



"صخرة الأساس"

BASIC ROCK

شرح لغة CSS من البداية إلى الاحتراف

تأليف: احمد حسونة عام 2019

نبذة مختصرة

هذا الكتاب يعد للقارئ بمثابة صخرة قوية ثابتة بفضل الله من الأساس في عالم الويب، مع لغة تنسيق الويب الساحرة CSS مع أهم تفاصيلها من البداية إلى الاحتراف بفضل الله، حيث تبدأ من الصفر وتعلم، ثم تتأسس، ثم تطبق، ثم تتعلم أكثر، ثم تطبق أكثر، ثم تتأسس وتعلم أكثر وأكثر، ثم تمارس أكثر وتستم في التطبيق، ثم تصل إلى الاحتراف بفضل الله

احمد حسونة

صخرة الأساس CSS

فهرس الموضوعان Subject Index

1.....	فهرس الموضوعان Subject Index
14.....	مقدمة أساسية - هامة جداً
18.....	هام جداً Very Important قبل الكتاب
19.....	رؤية الكتاب Vision of Book
20.....	قبل البداية في عالم CSS
21.....	محرر الكود المفضل Favorite Code Editor
23.....	ما هي لغة التصميم CSS
24.....	تأثير CSS على الصفحات
30.....	مختبر لغة CSS
31.....	وحدات القياس Measurement units
33.....	مكونات لغة CSS
34.....	بادئة المتصفحات Vendor Prefix
35.....	أهاك كتاب لغة CSS
36.....	أولاً: استخدام الطريقة inline style للعنصر.
37.....	ثانياً: استخدام الطريقة internal style للصفحة.
38.....	ثالثاً: استخدام الطريقة external style للملف.
40.....	الهدف Selector
41.....	الهدف الطبيعي Normal Selector
43.....	الهدف المعروف Id Selector
45.....	الهدف المصنف Class Selector
47.....	الهدف الكلي Universal Selector
49.....	الهدف بالخاصية Attribute Selector
59.....	الأهداف المتداخلة Combinators

60.....	عنصر داخل عنصر بأي طريقة
68.....	عنصر داخل عنصر بشكل مباشر
70.....	العنصر التالي
72.....	كل العناصر التالية أو أشقاء العنصر منه بعده
74.....	الكود الذي يغلب Active Code
78.....	الحجم أو الأبعاد Size or Dimensions
86.....	العوامش Margins
93.....	الحشو Padding
96.....	الاتجاه Direction
96.....	اتجاه العناصر direction
98.....	ضبط اليونيكود unicode-bidi
102.....	الإطارات Borders
117.....	اللون والخلفية Color and Background
117.....	لون الخلفية background-color
119.....	صورة الخلفية والتدرج background-image
131.....	تكرار الخلفية background-repeat
133.....	حجم الخلفية background-size
136.....	ارتباط أو تعلق الخلفية background-attachment
140.....	موضع الخلفية background-position
150.....	قصاصة الخلفية background-clip
153.....	مزج الخلفيات background-blend-mode
170.....	خاصية الخلفية background
172.....	تكوين إطار بالصورة border-image
177.....	استخدام الألوان Colors
182.....	خليط المزج mix-blend-mode
195.....	العزل isolation

197	المحتوي الزائد Overflow
200	خاصية المحتوي الزائد overflow
204	المحتوي الأفقي الزائد overflow-x
206	المحتوي الرأسى الزائد overflow-y
208	النصوص Texts
208	حجم الخط font-size
210	نوع الخط font-family
213	تقنيه الخط font- kerning
215	نمط الخط font- style
216	ثقل الخط font- weight
217	تنوع الخط font- variant
218	صناعة الخط بالمحدد @font-face
220	استيراد الخط بالمحدد @import
221	ضبط حجم الخط font-size-adjust
224	تمدد الخط font- stretch
227	طول السطر line- height
229	خاصية الخط font
230	محاذاة النص text-align
232	محاذاة آخر النص text-align-last
234	المسافة البادئة للنص text-indent
235	التفاف النص word wrap
242	مسافات بين الكلمات word-spacing
243	مسافات بين الحروف letter-spacing
244	خاصية النص الزائد text-overflow
246	خط ديكور النص text-decoration-line
250	لون ديكور النص text-decoration-color

251	text-decoration-style	شكل ديكور النص
253	text-decoration	خاصية ديكور النص
254	text-transform	تحويلات النص
256	text-shadow	ظل النص
262	hyphens	الواصلات
263	quotes	الاقتياس
264	tab-size	حجم رمز tab في النص
266	user-select	تكلم التحديد
268	writing-mode	طريقة الكتابة
270	hanging-punctuation	تعليق الترقيم
271	Display	العرض
272	visibility	ظهور العناصر
273	inline	عنصر على السطر
276	block	عناصر الكتلة
278	inline-block	عناصر مختلطة
279	change display	تغيير طبيعة العناصر
283	Images	الصور
283	Photo Exercises	تمارين الصور
291	Text with Images	النص مع الصور
297	filter	التأثيرات
322	object-fit	قص متناسب
323	object-position	موضع مع التحجيم
324	pseudo	الفئات والعناصر المستعارة
325	hover	اللمس
327	focus	التركيز
329	checked	الاختيار

331	empty	محتوي فارغ
333	first-child	أول عنصر فرعي
336	last-child	آخر عنصر فرعي
339	first-of-type	النوع الأول
343	last-of-type	النوع الأخير
347	nth-child	رقم محدد لكل فرع
351	nth-last-child	رقم محدد لكل فرع من الأسفل
353	nth-of-type	رقم محدد لكل نوع
355	nth-last-of-type	رقم محدد لكل نوع من الأسفل
357	only-child	الفرد فقط
359	only-of-type	النوع فقط
361	not	الكل ما عدا
364	disabled	التعطيل
365	enabled	التفعيل
366	read-only	القراءة فقط
367	read-write	غير مضبوط على القراءة فقط
368	required	المطلوب
369	optional	الغير المطلوب أو الاختياري
370	in-range	داخل المدى
371	out-of-range	خارج المدى
372	valid	تم التحقق منه
373	invalid	لم يتم التحقق منه
374	lang	اللغة
375	root	الأصل
376	target	هدف تم استهدافه
379	link	الرابط

380	visited	رابط تمت زيارته
381	active	الفاعلية
383	after	ما بعد
385	before	ما قبل
387	selection	شيء تم تحديده
390	first-letter	الحرف الأول
392	first-line	السطر الأول
394	Lists	القوائم
394	list-style-type	نوع القوائم
423	list-style-image	صورة القائمة
425	list-style-position	موضع القائمة
427	list-style	خاصية القائمة
429	Tables	الجدول
430		أولاً: أساس الجدول وبياناته وخلفية الصفحة
431		ثانياً: طمس إطار الجدول وعمل العرض بالكامل
432		ثالثاً: ضبط عناوين الجدول
433		رابعاً: ضبط الحشو للخلايا وعمل الإطار من الأسفل
435		خامساً: ضبط لون خلفية الصفوف الزوجية والفردية
437		أكواد تطور بناء الجدول
438		تصميم تطور بناء الجدول
440		جداول أخرى
475	caption-side	مكان عنوان الجدول
477	empty-cells	الخلايا الفارغة
478	table-layout	تنسيق الجدول
481	Position	الموضع
481		مفهوم الموضع

483	الموضوع الطبيعي والتغيير
485	أبعاد الحواف
490	أنواع الأوضاع
491	ظهور عنصر على آخر
493	ملخص مفهوم الموضوع
494	الوضع static
495	الوضع absolute
497	الوضع relative
499	الوضع fixed
507	الوضع sticky
510	التنظيم Float
510	معنى float
511	اتجاه العناصر نحو اليسار float left
514	اتجاه العناصر نحو اليمين float right
517	ال float مع العناصر الأخرى
522	عزل ال float ب clear
526	الصندوق box
526	ظل الصندوق box-shadow
538	تحجيم الصندوق box-sizing
542	ديكور السطر للصندوق box-decoration-break
547	الصندوق المنظم Box Model
548	أولاً: وضع المحتوى داخل div بدون أي تنسيق
549	ثانياً: لون النص والخلفية وحجم الخط
550	ثالثاً: العائمة والحشو ومحاذاة النص
551	رابعاً: الطول والعرض
552	خامساً: الإطار

553	سادساً : الإطار الخارجي
554	سابعاً : تظليل الصندوق والنص
556	مراحل تطور الصندوق
557	المرونة (Flex Box) Flexibility
557	الظهور المره display flex
561	محاذاة العناصر align-items
567	التفاف المحتوى flex-wrap
572	اتجاه المرونة flex-direction
576	تدفق المرونة flex-flow
579	محاذاة المحتوى align-content
591	ضبط المحتوى justify-content
597	محاذاة العنصر نفسه align-self
600	أساسه للمرونة flex-basis
603	ازدياد العناصر flex-grow
608	انكماش العناصر flex-shrink
612	الخاصية flex
613	ترتيب العناصر order
614	توسيط العنصر Centralize Element
615	توسيط العناصر Centralize Elements
616	عرض صور منظمة حسب الرغبة
620	المؤشر والعلامة Cursor and Caret
620	علامة الإدخال (الاقحام) في الكتابة caret-color
624	المؤشر cursor
627	حدث المؤشر pointer-events
628	التقطيع clipping
628	القصاصه clip

630	مسار القص clip-path
631	الأعمدة المتعددة Multi Columns
631	عدد الأعمدة column-count
633	الفارق بفة الأعمدة column-gap
634	فواصل الأعمدة column-rule
637	عرض العمود column-width
639	عنوانه مءمء للأعمدة column-span
642	ءعبئة الأعمدة column-fill
645	ءاصفة الأعمدة columns
646	النظام الشبكي Grid Layout
647	قالب الأعمدة grid-template-columns
651	قالب الصفوف grid-template-rows
653	قالب مناطق الخلايا grid-template-areas
664	القالب grid-template
668	مناطق الخلايا grid-area
670	فءوات بفة الخلايا grid-gap
676	صف البءاءة grid-row-start
678	صف النءاءة grid-row-end
682	الصف grid-row
684	عمود البءاءة grid-column-start
686	عمود النءاءة grid-column-end
690	العمود grid-column
692	ءلقائفة العمود grid-auto-columns
694	ءلقائفة الصف grid-auto-rows
696	ءلقائفة الءءفق grid-auto-flow
700	فءء ءءءم العناصء resize

703	scroll-behavior	أسلوب التمرير
705	Transform, Perspective	التحول والمنظور
706	2D	التحول ثنائي الأبعاد
706	rotate	تحول الاستدارة
708	rotateX	استدارة أفقية
710	rotateY	استدارة رأسية
712	skewX	الانحراف الأفقي
714	skewY	الانحراف الرأسي
716	skew	الانحراف الكلي
718	scaleX	المقياس الأفقي
720	scaleY	المقياس الرأسي
722	scale	المقياس الكلي
724	translateX	تحريك أفقي
726	translateY	تحريك أفقي
728	translate	تحريك كلي
730	matrix	مفهوم توزيعة التحول الكلي
732	matrix	عمل ال scaleX بواسطة
734	matrix	عمل ال scaleY بواسطة
736	matrix	عمل ال translateX بواسطة
738	matrix	عمل ال translateY بواسطة
740	matrix	عمل ال skewX بواسطة
742	matrix	عمل ال skewY بواسطة
744	matrix	عمل ال rotate بواسطة
748	matrix	عمل جميع التحولات بواسطة
750	Many Transform	أكثر من تحول
752	perspective	المنظور perspective والاستخدام الثلاثي الأبعاد 3D

- 758 perspective عمل منظور بالدالة
- 759 rotateZ تحول الاستدارة باستخدام
- 760 scaleZ التمدد المجسم
- 762 translateZ تحريك مجسم
- 764 Cube تكويب شكل مكعب ثلاثي الأبعاد
- 764 بداية تكويب المكعب
- 766 absolute ضبط الوضع الحر
- 767 ضبط الزوايا المناسبة
- 768 3D transform-style زيادة المشعة وضبط المنظور على
- 770 ضبط الإطار لمكونات المكعب الداخلية
- 772 عمل تحريك مجسم للكل بنصف قيمة كل مربع
- 774 rotate3d عمل استدارة ثلاثية الأبعاد
- 778 translate3d تحريك مجسم باستخدام
- 780 scale3d تمدد مجسم باستخدام
- 782 matrix3d تحويلات الـ 3D بواسطة
- 783 Transition الانتقال
- 784 transition الخاصة والمدة والتأخير مع الـ
- 792 transition-timing-function توظيف الوقت
- 798 cubic-bezier استخدام
- 804 transition الخاصة
- 806 Animation الحركة
- 807 @keyframes صناعة مؤثر الحركة بالمدد
- 809 تشغيل الحركة
- 811 animation-timing-function توظيف وقت الحركة
- 812 animation-delay تأخير الحركة
- 813 animation-iteration-count عدد التكرارات

814	animation-direction	توجيه الحركة
818	animation-fill-mode	نمط تتبع الحركة
820	animation-play-state	تشغيل وإيقاف الحركة
821	animation	خاصية التحريك
822	Infinite animation	الحركة اللانهائية
824	Animation	أمثلة على الـ
862	Tools	الأدوات
863	Text Box	مربع النص
865	Label	عنوان الشاشة
867	Buttons	الأزرار
870	Panel	اللوحة
871	Check Box	صندوق الاختيار
874	Radio	صندوق الراديو
877	Slider	شريط الاختيار
879	Combo Box	صندوق الاختيار
882	Toggle	صندوق المفتاح
885	Custom Counter	أداة العداد المخصص
888	styles.css	أكواد الملف
895		شاشة الأدوات
897	Scroll Bar	شريط تمرير الصفحة
903	Media Query	استعلامات الميديا
904	Media Query	البدء مع الـ
908	Media Query	تنوع الاستخدامات مع الـ
912		استخدام مع طبيعة الصفحة
914	Min and Max	تنسيق بينه أقصى وأقل للعرض
918	Inherit	الوراثة

921	تعرف المءعبراء Variables
924	ءمارفه عملفة practical examples
924	ءمرفه نص مءفور
925	ءعبفر شكل الرواء
926	عمل ءءول
928	قائمة رأسفة ءاءفة
930	قائمة ءانبفة والمءءو على الفاء
933	قائمة فف الأسفل
936	قائمة علوفة منبءة
939	مرفء نص بأفونة مء صورة
940	ءنسفق أءواء برمءفة ب CSS
942	الرواء المءفففة
944	ءءول بشرطف ءمرف وءلمفءاء
948	مرفء فءءرء وءءلوه
950	صفءة المءءءاء مع اسءءءاء صوره
957	مرفء فءقلب وءرءعش
959	ءنوفء رواء وازراء ومرفءاء نص
964	ءنوز وهءاءفا 
975	شءرء ءاصء
976	نشءرك على قراءءك
977	المراءء References
978	ءءاءم The End

مقدمة أساسية – هامة جداً

بسم الله الرحمن الرحيم والصلاة والسلام على أشرف المرسلين سيدنا محمد الذي أرسله الله رحمة للعالمين، وأشهد أن لا إله إلا الله وحده لا شريك له، الذي لم يتخذ ولدا ولم يكن له شريك في الملك وخلق كل شيء فقدره تقديرا ،،، ثم أما بعد...

الشيء الوحيد الذي أخبر الله رسوله محمد ﷺ أن يطلب فيه الزيادة هو العلم عندما قال الله سبحانه وتعالى في سورة طه (وَقُلْ رَبِّ زِدْنِي عِلْمًا) ، وأول ما نزل من القرآن الكريم (اقْرَأْ بِاسْمِ رَبِّكَ الَّذِي خَلَقَ (1) خَلَقَ الْإِنْسَانَ مِنْ عَلَقٍ (2) اقْرَأْ وَرَبُّكَ الْأَكْرَمُ (3) الَّذِي عَلَّمَ بِالْقَلَمِ (4) عَلَّمَ الْإِنْسَانَ مَا لَمْ يَعْلَمْ) ، ورفع الله من قدر وشأن العلماء حينما قال (يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنْكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ) ، وبه الله عز وجل أن العلماء لا يتساوون في القدر مع من لا يعلم فقال جل في علاه (قُلْ هَلْ يَسْتَوِي الَّذِينَ يَعْلَمُونَ وَالَّذِينَ لَا يَعْلَمُونَ إِنَّمَا يَتَذَكَّرُ أُولُو الْأَلْبَابِ) ، وأيضاً أخبرنا رب العزة سبحانه أن من يخشونه هم العلماء حين قال في كتابه العزيز (إِنَّمَا يَخْشَى اللَّهَ مِنْ عِبَادِهِ الْعُلَمَاءُ) ، وهو الذي علمنا عدد السنين والحساب حين قال (هُوَ الَّذِي جَعَلَ الشَّمْسُ ضِيَاءً وَالْقَمَرَ نُورًا وَقَدَرَهُ مَنَازِلَ لِتَعْلَمُوا عَدَدَ السِّنِينَ وَالْحِسَابِ مَا خَلَقَ اللَّهُ ذَلِكَ إِلَّا بِالْحَقِّ يُفَصِّلُ الْآيَاتِ لِقَوْمٍ يَعْلَمُونَ) ، وكما نرى أن العلم قدرة عالي ولا يطلبه إلا غالي أدركه علاه.

ولقد عجزت يازن الله تعالى أن أقدم أفضل ما لدي من علوم في مجالات الحاسب الآلي (الكمبيوتر)، لأنه يفيد في جميع مناحي الحياة، فإذا تحدثنا عن شركة تجارية فهي تستخدم الكمبيوتر لتدوين وحساب جميع أعمالها من إنتاج وتسويق ورواتب وغيرها، وإذا تحدثنا عن مدرسة أو كلية أو معهد، فجميعها تستخدم الكمبيوتر في أعمالها وتظهر نتائج الامتحانات والجدول، بل وبعض المؤسسات التعليمية الحديثة تفتح الطالب ببرامج ذكية متطورة، وفي الهيئات الحكومية يتم استخدام الكمبيوتر أيضاً، وأصبح الحاسب يفعل كل شيء في حياتنا فلقد صعد الإنسان في الآفاق والفضاء بالحاسب، ونشاهد به الفيديو ونشغل به الصوت ونطبع به الوثائق والمستندات ونصور به الصور والأوراق وندير به كافة المجالات، ولا تقتصر على هذا فحسب، بل ولو قلت لك أن الحاسب قد يغير في حياة أسرة بأكملها وينجيها بفضل الله، سوف تتعجب وتقول كيف؟! - والجواب هو كم من رجل كان على حافة الانهيار المادي والمعنوي بسبب فقده لوظيفته، ففتح الله عليه بتعلم الحاسب ورزقه الله بعمل جيد ورزق وفير، وكم من شاب كان على حافة الانحراف الأخلاقي بسبب الأخلاق السيئة المتداولة بين الشباب إلا من رحم ربي، فمع الله عليه بأن بدء في تعلم مجالات الحاسب فأحبها وبدء يبني مستقبله الحقيقي.

المحتوى الذي سوف تتعلمه يتكون من مواضيع تتبع بعضها بعض، ومنظمة بطريقة تسلسلية بحيث تأخذ بيدك خطوة بخطوة من البداية إلى الاحتراف بفضل الله، وسوف أتبع نظام في شرح هذا المحتوى مثل صعود الطائرة، ففي البداية يتم التنبيه على الركاب ببعض الإرشادات بالتفصيل الدقيق، ثم

في بداية الإقلاع يكون الأمر صعباً ثم نظير إلى أعلي ونخلق في الفضاء، وسوف نبدأ بإذن الله بالتفصيل ثم نعلوا.. ونعلوا ، ثم نخلق ونحتف بإذن الله، فإنني أنصح القارئ أن يهتم بجميع الموضوعات مرتبة، فعسى كلمة أن تغير بفضل الله، والمحتوي تم تأليفه بكامل الحب والإخلاص لجميع القراء، ولك يأخذ نسخ ولصق، بل وحتى المحتويات التاريخية الموجودة داخل المحتوى تم صياغتها بأساليب مختلفة من أكتف من مكان معلوماتي، والمؤلف وضح كل خبرته بكل الحب لجميع أحيائه القارئ لهذا المحتوى، والكمال لله وحده، ونحده لا نعلم إلا قليلاً كما قال الله في القرآن (وَيَسْأَلُونَكَ عَنِ الرُّوحِ ۗ قُلِ الرُّوحُ مِنْ أَمْرِ رَبِّي وَمَا أُوتِيتُمْ مِنَ الْعِلْمِ إِلَّا قَلِيلًا).

وفي الختام أسأل الله تبارك وتعالى بأسمائه الحسني وصفاته العلا أن يفيد جميع القراء بهذا المحتوى، ويغير في حياتهم إلى الأفضل إن شاء الله. كما أحب أن اشجع الجميع على العطاء في العلم، فإذا كنت تعلم أي معلومة لا تبخل على غيرك بها أبداً، وتأكد أن الله سوف يكفك بذلك ويمه عليك بمعلومات أفضل بكثير من الذي لديك، ولقد أمرنا رسول الله كامل الأخلاق أن نحب الخير لإخواننا كما نحب أنفسنا حبه قال (حب لأخيك ما تحب لنفسك)، وهذا الموضوع هام للغاية لأنه ينهض بالمجتمع ككل، فإذا تعلمت شيء، وعلمته لأخيك أو زميلك فسوف يأخذ فلكمك ثم يبتدع عليها ابتكاراً مختلفاً من ابتكارك، لأن الله خلق البشر بعقول ومميزات مختلفة، ثم يعلم زميلك معلوماتك ومعلوماته لزميل آخر، فببتدع ابتكاراً ثالثاً مختلفاً عن الأول والثاني، وهكذا، ولو حدث هذا على مستوى المجتمع

نعض المءءمء وءءءن اءءكاراءه واءءءاعاءه، ولا أقصء مءال الءاسب الآلى فءسب، لا، بل أقصء ءمفع المءالاء فف ءمفع مئاءف الءفاء، وءا ءابلء شءص ءبلء مئء معلومة وءءل علفء، ثم بعء ذلك ءلب مئء معلومة لا ءفعل مئءة، بل اعءفاء المعلومة على الفور ولمء له أن هءا هو الصءفء، وسوف يؤءر هءا علففة وبعفر ءالة إلى العءاء، وهءا ءءء معف شءصفاً أءء مء مءة، وأءبءء لف ءءءربة أن هءا هو الصءفء.

ءما أءب أن انه أنه بعفر مسموء لأي فرء أو شركة أو هفئة أو مئظمة أو مؤسسة أو ءءاع مء أي نوع أن ءءءء أو ءءاءر بعءا المءءوءف بأف نوع أف ءاه، أو بعفر ءءابءه، أو ءنسخه بعفر ءق لبعفر الاسم، أو أف عملفة ءءوئل ءءاب pdf ال أف نوع أو ءءءءل علفه، وله اسامء أف شءص ءسبب فف ذلك، وله أءر ءه فوم ءفاءة، وعءء الله ءءءمء الءصوم. وسوف ءءء شرح ففءو لهءا المءءوءف بالءفصءل مءانا للمشاهءة فءق على أكاڤمفة حسونة فف الموقع أو الففءوئوب على الرابءفه ءالفةه:


www.hassouna-academy.com

www.youtube.com/user/hassounaacademy

والسلام علفكم ورحمة الله وبرءاه.

المؤلف: اءمء حسونة




هام جداً Very Important قبل الكتاب

أمانة - **تمه الكتاب**، قراءة هذه الصفحة جيداً وصفحة نشرك على قراءتك وصفحة مقدمة أساسية وتليبه ما فيهم  فضلاً منكم وليس أمراً.



موقع حسونة أكاديمي ← <https://www.hassouna-academy.com>

قناة حسونة أكاديمي ← <https://www.youtube.com/user/HassounaAcademy>

بفضل الله، القناة عليها شرح كمبيوتر، وويندوز، ولينكس، وماك، وورد،
واكسل، وأكسس، وباور بوينت، ولغات برمجة كثيرة، وويب بتقنيات كثيرة،
وأمثلة كثيرة، وتطبيقات كثيرة، وموبايل، وأندرويد، وقواعد بيانات، ونظم،
وتصميم، وتحليل بيانات، وتحليل نظم، ومناهج دراسية، ووسائط متعددة،
ودردشة برمجة، وبرنامح مبيعات، وبرنامح مطامح، ومعلومات متعددة،
وقصص نجاح، وبرامح جاهزة، وفوتوشوب، وشرحات كثيرة بفضل الله.
رجاء! لا تنسي أن تشترك بالقناة  وفعل زر الجرس في القناة 
وسجل حساب على موقع حسونة أكاديمي  الآن.

يسعدنا انضمامك

<https://www.hassouna-academy.com/>

رؤفة الكءاب Vision of Book

- الءامل مع إصءارات CSS وءأنهم اصءار واءء
- الءوءفه قبل البءافة
- ءأسفس قوي فف CSS (ءأفر، وءءاء، مءوناء، ءءابة، اسءءءاء، هءءف)
- النصوص والءءم والصور والءشو والءوامش والاطارات
- الألوان والءلففاء والمءءوئ الزاء والنصوص والءرض
- الصور ثم الفءاء المسءءارة
- القواء والءءاول
- الموءء والءنظفم والمروءة
- الاعمءة والنظام الشبئف
- الءءول والمنظور
- ءءم الاءءقال والءرءة
- أءواء إءفاففة
- اسءءلاماء المفءفا
- مواءء إءفاففة وءمارف عملفة

قبل البداية في عالم CSS

قبل البدء في دراسة هذا الكتاب، يجب أن تكون على دراية كافية بعالم نظام التشغيل مثل، Windows أو Linux أو Mac، ثم همام جداً أن تتعلم كتاب "صخرة الأساس" في اللغة HTML قبل البدء في هذا الكتاب، ويمكنك العثور عليه من الرابط التالي:

<https://www.hassouna-academy.com/books>

ثم بعد أن تذاكر كتاب "صخرة الأساس" في اللغة HTML، تستطيع الدخول وبقوة في اللغة الرائعة CSS، بفضل الله. كتاب "صخرة الأساس" في اللغة HTML، يحتوي على موضوعات اللغة الأم للويب HTML بالتفصيل، ويشرح الوسوم والخصائص، بالتفصيل، ويبيد المفاهيم بوضوح، ويجعلك تدخل في عالم الويب من باب واسع، وسهل، ويشرح معك في الأمثلة العملية، حتى تفهم الغرض من كل وسم، وخاصة، ومثال، لذلك ننصح بكتاب "صخرة الأساس" في HTML.

انتبه! الآن يجب أن تعلم أنه أثناء دراسة هذا الكتاب، يجب معرفة كل ما يتعلق باللغة الأم للويب وهي اللغة HTML.

سوف نعمل على نفس الملف الذي تم استخدامه في كتاب "صخرة الأساس" للغة html، وهو الملف test.html.

محرر الكود المفضل Favorite Code Editor

قد سبق وتحدثنا في كتاب HTML "صخرة الأساس" عن محرر النصوص العادي Notepad، الموجود في ويندوز، أو ما يعادله في أنظمة التشغيل الأخرى مثل لينكس وماك، وتمت البداية على هذه المحررات التقليدية، لجعلك أكثر احترافية واعتمادية، لأنك تكتب الأكواد بنفسك، وبدون أي مساعدة، وبداية من الآن سوف نقوم بذكر محررات أكواد مختلفة، ومخصصة، وذكية، ومساعدة، ومريحة للعين، وأكثر كفاءة، والعمل عليها بسرعة وسهولة، لأنها تملك الأكواد بشكل تلقائي.

يوجد محررات أكواد كثيرة جداً يمكنك استخدامها بسهولة، مع العلم أن كل محرر كود مختلف عن الآخر، مثلاً برنامج **Notepad++**، أو برنامج **Atom**، أو برنامج **Komodo**، أو برنامج **Brackets**، أو برنامج **Sublime Text**، أو برنامج **Text Mate**، أو برنامج **Vim**، أو برنامج **Visual Studio Code** إلى آخر هذه البرامج الكثيرة، والتي أغلبها مجانية، ومفتوحة المصدر، وأيضاً مدعومة على أنظمة Windows وLinux وMac.

انتبه! ننصح عند بداية تعليمك ودراستك، ألا تعتمد كثيراً على كلمة هذه المحررات، وتحاول كتابة الأكواد كاملة بنفسك مرات متعددة، حتى تتعود وتحترف، ثم بعد ذلك، لا بأس أن تستخدم مساعدات المحررات المختلفة.

تحميل المحررات من الإنترنت وتنزيلها للنظام، بفضل الله سهل جداً،
 فيملائك التأكد من أن جهازك موصول بالإنترنت، ثم تابع الخطوات التالية:

- ادخل على موقع جوجل www.google.com.
- اكتب في خانة البحث كلمة "download"، ثم مسافة، ثم اسم البرنامج الذي تريد تحميله، مثلاً download brackets.
- ابحث عن رابط الموقع الأصلي للبرنامج، وغالباً يكون أول نتائج البحث، ثم ادخل عليه.
- ابحث داخل موقع البرنامج عن التحميل لنظام التشغيل الذي تستخدمه، ثم اضغط على زر "التحميل" أو "download"، لبداية التحميل.
- بعد انتهاء التحميل، قم بتشغيل الملف المسؤول عن تنزيل البرنامج، بطريقة نظام التشغيل الذي تعمل عليه، والتي من المفترض أنك تفهمها جيداً، ثم اتبع خطوات التنزيل مثل ازرار "next" او "install".
- افتح البرنامج، ثم اقرأ المساعدات والاستخدام والتلميحات، ثم استخدمه كثيراً لتعود عليه.

وعن طريق الخطوات السابقة تستطيع العمل على أي محرر تريده، مع العلم أن كل محرر له البيئة الخاصة به، ويوجد محررات تقبل منك إضافة حزم اضافيه بها لدعم لغات وكلمات أكثر.

ما هي لغة التصميم CSS

تم انشاء اللغة CSS، وتطويرها بعد ذلك، لتحسيه تصميم صفحات الويب والتأثير الأفضل عليها من حيث، النصوص، والمحاذاة، والخطوط، والألوان، والخلفية، والاطارات، والصور، والشفافية، والهوامش، والحشو، والروابط، وأشربة التنقل، والقوائم، والجداول، والأدوات، والمواضع، والتنظيم، والعرض، والدقة، والحجم، والتغيير، والتحرك، وسرعة الحركة، واختلاف الشاشات، وتقسيم الصفحات، وأشياء أخرى كثيرة.

الحروف CSS اختصار للكلمات Cascading Style Sheets ومعناها أوراق النمط المتتالي، ولكنه يملكه أن نبتعد قليلاً عن الترجمة الحرفية، ثم نقول إن معناها "صفحات الأنماط الانسيابية"، حيث أن لغة التصميم CSS تؤثر على صفحات الويب لتكون أكثر جمال وانسيابية.

اللغة CSS هي لغة تصميم، وليست لغة برمجة، حيث إن لغات البرمجة لها طقوس عمل واضحة، مثل تعريف المتغيرات وجمل الاختيار وحلقات التكرار والمصفوفات، إلى آخر هذه المفاهيم، والتي لا توجد في لغات التصميم.

عند كتابة ملف به أكواد لغة التصميم CSS، يأخذ الامتداد CSS. ثم يتم النداء عليه، كما سوف نتعلم فيما بعد إن شاء الله.

تأثير CSS على الصفحات

عندما تعمل باللغة CSS سوف يكون لها تأثير واضح ورائع على صفحات الويب، وهام جداً أن تعرف هذا التأثير حتى تعرف أهمية اللغة، ولتوضيح هذا التأثير لك، سوف نتكلم في البداية عن أمثلة واقعية من أرض الواقع، بالشرح والصور بفضل الله، وبعدها سوف تعرف قدر اللغة CSS.

- تخيل أنك تريد بناء البيت من الصفر، سوف تشتري الطوب والأسمت ومواد البناء لتأسيس البيت، وبعدها سيكون شكل البيت ليس كما تحب، فتسرع وتأتي بالنقاش ومواد الطلاء والسيراميك والديكور وغيرها، فيصبح شكل البيت رائع.

- أو تخيل مصنع السيارات عند تصنيع السيارة، يأتي بالحديد والأسلاك وغيرها ليتم تأسيس السيارة، ولكن سيكون شكل السيارة غير ماألوف لك، فيقوم المصنع بإحضار دهانات السيارة وطلاءها وتركيب الديكور لها، فيصبح شكل السيارة أكثر من رائع.

- أو تخيل مصنع الهاتف عند تصنيع الهاتف، يحضر اللوحات الإلكترونية والأسلاك وغيرها لتصنيع أساس الهاتف، فيكون شكل الهاتف لا يعجبك، فيأتي المصنع بالغطاء والديكورات، فيتم تغطية الهاتف، فيصبح شكل الهاتف ماألوف وجميل ورائع.

تخيل وركز جيداً فيما يلي:

1- بناء يتم تأسيسه، ثم تجهيزه:



صورة بداية تأسيس البناء، وكأننا نعمل على HTML للأساس



صورة البناء بعد الديكورات، وكأننا نعمل على CSS للتحسين

2- صناعة وتجهيز السيارة:



صورة بداية تأسيس السيارة، وكأننا نعمل على HTML للأساس



صورة السيارة بعد الديكورات، وكأننا نعمل على CSS للتجسيه

3- صناعة وتجهيز الهاتف



صورة بداية تأسيس الهاتف، وكأنا نعمل على HTML للأساس



صورة الهاتف بعد الديكورات، وكأنا نعمل على CSS للتحسين

بعد أن تم تخطيط الفكرة من الواقع، الآن تخطيط في عالم الويب، عن طريق عرض صفحة بدون CSS، ثم عرضها بلغة CSS كما يلي:

الشكل التالي يعرض صفحة ويب بدون تنسيقات CSS كما يلي:

Page
— □ ×

Full Name

Address

Salary

Number	Employee Name	Address	Salary
111	Mahmoud Samir Fayed	Cairo	7500
222	Adel AbdelSabour Ahmed	Giza	8500
333	Ahmed Mahmoud Hassouna	Cairo	7500
444	Amr ElSayed AbdElazez	Giza	4900
555	Ahmed Samy	Alex	7500
666	Eslam Adel Elnoby	Cairo	7500
777	Mahmoud Amr	Alex	7500

لاحظ أنها صفحة تم تأسيسها باللغة HTML فقط، ولاحظ وجود العناصر بدون أي ألوان أو تأثيرات أو تنسيقات.

الشكل التالي يعرض نفس صفحة الويب السابقة، بعد أن تم تحسيده العرض وتنسيقها باللغة CSS، وبدون أي تغيير في html كما يلي:

Page

Full Name

Address

Salary

Submit

Number	Employee Name	Address	Salary
111	Mahmoud Samir Fayed	Cairo	7500
222	Adel AbdelSabour Ahmed	Giza	8500
333	Ahmed Mahmoud Hassouna	Cairo	7500
444	Amr ElSayed AbdElazez	Giza	4900
555	Ahmed Samy	Alex	7500
666	Eslam Adel Elnoby	Cairo	7500
777	Mahmoud Amr	Alex	7500

مقدمة لغة CSS

تم تطوير لغة التصميم CSS من منظمة World Wide Web وهي منظمة شبكة الويب العالمية المشهورة، على يد اثنين من علماء الكمبيوتر، الأول هو "هاكون فيوم لاي" الرئيس التقني في برمجيات أوبرا، والثاني هو "بيرت بوس" عالم الكمبيوتر ومؤسس المتصفح Argo، كما يملك أن نذكر أن مطور هذه اللغة هو فريق CSS، وأول ظهور لهذه اللغة يوم 17 ديسمبر عام 1996م.

الفريق الذي يشترك في تطوير لغة التصميم CSS هم مجموعة كبيرة من المطورين، من الشركات المختلفة، ويمكنا الاطلاع على أسمائهم والشركات التابعة لها، من خلال الرابط التالي:

<https://www.w3.org/Style/CSS/members>

ودائماً يتم تطوير هذه اللغة الرائعة لتشمل تفاصيل أكثر، وتجعل صفحة الويب أكثر روعة.

وحدات القياس Measurement units

عند التعامل مع اللغة CSS نحتاج الى بعض وحدات القياس، للتعبير عه الذي نريه، مثلاً اريد ان أقول لك "طول عمر"، أنت الآن تنتظر الإجابة، وهي مثلاً "60 سننيمتر"، أو "70 سننيمتر"، ومثال آخر، وهو لو قلت لك، "وزن التفاح"، نعم أنت أيضاً الآن تنتظر الإجابة، وهي مثلاً أن أقول "5 كيلو جرام"، أو "7 كيلو جرام"، إلى آخر هذه الأمثلة المعروفة لدينا جميعاً، ولكنه اسأل نفسك سؤال، وفكر فيه، • هل لو قال لك شخص أنه يوجد شيء اسمه كيلو جرام، وشرحه لك بالتفصيل، سوف تفهم أكد، أم تفهم أكد لو تعاملت بهذا الكيلو جرام في البيع والشراء؟! •، بالتأكيد الإجابة هي الفهم مع التعامل، وكذلك وحدات CSS بعد الأمثلة والممارسة الكثيرة تفهم الوحدات أكد بفضل الله.

والآن سوف نتعرف على وحدات القياس التي نستخدمها في لغة التصميم التفصيلية CSS ومعني كل وحدة قياس.

وقد يتساءل البعض، لماذا سيتم معرفة وحدات القياس الآن؟!، والإجابة بفضل الله بسيطة جداً، وهي، تم ذلك لتكون معروفة لدي القارئ مع البداية، بحيث إذا تم استخدامها أمامك لا تكون غريبة عليك، فبتم فهم الكود بسهولة، ولا تشعر بالتعقيد.

الجدول التالي يوضح وحدات القياس ومعناها:

الوحدة	المعنى
الوحدات العادية (ثابتة)	
px	البيكسل pixel وهي نقطة على شاشة الكمبيوتر
cm	السنتمتر centimeter وهو مكون من 10 ملليمتر
mm	الملليمتر millimeters وهو 1 من 10 من السنتمتر
in	الإنش inch أو البوصة وهو يساوي 2.54 سنتمتر
pt	نقطة point، وتساوي 1/72 بوصة، و 1.33 بيكسل
pc	بيكا pica، وتساوي 1/6 بوصة، و 15.96 بيكسل، و 12 نقطة
الوحدات النسبية (تغير نسبياً)	
em	نفس الحجم الحالي، أي 2 الضعيف، و 3 ثلاث أضعاف، وهكذا
rem	مثل em ولكنها تأخذ من ال-root أي من الحجم الأصلي
ex	هو x-height، أي حجم الطول
ch	يقيس على أساس عرض نقشة الحرف صفر "0" للخط الحالي
vh	تساوي 1/100 بيكسل من قيمة إطار عرض الطول viewport
vw	تساوي 1/100 بيكسل من قيمة إطار عرض العرض viewport
vmin	هي 1/100 بيكسل من قيمة الأقل قيمة مقارنة بين الطول والعرض
vmax	هي 1/100 بيكسل من قيمة الأكبر قيمة مقارنة بين الطول والعرض
%	نسبة مئوية من الحجم الحالي، أي 200% الضعيف، وهكذا
وحدات أخرى (ربما نحتاجها)	
dpi	هي dots per inch أي عدد النقاط لكل بوصة 72 أو 96
dpcm	هي dots per centimeter أي عدد النقاط لكل سنتمتر
dppx	هي dots per px أي عدد النقاط لكل وحدة بيكسل
deg	هي الدرجة degree، مثل درجة الزاوية

مكونات لغة CSS

لغة التصميم CSS تتكون من شئيين، الخاصية **Attribute**، والقيمة **Value**، التي تأخذها هذه الخاصية، مع العلم أن تخصيص القيمة للخاصية لا يتم باستخدام الرمز = كما هو معتاد، ولكنه يتم باستخدام الرمز colon : للتخصيص، مثل الكود التالي:

```
color: red
```

وكما نلاحظ أنه تم تخصيص اللون الأحمر red لخاصية اللون color عن طريق الرمز : أي وكأننا نقول Color=Red، ولكنه مع لغة CSS تأتي السهولة، لتكون color: red بكل بساطة، أما لو أردنا أن نكتب خصائص كثيرة باللغة CSS نفصل بين كل خاصية وقيمتها بالرمز semicolon ; ; للكتابة خصائص متعددة، مثل الكود التالي:

```
color:red;width:5px;height:9px;
```

وكما نلاحظ أنه تم تخصيص اللون الأحمر red لخاصية اللون color، ثم فاصل ; ثم تخصيص القيمة 5 بيكسل لخاصية العرض width، ثم فاصل ; ثم تخصيص القيمة 9 بيكسل لخاصية الطول height، ثم فاصل أخير ; اختياري.

كما يمكنه أن تأخذ الخاصية أكثر من قيمة، مثل الكود التالي:

```
border:1px dotted red;
```

وكما نلاحظ خاصية الإطار تأخذ 1 بيكسل، ونمط خط منقط، واللون الأحمر، وكل هذا تم تخصيصه مرة واحدة عن طريق رمز كولين : واحد.

بادئة المتصفحات Vendor Prefix

تعلمنا بفضل الله أن تكون اللغة CSS، الأساسي عبارة عن خاصية attribute، وقيمة لها value، والآه **انتبه!** لأنه يوجد أصل هام نريد أن ننوه عليه قبل الشروع بالتفصيل في اللغة CSS، وهو بادئة المتصفح، أي أن كل متصفح له بادئته خاصة به، وعلى سبيل المثال، البادئة **-webkit-** لكل من Chrome, Safari, newer versions of Opera, almost all iOS browsers (including Firefox for iOS)، ومعني ذلك متصفحات جوجل كروم، وسفاري، والحديث من اوبرا، و iOS، والبادئة **-moz-** للمتصفح Firefox، والبادئة **-o-** لأوبرا القديم، والبادئة **-ms-** للمتصفحات (Internet Explorer, Microsoft Edge).

البادئة يتم وضعها قبل اسم الخاصية، في بعض الأحيان، وأحياناً قبل القيمة، ولكن **انتبه!** جيداً فلا يتم استخدامها في كل الخصائص والقيم، بل هي بعض الاستخدامات فقط، وركز جيداً في هذا حتى لا تختلط عليك الأمور، بل وحتى يوجد بعض الخصائص لها بادئة، وتعمل بدون كتابتها، وهناك بعض الخصائص والقيم، التي لا تعمل إلا بالبادئة، وهنا في هذا الكتاب، سوف يكون أكثر العمل على chrome، وله نستخدم البادئة إلا للضرورة فقط، فمع الآه **انتبه!** أنك لو رأيت **-webkit-** أو **-moz-** أو **-ms-** أو **-o-**، قبل أي خاصية، أو أي قيمة، فاعلم أنها بادئة وليست مع أصل الاسم أو القيمة.

أماكه كتابة لغة CSS

يتم كتابة أكواد لغة التصميم CSS بداخل أكواد html ليتم التأثر على عناصرها بشكل مباشر، ويمكك كتابة لغة CSS في ثلاث أماكه مختلفه كما يلي:

- 1- **بالعنصر نفسه، للتأثير عليه فقط Inline Style**
يتم استخدام خاصية html التي تسمى style لأي عنصر، لتوجيه تنسيق له، عن طريق كتابة أكواد CSS بهذه الخاصية.
- 2- **بالصفحة نفسها، للتأثير على عناصرها Internal Style**
يتم استخدام وسم html الذي يسمى style ، لتوجيه تنسيق لجميع عناصر الصفحة، عن طريق كتابة اسم العنصر ثم أقواسه مجموعة { } ثم كتابة أكواد CSS بداخل هذه الأقواس.
- 3- **بملف خارجي، والنداء عليه في الصفحات External Style**
يتم كتابة أكواد اللغة CSS في ملف نصي خارجي بنفس طريقة internal style، وبدون أي وسوم أو خصائص html، وهذا الملف يأخذ الامتداد CSS، ثم يتم النداء عليه بعد ذلك داخل أي صفحة من صفحات الموقع، عن طريق استخدام وسم HTML الذي يسمى link.

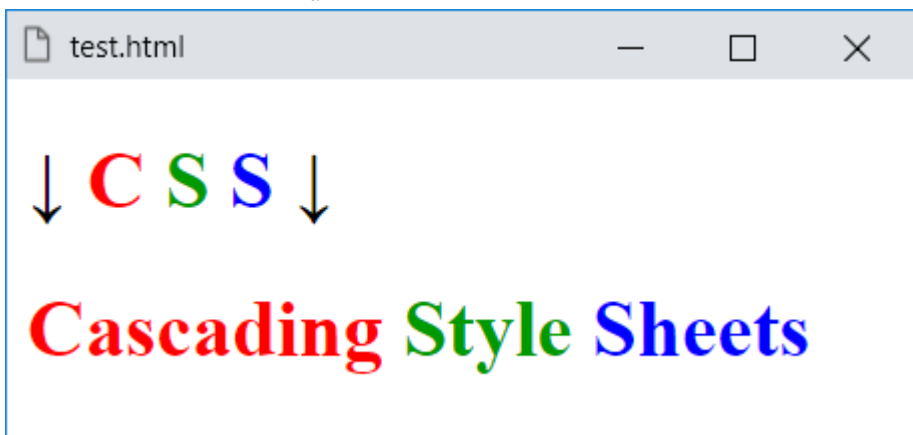
تم ذكر وسم style و link في كتاب "صخرة الأساس" HTML.

أولاً: استخدام الطريقة inline style للعنصر.
كود! قم بكتابة الكود التالي داخل العنصر body ثم احفظه:

```
<h1>
  &#x2193;
  <span style="color:#f00">C</span>
  <span style="color:#090">S</span>
  <span style="color:#00f">S</span>
  &#x2193;
</h1>

<h1>
  <span style="color:#f00">Cascading</span>
  <span style="color:#090">Style</span>
  <span style="color:#00f">Sheets</span>
</h1>
```

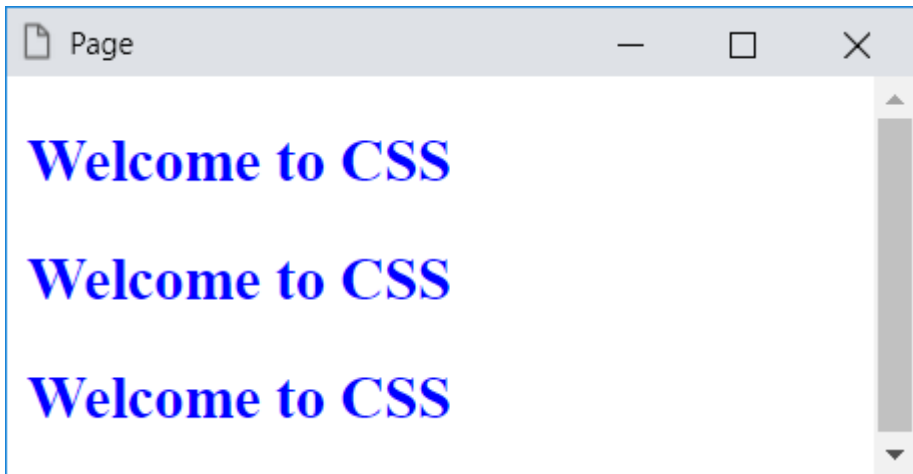
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



ثانفاً: اسءءءام الطرفة internal style للصفءة.
ءوء! قم بءءابة الكوء الءالف ثم اءفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page</title>
    <style>
      h2{
        color:blue;
      }
    </style>
  </head>
  <body>
    <h2>Welcome to CSS</h2>
    <h2>Welcome to CSS</h2>
    <h2>Welcome to CSS</h2>
  </body>
</html>
```

وبءء ءشءفل الصفءة ءظهر كما بالشءك الءالف:



ثالثاً: استخدام الطريقة external style للملف.

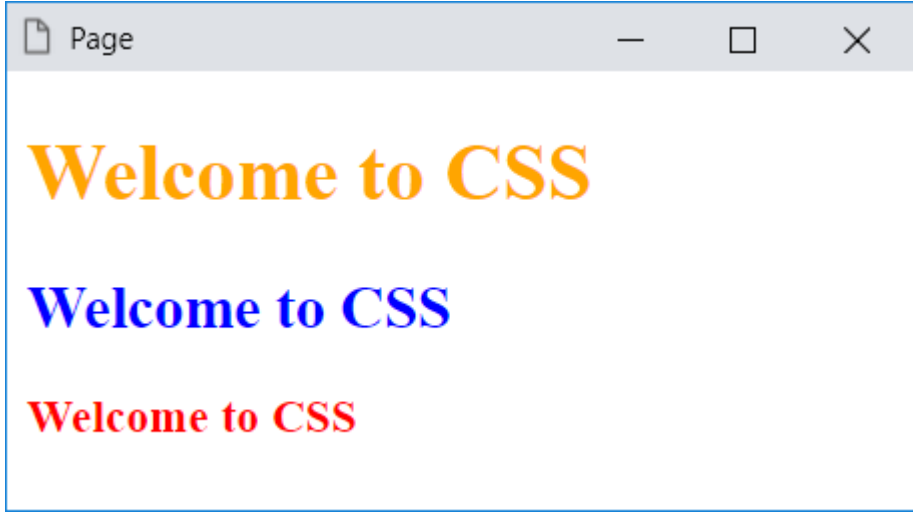
قم بفتح أي محرر أكواد، ثم أنشئ عليه ملف CSS جديد، بنفس مكان ملف التجربة باسم mystyle.css، ثم اكتب به الكود! التالي واحفظ:

```
h1{
  color:orange;
}
h2{
  color:blue;
}
h3{
  color:red;
}
```

كود! ثم قم بكتابة الكود التالي في ملف التجربة واحفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page</title>
    <link rel="stylesheet" href="mystyle.css">
  </head>
  <body>
    <h1>Welcome to CSS</h1>
    <h2>Welcome to CSS</h2>
    <h3>Welcome to CSS</h3>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



انتبه! لو أنشأت صفحة html جديدة في نفس المكان، ثم وضعت الوسم link بداخل الوسم head بها، سوف تتأثر نفس هذا التأيد، وتلويح الوسوم h1 و h2 و h3، مع العلم أن خاصية الارتباط rel للوسم link يتم ضبطها على القيمة stylesheet، وخاصية href للوسم link يتم ضبطها على مسار ملف التنسيق CSS.

الهدف Selector

المقصود بالكلمة selector هنا، هو العنصر المستهدف للتنسيق عليه.
انتبه! فإن هذا المفهوم هام جداً أكثر مما تتخيل، لفهم استخدامات لغة CSS، كما أنه سيساعدك لاستخدامات كثيرة فيما بعد بفضل الله.

تخيل معي أنك دخلت متجر كبير للحلوى، ثم أحبيت أن تشتري صنف معين،
ماذا تفعل؟، طبعي أن تفعل التالي:

- تسأل أحد عمال المتجر عن الصنف
- الذهاب الى مكان الصنف
- تسأل بائع الصنف عن النوع
- تشتري النوع من الصنف المقصود

كذلك ال selector داخل اللغة CSS، ويمكنه افتراض ما يلي:

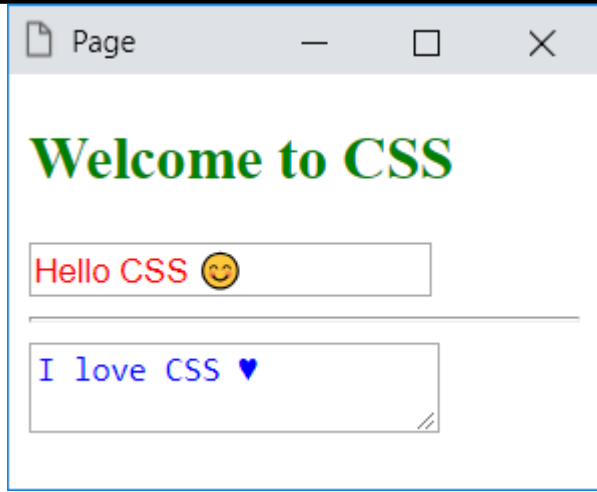
- تحديد مكان الكتابة
- تحديد العنصر أو العناصر
- معرفة كيفية التعامل معه
- تكتب الأكواد للتأثير عليه

الهدف الطبيعي Normal Selector

Normal Selector أو Type Selector، وهو أنه تكتب اسم الوسم tag بشكل صريح، كما تم استخدامه في طريقة internal style وطريقة external style، ثم التأيد عليه. **كود!** قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page</title>
    <style>
      h2{
        color:green;
      }
      input{
        color:red;
      }
      textarea{
        color:blue;
      }
    </style>
  </head>
  <body>
    <h2>Welcome to CSS</h2>
    <input value="Hello CSS &#x1F60A;"><hr>
    <textarea>I love CSS &hearts;</textarea>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



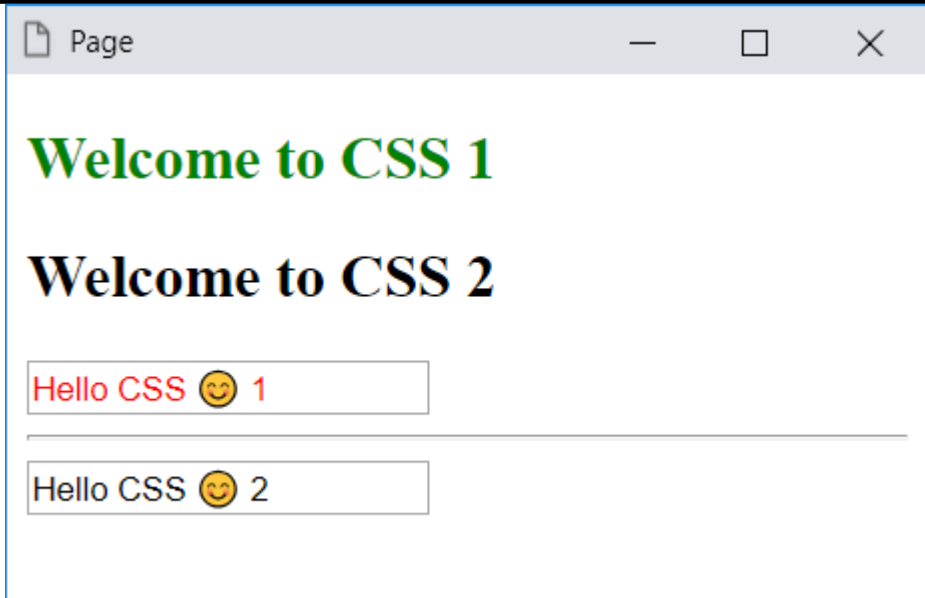
لاحظ أنه تم النداء على عناصر محددة بالاسم الطبيعي لها، بدون أي تغيير
لأسماء الوسوم، وهذه الوسوم هي `h2` و `input` و `textarea`،
وهذا بكل بساطة هو ال `Normal Selector`.

الهدف المعرف Id Selector

هو ان يتم إنشاء عنصر html ثم إعطاؤه id مخصص له كما تريد،
ثم بعد ذلك يتم توجيه تأثيره معك على هذا الـ id بشكل مباشر، كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page</title>
    <style>
      #my1{
        color:green;
      }
      #txt1{
        color:red;
      }
    </style>
  </head>
  <body>
    <h2 id="my1">Welcome to CSS 1</h2>
    <h2 id="my2">Welcome to CSS 2</h2>
    <input id="txt1" value="Hello CSS &#x1F60A; 1">
    <hr>
    <input id="txt2" value="Hello CSS &#x1F60A; 2">
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم استخدام الرمز # ثم وضع اسم ال id، للتأثير على العنصر الذي يحمل نفس اسم ال id.

لاحظ أنه لم يتأثر إلا العناصر التي تحمل نفس اسم ال id المقصود فقط، وهنا تم التأثر على id باسم my1 وآخر باسم txt1.

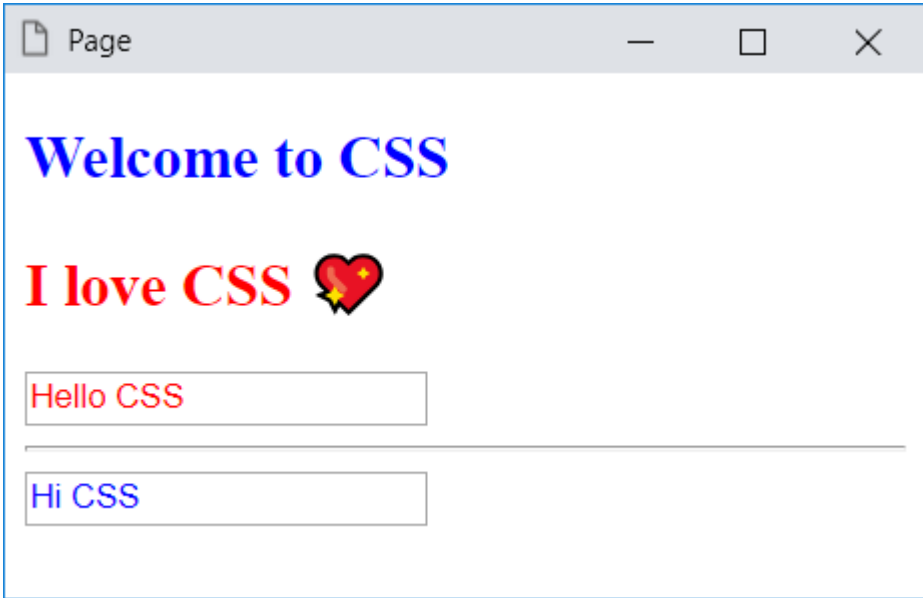
الهدف المصنف Class Selector

وهو أن تكتب أكواد CSS داخل اسم كلاس معين تسميه كما تريد، ثم بعد ذلك تخصص اسم الكلاس إلى خاصية html التي تسمى class لأي عنصر، فيتم التأثير عليه.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page</title>
    <style>
      .class1{
        color:red;
      }
      .class2{
        color:blue;
      }
    </style>
  </head>
  <body>
    <h2 class="class2">Welcome to CSS</h2>
    <h2 class="class1">I love CSS &#x1F496;</h2>
    <input class="class1" value="Hello CSS"><hr>
    <input class="class2" value="Hi CSS">
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن صنع الكلاس لكتابة أكواد به، لابد أن يبدأ برمز الدوت . قبل كتابة الاسم، ثم يتم تخصيص اسمه بدوت الدوت . لخاصية class لعناصر متعددة لتطبيق تنسيق واحد عليها.

انتبه! لو أن هناك عنصر وتريد أن يأخذ تنسيق أكثر من كلاس، فيتم ذلك عن طريق كتابة اسم الكلاس ثم مسافة، ثم اسم كلاس آخر، ثم مسافة، وهكذا لأي عدد، ومثال على ذلك الكود التالي:

```
<input class="class1 class2 class3">
```

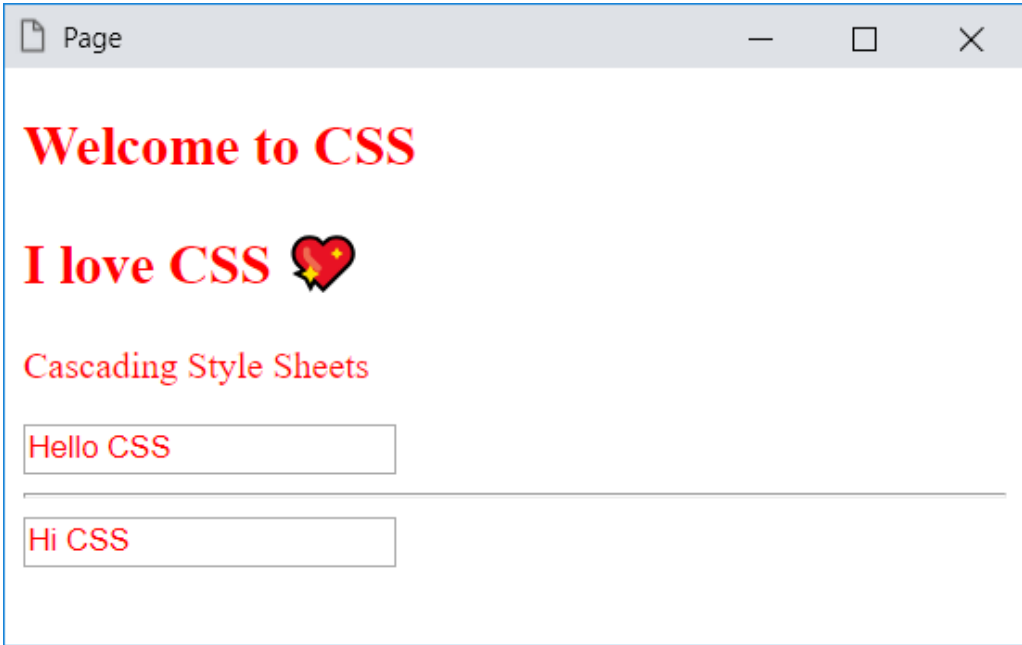
الهدف الكلي Universal Selector

وهو أن تحدد كل شيء، عن طريق رمز النجمة Asterisk * ثم كتابة الكود.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page</title>
    <style>
      *{
        color:red;
      }
    </style>
  </head>
  <body>
    <h2>Welcome to CSS</h2>
    <h2>I love CSS &#x1F496;</h2>
    <p>Cascading Style Sheets</p>
    <input value="Hello CSS"><hr>
    <input value="Hi CSS">
  </body>
</html>
```


وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



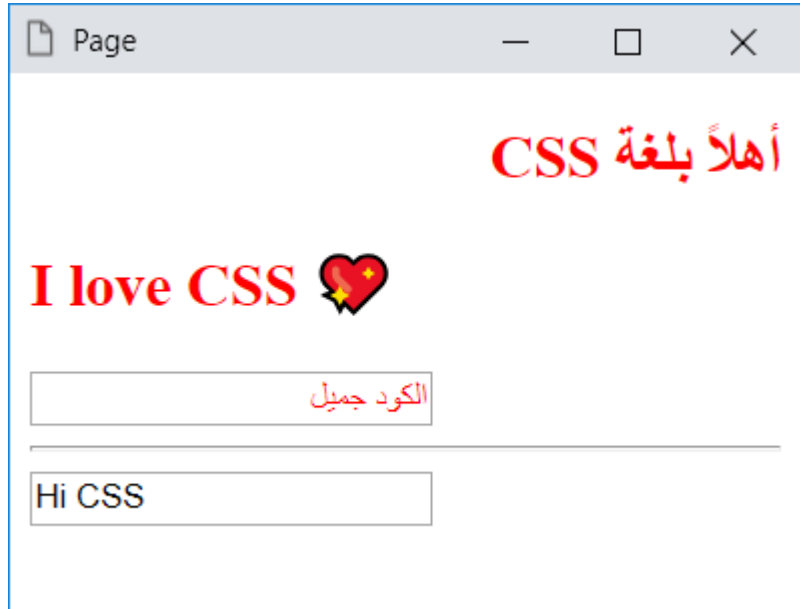
لاحظ كتابة الرمز * ثم الكود، وبعدها تأثر كل شيء بالأكواد الموجودة بعد الرمز * بشكل مباشر، وهذا معناه أن الرمز * هنا بمعنى "الك".

الهدف بالخاصية Attribute Selector

ومعني هذا أن يتم البحث عن خاصية وكتابة أكواد لها، ثم يتأثر أي عنصر يحمل تلك الخاصية بأي قيمة.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page</title>
    <style>
      [dir]{
        color:red;
      }
    </style>
  </head>
  <body>
    <h2 dir="rtl">أهلاً بلغة CSS</h2>
    <h2 dir="ltr">I love CSS &#x1F496;</h2>
    <input dir="rtl" value="الكود جميل"><hr>
    <input value="Hi CSS">
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم كتابة اسم الخاصية داخل أقواس مربعة [] ثم كتابة أكواد بعدها.

لاحظ أنه تم استخدام الخاصية dir للتجربة.

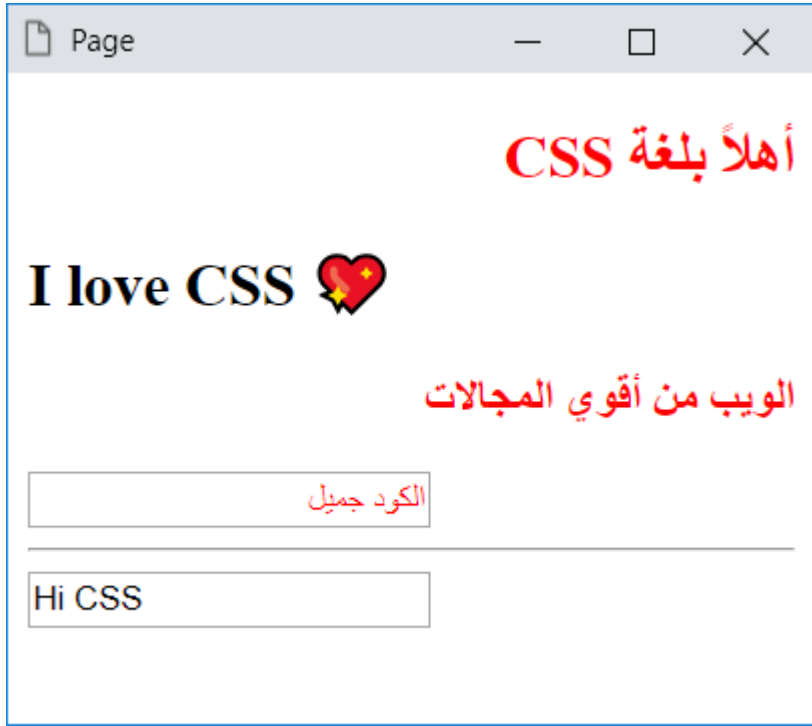
لاحظ أن جميع العناصر التي تحمل الخاصية dir بأي قيمة تم التأيد عليها، والتي لا تحمل الخاصية dir لم يتم التأيد عليها.

انتبه! فإنه يملك تحديد خاصية ذات قيمة محددة كما تريد، وذلك ما سوف ندرّب عليه فيما يلي.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page</title>
    <style>
      [dir=rtl]{
        color:red;
      }
    </style>
  </head>
  <body>
    <h2 dir="rtl">أهلاً بلغة CSS</h2>
    <h2 dir="ltr">I love CSS &#x1F496;</h2>
    <h3 dir="rtl">الويب من أقوى المجالات</h3>
    <input dir="rtl" value="الكود جميل"><hr>
    <input value="Hi CSS">
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن ما تأثر فقط هي العناصر التي تم ضبط خاصية dir لها على القيمة rtl.

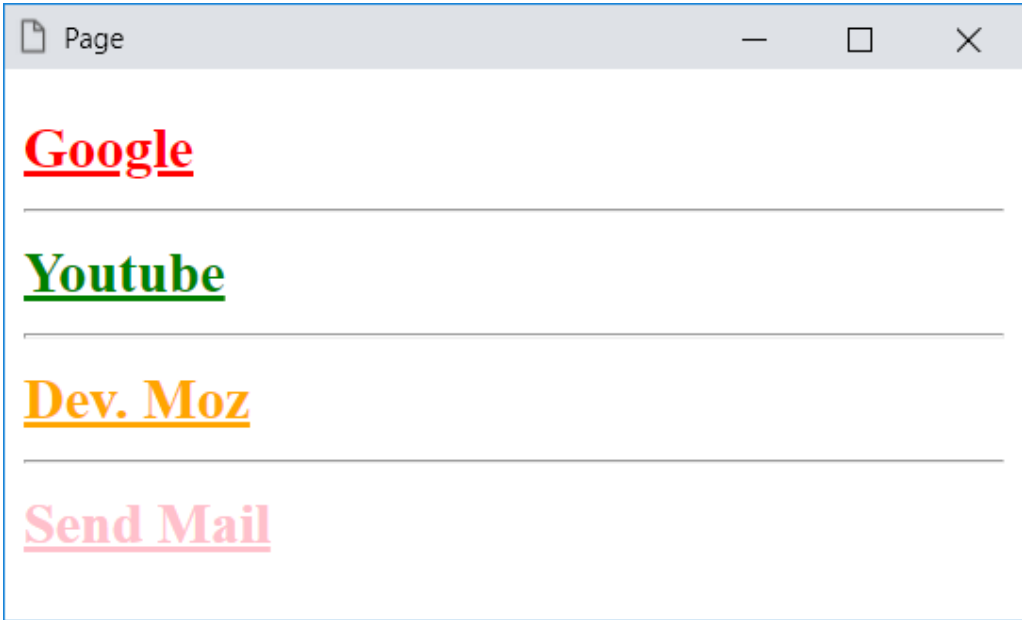
لاحظ أنك تستطيع استهداف أنواع محددة للعناصر مثل الـ `input` الذي يحمل النوع `text` أو النوع `button`، وهكذا لباقي الوسوم.

انتبه! فإنه يوجد اختيارات أخرى في `attribute selector` وسوف نوضحها في المثال التالي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>Page</title>
<style>
a[href="http://www.google.com"]{
  color:red;
}
a[href*="youtube"]{
  color:green;
}
a[href$=".org"]{
  color:orange;
}
a[href^="mailto"]{
  color:pink;
}
</style>
</head>
<body>
<h2>
<a href="http://www.google.com">Google</a><hr>
<a href="http://www.youtube.com">Youtube</a><hr>
<a href="http://developer.mozilla.org">Dev. Moz</a><hr>
<a href="mailto:testmail@domain.com">Send Mail</a>
</h2>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ التأثيرات في أكواد CSS في حالة الرمز = فقط المقصود التأثير على رابط جوجل بالكامل، أما عند وضع * قبل الرمز = فهذا يعني أنه لو كانت الخاصية بها الكلمة youtube في اسم مكان يؤثر عليها، أما عند وضع \$ قبل الرمز = فهذا يعني أن يكون آخر الخاصية org. في النهاية، أما عند وضع ^ قبل الرمز = فهذا يعني أن تبدأ الخاصية بالكلمة mailto في أولها، كما يوجد أيضاً ~ ومعناها أن يقارن القيمة بكلمات الخاصية لأن يسهل كل كلمة وأخرى مسافة، كما يوجد أيضاً |= ومعناها أن يقارن بالقيم التي بينها رمز - وغالباً تستخدم في اللغات.

انتبه! فإنه يمكن اشتراك عناصر متعددة في نفس تنسيق اللغة CSS،
وكأنني أقول عنصر "و" عنصر "و" عنصر "و" أو كأنني أقول
عنصر "و" كلاس "و" عنصر "و" معرف id "و" كلاس... وهكذا.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page</title>
    <style>
      input , .my1 , .my2 , #myh , p[dir=rtl]{
        color:red;
      }
    </style>
  </head>
  <body>

    <h2 class="my1">I Love CSS</h2>
    <h2 class="my2">CSS is Beautiful</h2>
    <input value="Text Box 1">
    <input value="Text Box 2">
    <h2 id="myh">I love CSS World &hearts;</h2>
    <p dir="rtl">أهلاً بكم في عالم الويب الجميل</p>

  </body>
</html>
```


وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم عمل تنسيق واحد، واشترك فيه عناصر متعددة، ومختلفة.

لاحظ اشتراك النوع normal والنوع class والنوع id والنوع attribute، في تنسيق واحد.

لاحظ أن هذه الطريقة توفر الوقت بشكل رائع، لأنها تضم تنسيق يتم كتابته مرة واحدة، ثم يتم تنفيذه على اختيارات مختلفة.

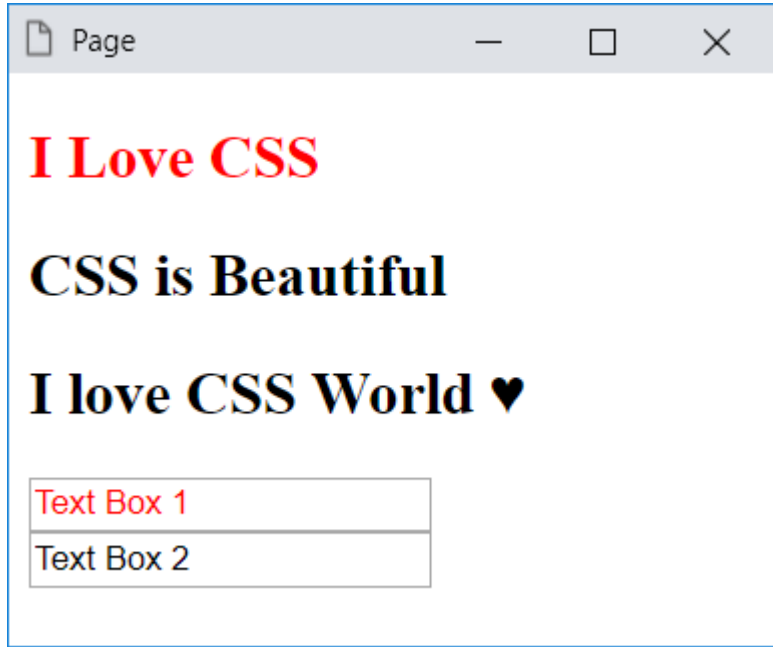
انتبه! المسافات الزائدة بين كل كوما وأخرى تنسيق فقط، ويمكنك ازالته
ويمكنك أيضاً أن تقصد وسم بشرط أن يحمل بداخلة كلاس معين، كما يلي.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page</title>
    <style>
      input.my1 , h2.my1{
        color:red;
      }
    </style>
  </head>
  <body>

    <h2 class="my1">I Love CSS</h2>
    <h2 class="my2">CSS is Beautiful</h2>
    <h2>I love CSS World &hearts;</h2>
    <input class="my1" value="Text Box 1">
    <input value="Text Box 2">

  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ في أكواد CSS أنه تم كتابة `input` ثم `دوت` . ثم الكلاس `my1`، وهذا يعني أن العنصر المقصود هو أي `input` يحمل الكلاس `my1`، ثم بعد ذلك كوما أي "و" ثم تم كتابة `h2` ثم `دوت` . ثم الكلاس `my1`، وهذا يعني أن العنصر المقصود هو أي `h2` يحمل الكلاس `my1`.

لاحظ أنه لم يتأثر إلا الوسوم `h2` والوسوم `input` فقط التي تحمل الكلاس `my1`.

الأهداف المتداخلة Combinators

وهي أن تقصد العناصر المتداخلة مع بعضها البعض، مثل ما يلي:

- عنصر بداخل عنصر، بداخل عنصر... وهكذا
- عنصر داخل عنصر بشكل غير مباشر
- عنصر بداخل عنصر بشكل مباشر
- عنصر بعد عنصر معيه بشكل مباشر
- كل العناصر بعد عنصر معيه بشكل مباشر
- عند فعل شيء معيه على عنصر

الى آخر هذه الاختيارات التي نحتاجها

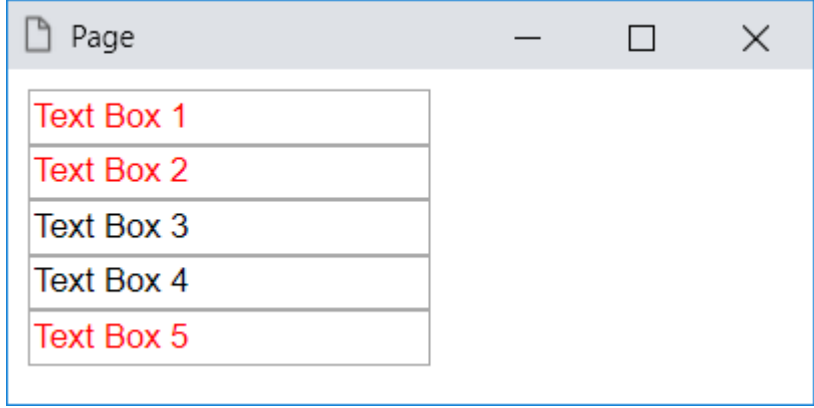
انتبه! فإنه لا مانع من أن تتنوع ال Selectors مع بعضها، بمعنى
أن تستخدم ال id بداخل class او normal، وهكذا.

عنصر داخل عنصر بأي طريقة

أو ما يسمى بـ Descendant combinator، ويتم هذا
باستخدام مسافة كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page</title>
    <style>
      div input{
        color:red;
      }
    </style>
  </head>
  <body>
    <div>
      <input value="Text Box 1">
      <input value="Text Box 2">
    </div>
    <input value="Text Box 3">
    <input value="Text Box 4">
    <div>
      <input value="Text Box 5">
    </div>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه يوجد في الصفحة 5 وسوم input الأول والثاني داخل الوسم div، والثالث والرابع على الصفحة، والخامس في وسم div آخر.

لاحظ في أكواد CSS أنه تم اختيار الوسم div ثم مسافة space ثم الوسم input، ومعنى ذلك أن المقصود كل input موجود في div.

لاحظ جيداً أن التأثير على الوسوم input الموجودة بداخل أي وسم div فقط.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page</title>
    <style>
      div div input{
        color:red;
      }
    </style>
  </head>
  <body>
    <div>

      <div>
        <input value="Text Box 1">
      </div>

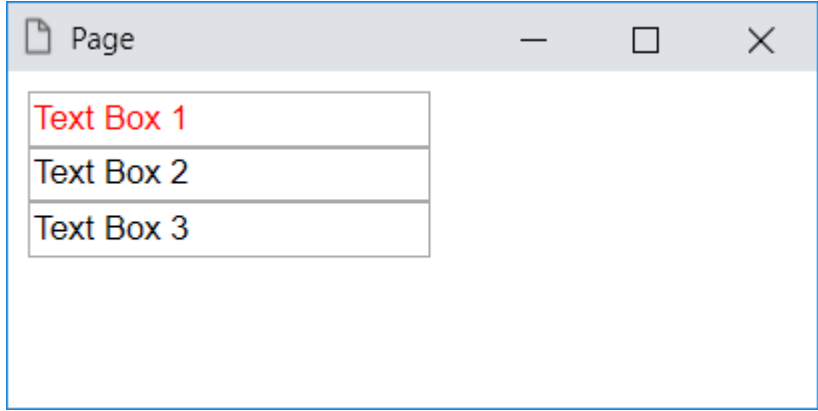
      <input value="Text Box 2">

    </div>

    <input value="Text Box 3">

  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ في أكواد CSS أنه تم كتابة div، ثم مسافة، ثم div، ثم input، وهذا يعني أن العنصر المقصود التأنيده عليه هو أي وسم input ولكنه بشرطيه، الشرط الأول، أن يكون بداخل وسم div، والشرط الثاني، أن يكون الـ div الذي يحتويه بداخل وسم div آخر.

لاحظ أن الـ input الذي تم تخصيص خاصية value له على القيمة Text Box 1 هو الذي تأثر فقط، لأنه هو فقط الذي تم وضعه في div ثم تم وضع هذه الـ div في div أخرى.

انتبه! فإنه بنفس الطريقة تستطيع أن تصل إلى عناصر مختلفة، داخل عناصر أخرى بسهولة.

كود! قم بكتابة الكود التالي ثم احفظه:

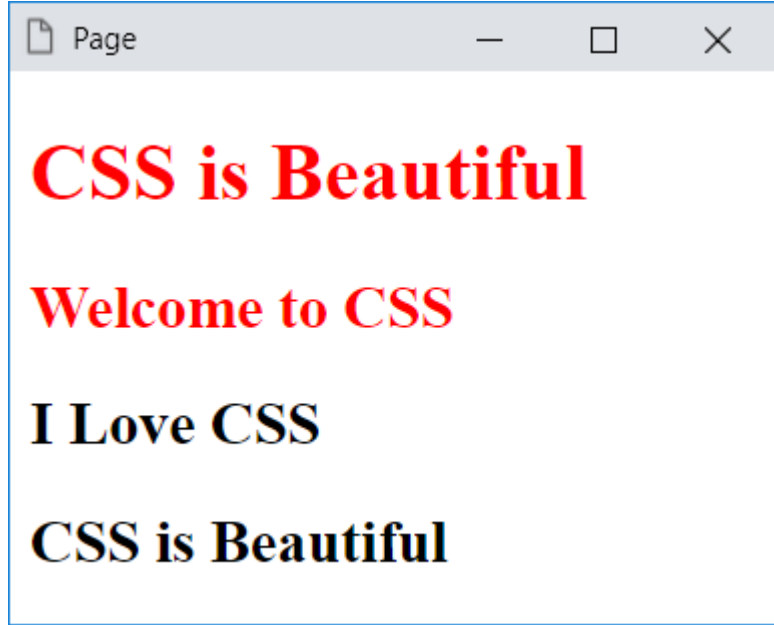
```
<!DOCTYPE html>
<html>
  <head>
    <title>Page</title>
    <style>
      div .my1{
        color:red;
      }
    </style>
  </head>
  <body>

    <div>
      <h1 class="my1">CSS is Beautiful</h1>
      <h2 class="my1">Welcome to CSS</h2>
    </div>

    <h2 class="my1">I Love CSS</h2>
    <h2 class="my1">CSS is Beautiful</h2>

  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ في أكواد CSS أنه تم كتابة `div`، ثم مسافة، ثم `my1`. ومعني هذا أن العنصر المقصود التأثير عليه، هو أي عنصر يحمل الكلاس `my1` بشرط أن يكون بداخل `div`.

لاحظ أنه يوجد وسميه `h2` يحملان الكلاس `my1` ولم يتأثروا لأنهم غير موجودين بوسم `div`.

انتبه! أنه تم الخلط بين النوع `normal selector` والنوع `class selector` في نفس العمل، مع العلم أنه بنفس هذا الفهم تستطيع الخلط بين باقي الـ `selectors` المختلفة.

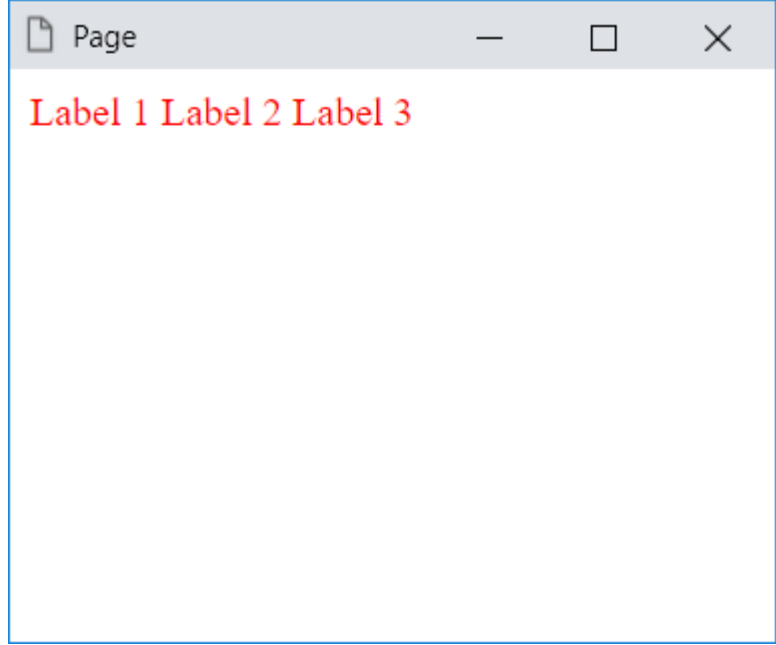
انتبه! فإن استخدام المسافة مع ال selectors المختلفة لا يفرق ما إذا كان العنصر المقصود، داخل عنصر بشكل مباشر، أو غير مباشر، وسوف يتضح ذلك من خلال المثال التالي.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page</title>
    <style>
      div label{
        color:red;
      }
    </style>
  </head>
  <body>

    <div>
      <label>Label 1</label>
      <span><label>Label 2</label></span>
      <label>Label 3</label>
    </div>

  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم إنشاء div وبداخلها label وspan يحمل label وlabel آخر.

لاحظ أن label 1 وlabel 3 داخل الdiv بشكل مباشر، أما label 2 داخل span أي ليس مباشر، ومع ذلك تم التأيد على الجميع دون أي اختلاف.

عنصر داخل عنصر بشكل مباشر

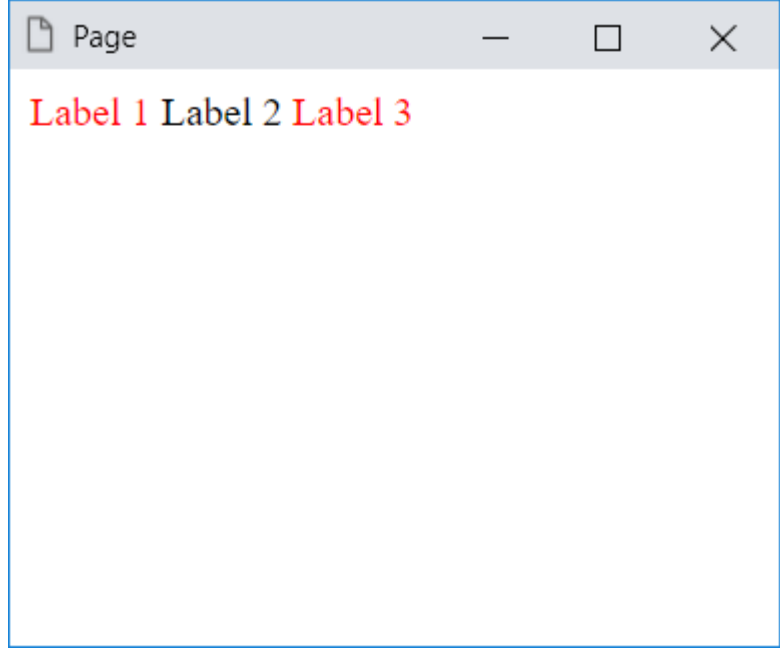
أو ما يعرف بـ Child combinator ويتم هذا باستخدام رمز القوس > كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page</title>
    <style>
      div > label{
        color:red;
      }
    </style>
  </head>
  <body>

    <div>
      <label>Label 1</label>
      <span><label>Label 2</label></span>
      <label>Label 3</label>
    </div>

  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم انشاء div وبداخلها label وبعده span يحمل label وبعده ال span يوجد label آخر.

لاحظ أن label 1 و label 3 داخل ال div بشكل مباشر، أما ال label 2 بداخل span أي ليس مباشر.

لاحظ أنه تم التأيد على وسوم label الموجودة داخل ال div بشكل مباشر فقط.

العنصر التالي

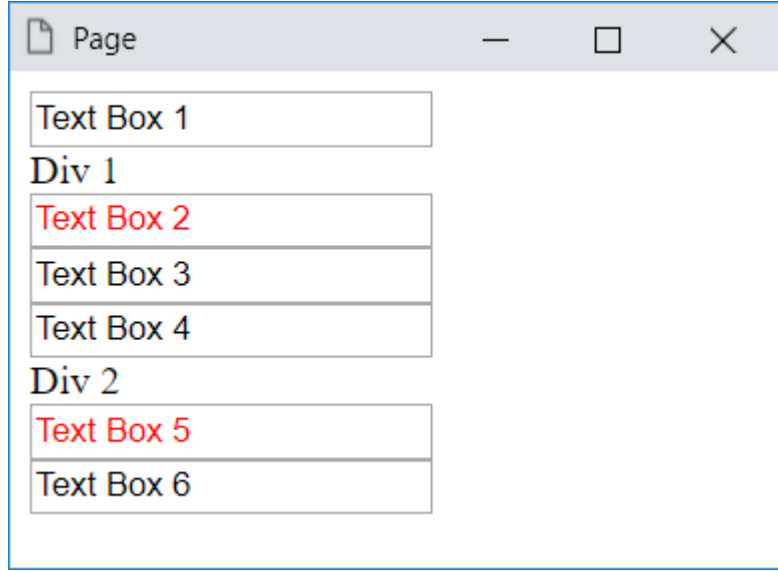
أو ما يعرف بـ Adjacent sibling combinator ويتم
هذا باستخدام الرمز + كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page</title>
    <style>
      div + input{
        color:red;
      }
    </style>
  </head>
  <body>
    <input value="Text Box 1">

    <div>Div 1</div>
    <input value="Text Box 2">
    <input value="Text Box 3">
    <input value="Text Box 4">

    <div>Div 2</div>
    <input value="Text Box 5">
    <input value="Text Box 6">
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ في أكواد CSS أنه تم كتابة div ثم + ثم input، ومعني ذلك أن العنصر المقصود هو أي وسم input يلي الـ div، مع العلم أنه لو كان هناك أكد من input يلي الـ div فسوف يتأثر واحد فقط.

لاحظ أن العناصر التي تأثرت هنا هم عنصريه، الأول Text Box 2 لأنه يلي Div 1، والثاني Text Box 5 لأنه يلي Div 2.

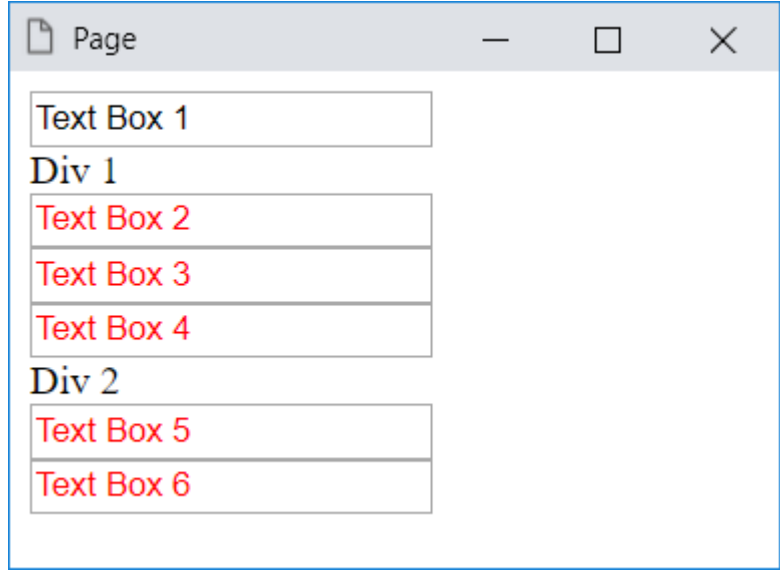
كل العناصر التالية أو أشقاء العنصر مع بعده
أو ما يعرف بـ General sibling combinator ويتم هذا
باستخدام الرمز ~ كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page</title>
    <style>
      div ~ input{
        color:red;
      }
    </style>
  </head>
  <body>
    <input value="Text Box 1">

    <div>Div 1</div>
    <input value="Text Box 2">
    <input value="Text Box 3">
    <input value="Text Box 4">

    <div>Div 2</div>
    <input value="Text Box 5">
    <input value="Text Box 6">
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ في أكواد CSS أنه تم كتابة div ثم ~ ثم input، ومعني ذلك أن العنصر المقصود هو جميع وسوم input التي تلي الوسم div.

لاحظ أن العناصر التي تأثرت هنا هم خمسة عناصر، وهم العناصر Text Box 4 و Text Box 3 و Text Box 2 لأنهم بعد Div 1، والعناصر Text Box 5 و Text Box 6 لأنهم بعد Div 2.

ويوجد أيضاً column combinator باستخدام || لاختيار خلية في جدول تنتمي إلى عمود، ولكنه **انتبه!** فهو نادر الاستخدام، وقل منه يتحدث عنه.

الكود الذي يغلب Active Code

عند كتابة أكواد تقصد نفسك العنصر، من منهم يغلب؟ وهذا هو الموضوع.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page</title>
    <style>
      h1{
        color:red;
      }
      h1{
        color:blue;
      }
    </style>
  </head>
  <body>

    <h1>Welcome to CSS</h1>

  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم تنسيق للوسم h1 مرتين، أولاً مرة باللون الأحمر، وثانياً مرة باللون الأزرق، والكود الذي غلب هنا هو الكود الأخير.

والآن، مقارنة أخري بين inline و internal:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page</title>
    <style>
      h1{ color:red; }
      h1{ color:blue; }
    </style>
  </head>
  <body>
    <h1 style="color:orange">Welcome to CSS</h1>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم كتابة تنسيقيهما للوسم `h1`، وتم توجيه `style` له بشكل مباشر، بطريقة `inline style` وهي التي تغلب على الكلد، ولكنه هناك كلمة وكان بها سحر، تغلب أكثر وهي `!important`، كما سنجرب الآن بفضل الله:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page</title>
    <style>
      h1{ color:red !important; }
      h1{ color:blue; }
    </style>
  </head>
  <body>
    <h1 style="color:orange">Welcome to CSS</h1>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه عند كتابة `important` بعد قيمة الخاصية غلبت على الكلا، حتى ولو كان آخر تنسيق، بل وحتى لو كان `inline style` أيضاً تغلب عليه، كما هو واضح في الصورة.

انتبه! لأن هذا المفهوم يجب أن تركز معه جيداً عند كتابة الكود في الصفحة `internal style` أو استدعاء ملفات `CSS` خارجية `external style`، وذلك حتى لا يأتي وقت وأنت تعمل تختلط عليك الكود وتنساء! لماذا لا يعمل؟ أو ما الذي حدث؟ أو هل الكود خطأ؟، إلى آخر هذه الأسئلة التي تحير المصممين، إذا أخذنا قاعدة، الكود الأخير يغلب، وكود ال `inline` يغلب عليه، وكتابة `important` تغلب على الكلا.

الحجم أو الأبعاد Size or Dimensions

ننظف الطول والعرض للعنصر شيءٍ معهم جداً، أثناء العمل على صفحة الويب، ويمكنك تحديد العرض عن طريق الخاصية width للعناصر، كما يمكنك أيضاً تحديد الطول عن طريق الخاصية height للعناصر.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      input{
        width:300px;
        height:100px;
      }
    </style>
  </head>
  <body>
    <input type="text" value="Text Box">
    <input type="submit">
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

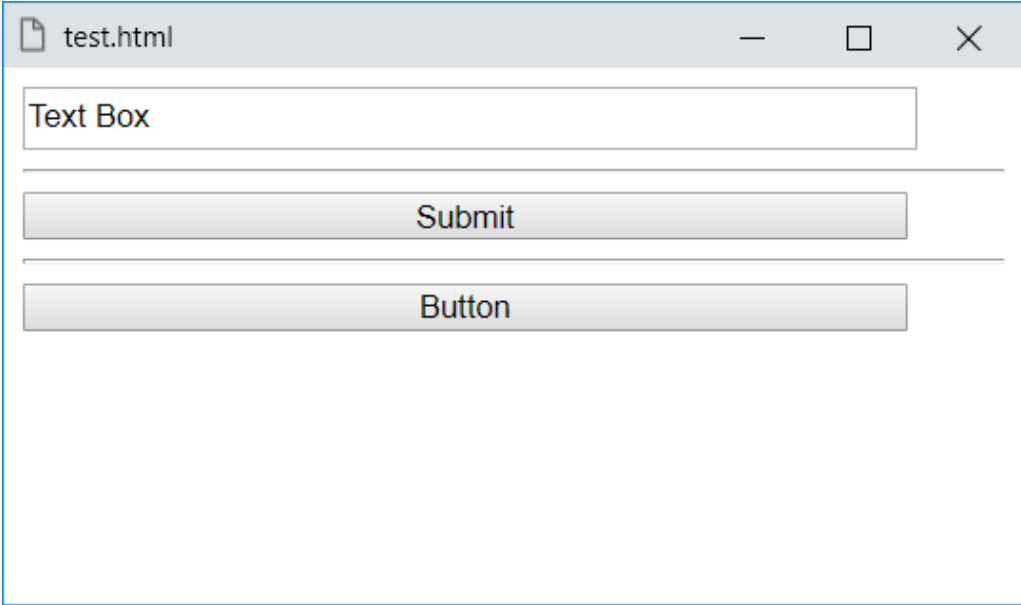


لاحظ وجود العناصر على صفحة الويب، مع التحكم في الطول والعرض لها، حيث أصبح العرض 300px والطول 100px لهم.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body{ height:200px; }
      input , button{
        width:90%;
        height:10%;
      }
    </style>
  </head>
  <body>
    <input type="text" value="Text Box"><hr>
    <input type="submit"><hr>
    <button>Button</button>
  </body>
</html>
```


وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

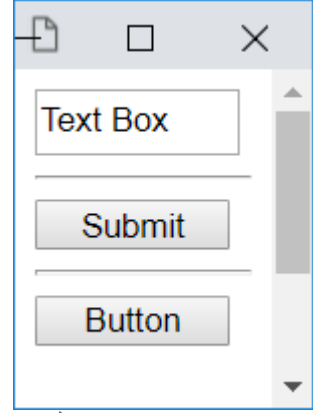


لاحظ ضبط طول عنصر الصفحة الأساسي body على 200px.

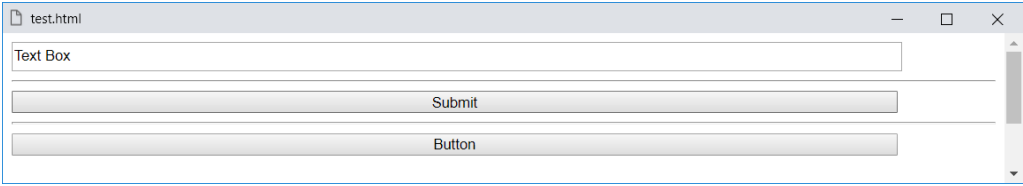
لاحظ ضبط الوسمين input و button على عرض 90% مع عرض المكان المتواجدان فيه، وضبط الطول على 10% مع طول المكان المتواجدان فيه.

لاحظ أن طول العناصر 10% وبما أن طول الوسم body تم ضبطه على القيمة 200px إذاً فإن طول العناصر هو 20px لأنها هي ناتج الـ 10% من الـ 200px للوسم body، ولو غيرت القيمة إلى 20px بدلاً من 10% ثم حدثت الصفحة، ستلاحظ أنه لا فرق.

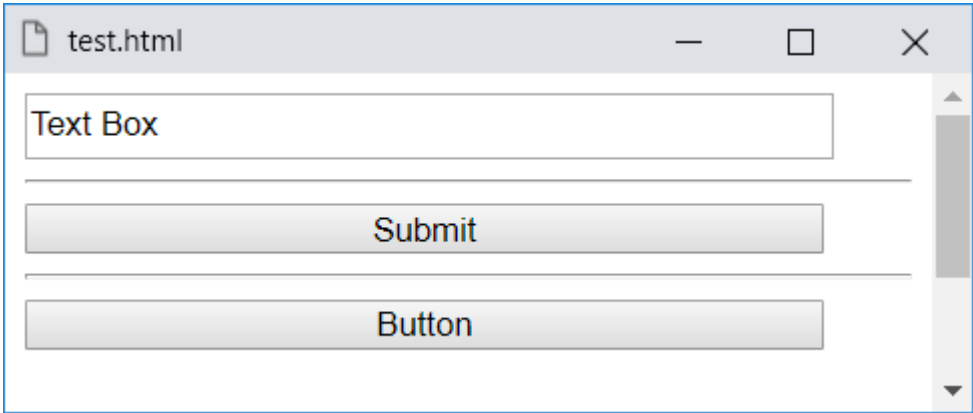
الآن سوف نقوم بتصغير عرض نافذة المتصفح ليكون مثل الشكل التالي:



الآن سوف نقوم بتكبير عرض نافذة المتصفح ليكون مثل الشكل التالي:



الآن سوف نقوم بتغيير عرض نافذة المتصفح ليكون مثل الشكل التالي:



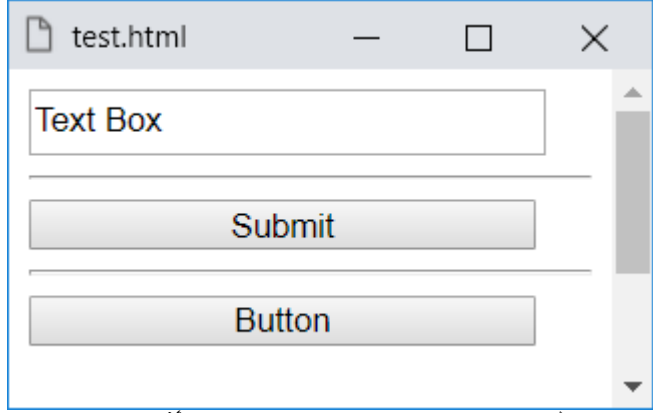
لاحظ أثر ثبات النسبة التي أعطيناها الى خاصية width للعناصر، سواء تم التصغير أو التكبير لنافذة المتصفح، حيث تكبر معها، وتصغر معها أيضاً.

ويمكن التحكم أيضاً في أقصى طول وعرض وأقل طول وعرض، ولكنه انتبه!
جيداً لتفهمها، لأن أثرها يظهر أثناء العرض.

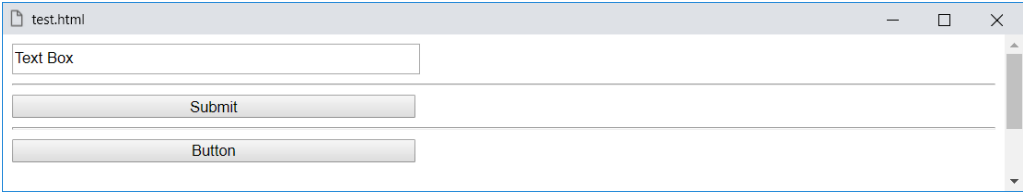
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body{ height:200px; }
      input , button{
        width:90%;
        height:20px;
        max-width:350px;
        min-width:200px;
      }
    </style>
  </head>
  <body>
    <input type="text" value="Text Box"><hr>
    <input type="submit"><hr>
    <button>Button</button>
  </body>
</html>
```

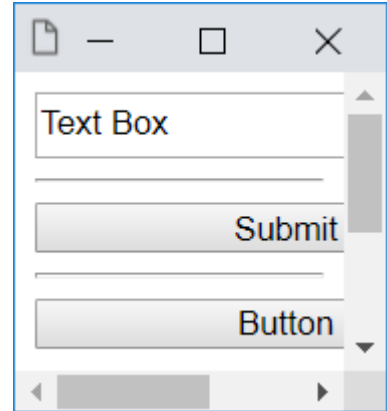
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



سوف نكتب عرض نافذة المتصفح أكثر من 350px وأنظر التالي:



لاحظ أن النسبة تعطلت أن نكتب العناصر بعد 350px لأنه تم ضبط خاصية max-width على 350px، أي أقصى قيمة. والآن سوف نصغر عرض نافذة المتصفح أقل من 200px وأنظر التالي:

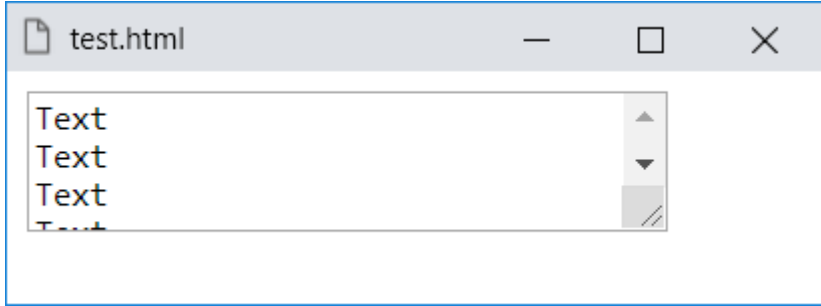


العناصر لم تستجيب للتصغير أقل من 200px بسبب min-width.

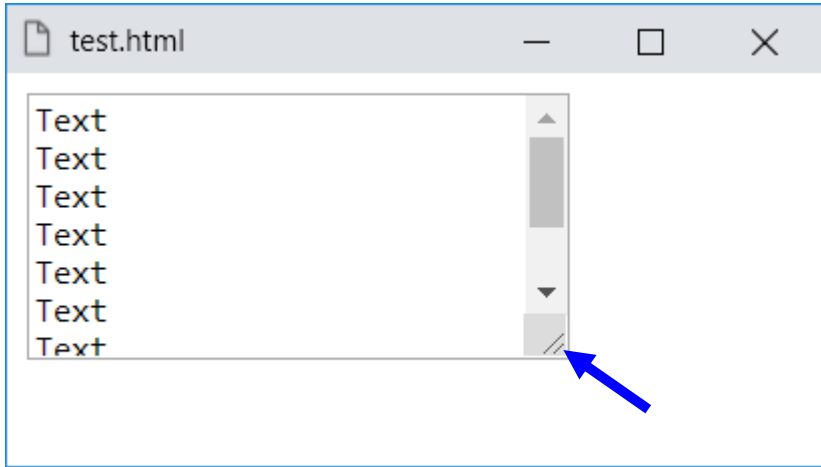
وبنفس الطريقة التي تم استخدامها وتجربتها بالنسب المئوية والتكبير والتصغير مع الخواص `min-width` و `max-width`، أيضاً يتم استخدامها مع الخواص `min-height` و `max-height`، ولكن للتجارب على الطول، وسنجرّب **بلود!** فقم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      textarea{
        width:250px;
        min-height:50px;
        max-height:100px;
      }
    </style>
  </head>
  <body>
    <textarea>
Text&#13;Text&#13;Text&#13;
Text&#13;Text&#13;Text&#13;
Text&#13;Text&#13;Text&#13;
    </textarea>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ استخدام الرمز `` للنزول سطر داخل `textarea`.
ولاحظ بدأ الطول مع `50px` بسبب الخاصية `min-height`.
والآن **انتبه!** حاول تكبير ال `textarea` مع صفحة الويب نفسها أثناء العرض، وسيقف عند طول معين بسبب الخاصية `max-height` كما يوضح الشكل التالي:



ولو تم زيادة قيمة الخاصية `max-height` للعنصر `textarea` سوف تستطيع تكبيره أكثر، معتمداً على قيمة `max-height`.

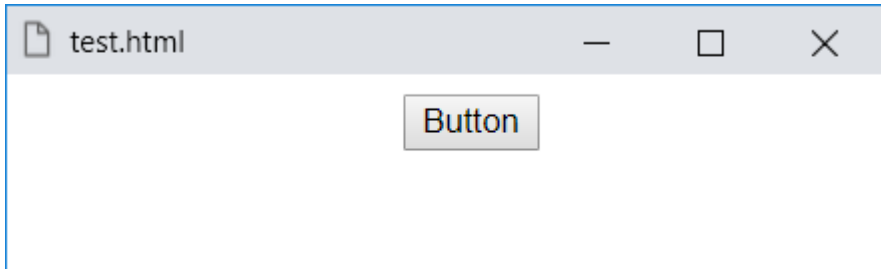
الهوامش Margins

تُضبط الهوامش في الويب، من المواضيع الهامة جداً، والهوامش هي الفراغات الخارجية لكل عنصر، أي الفراغات التي بين العناصر، حيث يوجد لكل عنصر هامش علوي، ويميني، وسفلي، ويساري، والخواص الخاصة بهم هي، `margin-top`، `margin-right`، `margin-bottom`، و `margin-left`، ويمكن استخدام خاصية `margin` وحدها لتشمل الاتجاهات الأربعة بالكامل، ويمكن إعطائهم قيمة بيكسل أو نسبة مئوية.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head></head>
<body>
  <button style="margin-left:150px;">
    Button
  </button>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

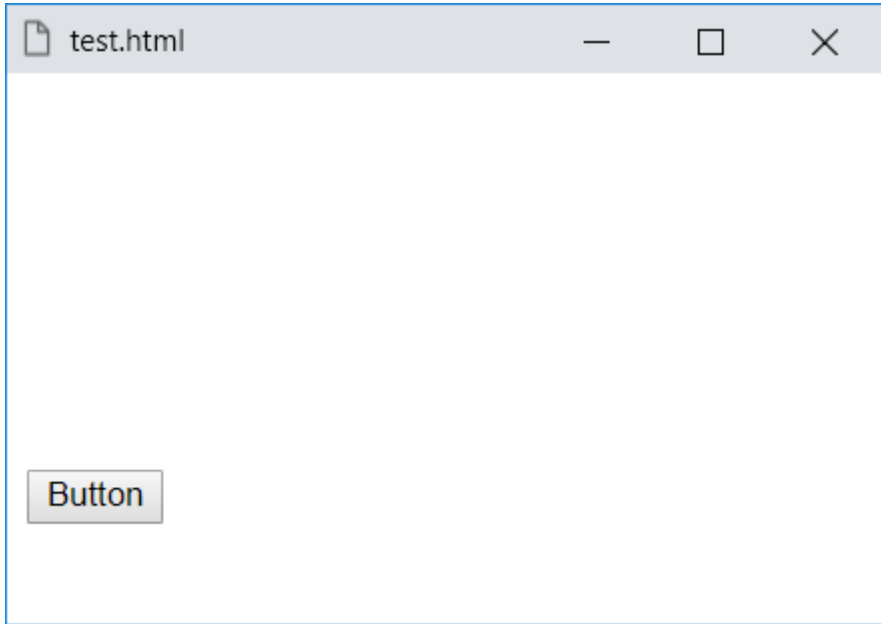


لاحظ ترك هامش للوسم button من اليسار 150px بسبب الخاصية .margin-left.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head></head>
  <body>
    <button style="margin-top:150px;">
      Button
    </button>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

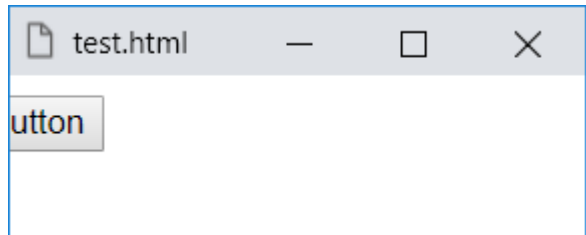


لاحظ ترك هامش للوسم button من الأعلى 150px بسبب الخاصية margin-top.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head></head>
  <body>
    <button style="margin-left:-25px;">
      Button
    </button>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



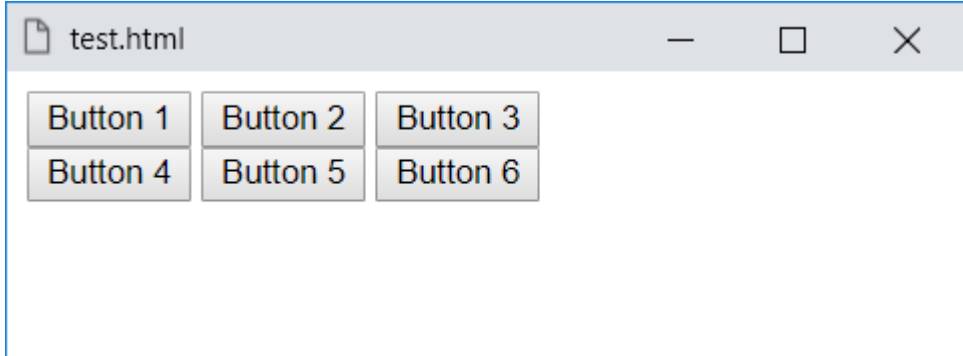
لاحظ مع إعطاء قيمة بيلسك بالسالب، يخرج العنصر من مسارة الطبيعي، الى خارج الصفحة.

انتبه! لو قمت بتجربة margin-right و margin-bottom له تلاحظ التأيد، لأنه يترك فراغ من الناحية العكسية للصفحة. والآن سوف نجرب كود بدون الهوامش، كما يلي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head></head>
  <body>
    <button>Button 1</button>
    <button>Button 2</button>
    <button>Button 3</button><br>
    <button>Button 4</button>
    <button>Button 5</button>
    <button>Button 6</button>
  </body>
</html>
```

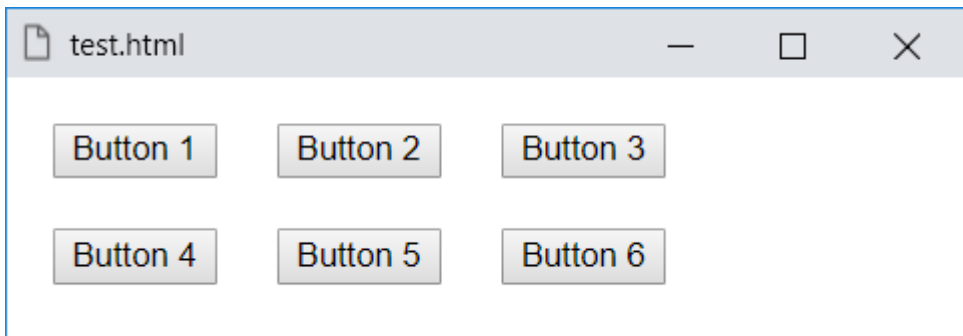
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



انتبه! فالآن سوف نجرب هوامش كلية لكل عنصر، كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      button{ margin:10px; }
    </style>
  </head>
  <body>
    <button>Button 1</button>
    <button>Button 2</button>
    <button>Button 3</button><br>
    <button>Button 4</button>
    <button>Button 5</button>
    <button>Button 6</button>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



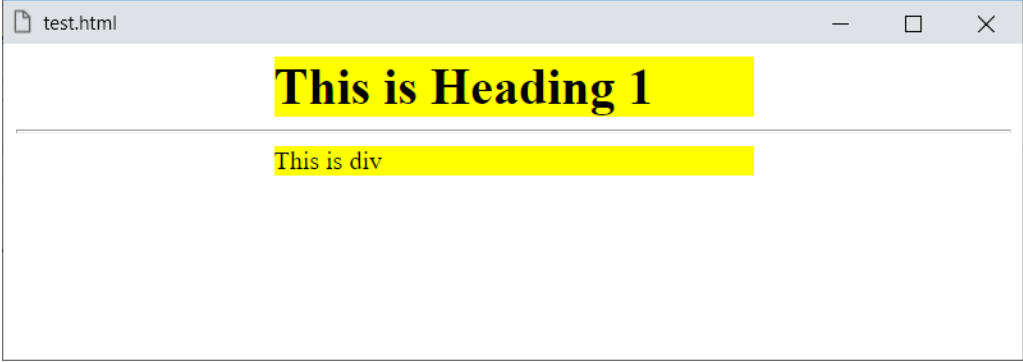
لاحظ أثر وجود هوامش 10px من الأربعة اتجاهات لجميع وسوم
 .button

انتبه! فإنه عن طريق الخاصية margin يمكنه جعل العناصر التي تنزل سطر في المنتصف، مثل الوسم div والوسم h1، وذلك عن طريق إعطاء العنصر عرض width ثم ضبط خاصية margin على القيمة auto، فيتم توزيع العنصر المتبقي بالتساوي على اليمين واليسار.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      div, h1{
        background-color:yellow;
        width:300px;
        margin:auto;
      }
    </style>
  </head>
  <body>
    <h1>This is Heading 1</h1>
    <hr>
    <div>This is div</div>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



انتبه! فإذا كان العرض كامل، فله تلاحظ هذا الفارق.

انتبه! فسوف نفترض، والآه تخيل معي، ان عرض الصفحة هو 650px وعرض الـ h1 هو 300px كما تم ضبطه، فإن المتبقي منه الـ 650 في الصفحة هو 350px، والآه سوف نقسم المتبقي على 2 ليكون $175 = 350 \div 2$ ، فيتم وضع هامش 175px على اليمين، ويتم وضع هامش 175px على اليسار، وهكذا.

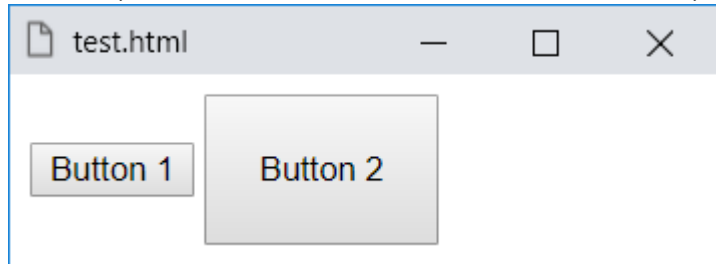
الحشو Padding

تُضبط الحشو في الويب، مع المواضيع العامة جداً، والحشو هو حشو داخلي لكل عنصر، أي فراغات مع داخل العناصر، حيث يوجد لكل عنصر، حشو علوي، ويميني، وسفلي، ويساري، والخواص الخاصة بهم هي، padding-top، padding-right، padding-bottom، وpadding-left، ويمكن استخدام خاصية padding وحدها لتشمل اتجاهات الحشو الأربعة بالكامل، ويمكن اعطائهم قيمة بيكسل أو نسبة مئوية، حسب الرغبة.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head></head>
  <body>
    <button>Button 1</button>
    <button style="padding:20px;">Button 2</button>
  </body>
</html>
```

وبعد تشغيل الصفحة يظهر أثر الحشو كما بالشكل التالي:

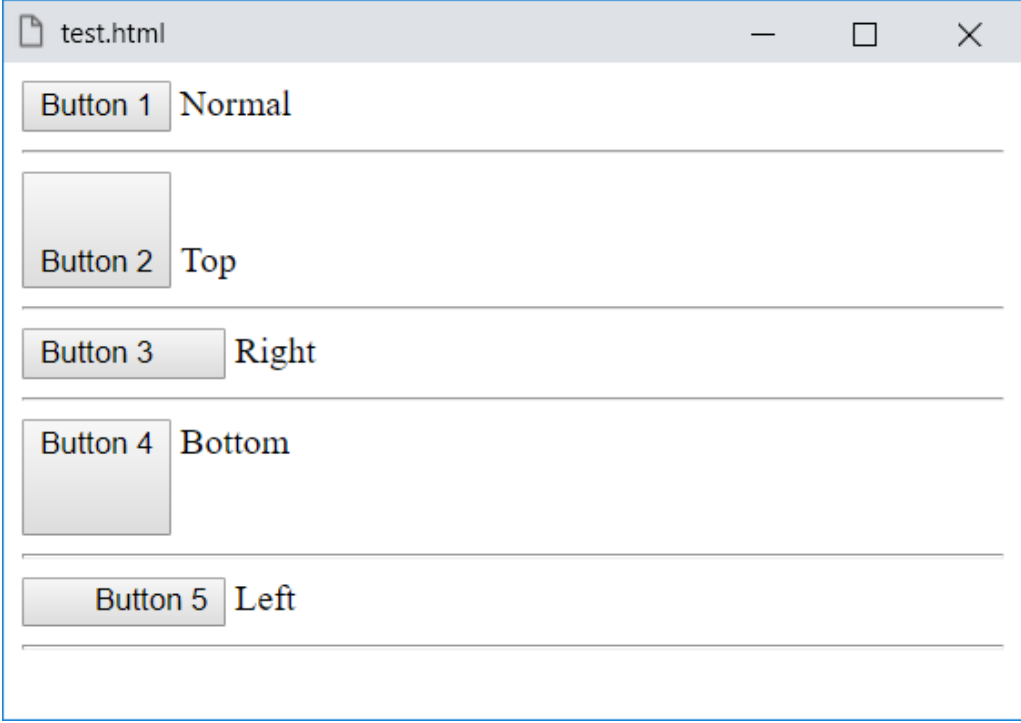


ولتجربة جميع اتجاهات الحشو، لا بد أن تركز في المثال التالي جيداً، حيث يتم فيه انشاء زر بدون حشو، ثم انشاء ازرار بعده تختلف في الحشو، وكتابة الحشو المستخدم بجوار الأزرار للتوضيح، كما يلي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head></head>
  <body>
    <button>Button 1</button> Normal <hr>
    <button style="padding-top:30px;">
      Button 2
    </button> Top <hr>
    <button style="padding-right:30px;">
      Button 3
    </button> Right <hr>
    <button style="padding-bottom:30px;">
      Button 4
    </button> Bottom <hr>
    <button style="padding-left:30px;">
      Button 5
    </button> Left <hr>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن Button 1 طبيعي ولا يوجد عليه أي أثر للحشو، ثم بعد ذلك لاحظ تأييد الحشو العلوي على Button 2، وتأييد الحشو من اليمين على Button 3، وتأييد الحشو السفلي على Button 4، وتأييد الحشو اليساري على Button 5.

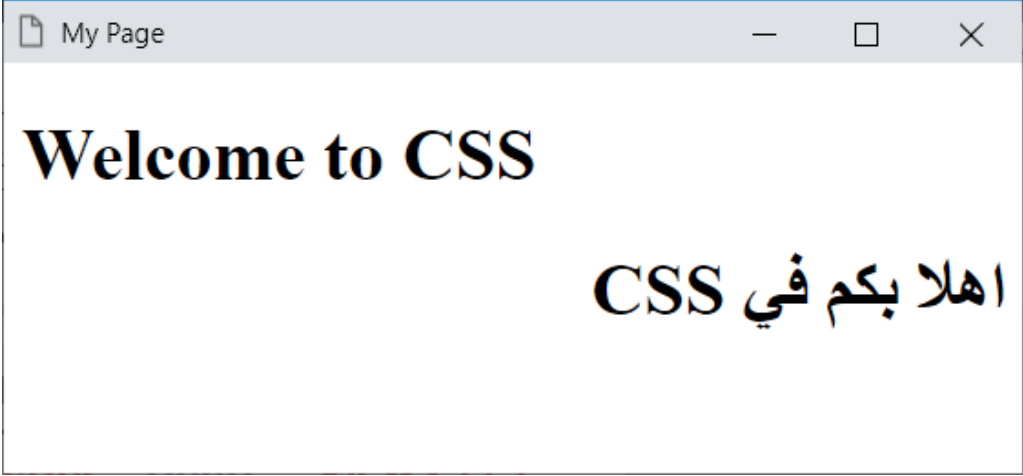
الاتجاه Direction

اتجاه العناصر direction

يمكنك تحديد اتجاهات العناصر، فإذا أنه تكون من اليسار الى اليمين، أو من اليمين الى اليسار، وسوف نوضح بفضل الله فيما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <title>
      My Page
    </title>
  </head>
  <body>
    <h1 style="direction:ltr">
      Welcome to CSS
    </h1>
    <h1 style="direction:rtl">
      CSS اهلا بكم في
    </h1>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن الوسم h1 الأول تم ضبط الاتجاه له على القيمة ltr أي من اليسار الى اليمين، وهي القيمة الافتراضية.

لاحظ أن الوسم h1 الثاني تم ضبط الاتجاه له على القيمة rtl أي من اليمين الى اليسار، فأصبح الاتجاه ملائم الى اللغة العربية.

منظّم اليونيكود unicode-bidi

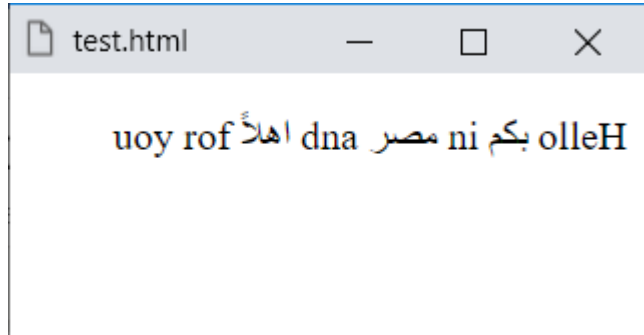
تعمل الخاصية unicode-bidi، جنباً إلى جنب مع خاصية direction، لتحديد كيفية التعامل مع النص ذا اتجاهيه في المستند، فعلى سبيل المثال، إذا كانت هناك فقرة من المحتوى تحتوي على نص من اليسار إلى اليمين، وأيضاً وفي نفس الوقت بها نص من اليمين إلى اليسار، فإنه يجب استخدام خوارزمية في الـ unicode معقدة لتحديد كيفية عرض النص، ولكه تتجاوز الخاصية unicode-bidi هذه الخوارزمية وتسمح للمطور بالتحكم أكثر في تضمين النص، والافتراضي لهذه الخاصية هي القيمة **normal**، ويمكّن استخدام القيمة **embed**، إذا كان العنصر من نوع inline، وتفتح هذه القيمة مستوى إضافي للتضمين فيما يتعلق بالخوارزمية ثنائية الاتجاه، ويتم إعطاء اتجاه مستوى التضمين هذا بواسطة خاصية الاتجاه direction، ويمكّن أيضاً استخدام القيمة **bidirectional-override**، للعناصر الـ inline، ويؤدي هذا إلى تخطي لما هو موجود، بالنسبة لعناصر حاوية الحفظ، ويؤدي أيضاً إلى التأثير على عناصر بداخل عناصر لمستوى الـ inline وليس الـ block، وهذا يعني أنه داخل العنصر، وتكون إعادة الترتيب متسلسلة بدقة وفقاً لخاصية الاتجاه، ويتم تجاهل الجزء الضمني من الخوارزمية ثنائية الاتجاه، ويمكّن استخدام القيمة **isolate**، وتشير هذه القيمة إلى أنه يجب حساب اتجاهية عنصر العنصر دون مراعاة محتوى هذا العنصر، وبالتالي يتم عزل العنصر عن أشقائه الذي معه في نفس المستوى، وعند تطبيق خوارزمية الدقة ثنائية الاتجاه، فإن عنصر الحاوية الخاص به

يعاملها كحرف واحد أو عدة أحرف استبدال كائنه الـ U+FFFC، مثلاً صورة، كما أنه يمكّن استخدام القيمة **isolate-override**، وتطبق هذه القيمة سلوك العزل للكلمة الرئيسية المعزولة عن المحتوى المحيط بها، وسلوك تجاوز الكلمة الرئيسية التي تتخطى الـ bidi إلى المحتوى الداخلي، ويمكّن استخدام القيمة **plaintext**، التي تجعل عناصر الاتجاهية محسوبة دون اعتبار حالتها الأصلية ثنائية الاتجاه أو قيمة خاصة الاتجاه direction، ويتم حساب الاتجاهية باستخدام قواعد P2 و P3 لخوارزمية الـ unicode ثنائية الاتجاه، وتسمح هذه القيمة بعرض البيانات المنسقة بالفعل باستخدام أداة تتبع خوارزمية الـ unicode ثنائية الاتجاه.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      p{
        direction:rtl; unicode-bidi: bidi-override;
      }
    </style>
  </head>
  <body>
    <p>Hello for you اهلاً and مصر in بكم </p>
  </body>
</html>
```

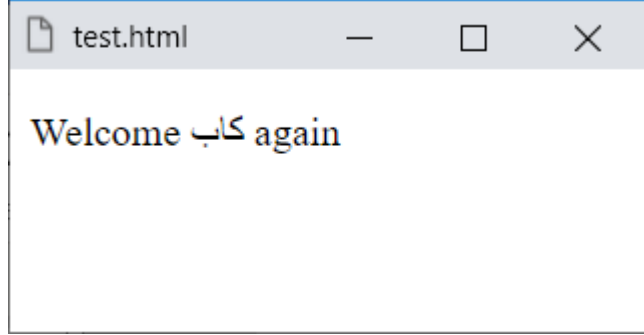
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



والآن حاول أن تقرأ الحروف الإنجليزية من اليمينه مثل الحروف العربية، مع العلم أنه إذا تم تغيير الاتجاه سوف يثبت الحروف الإنجليزية ويقرب العربية، وسنجرّب **بالكود!** التالي:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      p{
        direction:ltr; unicode-bidi: bidi-override;
      }
    </style>
  </head>
  <body>
    <p>Welcome بأك again</p>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



والآن جرب قراءة الحروف العربية من اليسار مثل الإنجليزي.

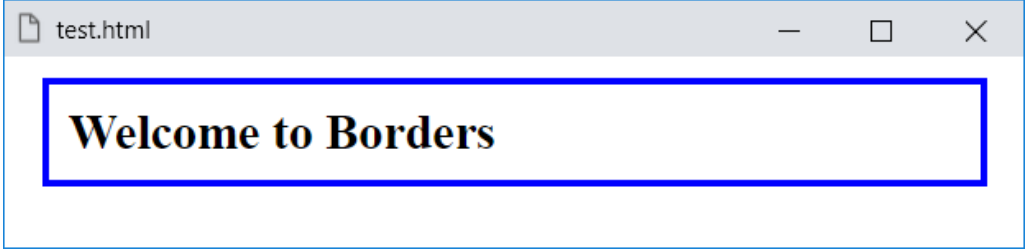
الإطارات Borders

الإطار هو الخط الذي يخلق بعنصر معيه، أو ما يعرف بالبرواز لتزيينه حواف العناصر، والإطار له أربع حواف، حافة عليا، وحافة اليمين، وحافة سفلي، وحافة اليسار.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      h2{
        margin:10px;
        padding:10px;
        border-width:3px;
        border-style:solid;
        border-color:blue;
      }
    </style>
  </head>
  <body>
    <h2>Welcome to Borders</h2>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لعمل إطار تم ضبط ما يلي:

تم ضبط سمك الإطار border-width على 3px.

تم ضبط نوع إطار border-style على solid أي صلب.

تم ضبط لون الإطار border-color على blue.

لاحظ أن الإطار تم من الحواف الأربعة بالكامل، بسمك 3px وخط صلب ولون أزرق.

انتبه! فإنه يمكنك إعطاء القيم الثلاثة للخاصية border مرة واحدة بدلاً من تحديد الثلاث خصائص السابقة، توفيراً للوقت، مثل الكود التالي:

```
border:3px solid blue;
```

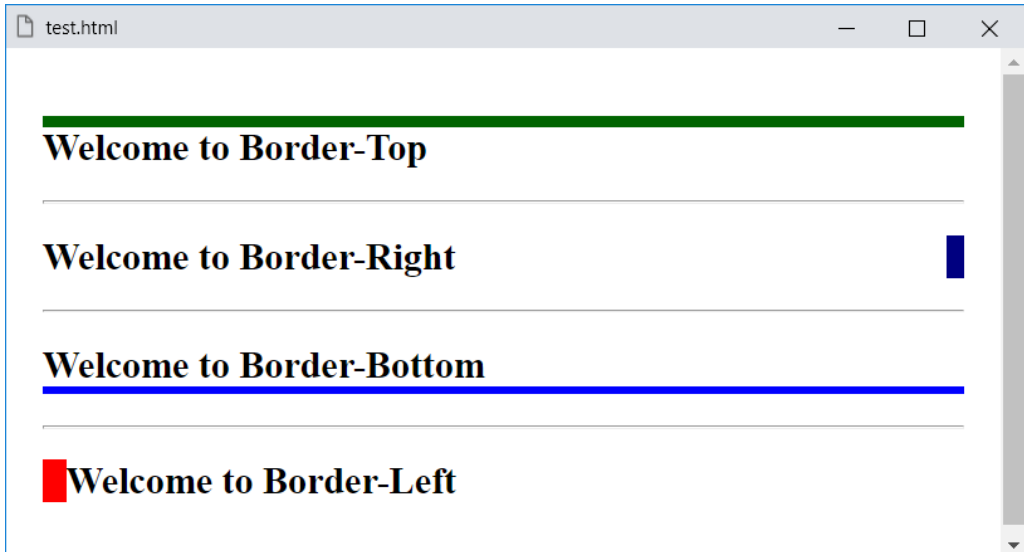
وبعد التشغيل سيتم إنشاء إطار بنفس الطريقة

كما يمكنك التحكم أيضاً في كل حافة بشكل مستقل، ليكون الإطار أكثر دقة وتفصيلاً، وتدقيقاً بفضل الله، وذلك كما يلي إن شاء الله:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head></head>
<body style="padding:15px;">
  <h2 style="border-top:7px solid darkgreen;">
    Welcome to Border-Top
  </h2><hr>
  <h2 style="border-right:11px solid navy;">
    Welcome to Border-Right
  </h2><hr>
  <h2 style="border-bottom:5px solid blue;">
    Welcome to Border-Bottom
  </h2><hr>
  <h2 style="border-left:15px solid red;">
    Welcome to Border-Left
  </h2>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

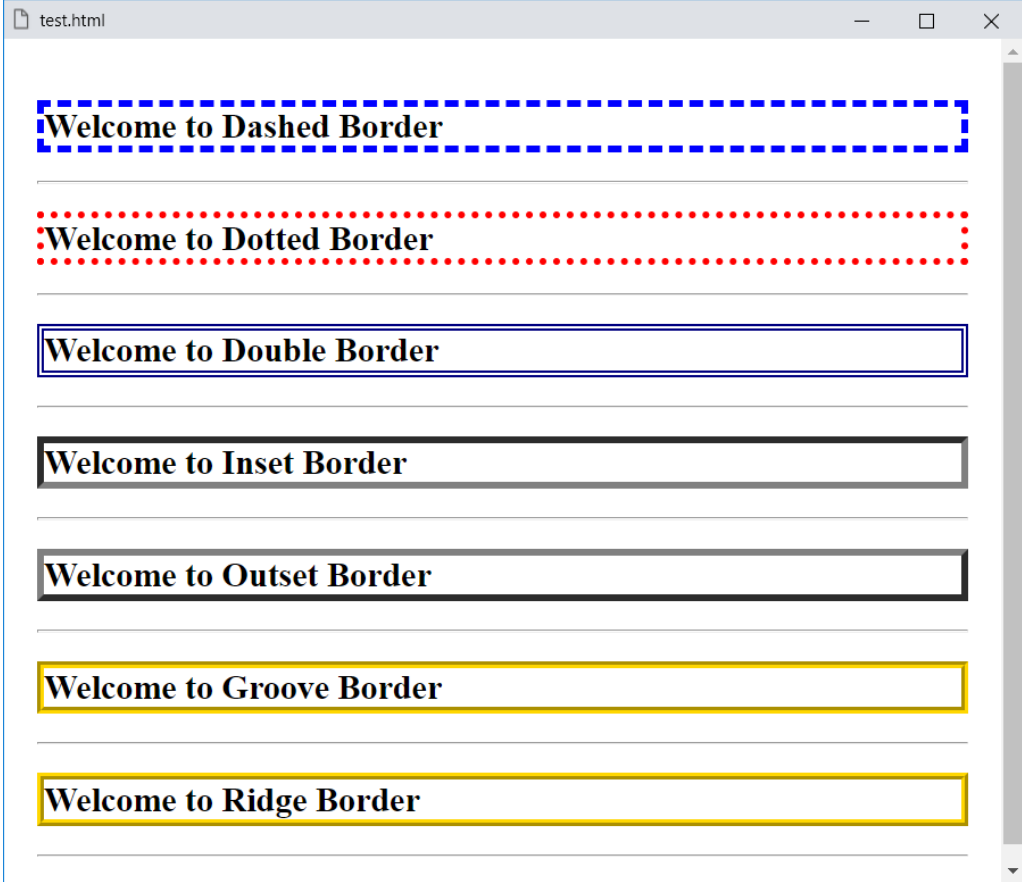


كما أنه يوجد أنواع أخرى من الإطارات، مثل خط منقطع، وخط منقط، وخط مزدوج، وأخري، كما يلي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head></head>
<body style="padding:15px;">
  <h2 style="border:5px dashed blue;">
    Welcome to Dashed Border
  </h2><hr>
  <h2 style="border:5px dotted red;">
    Welcome to Dotted Border
  </h2><hr>
  <h2 style="border:5px double navy;">
    Welcome to Double Border
  </h2><hr>
  <h2 style="border:5px inset gray;">
    Welcome to Inset Border
  </h2><hr>
  <h2 style="border:5px outset gray;">
    Welcome to Outset Border
  </h2><hr>
  <h2 style="border:5px groove gold;">
    Welcome to Groove Border
  </h2><hr>
  <h2 style="border:5px ridge gold;">
    Welcome to Ridge Border
  </h2><hr>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



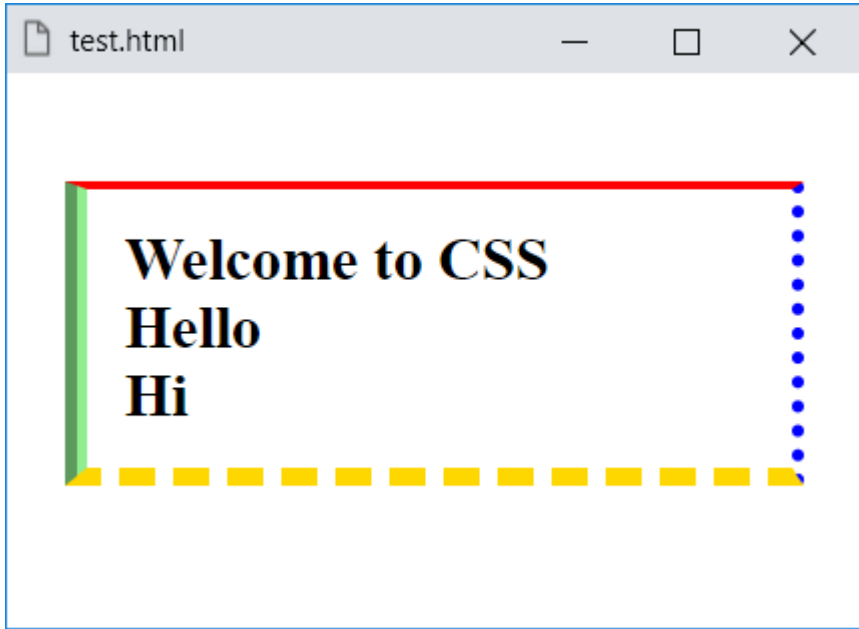
ومع كل أمثلة الإطارات السابقة، لاحظ التنوع في الإطارات، مع حيث النوع، أو الحجم، أو اللون.

انتبه! فإنه يمكنك أيضاً أن تنوع في حواف الإطارات المختلفة للعنصر الواحد بفضل الله، كما يلي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      h2{
        padding:15px;
        border-top:3px solid red;
        border-right:5px dotted blue;
        border-bottom:7px dashed gold;
        border-left:9px groove lightgreen;
      }
    </style>
  </head>
  <body style="padding:15px;">
    <h2>
      Welcome to CSS
    <br>
    Hello
    <br>
    Hi
    </h2>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ وجود تنوع مختلف في كل حافة من حواف الإطار، في نوع الإطار ولونه وحجمه.

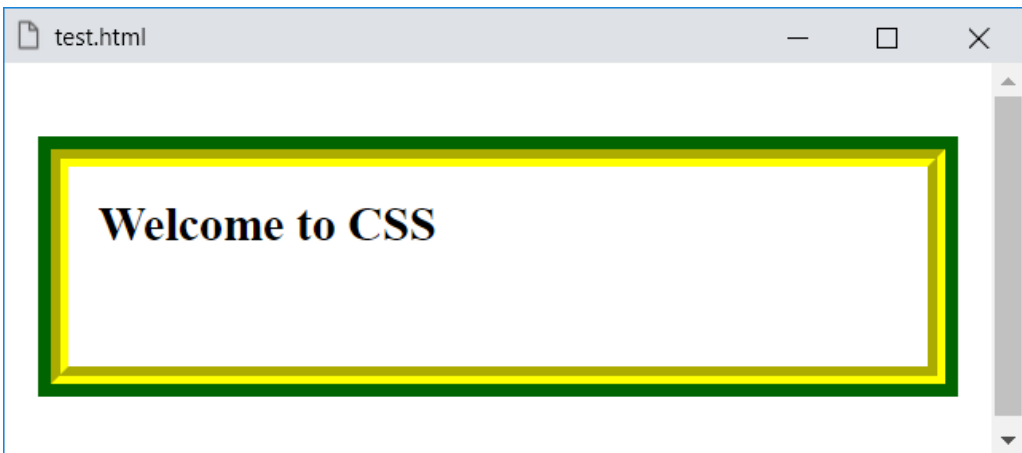
لاحظ أن الحشو padding للوسم h2 يظهر بين الإطار border له وبينه محتواه.

انتبه! فإنه يوجد خط خارجي للعناصر، مختلف عن الإطار الطبيعي واسمه outline، ويمكّن زيادته أيضاً للعنصر.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
h2{
height:70px;
padding:15px;
border:9px groove yellow;
outline:7px solid darkgreen;
}
</style>
</head>
<body style="padding:15px;">
<h2>Welcome to CSS</h2>
</body>
</html>
```

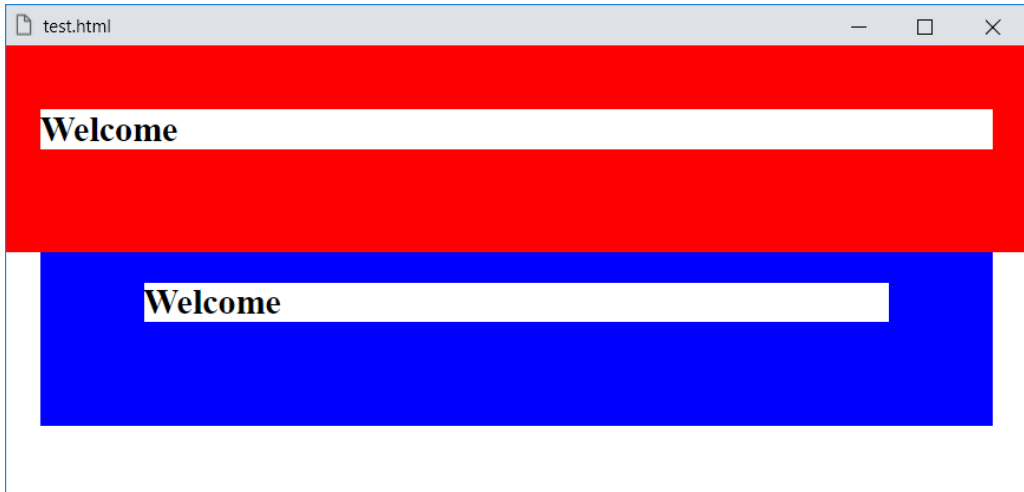
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ وجود خط خارجي outline بسمك 7px ونوع خط صلب solid ولونه أخضر غامق darkgreen.
انتبه! فإن الإطار تستطيع تفصيل حوافه مختلفة، أما الخط الخارجي فلا.
انتبه! فإن الإطار إذا كبد لا يخرج عن الصفحة، أما الخط الخارجي فيخرج ويغطي أيضاً على كل المحتويات بدون أي رعاية منه لذلك.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head></head>
  <body style="padding:15px;">
    <h2 style="outline:70px solid red">Welcome</h2>
    <h2 style="border:70px solid blue">Welcome</h2>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



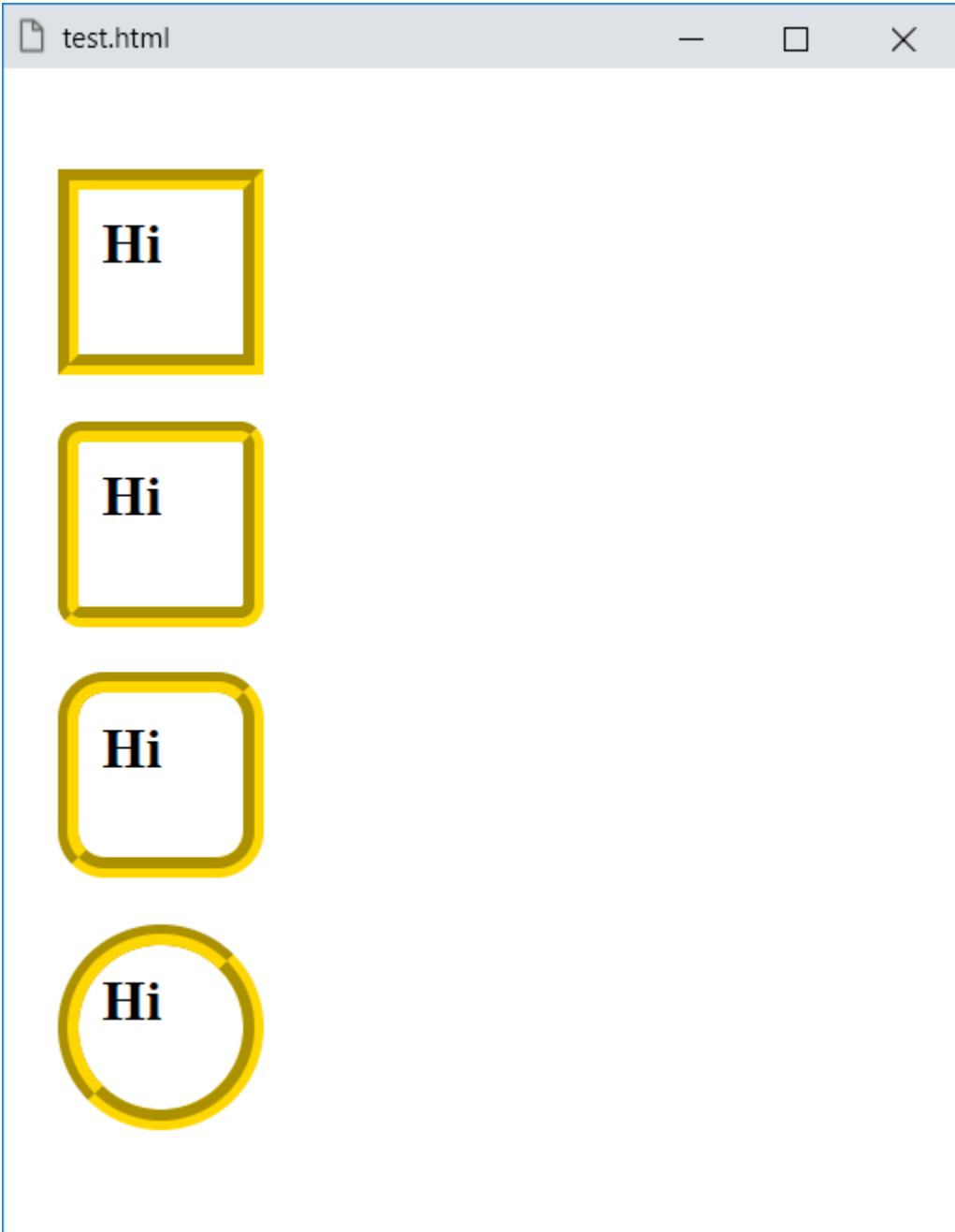
لاحظ في المثال السابق، أنه تم زيادة السمك إلى 70px لاختبار الإطار والخط الخارجي، واستجاب الإطار لعملية التكبير، أما الخط الخارجي، خرج عن الصفحة، وبدل وطغى على الإطار، فبسببه تم تغطية جزء من الإطار.

والآن سوف نبدأ مع تدوير الإطار radius أي نصف القطر من الجوانب الأربعة، وهم أعلى اليسار، وأعلى اليمين، وأسفل اليسار وأسفل اليمين.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  h2{
    padding:10px;
    width:50px;
    height:50px;
    border:9px groove gold;
  }
</style>
</head>
<body style="padding:15px;">
<h2>Hi</h2>
<h2 style="border-radius:10px;">Hi</h2>
<h2 style="border-radius:20px;">Hi</h2>
<h2 style="border-radius:45px;">Hi</h2>
</body>
</html>
```


وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

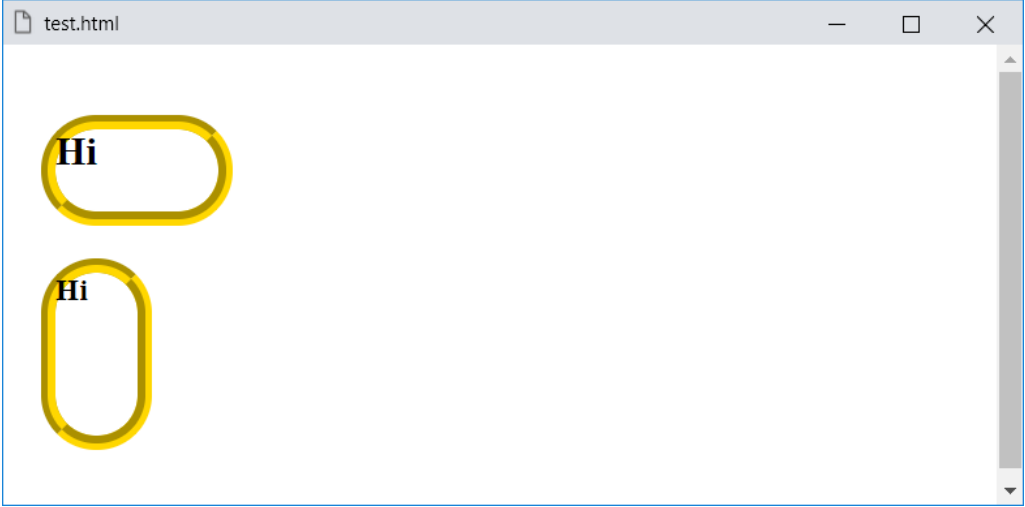


لاحظ أن أول وسم h2 لم يتم تدويره، ثم الذي يليه استدارا rounded مع الحواف الأربعة بمقدار 10px، ثم الذي يليه استدارا rounded مع الحواف الأربعة بمقدار 20px، ثم الذي يليه استدارا rounded مع الحواف الأربعة بمقدار 45px حتى وصل الى شكل دائري، وذلك لأن الطول موافق للعرض، أما لو كان غير ذلك فسيكون شكل بيضاوي طولي أو عرضي، وكل هذا بخاصية `border-radius`.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      h2{ width:100px; height:50px; }
      h3{ width:50px; height:100px; }
      h2,h3{
        border:9px groove gold;
      }
    </style>
  </head>
  <body style="padding:15px;">
    <h2 style="border-radius:35px;">Hi</h2>
    <h3 style="border-radius:35px;">Hi</h3>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن الوسم h2 عرضه أكبر من طوله فأحدث شكل بيضاوي أفقي بعد تخصيص خاصية border-radius له.

لاحظ أن الوسم h3 طوله أكبر من عرضه فأحدث شكل بيضاوي رأسي بعد تخصيص خاصية border-radius له.

لاحظ أنه عند ضبط الخاصية border-radius تتأثر كل الزوايا مع كل الجهات بالاستدارة مرة واحدة بمقدار واحد، وهو المقدار الذي تم تخصيصه للخاصية.

كما يمكنك أيضاً التحكم في استدارة كل زاوية وحدها بفضل الله، كما يلي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      h2,h3{
        height:50px; border:9px ridge red;
      }
      h2{
        border-top-left-radius:25px;
        border-top-right-radius:25px;
        margin-bottom:0;
      }
      h3{
        border-bottom-left-radius:25px;
        border-bottom-right-radius:25px;
        margin-top:0;
      }
    </style>
  </head>
  <body style="padding:15px;">
    <h2></h2>
    <h3></h3>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم التحكم بفضل الله، في كل زاوية وحدها، بالخصائص المحددة،
 فالخاصية `border-top-left-radius` أعلي يساراً
 والخاصية `border-top-right-radius` أعلي يميناً
 والخاصية `border-bottom-left-radius` أسفل يساراً
 والخاصية `border-bottom-right-radius` أسفل يميناً
 ليكون التحكم في العرض أكثر قوة وتفصيل بفضل الله.

وبعد كل هذا التفصيل بفضل الله رجاء **انتبه!** لأنه يمكنك تفصيل الإطار كل جزء كما تحب، ويمكنك أيضاً ضبطه مرة واحدة، حسب الرغبة والاحتياج.

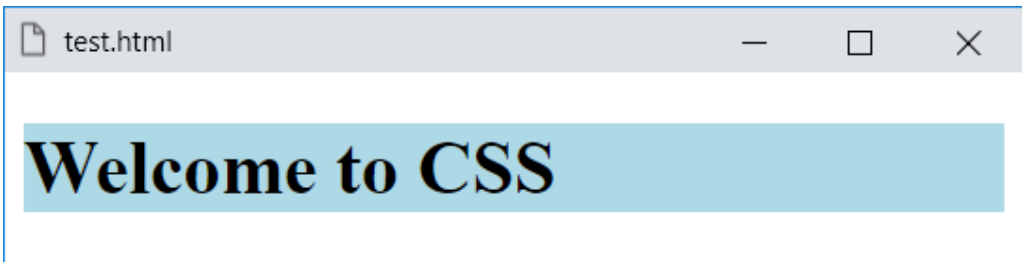
اللون والخلفية Color and Background

لون الخلفية background-color

تغيير لون خلفية العناصر وخاصية background-color:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      h1{ background-color:lightblue; }
    </style>
  </head>
  <body>
    <h1>Welcome to CSS</h1>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم تغيير لون الخلفية للوسم h1 عن طريق تخصيص قيمة اللون lightblue للخاصية background-color.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      h1{
        background-color:navy;
        color:lightblue;
      }
    </style>
  </head>
  <body>
    <h1>Welcome to CSS</h1>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



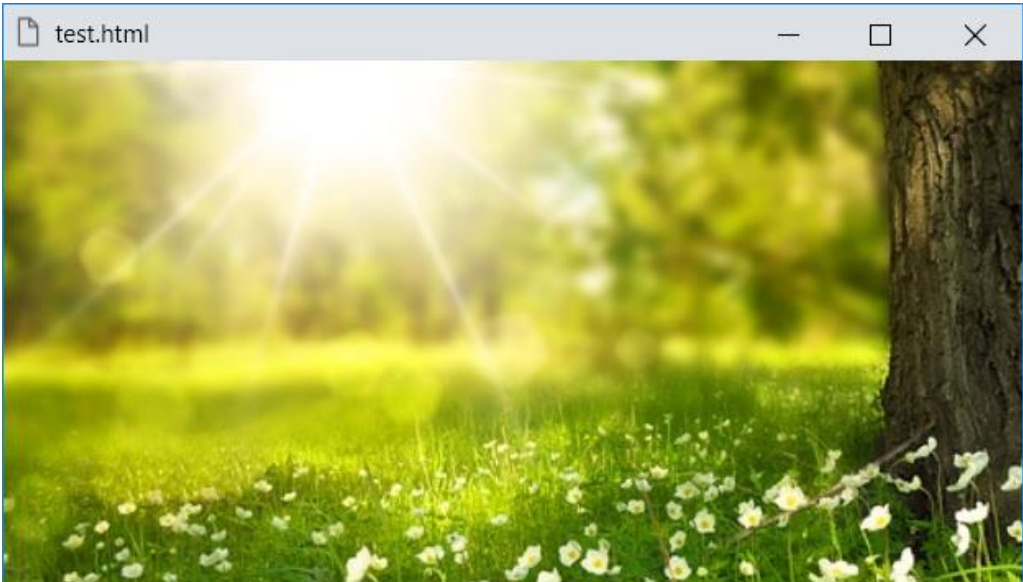
لاحظ أنه تم ضبط لون الخلفية للوسم h1 على اللون navy وتم ضبط لون النص لنفس الوسم على اللون lightblue.

صورة الخلفية والتدرج background-image

تغيير صورة الخلفية وخاصية background-image:
وباستخدام الصورة tree.jpg قم بكتابة الكود! التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    body{
      background-image:url("tree.jpg");
    }
  </style>
</head>
<body>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن العنصر المقصود بالتنسيق هنا هو العنصر body أي الصفحة نفسها، أي أننا نريد تغيير أو تعيين صورة الخلفية لتظهر في الصفحة للوسم .body.

لاحظ أنه للتحكم في صورة الخلفية للعناصر يتم استخدام الخاصية .background-image

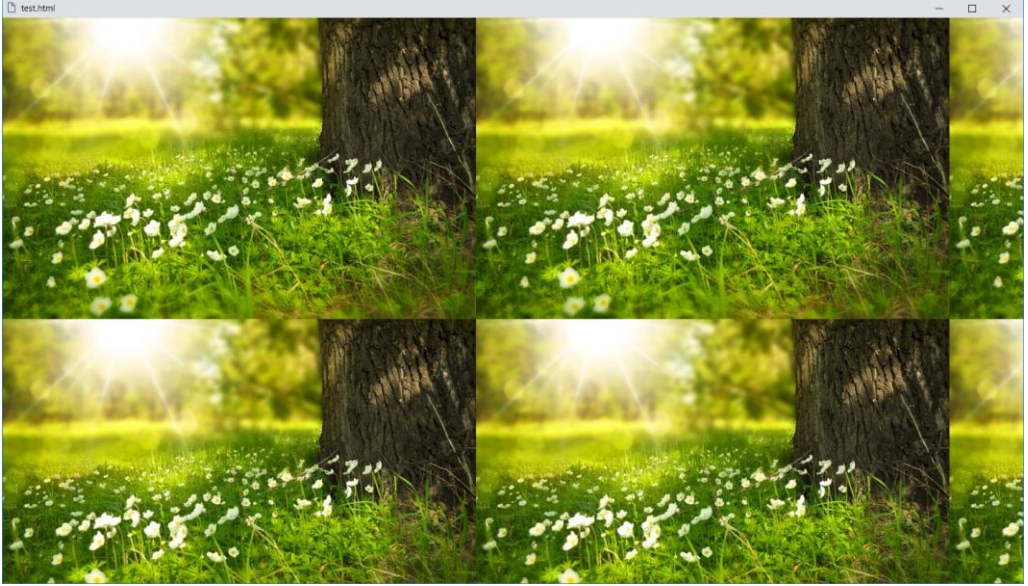
لاحظ أن الخاصية background-image تأخذ قيمة لمسار صورة، مع طريق الدالة url، ثم بعدها قوسيه () وبداخل القوسيه رمز اقتباس "" وبداخل الرمز الاقتباس اسم أو مسار الصورة، ليكون ناتج ذلك هو التالي:

```
background-image:url("tree.jpg")
```

لتعيين صورة خلفية للوسم body فتكون خلفية للصفحة.

انتبه! فإنه يمكنك وضع اسم الصورة بدون الرمز الاقتباس "" ولكنه له تعمل عند وجود مسافات في الاسم مثل "my tree.jpg"، فلذلك تأكد أنه من الأفضل استخدام الرمز الاقتباس "" دائماً في هذه الحالة.

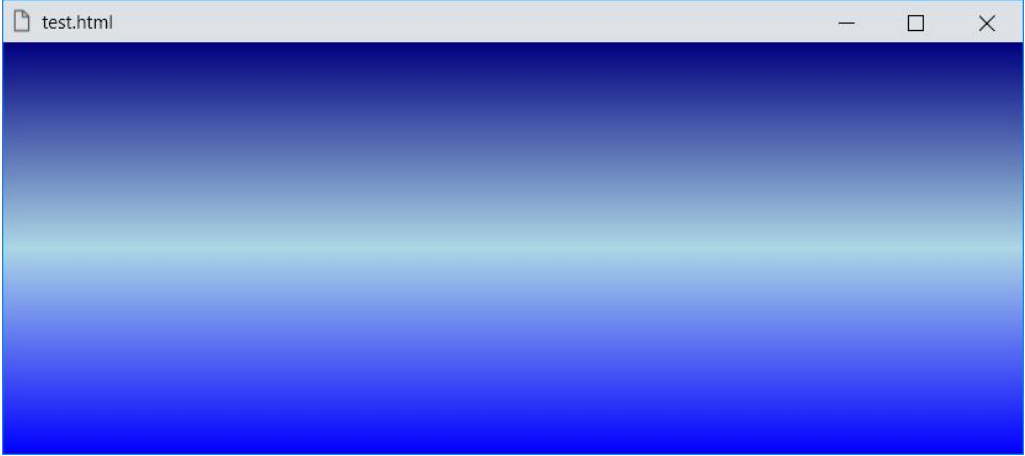
انتبه! فإن الصورة المعروضة أمامك تتكرر إذا زاد المتصفح عن حجمها الأصلي طويلاً أو عرضاً، ولملاحظة ذلك، أنظر إلى الشكل التالي بعد تكبير نافذة المتصفح:



يملك تخصيص تدرج لوني خطي الى لون الخلفية، مع تخصيص خاصية الطول للوسم body لتحديد إلى أيه ينتهي التدرج، ثم يتكرر كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body{
        height:250px;
        background-image:linear-gradient(navy,lightblue,blue);
      }
    </style>
  </head>
  <body>
  </body>
</html>
```

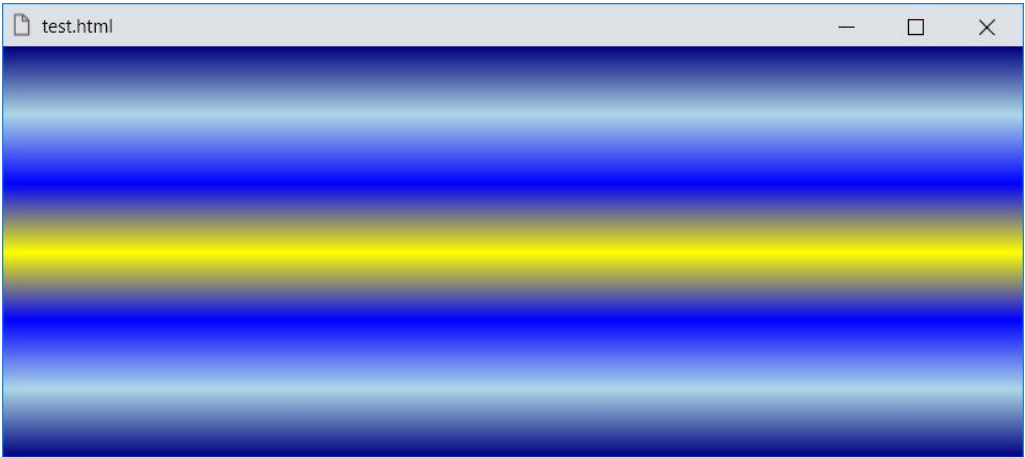
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم استخدام الدالة linear-gradient لإنشاء تدرج لوني خطي، وتم كتابة ثلاث ألوان بداخل الأقواس، مع العلم أنك تستطيع زيادة هذه الألوان كما تريد، ولو تم استخدام ألوان مختلفة فسيغير الشكل، مثلاً لو تم استخدام القيم التالية:

(navy, lightblue, blue, yellow, blue, lightblue, navy)

فسيلوون الشكل كما يلي:



عند استخدام التدرج الخطي، يمكنه تحديد الاتجاه، كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{
    height:250px;
    background-image:linear-gradient(to right,red,blue);
  }
</style>
</head>
<body>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



عند استخدام التدرج الخطي، يمكن تحديد درجة الزاوية، كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{
    height:250px;
    background-image:linear-gradient(290deg,red,blue);
  }
</style>
</head>
<body>
</body>
</html>
```

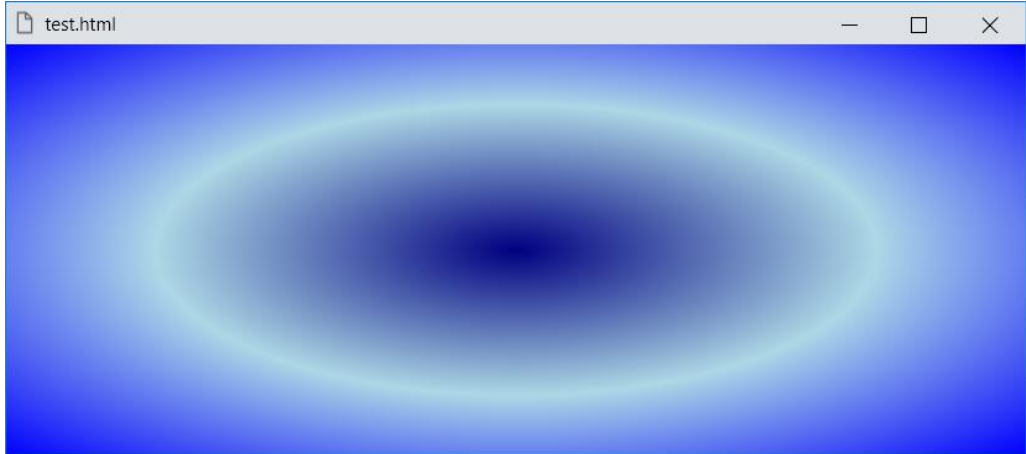
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



يملك تخصيص تدرج لوني شعاعي الى لون الخلفية، مع تخصيص خاصية الطول للوسم body لتحديد إلى أيه ينتهي التدرج، ثم يتكرر كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{
    height:250px;
    background-image:radial-gradient(navy,lightblue,blue);
  }
</style>
</head>
<body>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

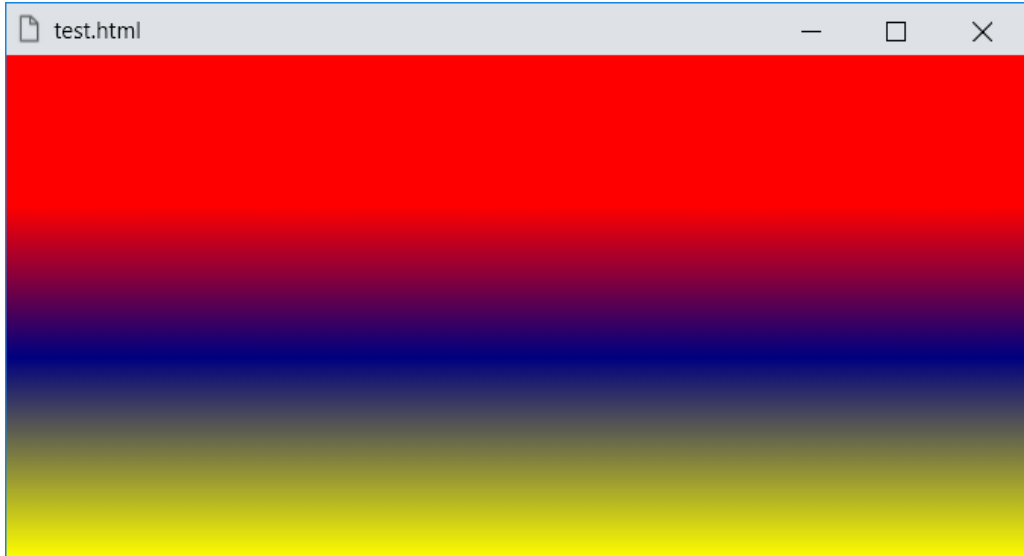


لاحظ أنه تم استخدام الدالة radial-gradient لإنشاء تدرج لوني شعاعي بثلاث ألوان، ويمكّن زيادة الألوان.

وعند استخدام التدرج الخطي والشعاعي نستطيع تحديد النسب تصاعدياً،
كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{
    height:250px;
    background-image:linear-gradient(red 30%,navy 60%,yellow 100%);
  }
</style>
</head>
<body>
</body>
</html>
```

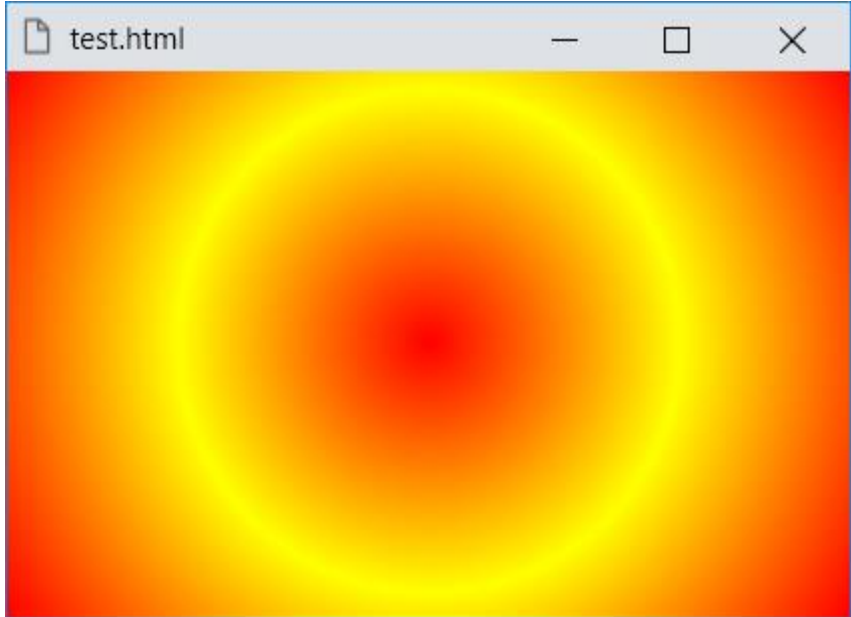
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



عند استخدام التدرج الشعاعي، يملك تحديده كدائرة، كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{
    height:200px;
    background-image:radial-gradient(circle, red, yellow, red);
  }
</style>
</head>
<body>
</body>
</html>
```

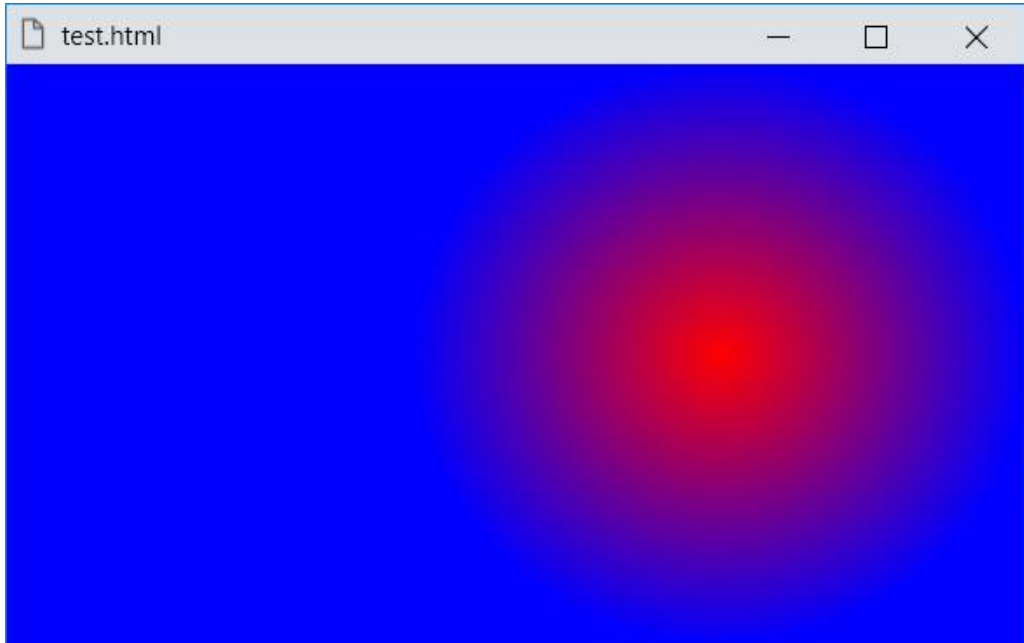
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



انتبه! عند استخدام التدرج الشعاعي، يمكن تحديد نسبة الجانب، كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{
    height:250px;
    background-image:radial-gradient(closest-side at 70%,red,blue);
  }
</style>
</head>
<body>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

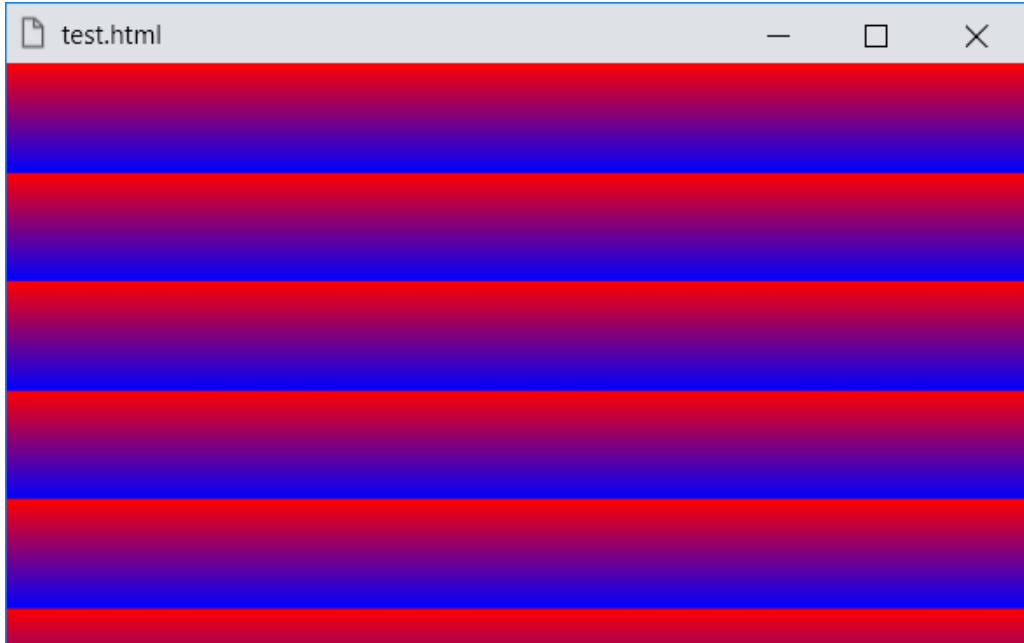


هذا الجانب الأفقي، وباستخدام 70% 70% closest-side سننقص الجانب الأفقي والرأسي معاً.

فاءك انءءاء ءءء لونف ءءف مءءءر؁ بالءالة repeating-linear-gradient
؁ كما فلف:
ءوء! قم بءءابة الكوء ءءالف ءم اءفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{
    height:250px;
    background-image:repeating-linear-gradient(red 50px,blue 100px);
  }
</style>
</head>
<body>
</body>
</html>
```

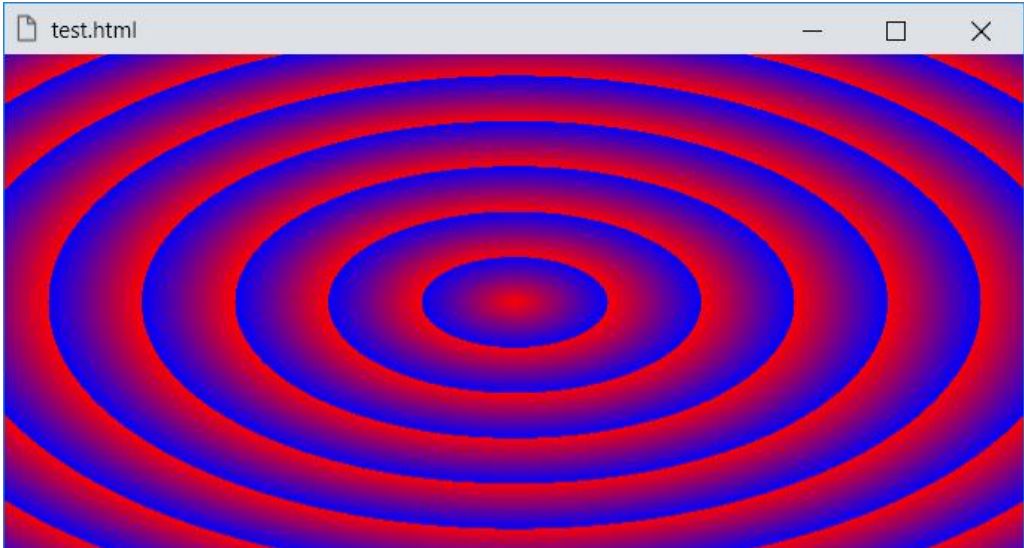
وبء ءءءفء الصفاء ءظءه كما بالشءك ءءالف:



يمكنك انشاء تدرج شعاعي متكرر، بالدالة `repeating-radial-gradient` كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{
    height:250px;
    background-image:repeating-radial-gradient(red 50px,blue 100px);
  }
</style>
</head>
<body>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



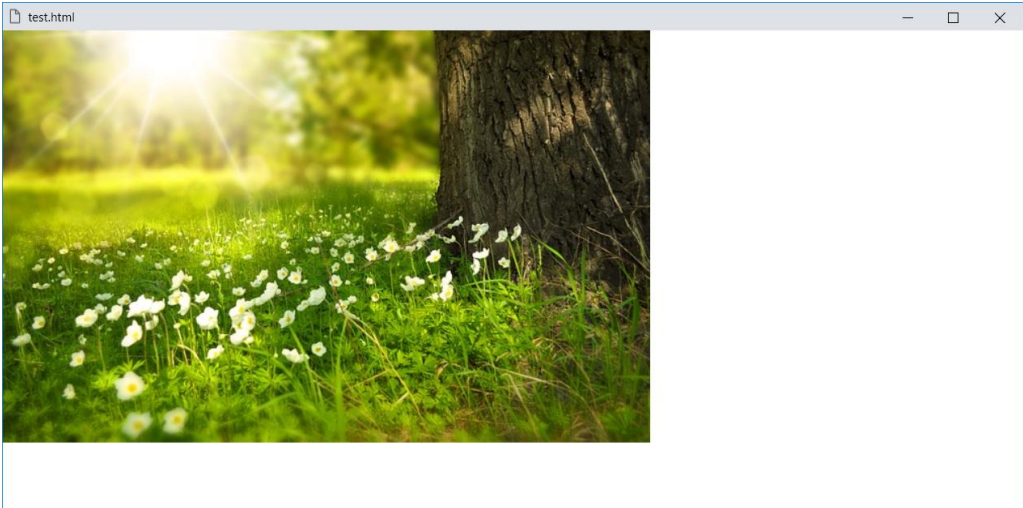
ءءار الخلففة background-repeat

ءءار صورة الخلففة وءاصفة background-repeat:

ءوء! قم بءءابة الكوء الألفف ءم اءفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    body{
      background-image:url("tree.jpg");
      background-repeat:no-repeat;
    }
  </style>
</head>
<body>
</body>
</html>
```

وءء ءءغفل الصفة ءظهم كما بالشكل الألفف:



لاحظ أنه يمكن التحكم في تكرر صورة الخلفية عن طريق الخاصية background-repeat بسهولة.

لاحظ أنه مع القيمة no-repeat لم يحدث تكرر، أفقي أو رأسي.

انتبه! فإن القيمة الافتراضية، هي repeat، أي تكرر كلي.

انتبه! لو تم استخدام القيمة repeat-x فسوف يحدث تكرر أفقي.

انتبه! لو تم استخدام القيمة repeat-y فسوف يحدث تكرر رأسي.

انتبه! لو تم استخدام القيمة space فسوف يحدث تكرر مع مسافات.

انتبه! لو تم استخدام القيمة round فسوف يحدث تكرر بدون مسافات.

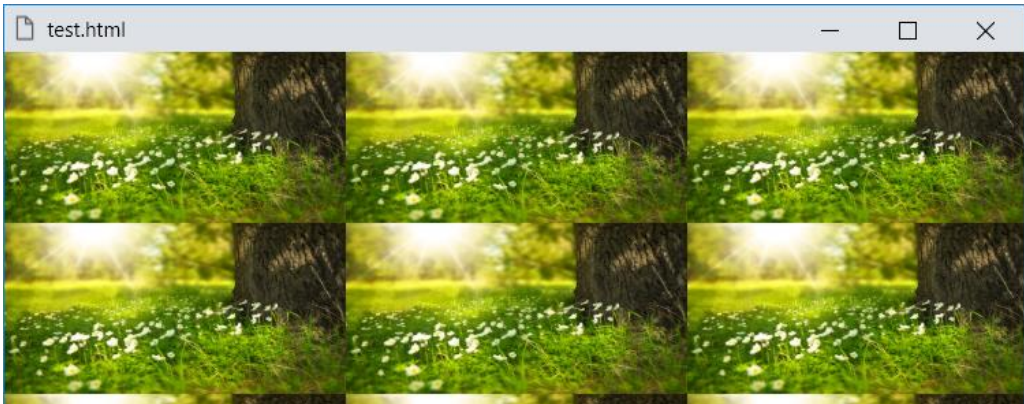
ءءم الخلففة background-size

ءءم صورة الخلففة وءاصفة background-size:

ءوء! قم بءءابة الكوء الألفف ءم اءفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    body{
      background-image:url("tree.jpg");
      background-size:200px 100px;
    }
  </style>
</head>
<body>
</body>
</html>
```

وءء ءءءعل الصفءة ءظهم ءم بالشءك الألفف:

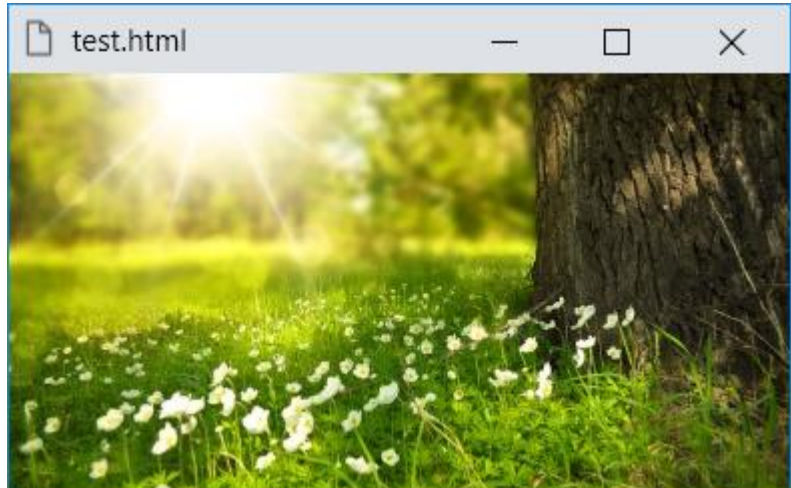


لاءظ أه الءاصفة background-size للوءم body آءءء
الءفة 200px 100px، ومعناها 200 للعرض و100 للءول.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body{
        background-image:url("tree.jpg");
        background-size:cover;
      }
    </style>
  </head>
  <body></body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



ولو تم ءءبءرها، ءكون مءء الشءك الءالء:



ومء هنا نلاءظ ءأءبء القءمة cover على الءلففة، ءفء ءعلءها ءءظف الشاشء ءائماً، وءمءك اسءءءام القءمة contain للءأءء مء أن الصورة مرءبة بالءامل.

ارتباط أو تعلق الخلفية background-attachment

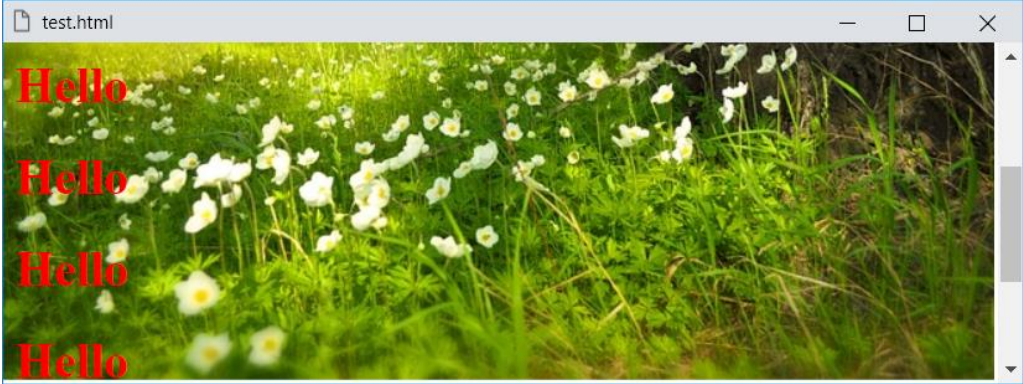
تعلق صورة الخلفية وخاصية background-attachment :
انتبه! فإنه مع أجل رؤية تأثير هذه الخاصية، لابد من وضع محتوى في صفحة الويب، بحيث يظهر شريط التمرير scroll حيث ننزل ونصعد، لنرى ما سوف يحدث، ولذلك سوف نختار الـ h1 ونلونه باللون الأحمر ليظهر على صورة الشجرة، مع العلم أننا لنستخدم background-attachment الآن، ليظهر التأثير، **انتبه!** رجاء التأكيد.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      h1{ color:red; }
      body{
        background-image:url("tree.jpg");
        background-repeat:no-repeat;
      }
    </style>
  </head>
  <body>
    <h1>Hello</h1><h1>Hello</h1><h1>Hello</h1>
    <h1>Hello</h1><h1>Hello</h1><h1>Hello</h1>
    <h1>Hello</h1><h1>Hello</h1><h1>Hello</h1>
  </body>
</html>
```

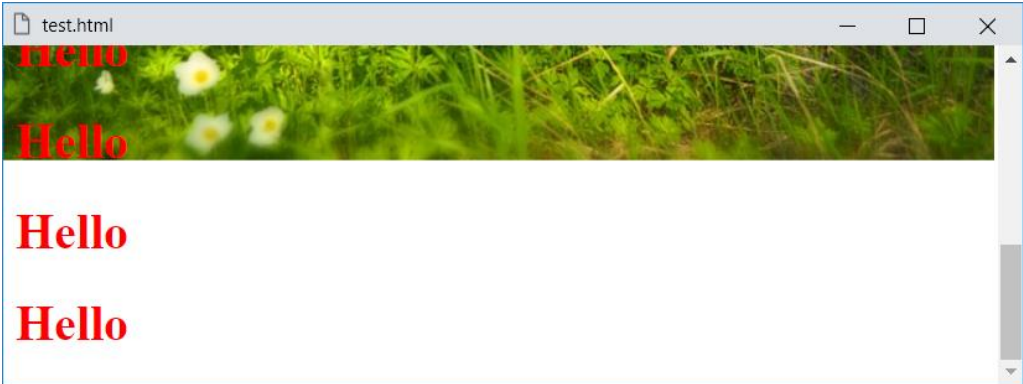
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



والآن نزل بشريط التمرير scroll ليكوه الشكل كما يلي:



بعد النزول لاحظ أنك تترك الصورة بالأعلى، وتذهب للأسفل، مثل التالي:



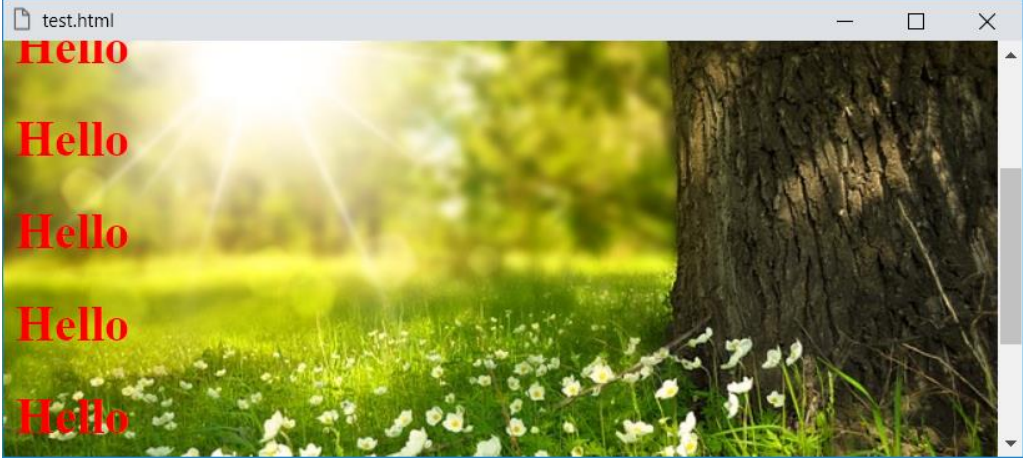
والآن سوف نرى تأثير الخاصية `background-attachment` ولكن
انتبه! جيداً وركز مع شريط التمرير `.scroll`.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1{ color:red; }
body{
background-image:url("tree.jpg");
background-repeat:no-repeat;
background-attachment:fixed;
}
</style>
</head>
<body>
<h1>Hello</h1><h1>Hello</h1><h1>Hello</h1>
<h1>Hello</h1><h1>Hello</h1><h1>Hello</h1>
<h1>Hello</h1><h1>Hello</h1><h1>Hello</h1>
</body>
</html>
```

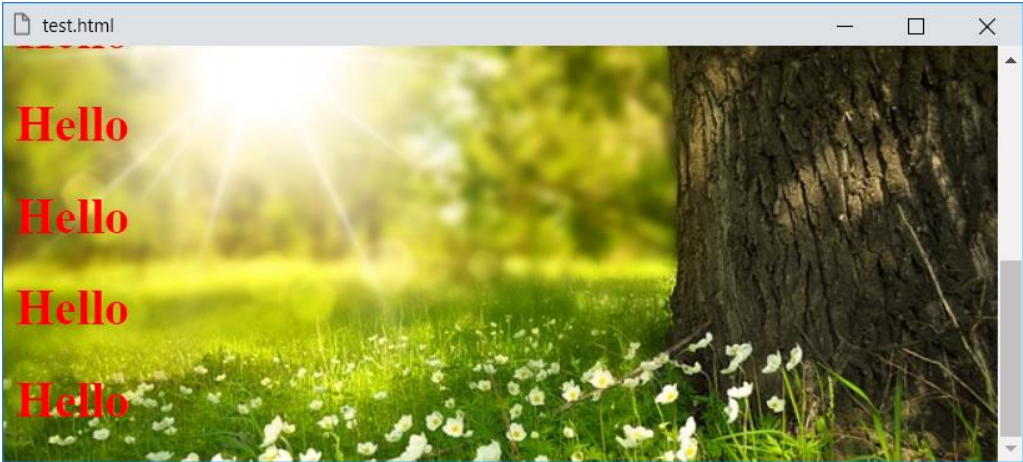
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



والآن نازل بشرفء الءمرف scroll لفاء الشءل كما فلف:



والآن نازل مرة آءرف بشرفء الءمرف scroll لفاء الشءل كما فلف:



لأء ثبات الصورة، بسبب اسءءءام القفمة fixed للءاصفة
background-attachment، مع العلم أن القفمة
scroll هف الافتراضفة، والقفمة local ءشبه scroll.

موضوع الخلفية background-position

موضوع صورة الخلفية وخاصية background-position:

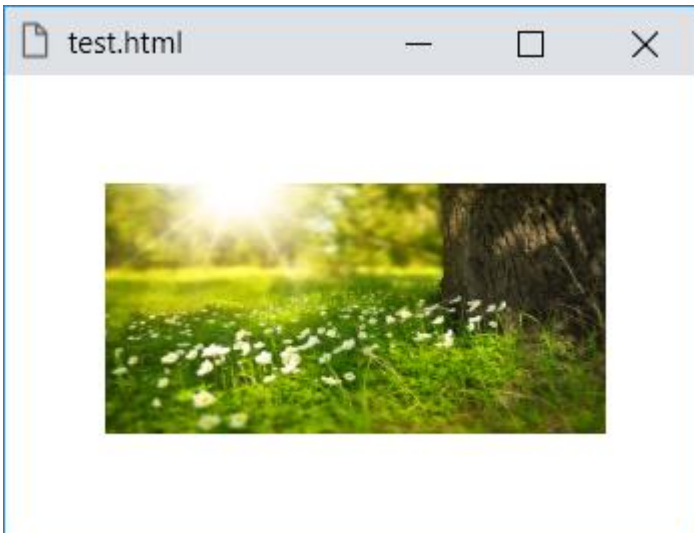
كود! قم بكتابة الكود التالي ثم احفظه:

```

<!DOCTYPE html>
<html>
<head>
<style>
  body{
    background-image:url("tree.jpg");
    background-repeat:no-repeat;
    background-size:200px 100px;
    background-attachment:fixed;
    background-position:center;
  }
</style>
</head>
<body></body>
</html>

```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

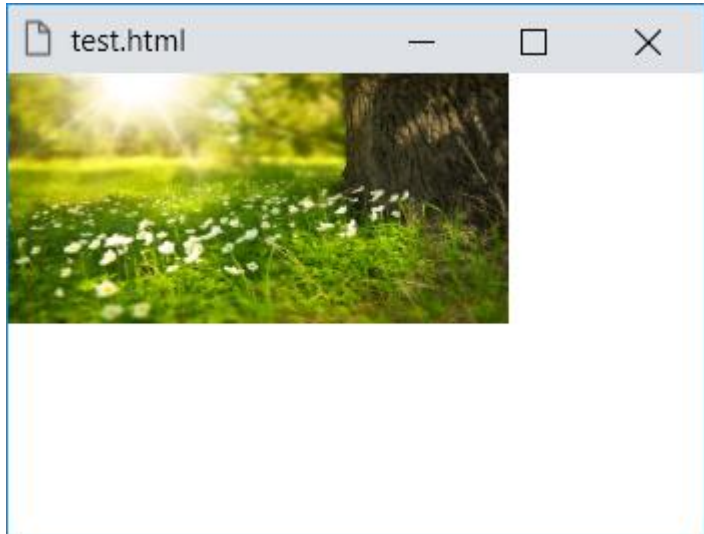


لاحظ أن الخلفية أصبحت في المنتصف بالقيمة center.

كوء! قم بءآابة الكوء الآلفف ثم اءفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{
    background-image:url("tree.jpg");
    background-repeat:no-repeat;
    background-size:200px 100px;
    background-attachment:fixed;
    background-position:left top;
  }
</style>
</head>
<body></body>
</html>
```

وءء ءءءفء الصفءة ءظءه كما بالشءك الآلفف:

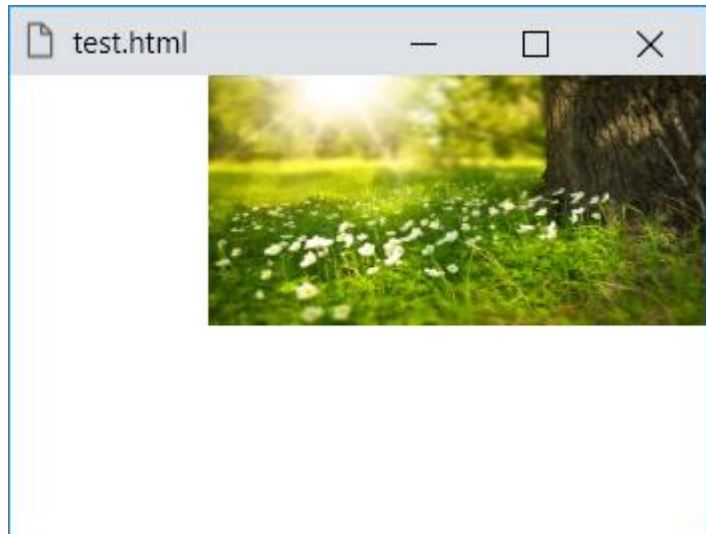


لاءظ أن الءلففة أءبءء في الأعلى يساراً ، بسبب القفمفءه left و top.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{
    background-image:url("tree.jpg");
    background-repeat:no-repeat;
    background-size:200px 100px;
    background-attachment:fixed;
    background-position:right top;
  }
</style>
</head>
<body></body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

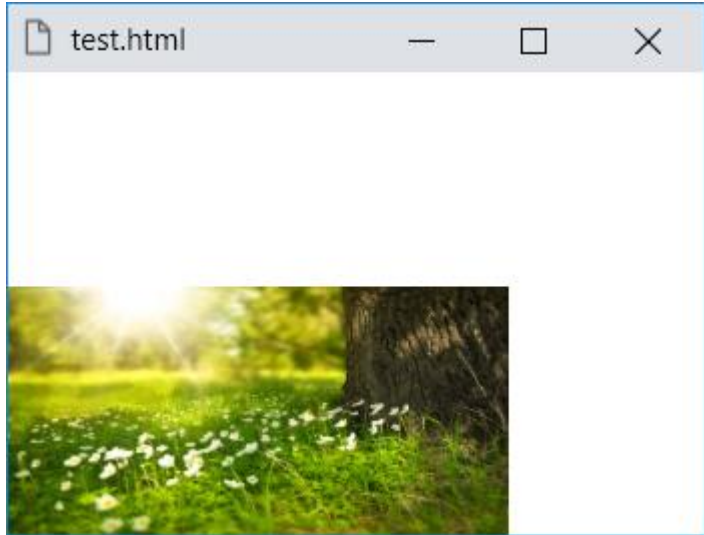


لاحظ أن الخلفية في الأعلى يميناً، بسبب القيمتين `top` و `right`.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{
    background-image:url("tree.jpg");
    background-repeat:no-repeat;
    background-size:200px 100px;
    background-attachment:fixed;
    background-position:left bottom;
  }
</style>
</head>
<body></body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

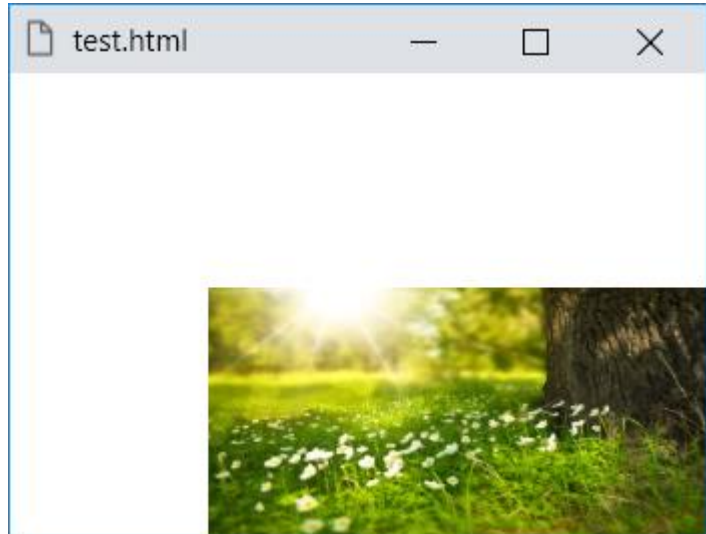


لاحظ أن الخلفية في الأسفل يساراً، بسبب القيمتين `bottom` و `left`.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{
    background-image:url("tree.jpg");
    background-repeat:no-repeat;
    background-size:200px 100px;
    background-attachment:fixed;
    background-position:right bottom;
  }
</style>
</head>
<body></body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

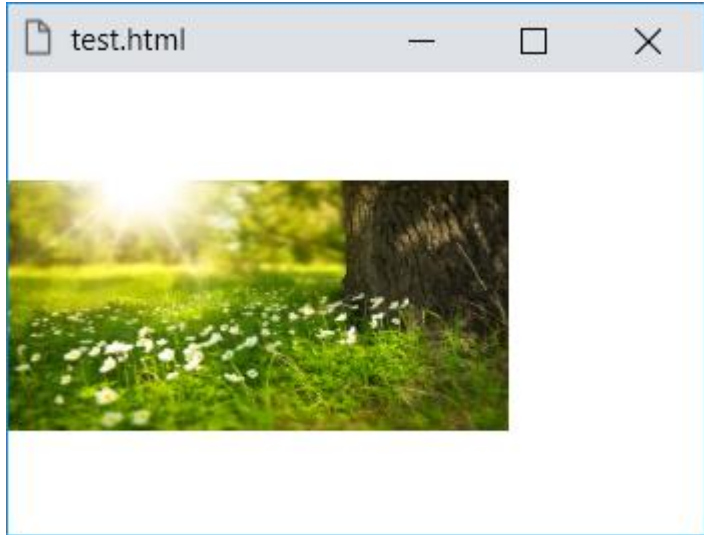


لاحظ الخلفية في الأسفل يميناً، بسبب القيمتين `bottom` و `right`.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{
    background-image:url("tree.jpg");
    background-repeat:no-repeat;
    background-size:200px 100px;
    background-attachment:fixed;
    background-position:left center;
  }
</style>
</head>
<body></body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

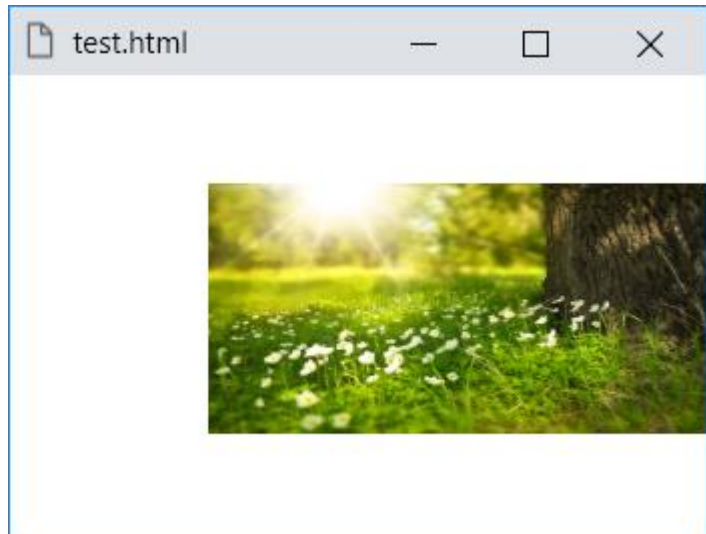


لاحظ الخلفية في الوسط يساراً ، بسبب القيمتين center و left.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{
    background-image:url("tree.jpg");
    background-repeat:no-repeat;
    background-size:200px 100px;
    background-attachment:fixed;
    background-position:right center;
  }
</style>
</head>
<body></body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

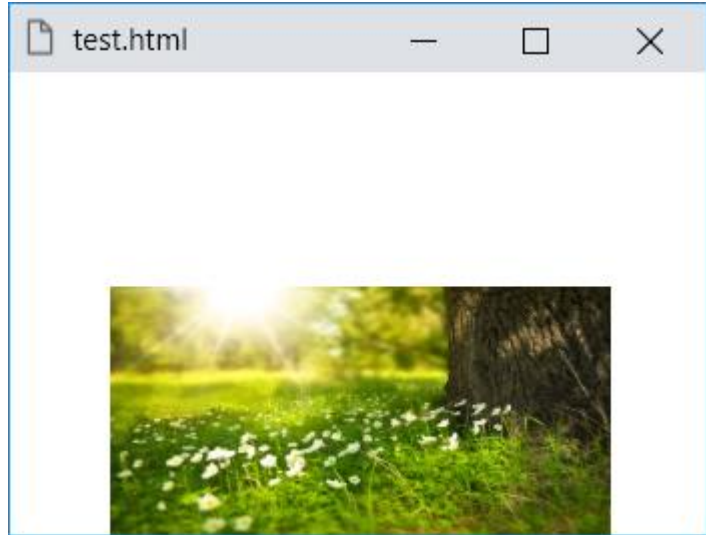


لاحظ الخلفية في الوسط يمينا، بسبب القيمتين center و right.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{
    background-image:url("tree.jpg");
    background-repeat:no-repeat;
    background-size:200px 100px;
    background-attachment:fixed;
    background-position:bottom center;
  }
</style>
</head>
<body></body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

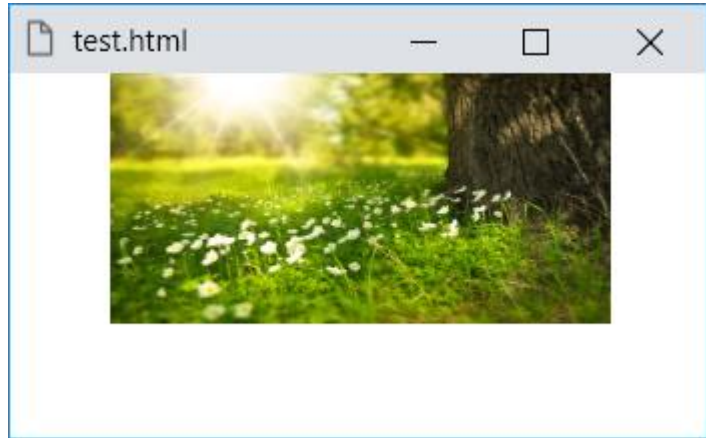


لاحظ الخلفية بالوسط والأسفل، بسبب القيمتيه bottom و center.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{
    background-image:url("tree.jpg");
    background-repeat:no-repeat;
    background-size:200px 100px;
    background-attachment:fixed;
    background-position:top center;
  }
</style>
</head>
<body></body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

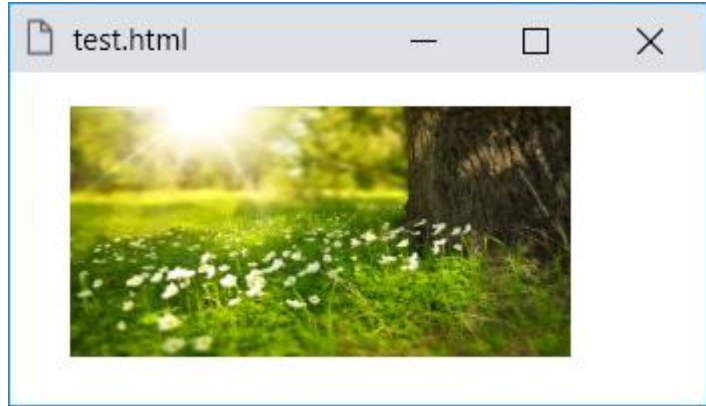


لاحظ الخلفية بالوسط والأعلى، بسبب القيمتين `top` و `center`.

ءوء! قم بءءابة الكوء الءاءف ءم اءفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{
    background-image:url("tree.jpg");
    background-repeat:no-repeat;
    background-size:200px 100px;
    background-attachment:fixed;
    background-position:30% 40%;
  }
</style>
</head>
<body></body>
</html>
```

وءء ءءءفء الصفاء ءظءه ءم بالشء الءاءف:



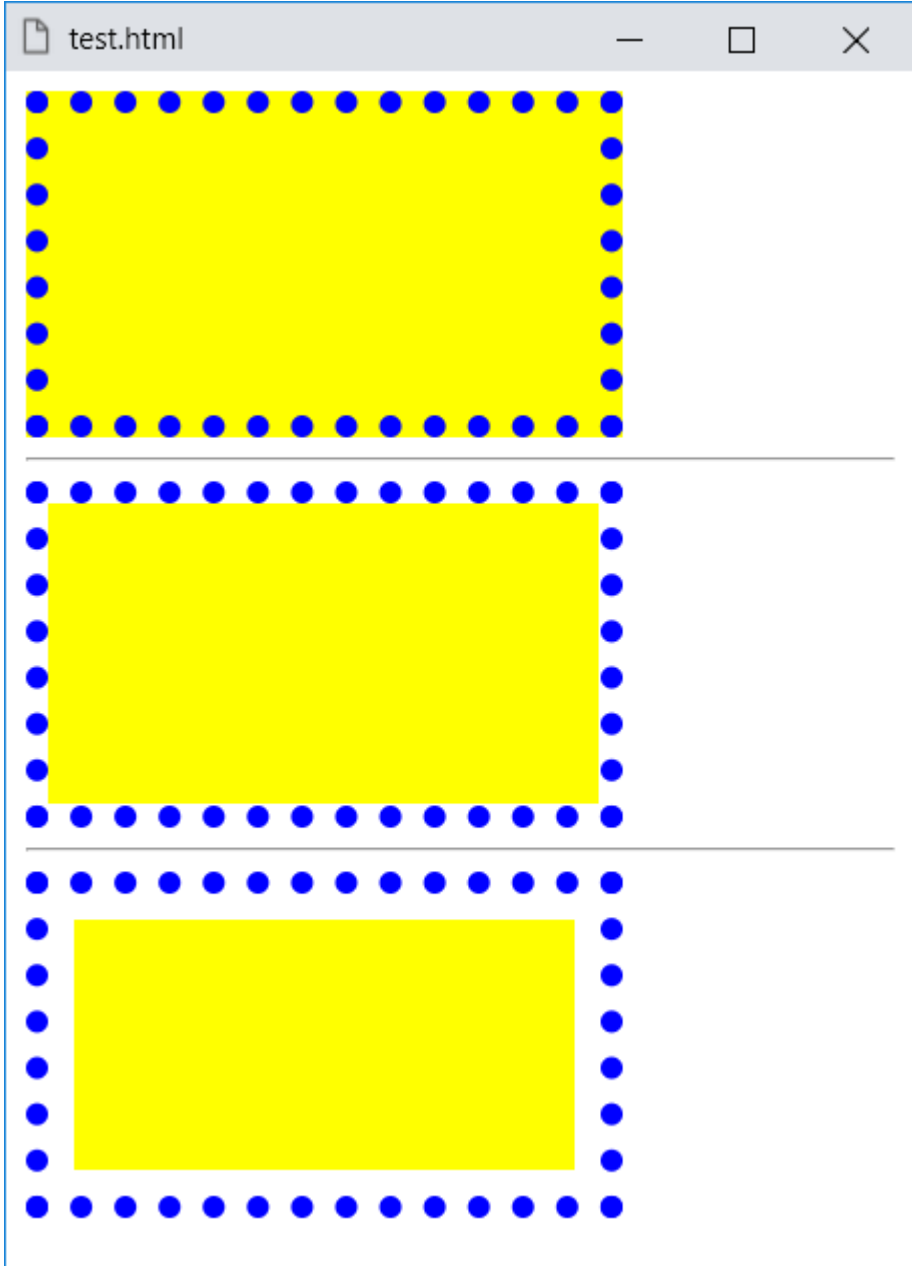
لاءظ أنه ءم ءءك 30% مء العرض مء نااءفة البءار، ولاءظ أنه ءم ءءك 40% مء الطوء مء النااءفة العلوءة، مء العلم أنه فمءءءءء بءفمة البفءسل أفضاً.

background-clip قصاصة الخلفية

نوع قصاصة صورة الخلفية وخاصة `:background-clip`
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    div{
      width:200px; height:100px; padding:10px;
      background-color:yellow;
      border:9px dotted blue;
    }
    #d1{
      background-clip:border-box;
    }
    #d2{
      background-clip:padding-box;
    }
    #d3{
      background-clip:content-box;
    }
  </style>
</head>
<body>
  <div id="d1"></div><hr>
  <div id="d2"></div><hr>
  <div id="d3"></div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم تحديد عرض width و طول height وحشو padding ولون خلفية background-color وإطار منقط dotted كيبير ليظهر ما خلفه، وكل ذلك لعمل إظهار لتأثير القصاصه بالكامل.

لاحظ أنه مع القيمة border-box الشكل كما هو، وذلك لأن القيمة border-box هي الافتراضية.

لاحظ أنه مع القيمة padding-box يظهر الإطار خارج الخلفية.

لاحظ أنه مع القيمة content-box يظهر الإطار خارج الخلفية، وخارج الحشو أيضاً.

انتبه! فإن هناك خاصية أخرى تفعل نفس الشيء، وتأخذ نفس القيم، ولكنها تأخذ من أصل الصورة، وهي background-origin، وتفيد عندما تكون أصل الخلفية صورة.

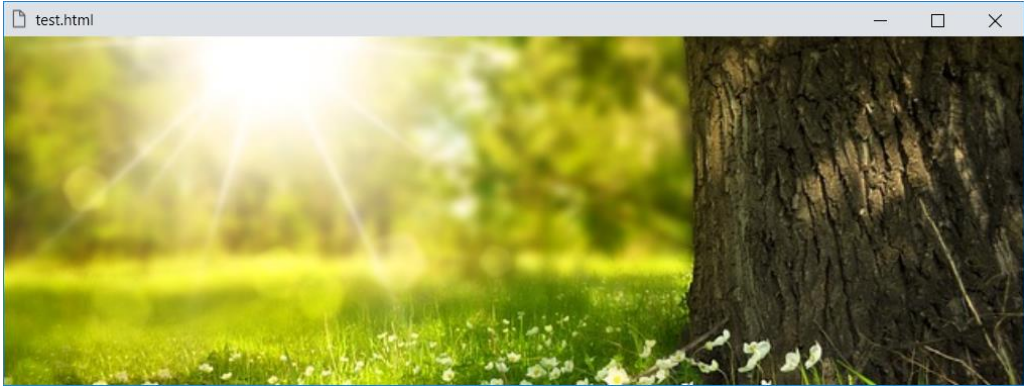
مزء الخلففاء background-blend-mode

مزء الخلففاء وءاصفة background-blend-mode، وهف ممءعة ءءاً، وستشعر كأنك ءصء مؤءراء بفءامء الفوءوشوب بسهولة:

ءعودنا مع قبل معء ءءابة الكوء ءءالف:

```
background-image:url('tree.jpg');
```

فظهر الشءل ءءالف:



وعء ءءابة الكوء ءءالف:

```
background-image:linear-gradient(to left,yellow,green);
```

فظهر الشءل ءءالف:

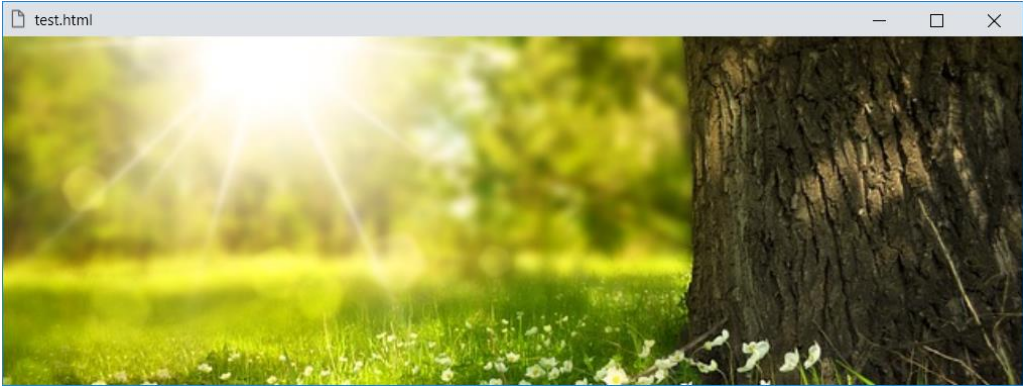


ولكن الآن، هل يمكن أن نقوم بعمل مزيج أو خليط بين الخلفيتين، أي بين خلفية التدرج اللوني، وخلفية الصورة؟!، الإجابة هي، نعم بفضل الله، وذلك عن طريق خاصية `background-blend-mode`.

والآن، لو تم كتابة الكود التالي:

```
background-image:url('tree.jpg'),linear-gradient(to left,red,blue);
```

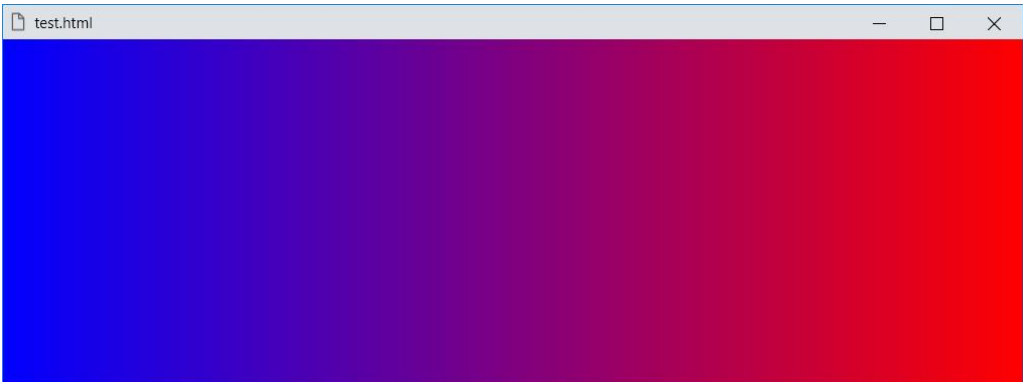
ستظهر خلفية الصورة، لأنها أولاً، كما يلي:



ولو تم كتابة الكود التالي:

```
background-image:linear-gradient(to left,red,blue),url('tree.jpg');
```

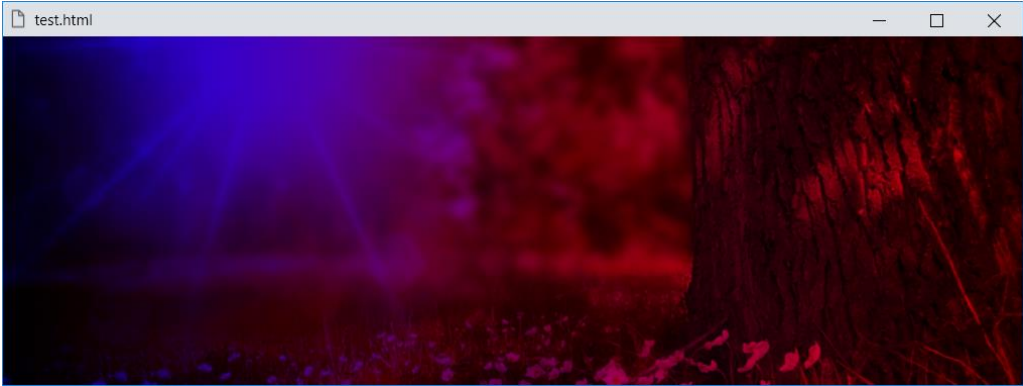
ستظهر خلفية التدرج اللوني، لأنها أولاً، كما يلي:



الوضع السابق هو normal، وهو الأولي فالأولي، وهو الافتراضي.
ولكن مع الوضع multiply، وندب الألوان يختلف الأمر كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{
    background-size:cover;
    background-image:linear-gradient(to left,red,blue),
      url('tree.jpg');
    background-blend-mode:multiply;
  }
</style>
</head>
<body> </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

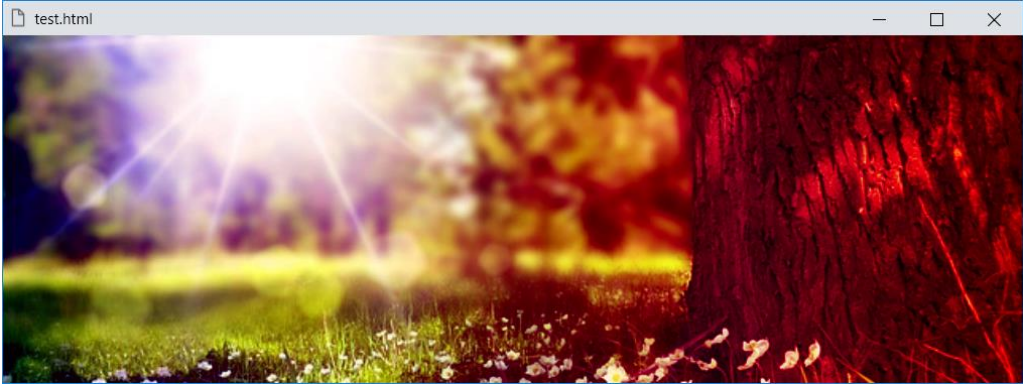


لاحظ أنه تم المزج بين التدرج اللوني، وصورة الخلفية، حيث الناتج الأخير هو نتيجة ندب الألوان العلوية والسفلية، أي طبقة سوداء تؤدي إلى مثلها، وطبقة بيضاء تؤدي إلى أي تغيير، فيؤثر كصورتيه بفيلم شفاف متداخل.

ومع الوضع overlay، والتغطية يختلف الأمر كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
body{
background-size:cover;
background-image:linear-gradient(to left,red,blue),
url('tree.jpg');
background-blend-mode:overlay;
}
</style>
</head>
<body> </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

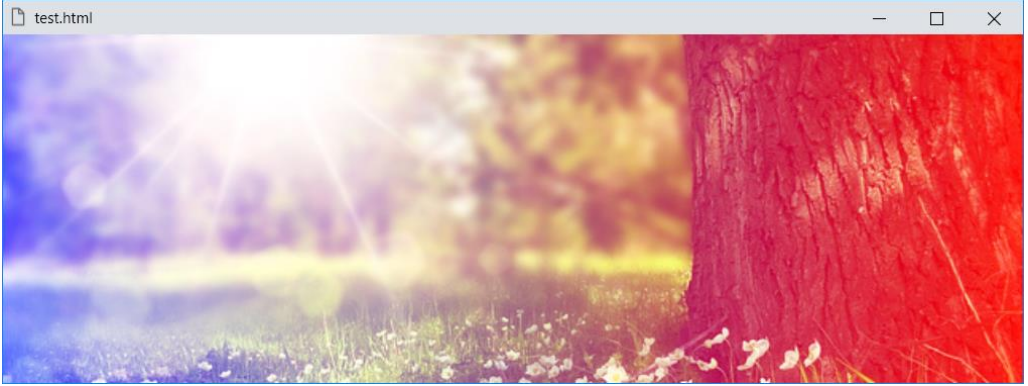


لاحظ المزج بين التدرج، والصورة، حيث الناتج الأخير هو نتيجة الضرب إذا كان اللون السفلي داكن، أو الشاشة إذا كان اللون السفلي ساطع، مع الضوء الثابت وتبديل الطبقات.
ومع الوضع screen، والشاشة يختلف الأمر كما يلي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{
    background-size:cover;
    background-image:linear-gradient(to left,red,blue),
      url('tree.jpg');
    background-blend-mode:screen;
  }
</style>
</head>
<body> </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



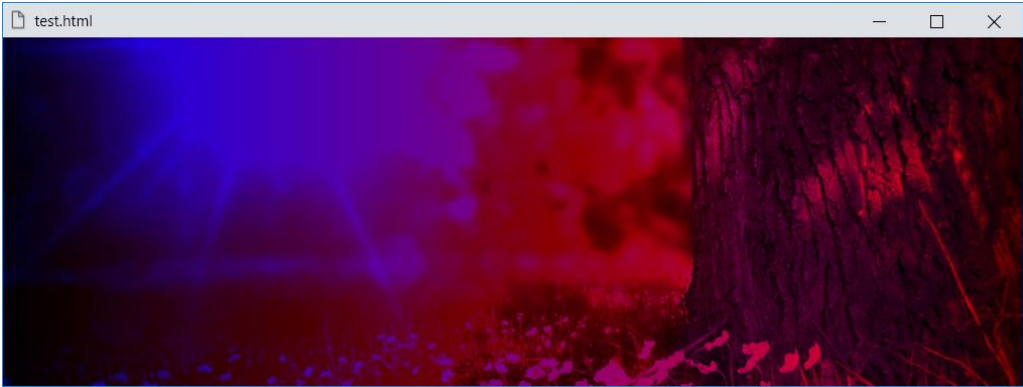
لاحظ المزج بين التدرج، والصورة، حيث الناتج الأخير هو نتيجة قلب الألوان وضربها وعكس تلك القيمة، ولا تؤدي الطبقة السوداء إلى أي تغيير، وتؤدي الطبقة البيضاء إلى طبقة نهائية بيضاء، فيكون التأثير هو مثل صورتين مشرقتين على شاشة العرض.

ومع الوضع `darken`، والأدكن يختلف الأمر كما يلي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{
    background-size:cover;
    background-image:linear-gradient(to left,red,blue),
      url('tree.jpg');
    background-blend-mode:darken;
  }
</style>
</head>
<body> </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

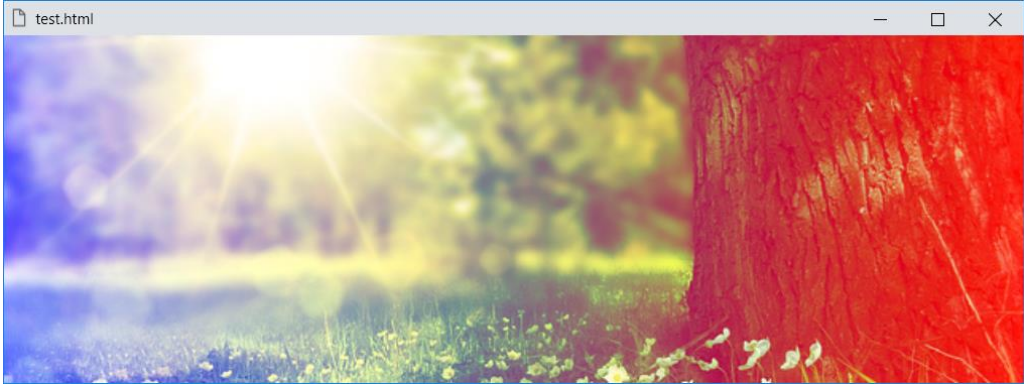


لاحظ المزج بين التدرج، والصورة، حيث الناتج الأخير هو توحيد الألوان الداكنة.

ومع الوضع lighten، والأسطح يختلف الأمر كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{
    background-size:cover;
    background-image:linear-gradient(to left,red,blue),
      url('tree.jpg');
    background-blend-mode:lighten;
  }
</style>
</head>
<body> </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

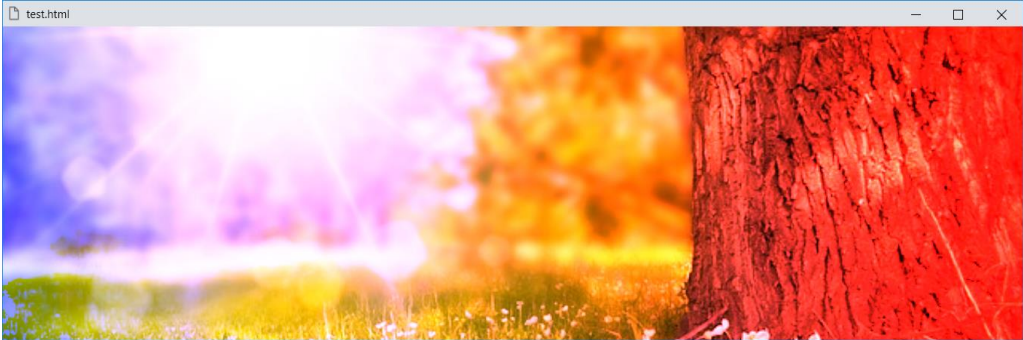


لاحظ المزج بين التدرج، والصورة، حيث الناتج الأخير هو توحيد الألوان الساطعة.

ومع الوضع color-dodge، وإنقاص الكثافة يختلف الأمر كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{
    background-size:cover;
    background-image:linear-gradient(to left,red,blue),
      url('tree.jpg');
    background-blend-mode:color-dodge;
  }
</style>
</head>
<body> </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

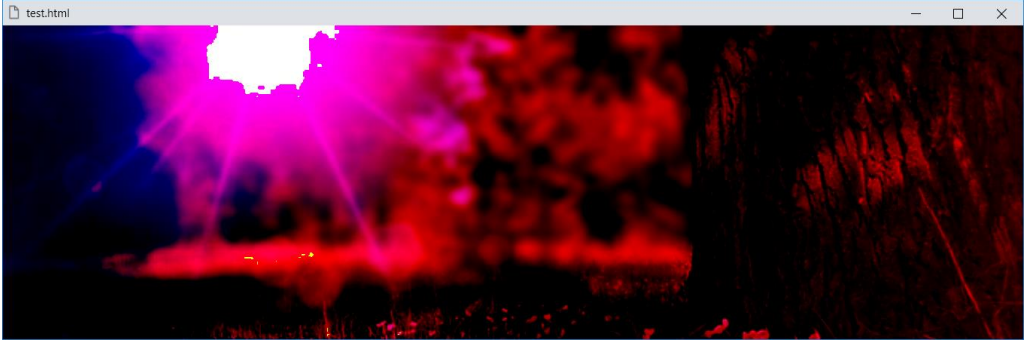


لاحظ المزج بين التدرج، والصورة، حيث الناتج الأخير هو نتيجة قسمة اللون السفلي على عكس اللون العلوي، فالطبقة السوداء لا تؤدي إلى أي تغيير، وتكون المقدمة ذات اللون العكسي للخلفية إلى لون مضىء بالكامل.

ومع الوضع color-burn، وحرق الألوان يختلف الأمر كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
body{
background-size:cover;
background-image:linear-gradient(to left,red,blue),
url('tree.jpg');
background-blend-mode:color-burn;
}
</style>
</head>
<body> </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

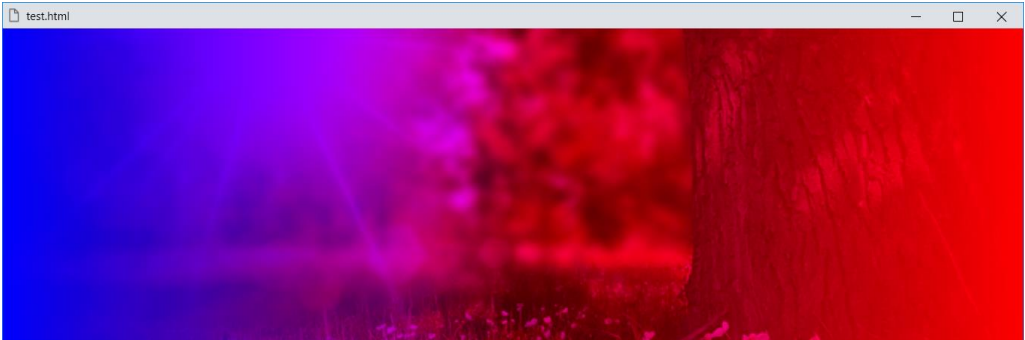


لاحظ المزج بين التدرج، والصورة، حيث الناتج الأخير هو نتيجة عكس اللون السفلي، وقسمة القيمة على اللون الأعلى، وعكس تلك القيمة، حيث الطبقة البيضاء لا تؤدي إلى أي تغيير، وتؤدي الطبقة ذات اللون العكسي للخلفية إلى صورة نهائية سوداء، فيظهر وكأنه محروق.

ومع الوضع hard-light، والضوء الخشبي يختلف الأمر كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{
    background-size:cover;
    background-image:linear-gradient(to left,red,blue),
      url('tree.jpg');
    background-blend-mode:hard-light;
  }
</style>
</head>
<body> </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

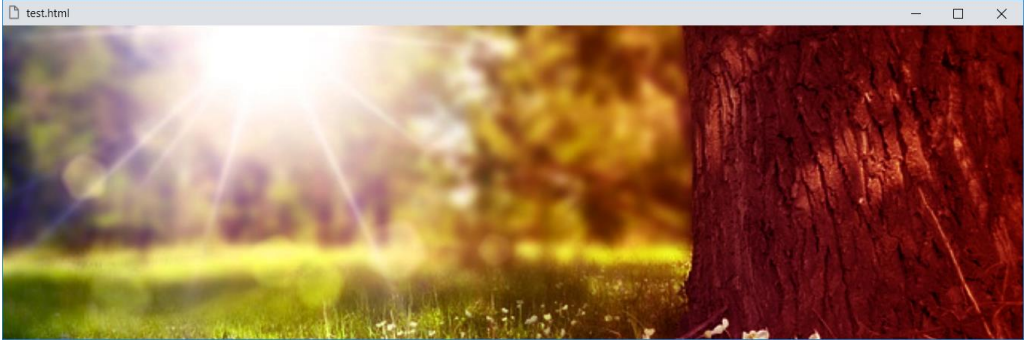


لاحظ المزج بين التدرج، والصورة، حيث الناتج الأخير هو نتيجة الضرب إذا كان اللون الأعلى أغمق أو الشاشة إذا كان اللون الأعلى أخف.

ومع الوضع soft-light، والضوء الناعم يختلف الأمر كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
body{
background-size:cover;
background-image:linear-gradient(to left,red,blue),
url('tree.jpg');
background-blend-mode:soft-light;
}
</style>
</head>
<body> </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

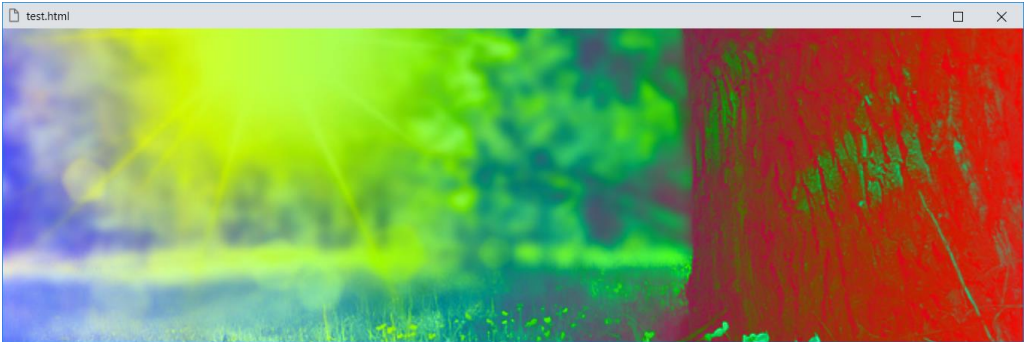


لاحظ المزج بين التدرج، والصورة، حيث الناتج الأخير يشبه الضوء الصلب، ولكنه أكثر نعومة، ويعمل وضع المزج هذا بشكل مشابه للضوء المتسلط، ويتشابه هذا التأثير وكأنه تم تسليط ضوء كشاف على الخلفية.

ومع الوضع difference، والفروقات يختلف الأمر كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{
    background-size:cover;
    background-image:linear-gradient(to left,red,blue),
      url('tree.jpg');
    background-blend-mode:difference;
  }
</style>
</head>
<body> </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ المزج بين التدرج، والصورة، حيث الناتج الأخير هو نتيجة لطح اللوني الداكن مع اللون الأسطح، فلا يكون للطبقة السوداء أي تأثير، بينما تقوم الطبقة البيضاء بتغيير لون الطبقة الأخرى.

ومع الوءء exclusion، والاسءبعاء فءءك الأمر كما فلفف:
ءوء! قم بءءابة الءوء الءالف ثم اءفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{
    background-size:cover;
    background-image:linear-gradient(to left,red,blue),
                        url('tree.jpg');
    background-blend-mode:exclusion;
  }
</style>
</head>
<body> </body>
</html>
```

وبعء ءشءفء الصءفة ءظهر كما بالشءك الءالف:

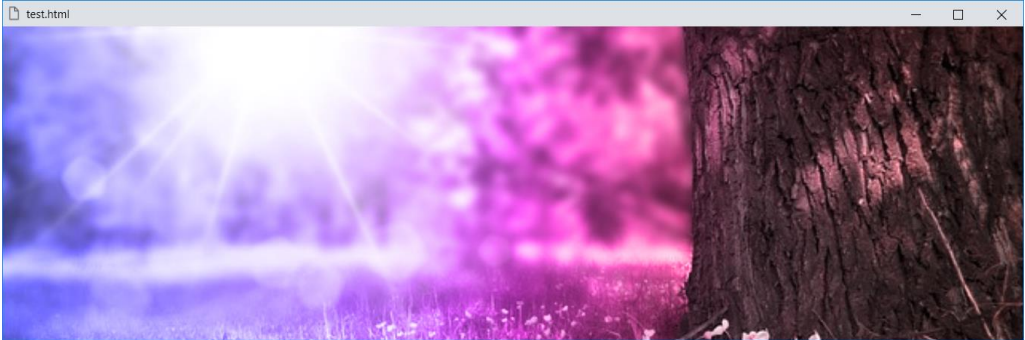


لاءظ المءء ففء الءءءء، والصورة، ءفء الءاءء الأءفء مشابه للءرق، ولكء بأقل ءبافء، كما هو الءال مع الاءءلاف، فلا ففء للءبقة السوداء أف ءأءء، ففءما ءقوم الءبقة الففءاء بءفءفء لون الءبقة الأءرى.

ومع الوضع hue، ودرجة اللون يختلف الأمر كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{
    background-size:cover;
    background-image:linear-gradient(to left,red,blue),
      url('tree.jpg');
    background-blend-mode:hue;
  }
</style>
</head>
<body> </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

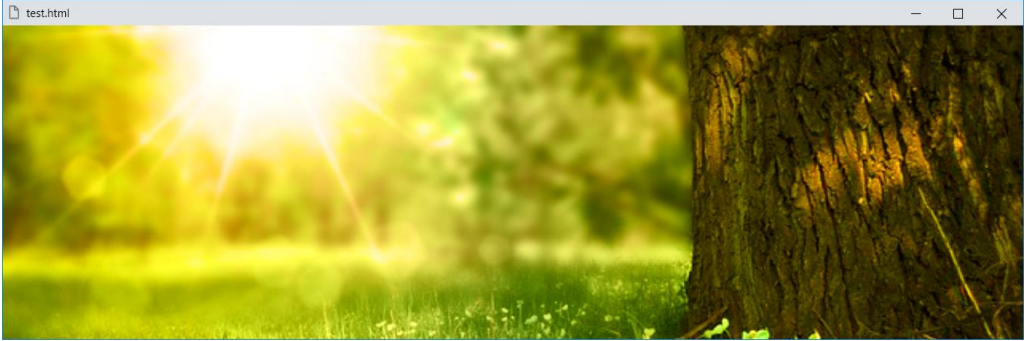


لاحظ المزج بين التدرج، والصورة، حيث الناتج الأخير له لون أعلى، بينما يستخدم تشبع وإضاءة اللون السفلي.

ومع الوضع saturation، والتشبع يختلف الأمر كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
body{
background-size:cover;
background-image:linear-gradient(to left,red,blue),
url('tree.jpg');
background-blend-mode:saturation;
}
</style>
</head>
<body> </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

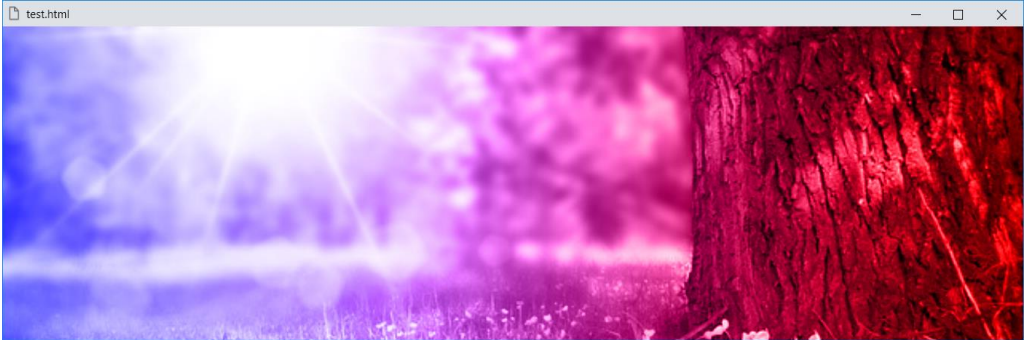


لاحظ المزج بين التدرج، والصورة، حيث الناتج الأخير له تشبع مه أعلى الألوان، في حين يستخدم ال hue وسطوع اللون السفلي.

ومع الوضوح color، واللون يختلف الأمر كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{
    background-size:cover;
    background-image:linear-gradient(to left,red,blue),
      url('tree.jpg');
    background-blend-mode:color;
  }
</style>
</head>
<body> </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

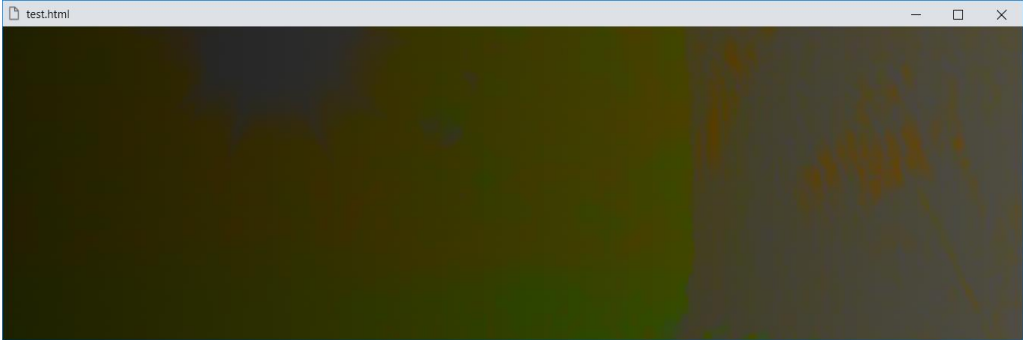


لاحظ المزج بين التدرج، والصورة، حيث الناتج الأخير له لون وتشبع من أعلى لون، بينما يستخدم لمعان اللون السفلي، فيحافظ التأثير على مستويات اللون الرمادي ويمكن استخدامه لتلوين المقدمة.

ومع الوضع color، واللحمان يختلف الأمر كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{
    background-size:cover;
    background-image:linear-gradient(to left,red,blue),
      url('tree.jpg');
    background-blend-mode:luminosity;
  }
</style>
</head>
<body> </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ المزج بين التدرج، والصورة، حيث يحتوي الناتج الأخير على لمعان اللون العلوي، في حين يستخدم لون وتشبع اللون السفلي.

خاصية الخلفية background

خاصية الخلفية background، تجمع القيم المتعلقة بها، حيث يتم ضبط، اللون، والصورة، والموضع مع الحجم، والتكرار، والارتباط، وقصاصة لون الخلفية، وقصاصة صورة الخلفية، حيث تفصل بين كل قيمة وأخرى فقط بمسافة.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{
    height:300px;
    padding:20px;
    border:3px dashed blue;
    background:yellow url('tree.jpg')
                center right/50% no-repeat
                fixed content-box content-box;
  }
</style>
</head>
<body>
  <div></div>
</body>
</html>
```

وبءء ءشءءة الصءفة ءظهء كما بالشكل ءءءء:



ولءلك **انءبه!** فلو ءم ءءابة الكوء ءءءء، لضبط لون الخلففة سوف ءعمل:

```
background:red;
```

ولو ءم ءءابة الكوء ءءءء، لضبط صورة الخلففة سوف ءعمل:

```
background:url('tree.jpg');
```

ولو ءم ءءابة الكوء ءءءء، لضبط لون وصورة الخلففة سوف ءعمل:

```
background:red url('tree.jpg');
```

وهءلءاً، فرءاء الفهء ءءءاً عنء العمل.

تكوين إطار بالصورة border-image

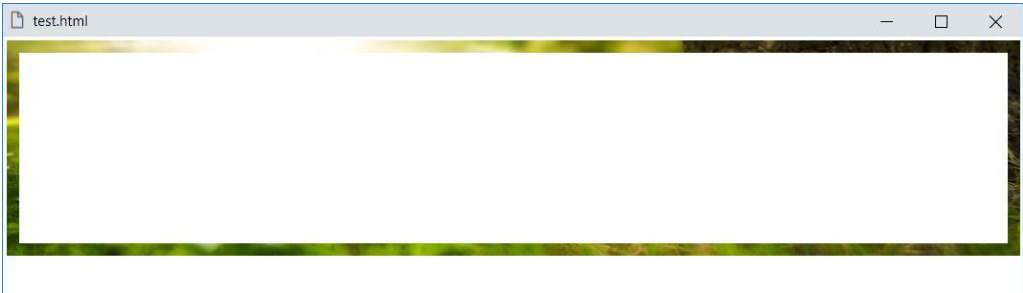
عند استخدام صورة لتكوين إطار منها، في البداية تخيل وكأن هناك صورة، ثم يتم قطع حوافها لاستخدامها كإطار، والخاصية العامة لتكوين الإطار من الصورة هي border-image، ولكي يتفرع منها بعض الخصائص، ولفهم تلك الخصائص، تابع الجدول التالي:

الخاصية	التفصيل
border	لا بد من تحديد إطار عادي أولاً، قبل الاستخدام، وعلى الأقل سمك الإطار ونوع الإطار
border-image-source	مصدر الصورة، وتأخذ url لرابط الصورة
border-image-width	حجم الإطار، ويمكن تجاهلها باستخدام حجم الإطار العادي border
border-image-slice	رقم نسبة يتم قصها من الصورة لاستخدامها في الإطار
border-image-repeat	تكرار صورة الإطار، والقيمة الافتراضية لها هي stretch وهي لتمديد الصورة المستخدمة لحدود الإطار، والقيمة repeat وround لتكرار صورة الإطار، والقيمة space لوضع مسافات بين صورة الإطار
border-image-outset	حجم مقدار خروج صورة الإطار عن موضعها
border-image	هي لتجميع قيم border-image دفعة واحدة

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{
    height:150px;
    border:10px solid;
    border-image-source: url('tree.jpg');
    border-image-width: 10px;
    border-image-slice: 10;
    border-image-repeat: round;
  }
</style>
</head>
<body>
  <div></div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم قطع حواف الصورة، لاستخدامها كإطار.

والآن سوف نقوم بتحميل صورة للإطار موجودة على موقع
 mdn.mozillademos.org مع الرابط التالي:

<https://mdn.mozillademos.org/files/4127/border.png>

وهذه الصورة اسمها border.png وهي التالية:



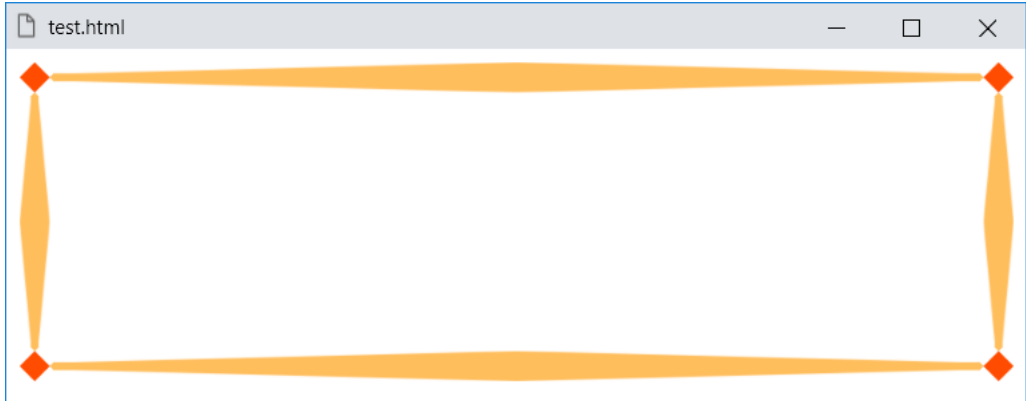
والآن سوف نقوم بعمل مثال على هذه الصورة بفضل الله، كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      div{
        height:150px;
        border:20px solid;
        border-image:url('border.png') 100;
      }
    </style>
  </head>
  <body>
    <div></div>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



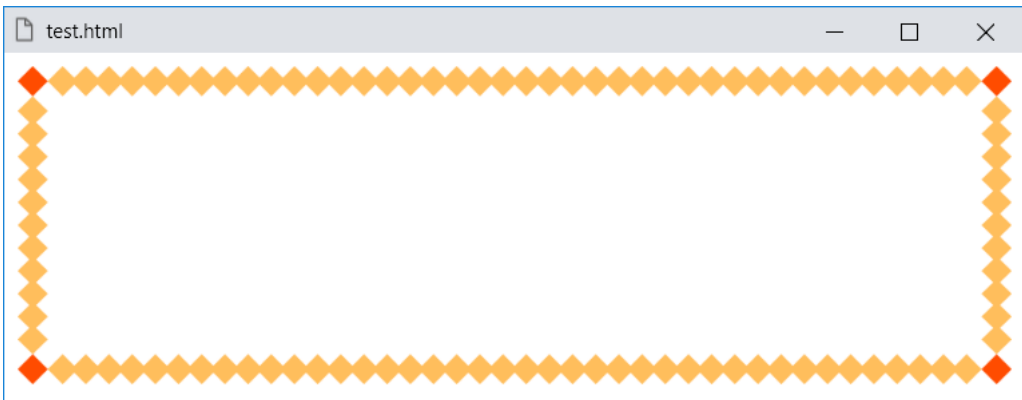
لاحظ أنه تم استخدام الخاصية العامة `border-image` بسهولة،
وتم كتابة المسار بال `url` ثم 100 لأخذ الصورة بالكامل، فأصبحت
كاملة على الحواف الأربعة.
ولو استبدلنا القيمة 100 بالقيمة 50 سوف يكون الناتج كما يلي:



ولو تم تكرارها، سيختلف الأمر، كما سنفعل في المثال التالي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      div{
        height:150px;
        border:20px solid;
        border-image:url('border.png') 30 round;
      }
    </style>
  </head>
  <body>
    <div></div>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



استخدام الألوان Colors

في اللغة CSS يوجد أساليب أخرى لاستخدام الألوان.

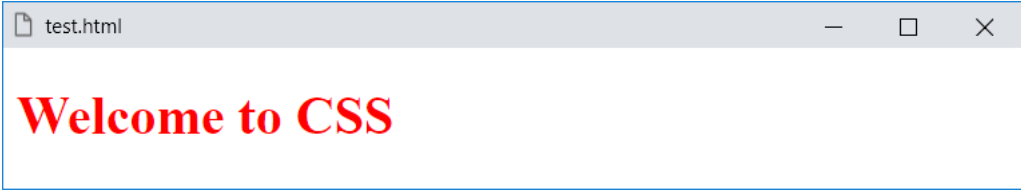
عند استخدام الألوان في لغة CSS يملك كتابة اسم اللون صريح مثل الألوان red و green و blue و orange الى آخر هذه الألوان، ويمك أيضاً استخدام طريقة RGB مثل #FF0000 او #F00 أو #0F0 وهكذا، وهذيك الطريقتيه تم شرحهم باستفاضة من قبل في كتاب الويب "صخرة الأساس" عن اللغة HTML.

دالة rgb لإنتاج اللون

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      h1{
        color:rgb(255,0,0);
      }
    </style>
  </head>
  <body>
    <h1>Welcome to CSS</h1>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم تحديد اللون الأحمر بدون كتابة الاسم red أو #F00 أو #FF0000.

لاحظ أنه لتحديد اللون، تم كتابة **rgb** ثم قوسيه () ثم بداخل القوسيه كتبنا 255 ثم فاصلة " , " ثم صفر ثم فاصلة " , " ثم صفر، ومعني هذا أنه تم إعطاء الأحمر 255 درجة، ثم الأخضر صفر، ثم الأزرق صفر، وبهذه الطريقة نستطيع تكوين أي لون نريده من خليط الألوان، بمعرفة الدرجات المطلوبة لكل لون.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head></head>
  <body>
    <h2 style="background-color:rgb(5,200,170);">
      Welcome to CSS
    </h2>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ انتاج لونه آخر مختلف بطريقة درجات .rgb.

دالة rgba للشفافية مع اللونه
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body{ background:url('tree.jpg'); }
      h1{
        background-color:rgba(0,255,0,0.3);
      }
    </style>
  </head>
  <body>
    <h1>
      Welcome to CSS<br>
      Hello
    </h1>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

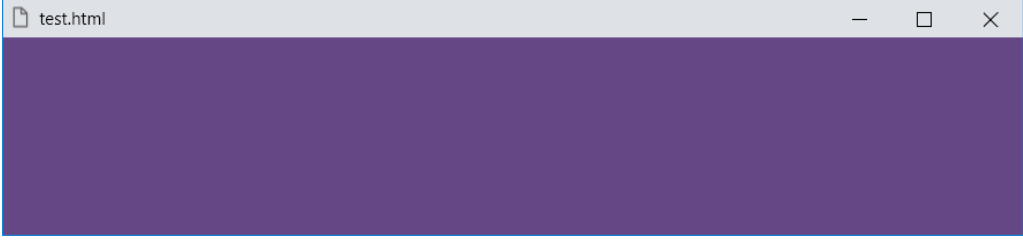


لاحظ أنه تم تلوين اللون الأخضر باستخدام الدالة rgba ثم تم استخدام القيمة 0.3 وذلك ليكون اللون ظاهر بنسبة 30% أي يكون نسبة الشفافية 70%، وبفضل الله تم استخدام صورة في خلفية الـ body لوضوح الشفافية ولتسهيل المفهوم لك سريعاً.

دالة hsl لإنتاج اللون (درجة اللون، نسبة الإشباع، نسبة السطوع)
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body{
        background:hsl(270,30%,40%);
      }
    </style>
  </head>
  <body></body>
</html>
```

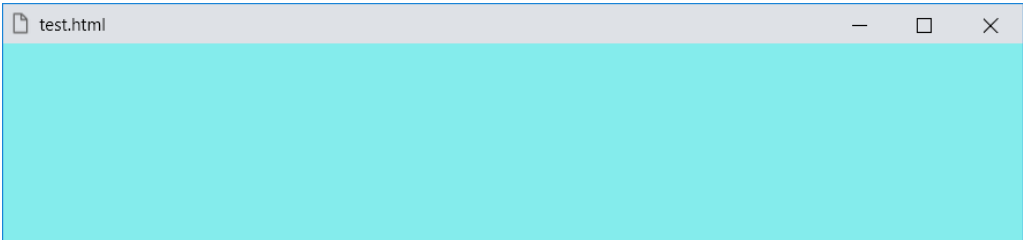
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



دالة `hsla` لإنتاج اللون مع الشفافية
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body{
        background:hsla(180,90%,45%,0.5);
      }
    </style>
  </head>
  <body></body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



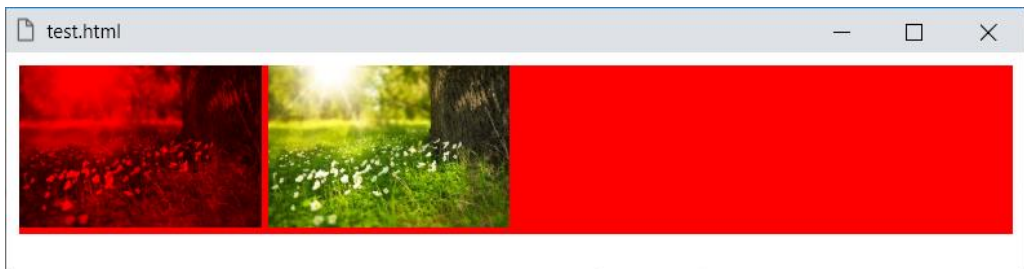
خليط المزيج mix-blend-mode

سوف نجرب مزج darken مثال بالكود! التالي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
div{
background-color: red;
}
#i1{ mix-blend-mode: darken; }
</style>
</head>
<body>
<div>


</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

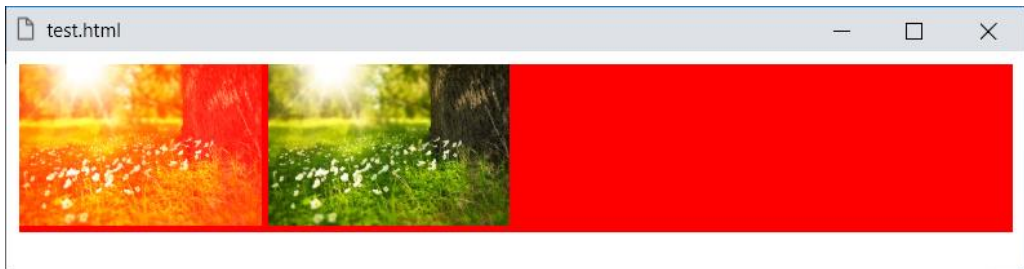


لاحظ اختلاف الصورة التي تم ضبط خليط المزج لها، مع الصورة التي لم يتم تنسيقها، مع العلم ان القيمة الافتراضية لهذه الخاصية هي القيمة .normal

سوف نجرب مزج lighten مثال بالكود! التالي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{
    background-color: red;
  }
  #i1{ mix-blend-mode: lighten; }
</style>
</head>
<body>
<div>
  
  
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

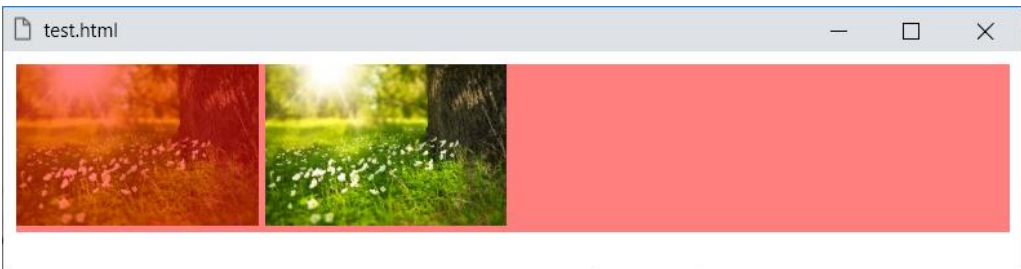


لاحظ اختلاف الصورة التي تم ضبط خليط المزج لها، عه الصورة التي لم يتم تنسيقها.

سوف نجرب مزج overlay مثال **بالكود!** التالي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{
    background-color: rgba(255,0,0,0.5);
  }
  #i1{ mix-blend-mode: overlay; }
</style>
</head>
<body>
<div>
  
  
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

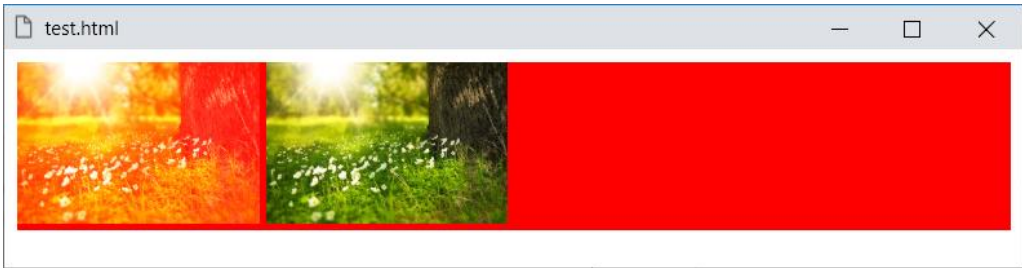


لاحظ اختلاف الصورة التي تم ضبط خليط المزج لها، مع الصورة التي لم يتم تنسيقها، ورجاء **انتبه!** لوجود الشفافية.

سوف نجرب مزج screen مثال **بالكود!** التالي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{
    background-color: red;
  }
  #i1{ mix-blend-mode: screen; }
</style>
</head>
<body>
<div>
  
  
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ اختلاف الصورة التي تم ضبط خليط المزج لها، عه الصورة التي لم يتم تنسيقها.

سوف نجرب مزج multiply مثال بالكود! التالي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{
    background-color: red;
  }
  #i1{ mix-blend-mode: multiply; }
</style>
</head>
<body>
<div>
  
  
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

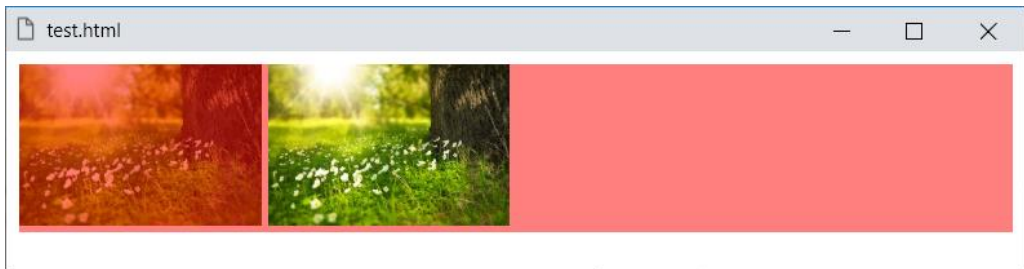


لاحظ اختلاف الصورة التي تم ضبط خليط المزج لها، مع الصورة التي لم يتم تنسيقها.

سوف نجرب مزج color-dodge مثال **بالكود!** التالي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{
    background-color: rgba(255,0,0,0.5);
  }
  #i1{ mix-blend-mode: color-dodge; }
</style>
</head>
<body>
<div>
  
  
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

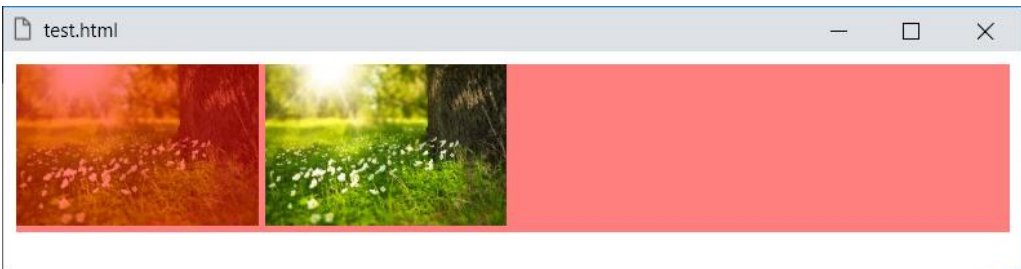


لاحظ اختلاف الصورة التي تم ضبط خليط المزج لها، عه الصورة التي لم يتم تنسيقها، ورجاء **انتبه!** لوجود الشفافية.

سوف نجرب مزج color-burn مثال **بالكود!** التالي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{
    background-color: rgba(255,0,0,0.5);
  }
  #i1{ mix-blend-mode: color-burn; }
</style>
</head>
<body>
<div>
  
  
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

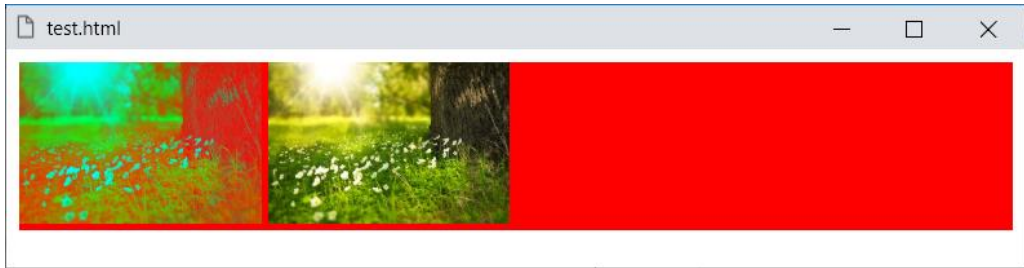


لاحظ اختلاف الصورة التي تم ضبط خليط المزج لها، مع الصورة التي لم يتم تنسيقها، ورجاء **انتبه!** لوجود الشفافية.

سوف نجرب مزج difference مثال بالكود! التالي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{
    background-color: red;
  }
  #i1{ mix-blend-mode: difference; }
</style>
</head>
<body>
<div>
  
  
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ اختلاف الصورة التي تم ضبط خليط المزج لها، مع الصورة التي لم يتم تنسيقها.

سوف نجرب مزج exclusion مثال بالكود! التالي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{
    background-color: red;
  }
  #i1{ mix-blend-mode: exclusion; }
</style>
</head>
<body>
<div>
  
  
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ اختلاف الصورة التي تم ضبط خليط المزج لها، مع الصورة التي لم يتم تنسيقها.

سوف نجرب مزج hue مثال **بالكود!** التالي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{
    background-color: red;
  }
  #i1{ mix-blend-mode: hue; }
</style>
</head>
<body>
<div>
  
  
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ اختلاف الصورة التي تم ضبط خليط المزج لها، مع الصورة التي لم يتم تنسيقها.

سوف نجرب مزج saturation مثال بالكود! التالي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{
    background-color: red;
  }
  #i1{ mix-blend-mode: saturation; }
</style>
</head>
<body>
<div>
  
  
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

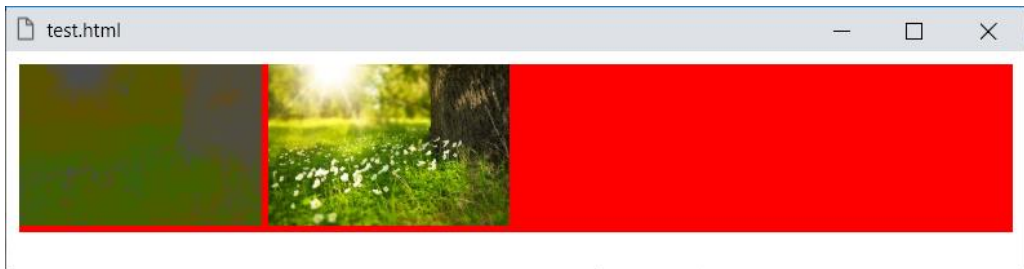


لاحظ اختلاف الصورة التي تم ضبط خليط المزج لها، مع الصورة التي لم يتم تنسيقها.

سوف نجرب مزج color مثال بالكود! التالي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{
    background-color: red;
  }
  #i1{ mix-blend-mode: color; }
</style>
</head>
<body>
<div>
  
  
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



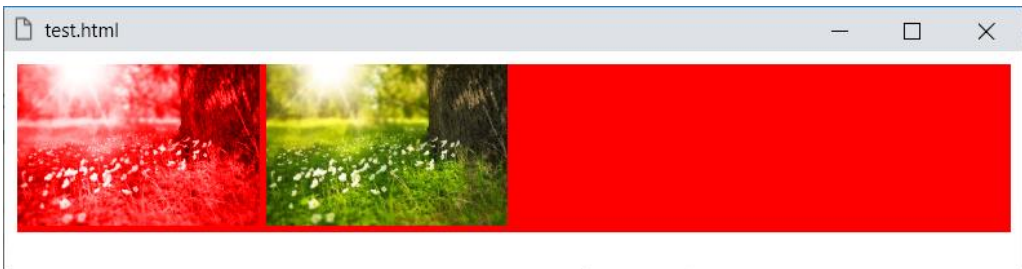
لاحظ اختلاف الصورة التي تم ضبط خليط المزج لها، مع الصورة التي لم يتم تنسيقها.

سوف نجرب مزج luminosity مثال بالكود! التالي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
div{
background-color: red;
}
#i1{ mix-blend-mode: luminosity; }
</style>
</head>
<body>
<div>


</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ اختلاف الصورة التي تم ضبط خليط المزج لها، مع الصورة التي لم يتم تنسيقها.

العزل isolation

سوف نجرب مثال بالكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .a { background-color: rgb(0,255,0); }
    #b { width: 200px; height: 210px; }
    .c {
      width: 100px; height: 100px;
      border: 1px solid black; padding: 2px;
      mix-blend-mode: difference;
    }
    #d { isolation: auto; }
    #e { isolation: isolate; }
  </style>
</head>
<body>
  <div id="b" class="a">
    <div id="d">
      <div class="a c">auto</div>
    </div>
    <div id="e">
      <div class="a c">isolate</div>
    </div>
  </div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



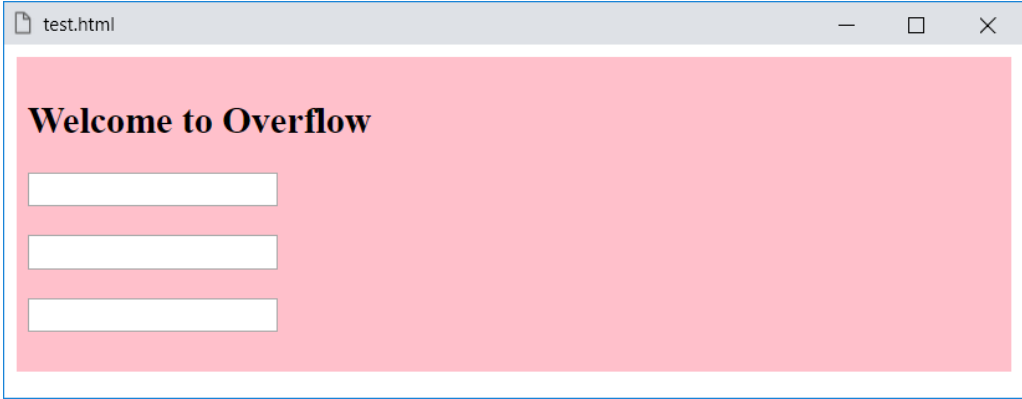
المحتوي الزائد Overflow

عند ضبط عرض وطول لوسم معين مثل p أو div أو أي وسم به محتوى، ثم تم وضع نصوص أو عناصر حتى زادت عن الطول أو العرض المحدد لهذا الوسم، فحينئذ يجب أن تتعرف على موضوع overflow لتستطيع التعامل بفهم مع هذه الحالات.

رجاء **انتبه!**، وركز جيداً مع المثال التالي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    div{ background: pink; padding: 7px; }
  </style>
</head>
<body>
  <div>
    <h2>Welcome to Overflow</h2>
    <input><br><br>
    <input><br><br>
    <input><br><br>
  </div>
</body>
</html>
```

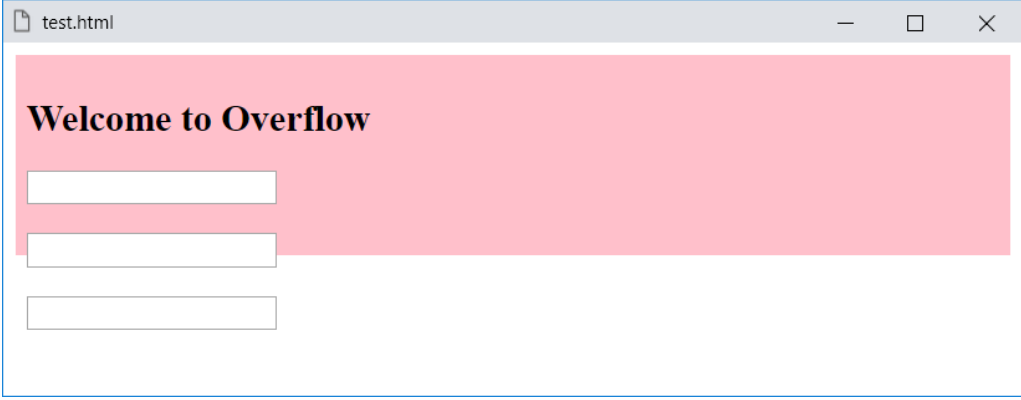
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن طول الـ div زاد بشكل تلقائي حسب ما بداخلها.
والآن، رجاء **انتبه!**، وركز جيداً مع المثال التالي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{ height:115px;background:pink;padding:7px; }
</style>
</head>
<body>
<div>
  <h2>Welcome to Overflow</h2>
  <input><br><br>
  <input><br><br>
  <input><br><br>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



انتبه! ماذا حدث؟!، لقد حدث شيء غيّر مقبول، لقد خرجت العناصر من الوسم div وكأنها ليست فيه، كما يوجد لنا لون خلفية الـ div، ومن هنا أحبائي تأتي أهمية موضوع الـ Overflow.

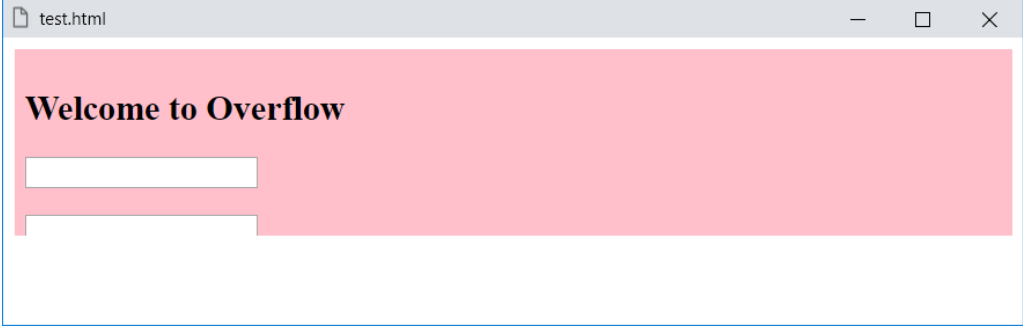
خاصية المحتوى الزائد overflow

هذه الخاصية قيمتها الافتراضية، هي **visible**، أي مرئي كما هو واضح في المثال السابق، أي لو زاد المحتوى قم بإظهاره، والقيمة **hidden** تخفي المحتوى الزائد، والقيمة **scroll** تظهر شريط تمرير للتحكم في المحتوى الزائد أثناء عرض صفحة الويب، والقيمة **auto** تجعل الموضوع أوتوماتيكياً، أي لو هناك محتوى يظهر شريط التمرير، والا فلا يظهر، وكل هذا حسب اختيارك وتصميمك لصفحة الويب الخاصة بك.

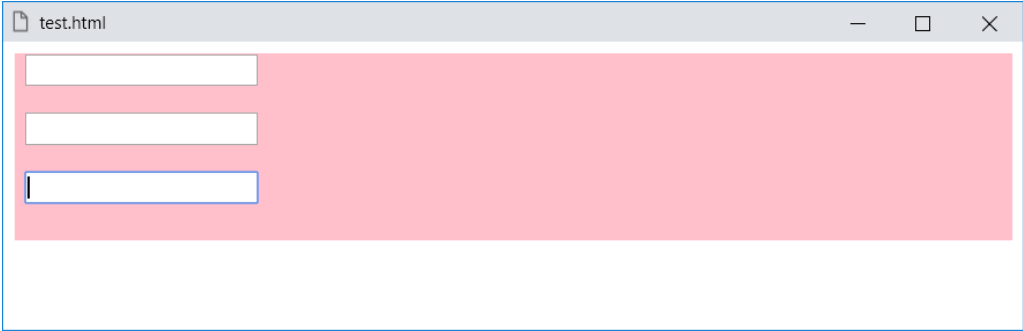
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    div{ height:115px;background:pink;padding:7px; }
  </style>
</head>
<body>
  <div style="overflow:hidden">
    <h2>Welcome to Overflow</h2>
    <input><br><br>
    <input><br><br>
    <input><br><br>
  </div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم اختفاء المحتوى الزائد عن الطول، ولكنه **انتبه!** فإذا ضغطت بالماوس لتجعل المؤشر داخل مربع النص، تم تغطية على زر tab من الكيبورد أكد من مرة، سيكون الشكل كما يلي:



ومع هنا تعرف أن المحتوى مخفي، لكنه مفعّل بشكل طبيعي.

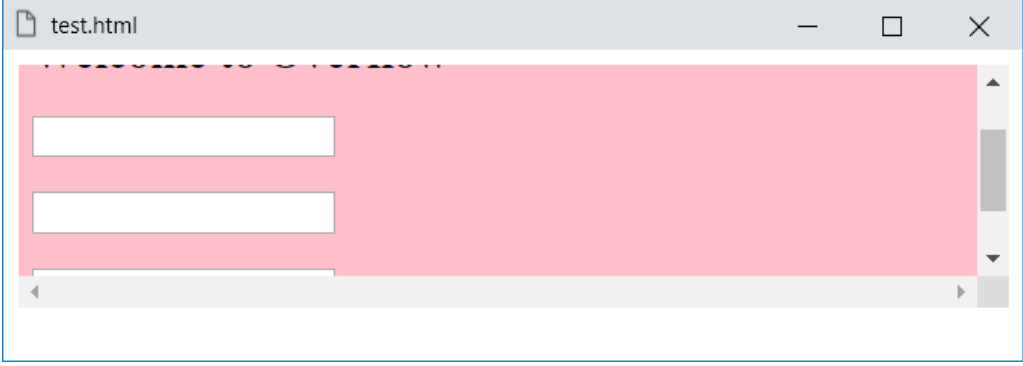
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    div{ height:115px;background:pink;padding:7px; }
  </style>
</head>
<body>
  <div style="overflow:scroll">
    <h2>Welcome to Overflow</h2>
    <input><br><br>
    <input><br><br>
    <input><br><br>
  </div>
</body>
</html>
```

لاحظ شريط التمرير بعد تشغيل الصفحة كما بالشكل التالي:



حيث لو نزلت بشرط التمير، تكون مثل الشكل التالي:



وهكذا، حتى تصل لما تريد.

ولو استخدمت القيمة auto تظهر شرط التمير فقط لو هناك محتوى زائد، ولو استخدمت القيمة visible له تشعر باختلاف، لأنها هي الافتراضية، كما تم ذكر ذلك من قبل.

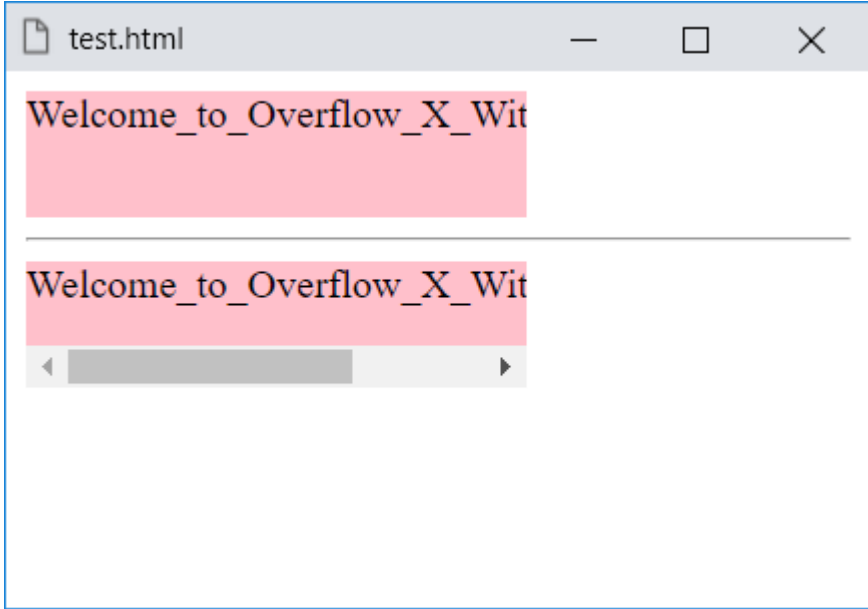
المحتوي الأفقي الزائد overflow-x

عند التعامل مع خاصية overflow نلاحظ أنها ركزت على شريط التمرير الرأسى وشريط التمرير الأفقي معاً، ولكن الآن، ومع الخاصية overflow-x نستطيع التحكم في شريط التمرير الأفقي فقط، وهي تأخذ نفس القيم، سواء auto أو hidden أو scroll أو visible الافتراضية.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    div{ width:200px;height:50px;background:pink; }
  </style>
</head>
<body>
  <div style="overflow-x:hidden">
    Welcome_to_Overflow_X_With_Long_Word
  </div>
  <hr>
  <div style="overflow-x:scroll">
    Welcome_to_Overflow_X_With_Long_Word
  </div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ اختفاء المحتوى الزائد أفقياً، في الـ div الأولي، وكأنه غير موجود على الصفحة.

لاحظ ظهور شريط التمرير الأفقي فقط، وعدم ظهور شريط التمرير الرأسي في الـ div الثانية.

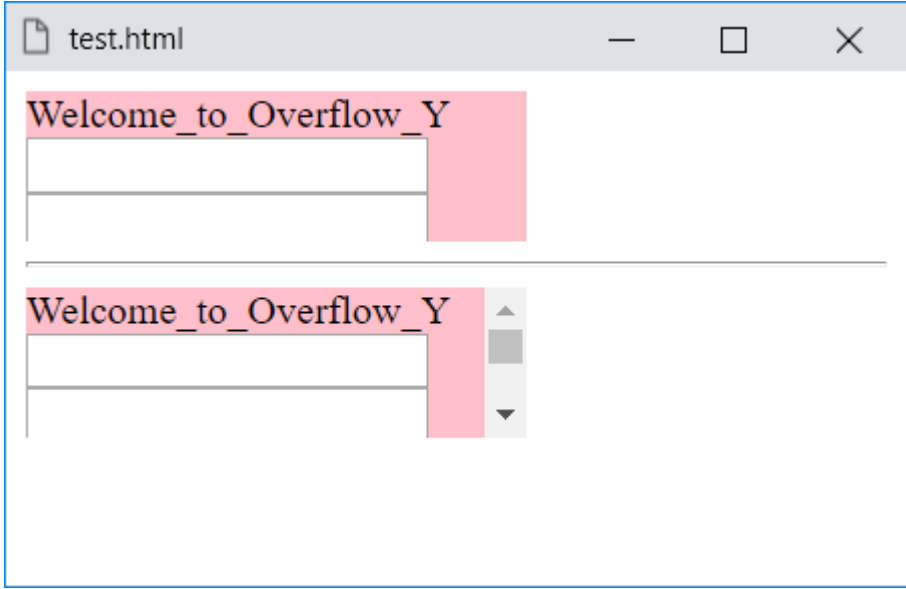
المحتوي الرأسى الزائد overflow-y

عند التعامل مع خاصية overflow نركز على شريطية تمرير رأسى وأفقى، وخاصية overflow-x نركز على شريط تمرير أفقى فقط، ولكن الآن، ومع الخاصية overflow-y نستطيع التحكم في شريط التمرير الرأسى فقط، والخاصية overflow-y تأخذ نفس القيم، سواء hidden أو scroll أو auto أو visible الافتراضية.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    div{ width:200px;height:60px;background:pink; }
  </style>
</head>
<body>
  <div style="overflow-y:hidden">
    Welcome_to_Overflow_Y<br>
    <input><br><input><br><input>
  </div> <hr>
  <div style="overflow-y:scroll">
    Welcome_to_Overflow_Y<br>
    <input><br><input><br><input>
  </div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ اختلاف المحتوي الزائد رأسياً، في الـ div الأولي، وكأنه غير موجود على الصفحة.

لاحظ ظهور شريط التمرير الرأسي فقط، وعدم ظهور شريط التمرير الأفقي في الـ div الثانية.

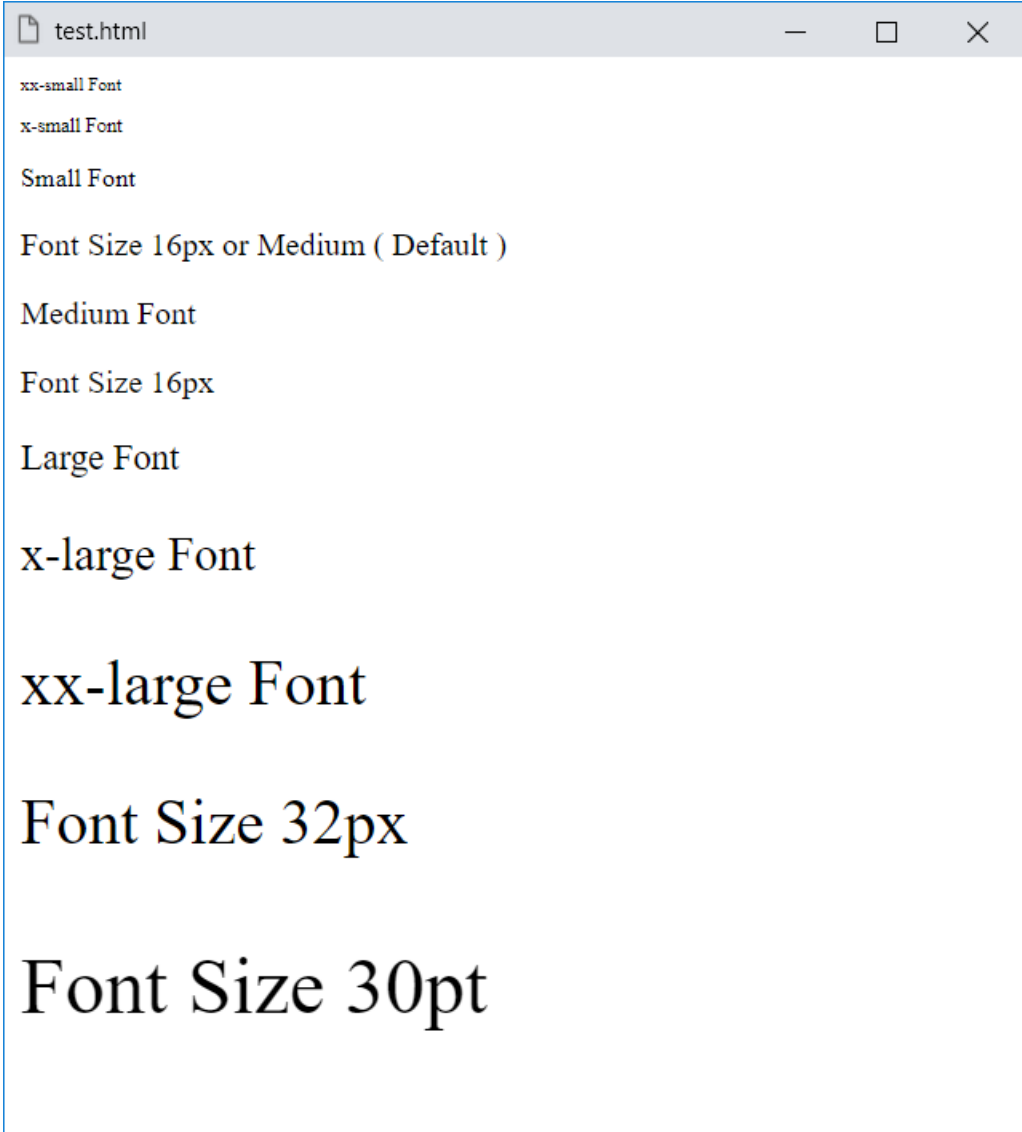
Texts النصوص

font-size حجم الخط

خاصية font-size للتحكم في حجم الخط كما تريد، وللتجارب على أحجام الخط تابع ما يلي بتدقيق تام:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head></head>
  <body>
    <p style="font-size:xx-small">xx-small Font</p>
    <p style="font-size:x-small">x-small Font</p>
    <p style="font-size:small">Small Font</p>
    <p>Font Size 16px or Medium ( Default )</p>
    <p style="font-size:medium">Medium Font</p>
    <p style="font-size:16px">Font Size 16px</p>
    <p style="font-size:large">Large Font</p>
    <p style="font-size:x-large">x-large Font</p>
    <p style="font-size:xx-large">xx-large Font</p>
    <p style="font-size:32px">Font Size 32px</p>
    <p style="font-size:30pt">Font Size 30pt</p>
  </body>
</html>
```

وءءء ءءءفءل الصفاءة ءظهء ءءا بالشءك ءءالف:



لاءظ أه الءءم الاءءراءفءف هو 16px أو Medium، ولاءظ أنه فمءك ءءابة اسم ءءءم الءء، مءء large و small وءفءها، و فمءك ءءابة بءفءة px أو pt أو ءءف النسب % وءفءها.

نوع الخط font-family

خاصية font-family للتحكم في نوع الخط، مثل نوع الخط Arial أو مثل نوع الخط Tahoma أو مثل نوع الخط Verdana أو مثل نوع الخط Andalus أو مثل نوع الخط Consolas أو مثل نوع الخط Times New Roman أو مثل نوع الخط Impact وغيرها الكثير والكثير من أنواع الخطوط المختلفة.

انتبه! فإن نوع الخط متوقف ومعتمد على الخطوط التي تم تنزيلها على جهاز كمبيوتر الشخص الذي يقوم بتشغيل الصفحة، وأيضاً معتمد على المتصفح الذي تم تشغيل الصفحة عليه لدعم نوع الخط، فمثلاً لو طلبت نوع الخط impact لصفحتك، ثم تم فتح الصفحة على جهاز لا يوجد فيه النوع impact له يتم العرض لصفحة الويب كما اختبرته.

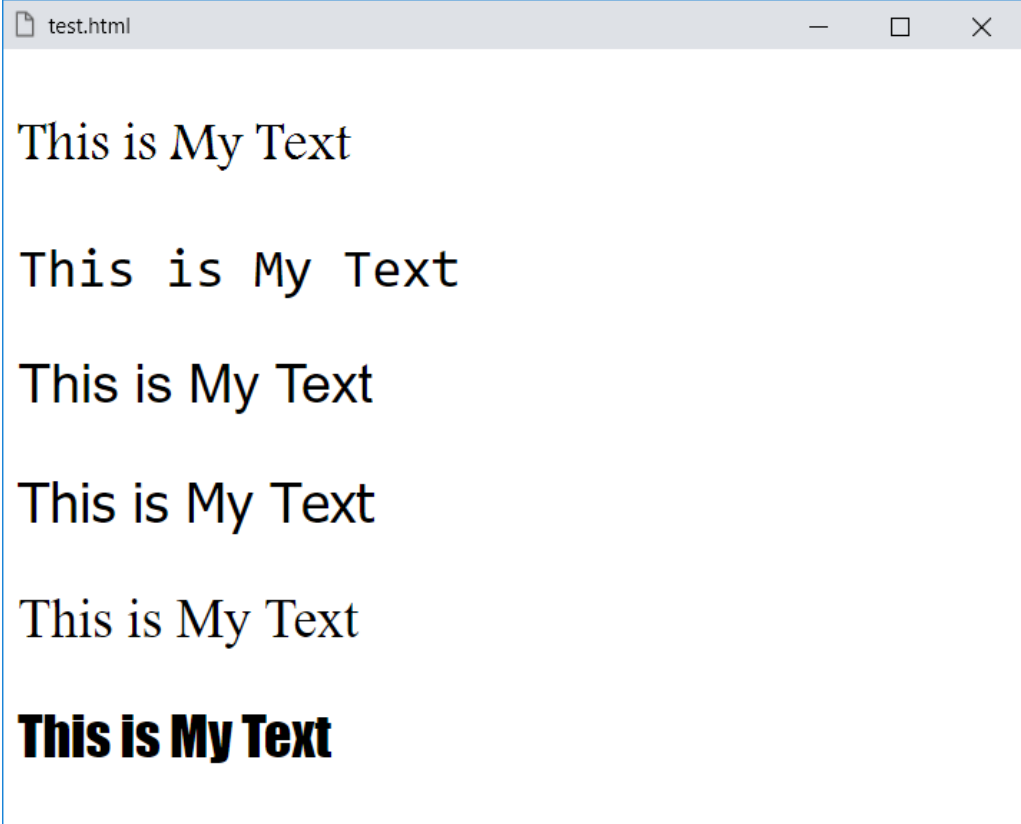
انتبه! فإنه عند تخصيص نوع الخط لفقرة أو أي عنصر به كلام، يتم استخدام الخاصية font-family، ويتم اسناد اسم الخط لها، مثلاً font-family:arial أو font-family:andalus، وأيضاً يمكنك وضع " مثل font-family:"tahoma"، وأيضاً يمكنك استخدام ' مثل font-family:'consolas'، وهكذا، ولكنه **انتبه!** فلو اسم الخط به مسافات مثل Times New Roman فيجب استخدام " أو ' لكتابه اسم الخط.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
p{ font-size:30px; }
#p1{ font-family:andalus; }
#p2{ font-family:consolas; }
#p3{ font-family:"arial"; }
#p4{ font-family:'tahoma'; }
#p5{ font-family:"times new roman"; }
#p6{ font-family:impact, verdana, arial; }
</style>
</head>
<body>
<p id="p1">This is My Text</p>
<p id="p2">This is My Text</p>
<p id="p3">This is My Text</p>
<p id="p4">This is My Text</p>
<p id="p5">This is My Text</p>
<p id="p6">This is My Text</p>
</body>
</html>
```

انتبه! فقبل عرض الكود لاحظ وجود الكود الذي يعمل على #p6 والذي خصصنا له أكد من قيمة نوع خط، وبعبارة أخرى كل نوع وآخر كوما، فقد تتساءل، ما هذا!؟، والإجابة بفضل الله، هو البديل، أي تم استخدام النوع impact ولو لم يجده سيستخدم verdana ولو لم يجده سيستخدم arial وهكذا، لباقي البدائل.

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم تثبيت حجم الخط لجميع الفقرات ليكون 30px ومع ذلك تلاحظ اختلاف نقشة كل خط عن الآخر، ومع أجل ذلك رجاء **انتبه!** فإنها معلومة هامة جداً، لأنها تؤثر على تصميم الموقع بالكلية، لأن كل نوع خط له نقشة مختلف، فهناك نقشة كبير، ونقشة صغير، ونقشة ثقيل، وهكذا، فيختلف العرض حسب النقشة.

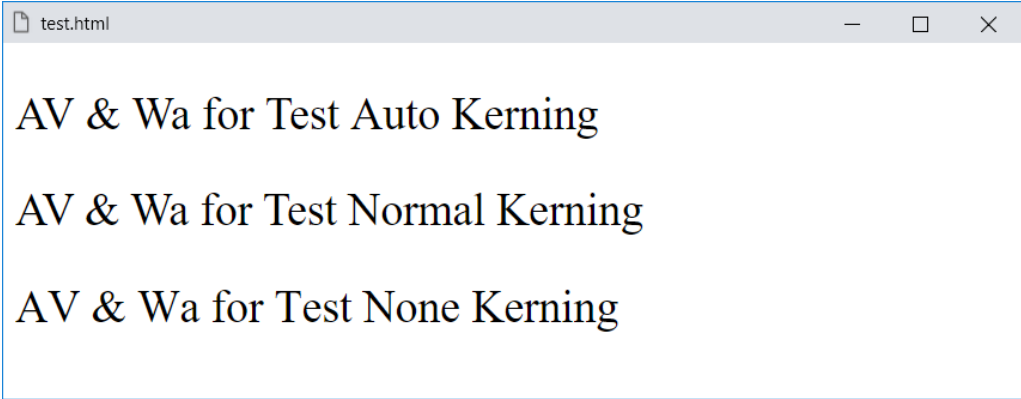
تقنية الخط font-kerning

خاصية font-kerning للتحكم في المسافات بين الحروف لكل خط،
ولكنه رجاء انتبه! فلا بد أن يدعم نوع الخط المستخدم بيانات kerning
حتى تدمي التأثير، وإذا كانت غير مدعومة، فله يكون هناك أي تأثير.

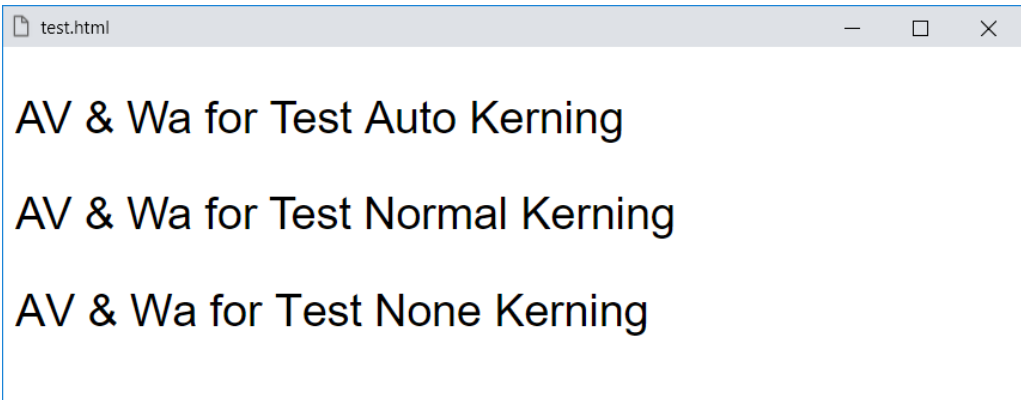
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      p{ font-size:30px; font-family:serif; }
    </style>
  </head>
  <body>
    <p style="font-kerning:auto;">
      AV & Wa for Test Auto Kerning
    </p>
    <p style="font-kerning:normal;">
      AV & Wa for Test Normal Kerning
    </p>
    <p style="font-kerning:none;">
      AV & Wa for Test None Kerning
    </p>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



انتبه! فإن القيمة auto هي القيمة الافتراضية، ولاحظ أن الاختلاف ضئيل جداً، وذلك لأن المسافات بين الحروف تعتبر شيء قليل، ولذلك تم استخدام الحروف AV والحروف Wa ليظهر التأنيب في بداية كل فقرة، ولو تم استخدام النوع arial بدلاً من النوع serif سيكون العرض كما بالشكل التالي:



ومع هنا نلاحظ اختلاف ال Kerning بين النوع serif والنوع arial كما هو واضح، بفضل الله.

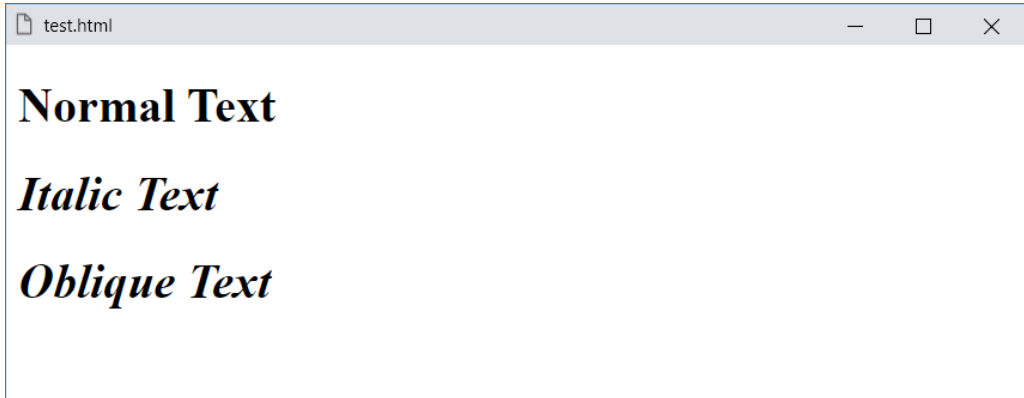
نمط الخط font-style

خاصية font-style للتحكم في نمط الخط، فعمل هو مائل italic، أم عادي normal، أم منحرف Oblique، وسوف نجرّب فيما يلي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head></head>
  <body>
    <h1 style="font-style:normal">Normal Text</h1>
    <h1 style="font-style:italic">Italic Text</h1>
    <h1 style="font-style:oblique">Oblique Text</h1>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



انتبه! فإن القيمة normal هي الافتراضية، ولو اردت جعل الخط مائل يمكنك استخدام italic أو Oblique.

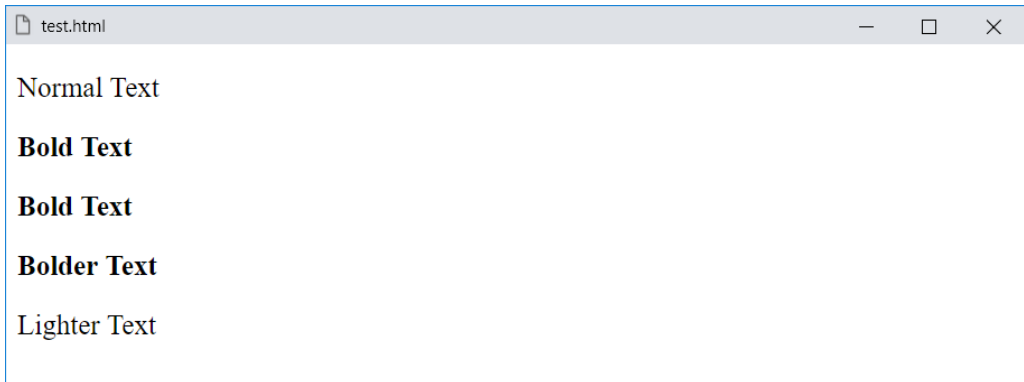
ثقل الخط font-weight

خاصية font-weight للتحكم في ثقل الخط، فعمل هو عادي بالقيمة normal، أو ثقيل bold، أو أثقل منه bolder أو رقم 100 أو 200 أو 300 وحتى 900، وغير ذلك، ولكه الاختلاف سيكون غير واضح إلا مع العادي والثقيل، وحسب نوع الخط، وسوف نجد فيما يلي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>p{font-size:20px;}</style>
  </head>
  <body>
    <p style="font-weight:normal">Normal Text</p>
    <p style="font-weight:700">Bold Text</p>
    <p style="font-weight:bold">Bold Text</p>
    <p style="font-weight:bolder">Bolder Text</p>
    <p style="font-weight:lighter">Lighter Text</p>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

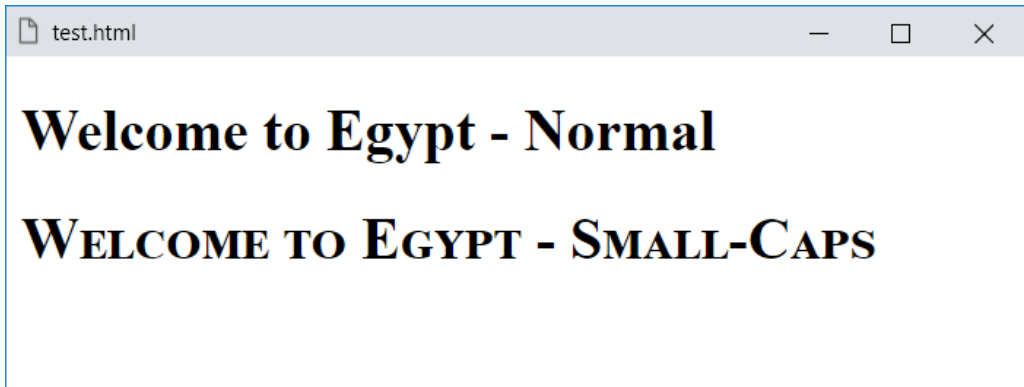


تنوع الخط font-variant

خاصية font-variant للتحكم في تنوع حالة الأحرف للخط، وسوف يتضح لك إن شاء الله، بعد كتابة الكود ورؤية الصفحة، كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head></head>
<body>
  <h1 style="font-variant:normal">
    Welcome to Egypt - Normal
  </h1>
  <h1 style="font-variant:small-caps">
    Welcome to Egypt - Small-Caps
  </h1>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



@font-face صنع الخط بالمدد

سوف نتعرف بفضل الله على المدد @font-face لصناعة الخطوط، وله قوسيه مجموعة { } ليكتب بينهم أنواع صنع الخطوط.

وفي البداية لابد أن نعلم أن الخطوط عبارة عن مجموعة ملفات، وهذه الملفات، مثل أي ملفات طبيعية، حيث لها امتداد، وعند العمل على الويب، نستخدم بعض الامتدادات، وأشهر هذه الامتدادات هو الامتداد TTF، ويوجد أيضاً الامتداد WOFF، والامتداد WOFF 2.0، والامتداد OTF، والامتداد SVG، والامتداد EOT.

أيضاً كل متصفح يقبل أنواع خطوط معينة، فلذلك يرجى مراعاة دعم المتصفحات لأنواع الخطوط.

ولصناعة خط جديد، يتم انشاء مدد @font-face ثم بداخله يتم ضبط الخاصية font-family على اسم جديد تماماً، والذي سوف تستخدمه فيما بعد، إذا اردت استخدام هذا الخط، ويتم أيضاً ضبط الخاصية SRC على مسار ملف الخط الذي تريده.

والآن للتجربة، سوف نقوم بتحميل خط Font صالح لصفحات الويب من الانترنت باسم Arbutus، والملف هذا الخط اسمه بالامتداد هو Arbutus-Regular.ttf، والتجربة بفضل الله كما يلي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  @font-face{
    font-family:MyNewFont;
    src:url('Arbutus-Regular.ttf');
  }
</style>
</head>
<body>
  <h1 style="font-family:MyNewFont;">
    Welcome to My New Font
  </h1>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



ولو أردنا اسناد ملفات أخرى نستخدم كوما كفاصلة، مثل الكود التالي:

```
src:url('Arbutus-Regular.ttf'),
  url('Arbutus-Regular.woff'),
  url('Arbutus-Regular.woff2');
```

استيراد الخط بالمحدد @import

المحدد @import هو في الأساس لاستيراد بعض الروابط أو بعض الملفات، حيث يتم كتابة المحدد @import ثم url وبداخلها الرابط الذي نريد استيراده، والآن سوف نستورد خط font من خطوط جوجل ونستخدمه، حيث يوجد خطوط متعددة في الرابط التالي:

<https://fonts.google.com/>

والآن سوف اختار خط اسمه Charmonman من الرابط التالي:

<https://fonts.google.com/specimen/Charmonman>

والآن سوف أنضغط على Select This Font، ثم الضغط على Family Selected، ثم اختيار @IMPORT، فيظهر رابط الاستيراد جاهز، وهو كما يلي:

```
<style>
@import url('https://fonts.googleapis.com/css?family=Charmonman');
</style>
```

ويوجد كود الاستخدام، وهو كما يلي:

```
font-family: 'Charmonman', cursive;
```

وبعد الاستخدام تكون التجربة مثل الشكل التالي:



ولكن **انتبه!** فإذا استخدمت المحدد @font-face و @import معاً، أجعل المحدد @import هو الذي بالأعلى ليعمل.

منبظ حجم الخط font-size-adjust

خاصية font-size-adjust للتحكّم في منبظ الخط عند عدم تواجد النوع المحدد، وهذه الخاصية مفيدة لأنه يتم تحديد وضوح الخطوط، خصوصاً في حالة الحروف الصغيرة، أكثر من الأحرف الكبيرة، ويمكن أن تصبح القراءة بها مشكلة عندما يكون نوع الخط الأول غير متاح، ويكون بديلها نسبة أبعاده مختلفة بشكل كبير، أي نسبة حجم الأحرف الصغيرة إلى حجم الخط.

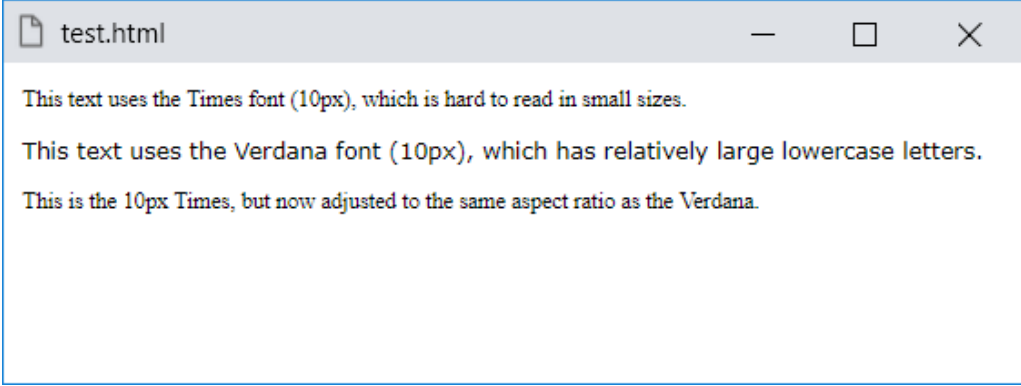
ولاستخدام هذه الخاصية بطريقة متوافقة مع المتصفحات التي لا تدعم منبظ حجم الخط font-size-adjust، يتم تحديدها كرقم يتم ضرب خاصية حجم الخط font-size به، وهذا يعني أن القيمة المحددة للموقع يجب أن تكون نسبة العرض إلى الارتفاع لنوع الخط الأول، والآه دعونا نجرب لتفهم الأمر بفضل الله.

وحتى نجرب بمثال معتمد، احضرت لكم مثال من موقع mozilla.org لاختبار خاصية font-size-adjust، وهذا المثال به فقرات، ومكتوب بداخل كل فقرة نص يعبر عن التجربة.

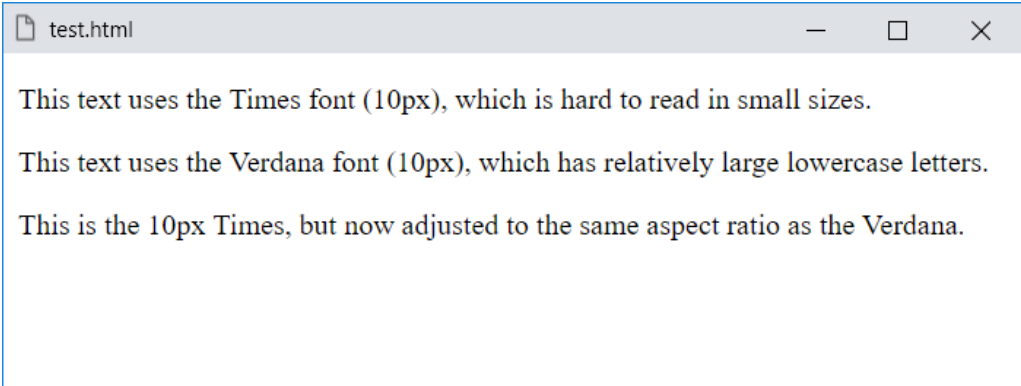
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
.times {
  font-family: Times, serif;
  font-size: 10px;
}
.verdana {
  font-family: Verdana, sans-serif;
  font-size: 10px;
}
.adjtimes {
  font-family: Times, sans-serif;
  font-size-adjust: 0.58;
  font-size: 10px;
}
</style>
</head>
<body>
<p class="times">
This text uses the Times font (10px), which is hard to
read in small sizes.</p>
<p class="verdana">
This text uses the Verdana font (10px), which has
relatively large lowercase letters.</p>
<p class="adjtimes">
This is the 10px Times, but now adjusted to the same
aspect ratio as the Verdana.</p>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



ولو تم إزالة تنسيق CSS بالكامل، سوف يكون ناتج صفحة الويب كما بالشكل التالي:



إذاً رجاء **انتبه!** فإن الخاصية font-size-adjust في CSS لضبط حجم الخط أثناء العرض، حيث يجب أن يكون تم اختياره على أساسه طول الأحرف الصغيرة lowercase بدلاً من طول الأحرف الكبيرة uppercase وهذا يفيد لوضوح الخطوط وسهولة قراءتها حتى ولو كانت كلها حروف صغيرة.

تمدد الخط font-stretch

خاصية font-stretch لتمدد الخط، وهي تعرض لبعض أنواع الخطوط أوجه إضافية، بحيث تكون فيها الأحرف أضيق من الوجه العادي، أي وجوه مكثفة، أو تكون أوسع من الوجه العادي أي وجوه موسعة، ويمكنك استخدام امتداد الخط font-stretch لتحديد وجه مكثف أو موسع من هذه الخطوط، ولكه **انتبه!** فإذا كان نوع الخط الذي تستخدمه لا يعرض الوجوه المكثفة أو الموسعة، فله يكون لهذه الخاصية أي تأثير.

حيث تأخذ الخاصية font-stretch، مجموعة من القيم، فالقيمة normal لعدم التمدد وهي الافتراضية للخاصية، والقيمة expanded للتوسع، والقيمة extra-expanded للتوسع الكبير، والقيمة ultra-expanded لتوسع كبير جداً، والقيمة extra-condensed للكثافة عالية، والقيمة ultra-condensed للكثافة عالية جداً، وهناك أيضاً القيمة semi-expanded لعمل شبه توسع، والقيمة semi-condensed لعمل شبه تكثيف.

وحتى نجرب بمثال معتمد، احضرت لكم مثال من موقع mozilla.org لاختبار خاصية font-stretch، وهذا المثال به فقرات، ومكتوب بداخل كل فقرة نص يعبر عن التجربة.

كود! حيث أكواد HTML كما يلي:

```
<div>
  <p class="condensed">an elephantine lizard</p>
  <p class="normal">an elephantine lizard</p>
  <p class="expanded">an elephantine lizard</p>
</div>
```

وأكواد CSS كما يلي:

```
@font-face {
  src: url('https://mdn.mozillademos.org/files/16014/LeagueMonoVariable.ttf');
  font-family: 'LeagueMonoVariable';
  font-style: normal;
}

.container {
  border: 10px solid #f5f9fa;
  padding: 0 1rem;
  font: 1.5rem 'LeagueMonoVariable', sans-serif;
}

.condensed {
  font-stretch: 50%;
}

.normal {
  font-stretch: 100%;
}

.expanded {
  font-stretch: 200%;
}
```

ويكون الناتج من الموقع مثل الشكل التالي:

an elephantine lizard

an elephantine lizard

an elephantine lizard

وإذا اردت قراءة تفاصيل أكثر عن الخاصية font-stretch لتمدد الخط، يمكنك الدخول على الرابط التالي:

<https://developer.mozilla.org/en-US/docs/Web/CSS/font-stretch>

وهو رابط من موقع developer.mozilla.org بشرح هذه الخاصية بالتفصيل.

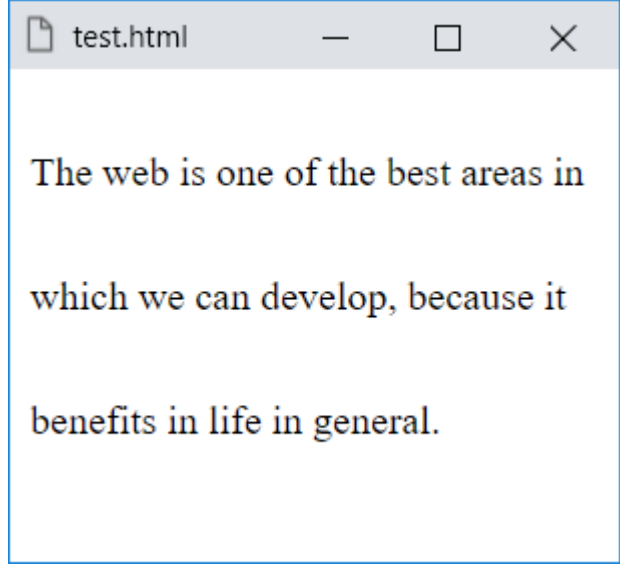
طول السطر line-height

خاصية line-height للتحكم في طول كل سطر، والتجربة كما يلي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      p{
        line-height:50px;
      }
    </style>
  </head>
  <body>
    <p>
      The web is one of the best
      areas in which we can develop,
      because it benefits in life in general.
    </p>
  </body>
</html>
```

وبعد تشغيل الصفحة بعرض قليل تظهر كما بالشكل التالي:



لاحظ أنه تم تقليد عرض نافذة المتصفح، لتظهر الفقرة على أكثر من سطر بشكل متعمد.

لاحظ أن كل سطر أصبح طوله 50px لأنه تم ضبط خاصية طول السطر line-height على القيمة 50px.

لاحظ أن الـ 50px للسطر كله، أي ما فوقه وما تحته والكلاص.

خاصية الخط font

خاصية الخط font، تجمع القيم المتعلقة بها، حيث يتم للخط ضبط، نمط font-style، واختلاف font-variant، وثقل font-weight، وحجم مع طول سطر font-size/line-height، ونوع الخط font-family، كلهم في خاصية واحدة، وهي خاصية font، وللتجربة تابع المثال التالي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    p{
      font:italic small-caps bold 40px/50px impact;
    }
  </style>
</head>
<body>
  <p>Welcome to Font Attribute</p>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



محاذاة النص text-align

خاصية محاذاة الخط text-align، تجعلك تتحكم في مكان الخط كتنسيق، فهل هو على اليمين، أم على اليسار، أم في المنتصف، الى آخر هذه الأنواع المشهورة من المحاذاة، والقيمة الافتراضية لها هي left. **كود!** قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head></head>
<body>
  <p style="text-align:left">
    Welcome to Text-Align Attribute
    Welcome to Text-Align Attribute
    Welcome to Text-Align Attribute
  </p>
  <p style="text-align:right">
    Welcome to Text-Align Attribute
    Welcome to Text-Align Attribute
    Welcome to Text-Align Attribute
  </p>
  <p style="text-align:center">
    Welcome to Text-Align Attribute
    Welcome to Text-Align Attribute
    Welcome to Text-Align Attribute
  </p>
  <p style="text-align:justify">
    Welcome to Text-Align Attribute
    Welcome to Text-Align Attribute
    Welcome to Text-Align Attribute
  </p>
</body>
</html>
```

وبعد تشغفل الصفءة تظهر كما بالشكل التالي:



لاءظ فف الفءة الأولى left وهو الافءامف، والفءة الثانية right ءفء المءااة على الئمف، والفءة الثالثة center ءفء النص فف المنءصف، والفءة الرابعة justify ءفء تم مءااة النص فأصء كل سءر موازف لباقي الأسءر.

انءبه! فانه فوءء ءاصفة text-justify لءءفءف نوع المءااة، وءأءء القفمة inter-word ءه المءااة بفه الكلمات، وءأءء القفمة inter-character ءه المءااة بفه الءروف.

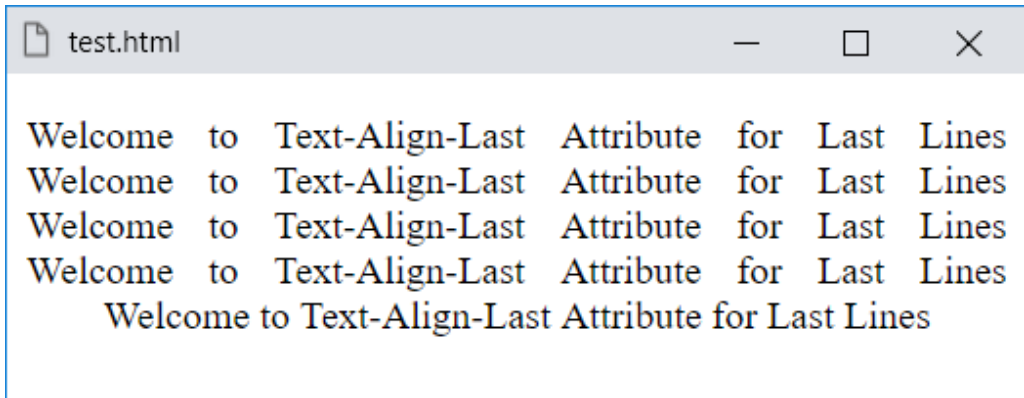
محاذاة آخر النص text-align-last

خاصية محاذاة آخر الخط text-align-last، حيث تقوم بعمل محاذاة عادية للنص كما تعلمنا من قبل، ولكنه آخر سطر في الفقرة تختلف محاذاته عما قبله، والتجربة بفضل الله كما يلي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head></head>
<body>
  <p style="text-align:justify;text-align-last:center">
    Welcome to Text-Align-Last Attribute for Last Lines
    Welcome to Text-Align-Last Attribute for Last Lines
    Welcome to Text-Align-Last Attribute for Last Lines
    Welcome to Text-Align-Last Attribute for Last Lines
    Welcome to Text-Align-Last Attribute for Last Lines
  </p>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

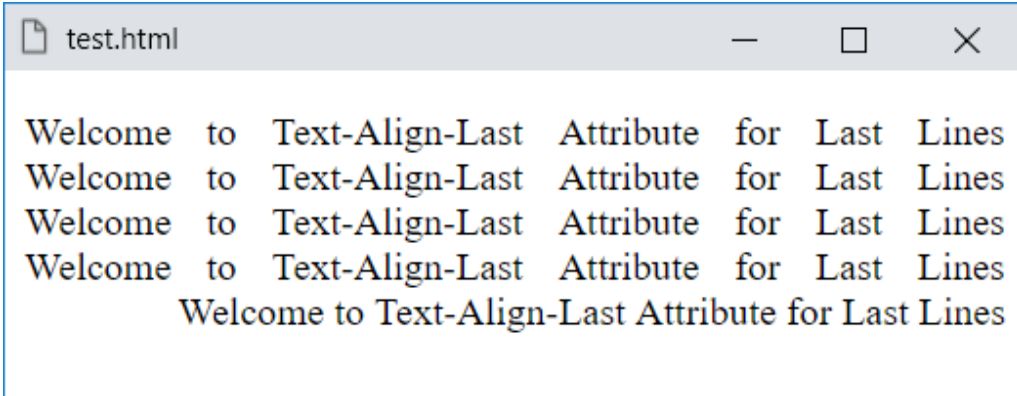


لاحظ محاذاة الفقرة، فيما عدا آخر سطر في المنتصف.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head></head>
<body>
<p style="text-align:justify;text-align-last:right">
Welcome to Text-Align-Last Attribute for Last Lines
Welcome to Text-Align-Last Attribute for Last Lines
Welcome to Text-Align-Last Attribute for Last Lines
Welcome to Text-Align-Last Attribute for Last Lines
Welcome to Text-Align-Last Attribute for Last Lines
</p>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



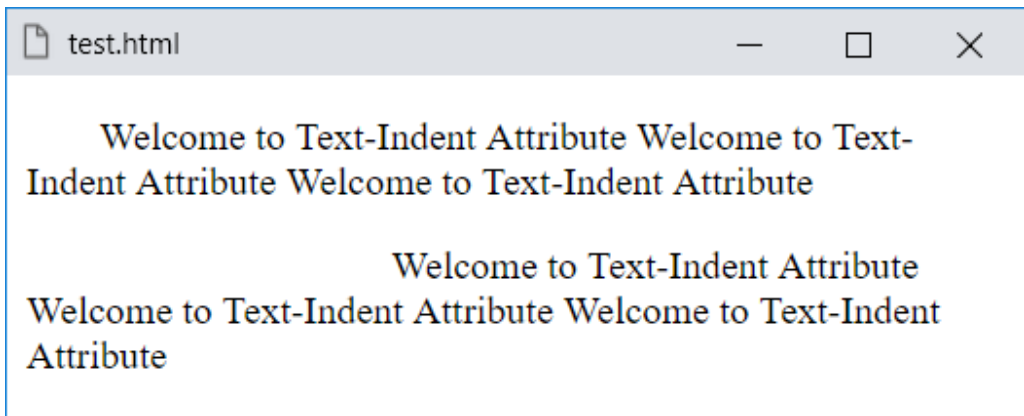
لاحظ محاذاة الفقرة، فيما عدا آخر سطر في اليمين، ويمكنك جعله في اليسار، أو غير ذلك.

المسافة البادئة للنص text-indent

يمكّن ضبط قيمة للمسافة البادئة للفقرات بالخاصية text-indent، ولاحظ مسافة فقرات بادئة 30px للأولي و150px للثانية كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head></head>
<body>
<p style="text-indent:30px;">
  Welcome to Text-Indent Attribute
  Welcome to Text-Indent Attribute
  Welcome to Text-Indent Attribute
</p>
<p style="text-indent:150px;">
  Welcome to Text-Indent Attribute
  Welcome to Text-Indent Attribute
  Welcome to Text-Indent Attribute
</p>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



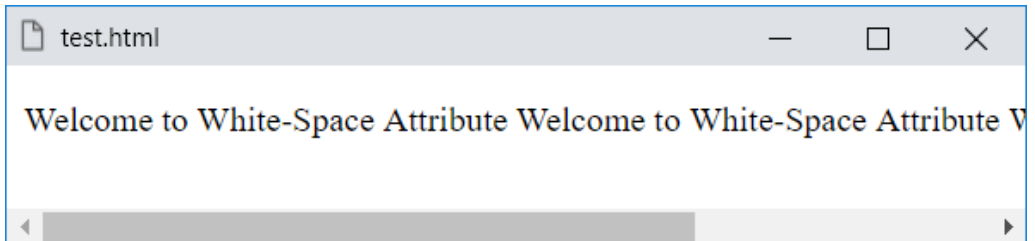
التفاف النص word wrap

خاصية **white-space**، فعند كتابة فقرة كبيرة يتم عمل التفاف للنص word wrap بشكل تلقائي واضح، عند تصغير عرض نافذة المتصفح، ودور الخاصية **white-space** هي التحكم في هذا النص الزائد، والقيمة الافتراضية لها هي **normal**.
ويمكنك منع التفاف النص بالقيمة **nowrap**، فيظهر شريط التمرير **scroll** الأفقي أسفل الصفحة، للتحكم منه في قراءة النص، كما بالمثال التالي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head></head>
<body>
  <p style="white-space:nowrap;">
    Welcome to White-Space Attribute
    Welcome to White-Space Attribute
    Welcome to White-Space Attribute
  </p>
</body>
</html>
```

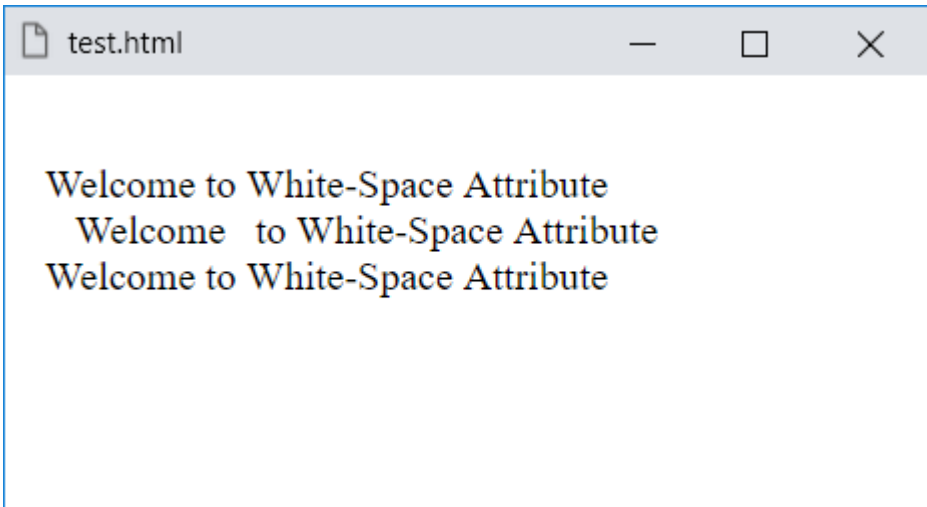
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



ويمكنك أيضاً ضبطها على القيمة pre لتكون مثل الـ pre، فتظهر
 بما قمت أنت بكتابته، كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head></head>
<body>
  <p style="white-space:pre;">
    Welcome to White-Space Attribute
      Welcome to White-Space Attribute
    Welcome to White-Space Attribute
  </p>
</body>
</html>
```

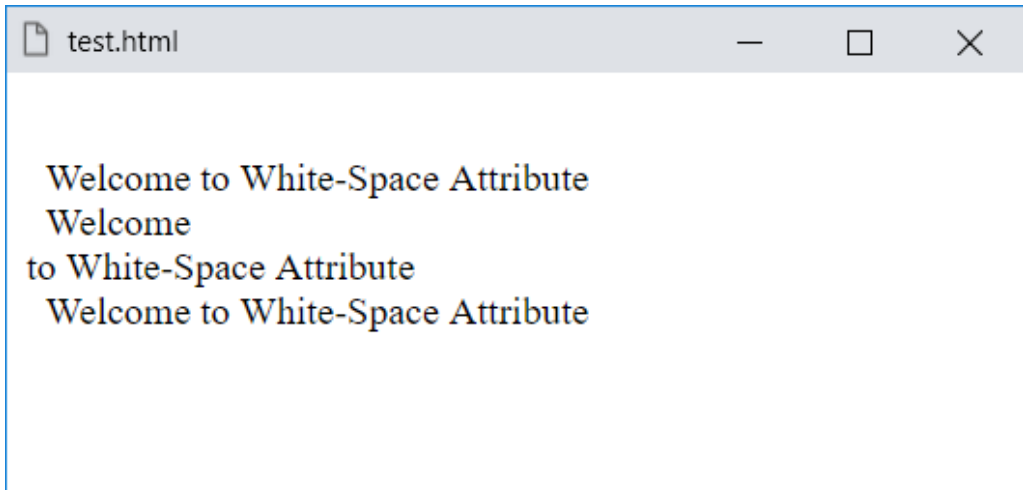
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



ويمكنك استخدام pre-wrap ليتم الاحتفاظ بتسلسل مساحة بيضاء، ويتم تقسيم السطور عند السطر الجديد بالوسم br، وعند الضرورة لملء مربعات السطور، وباختصار، الحفاظ على الـ pre مع المسافات والتفاف النص، كما بالمثال التالي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head></head>
<body>
  <p style="white-space:pre-wrap;">
    Welcome to White-Space Attribute
    Welcome<br>to White-Space Attribute
    Welcome to White-Space Attribute
  </p>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

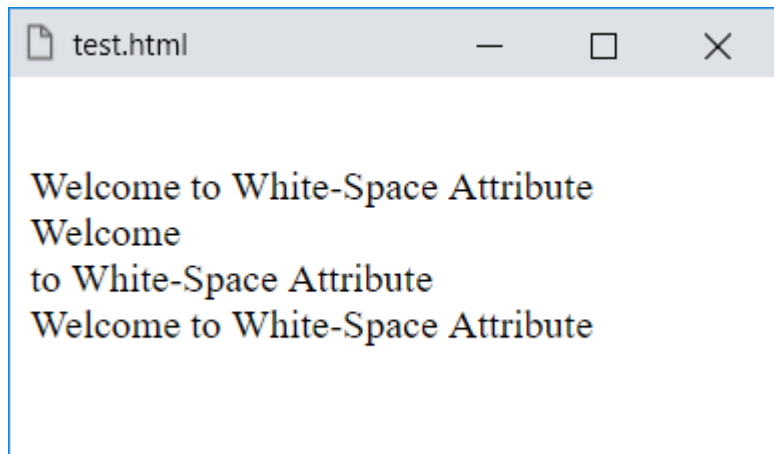


ويمكنك استخدام القيمة pre-line، ليتم إنهاء تسلسل الفراغات البيضاء في بداية السطور، ويتم تقسيم السطور عند الوسم br، وعند الضرورة لملء مربعات الخطوط، وباختصار، الحفاظ على ال pre مع حذف المسافات الموجودة به، كما بالمثال التالي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head></head>
<body>
  <p style="white-space:pre-line;">
    Welcome to      White-Space Attribute
      Welcome<br>to White-Space Attribute
    Welcome to White-Space      Attribute
  </p>
</body>
</html>
```

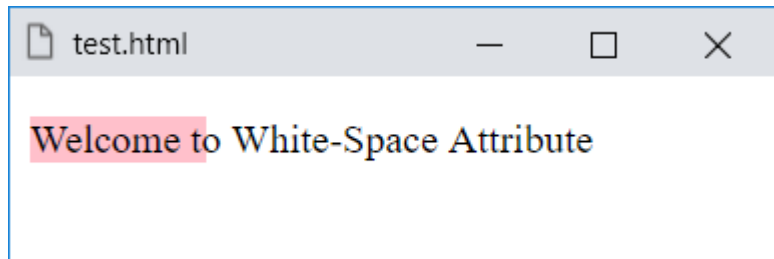
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



انتبه! فإنه إذا تم استخدام القيمة nowrap، وتخصيص عرض قليل ولا يكفي للنص، سوف يخرج النص خارج الفقرة، او الوسم، وللتوضيح سوف نخصص خلفية للفقرة ليظهر خروج النص، كما بالمثال التالي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    p{
      width:70px;
      background:pink;
      white-space:nowrap;
    }
  </style>
</head>
<body>
  <p>
    Welcome to White-Space Attribute
  </p>
</body>
</html>
```

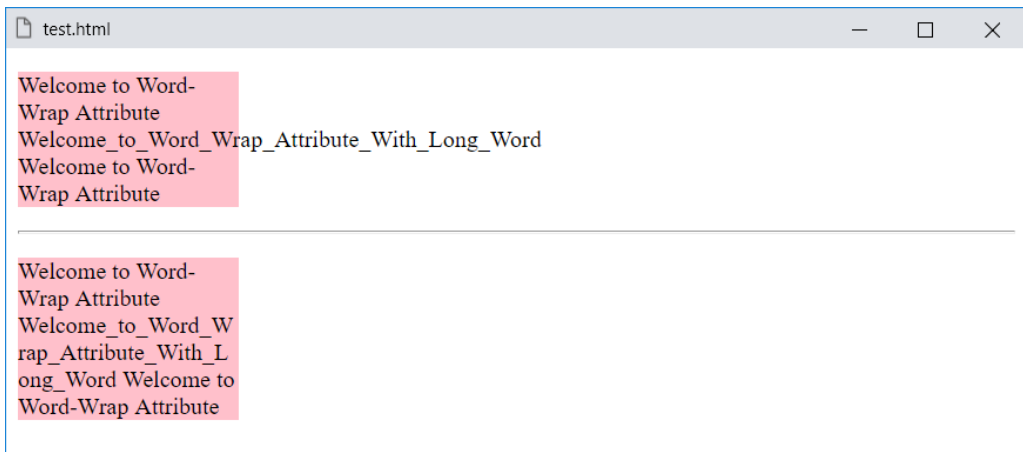
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



خاصية **word-wrap** تفيد عند استخدام كلمة كبيرة، كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>p{ width:150px;background:pink; }</style>
</head>
<body>
  <p>
    Welcome to Word-Wrap Attribute
    Welcome_to_Word_Wrap_Attribute_With_Long_Word
    Welcome to Word-Wrap Attribute
  </p>
  <hr>
  <p style="word-wrap:break-word;">
    Welcome to Word-Wrap Attribute
    Welcome_to_Word_Wrap_Attribute_With_Long_Word
    Welcome to Word-Wrap Attribute
  </p>
</body>
</html>
```

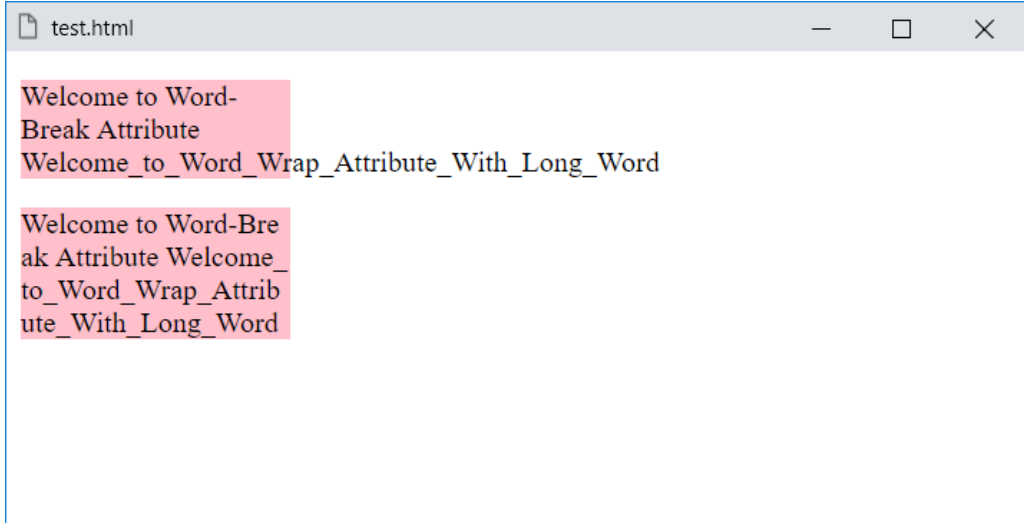
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



خاصية **word-break** تفيد عند استخدام كلمة كبيرة، كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>p{ width:150px;background:pink; }</style>
</head>
<body>
  <p style="word-break:keep-all;">
    Welcome to Word-Break Attribute
    Welcome_to_Word_Wrap_Attribute_With_Long_Word
  </p>
  <p style="word-break:break-all;">
    Welcome to Word-Break Attribute
    Welcome_to_Word_Wrap_Attribute_With_Long_Word
  </p>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

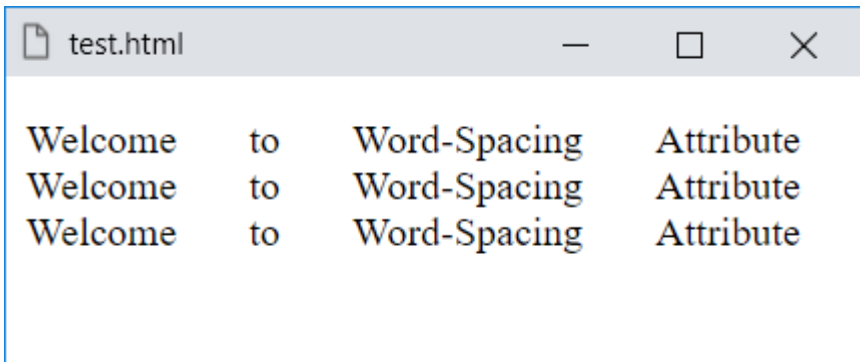


مسافات بين الكلمات word-spacing

الخاصية word-spacing تبت مسافات بين كل كلمة وأخرى، وللتجربة، سوف يتم استخدام المثال التالي بفضل الله: **كود!** قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head></head>
<body>
  <p style="word-spacing:25px;">
    Welcome to Word-Spacing Attribute
    Welcome to Word-Spacing Attribute
    Welcome to Word-Spacing Attribute
  </p>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



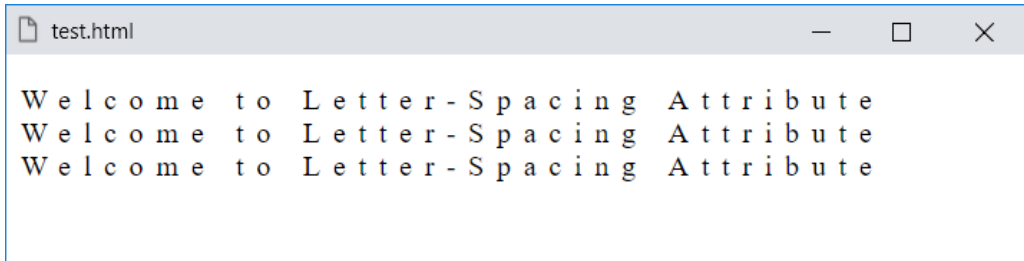
لاحظ أنه تم تثبيت 25px بين كل كلمة وأخرى.

مسافات بين الحروف letter-spacing

الخاصية letter-spacing تبت مسافات بين كل حرف وأخر،
وللتجربة، سوف يتم استخدام المثال التالي بفضل الله:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head></head>
<body>
  <p style="letter-spacing:7px;">
    Welcome to Letter-Spacing Attribute
    Welcome to Letter-Spacing Attribute
    Welcome to Letter-Spacing Attribute
  </p>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم تثبيت 7px بين كل حرف وأخر.

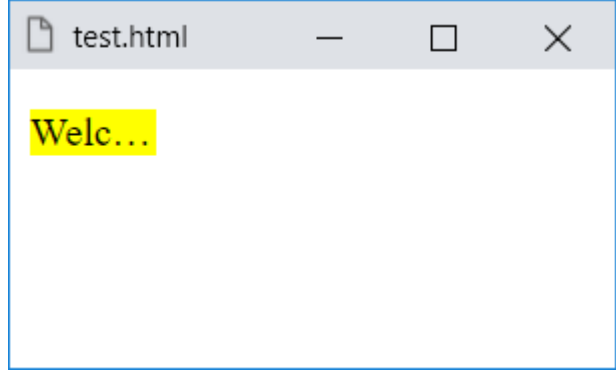
خاصية النص الزائد text-overflow

وهي تستخدم عند زيادة النص، ثم إخفاؤه بخاصية overflow، والقيمة clip هي الافتراضية لها، حيث تقص او تخفي النص، ويمكك اعطاؤها القيمة ellipsis لإظهار نقاط "..." في آخر النص، ويمكك كتابة نص ليظهر بدلاً من النقاط (النص لمتصفح Firefox فقط)، والتجربة بالمثال التالي سوف تبين لنا هذا بفضل الله، كما يلي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    p{
      background:yellow;
      width:50px;
      white-space:nowrap;
      overflow:hidden;
    }
  </style>
</head>
<body>
  <p style="text-overflow:ellipsis">
    Welcome to Text Overflow
  </p>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ ظهور نقاط "..." بدلاً من النص المختفي.

لاحظ أننا لم نجرب القيمة clip لأنها هي الافتراضية.

لاحظ لم نجرب قيمة نص بديل لأنها حتى الآن تعمل فقط على Firefox.

خط ديكور النص text-decoration-line

يمكنك مع خلال هذه الخاصية عمل زخرفة للنص، ونضع به خط، كما يلي:

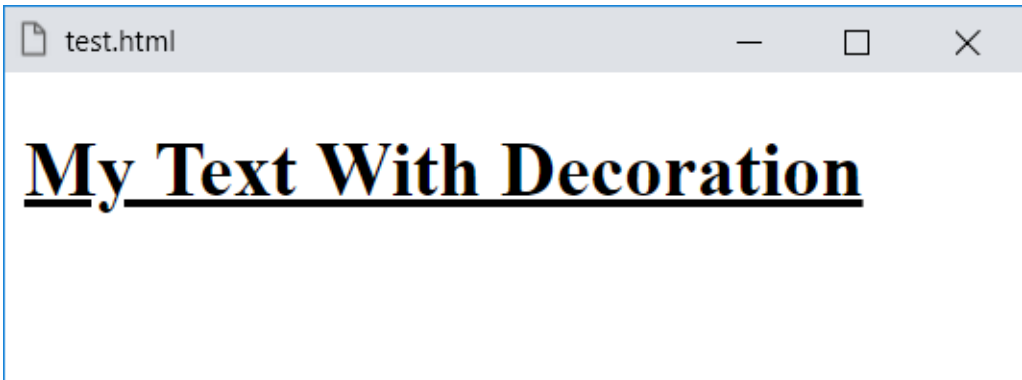
كود! قم بكتابة الكود التالي ثم احفظه:

```

<!DOCTYPE html>
<html>
<head>
  <style>
    h1{
      text-decoration-line:underline;
    }
  </style>
</head>
<body>
  <h1>My Text with Decoration</h1>
</body>
</html>

```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن يوجد خط تحت النص، بالقيمة underline.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    h1{
      text-decoration-line:line-through;
    }
  </style>
</head>
<body>
  <h1>My Text with Decoration</h1>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

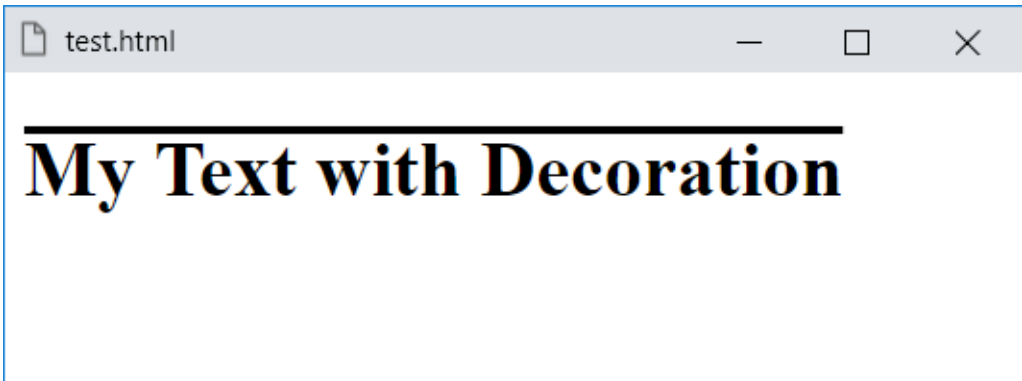


لاحظ أنه يوجد خط في منتصف النص، بسبب القيمة line-through.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    h1{
      text-decoration-line:overline;
    }
  </style>
</head>
<body>
  <h1>My Text with Decoration</h1>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

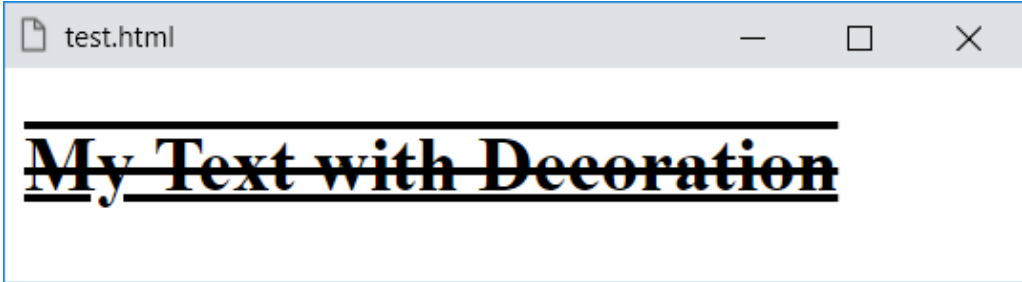


لاحظ أن يوجد خط فوق النص، بالقيمة overline.

والآن **انتبه!** فإنه مع التركيز مع الأمثلة السابقة، يوجد سؤال لابد أن تسأله لنفسك، وهو، كيف أجمع بينهم!؟، والجواب بفضل الله كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  h1{
    text-decoration-line:overline line-through underline;
  }
</style>
</head>
<body>
<h1>My Text with Decoration</h1>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



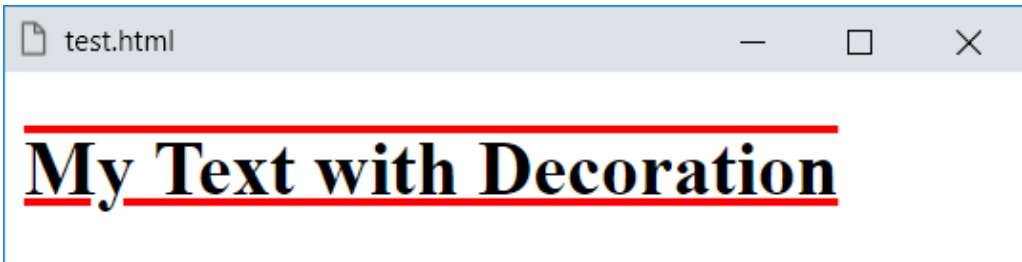
لاحظ أنه تم اجمع بينهم واستخدامهم الثلاثة لنفس النص، مع طريق مسافة فقط بين كل قيمة وأخرى، بفضل الله بسهولة.

لون ديكور النص text-decoration-color

يمكنك تغيير لون زخرفة للنص عن طريق هذه الخاصية، كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    h1{
      text-decoration-line:overline underline;
      text-decoration-color:red;
    }
  </style>
</head>
<body>
  <h1>My Text with Decoration</h1>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



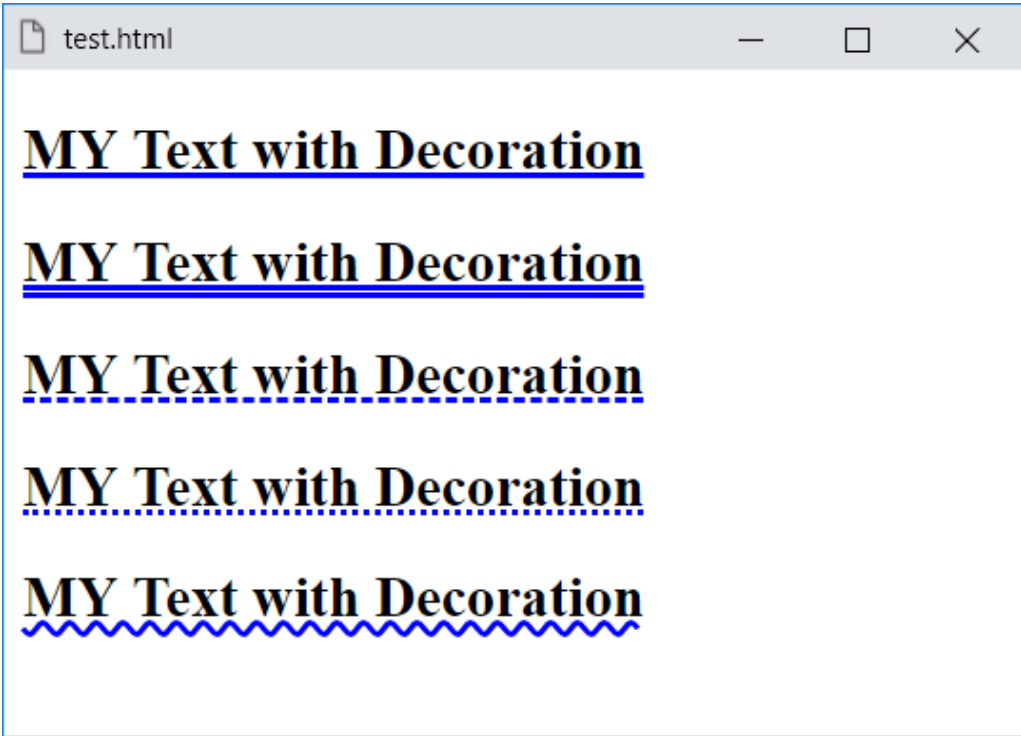
لاحظ تغيير لون الديكور، بالخاصية text-decoration-color.

شكل ديكور النص text-decoration-style

يمكنك تغيير شكل زخرفة النص عن طريق هذه الخاصية، كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    h2{
      text-decoration-line:underline;
      text-decoration-color:blue;
    }
  </style>
</head>
<body>
  <h2 style="text-decoration-style:solid">
    MY Text with Decoration</h2>
  <h2 style="text-decoration-style:double">
    MY Text with Decoration</h2>
  <h2 style="text-decoration-style:dashed">
    MY Text with Decoration</h2>
  <h2 style="text-decoration-style:dotted">
    MY Text with Decoration</h2>
  <h2 style="text-decoration-style:wavy">
    MY Text with Decoration</h2>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



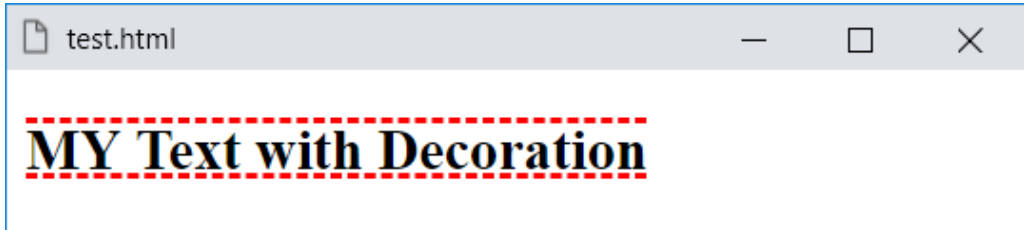
لاحظ أنه تم استخدام أكثر من نمط style بفضل الله، حيث استخدمنا الخط العادي solid وهو الافتراضي، والخط المزدوج double، وخط به تقطيع dashed، وخط به نقط dotted، وخط مموج .wavy.

خاصية ديكور النص text-decoration

خاصية ديكور النص text-decoration، تجمع القيم المتعلقة بها، حيث يتم ضبط، خط ديكور النص text-decoration-line، ويتم ضبط اللون لديكور النص text-decoration-color، ويتم ضبط النمط لديكور النص text-decoration-style، كلهم في خاصية واحدة، وسوف نجرب هذا بفضل الله فيما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    h2{
      text-decoration:underline overline red dashed;
    }
  </style>
</head>
<body>
  <h2>MY Text with Decoration</h2>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



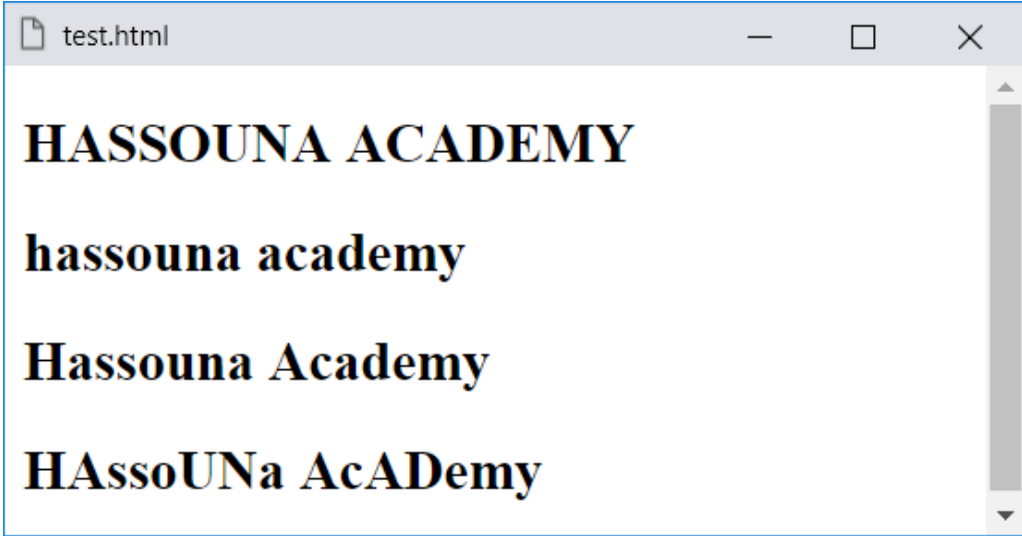
تحويلات النص text-transform

هي خاصية تمنحك التحكم في تقلبات أو تحويلات النصوص، مثلاً حالة الحروف الكبيرة uppercase والصغيرة lowercase أو الاثنية معاً بضابط طريقة camelCase، وبفضل الله المثال التالي يبيّن ذلك بوضوح، ولكنه **انتبه!** رجاء التركيز في المثال، كما يلي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head></head>
<body>
  <h2 style="text-transform:uppercase">
    hassouna academy
  </h2>
  <h2 style="text-transform:lowercase">
    HASSOUNA ACADEMY
  </h2>
  <h2 style="text-transform:capitalize">
    hassouna academy
  </h2>
  <h2 style="text-transform:capitalize">
    hAssoUNa acADemy
  </h2>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن النص الأول كانت حروف صغيرة وتم تحويلها الى حروف كبيرة،
ع طريق القيمة `uppercase`.

لاحظ أن النص الثاني كانت حروف كبيرة وتم تحويلها الى حروف صغيرة،
ع طريق القيمة `lowercase`.

لاحظ أن النص الثالث به حروف صغيرة وتم تحويلها `camelCase`،
ع طريق القيمة `capitalize`.

لاحظ أن القيمة `capitalize` لا تمس الحروف الموجودة بالمنتصف،
وتعدل أول حرف من كل كلمة فقط.

ظل النص text-shadow

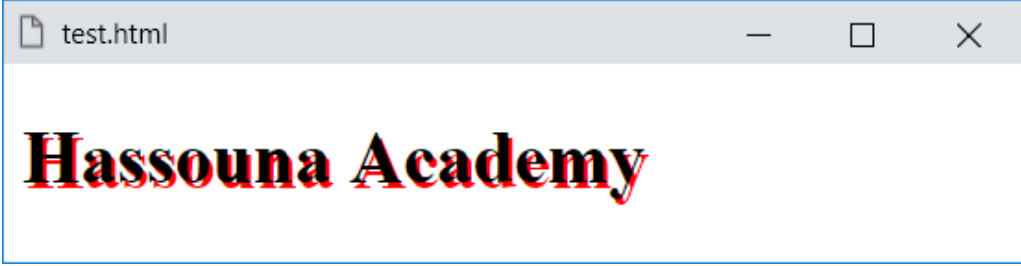
- يستخدم ظل النص أو text-shadow في تحسين شكل النص، ولفت الانتباه له، ولكه هيا بنا نتفهم على بعض الأمور في البداية:
- الظل هو نسخة ثانية من النص، وتكون خلف النسخة الأصلية
 - يتم تحديد قيمة أفقية x لتحديد موضع الظل أفقياً
 - يتم تحديد قيمة رأسية y لتحديد موضع الظل رأسياً
 - يتم تحديد قيمة radius، بحيث لو زادت يكون ظل مطموس وغير واضح
 - يتم تحديد اللون، الذي سوف يظهر الظل به
- والآن هيا نجرب، كما بالمثال التالي:
- كود!** قم بكتابة الكود التالي ثم احفظه:

```

<!DOCTYPE html>
<html>
<head>
  <style>
    h1{
      text-shadow:2px 2px red;
    }
  </style>
</head>
<body>
  <h1>
    Hassouna Academy
  </h1>
</body>
</html>

```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ انشاء ظل أحمر، وتحريكه أفقياً 2px ورأسياً 2px.

والآن بالمثال التالي سوف يتضح إن شاء الله أن الظل نسخة من النص.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    h1{
      text-shadow:35px 25px red;
    }
  </style>
</head>
<body>
  <h1>
    Hassouna Academy
  </h1>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

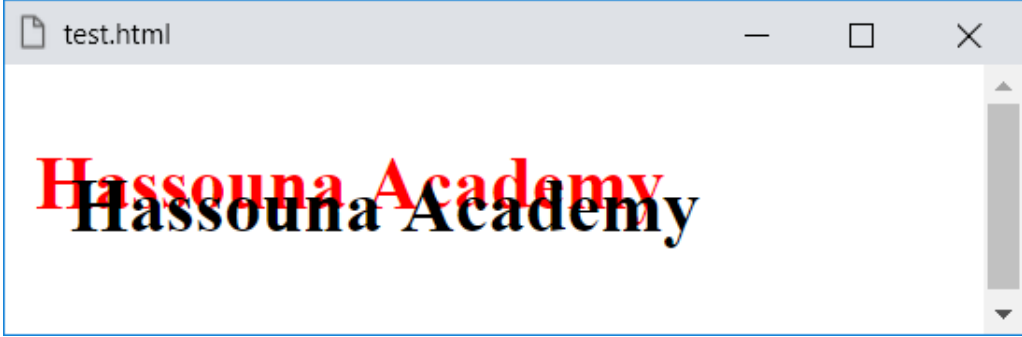


لاحظ انشاء ظل أحمر، وإبعاده أفقياً 35px ورأسياً 25px، فظهر وكأنه نص آخر.

ويمكن تحريكه بقيمة سالبة، ليظهر في الاتجاه العكسي، كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  h1{
    padding:20px;
    text-shadow:-15px -10px red;
  }
</style>
</head>
<body>
  <h1>Hassouna Academy</h1>
</body>
</html>
```

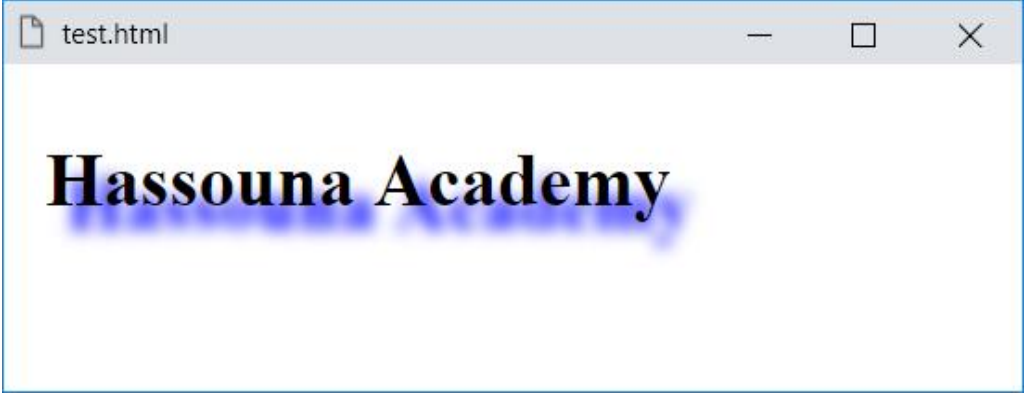
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



ويمكن زيادة قيمة radius للتشويش، كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  h1{
    padding:10px;
    text-shadow:9px 10px 11px blue;
  }
</style>
</head>
<body>
  <h1>Hassouna Academy</h1>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن ظل النص تحرك أفقياً بقيمة 9px.

لاحظ أن ظل النص تحرك رأسياً بقيمة 10px.

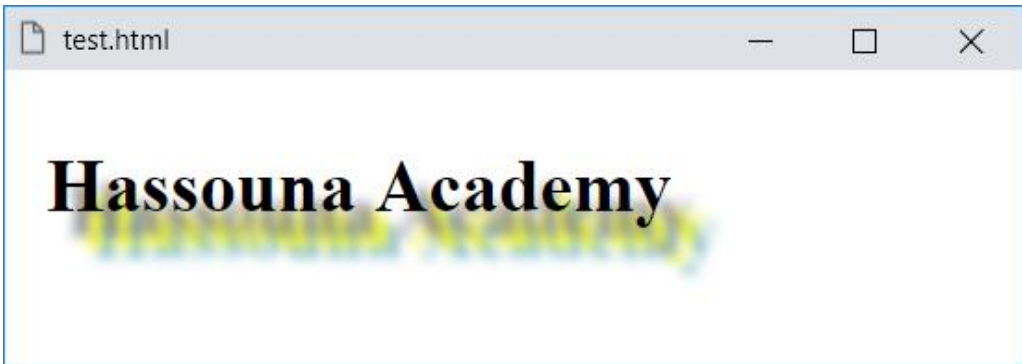
لاحظ أن ظل النص تم التشويش عليه بقيمة 11px.

انتبه! فإنه يمكنك إنشاء ظل جميل للنص، عن طريق إنشاء أكثر من ظل بفضل الله وحده، فكل قليلاً في القدرة التي أعطاها لك الله رب العالمين في التفكير والتمييز بفضل الله، سوف تجد أنه يمكنك أن تصنع أكثر من ظل، عن طريق إنشاء ظل أول، ثم ظل ثاني، ثم ظل ثالث، وهكذا...، ولكن السؤال الذي يطرح نفسه هنا، هو، كيف أفصل بينهم؟!، والإجابة هي بكل سهولة بفضل الله عن طريق كوما "،" بين كل ظل وآخر، وكما قلت لك **انتبه!** فلا تنسى فضل الله عليك في أن تفهم، وتفكر، والمثال كما يلي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1{
padding:10px;
text-shadow:9px 10px 7px gray,
15px 15px 7px yellow,
20px 20px 7px #71b2c7;
}
</style>
</head>
<body>
<h1>Hassouna Academy</h1>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



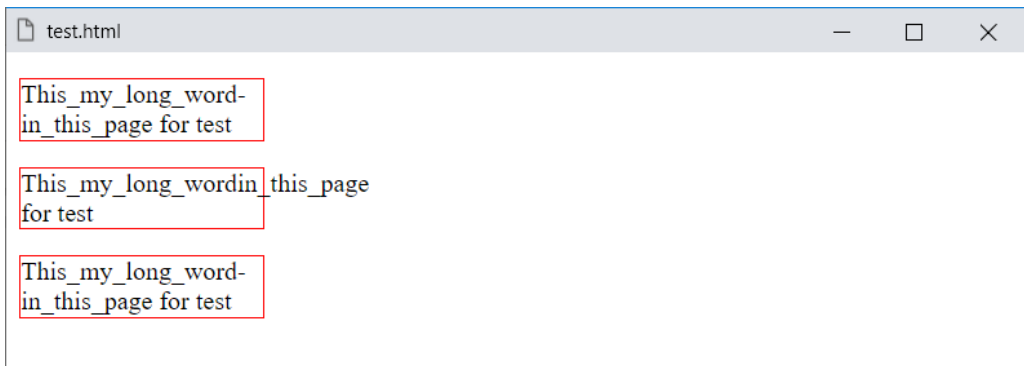
لاحظ انشاء ظل رمادي، ثم ظل أصفر أبعد منه، ثم ظل من مشتقات الأزرق ابعد من الأصفر، فظهر الشكل كما هو واضح بالصورة بفضل الله.

الواصلات hyphens

تحدد الخاصية hyphens كيفية ربط الكلمات عند التقاف النص عبر أسطر متعددة، ويمكنك منع الواصلات بالقيمة none، أو استخدام الواصلات في نقاط محددة يدوياً داخل النص بالقيمة manual، أو السماح للمتصفح بإدخال الواصلات تلقائياً عند الضرورة بالقيمة auto، وسوف نجرب **بالكود!** التالي إن شاء الله:

```
<!DOCTYPE html>
<html>
<head>
<style>
  p{ width:150px; border:1px solid red; }
  #p1{ hyphens: manual; }
  #p2{ hyphens: none; }
  #p3{ hyphens: auto; }
</style>
</head>
<body>
<p id="p1">This_my_long_word&shy;in_this_page for test</p>
<p id="p2">This_my_long_word&shy;in_this_page for test</p>
<p id="p3">This_my_long_word&shy;in_this_page for test</p>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

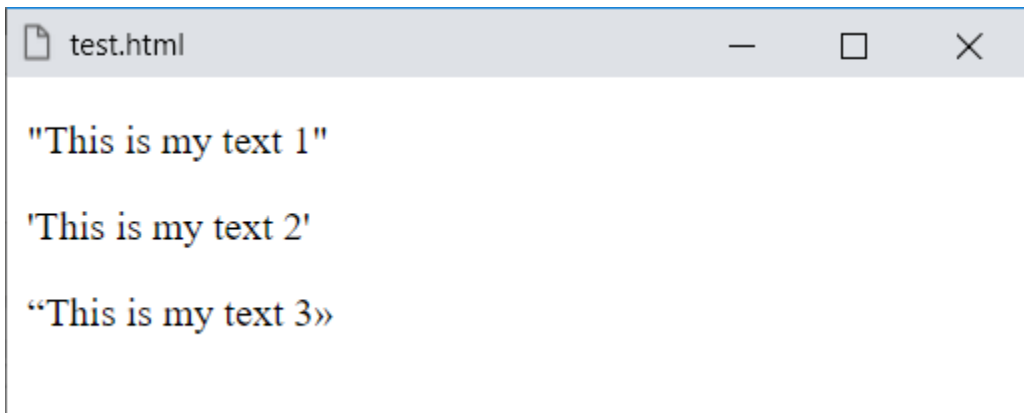


الاقتباس quotes

يمكّن التحكم في الاقتباس، بدونه ونضع أي زيادات في النص.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    #p2{ quotes: " ' " ' "; }
    #p3{ quotes: "“” "»"; }
  </style>
</head>
<body>
  <p><q id="p1">This is my text 1</q></p>
  <p><q id="p2">This is my text 2</q></p>
  <p><q id="p3">This is my text 3</q></p>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



حجم رمز tab في النص tab-size

الخاصية tab-size تأخذ قيمة رقمية لتقوم بتحديد عدد المسافات في ال tab الواحدة، والافتراضي لها هو 8، وعند كتابة النصوص يمكن احتياج وضع tab، بدلاً من space، ولكن لو تم وضع tab فله تملكه مع رؤيته على الصفحة، فلذلك سوف نستخدم الـ pre ليثبت النص كما هو، ثم نجرب tab-size، كما يلي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    #p1{ tab-size:20; }
    #p2{ tab-size:15; }
    #p3{ tab-size:10; }
    #p4{ tab-size:8; }
  </style>
</head>
<body>
  <pre id="p1">This      is      my      paragraph 1</pre>
  <pre id="p2">This      is      my      paragraph 2</pre>
  <pre id="p3">This      is      my      paragraph 3</pre>
  <pre id="p4">This      is      my      paragraph 4</pre>
  <pre id="p5">This      is      my      paragraph 5</pre>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

```
test.html
This          is          my          paragraph 1
This          is          my          paragraph 2
This          is          my          paragraph 3
This          is          my          paragraph 4
This          is          my          paragraph 5
```

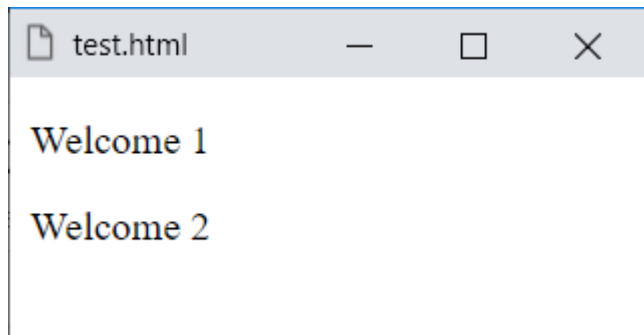
لاحظ أنه تم استخدام الوسم `pre` لتشعر صفحة الويب بال `tab`، ولاحظ أن آخر وسم `p` لم يتم تنسيقه، مع العلم أن الذي قبله تم ضبط خاصية `tab-size` له على 8، والوسميه ظاهريه على الصفحة وكأن مسافاتهم واحدة كما هو واضح بالشكل السابق، لأنه كما ذكرنا من قبل أن الافتراضي لخاصية `tab-size` هو 8.

تحكم التحدید user-select

الخاصية user-select تأخذ القيمة auto كقيمة افتراضية، ويمكن ضبطها على القيمة none لمنع التحدید.
كود! قم بكتابة الكود التالي ثم احفظه:

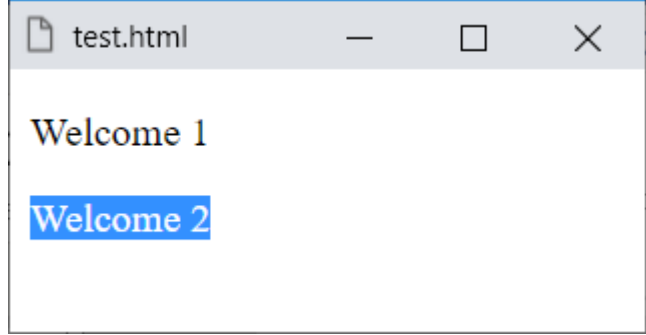
```
<!DOCTYPE html>
<html>
  <head>
    <style>
      #p1{ user-select: none; }
      #p2{ user-select: text; }
    </style>
  </head>
  <body>
    <p id="p1">Welcome 1</p>
    <p id="p2">Welcome 2</p>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



والآن قم بتجربة تحدید الـ id المكتوب بداخله Welcome 1، فله تستطيع تحدیده، لأنه أخذ القيمة none لخاصية user-select.

والآن اضغط على Ctrl+A على صفحة الويب لتدي مثل الشكل التالي:



وكما تلاحظ تم التحديد على الكك فيما عدا الوسم p الأول، ولكنه الوسم الثاني أخذ القيمة text لتحديد النص، فتم تحديده لأنه قابل للتحديد.

طريقة الكتابة writing-mode

هذه الخاصية تتحكم في الكتابة، فيتم قلب النص بشكل كلي، حيث يمكنه استخدام القيمة horizontal-tb للأفقي، والقيمة vertical-rl للرأسي أفقياً إذاً يمينا، والقيمة vertical-lr للرأسي وأفقياً إذاً يساراً، وسوف يتم التجربة بفضل الله في المثال التالي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      #b1{ writing-mode: vertical-rl; }
      #b2{ writing-mode: vertical-lr; }
      #b3{ writing-mode: horizontal-tb; }
      #ar{ direction:rtl; }
    </style>
  </head>
  <body>
    <p>Welcome to <b id="b1">Egypt</b> country</p>
    <p id="ar">دائماً <b id="b2">مصر</b> أهلاً بكم في</p>
    <p>Welcome to <b id="b3">Egypt</b> country</p>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



تعليق الترقيم hanging-punctuation

تحدد الخاصية hanging-punctuation علامات الترقيم المعلقة، ما إذا كان يجب أن يتم تعليق علامة الترقيم في بداية سطر النص أو نهايته، وقد يتم وضع علامات الترقيم المعلقة خارج مربع الخط، ولكنه **انتبه!**، فإن هذه الخاصية تعمل فقط على متصفح safari إصدار 10 فيما اعلى، وحتى الآن لا تعمل على أي متصفح آخر، ولذلك فلم يتم شرحها أو تجربتها في هذا الكتاب.

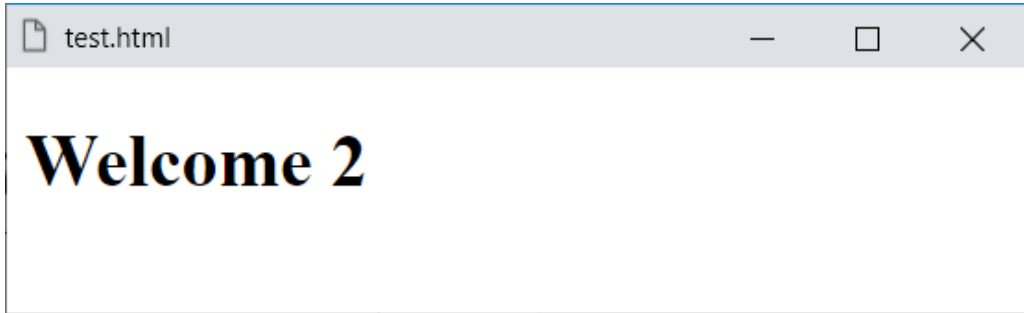
العرض Display

المقصوء هنا هو طرفة عرض العناصر على صفءة الوفب؁ أو كما ءكرنا مع قبل فف ءئاب "صخرة الأساس" حسونة أكاڤمفة فف اللغة html أنه هنا ء عناصر ءنزل سطر مءل الوسء h1 والوسء div وعناصر أءرفف لا ءنزل سطر مءل الوسء img والوسء span؁ ولكه الآن مع موضوء العرض display سوف فكون الأمر أءءر ءفصفاً مع ءلك إن شاء الله.

ءوء! قم بءءابة الءوء ءءالف اءءل العنصر body ثم اءفظه:

```
<input style="display:none">  
<h1 style="display:none">Welcome 1</h1>  
<h1>Welcome 2</h1>
```

وبء ءشءفل الصفءة ءظهر كما بالشءك ءءالف:



لاءظ أنه عءء ضبط الخاصفة display على القفمة none ففم ءءف العنصر ءماماً وءأنه ففءف موءوء على الصفءة.

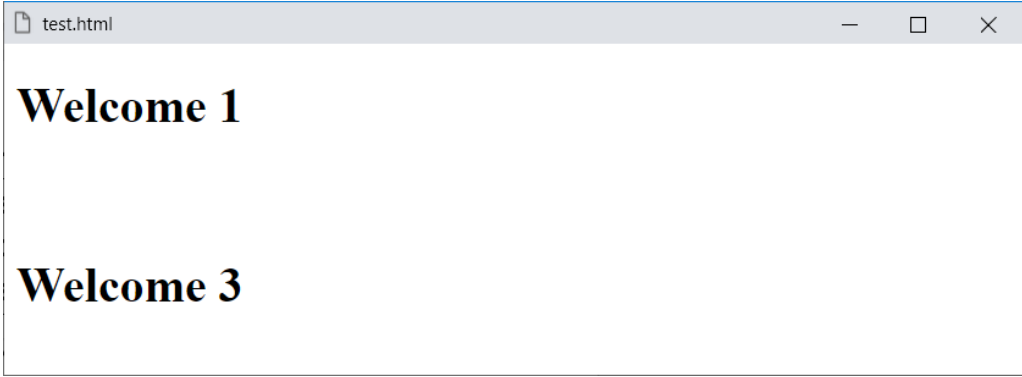
ظهور العناصر visibility

يمكن إخفاء العناصر وإظهارها مع الاحتفاظ بمكانها محجوز كما هو.

كود! قم بكتابة الكود التالي داخل العنصر body ثم احفظه:

```
<h1 style="visibility:visible">Welcome 1</h1>
<h1 style="visibility:hidden">Welcome 2</h1>
<h1 style="visibility:visible">Welcome 3</h1>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ الفارق الموجود بين Welcome 1 و Welcome 3 وذلك لأن مكان Welcome 2 محفوظ على الصفحة، ولكنه مخفي وغير ظاهر، بعكس display:none التي تحذف العنصر تماماً.

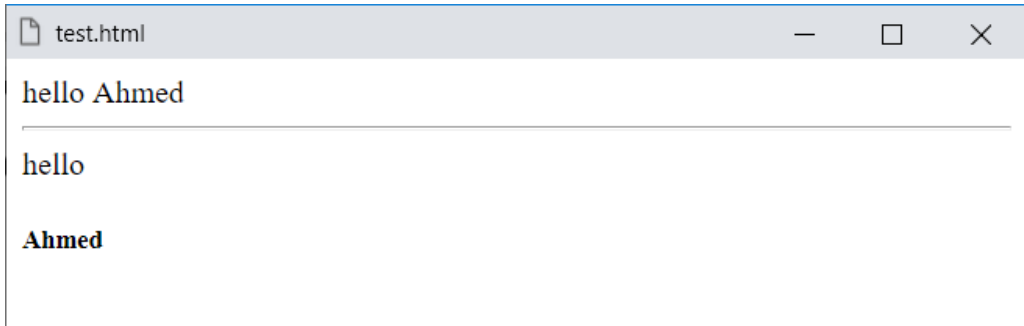
عنصر على السطر inline

هو العنصر الذي يكون مع النسق، وكأنه حرف يتم كتابته أو كلمة في سياق النص، مثل الوسم span والوسم b والوسم i والوسم u والوسم a، الى آخر هذه الوسوم.

كود! قم بكتابة الكود التالي داخل العنصر body ثم احفظه:

```
hello  
<span>Ahmed</span>  
<hr>  
hello  
<h5>Ahmed</h5>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



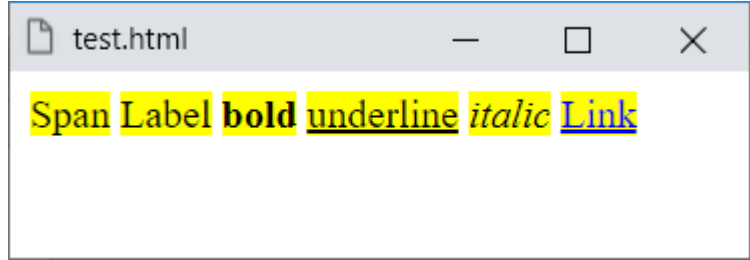
لاحظ أنه عند استخدام الوسم span في سياق النص، أصبح طبيعياً وكأنه مكتوب مع النص، باختلاف الوسم h5 الذي لم يسير مع النص.

انتبه! فإن العناصر الـ inline لا تقبل زيادة في الطول أو العرض، كما سيوضح المثال التالي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      span, label, b, u, i, a{
        background-color:yellow;
        width:300px;
        height:300px;
      }
    </style>
  </head>
  <body>
    <span>Span</span>
    <label>Label</label>
    <b>bold</b>
    <u>underline</u>
    <i>italic</i>
    <a href="#">Link</a>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم التأيد بخاصية width وخاصية height على الوسم span والوسم label والوسم b والوسم u والوسم i والوسم a ومع ذلك له يتأثر أحدهم أبداً عند التغيير في العرض أو الطول، فهو يزيد تلقائياً حسب محتواه فقط، فإذا تم زيادة حجم الخط للعناصر التي من نوع inline عن طريق الخاصية font-size، فسوف يتأثر ويزيد الطول والعرض حسب المحتوى وحجمه على الصفحة.

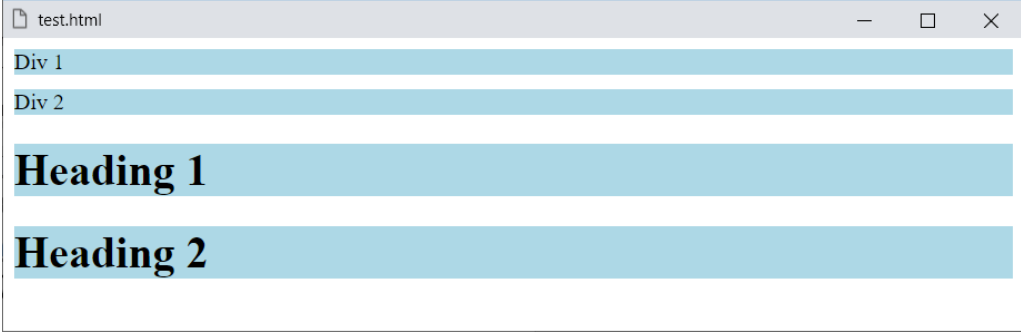
عناصر الكتلة block

هو العنصر الذي يجزئ لنفسه مكان كامل، ويأخذ عرض الصفحة بالكامل بشكل افتراضي، وينزل سطر في آخره، مثل الوسم `div` والوسم `h1` والوسم `hr` والوسم `form` والوسم `pre` والوسم `table` والوسم `ol` والوسم `nav`، الى آخر هذه الوسوم.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      div,h1,h2{
        margin-bottom:10px;
        background-color:lightblue;
      }
    </style>
  </head>
  <body>
    <div>Div 1</div>
    <div>Div 2</div>
    <h1>Heading 1</h1>
    <h1>Heading 2</h1>
  </body>
</html>
```

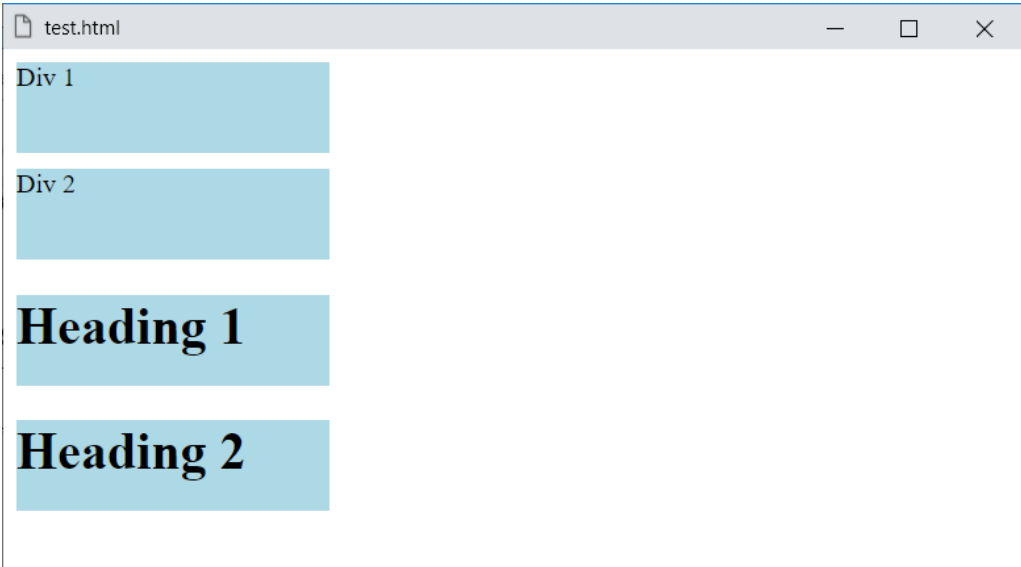
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه عند استخدام الـ div والـ h1 والـ h2، تم أخذ عرض الصفحة بالكامل مع سطر إضافي لكل عنصر، وهذه العناصر يمكن التحكم في طولها وعرضها، فإذا تم زيادة الكود التالي:

```
width:190px; height:55px;
```

سيكون عرض الصفحة كما يلي:



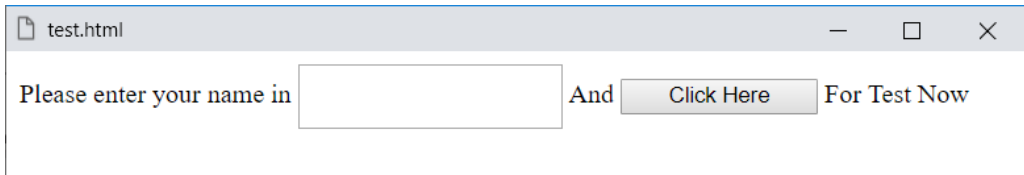
عناصر مختلفة inline-block

وهي العناصر التي تجمع بين النوع inline والنوع block معاً، حيث تتميز بأنها يمكن وضعها على سطر ووسط الكتابة مثل ال inline، وأيضاً يمكن زيادة طولها وعرضها بشكل طبيعي مثل ال block، مثل الوسم input والوسم button ووسوم أخرى مثلهم.

كود! قم بكتابة الكود التالي داخل العنصر body ثم احفظه:

```
Please enter your name in
<input style="height:33px;"> And
<button style="width:120px;">
  Click Here
</button>
For Test Now
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم استخدام الوسمين input و button، بين النص، وفي نفس الوقت يمكن التحكم في الطول والعرض بشكل طبيعي.

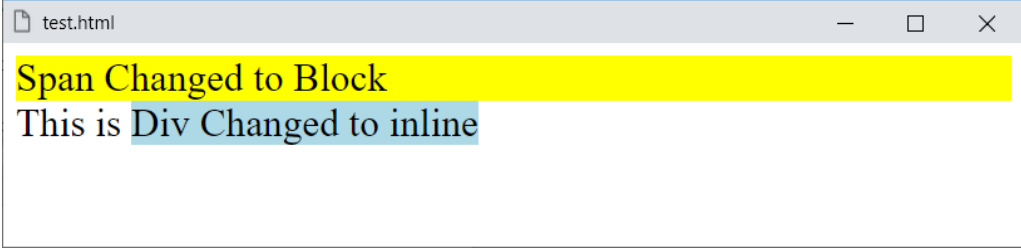
تغيير طبيعة العناصر change display

يملك استخدام أي عنصر مع قلب أو تغيير طبيعته الافتراضية، فمثلاً نجعل الـ `div` بطبيعة `inline` أو نغير الـ `span` الى `block`، أو نحول الـ `button` الى `block`، وهكذا، وذلك كما يلي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      *{ font-size:25px; }
      span{ background-color:yellow }
      div{ background-color:lightblue }
    </style>
  </head>
  <body>
    <span style="display:block">
      Span Changed to Block
    </span>
    This is
    <div style="display:inline">
      Div Changed to inline
    </div>
  </body>
</html>
```


وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ تغيير طبيعة العناصر عن طريق الخاصية display لأي عنصر.
 لاحظ أنه تم تحويل الـ span الى طبيعة الـ block.
 لاحظ أنه تم تحويل الـ div الى طبيعة الـ inline.
 لاحظ أنه يمكن استخدام أي عنصر بأي طبيعة تريدها، بفضل الله.

انتبه! فإه الخاصية display تأخذ قيم أخرى كثيرة، وكل قيمة تؤثر على هيئة العنصر وسوف يتم إظهار الفارق بين كل قيمة وأخرى كما يلي:

انتبه! سوف نقوم بضبط الخاصية width، والخاصية height، والخاصية background-color، والخاصية color، والخاصية border، والخاصية outline، والخاصية padding، والخاصية margin، للوسم div، وسوف يتم وضع حرف قبل كل div وحرف بعدها ليظهر التأثير جوار النص، وسوف يتم وضع فاصل hr بعد كل div ليظهر كل عنصر واضحا، ثم بعد ذلك سوف نري تأثير كل قيمة منهم على العنصر بعد تحديد قيم لكل هذه الخواص.

كود! قم بكتابة الكود التالي ثم احفظه:

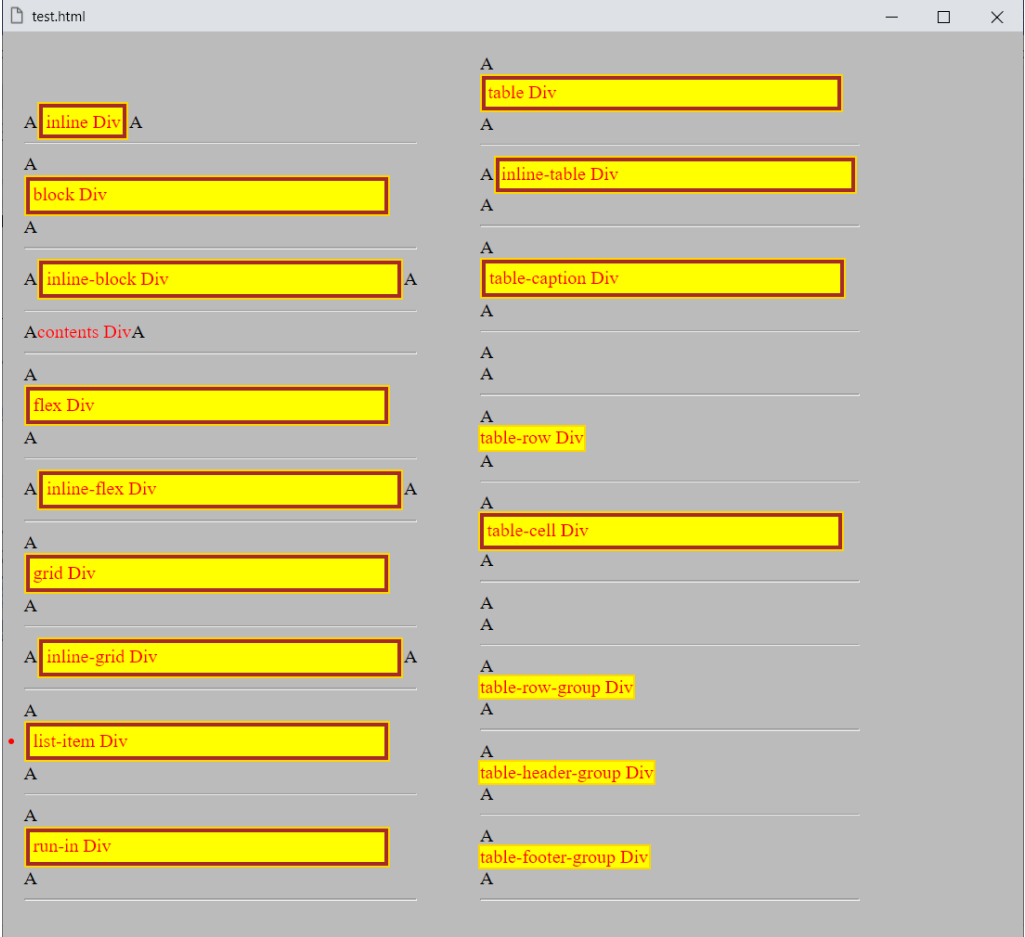
```
<!DOCTYPE html>
<html>
<head>
<style>
  body{ background:#bbb; padding:10px; }
  div{
    width:300px; height:20px;
    background-color:yellow; color:red;
    border:3px solid brown; outline:2px solid gold;
    padding:3px; margin:2px;
  }
</style>
</head>
<body>

<form style="display:inline-block; margin-right:50px;">
  <div style="display:inline">inline Div</div><hr>
  <div style="display:block">block Div</div><hr>
  <div style="display:inline-block">inline-block Div</div><hr>
  <div style="display:contents">contents Div</div><hr>
  <div style="display:flex">flex Div</div><hr>
  <div style="display:inline-flex">inline-flex Div</div><hr>
  <div style="display:grid">grid Div</div><hr>
  <div style="display:inline-grid">inline-grid Div</div><hr>
  <div style="display:list-item">list-item Div</div><hr>
  <div style="display:run-in">run-in Div</div><hr>
</form>

<form style="display:inline-block;">
  <div style="display:table">table Div</div><hr>
  <div style="display:inline-table">inline-table Div</div><hr>
  <div style="display:table-caption">table-caption Div</div><hr>
  <div style="display:table-column">table-column Div</div><hr>
  <div style="display:table-row">table-row Div</div><hr>
  <div style="display:table-cell">table-cell Div</div><hr>
  <div style="display:table-column-group">table-column-group Div</div><hr>
  <div style="display:table-row-group">table-row-group Div</div><hr>
  <div style="display:table-header-group">table-header-group Div</div><hr>
  <div style="display:table-footer-group">table-footer-group Div</div><hr>
</form>

</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن من القيم ما يقبل النص في اوله و آخره، ومنهم من لا يقبل النص، ومنهم من يقبل النص في أوله ولا يقبله في آخره، ومنهم من وضع نقطة مثل الـ `u1`، ومنهم من اختفى تماماً، ومنهم من قبل الإطار، ومنهم من لا يقبل الإطار، ومنهم من قبل لون الخلفية، ومنهم من لا يقبل لون الخلفية، ومنهم من قبل الطول والعرض، ومنهم من لا يقبل الطول والعرض، وهكذا، وسوف تحتاج إليهم حسب تنسيقك.

الصور Images

تمارين الصور Photo Exercises

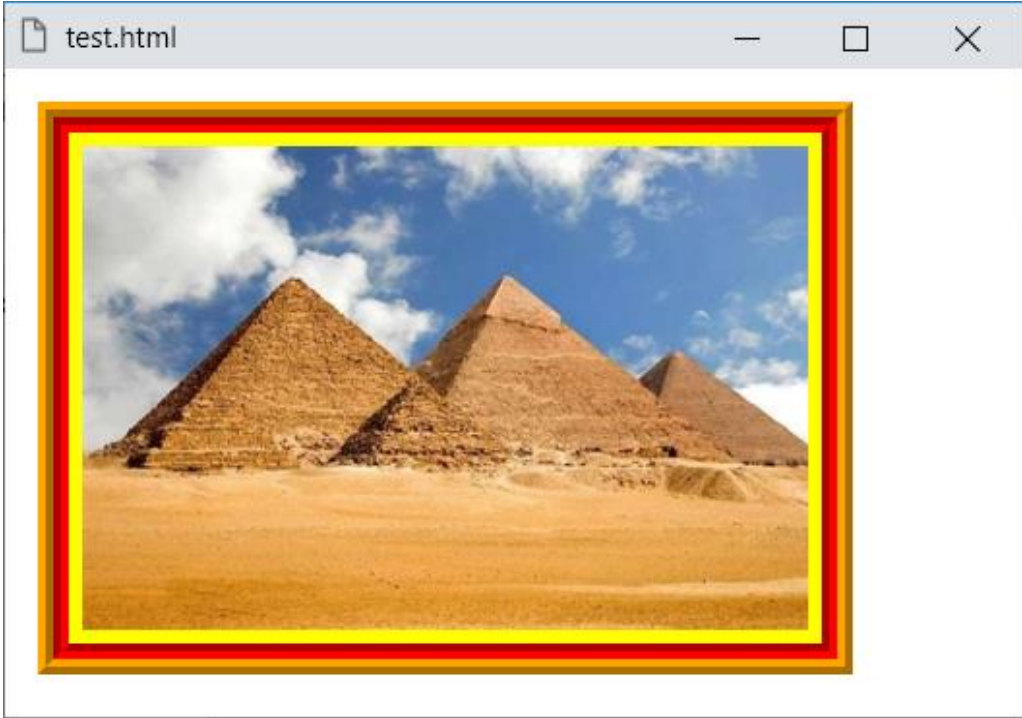
سوف نتعامل مع الصور، حيث نغير فيها كما نريد، بالخبرات السابقة التي تعلمناها حتى الآن بفضل الله، وذلك عن طريق مجموعة من الأمثلة المختلفة، كما يلي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
body{ padding:10px; }
img{
width:300px; height:200px;
background:yellow;
border:6px groove red;
outline:7px ridge orange;
padding:6px; margin:2px;
}
</style>
</head>
<body>

</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



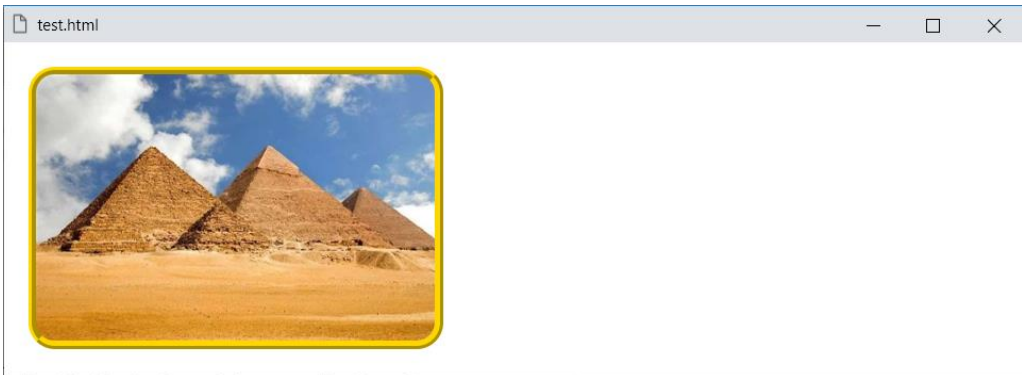
لاحظ أنه تم تكويبه إطار بارز وإطار محفور، وتم استخدام ال padding كحلية باللون الأصفر، وذلك لأن لون خلفية الصورة أصفر، والحشو حاجز بين الصورة وخلفيتها.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
body{ padding:10px; }
img{
width:300px; height:200px;
border:6px ridge gold;
border-radius:20px;
}
</style>
</head>
<body>

</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

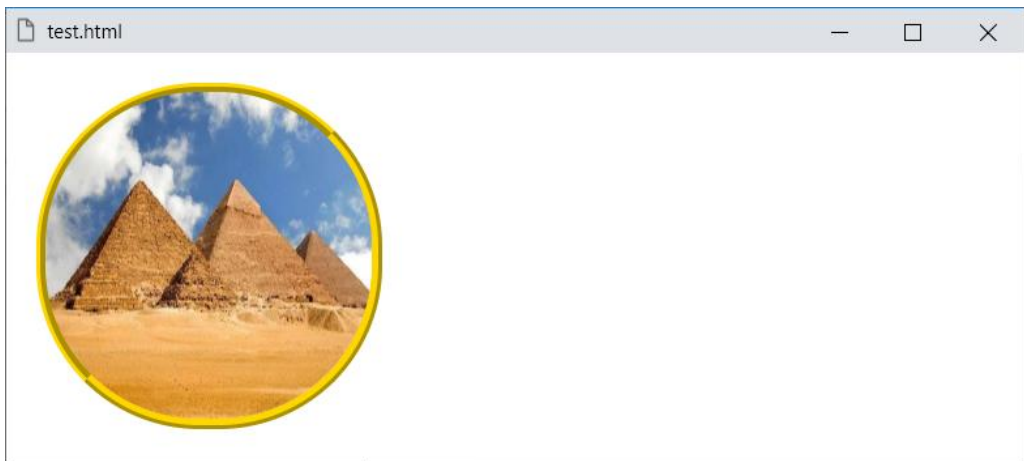


لاحظ تأثير ال border-radius على بروز الصورة.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    body{ padding:10px; }
    img{
      width:200px; height:200px;
      border:6px ridge gold;
      border-radius:100px;
    }
  </style>
</head>
<body>
  
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



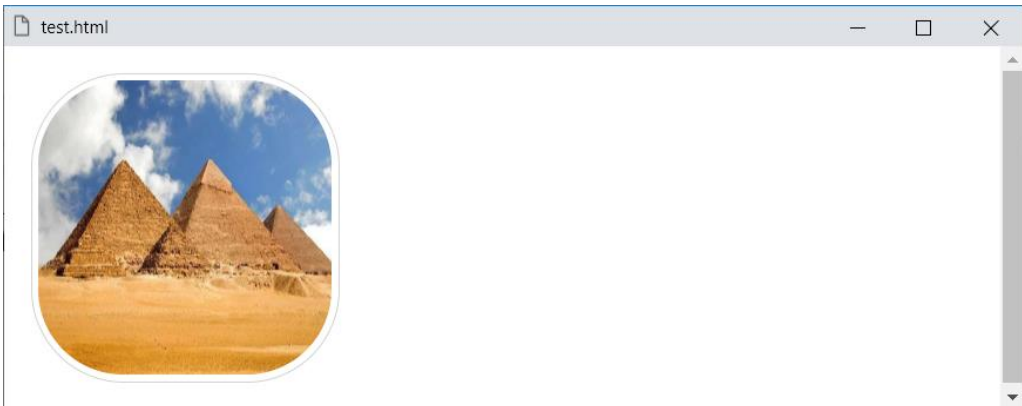
لاحظ تحويل الصورة لدائرة باستخدام الـ .border-radius

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
body{ padding:10px; }
img{
width:200px; height:200px;
padding:5px;
border:0.5px solid lightgray;
border-radius:30%;
}
</style>
</head>
<body>

</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ استخدام إطار خفيف حول الصورة.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    body{ padding:10px; }
    img{
      width:200px; height:200px;
      display:block;
      margin:auto;
    }
  </style>
</head>
<body>
  
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

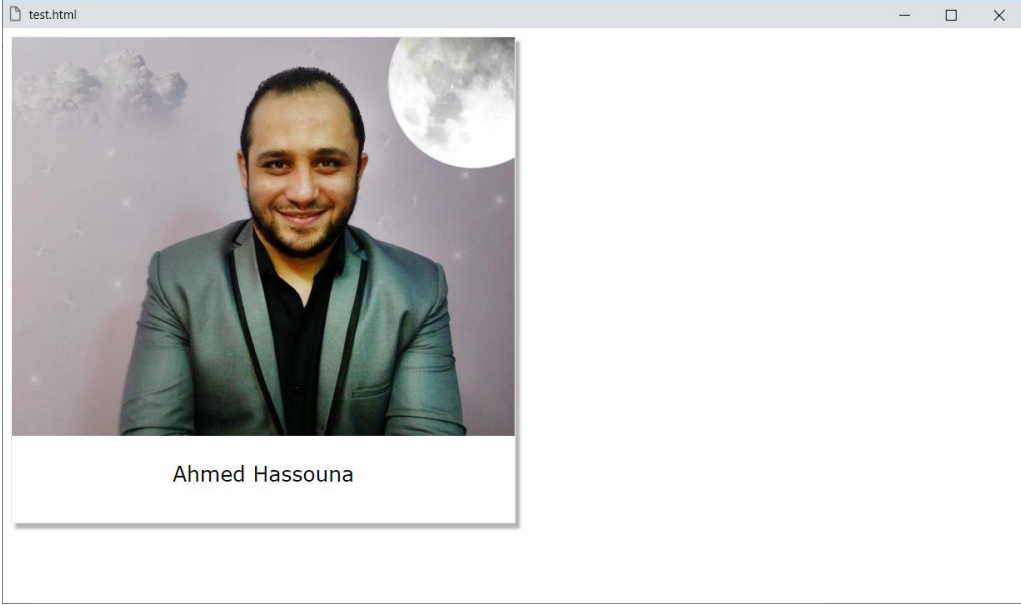


لاحظ أننا جعلنا الصورة في المنتصف، وذلك عن طريق الـ `display:block` والـ `margin:auto`.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .image-card{
      width:50%;
      border:1px solid #eee;
      box-shadow:4px 5px 3px rgba(10,10,10,0.3);
    }
    .image-card img{
      width:100%;
    }
    .image-card p{
      font-family:verdana;
      font-size:20px;
      text-align:center;
      padding-bottom:15px;
    }
  </style>
</head>
<body>
  <div class="image-card">
    
    <p>Ahmed Hassouna</p>
  </div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم عمل كارت صور، باستخدام لون شفاف لظل الصندوق، وإطار خفيف، والوسم p للكتابة title الصورة.

النص مع الصور Text with Images

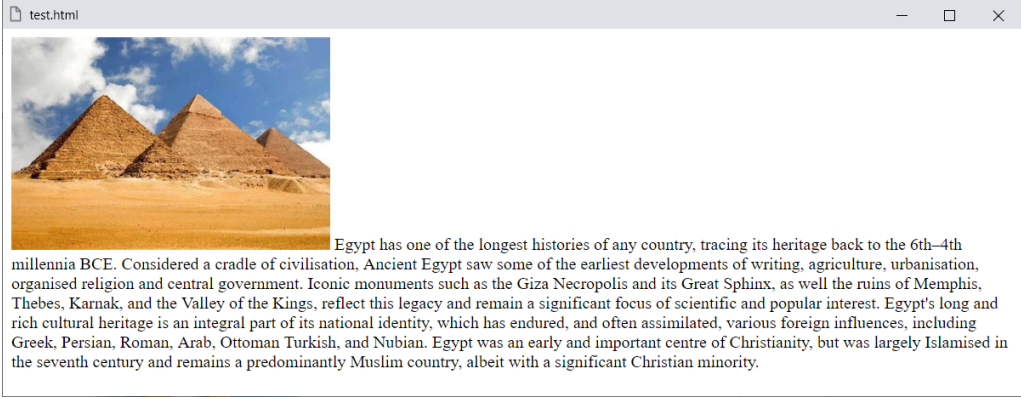
كنا قديما نستخدم الخاصية align للوسم img في لغة html، ولكنها الآن أصبحت قديمة deprecated، وسوف نتعلم تنظيم ودمج النصوص بجوار الصور في لغة CSS، عن طريق الخاصية vertical-align، والمثال التالي يوضح الوضع الطبيعي إذا تم إضافة نص بجوار الصورة:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
img{
width:300px; height:200px;
}
</style>
</head>
<body>

<span>
Egypt has one of the longest histories of any country,
tracing its heritage back to the 6th-4th millennia BCE.
Considered a cradle of civilisation,
Ancient Egypt saw some of the earliest developments of writing,
agriculture, urbanisation, organised religion and central government.
Iconic monuments such as the Giza Necropolis and its Great Sphinx,
as well the ruins of Memphis, Thebes, Karnak, and the Valley of the Kings,
reflect this legacy and remain a significant focus of scientific and popular interest.
Egypt's long and rich cultural heritage is an integral part of its national identity,
which has endured, and often assimilated, various foreign influences,
including Greek, Persian, Roman, Arab, Ottoman Turkish, and Nubian.
Egypt was an early and important centre of Christianity,
but was largely Islamised in the seventh century and remains a predominantly Muslim country,
albeit with a significant Christian minority.
</span>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



وهنا تم اختيار الـ `span` لكتابة النص بداخلة، لينتبه لنا وكأنه نص عادي، ونلاحظ أن النص بدأ من أسفل الصورة، ثم تمت الكلمة بعدها.

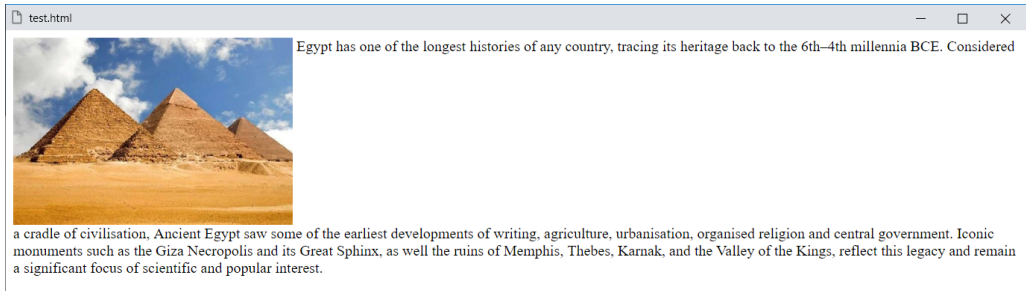
الخاصية `vertical-align` قيمتها الافتراضية هي `baseline`، وتأخذ القيمة `top` أو `text-top` ليبدأ النص من أعلى الصورة، وتأخذ القيمة `middle` ليبدأ النص من الوسط، وتأخذ القيمة `bottom` أو `text-bottom` ليبدأ النص من الأسفل، وتأخذ القيمة، كما يوجد قيم أخرى لهذه الخاصية، حسب الاحتياج.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
img{
width:300px; height:200px;
vertical-align:top;
}
</style>
</head>
<body>

<span>
Egypt has one of the longest histories of any country,
tracing its heritage back to the 6th-4th millennia BCE.
Considered a cradle of civilisation,
Ancient Egypt saw some of the earliest developments of writing,
agriculture, urbanisation, organised religion and central government.
Iconic monuments such as the Giza Necropolis and its Great Sphinx,
as well the ruins of Memphis, Thebes, Karnak, and the Valley of the Kings,
reflect this legacy and remain a significant focus of scientific and popular interest.
</span>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



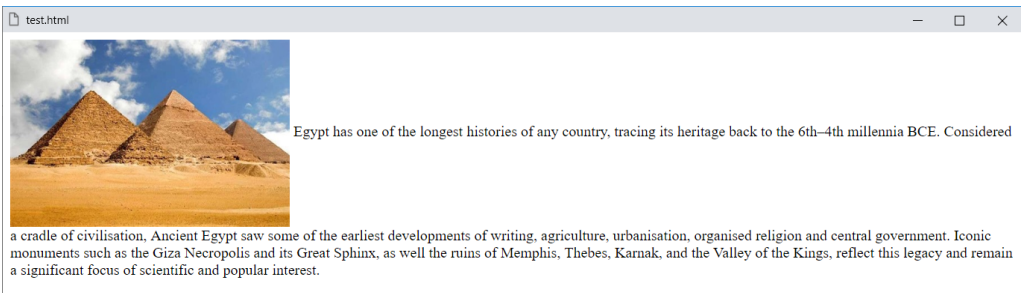
لاحظ أن النص بدأ من أعلى الصورة، ثم تمت الكلمة بعدها.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
img{
width:300px; height:200px;
vertical-align:middle;
}
</style>
</head>
<body>

<span>
Egypt has one of the longest histories of any country,
tracing its heritage back to the 6th-4th millennia BCE.
Considered a cradle of civilisation,
Ancient Egypt saw some of the earliest developments of writing,
agriculture, urbanisation, organised religion and central government.
Iconic monuments such as the Giza Necropolis and its Great Sphinx,
as well the ruins of Memphis, Thebes, Karnak, and the Valley of the Kings,
reflect this legacy and remain a significant focus of scientific and popular interest.
</span>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

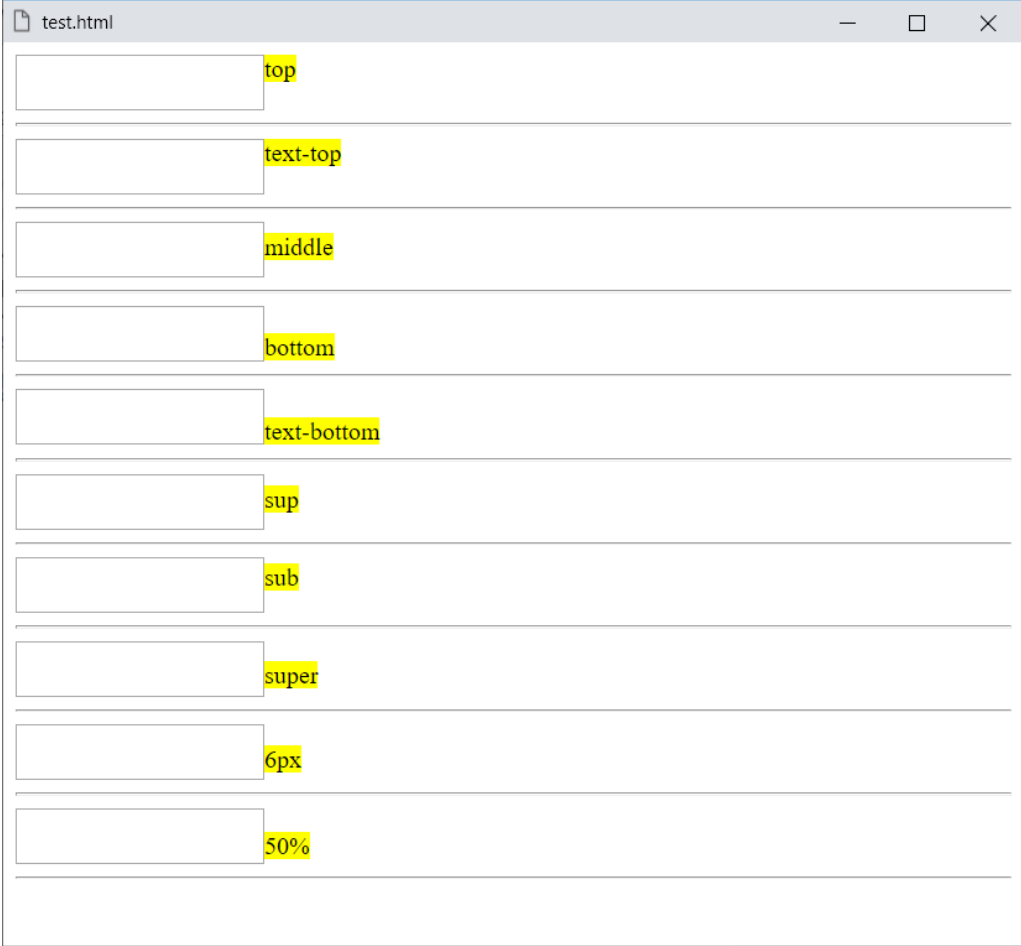


لاحظ أن النص بدأ من أوسط الصورة، ثم تمت الكلمة بعدها.

انتبه! فإه الخاصية vertical-align ليست فقط مع الصور، ولكنه مع باقي العناصر، وسوف نعرض مثال يبيه لنا تأثيرها، كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
span{ background:yellow; } input{ height:30px; }
input.v1{ vertical-align:top; }
input.v2{ vertical-align:text-top; }
input.v3{ vertical-align:middle; }
input.v4{ vertical-align:bottom; }
input.v5{ vertical-align:text-bottom; }
input.v6{ vertical-align:sup; }
input.v7{ vertical-align:sub; }
input.v8{ vertical-align:super; }
input.v9{ vertical-align:6px; }
input.v10{ vertical-align:50%; }
</style>
</head>
<body>
<input class="v1"><span>top</span><hr>
<input class="v2"><span>text-top</span><hr>
<input class="v3"><span>middle</span><hr>
<input class="v4"><span>bottom</span><hr>
<input class="v5"><span>text-bottom</span><hr>
<input class="v6"><span>sup</span><hr>
<input class="v7"><span>sub</span><hr>
<input class="v8"><span>super</span><hr>
<input class="v9"><span>6px</span><hr>
<input class="v10"><span>50%</span><hr>
</body>
</html>
```


وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أننا قمنا باستخدام الـ `input`، وبجواره نص، ولاحظ أن كل قيمة يظهر تأثيرها، ولاحظ أنه يمكن استخدام قيمة مثل `px` أو `%` للتأثير على خاصية `vertical-align`.

التأثيرات filter

يملك التأثير على الصور، حيث يتم عمل تشويش، أو تفتيح، أو تغييرات أخرى على الصور، ولهذا نستخدم الخاصية filter، ثم نعطي لها دوال جاهزة في CSS لتنفيذ ما نريد بفضل الله.

أولاً: التشويش blur

كود! قم بكتابة الكود التالي ثم احفظه:

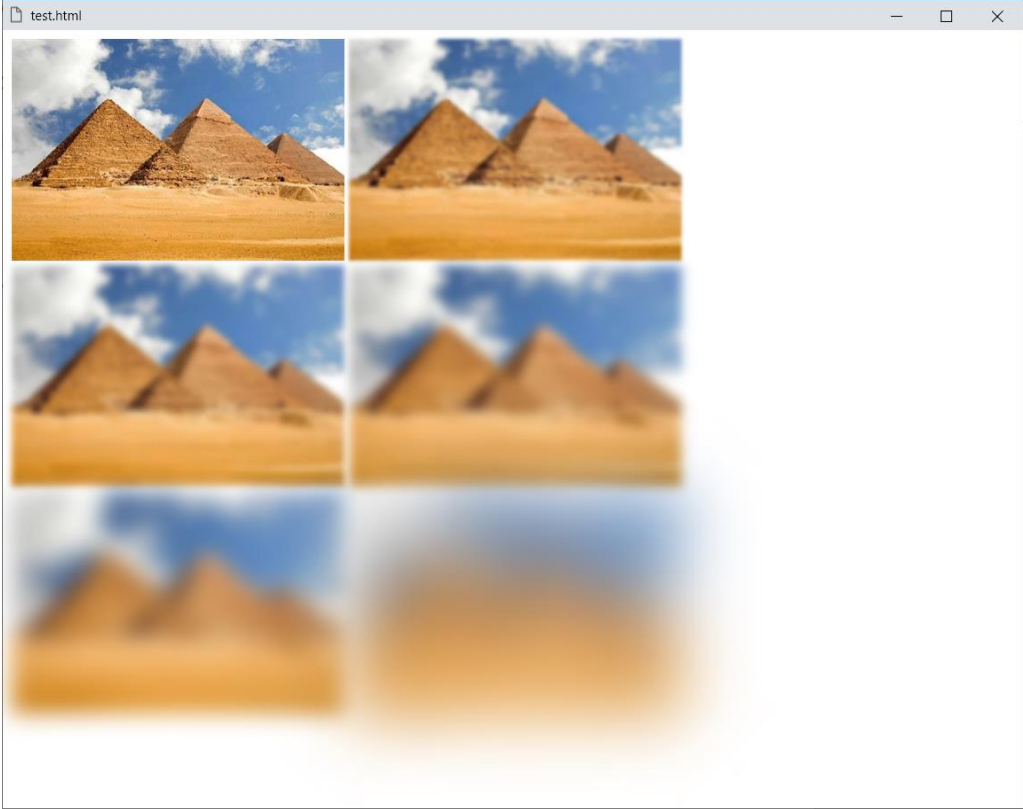
```
<!DOCTYPE html>
<html>
<head>
<style>
img{ width:300px; height:200px; }
img.i1{ filter:blur(1.5px); }
img.i2{ filter:blur(3px); }
img.i3{ filter:blur(5px); }
img.i4{ filter:blur(10px); }
img.i5{ filter:blur(30px); }
</style>
</head>
<body>






</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



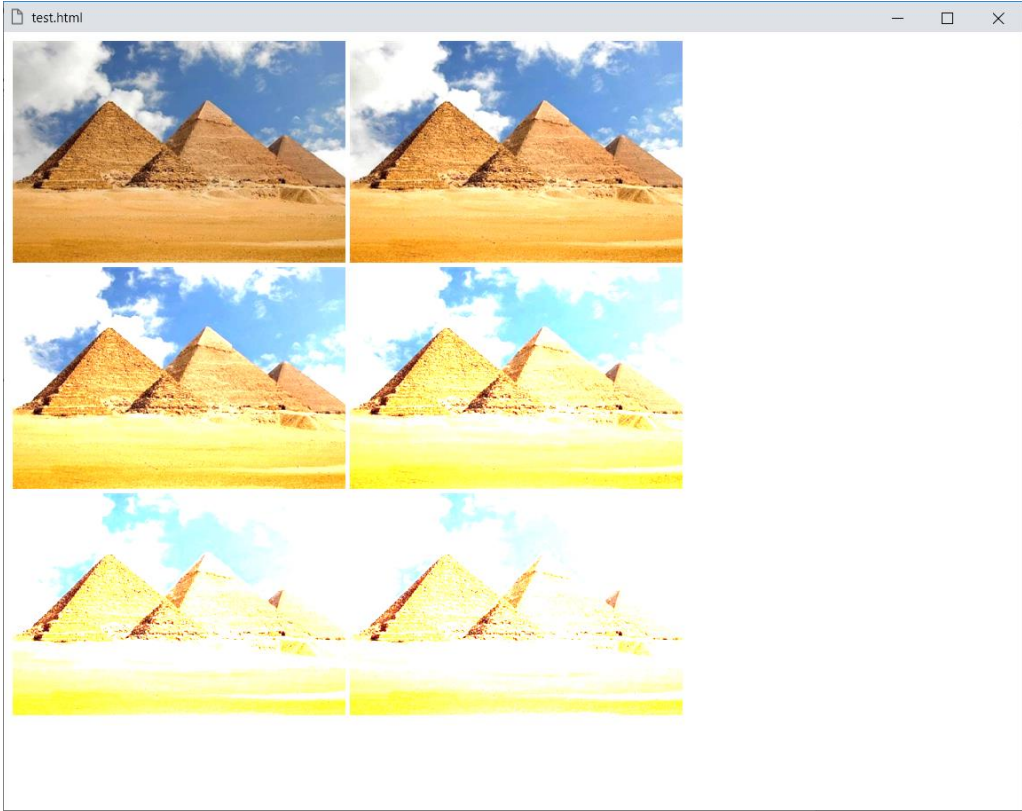
لاحظ أنه تم استخدام الدالة `blur` للتشويش، ولاحظ أنه تم وضع الصورة الأولى بدون تأثير، ثم صورة أخرى تشويشت بمقدار `1.5px`، ثم أخرى تشويشت بمقدار `3px`، ثم أخرى تشويشت بمقدار `5px`، ثم أخرى تشويشت بمقدار `10px`، ثم أخرى تشويشت بمقدار `30px`، وكلما زاد التشويش قلت القدرة على الرؤية.

ثانياً: السطوع brightness

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      img{ width:300px; height:200px; }
      img.i1{ filter:brightness(120%); }
      img.i2{ filter:brightness(150%); }
      img.i3{ filter:brightness(200%); }
      img.i4{ filter:brightness(250%); }
      img.i5{ filter:brightness(300%); }
    </style>
  </head>
  <body>
    
    
    
    
    
    
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

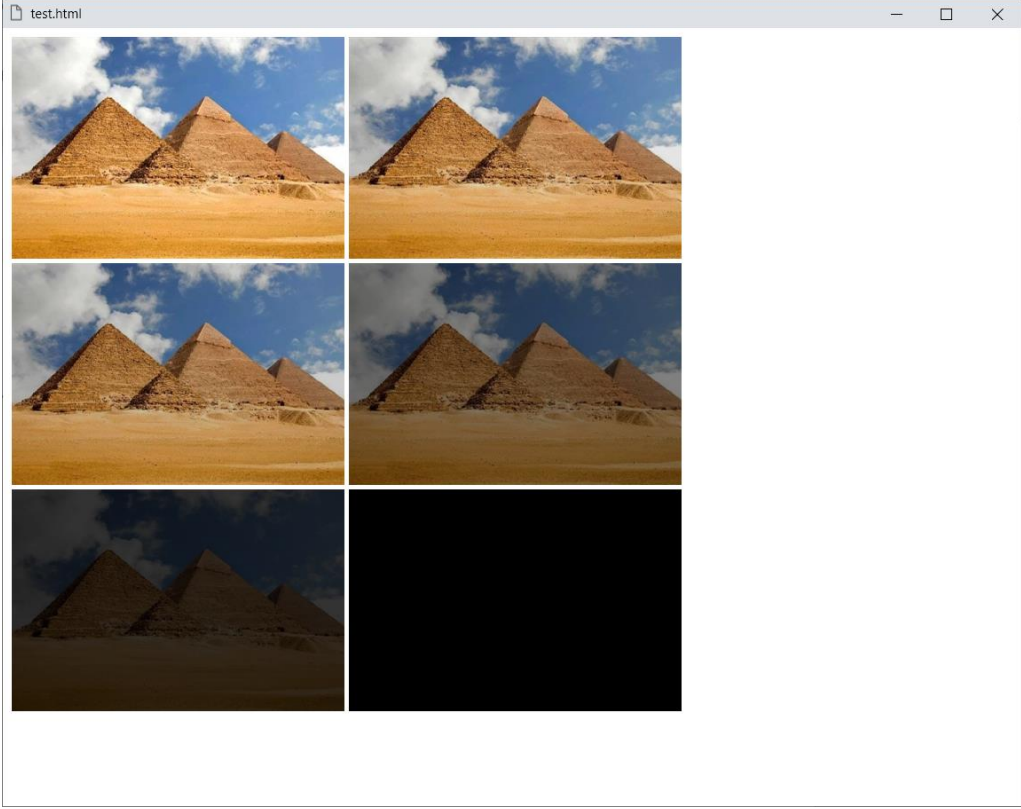


لاحظ أنه تم استخدام الدالة `brightness` للسطوع، ولاحظ أنه تم وضع الصورة الأولى بدون تأثير، ثم صورة أخرى سطعت بمقدار 120%، ثم أخرى سطعت بمقدار 150%، ثم أخرى سطعت بمقدار 200%، ثم أخرى سطعت بمقدار 250%، ثم أخرى سطعت بمقدار 300%، وكلما زاد السطوع زادت إضاءة الصورة.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      img{ width:300px; height:200px; }
      img.i1{ filter:brightness(90%); }
      img.i2{ filter:brightness(80%); }
      img.i3{ filter:brightness(50%); }
      img.i4{ filter:brightness(20%); }
      img.i5{ filter:brightness(0%); }
    </style>
  </head>
  <body>
    
    
    
    
    
    
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



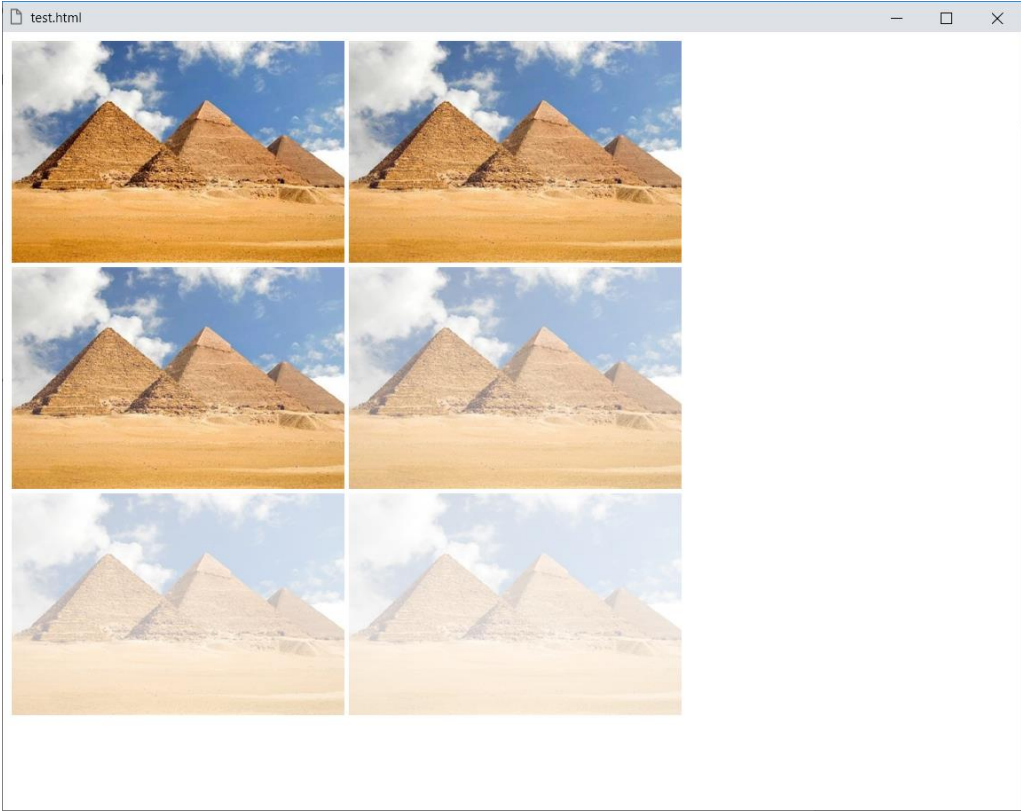
لاحظ أنه عندما قلت نسبة السطوع قلت الإضاءة على الصورة، حتى تصبح سوداء تماماً وغير مرئية.

ثالثاً: الشفافية opacity

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      img{ width:300px; height:200px; }
      img.i1{ filter:opacity(90%); }
      img.i2{ filter:opacity(80%); }
      img.i3{ filter:opacity(50%); }
      img.i4{ filter:opacity(30%); }
      img.i5{ filter:opacity(20%); }
    </style>
  </head>
  <body>
    
    
    
    
    
    
  </body>
</html>
```


وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

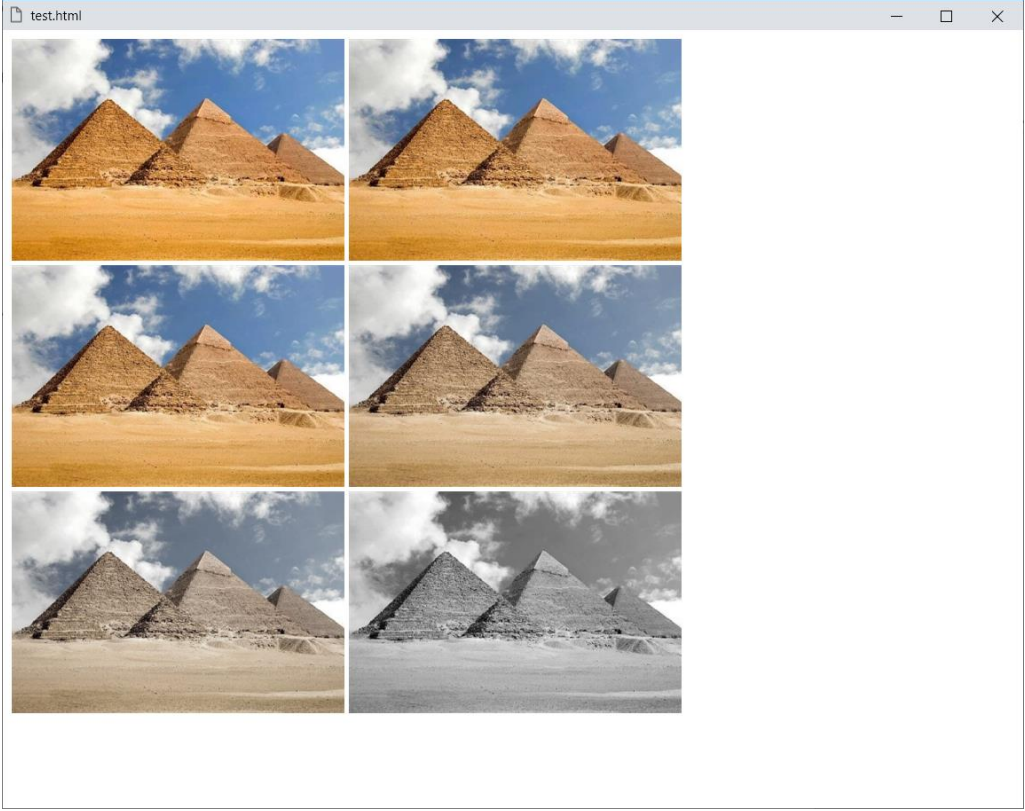


لاحظ أنه تم استخدام الدالة opacity للشفافية، ولاحظ أنه تم وضع الصورة الأولى بدون تأثير، ثم صورة أخرى ظهرت بمقدار 90%، ثم أخرى ظهرت بمقدار 80%، ثم أخرى ظهرت بمقدار 50%، ثم أخرى ظهرت بمقدار 30%، ثم أخرى ظهرت بمقدار 20%، وكلما قلت نسبة الشفافية زاد اختفاء الصورة.

رابعاً: التشبع saturate
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      img{ width:300px; height:200px; }
      img.i1{ filter:saturate(90%); }
      img.i2{ filter:saturate(80%); }
      img.i3{ filter:saturate(50%); }
      img.i4{ filter:saturate(30%); }
      img.i5{ filter:saturate(0%); }
    </style>
  </head>
  <body>
    
    
    
    
    
    
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

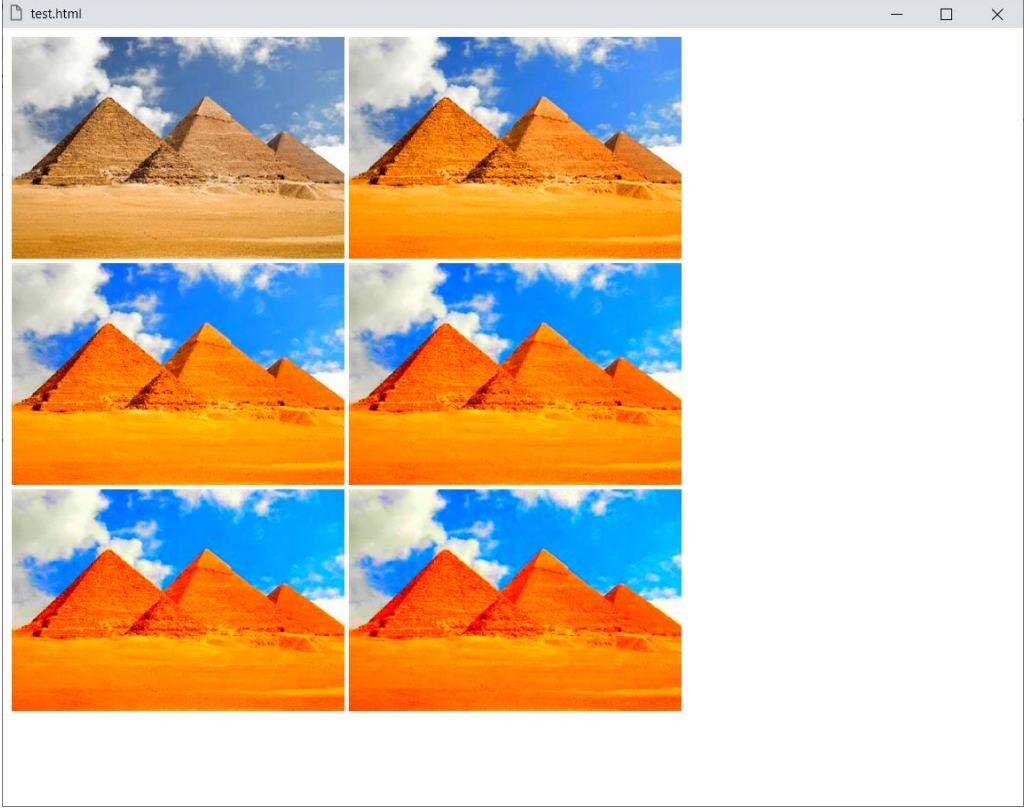


لاحظ أنه تم استخدام الدالة `saturate` لتشبع الألوان، ولاحظ أنه تم وضع الصورة الأولى بدون تأييد، ثم صورة أخرى كان التشبع بمقدار 90%، ثم أخرى كان التشبع بمقدار 80%، ثم أخرى كان التشبع بمقدار 50%، ثم أخرى كان التشبع بمقدار 30%، ثم أخرى كان التشبع بمقدار 0%، وكلما قلت نسبة التشبع زاد اختفاء الألوان عن الصورة.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      img{ width:300px; height:200px; }
      img.i1{ filter:saturate(190%); }
      img.i2{ filter:saturate(290%); }
      img.i3{ filter:saturate(390%); }
      img.i4{ filter:saturate(490%); }
      img.i5{ filter:saturate(600%); }
    </style>
  </head>
  <body>
    
    
    
    
    
    
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



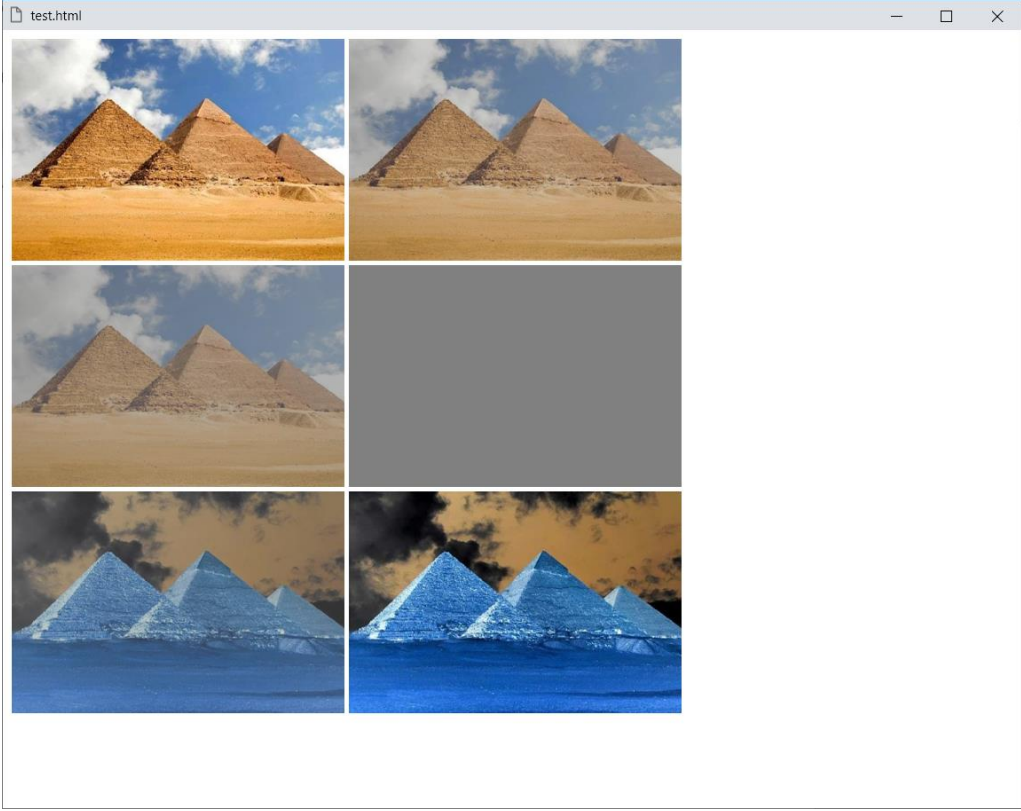
لاحظ أنه مع زيادة نسبة التشبع أكثر من 100% كان تشبع الألوان شديد جداً، بخلاف الصورة الطبيعية.

خامساً: الإعكاس invert

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      img{ width:300px; height:200px; }
      img.i1{ filter:invert(20%); }
      img.i2{ filter:invert(30%); }
      img.i3{ filter:invert(50%); }
      img.i4{ filter:invert(80%); }
      img.i5{ filter:invert(100%); }
    </style>
  </head>
  <body>
    
    
    
    
    
    
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



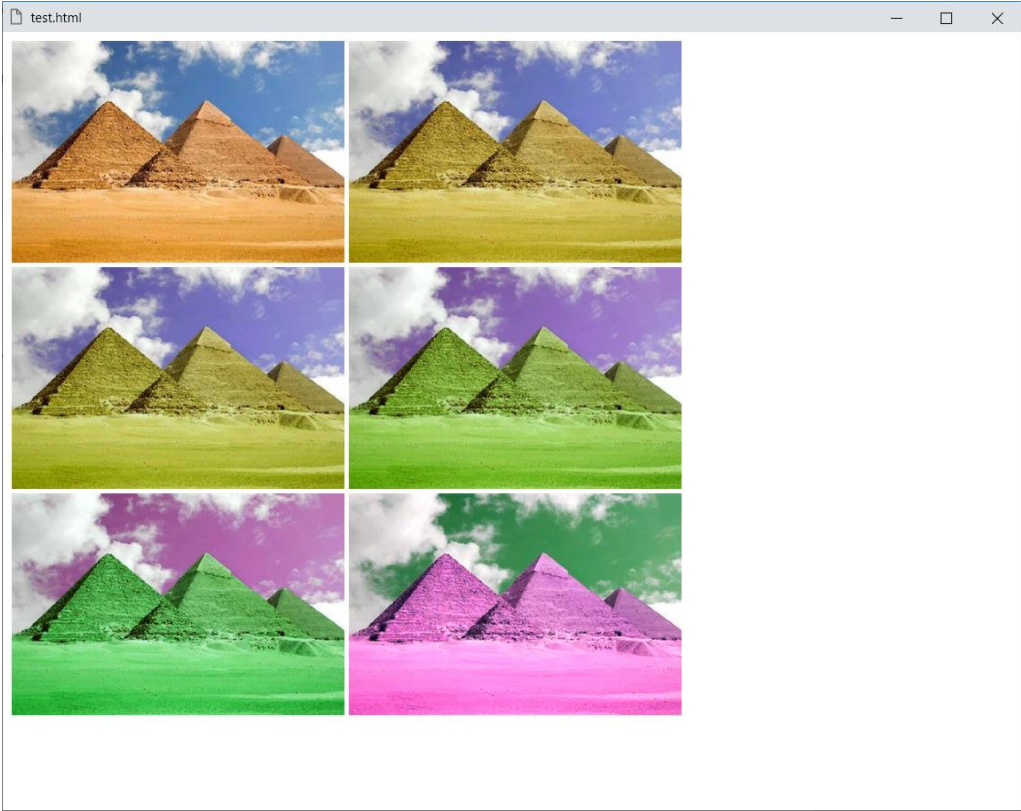
لاحظ أنه تم استخدام الدالة `invert` لعكاس الألوان، ولاحظ أنه تم وضع الصورة الأولى بدون تأخير، ثم صورة أخرى كانت نسبة العكاس بمقدار 20%، ثم أخرى كانت نسبة العكاس بمقدار 30%، ثم أخرى كانت نسبة العكاس بمقدار 50%، ثم أخرى كانت نسبة العكاس بمقدار 80%، ثم أخرى كانت نسبة العكاس بمقدار 100%.

لاحظ أن القيمة 50% هي حلقة الوصل بين العكسيين.

سادساً: تغيرات الألوان hue-rotate كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      img{ width:300px; height:200px; }
      img.i1{ filter:hue-rotate(20deg); }
      img.i2{ filter:hue-rotate(30deg); }
      img.i3{ filter:hue-rotate(50deg); }
      img.i4{ filter:hue-rotate(80deg); }
      img.i5{ filter:hue-rotate(270deg); }
    </style>
  </head>
  <body>
    
    
    
    
    
    
  </body>
</html>
```


وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

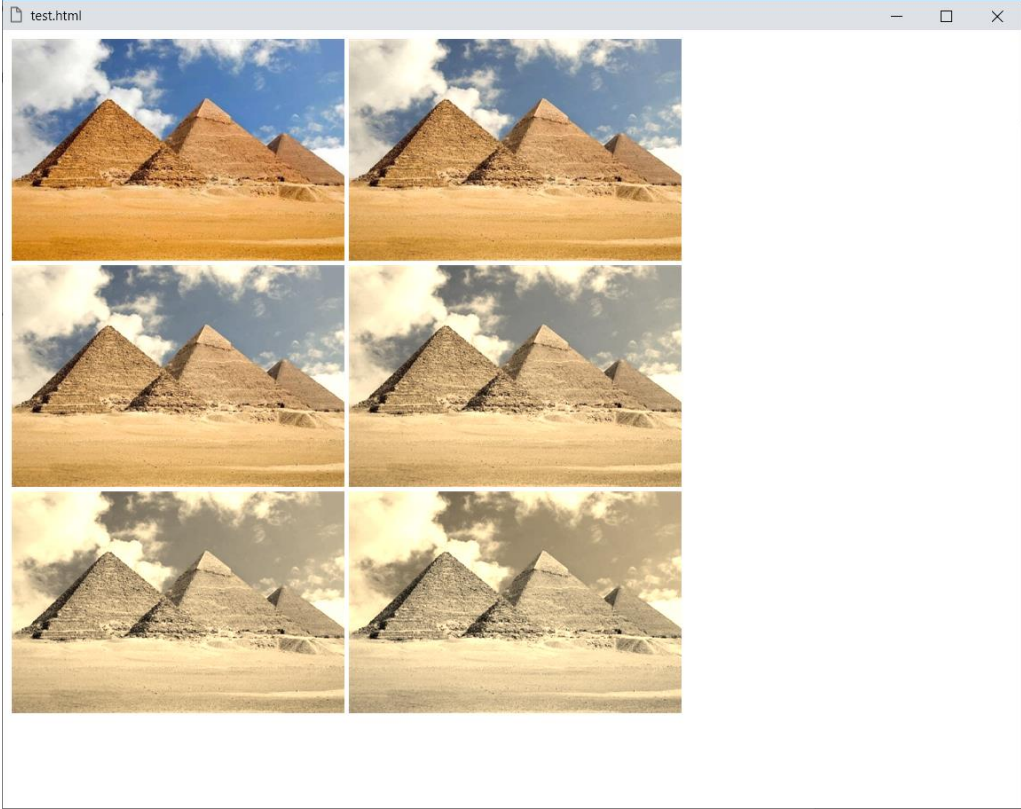


لاحظ أنه تم استخدام الدالة hue-rotate لاستدارة الألوان، ولاحظ أنه تم وضع الصورة الأولى بدون تأييد، ثم صورة أخرى كانت استدارة اللون بمقدار 20 درجة، ثم أخرى كانت استدارة اللون بمقدار 30 درجة، ثم أخرى كانت استدارة اللون بمقدار 50 درجة، ثم أخرى كانت استدارة اللون بمقدار 80 درجة، ثم أخرى كانت استدارة اللون بمقدار 270 درجة، وكلما تغيرت درجة الاستدارة تغيرت الألوان في الصورة.

سابعاً: تأييد السبيدج sepia
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      img{ width:300px; height:200px; }
      img.i1{ filter:sepia(30%); }
      img.i2{ filter:sepia(50%); }
      img.i3{ filter:sepia(70%); }
      img.i4{ filter:sepia(90%); }
      img.i5{ filter:sepia(100%); }
    </style>
  </head>
  <body>
    
    
    
    
    
    
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

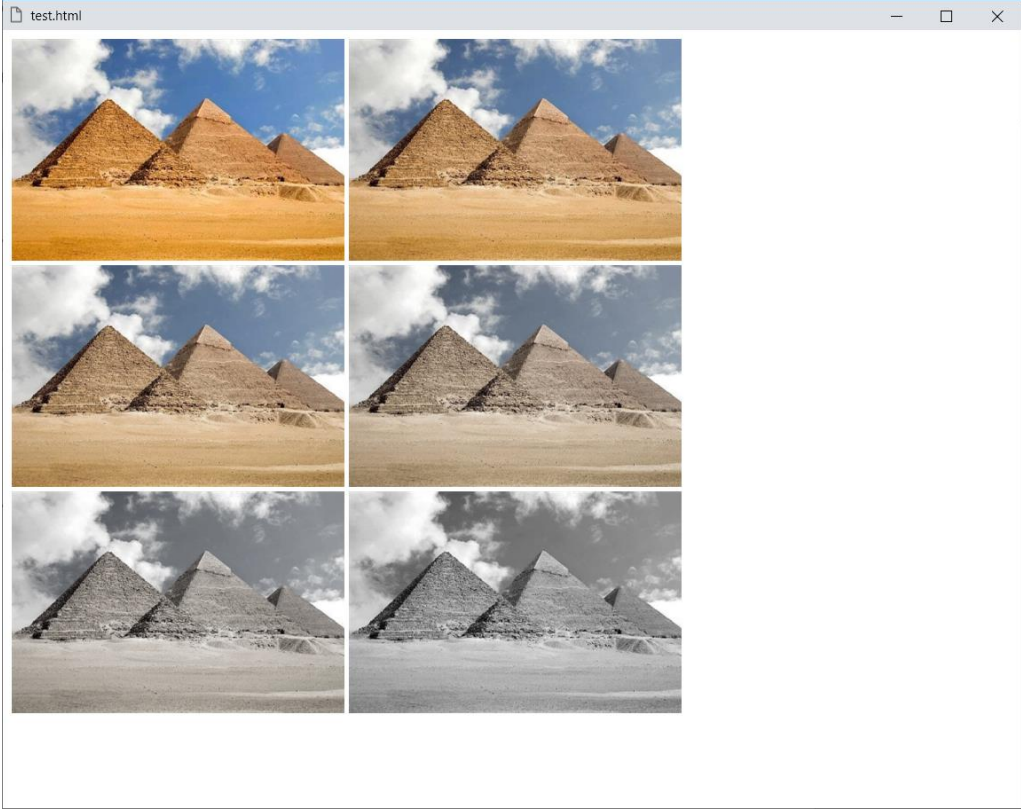


لاحظ أنه تم استخدام الدالة `sepia` لتأثير السبيدج على الألوان، ولاحظ أنه تم وضع الصورة الأولى بدون تأثير، ثم صورة أخرى كانت نسبة السبيدج بمقدار 30%، ثم أخرى كانت نسبة السبيدج بمقدار 50%، ثم أخرى كانت نسبة السبيدج بمقدار 70%، ثم أخرى كانت نسبة السبيدج بمقدار 90%، ثم أخرى كانت نسبة السبيدج بمقدار 100%.

ثامناً : المقياس الرمادي للألوان grayscale كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      img{ width:300px; height:200px; }
      img.i1{ filter:grayscale(30%); }
      img.i2{ filter:grayscale(50%); }
      img.i3{ filter:grayscale(70%); }
      img.i4{ filter:grayscale(90%); }
      img.i5{ filter:grayscale(100%); }
    </style>
  </head>
  <body>
    
    
    
    
    
    
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم استخدام الدالة `opacity` لتأثير المقياس الرمادي على الألوان، ولاحظ أنه تم وضع الصورة الأولى بدون تأثير، ثم صورة أخرى كانت نسبة المقياس الرمادي بمقدار 30%، ثم أخرى كانت نسبة المقياس الرمادي بمقدار 50%، ثم أخرى كانت نسبة المقياس الرمادي بمقدار 70%، ثم أخرى كانت نسبة المقياس الرمادي بمقدار 90%، ثم أخرى كانت نسبة المقياس الرمادي بمقدار 100%، حتى وصلت كصورة قديمة أبيض وأسود.

تاسعاً: التباين contrast
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
img{ width:200px; height:100px; }
img.i1{ filter:contrast(30%); }
img.i2{ filter:contrast(50%); }
img.i3{ filter:contrast(70%); }
img.i4{ filter:contrast(90%); }
img.i5{ filter:contrast(120%); }
img.i6{ filter:contrast(220%); }
img.i7{ filter:contrast(320%); }
</style>
</head>
<body>

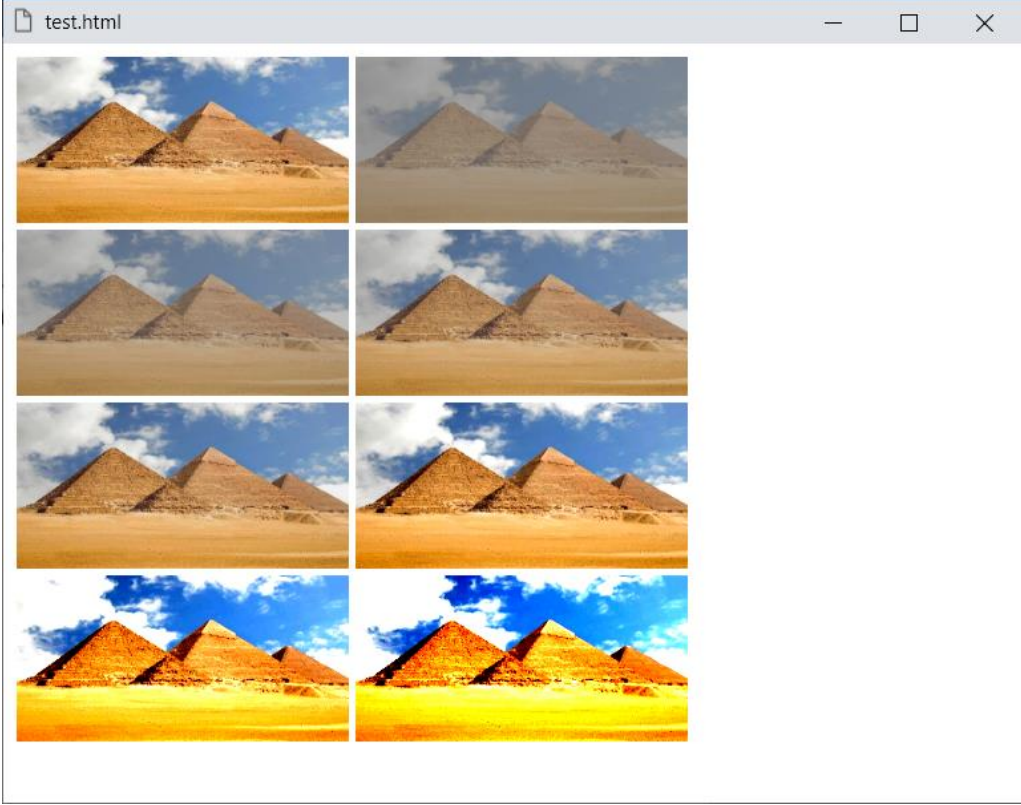







</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

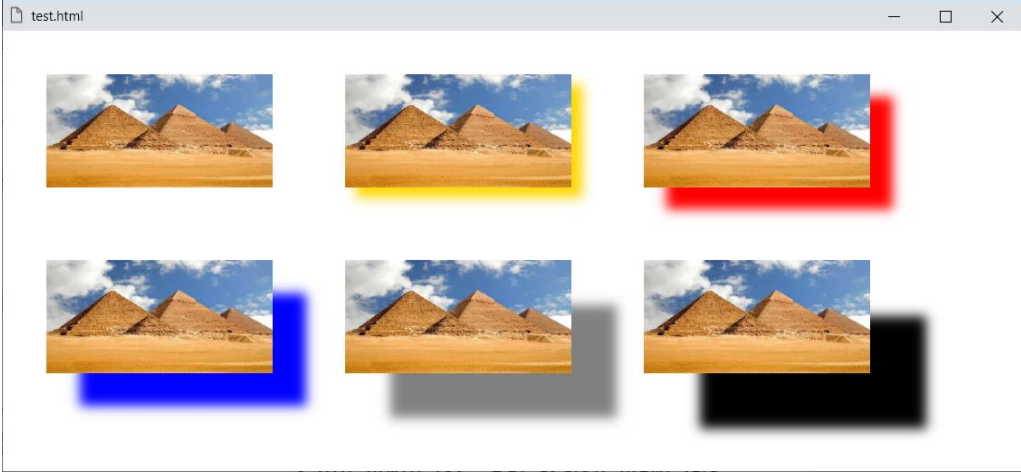


لاحظ أنه تم استخدام الدالة `contrast` لتأثير التباين على الألوان، ولاحظ أنه تم وضع الصورة الأولى بدون تأثير، ثم صورة أخرى كانت نسبة التباين بمقدار 30%، ثم أخرى كانت نسبة التباين بمقدار 50%، ثم أخرى كانت نسبة التباين بمقدار 70%، ثم أخرى كانت نسبة التباين بمقدار 90%، ثم أخرى كانت نسبة التباين بمقدار 120%، ثم أخرى كانت نسبة التباين بمقدار 220%، ثم أخرى كانت نسبة التباين بمقدار 320%.

وأخيراً: تأييد اسقاط الظل بالدالة drop-shadow كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      img{ width:200px; height:100px; margin:30px; }
      img.i1{ filter:drop-shadow(10px 10px 5px gold); }
      img.i2{ filter:drop-shadow(20px 20px 5px red); }
      img.i3{ filter:drop-shadow(30px 30px 5px blue); }
      img.i4{ filter:drop-shadow(40px 40px 5px gray); }
      img.i5{ filter:drop-shadow(50px 50px 5px black); }
    </style>
  </head>
  <body>
    
    
    
    
    
    
  </body>
</html>
```


وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



انتبه! فلقد تم شرح الظل من قبل بالتفصيل بفضل الله، والآن لاحظ وجود ظلال مختلفة على الصور، فهناك الذهبي gold، والأحمر red، والأزرق blue، والرمادي gray، والأسود black، بمسافات مختلفة.

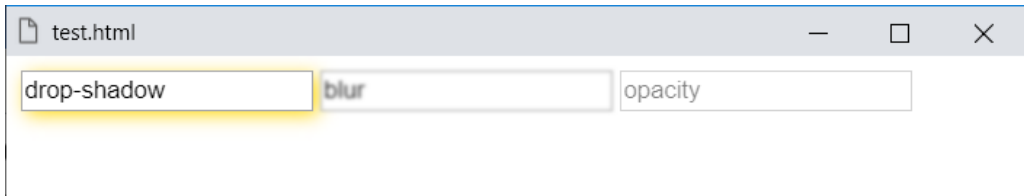
يوجد أيضاً دالة url، وهي تأخذ منك ملف xml به محتوى svg ليؤثر على الخاصية filter، ولكه **انتبه!** فإن موضوع svg، له دراسة خاصة، وإن شاء الله تستطيع ان تبحث عن كتاب "صناعة الأساس" في موضوع الـ SVG.

انتبه! جيداً، فإن الخاصية filter لا تخص الصور فقط، بل يمكن تطبيقها على عناصر متعددة، وسنجرّب فيما يلي إن شاء الله.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    #i1{ filter:drop-shadow(3px 3px 5px gold); }
    #i2{ filter:blur(1px); }
    #i3{ filter:opacity(50%); }
  </style>
</head>
<body>
  <input id="i1" value="drop-shadow">
  <input id="i2" value="blur">
  <input id="i3" value="opacity">
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم التأيد بالخاصية filter على الـ input.

object-fit مناسب

يملك التحكم بالتجسيم مع ملاحظة الفرق، وسنجد بالأكود! التالي:

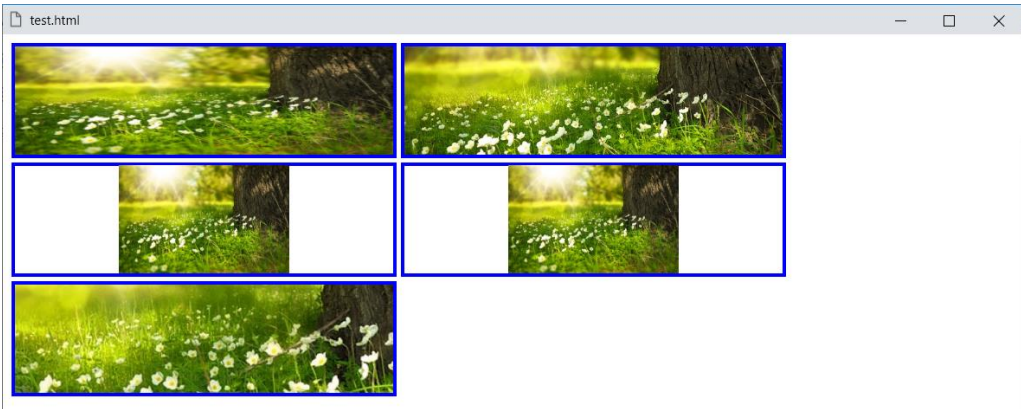
```
<!DOCTYPE html>
<html>
<head>
<style>
img{ border:3px solid blue; }
#i1{ object-fit: fill; }
#i2{ object-fit: cover; }
#i3{ object-fit: contain; }
#i4{ object-fit: scale-down; }
#i5{ object-fit: none; }
</style>
</head>
<body>





</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



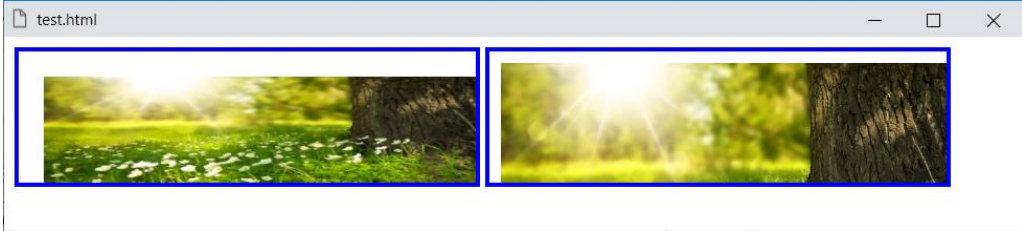
موضحة مع التوجيه object-position

يمكّن ضبط الموضحة بالتوجيه مع ملاحظة الفرق، وسنجرّب بالكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
img{ border:3px solid blue; }
#i1{ object-fit: fill; object-position: 19px 19px; }
#i2{ object-fit: cover; object-position: 9px 9px; }
</style>
</head>
<body>


</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم وضع فرق من اليسار ومن أعلى.

الفئات والعناصر المستعارة pseudo

الفئات المستعارة هي pseudo-class، والعناصر المستعارة هي pseudo-element، ويتم استخدام مثل هذه الأشياء لتحديد حالات خاصة لعنصر محدد، مثل لمس العنصر، أو العنصر الأول في سلسلة محددة، أو تحديد العنصر، أو تحديد النص، وأشياء أخرى كثيرة، وعند التعامل مع الفئات المستعارة نضع كولو ":", والعناصر المستعارة نضع " :::", بعد اسم الـ class أثناء تنسيق CSS، ثم اسم الفئة المستعارة أو العنصر المستعار، ولفهم هذه الفئات والعناصر المستعارة، سوف نصنع أمثلة ونلاحظ التأثيرات إن شاء الله.

وللتسهيل تخيل وكأننا نريد عمل تنسيق معين أثناء لمس الزر بالماوس، أي عند مرور المؤشر على الزر، فيكون الكود مثل التالي:

```
button:hover
```

ثم بعد ذلك أقواس المجموعة {} ثم أكواد اللغة CSS، ويتم تطبيق التنسيق الموجود داخل الأقواس بعد الـ hover فقط عند مرور المؤشر، ولو تم إبعاد المؤشر مرة أخرى، يعود التنسيق الأصلي كما كان، وهكذا، يتم لمس الزر يظهر تنسيق hover، وعدم اللمس يزول تنسيق hover. أو تخيل لو نريد تنسيق عند تحديد نص الـ input والكود التالي:

```
input::selection
```

ليتأثر النص، أثناء تحديده فقط، وهكذا... وسوف نفضل إن شاء الله.

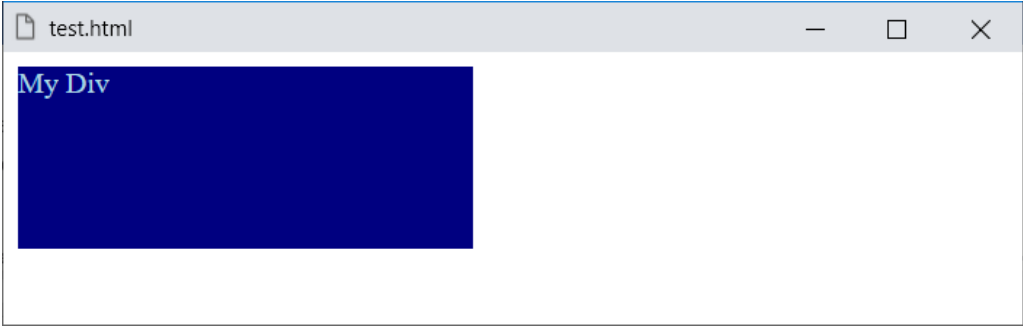
اللمس hover

هو يخص لمس العناصر، لعمل تنسيق معين عليها أثناء اللمس فقط.

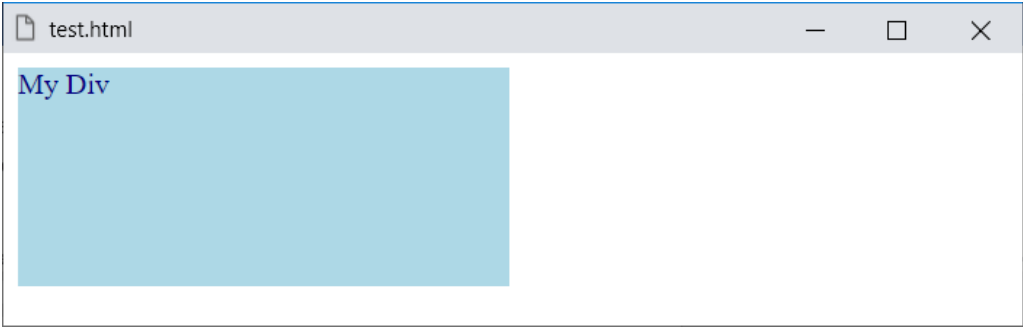
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      div{
        width:250px; height:100px;
        background:navy; color:lightblue;
      }
      div:hover{
        width:270px; height:120px;
        background:lightblue; color:navy;
      }
    </style>
  </head>
  <body>
    <div>My Div</div>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



وعند اللمس يكون الشكل كما يلي:



لاحظ أن الوسم div تم تنسيقه على عرض 250 وطول 100 ولون خلفية ازرق غامق ولون نص ازرق فاتح.

لاحظ عند لمس الوسم div يتحول تنسيقه الى عرض 270 وطول 120 ولون خلفية ازرق فاتح ولون نص ازرق غامق.

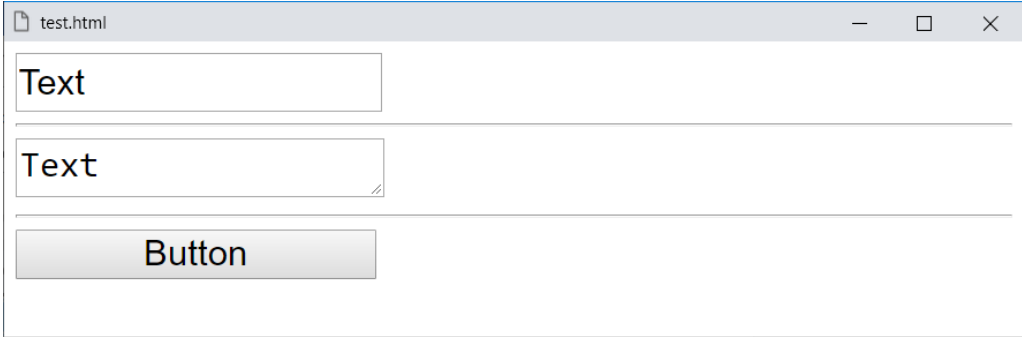
التركيز focus

يتم التنسيق عند دخول المؤشر وثباته في عنصر مثل الـ `input`.

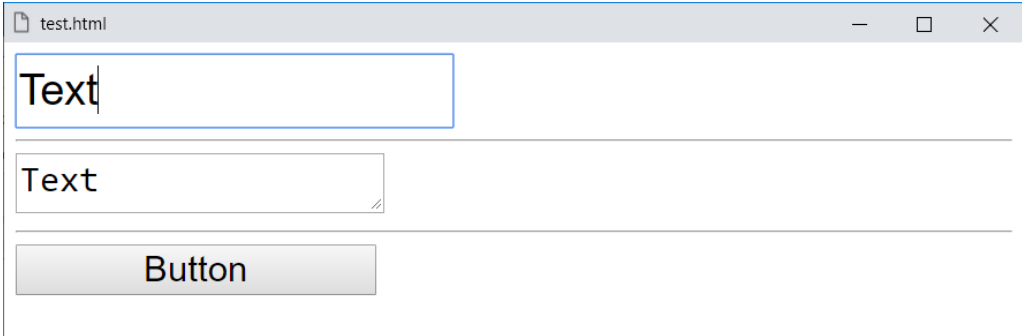
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
input, textarea, button{
width:250px; height:35px;
font-size:25px;
}
input:focus, textarea:focus, button:focus{
width:300px; height:45px;
font-size:30px;
}
</style>
</head>
<body>
<input value="Text"><hr>
<textarea>Text</textarea><hr>
<button>Button</button>
</body>
</html>
```

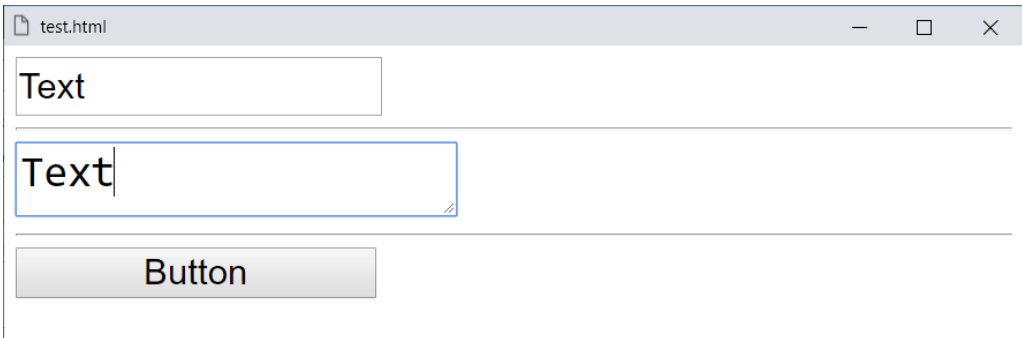

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



والآن انتبه! واضغط الزر tab لتدري مثل الشكل التالي:



والآن انتبه! واضغط الزر tab لتدري مثل الشكل التالي:



ولو ضغطت tab اخذني سترني التأثير على الزر أيضاً.

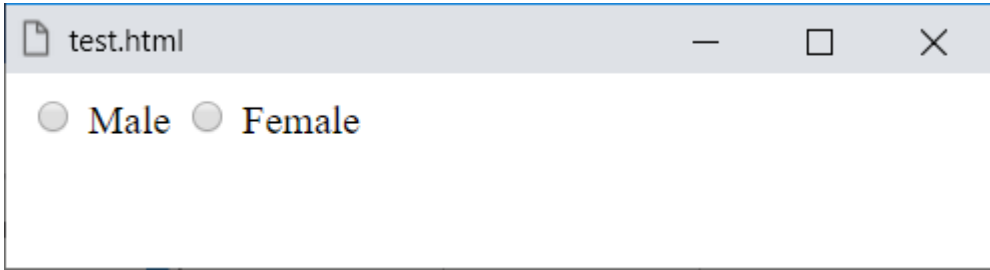
الاختيار checked

يوجد رسوم بها أنواع يتم تحديدها أو اختيارها، مثل الـ radio.

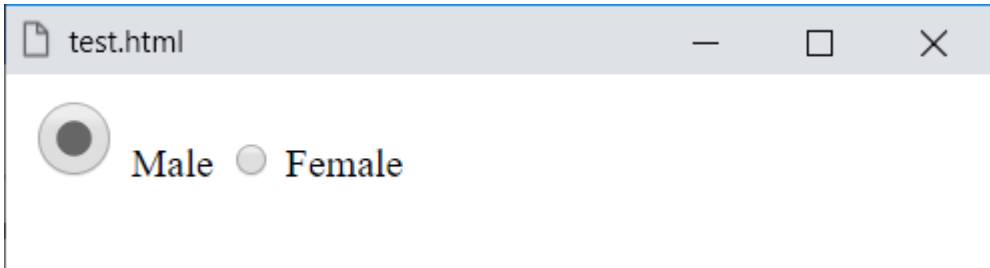
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      input[type=radio]:checked{
        width:30px; height:30px;
        font-size:25px;
      }
    </style>
  </head>
  <body>
    <input type="radio" name="g" id="m">
    <label for="m">Male</label>
    <input type="radio" name="g" id="f">
    <label for="f">Female</label>
  </body>
</html>
```

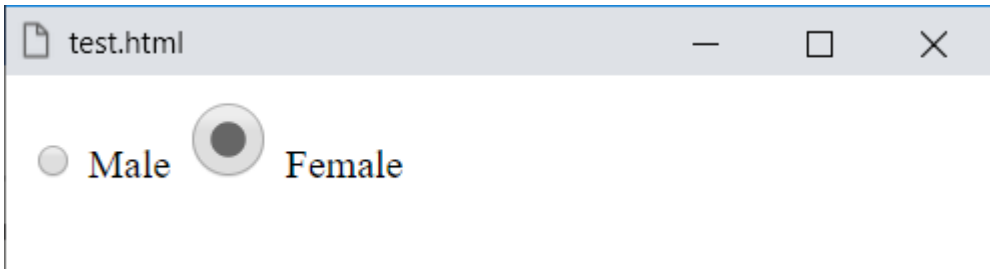
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



والآن انتبه! وحدد male لترى مثل الشكل التالي:



والآن انتبه! وحدد female لترى مثل الشكل التالي:



ولو قمت بعمل نفس التأثيرات على أشياء بها اختيار أخري مثل النوع checkbox سوف يتأثر أيضاً.

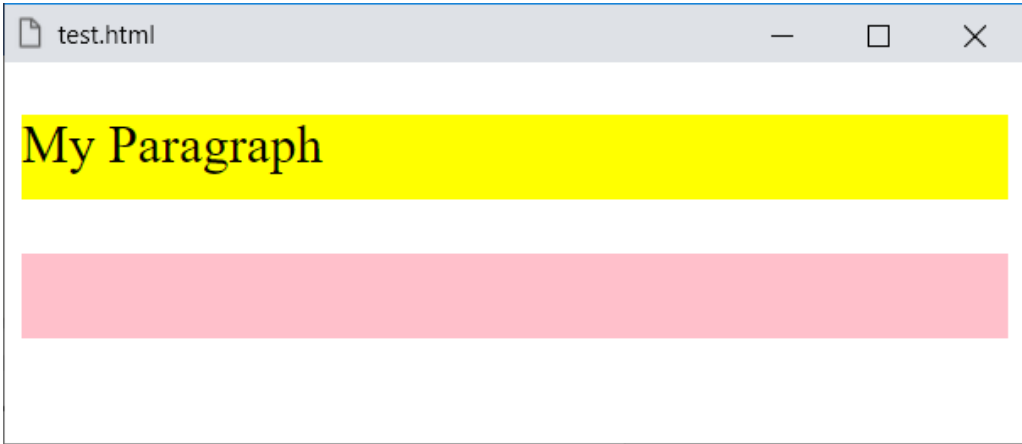
محتوي فارغ empty

عمل تنسيق معيه عندما يكون محتوى العنصر فارغ.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      p{
        font-size:25px;
        height:40px;
        background:yellow;
      }
      p:empty{
        background:pink;
      }
    </style>
  </head>
  <body>
    <p>My Paragraph</p>
    <p></p>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أننا قمنا بتمييز اللون الـ pink للفقرات الفارغة، فإذا كان هناك أي فقرات بالوسم p فارغة ولا يوجد بها محتوى، فسوف تتحول إلى التسيق، والذي هنا هو لون خلفية pink.

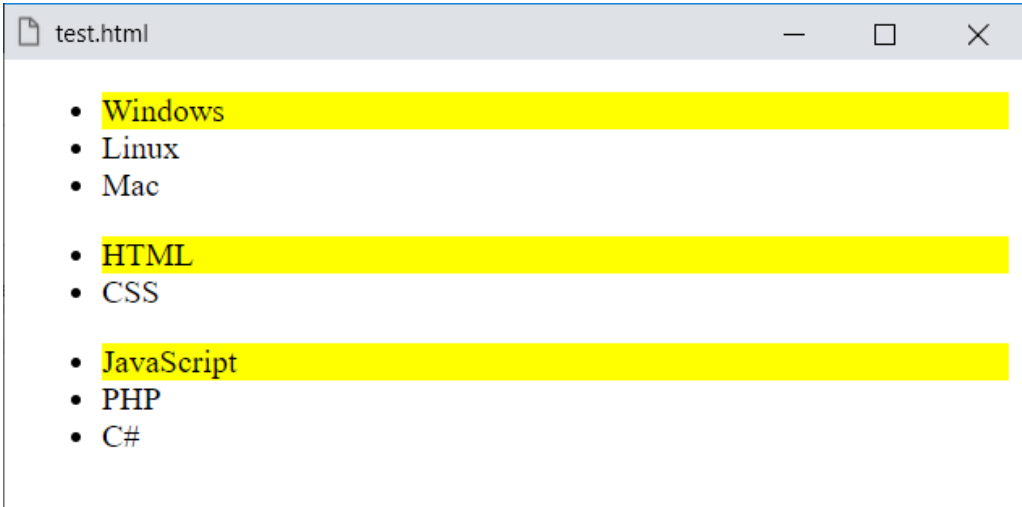
أول عنصر فرعي first-child

عند وجود وسوم بها وسوم أخرى متفرعة منها، يتم تطبيق التنسيق على أول عنصر من كل مجموعة، وسنوضح فيما يلي ان شاء الله.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      li:first-child{
        background:yellow;
      }
    </style>
  </head>
  <body>
    <ul>
      <li>Windows</li>
      <li>Linux</li>
      <li>Mac</li>
    </ul>
    <ul>
      <li>HTML</li>
      <li>CSS</li>
    </ul>
    <ul>
      <li>JavaScript</li>
      <li>PHP</li>
      <li>C#</li>
    </ul>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

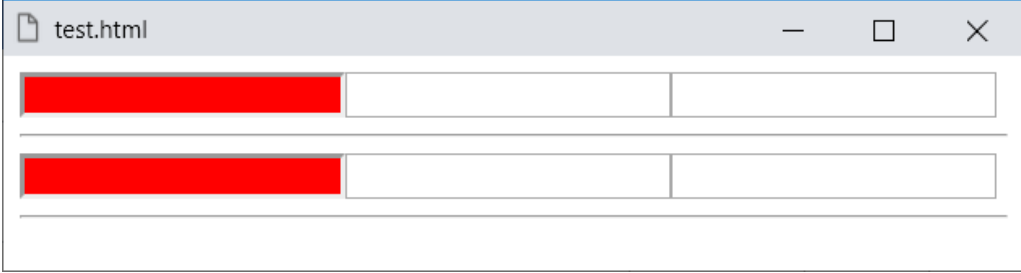


لاحظ أنه تم عمل ثلاثة وسوم UI، وكل منها يحمل بعض العناصر، **وانتبه!** ولاحظ أن التنسيق تم تطبيقه على أول عنصر من كل مجموعة من الوسوم UI.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    input:first-child{ background:red; }
  </style>
</head>
<body>
  <div><input><input><input></div><hr>
  <div><input><input><input></div><hr>
</body>
</html>
```

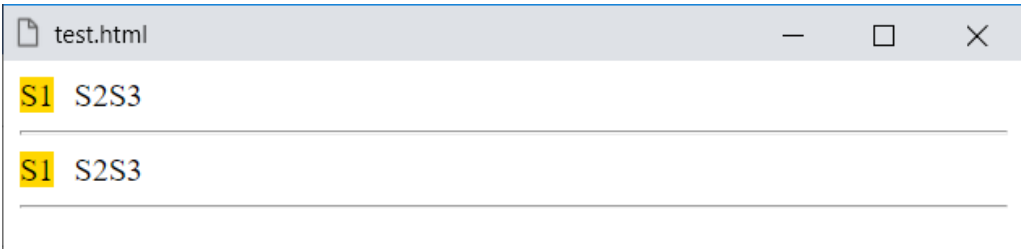
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم عمل وسميه div، وبداخل كل وسم منهم ثلاثة وسموم input، وتم تحديد لون خلفية أحمر red، لأول كل وسم input، فتم تطبيق التنسيق على أول input في كل مجموعة، ويمكن تطبيق نفس الفكرة، بتنسيق آخر على الومسم span، كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
span:first-child{ background:gold; margin-right:10px; }
</style>
</head>
<body>
<div><span>S1</span><span>S2</span><span>S3</span></div><hr>
<div><span>S1</span><span>S2</span><span>S3</span></div><hr>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



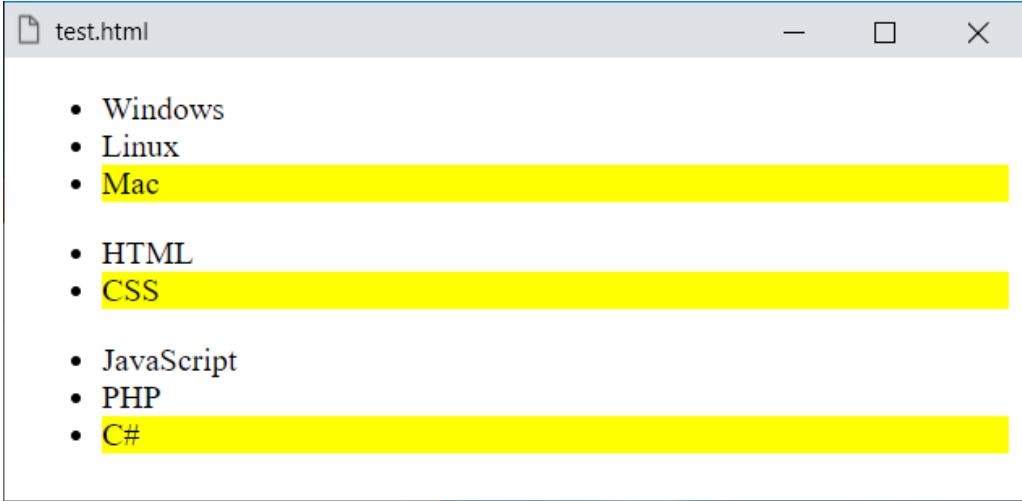
آخر عنصر فرعي last-child

عند وجود وسوم بها وسوم أخري متفرعة منها، يتم تطبيق التنسيق على آخر عنصر من كل مجموعة، وسنوضح فيما يلي ان شاء الله.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      li:last-child{
        background:yellow;
      }
    </style>
  </head>
  <body>
    <ul>
      <li>Windows</li>
      <li>Linux</li>
      <li>Mac</li>
    </ul>
    <ul>
      <li>HTML</li>
      <li>CSS</li>
    </ul>
    <ul>
      <li>JavaScript</li>
      <li>PHP</li>
      <li>C#</li>
    </ul>
  </body>
</html>
```

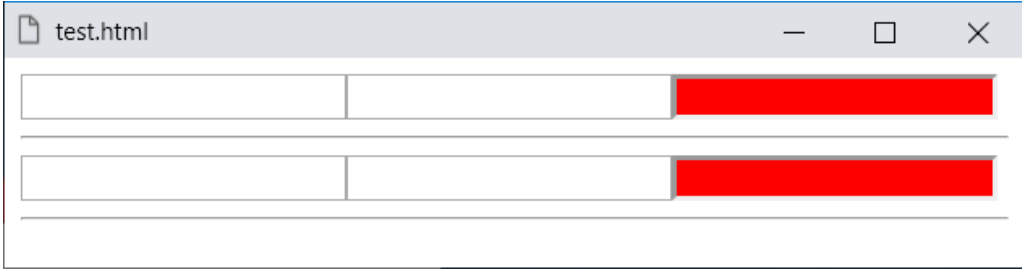
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم عمل ثلاثة وسوم UI، وكل منها يحمل بعض العناصر،
وانتبه! ولاحظ أن التنسيق تم تطبيقه على آخر عنصر من كل مجموعة
من الوسوم UI، وفيما يلي، تطبيق آخر:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
input:last-child{ background:red; }
</style>
</head>
<body>
<div><input><input><input></div><hr>
<div><input><input><input></div><hr>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

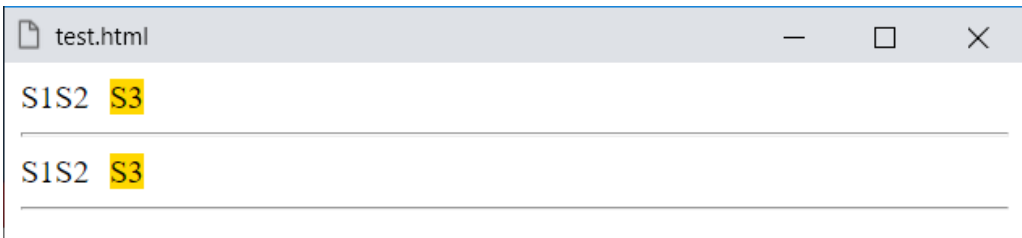


لاحظ أنه تم عمل وسميه div، وبداخل كل وسم منهم ثلاثة وسموم input، وتم تحديد لون خلفية أحمر red، لآخر كل وسم input، فتم تطبيق التنسيق على آخر input في كل مجموعة، ويمكن تطبيق نفس الفكرة، بتنسيق آخر على الـ span، كما يلي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
span:last-child{ background:gold; margin-left:10px; }
</style>
</head>
<body>
<div><span>S1</span><span>S2</span><span>S3</span></div><hr>
<div><span>S1</span><span>S2</span><span>S3</span></div><hr>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



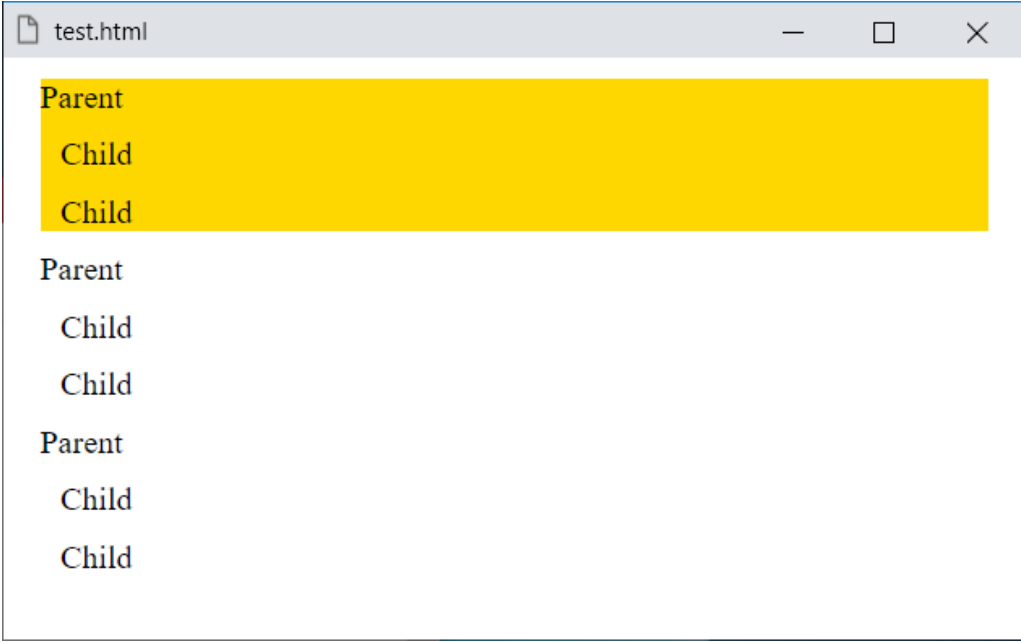
النوع الأول first-of-type

هو يشبه كثيراً ال first-child، ولكنه يركز على النوع.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{ margin:10px; }
  .parent:first-child{
    background:gold;
  }
</style>
</head>
<body>
<div class="parent">Parent
  <div>Child</div>
  <div>Child</div>
</div>
<div class="parent">Parent
  <div>Child</div>
  <div>Child</div>
</div>
<div class="parent">Parent
  <div>Child</div>
  <div>Child</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

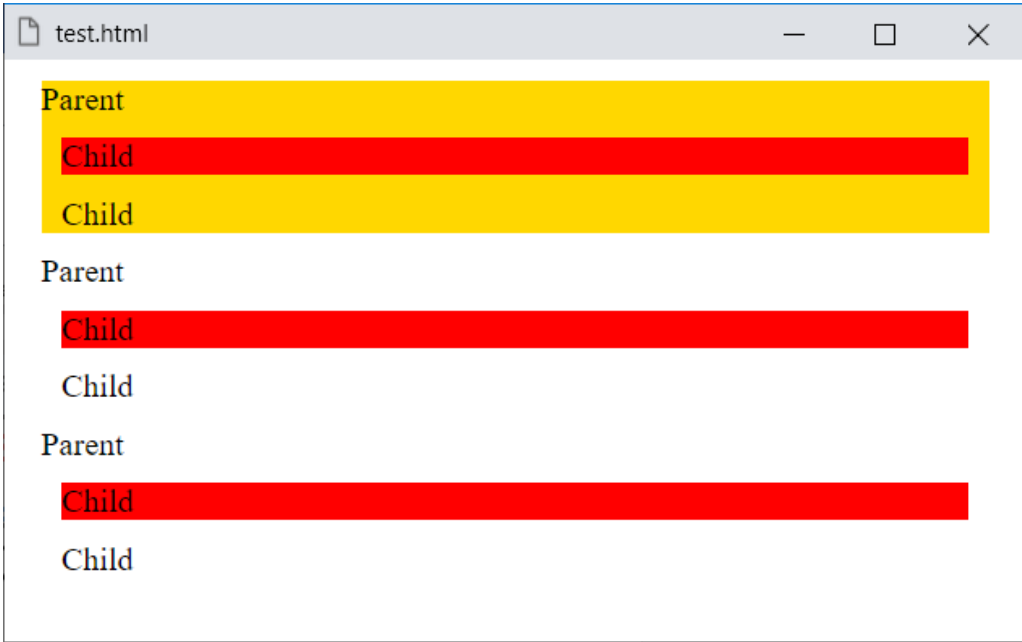


هذا المثال طبيعي، كما تعلمنا من قبل، ولكنه ماذا لو أردنا ثبات هذا التمييز، مع تمييز آخر للفروع!، وهنا نستطيع أن نقوم بحل هذه المشكلة، من خلال الفئة المستعارة `first-of-type`، كما سيوضح لنا إن شاء الله المثال التالي.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{ margin:10px; }
  div:first-of-type{
    background:red;
  }
  .parent:first-child{
    background:gold;
  }
</style>
</head>
<body>
<div class="parent">Parent
  <div>Child</div>
  <div>Child</div>
</div>
<div class="parent">Parent
  <div>Child</div>
  <div>Child</div>
</div>
<div class="parent">Parent
  <div>Child</div>
  <div>Child</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



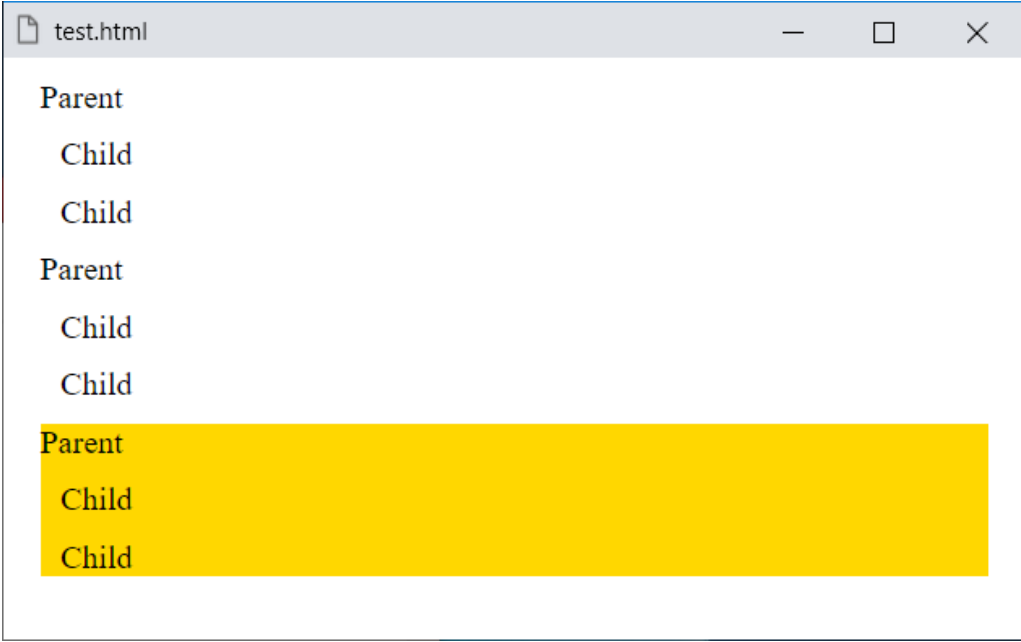
ونلاحظ هنا أننا فرقنا بين الأصل parent والفرع child بسهولة
التعامل مع الفئات المستعارة first-child, first-of-type.

النوع الأخير last-of-type هو يشبه كثيراً الـ last-child، ولكنه يركز على النوع.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{ margin:10px; }
  .parent:last-child{
    background:gold;
  }
</style>
</head>
<body>
<div class="parent">Parent
  <div>Child</div>
  <div>Child</div>
</div>
<div class="parent">Parent
  <div>Child</div>
  <div>Child</div>
</div>
<div class="parent">Parent
  <div>Child</div>
  <div>Child</div>
</div>
</body>
</html>
```


وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

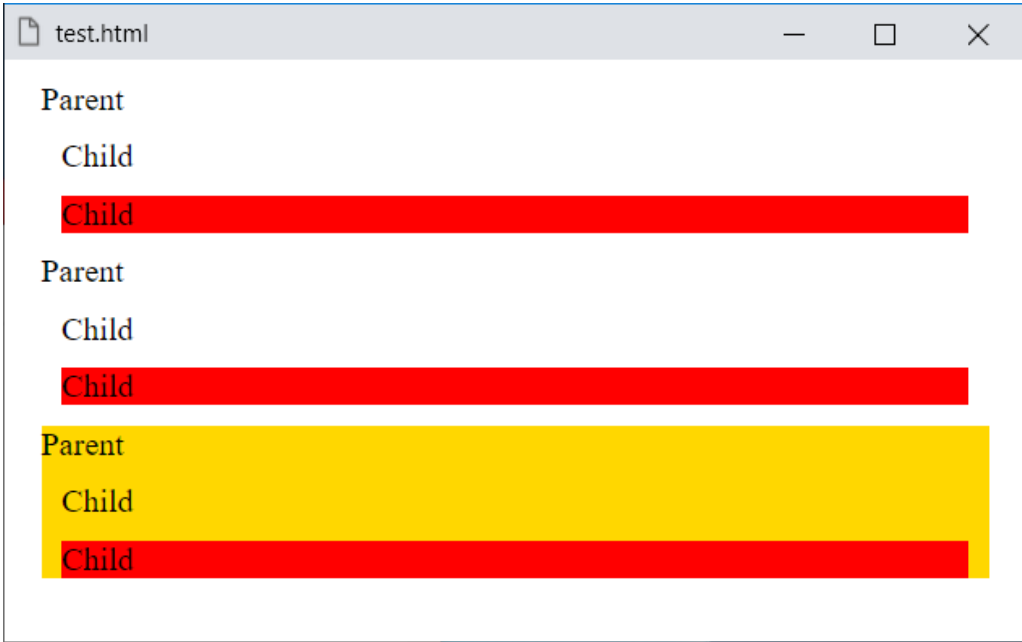


هذا المثال طبيعي، كما تعلمنا من قبل، ولكنه ماذا لو أردنا ثبات هذا التمييز، مع تمييز آخر للفروع!، وهنا نستطيع أن نقوم بحل هذه المشكلة، من خلال الفئة المستعارة `last-of-type`، كما سيوضح لنا إن شاء الله المثال التالي.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    div{ margin:10px; }
    div:last-of-type{
      background:red;
    }
    .parent:last-child{
      background:gold;
    }
  </style>
</head>
<body>
  <div class="parent">Parent
    <div>Child</div>
    <div>Child</div>
  </div>
  <div class="parent">Parent
    <div>Child</div>
    <div>Child</div>
  </div>
  <div class="parent">Parent
    <div>Child</div>
    <div>Child</div>
  </div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



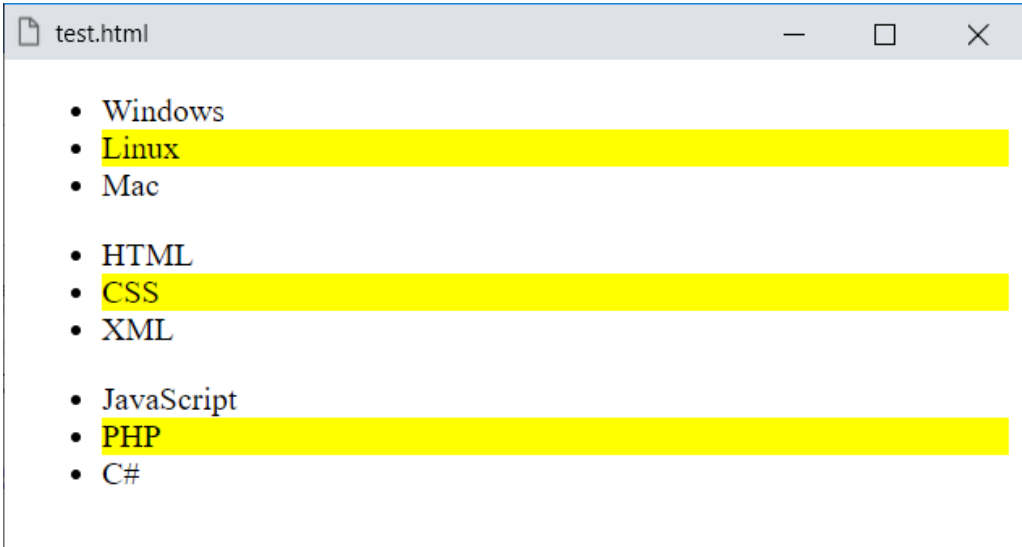
ونلاحظ هنا أننا فرقنا بين الأصل parent والفرع child بسهولة
التعامل مع الفئات المستعارة last-child و last-of-type.

رقم محدد لكل فرع nth-child

هنا لا نتحدث عن أول أو آخر فرع، بل نتحدث عن رقم يتم اختياره، أو تحديده بسهولة تامة، بفضل الله، والحروف nth تعني ترتيب رقمي مثل 1st و 2nd و 3rd و 4th، وهكذا، وبالطبع كلمة child تعني فرع، وباستخدام nth-child ننسق بترتيب محدد، مثلاً ثاني فرع.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  li:nth-child(2){ background:yellow; }
</style>
</head>
<body>
<ul>
  <li>Windows</li>
  <li>Linux</li>
  <li>Mac</li>
</ul>
<ul>
  <li>HTML</li>
  <li>CSS</li>
  <li>XML</li>
</ul>
<ul>
  <li>JavaScript</li>
  <li>PHP</li>
  <li>C#</li>
</ul>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



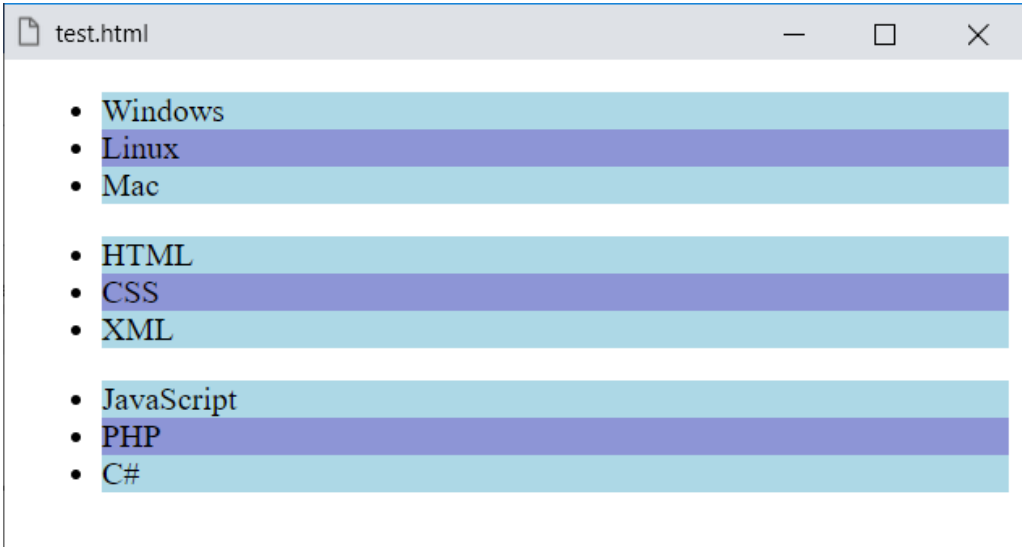
لاحظ أنه تم وضع أقواس عادية () بعد الفئة nth-child، وتم وضع الرقم 2 بداخلها، أي أننا نقصد ثاني فرع li من كل مجموعة، وبالفعل تم وضع خلفية صفراء لثاني كل فرع.

انتبه! فإنه يمكن كتابة أي رقم آخر بدلاً من الرقم 2 لتطبيق التنسيق على عناصر بتدبير آخر، كما أنه يمكن أيضاً كتابة كلمات لتحكم معيها، فمثلاً الكلمة odd للتطبيق على الفروع الفردية، والكلمة even للتطبيق على الفروع الزوجية.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    li:nth-child(odd){
      background:lightblue;
    }
    li:nth-child(even){
      background:#8d95d6;
    }
  </style>
</head>
<body>
  <ul>
    <li>Windows</li>
    <li>Linux</li>
    <li>Mac</li>
  </ul>
  <ul>
    <li>HTML</li>
    <li>CSS</li>
    <li>XML</li>
  </ul>
  <ul>
    <li>JavaScript</li>
    <li>PHP</li>
    <li>C#</li>
  </ul>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



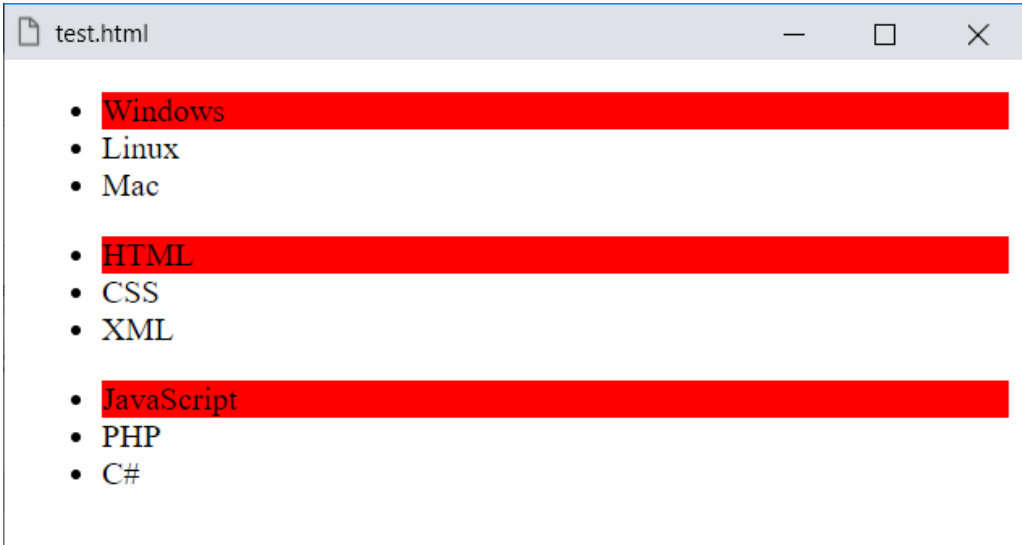
لاحظ أنه تم تلوين الترتيب الفردي باللون الأزرق الفاتح، وتم تلوين الترتيب الزوجي بلون آخر، فأصبح الشكل جمالياً.

رقم محدد لكل فرع من الأسفل nth-last-child

هنا نتحدث عن رقم يتم اختياره من الآخر، أو تحديده بسهولة تامة من الأسفل، بفضل الله، وباستخدام nth-last-child ننسق بترتيب محدد، مثلاً ثاني فرع من الأسفل، أو ثالث فرع من الأسفل.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  li:nth-last-child(3){ background:red; }
</style>
</head>
<body>
<ul>
<li>Windows</li>
<li>Linux</li>
<li>Mac</li>
</ul>
<ul>
<li>HTML</li>
<li>CSS</li>
<li>XML</li>
</ul>
<ul>
<li>JavaScript</li>
<li>PHP</li>
<li>C#</li>
</ul>
</body>
</html>
```


وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن المقصود هو تلك فرع من الأسفل، ومعناها أن يبدأ العد من آخر عنصر.

رقم محدد لكل نوع nth-of-type

وهنا نضع رقم محدد للنوع، ويبدأ العد من أول عنصر.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    li:nth-of-type(3){ background:gold; }
  </style>
</head>
<body>
  <ul>
    <li>Windows</li>
    <li>Linux</li>
    <li>Mac</li>
  </ul>
  <ul>
    <li>HTML</li>
    <li>CSS</li>
    <li>XML</li>
  </ul>
  <ul>
    <li>JavaScript</li>
    <li>PHP</li>
    <li>C#</li>
  </ul>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

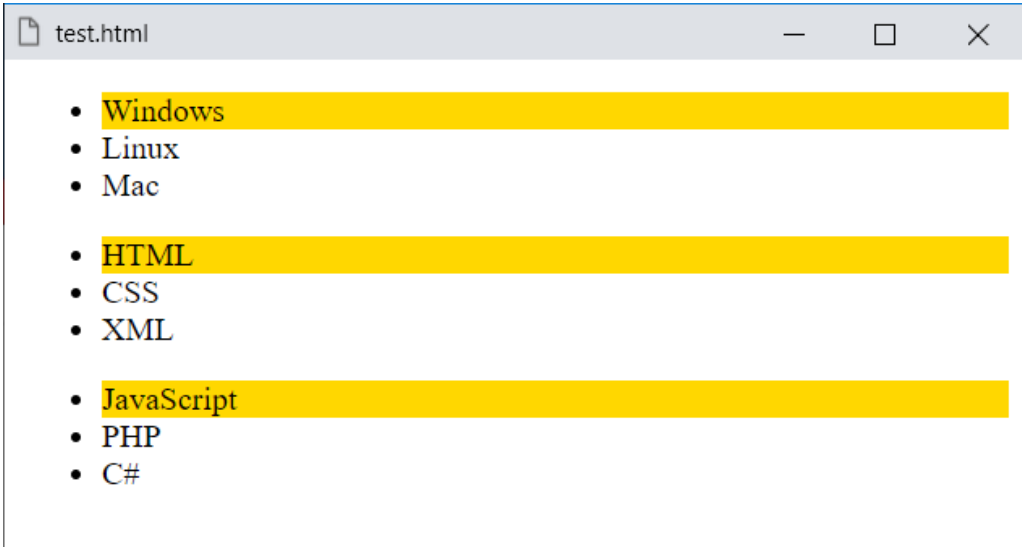


لاحظ أننا نقصد تلك فئة من كل نوع `li`، وبالفعل تم وضع خلفية زهيدة لتلك كل نوع.

رقم محدد لكل نوع من الأسفل `nth-last-of-type`
وهنا نضج رقم محدد للنوع، ويبدأ العد من آخر عنصر.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      li:nth-last-of-type(3){ background:gold; }
    </style>
  </head>
  <body>
    <ul>
      <li>Windows</li>
      <li>Linux</li>
      <li>Mac</li>
    </ul>
    <ul>
      <li>HTML</li>
      <li>CSS</li>
      <li>XML</li>
    </ul>
    <ul>
      <li>JavaScript</li>
      <li>PHP</li>
      <li>C#</li>
    </ul>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



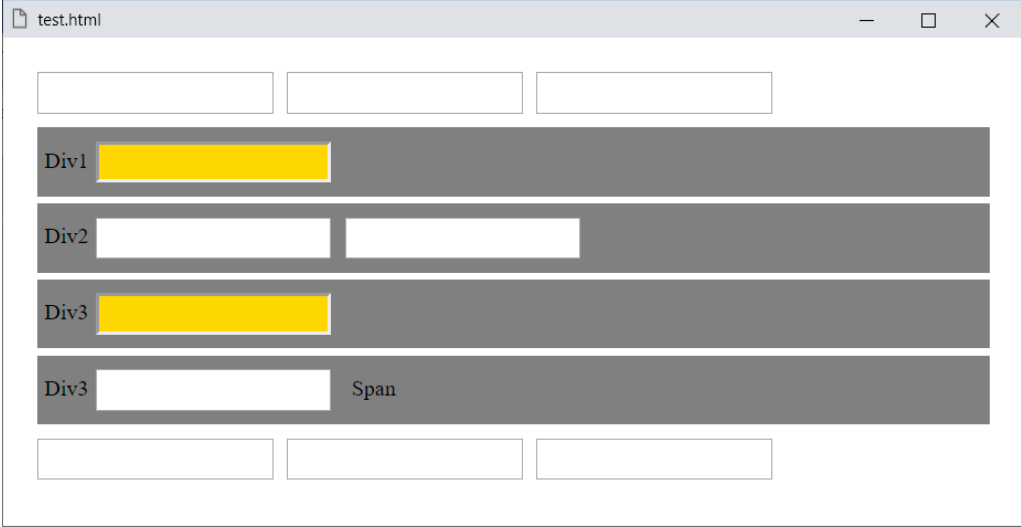
لاحظ أننا نغص ثلاث فروع من كل نوع Li من الأسفل، وبالفعل تم وضع خلفية ذهبية لثالث كل نوع من الأسفل.

الفروع فقط only-child

المقصود هنا هو التأثير على العنصر فقط لو كان له أصل parent وليس له نظراء siblings، سواء نظراء من نفس نوعه أو أي وسمة من نوع آخر، وسوف نفهم إن شاء الله من المثال التالي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      *{ margin:5px; padding:5px; }
      div{ background:gray; }
      input:only-child{ background:gold; }
    </style>
  </head>
  <body>
    <input><input><input>
    <div>Div1<input></div>
    <div>Div2<input><input></div>
    <div>Div3<input></div>
    <div>Div3<input><span>Span</span></div>
    <input><input><input>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ وجود ثلاثة وسوم `input`، ثم وسوم `div` بداخله وسوم `input` واحد، ثم وسوم `div` بداخله وسوم `input`، ثم وسوم `div` بداخله وسوم `input` واحد ووسوم `span` واحد، ثلاثة وسوم `input`.

لاحظ أن الذي تأثر فقط هو وسوم `input` الموجود في حاوية `div` بمفرده، وبدون أي نظائر.

انتبه! ولاحظ أنه في حالة وجود نظير من نفس النوع، أو من نوع آخر فلم يتم تنسيق أي واحد منهم.

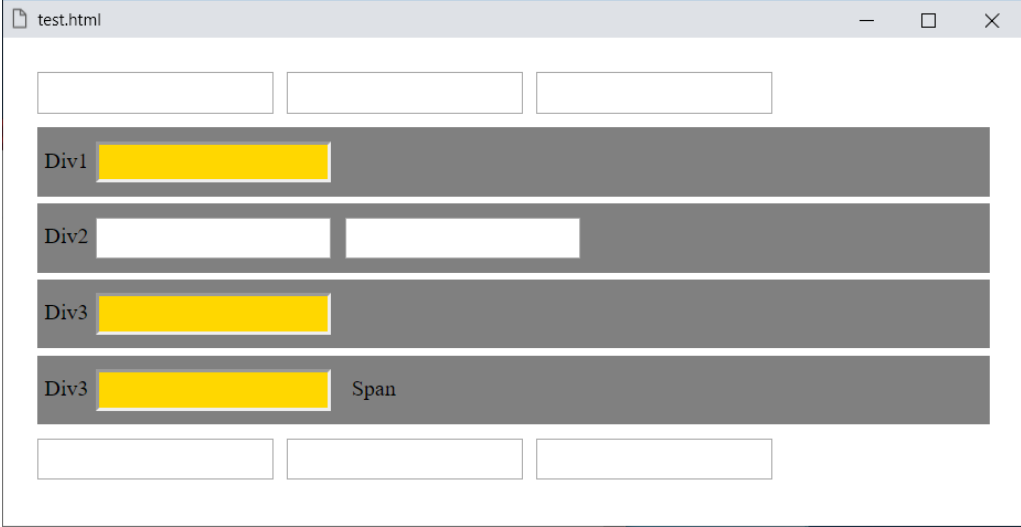
النوع فقط only-of-type

المقصود هنا هو التأثير على نوع العنصر فقط لو كان له أصل parent وليس له نظراء siblings، مع نفسه نوعه، وسوف نفهم إن شاء الله مع المثال التالي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      *{ margin:5px; padding:5px; }
      div{ background:gray; }
      input: only-of-type{ background:gold; }
    </style>
  </head>
  <body>
    <input><input><input>
    <div>Div1<input></div>
    <div>Div2<input><input></div>
    <div>Div3<input></div>
    <div>Div3<input><span>Span</span></div>
    <input><input><input>
  </body>
</html>
```


وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ وجود ثلاثة وسوم `input`، ثم وسوم `div` بداخله وسوم `input` واحد، ثم وسوم `div` بداخله وسوم `input`، ثم وسوم `div` بداخله وسوم `input` واحد، ثم وسوم `input` واحد، ثم وسوم `div` بداخله وسوم `input` واحد ووسوم `span` واحد، ثلاثة وسوم `input`.

لاحظ أن الذي تأثر هو وسوم `input` الموجود في حاوية `div` بمفرده، وبدون أي نظائر مع نفس نوعه، ومع نظيره `span` تأثر أيضاً.

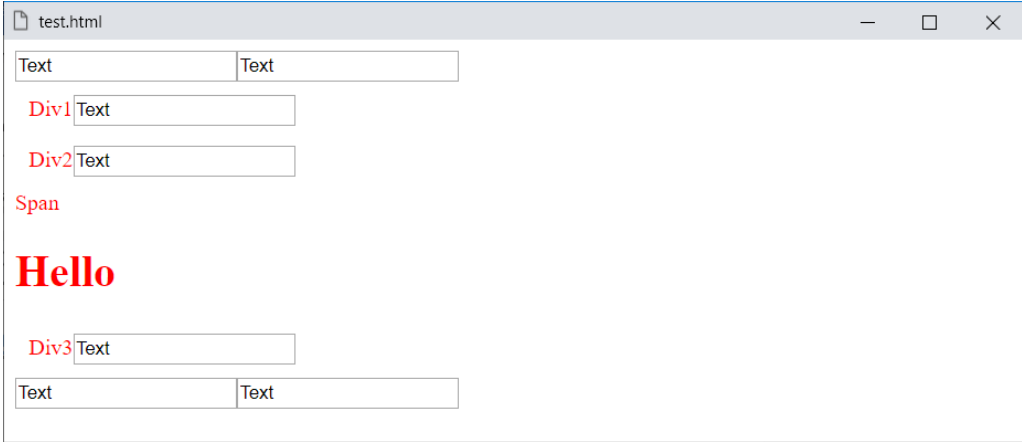
انتبه! ولاحظ أنه في حالة وجود نظير مع نفس النوع، فلم يتم تنسيق أي واحد منهم.

الل ما عدا not

المقصود هنا أنه يتم تطبيق التنسيق على جميع الوسوم بالكامل فيما عدا
وسم يتم كتابته بين قوسيه عاديه (.).
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      div{ margin:5px; padding:5px; }
      :not(input){ color:red; }
    </style>
  </head>
  <body>
    <input value="Text"><input value="Text">
    <div>Div1<input value="Text"></div>
    <div>Div2<input value="Text"></div>
    <span>Span</span><h1>Hello</h1>
    <div>Div3<input value="Text"></div>
    <input value="Text"><input value="Text">
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

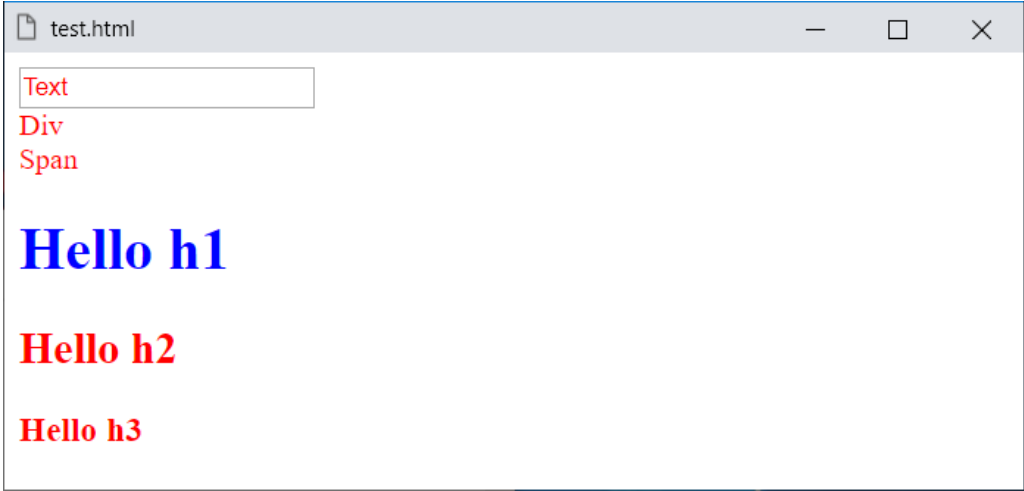


لاحظ أن التنسيق تم على كل الوسوم فيما عدا الوسم input في أي مكان، حتى ولو كان في وسم يحمل تنسيق مثل div.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      h1,h3{ color:blue; }
      :not(h1){ color:red; }
    </style>
  </head>
  <body>
    <input value="Text"><div>Div</div>
    <span>Span</span>
    <h1>Hello h1</h1>
    <h2>Hello h2</h2>
    <h3>Hello h3</h3>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم تنسيق الؤسم h1 والؤسم h3، ولكه بعد كؤد الاستثناء، والذي هو "`:not(h1){ color:red; }`" تم تنسيق الك بما في ذلك الؤسم h3، والذي بقي باللون الأزرق هو الؤسم h1، لأنه هو المستثنى.

انتبه! فإنه يمكن استخدام `:is` و `:where` و `:has` للتطابق، ويمكن الاطلاع على اختياراته أكدّ قءما مه الرابط التالي:

<https://www.w3.org/TR/selectors-4/>

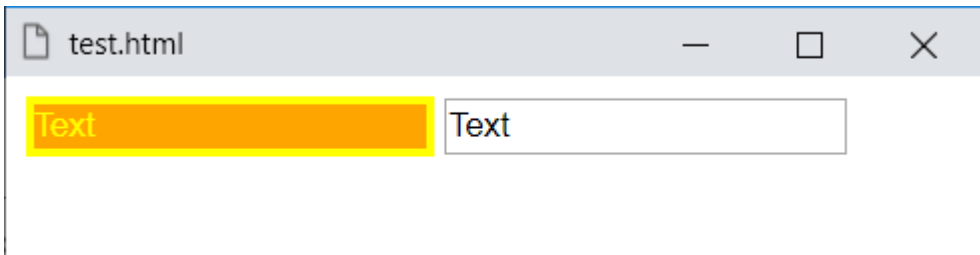
التعطيل disabled

هنا يتم التنسيق على عناصر معطلة بالفئة disabled، وسيتم التوضيح فيما يلي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
input:disabled{
border:3px solid yellow;
background:orange;
color:yellow;
}
</style>
</head>
<body>
<input value="Text" disabled>
<input value="Text">
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن العناصر التي تأثرت بالتنسيق هي العناصر الغير مفعلة disabled فقط.

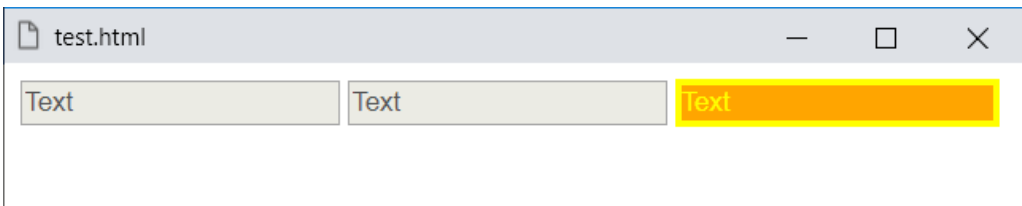
التفعيل enabled

هنا يتم التنسيق على عناصر مفعلة بالفئة enabled، وسيتم التوضيح فيما يلي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
input:enabled{
border:3px solid yellow;
background:orange;
color:yellow;
}
</style>
</head>
<body>
<input value="Text" disabled>
<input value="Text" disabled>
<input value="Text">
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن العناصر التي تأثرت بالتنسيق هي العناصر المفعلة enabled فقط.

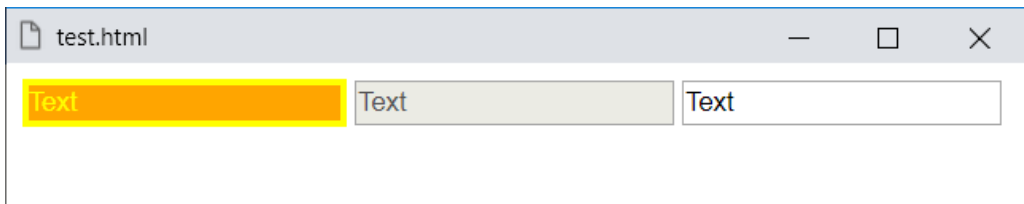
القراءة فقط read-only

هنا يتم التنسيق على عناصر تم ضبطها من قبل على القراءة فقط بالفئة read-only، وسيتم التوضيح فيما يلي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
input:read-only{
border:3px solid yellow;
background:orange;
color:yellow;
}
</style>
</head>
<body>
<input value="Text" readonly>
<input value="Text" disabled>
<input value="Text">
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن العنصر الذي تم ضبطه على القراءة فقط قد تأثر.

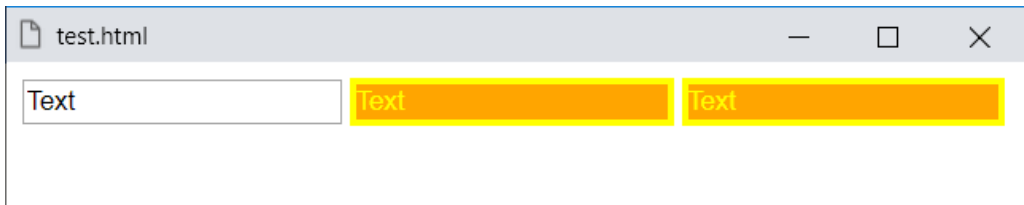
غفة مضبوء على القراءء فقط read-write

هنا المقصوء هو العنصر الءى لم فءم مضبوء بـ HTML ءَ readonly، ءءى ولو ءاه غفة مفعء disabled سبءاءء.

ءوء! قم بءءابة الءوء الءاءى ءم اءفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  input:read-write{
    border:3px solid yellow;
    background:orange;
    color:yellow;
  }
</style>
</head>
<body>
  <input value="Text" readonly>
  <input value="Text" disabled>
  <input value="Text">
</body>
</html>
```

وبءء ءشءفل الصفاءء ءظهر ءما بالشءك الءاءى:



لاءظ أن أى عنصر لم فءم مضبوء ءَ readonly قد ءاءءء بالءنسفق.

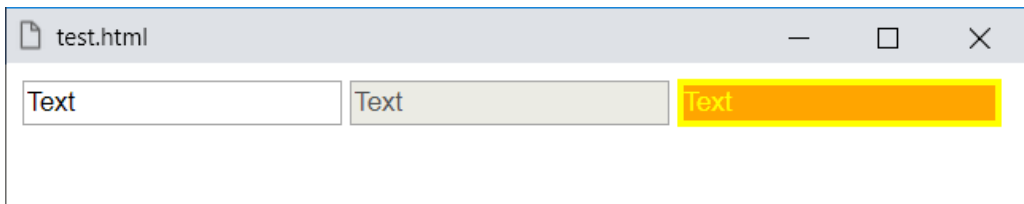
المطلوب required

هنا العنصر المقصود، هو الذي تم تحديده كمطلوب مع html وهو
 .required

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    input:required{
      border:3px solid yellow;
      background:orange;
      color:yellow;
    }
  </style>
</head>
<body>
  <input value="Text" readonly>
  <input value="Text" disabled>
  <input value="Text" required>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



وهنا تأثر الذي تم ضبطه على required مع اللغة html فقط.

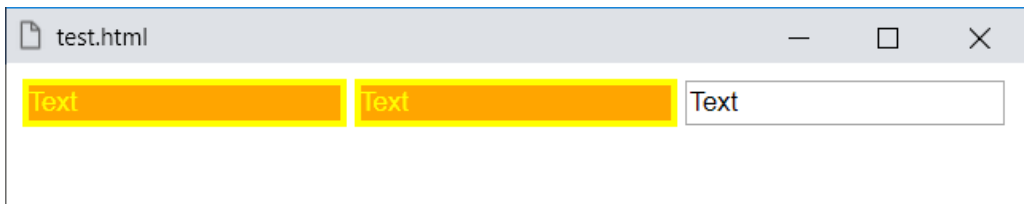
الغير المطلوب أو الاختياري optional

هنا العنصر المقصود، هو الذي تم تحديده كمدخل عادي مع html وهو optional أي اختياري.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
input:optional{
border:3px solid yellow;
background:orange;
color:yellow;
}
</style>
</head>
<body>
<input value="Text" readonly>
<input value="Text" disabled>
<input value="Text" required>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



وهنا تأثر الذي لم يتم ضبطه على required مع اللغة html فقط، وما لم يحدد ك required فهو اختياري.

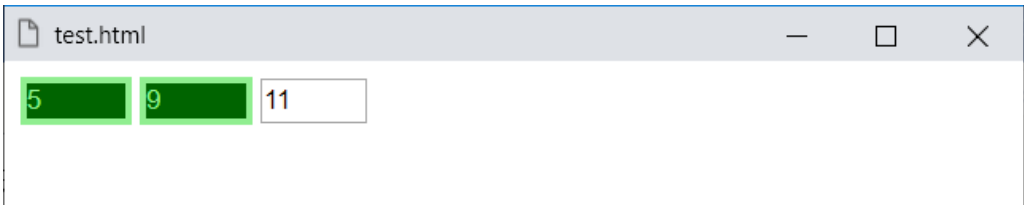
داخل المدى in-range

نستطيع أن نقوم بعمل تنسيق على العناصر المنضبطة داخل المدى كالتالي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
input:in-range{
border:3px solid lightgreen;
background:darkgreen;
color:lightgreen;
}
</style>
</head>
<body>
<input type="number" min="3" max="9" value="5">
<input type="number" min="3" max="9" value="9">
<input type="number" min="3" max="9" value="11">
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن النوع number، ولاحظ الذي تأثر فقط هي العناصر التي داخل المدى المحدد.

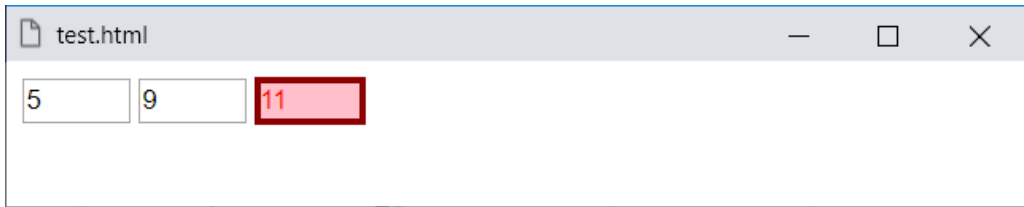
خارج المدى out-of-range

نستطيع أن نقوم بعمل تنسيق على العناصر الخارجة عن المدى كالتالي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
input:out-of-range{
border:3px solid darkred;
background:pink;
color:red;
}
</style>
</head>
<body>
<input type="number" min="3" max="9" value="5">
<input type="number" min="3" max="9" value="9">
<input type="number" min="3" max="9" value="11">
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن النوع number، ولاحظ الذي تأثر فقط هي العناصر التي خرجت عن المدى المحدد.

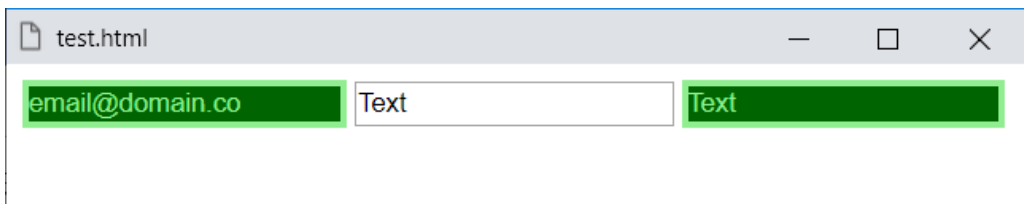
تم التحقق منه valid

وهنا يمكننا توكيز التنسيق على العناصر التي تم التحقق منها سواء مطلوبة required أو رقم number أو بريد email أو غير ذلك.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
input:valid{
border:3px solid lightgreen;
background:darkgreen;
color:lightgreen;
}
</style>
</head>
<body>
<input type="email" value="email@domain.co" >
<input type="email" value="Text" >
<input value="Text" required>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم تفعيل التنسيق فقط على العناصر التي تم التحقق منها.

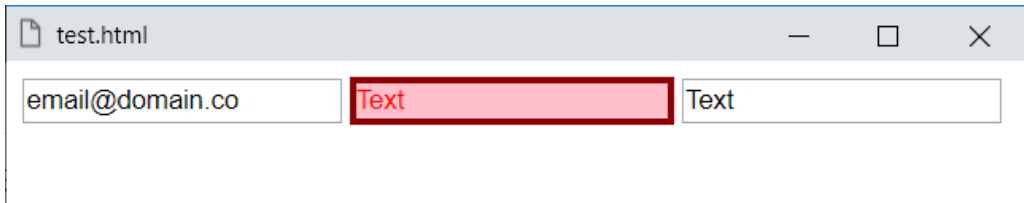
لم يتم التحقق منه invalid

وهنا يمكننا تذكير التنسيق على العناصر التي لم يتم التحقق منها سواء مطلوبة required أو رقم number أو بريد email أو غير ذلك.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
input:invalid{
border:3px solid darkred;
background:pink;
color:red;
}
</style>
</head>
<body>
<input type="email" value="email@domain.co" >
<input type="email" value="Text" >
<input value="Text" required>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



test.html

email@domain.co Text Text

لاحظ أنه تم تفعيل التنسيق فقط على العناصر التي لم يتم التحقق منها.

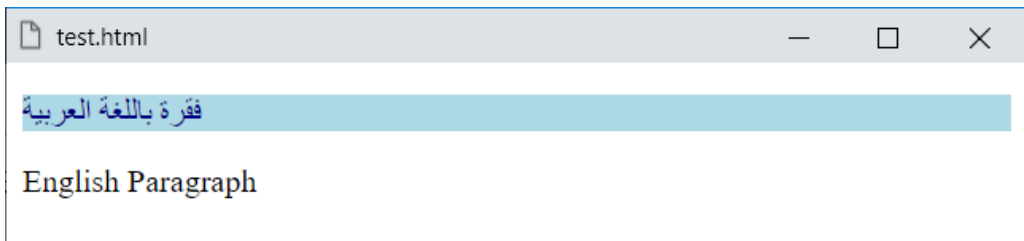
اللغة lang

يمكّن التنسيق على لغة محددة من اللغات، كما سنوضح إن شاء الله.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  p:lang(ar){
    background:lightblue;
    color:navy;
  }
</style>
</head>
<body>
  <p lang="ar">فقرة باللغة العربية</p>
  <p lang="en">English Paragraph</p>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن المحدد lang بعده قوسيه وبهم اللغة العربية ar، ولذلك هي التي تم التأيد عليها بالتنسيق.

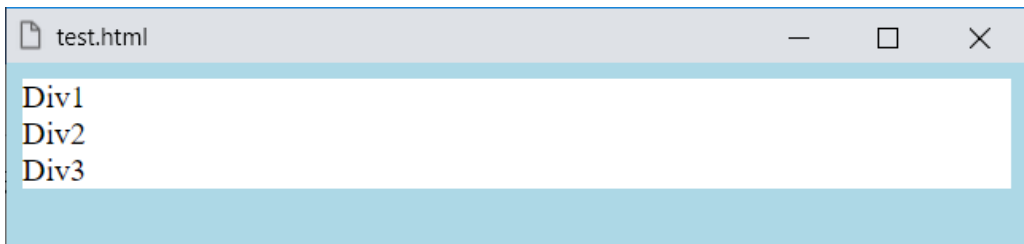
الأصل root

المقصود هنا هو جذور أو أصل المستند، مع العلم أن مستند html الأصل الخاص به هو دائماً يكون الوسم .html.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    :root{
      background:lightblue;
    }
    body{ background:#fff; }
  </style>
</head>
<body>
  <div>Div1<div><div>Div2<div><div>Div3<div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن التأثير على الوسم html، والمحتوي الأبيض تم تخصيصه للوسم .body.

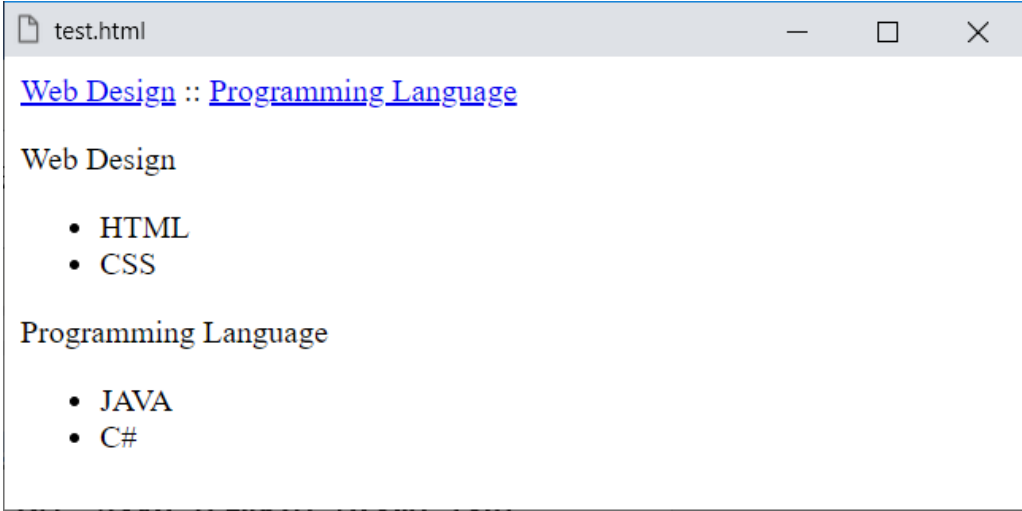
هدف تم استهدافه target

هنا نستطيع التنسيق على المحتوى الذي تم استهدافه برابط، كالتالي:

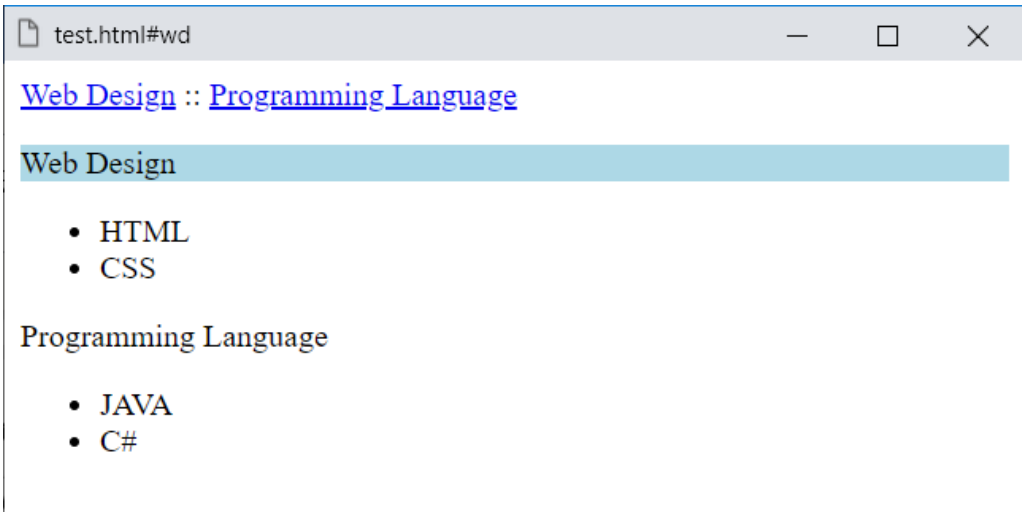
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      :target{
        background:lightblue;
      }
    </style>
  </head>
  <body>
    <a href="#wd">Web Design</a> ::
    <a href="#pl">Programming Language</a>
    <p id="wd">Web Design</p>
    <ul>
      <li>HTML</li>
      <li>CSS</li>
    </ul>
    <p id="pl">Programming Language</p>
    <ul>
      <li>JAVA</li>
      <li>C#</li>
    </ul>
  </body>
</html>
```

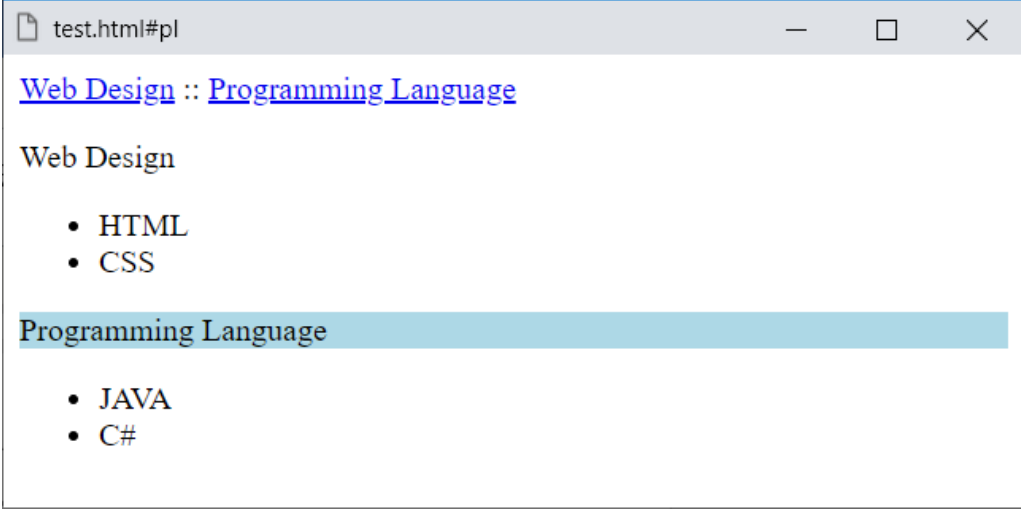
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم عمل روابط داخلية، ومحتوي يتم استهدافه لها، إذا تم الضغط عليها، ولاحظ أن الصفحة تظهر بشكل عادي ولم يتأثر شيء عليها حتى الآن، ولكنه رجاء **انتبه!**، فإذا تم الضغط على الرابط سوف يحدث شيء آخر، فالآن سوف نضغط على الرابط الأول، ليكون مثل التالي:



والآن سوف نضغط على الرابط الثاني، ليكون مثل التالي:



وكما نلاحظ فإن التنسيق، هام جداً لتحديد الهدف الذي تم استهدافه.

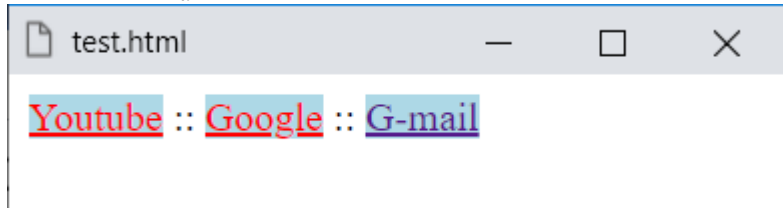
الرابط link

هنا يتم التنسيق على الرابط الذي لم تتم زيارته.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  a:link{
    background:lightblue; color:red
  }
</style>
</head>
<body>
<a href="http://www.youtube.com">Youtube</a> ::
<a href="http://www.google.com">Google</a> ::
<a href="http://www.gmail.com">G-mail</a>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم عمل تنسيق على الروابط التي لم تتم زيارتها، مع العلم أن الموقع gmail.com تمت زيارته، ولكنه تأثر بخلفية المحدد link لأنه لم يخصص له خلفية.

رابط تمت زيارته visited

هنا يتم التنسيق على الرابط الذي تمت زيارته بالفعل.

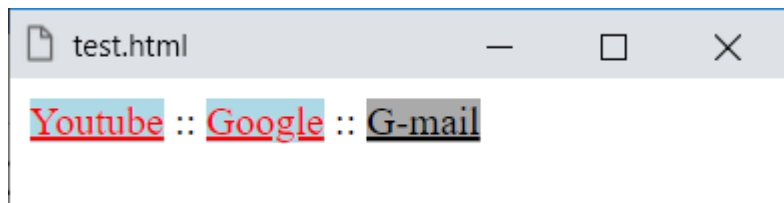
كود! قم بكتابة الكود التالي ثم احفظه:

```

<!DOCTYPE html>
<html>
<head>
<style>
  a:link{ background:lightblue; color:red }
  a:visited{ background:#aaa; color:black }
</style>
</head>
<body>
<a href="http://www.youtube.com">Youtube</a> ::
<a href="http://www.google.com">Google</a> ::
<a href="http://www.gmail.com">G-mail</a>
</body>
</html>

```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم عمل تنسيق على الروابط التي لم تتم زيارتها، مع العلم أن الموقع gmail.com تمت زيارته، وتم عمل تخصيص له حتى لا يتأثر بـ link، حيث أخذ تنسيق المحدد visited بالكامل.

الفاعلية active

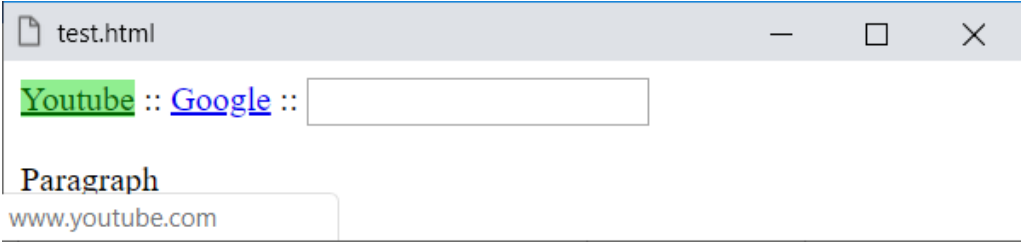
هنا رجاء **انتبه!**، وركز حتى تستطيع رؤية التأثير، حيث المقصود بالمحدد active فاعلية العنصر، وسوف نحاول التوضيح كالتالي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      a:active, input:active, p:active{
        background:lightgreen;
        color:darkgreen
      }
    </style>
  </head>
  <body>
    <a
href="http://www.youtube.com">Youtube</a>
::
    <a
href="http://www.google.com">Google</a> ::
    <input> <p>Paragraph</p>
  </body>
</html>
```

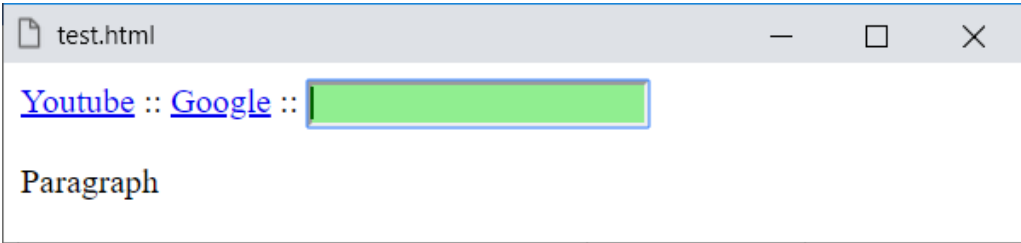
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن الوضع طبيعي جداً، ولم يتأثر أي شيء، ولكنه **انتبه!** وانضغط واستمر بالضغط على رابط يوتيوب ليكون مثل الشكل التالي:



وهذا هو المحدد active، ورجاء **انتبه!** ولاحظ قبل رفع الماوس عن على الرابط، والآه لو ضغطت بالماوس على input يكون كالتالي:



وهكذا، ولكنه **انتبه!** فله تدرى التأثير إلا أثناء الاستمرار بالضغط على الماوس فقط.

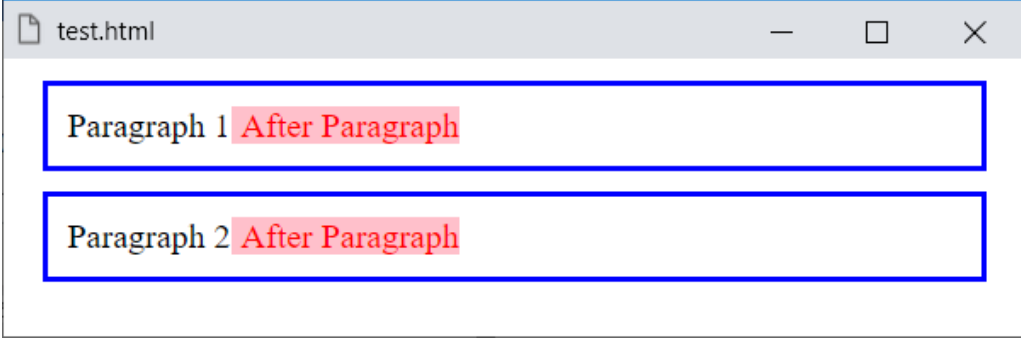
ما بعد after

المحدد after هو ال Pseudo Elements، وهو ما بعد العنصر، ويمك استخدامه خاصة content في لغة CSS وتخصيص نص لها لرؤيته بعد العنصر المقصود.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      p{
        margin:10px; padding:10px;
        border:2px solid blue;
      }
      p::after{
        content:'After Paragraph';
        background:pink;
        color:red;
        padding-left:5px;
      }
    </style>
  </head>
  <body>
    <p>Paragraph 1</p>
    <p>Paragraph 2</p>
  </body>
</html>
```


وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أننا نتحكم في وضع أشياء بعد العناصر، باستخدام الخاصية content و نتحكم في تنسيق لهذه الأشياء.

لاحظ الحشو padding والعامش margin والإطار border للفقرات.

لاحظ أن المحدد after يتم استخدامه بواسطة ":", وهو من العناصر المستعارة في لغة CSS.

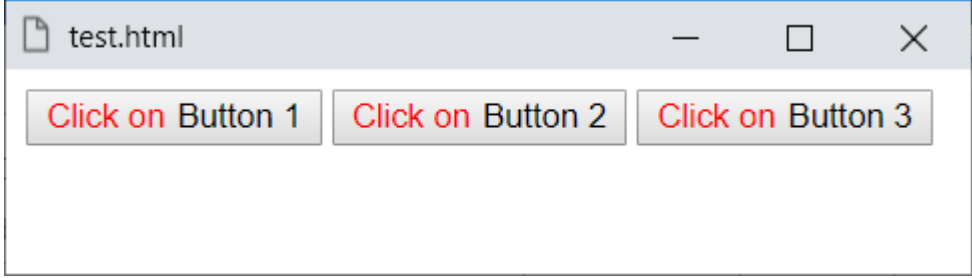
ما قبل before

المحدد before هو ال Pseudo Elements، وهو ما قبل العنصر، ويمك استخدامه خاصة content في لغة CSS وتخصيص نص لها لرؤيته بعد العنصر المقصود.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    button::before{
      content:'Click on';
      color:red;
      padding-right:5px;
    }
  </style>
</head>
<body>
  <button>Button 1</button>
  <button>Button 2</button>
  <button>Button 3</button>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



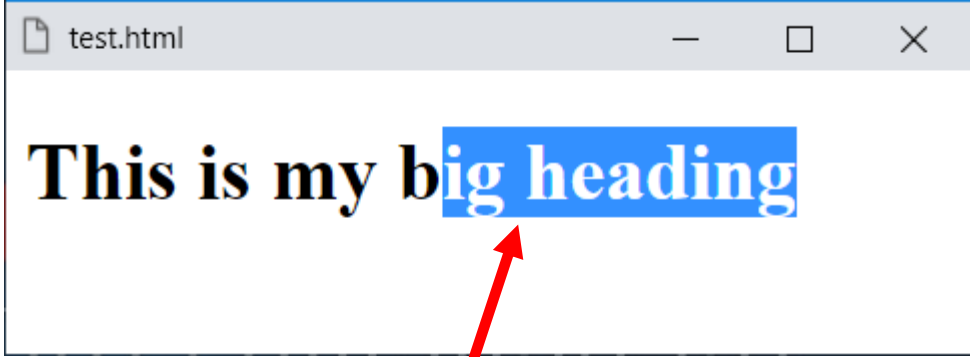
لاحظ أننا نتحكم في وضع أشياء قبل العناصر، باستخدام الخاصية content وتتحكم في تنسيق لهذه الأشياء.

لاحظ أن المحدد before يتم استخدامه بواسطة ":", وهو من العناصر المستعارة في لغة CSS.

لاحظ أننا تدخلنا تدخل مباشر في هيكلية الـ button.

شيء تم تحديده selection

المحدد selection هو ال Pseudo Elements، وهو ما يتم تحديده، فمع الطبيعي أنه إذا تم تحديد شيء يكون مثل التالي:



جزء محدد من النص

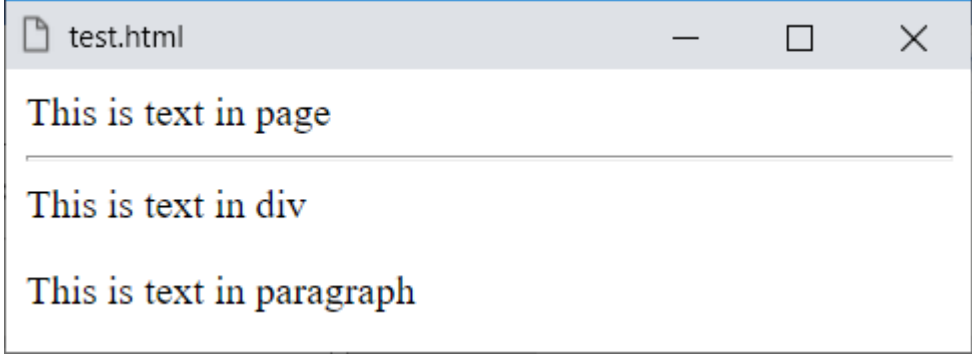
وهنا يأتي المحدد selection ويلعب دوراً هاماً، حيث يغير تنسيقات الشيء المحدد، مثل الموجود أمامك في الشكل السابق.

الآن بفضل الله سوف نشرح مثال، لتفهم منه هذا الموضوع إن شاء الله ببساطة، ولكن **انتبه!** فيجب عليك التركيز والتجربة.

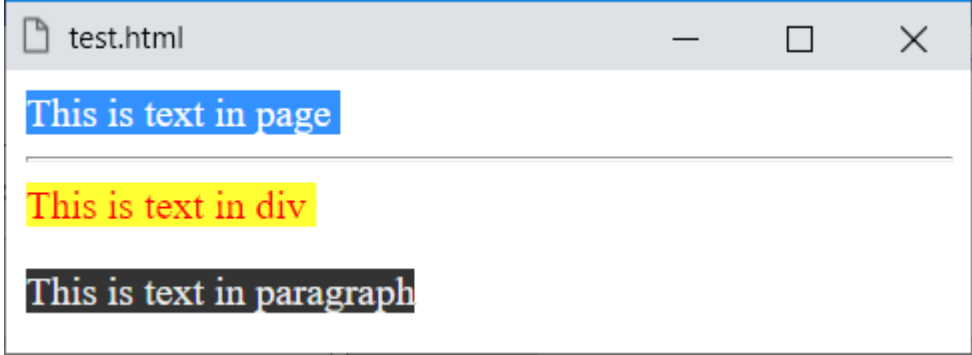
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      div::selection{
        background:yellow;
        color:red;
      }
      p::selection{
        background:#222;
        color:#eee;
      }
    </style>
  </head>
  <body>
    This is text in page<hr>
    <div>This is text in div</div>
    <p>This is text in paragraph</p>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



والآن قم بتحديدهم جميعاً ، أو اضغط Ctrl+A وأنت على الصفحة لتحديد الكل، وسوف ترى مثل الشكل التالي:



لاحظ أن الطبيعي هو النص المكتوب على الصفحة، وتم التأثير بشكل مختلف لكل من الذي يتم تحديده للوسم div والوسم p.

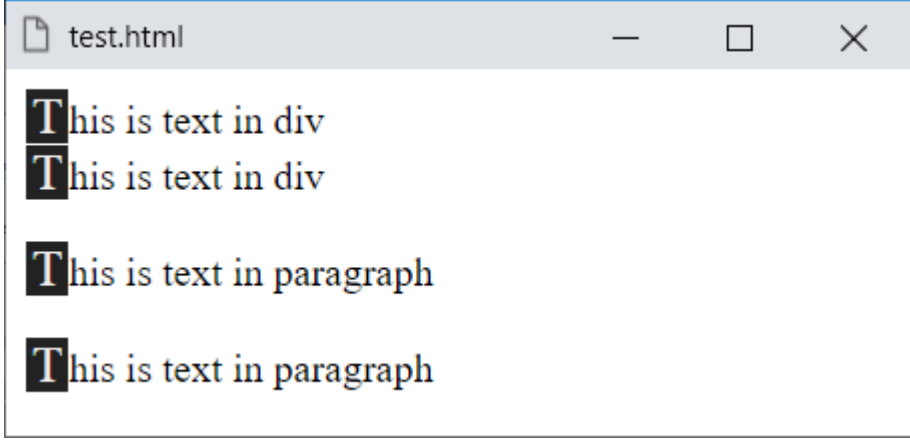
الحرف الأول first-letter

المحدد first-letter مع ال Pseudo Elements، وهو الحرف الأول مع العنصر، وسوف نبيد فيما يلي إن شاء الله.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      div::first-letter, p::first-letter{
        padding-left:3px;
        padding-right:3px;
        font-size:20px;
        background:#222;
        color:#eee;
      }
    </style>
  </head>
  <body>
    <div>This is text in div</div>
    <div>This is text in div</div>
    <p>This is text in paragraph</p>
    <p>This is text in paragraph</p>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم التحكم في تنسيقات أول حرف مه كل وسم div، وأول حرف مه كل وسم p.

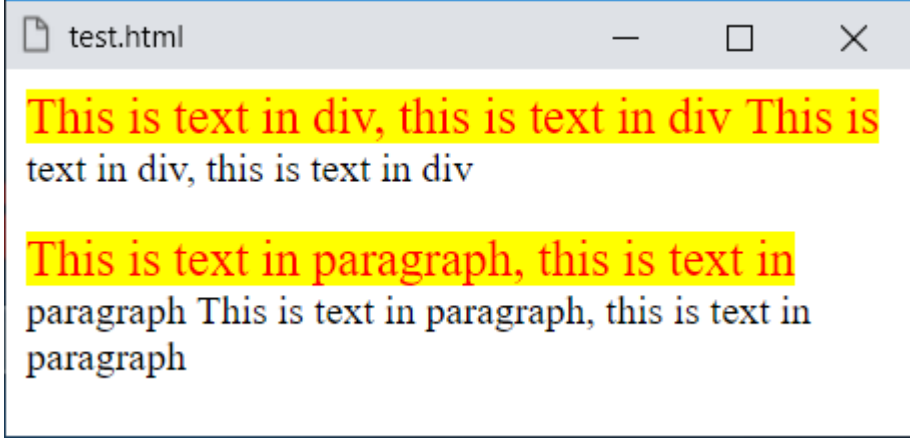
السطر الأول first-line

المحدد first-line مع ال Pseudo Elements، وهو السطر الأول مع العنصر، وسوف نبيد فيما يلي إن شاء الله.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
div::first-line, p::first-line{
padding-left:3px;
padding-right:3px;
font-size:20px;
background:yellow;
color:red;
}
</style>
</head>
<body>
<div>
This is text in div, this is text in div
This is text in div, this is text in div
</div>
<p>
This is text in paragraph, this is text in paragraph
This is text in paragraph, this is text in paragraph
</p>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم التحكم في تنسيقات أول سطر من كل وسم `div`، وأول سطر من كل وسم `p`.

Lists القوائم

العمل مع القوائم في لغة CSS شيء مهم وجميل، لأنك تتحكم فيه بكل تفاصيله، وتستطيع التغيير كما تريد، بصرف النظر عن كونه `ol` أو `ul`.

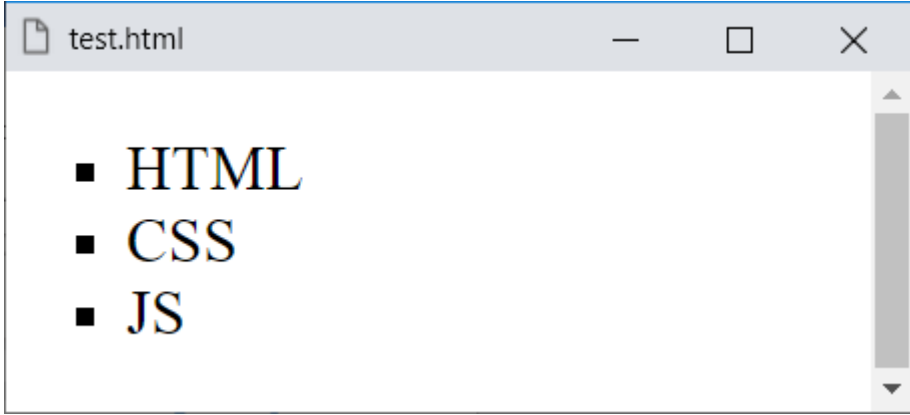
نوع القوائم `list-style-type`

يمكنك التحكم في نوع القائمة كما يلي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      ul{
        font-size:25px;
        list-style-type:square;
      }
    </style>
  </head>
  <body>
    <ul>
      <li>HTML</li>
      <li>CSS</li>
      <li>JS</li>
    </ul>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



وهنا استخدامنا الخاصية `list-style-type` لتغيير نوع النقاط، وتم تخصيص القيمة `square` لها حتى يتحول شكل النقاط الى شكل المربع، كما هو واضح في الشكل.

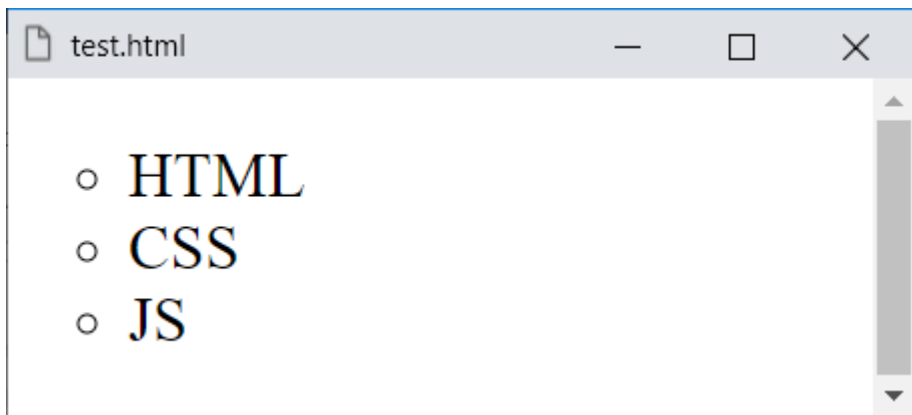
لاحظ أنه تم التغيير باللغة CSS في نوع الوسم `ul` بدون أي تدخل في اللغة html بأي شكل.

انتبه! فإن القيمة الافتراضية الخاصية `list-style-type` هي `disc`، وهي النقطة العادية.

يمكنك التحويل الى دائرة مفرغة باستخدام القيمة `.circle`
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      ul{
        font-size:25px;
        list-style-type:circle;
      }
    </style>
  </head>
  <body>
    <ul>
      <li>HTML</li>
      <li>CSS</li>
      <li>JS</li>
    </ul>
  </body>
</html>
```

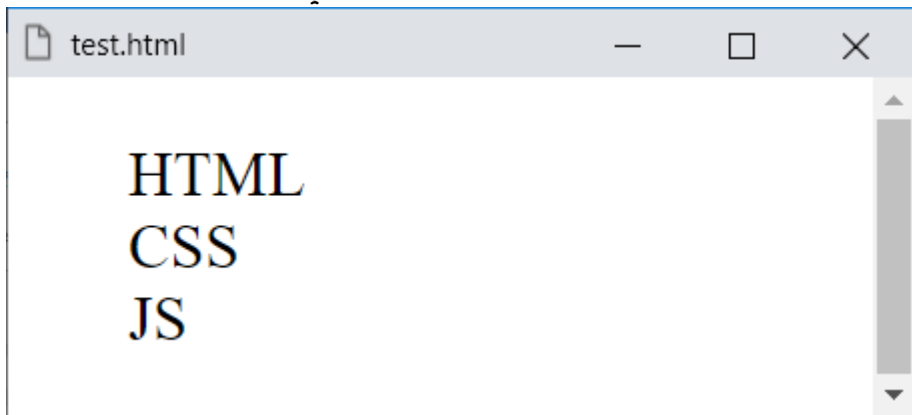
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



يملك التحويل إلى لا شيء، عن طريق القيمة `.none`.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  ul{
    font-size:25px;
    list-style-type:none;
  }
</style>
</head>
<body>
<ul>
<li>HTML</li>
<li>CSS</li>
<li>JS</li>
</ul>
</body>
</html>
```

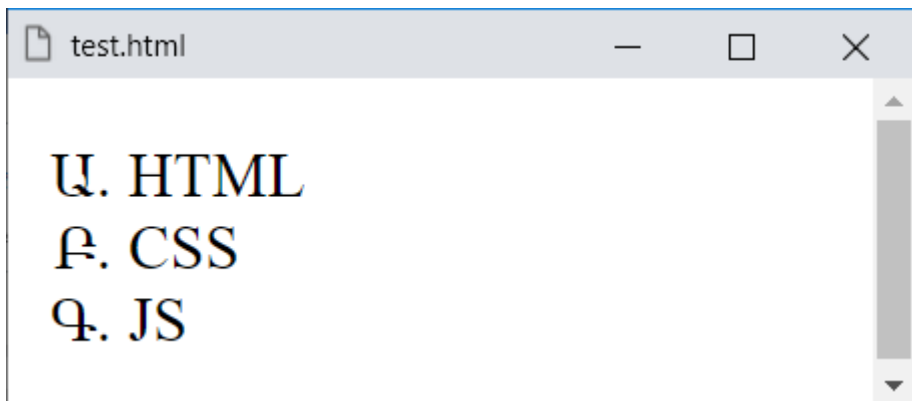
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



يمكنك التحويل الى الأرميني عن طريق القيمة .armenian
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      ul{
        font-size:25px;
        list-style-type:armenian;
      }
    </style>
  </head>
  <body>
    <ul>
      <li>HTML</li>
      <li>CSS</li>
      <li>JS</li>
    </ul>
  </body>
</html>
```

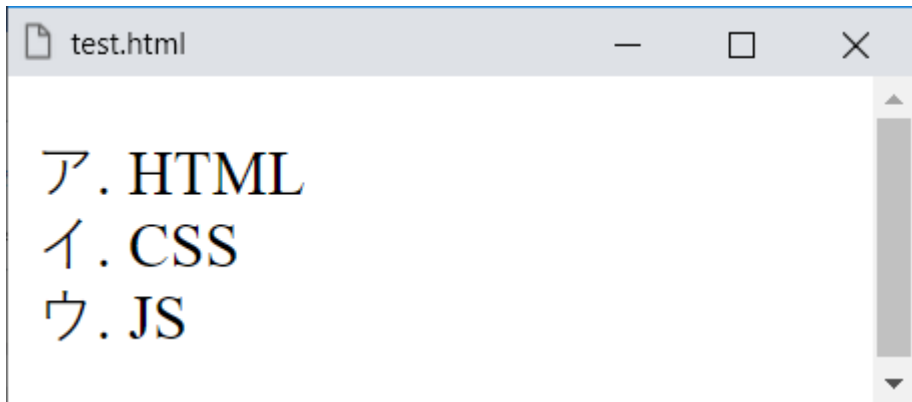
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



يملك التحويل الى كاتاكانا اليابانية عن طريق القيمة .katakana
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  ul{
    font-size:25px;
    list-style-type:katakana;
  }
</style>
</head>
<body>
<ul>
<li>HTML</li>
<li>CSS</li>
<li>JS</li>
</ul>
</body>
</html>
```

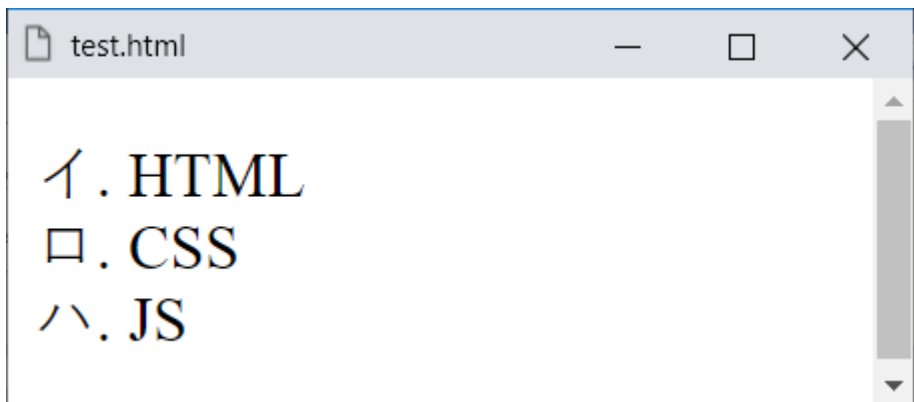
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



.katakana-iroha تحويل الـ katakana-iroha بالقيمة
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      ul{
        font-size:25px;
        list-style-type:katakana-iroha;
      }
    </style>
  </head>
  <body>
    <ul>
      <li>HTML</li>
      <li>CSS</li>
      <li>JS</li>
    </ul>
  </body>
</html>
```

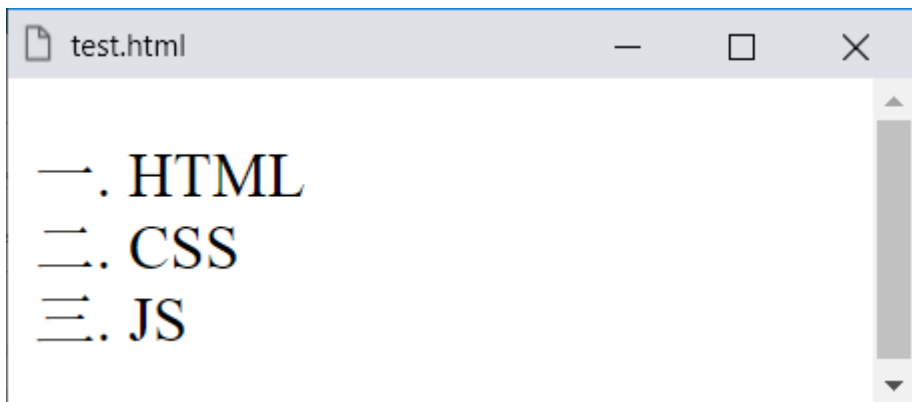
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



التحويل لـ cjk صينية يابانية كورية بالقيمة cjk-ideographic.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      ul{
        font-size:25px;
        list-style-type:cjk-ideographic;
      }
    </style>
  </head>
  <body>
    <ul>
      <li>HTML</li>
      <li>CSS</li>
      <li>JS</li>
    </ul>
  </body>
</html>
```

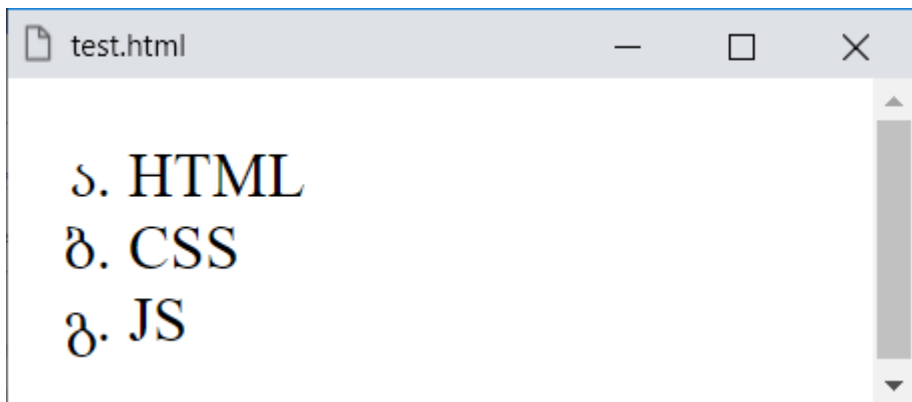
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



التحويل الى الجورجية بالقيمة .georgian
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      ul{
        font-size:25px;
        list-style-type:georgian;
      }
    </style>
  </head>
  <body>
    <ul>
      <li>HTML</li>
      <li>CSS</li>
      <li>JS</li>
    </ul>
  </body>
</html>
```

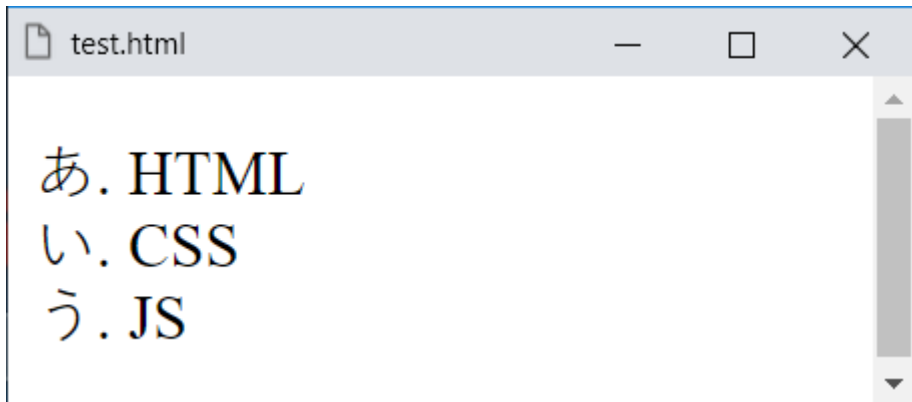
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



تحويل الى هيراغانا اليابانية بالقيمة hiragana.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
ul{
font-size:25px;
list-style-type:hiragana;
}
</style>
</head>
<body>
<ul>
<li>HTML</li>
<li>CSS</li>
<li>JS</li>
</ul>
</body>
</html>
```

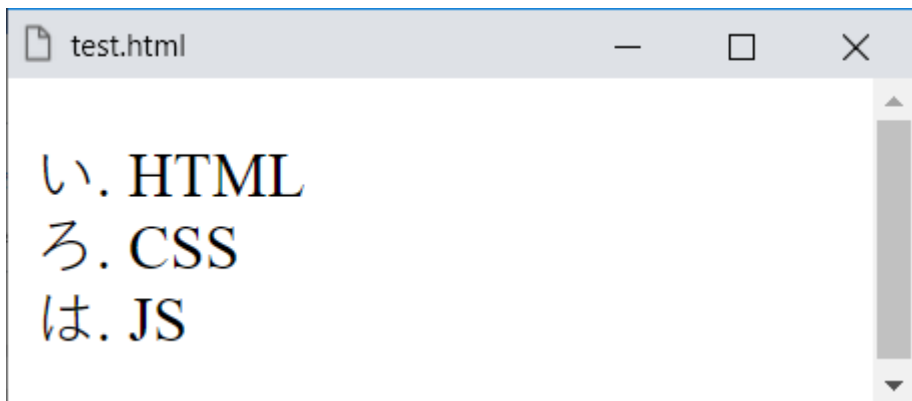
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



.hiragana-iroha hiragana-iroha بالقيمة تحويل الى
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      ul{
        font-size:25px;
        list-style-type:hiragana-iroha;
      }
    </style>
  </head>
  <body>
    <ul>
      <li>HTML</li>
      <li>CSS</li>
      <li>JS</li>
    </ul>
  </body>
</html>
```

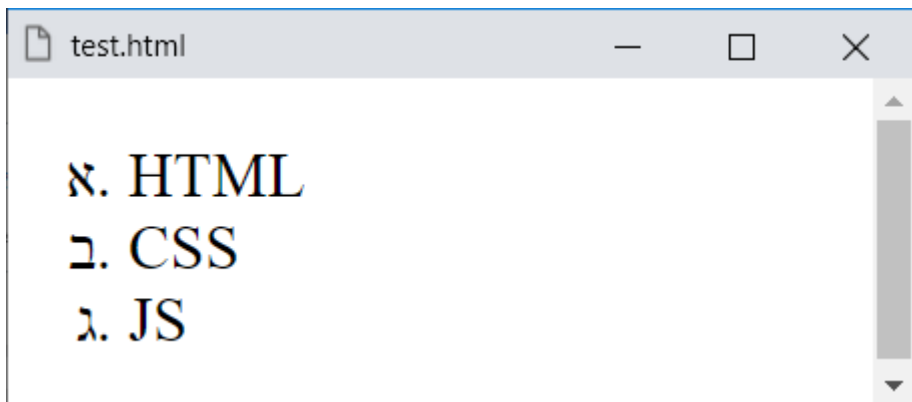
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



التحويل الى اللغة العبرية بالقيمة .hebrew
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      ul{
        font-size:25px;
        list-style-type:hebrew;
      }
    </style>
  </head>
  <body>
    <ul>
      <li>HTML</li>
      <li>CSS</li>
      <li>JS</li>
    </ul>
  </body>
</html>
```

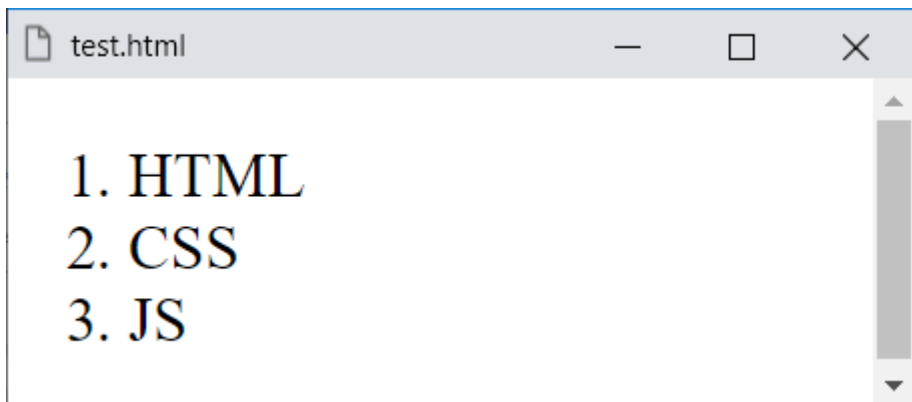
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



التحويل الى ارقام بالقيمة decimal.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      ul{
        font-size:25px;
        list-style-type:decimal;
      }
    </style>
  </head>
  <body>
    <ul>
      <li>HTML</li>
      <li>CSS</li>
      <li>JS</li>
    </ul>
  </body>
</html>
```

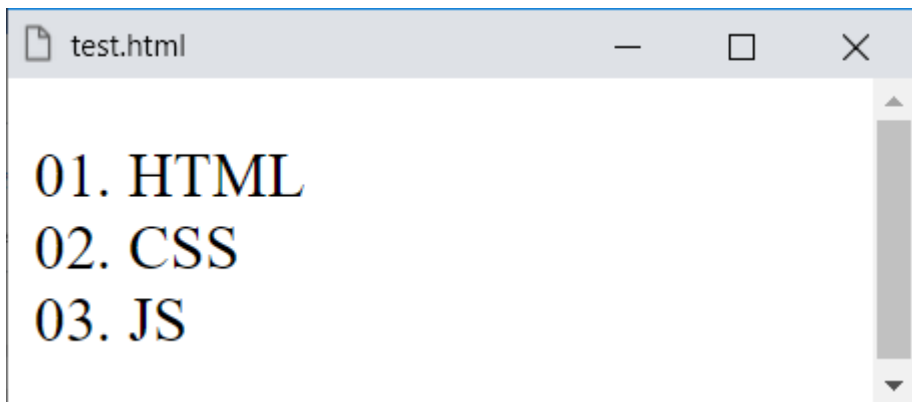
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



تحويل لأرقام مسبقة بصفر بالقيمة decimal-leading-zero.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
ul{
font-size:25px;
list-style-type:decimal-leading-zero;
}
</style>
</head>
<body>
<ul>
<li>HTML</li>
<li>CSS</li>
<li>JS</li>
</ul>
</body>
</html>
```

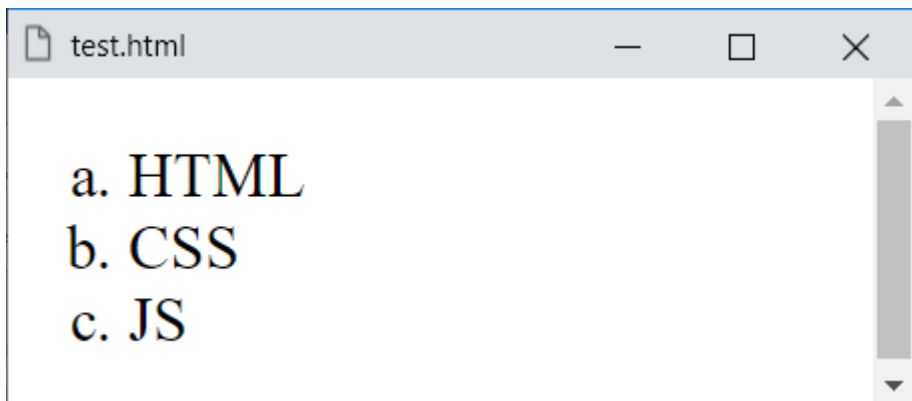
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



تحويل الى ابجدية انجليزية صغيرة بالقيمة lower-alpha.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      ul{
        font-size:25px;
        list-style-type:lower-alpha;
      }
    </style>
  </head>
  <body>
    <ul>
      <li>HTML</li>
      <li>CSS</li>
      <li>JS</li>
    </ul>
  </body>
</html>
```

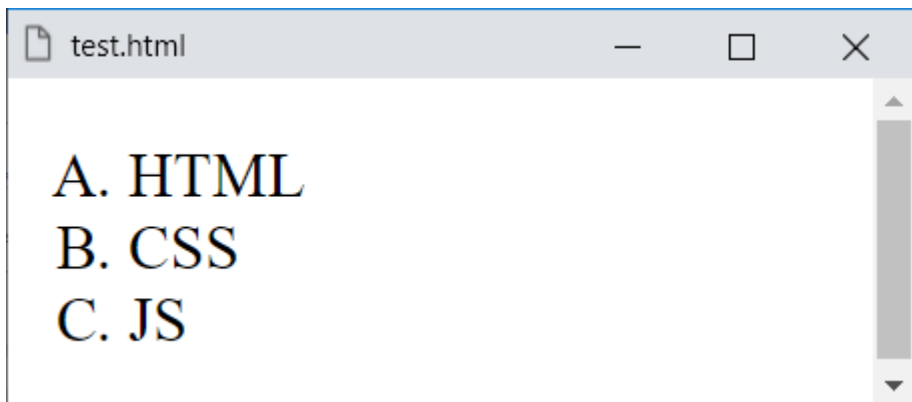
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



التحويل الى ابجدية انجليزية كبيرة بالقيمة upper-alpha.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
ul{
font-size:25px;
list-style-type:upper-alpha;
}
</style>
</head>
<body>
<ul>
<li>HTML</li>
<li>CSS</li>
<li>JS</li>
</ul>
</body>
</html>
```

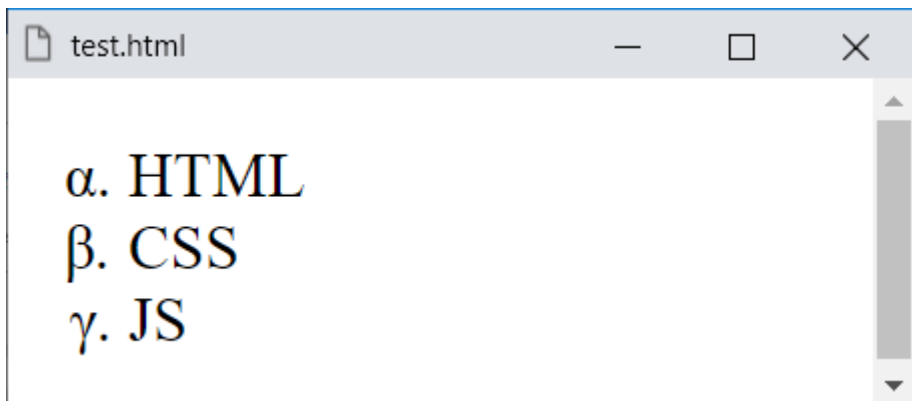
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



ترقيم اغريقي او يوناني صغيد بالقيمة lower-greek.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      ul{
        font-size:25px;
        list-style-type:lower-greek;
      }
    </style>
  </head>
  <body>
    <ul>
      <li>HTML</li>
      <li>CSS</li>
      <li>JS</li>
    </ul>
  </body>
</html>
```

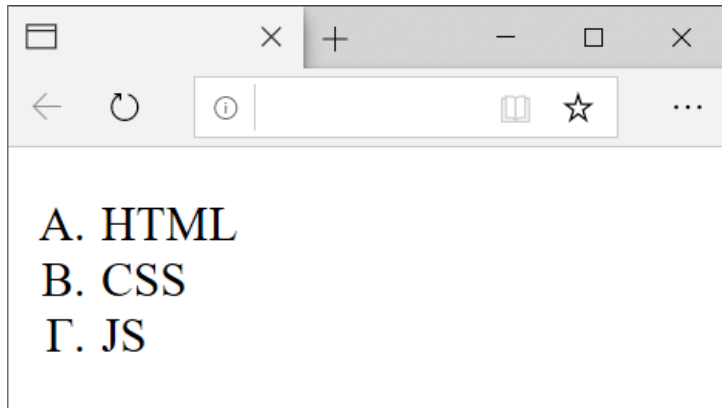
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



ترقيم اغريقي او يوناني كبير بالقيمة upper-greek.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
ul{
font-size:25px;
list-style-type:upper-greek;
}
</style>
</head>
<body>
<ul>
<li>HTML</li>
<li>CSS</li>
<li>JS</li>
</ul>
</body>
</html>
```

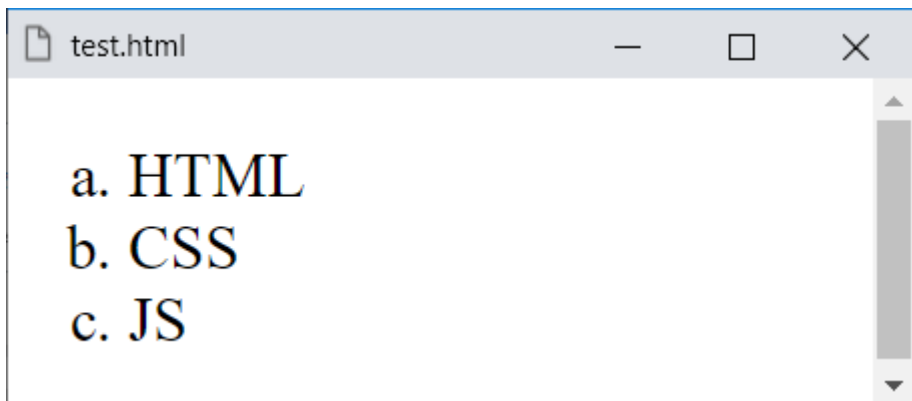
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



تحويل الى لاتيني صغير بالقيمة `.lower-latin`
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      ul{
        font-size:25px;
        list-style-type:lower-latin;
      }
    </style>
  </head>
  <body>
    <ul>
      <li>HTML</li>
      <li>CSS</li>
      <li>JS</li>
    </ul>
  </body>
</html>
```

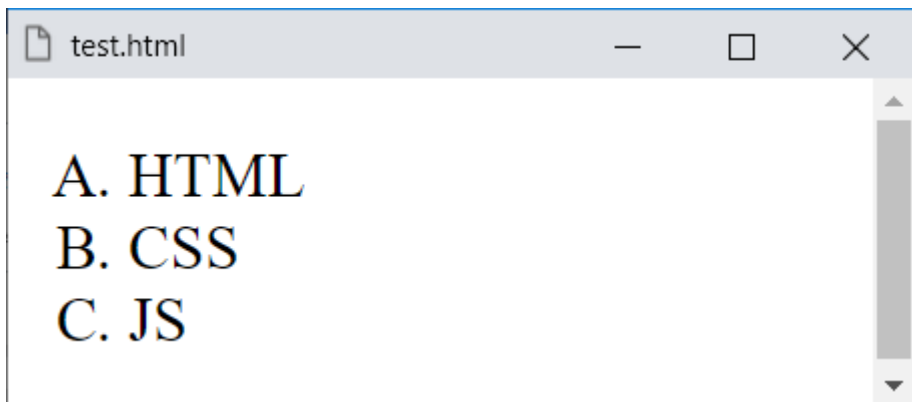
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



تحويل الى لاتيني كبير بالقيمة .upper-latin
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  ul{
    font-size:25px;
    list-style-type:upper-latin;
  }
</style>
</head>
<body>
<ul>
<li>HTML</li>
<li>CSS</li>
<li>JS</li>
</ul>
</body>
</html>
```

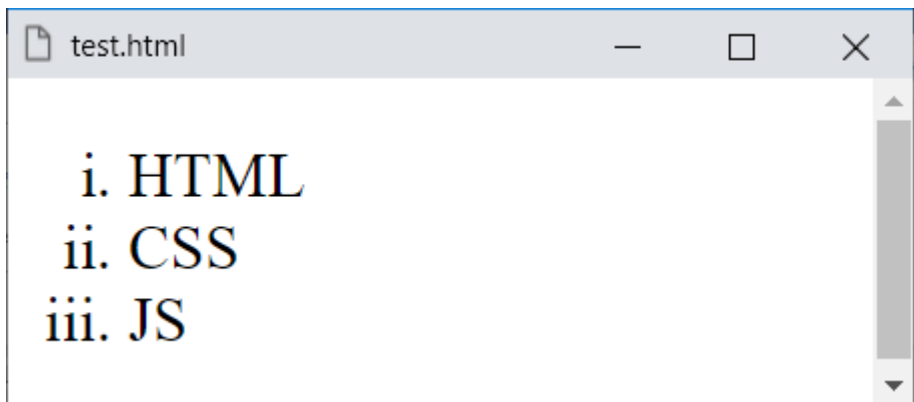
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



تحويل الى روماني صغير بالقيمة .lower-roman
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      ul{
        font-size:25px;
        list-style-type:lower-roman;
      }
    </style>
  </head>
  <body>
    <ul>
      <li>HTML</li>
      <li>CSS</li>
      <li>JS</li>
    </ul>
  </body>
</html>
```

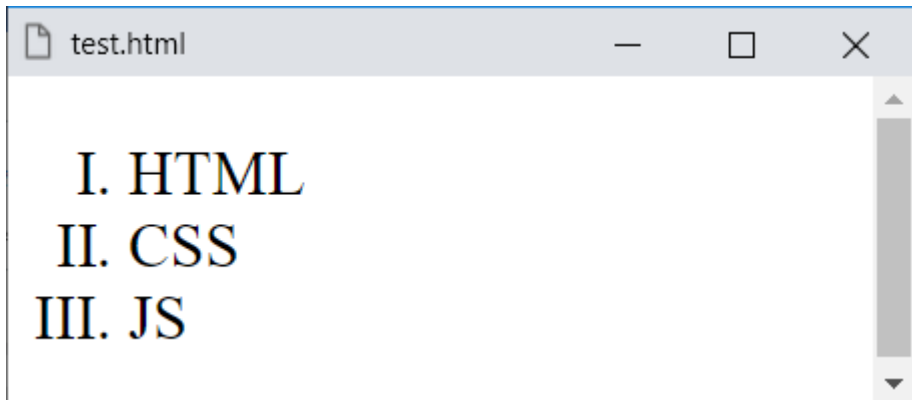
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:




تحويل الى روماني كبير بالقيمة upper-roman.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  ul{
    font-size:25px;
    list-style-type:upper-roman;
  }
</style>
</head>
<body>
<ul>
<li>HTML</li>
<li>CSS</li>
<li>JS</li>
</ul>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



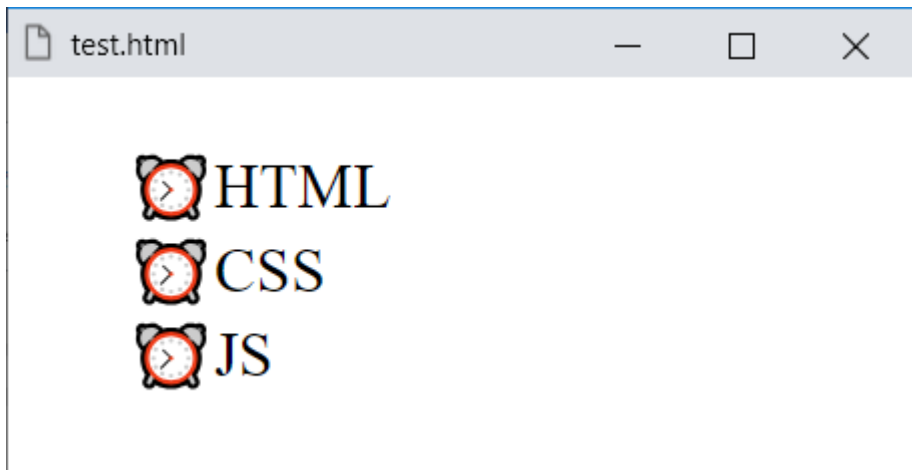
انتبه! جيداً ولاحظ أن التمارين السابقة كانت على الوسم ul بشكل متعمد، وذلك لئتم الإثبات لك أن لغة CSS قوية مع لغة html، حيث تغير أصلها، ولو استخدمت الوسم ol وغيرته الى disc سينغير معك بشكل طبيعي، وكأنه ul، ولكنه ما سبق للتدريب، أما بالنسبة لك، فننصحك أن تستخدم كل شيء في موضعه، فإذا استخدمت نقاط عليك بالوسم ul وإذا استخدمت عناصر ترتيب عليك بالوسم ol، كما تعلمت في كتاب html "صخرة الأساس" من قبل.

يمكك أيضاً انشاء قائمة برمز مخصص، وسبق وأن تحدثنا مع الرموز باستفاضة في كتاب "صخرة الأساس" للغة html، كما يوجد رقم hex لكل رمز حرفي مثل ♥، أو ☺، أو ♣، وهكذا، فإنه وبفضل الله يمكك انشاء ترقيم باستخدام رمز مع هذه الرموز كما نريد، وهنا سوف نستخدم رمز المنبه ، وهو بكون hexadecimal بالرقم "23F0"، ولاستخدام هذا الرقم السادس عشر في الخاصية content في لغة ال CSS يتم وضع باق سلاش قبل الرقم السادس عشر، فيكون "23F0\" داخل القيمة التي سوف نعطيها للخاصية content أثناء الاستخدام، وسوف يكون المثال بفضل الله كما يلي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    ul{ list-style-type:none; font-size:25px; }
    ul li::before{ content:'\23F0' }
  </style>
</head>
<body>
  <ul>
    <li>HTML</li>
    <li>CSS</li>
    <li>JS</li>
  </ul>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

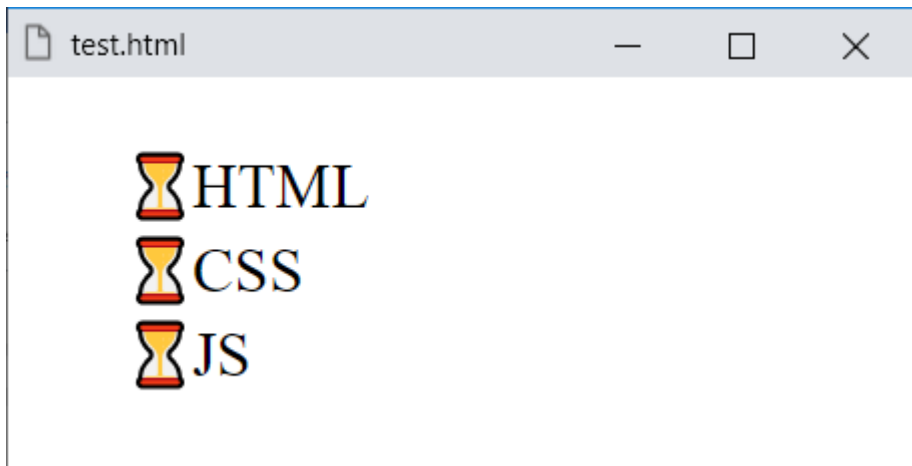


لاحظ استخدام المحدد before للوسوم li فقط بداخل الوسم ul.

وبنفس الطريقة، يمكن استخدام رمز الساعة الرملية، كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      ul{ list-style-type:none; font-size:25px; }
      ul li::before{ content:'\23F3' }
    </style>
  </head>
  <body>
    <ul>
      <li>HTML</li>
      <li>CSS</li>
      <li>JS</li>
    </ul>
  </body>
</html>
```

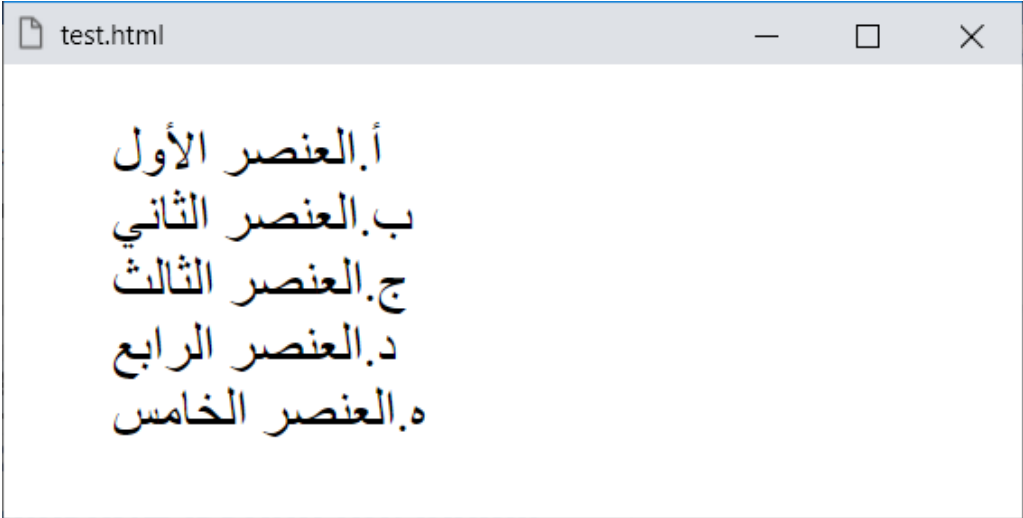
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



ويمكن إنشاء ترميز باللغة العربية لخمسة عناصر كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
ul{ list-style-type:none; font-size:25px; }
ul li:nth-child(1)::before{ content:'\0623.'; }
ul li:nth-child(2)::before{ content:'\0628.'; }
ul li:nth-child(3)::before{ content:'\062C.'; }
ul li:nth-child(4)::before{ content:'\062F.'; }
ul li:nth-child(5)::before{ content:'\0647.'; }
</style>
</head>
<body>
<ul>
<li>العنصر الأول</li>
<li>العنصر الثاني</li>
<li>العنصر الثالث</li>
<li>العنصر الرابع</li>
<li>العنصر الخامس</li>
</ul>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



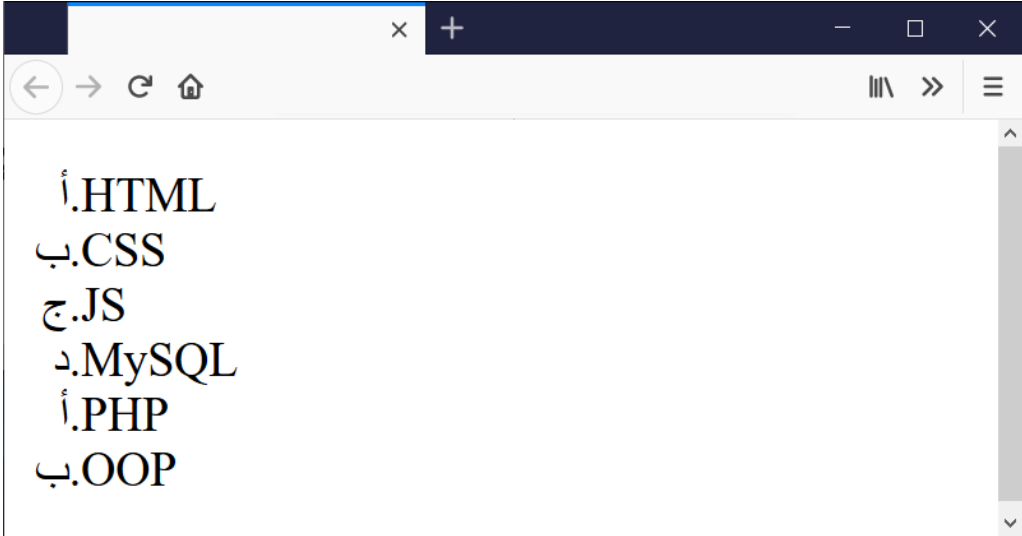
لاحظ أنه تم تجميع هذا الكود بناء على خبراتك السابقة مع هذا الكتاب مع أوله بفضل الله، **فأولاً** الرقم 0623 للحرف "أ" والرقم 0628 للحرف "ب" والرقم 062C للحرف "ج" والرقم 062F للحرف "د" والرقم 0647 للحرف "ه"، **وثانياً** نقصد رقم العنصر li بشرط يكون بداخل العنصر ul، ثم نستخدم المحدد before لوضع الحرف قبله، ثم بعد الحرف نقطة "."، **وثالثاً** يمكن أن تزيد عن خمسة عناصر بنفس الطريقة ان شئت.

انتبه! فإنه يوجد طريقة أخرى أكد احترافية، وإمكانية تجربتها رجاء تشغيل الصفحة على المتصفح Firefox، وهي كما يلي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
@counter-style arabic{
system:cyclic;
symbols: '\0623' '\0628' '\062C' '\062F';
suffix: ".";
}
ul {
list-style-type: arabic;
font-size:30px;
}
</style>
</head>
<body>
<ul>
<li>HTML</li>
<li>CSS</li>
<li>JS</li>
<li>MySQL</li>
<li>PHP</li>
<li>OOP</li>
</ul>
</body>
</html>
```

وبعد تشغيل الصفحة على Firefox تظهر كما بالشكل التالي:



لاحظ أن المحدد @counter-style لصناعة الترتيب، ولاحظ أنه تم إنشاء ترتيب باسم arabic والذي تم النداء عليه للوسم ul بعد ذلك، ولاحظ أنه في الخاصية symbols تم وضع نفس الرموز التي تم استخدامها للحروف العربية بدلاً من الحروف نفسها، وبضبط system:cyclic فهو يعيد الترتيب من جديد إذا وصل إلى نهاية الرموز الموضوعة وتم وضع عناصر زائدة لظهور مثل هذا، والخاصية suffix هي الفاصل بين الترتيب والمحتوي li.

صورة القائمة list-style-image

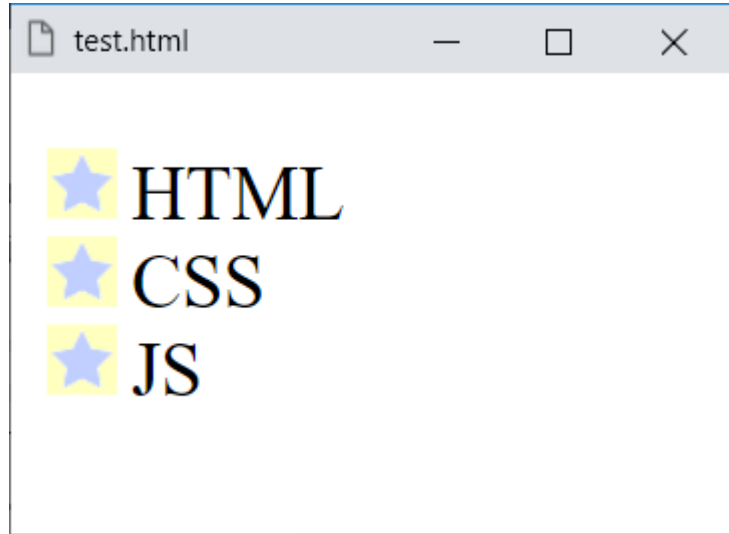
هو موضوع شيق في القوائم، حيث إنك تستخدم صورة للقائمة بدلاً من الحروف المعروفة، وحتى الغيد معروفة، ولكن سنحتاج صورة لنجرب بها هذا الموضوع، والآن سوف نقوم بتحميل صورة للإطار موجودة على موقع mdn.mozillademos.org مع الرابط التالي لنستخدمها:

<https://mdn.mozillademos.org/files/11981/starsolid.gif>

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      ul {
        list-style-image:url('starsolid.gif');
        font-size:30px;
      }
    </style>
  </head>
  <body>
    <ul>
      <li>HTML</li>
      <li>CSS</li>
      <li>JS</li>
    </ul>
  </body>
</html>
```


وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



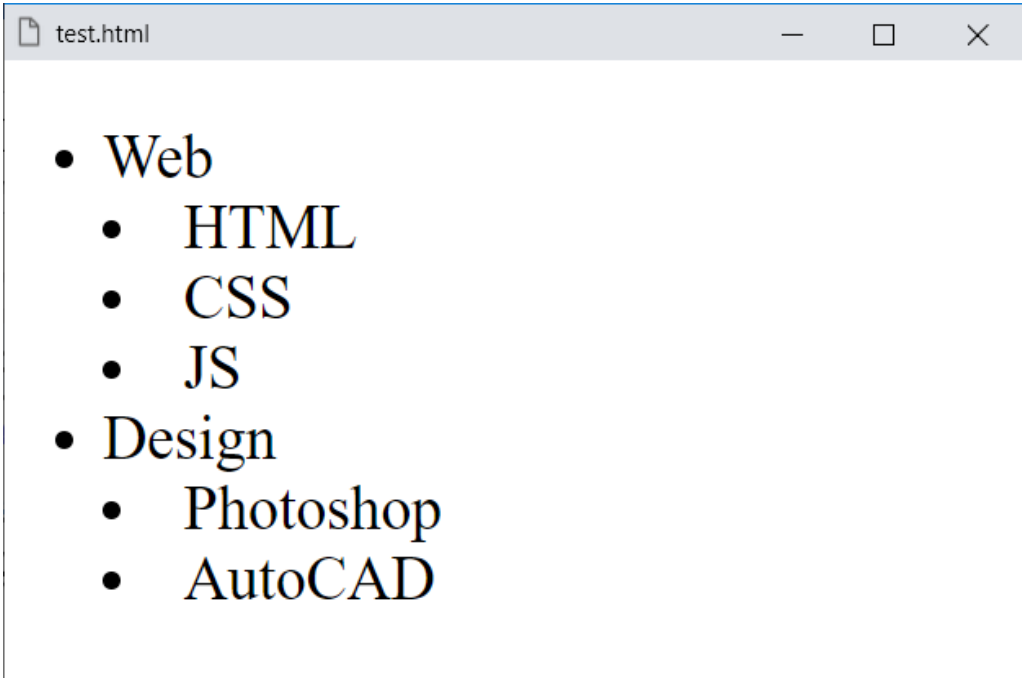
لاحظ أن نقاط القائمة ul تحولت إلى شكل الصورة، عن طريق ضبط الخاصية `list-style-image` على مكان ملف الصورة بـ `url`.

موضوع القائمة list-style-position

هو موضوع عناصر القائمة، ويمكّن توضيحه بالمثال التالي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    ul{ font-size:30px; }
    ul li.parent{ list-style-position:outside; }
    ul li.child{ list-style-position:inside; }
  </style>
</head>
<body>
  <ul>
    <li class="parent">Web</li>
    <li class="child">HTML</li>
    <li class="child">CSS</li>
    <li class="child">JS</li>
    <li class="parent">Design</li>
    <li class="child">Photoshop</li>
    <li class="child">AutoCAD</li>
  </ul>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



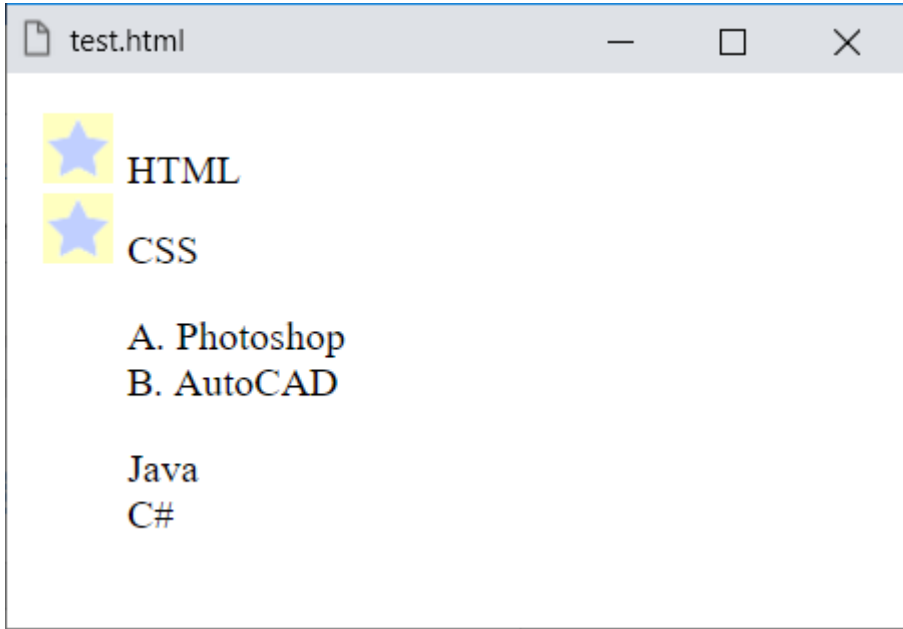
لاحظ عمل كلاس باسم parent للوسوم li الموجودة داخل ul، وضبط خاصية list-style-position له على القيمة outside وهي الافتراضية، وتم عمل كلاس باسم child للوسوم li بداخل ul، وضبط خاصية list-style-position له على القيمة inside وهي تجعل العناصر تكون بالداخل قليلاً عن الطبيعي، كما هو واضح في الصورة السابقة.

خاصية القائمة list-style

كما تعودنا من قبل من خصائص الـ CSS، وهي خاصية عامة تشمل القوائم، وسوف نبيه إن شاء الله فيما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      ul{ list-style:url('starsolid.gif'); }
      ol{ list-style:upper-alpha inside; }
      ul.lang{ list-style:none; }
    </style>
  </head>
  <body>
    <ul>
      <li>HTML</li><li>CSS</li>
    </ul>
    <ol>
      <li>Photoshop</li><li>AutoCAD</li>
    </ol>
    <ul class="lang">
      <li>Java</li><li>C#</li>
    </ul>
  </body>
</html>
```

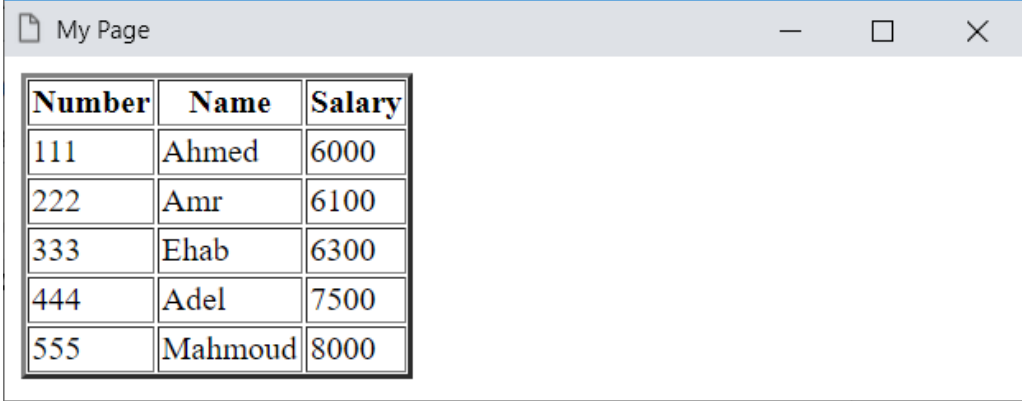
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن الخاصية `list-style` يمكن إعطاؤها قيمة `none`، أو قيمة `inside`، أو قيمة `upper-alpha`، أو قيمة رابط صورة، وذلك ببساطة لأنها خاصة عامة للقوائم.

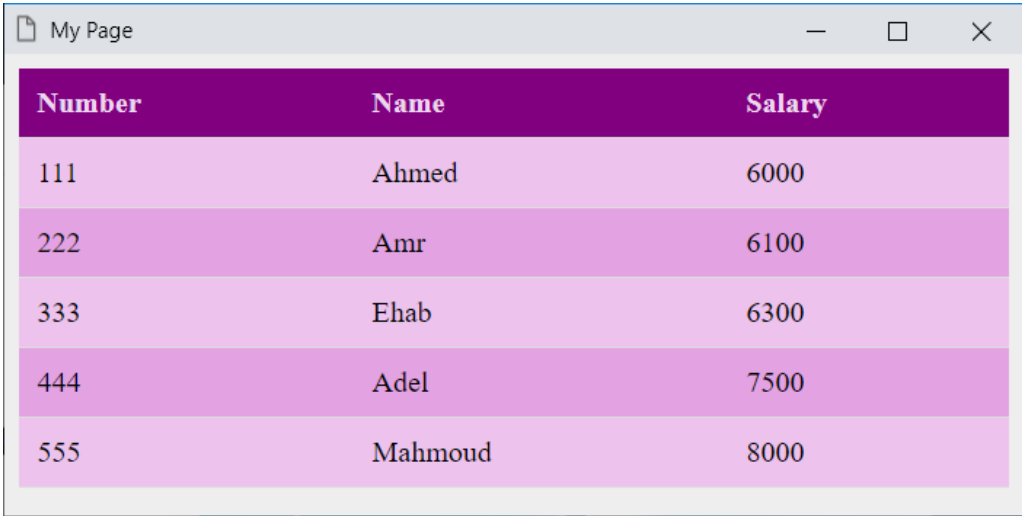
الجدول Tables

والآن نريد أن نغير مه شكل الجدول التقليدي، والذي كما بالشكل التالي:



Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Ehab	6300
444	Adel	7500
555	Mahmoud	8000

ونصل الى ان نجعل الجدول مثل الشكل التالي، أو ما يشابهه:



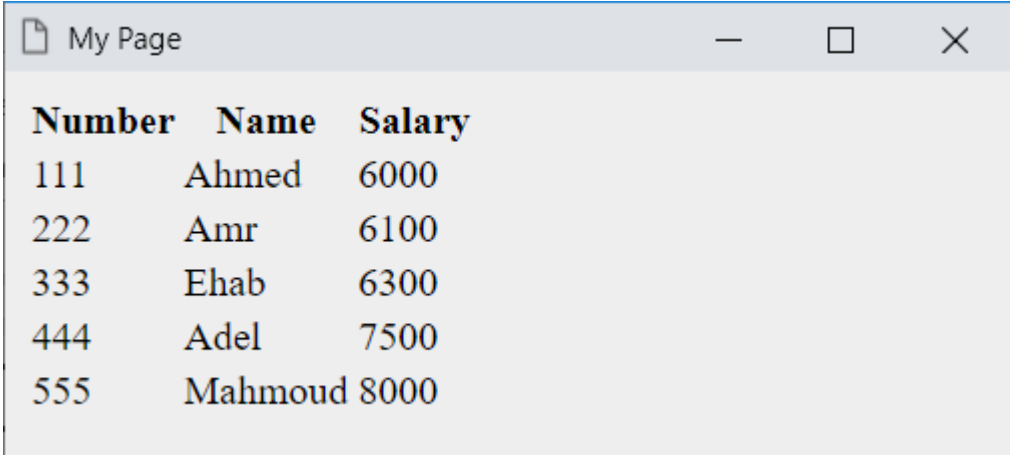
Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Ehab	6300
444	Adel	7500
555	Mahmoud	8000

لاحظ أن لغة الـ CSS و صنعت تنسيق جذاب وجميل للجدول، بفضل الله، ونده الآن سوف نستخدم خبراتنا السابقة التي تعلمناها مه هذا الكتاب، مه أوله، وسوف نحاول أن ندرج في صناعة هذا الجدول لتنفعمه جيداً.

أولاً : أساس الجدول وبياناته وخلفية الصفحة
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
  body{ background:#eee; }
</style>
</head>
<body>
<table>
<tr><th>Number</th><th>Name</th><th>Salary</th></tr>
<tr><td>111</td><td>Ahmed</td><td>6000</td></tr>
<tr><td>222</td><td>Amr</td><td>6100</td></tr>
<tr><td>333</td><td>Ehab</td><td>6300</td></tr>
<tr><td>444</td><td>Adel</td><td>7500</td></tr>
<tr><td>555</td><td>Mahmoud</td><td>8000</td></tr>
</table>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

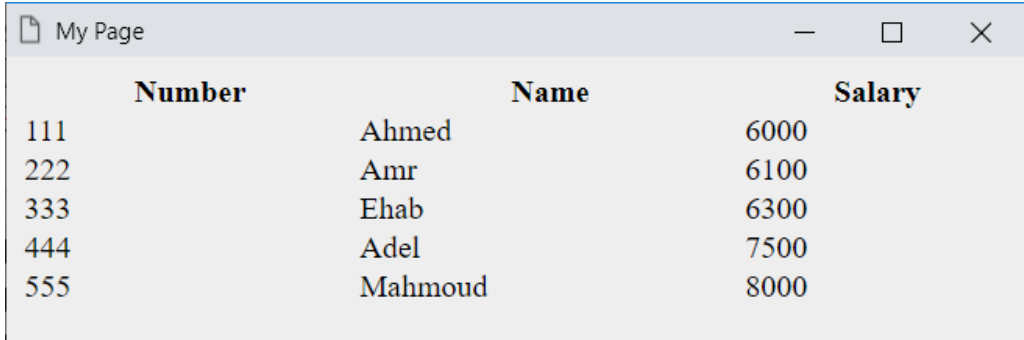


Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Ehab	6300
444	Adel	7500
555	Mahmoud	8000

ثانياً: طمس إطار الجدول وعمل العرض بالكامل
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
body{ background:#eee; }
table{ border-collapse: collapse; width:100%; }
</style>
</head>
<body>
<table>
<tr><th>Number</th><th>Name</th><th>Salary</th></tr>
<tr><td>111</td><td>Ahmed</td><td>6000</td></tr>
<tr><td>222</td><td>Amr</td><td>6100</td></tr>
<tr><td>333</td><td>Ehab</td><td>6300</td></tr>
<tr><td>444</td><td>Adel</td><td>7500</td></tr>
<tr><td>555</td><td>Mahmoud</td><td>8000</td></tr>
</table>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



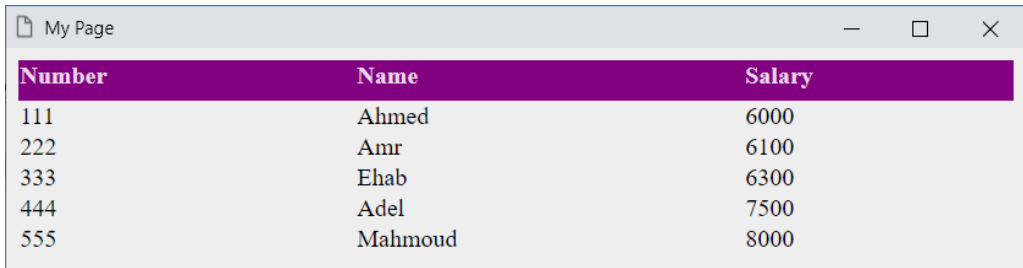
Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Ehab	6300
444	Adel	7500
555	Mahmoud	8000

طمس الإطارات بـ collapse، ولرؤيتها اضبط إطار الجدول إن شئت.

ثالثاً : ضبط عناوين الجدول
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
body{ background:#eee; }
table{ border-collapse: collapse; width:100%; }
th{
height:25px; text-align:left; vertical-align: top;
background-color: #800080; color: #edd8ed;
}
</style>
</head>
<body>
<table border="1">
<tr><th>Number</th><th>Name</th><th>Salary</th></tr>
<tr><td>111</td><td>Ahmed</td><td>6000</td></tr>
<tr><td>222</td><td>Amr</td><td>6100</td></tr>
<tr><td>333</td><td>Ehab</td><td>6300</td></tr>
<tr><td>444</td><td>Adel</td><td>7500</td></tr>
<tr><td>555</td><td>Mahmoud</td><td>8000</td></tr>
</table>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

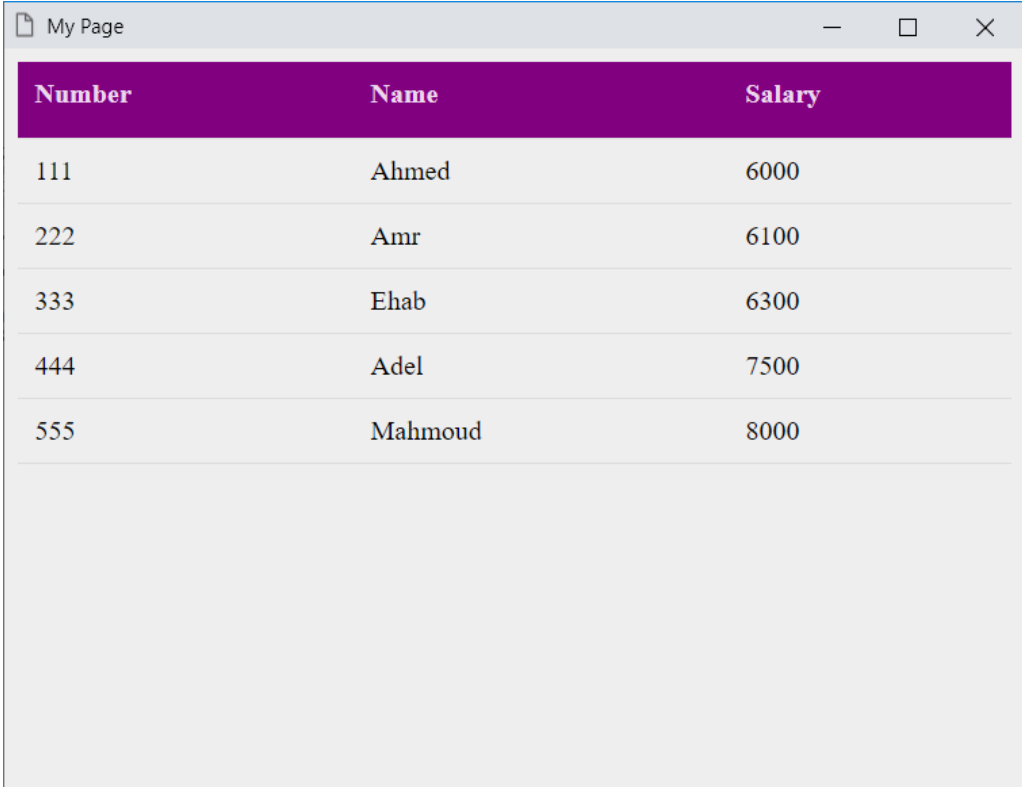


Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Ehab	6300
444	Adel	7500
555	Mahmoud	8000

رابعاً: ضبط الحشو للخلايا وعمل الإطار من الأسفل
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
body{ background:#eee; }
table{ border-collapse: collapse; width:100%; }
th{
height:25px; text-align:left; vertical-align: top;
background-color: #800080; color: #edd8ed;
}
th, td{ padding:10px; border-bottom:1px solid #ddd; }
</style>
</head>
<body>
<table>
<tr><th>Number</th><th>Name</th><th>Salary</th></tr>
<tr><td>111</td><td>Ahmed</td><td>6000</td></tr>
<tr><td>222</td><td>Amr</td><td>6100</td></tr>
<tr><td>333</td><td>Ehab</td><td>6300</td></tr>
<tr><td>444</td><td>Adel</td><td>7500</td></tr>
<tr><td>555</td><td>Mahmoud</td><td>8000</td></tr>
</table>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

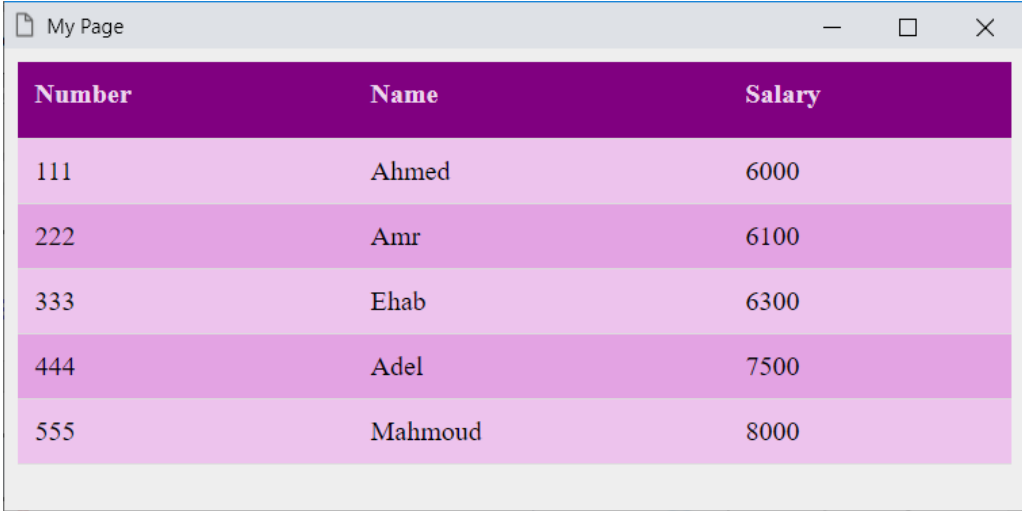


Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Ehab	6300
444	Adel	7500
555	Mahmoud	8000

خامساً: ضبط لون خلفية الصفوف الزوجية والفردية كود! قم بكتابة الكود التالي ثم احفظه:

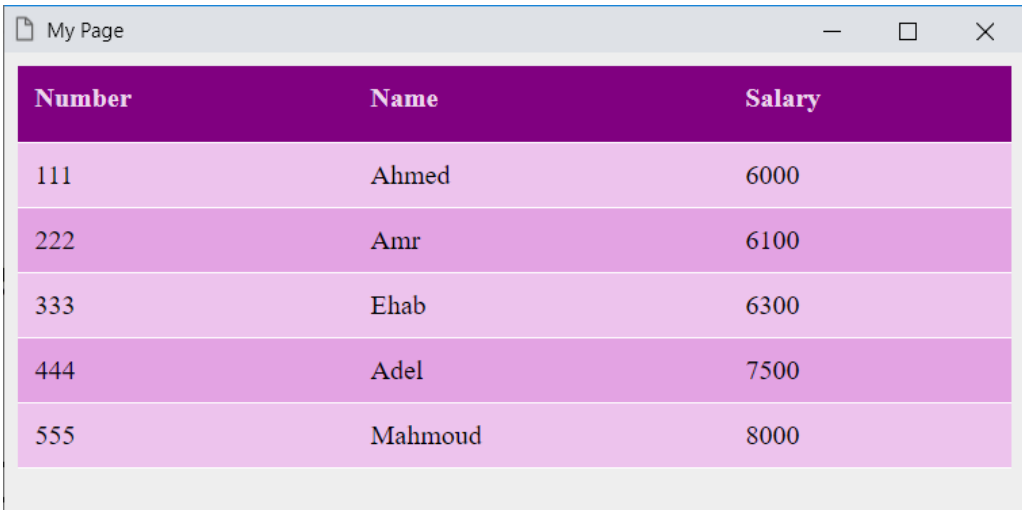
```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
body{ background:#eee; }
table{ border-collapse: collapse; width:100%; }
th{
height:25px; text-align:left; vertical-align: top;
background-color: #800080; color: #edd8ed;
}
th, td{ padding:10px; border-bottom:1px solid #ddd; }
tr:nth-child(even){ background-color: #edc3ed; }
tr:nth-child(odd){ background-color: #e3a3e3; }
</style>
</head>
<body>
<table>
<tr><th>Number</th><th>Name</th><th>Salary</th></tr>
<tr><td>111</td><td>Ahmed</td><td>6000</td></tr>
<tr><td>222</td><td>Amr</td><td>6100</td></tr>
<tr><td>333</td><td>Ehab</td><td>6300</td></tr>
<tr><td>444</td><td>Adel</td><td>7500</td></tr>
<tr><td>555</td><td>Mahmoud</td><td>8000</td></tr>
</table>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Ehab	6300
444	Adel	7500
555	Mahmoud	8000

والآن بفضل الله وصلنا الى جدول بشكل جميل، وإذا جعلنا الإطار السفلي باللون الأبيض، سيكون كما بالشكل التالي:



Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Ehab	6300
444	Adel	7500
555	Mahmoud	8000

أكواء ءطور بناء الءءول

الآن سوف نساءر أكواء CSS فقط، والءف مه ءلالها تم الءصول الء الءءول السابق بلك ءنساءقائه.
أولاً:

```
table{ border-collapse: collapse; width:100%; }
```

ثم:

```
th{  
  height:25px; text-align:left; vertical-align: top;  
  background-color: #800080; color: #edd8ed;  
}
```

ثم:

```
th, td{ padding:10px; border-bottom:1px solid #fff; }
```

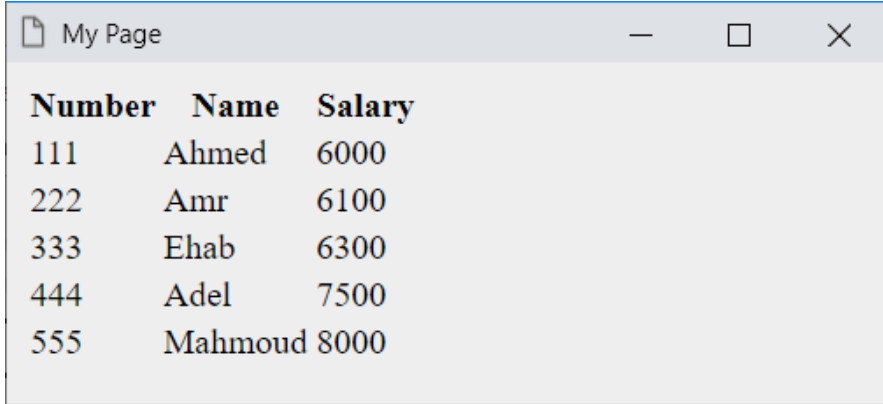
ثم:

```
tr:nth-child(even){ background-color: #edc3ed; }  
tr:nth-child(odd){ background-color: #e3a3e3; }
```

انءبه! ولاءظ أن الأكواء مع فعمها وءرءببها سءهل بفضل الله.

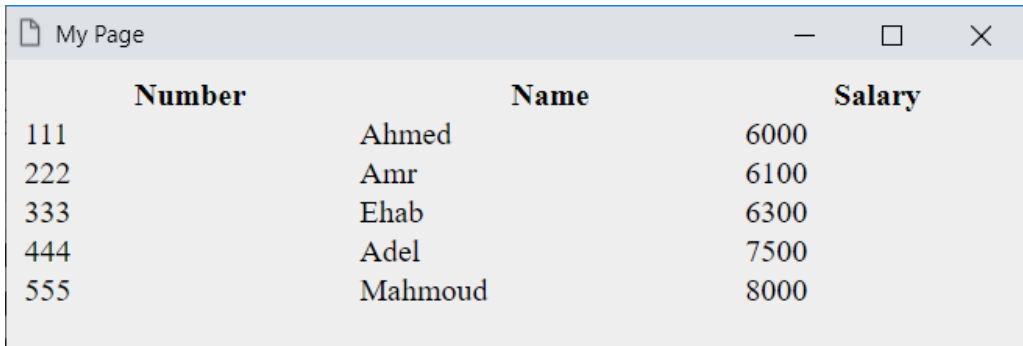
تصميم تطور بناء الجدول

الآن سوف نسرد صور فقط، والتي من خلالها تم الوصول الى الجدول السابق بكل تنسيقاته.
 أولاً :



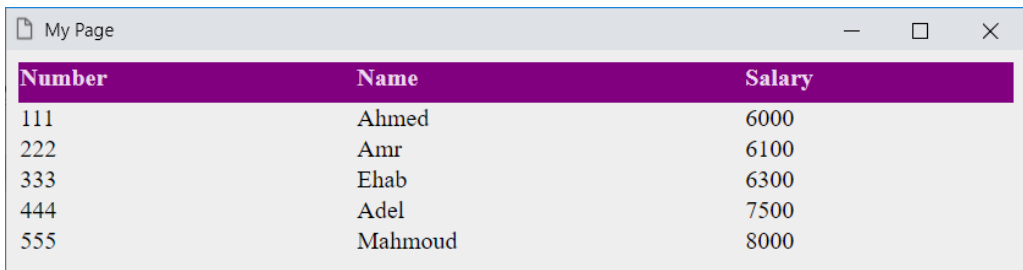
Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Ehab	6300
444	Adel	7500
555	Mahmoud	8000

ثم ^



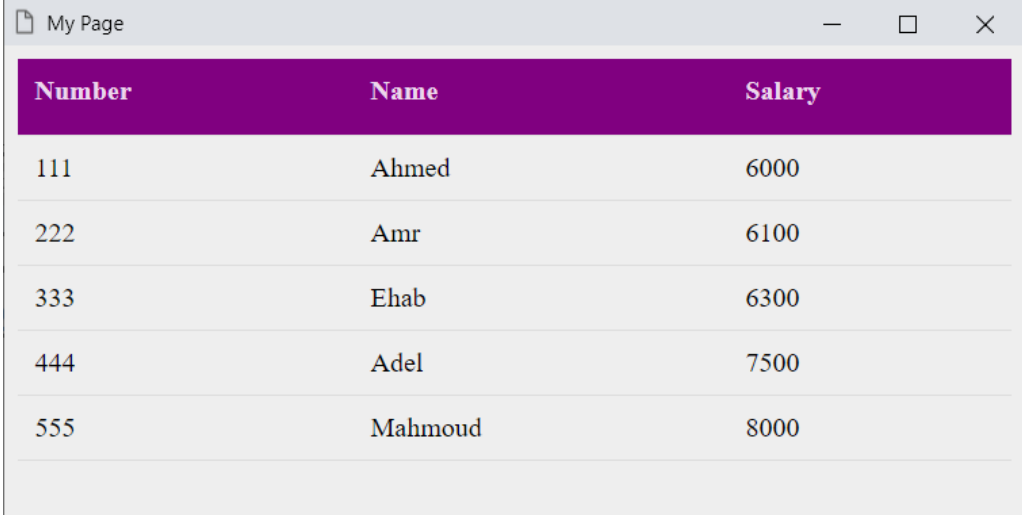
Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Ehab	6300
444	Adel	7500
555	Mahmoud	8000

ثم ^



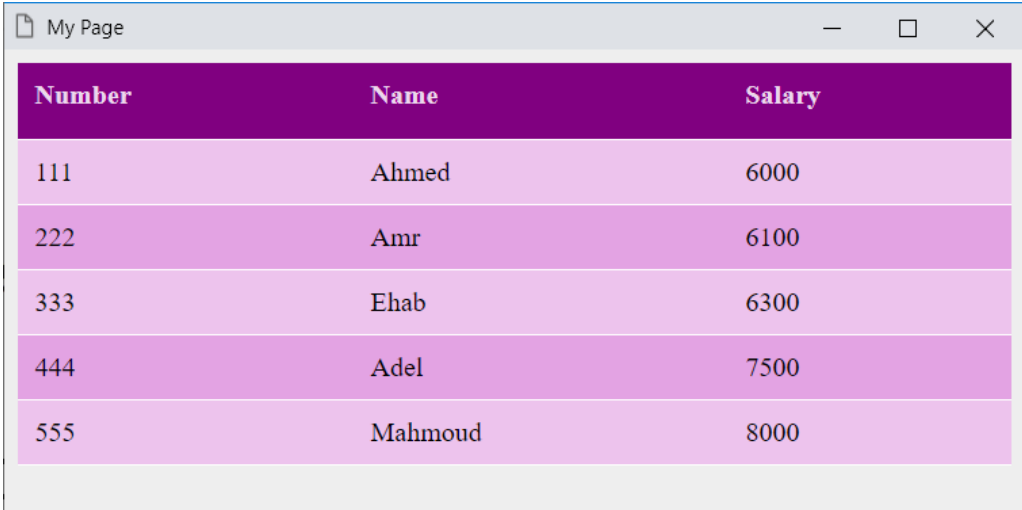
Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Ehab	6300
444	Adel	7500
555	Mahmoud	8000

نم:



Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Ehab	6300
444	Adel	7500
555	Mahmoud	8000

نم:



Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Ehab	6300
444	Adel	7500
555	Mahmoud	8000

جداول أخرى

بفضل الله سوف نبدأ تأليف تصاميم مجموعة مختلفة من الجداول، مع حيث الألوان، والنقوش، وأشياء أخرى كثيرة، لتناسب مع جميع الأغراض، وبدون شرح، لأن كل ما سنستخدمه إن شاء الله، تم شرحه فيما سبق، وبفضل الله سوف نعمل مثل التالي إن شاء الله:

Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Elab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khalid	8400

Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Elab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khalid	8400

Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Elab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khalid	8400

Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Elab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khalid	8400

Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Elab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khalid	8400

Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Elab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khalid	8400

Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Elab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khalid	8400

Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Elab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khalid	8400

Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Elab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khalid	8400

Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Elab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khalid	8400

Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Elab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khalid	8400

Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Elab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khalid	8400

Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Elab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khalid	8400

Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Elab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khalid	8400

Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Elab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khalid	8400

Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Elab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khalid	8400

Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Elab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khalid	8400

Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Elab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khalid	8400

هذه الصور صغيرة، وبالتفصيل ستتضح التفاصيل، وأول جدول تم عمله.

انتبه! وركز جيداً، لأننا سوف نستخدم أشياء متداخلة لنقوم بتلويح شيء معيه، فيجب عليك أن تركز مع كل خاصية يتم كتابتها إذا كنت تريد الفهم، وأيضاً لا تنسى أنه لم يتم الشرح كما ذكرنا من قبل.

كود! " **جدول 1** " قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
body{ background:#eee; }
table{ border-collapse: collapse; width:100%; }
th{
height:25px; text-align:left; vertical-align: top;
background-color: #00801e; color: #dbf2e1;
}
th, td{ padding:10px; border-bottom:1px solid #fff; }
tr:nth-child(even){ background-color: #96e0a7; }
tr:nth-child(odd){ background-color: #b7ebc4; }
</style>
</head>
<body>
<table>
<tr><th>Number</th><th>Name</th><th>Salary</th></tr>
<tr><td>111</td><td>Ahmed</td><td>6000</td></tr>
<tr><td>222</td><td>Amr</td><td>6100</td></tr>
<tr><td>333</td><td>Ehab</td><td>6300</td></tr>
<tr><td>444</td><td>Adel</td><td>7500</td></tr>
<tr><td>555</td><td>Mahmoud</td><td>8000</td></tr>
<tr><td>666</td><td>Eslam</td><td>9000</td></tr>
<tr><td>777</td><td>Khaled</td><td>8400</td></tr>
</table>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

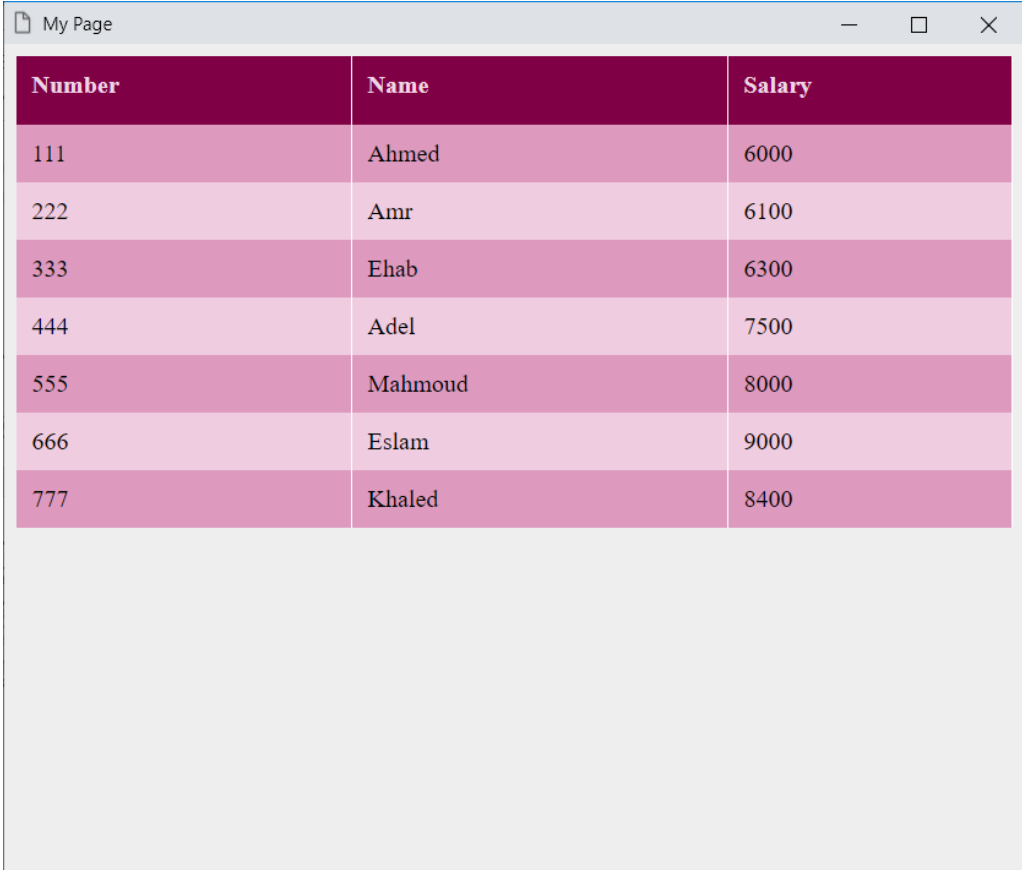


Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Ehab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khaled	8400

كود! " جدول 2 " قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
body{ background:#eee; }
table{ border-collapse: collapse; width:100%; }
th{
height:25px; text-align:left; vertical-align: top;
background-color: #800045; color: #f0d6e4;
}
th, td{ padding:10px; border-right:1px solid #fff; }
tr:nth-child(even){ background-color: #dd9abe; }
tr:nth-child(odd){ background-color: #efccdf; }
</style>
</head>
<body>
<table>
<tr><th>Number</th><th>Name</th><th>Salary</th></tr>
<tr><td>111</td><td>Ahmed</td><td>6000</td></tr>
<tr><td>222</td><td>Amr</td><td>6100</td></tr>
<tr><td>333</td><td>Ehab</td><td>6300</td></tr>
<tr><td>444</td><td>Adel</td><td>7500</td></tr>
<tr><td>555</td><td>Mahmoud</td><td>8000</td></tr>
<tr><td>666</td><td>Eslam</td><td>9000</td></tr>
<tr><td>777</td><td>Khaled</td><td>8400</td></tr>
</table>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:




Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Ehab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khaled	8400

كود! " جدول 3 " قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
body{ background:#eee; }
table{ border-collapse: collapse; width:100%; }
th{
height:25px; text-align:left; vertical-align: top;
background-color: #007b80; color: #c7edef;
}
th, td{ padding:10px; }
tr:nth-child(even){ background-color: #6bcace; }
tr:nth-child(odd){ background-color: #b0e7ea; }
</style>
</head>
<body>
<table>
<tr><th>Number</th><th>Name</th><th>Salary</th></tr>
<tr><td>111</td><td>Ahmed</td><td>6000</td></tr>
<tr><td>222</td><td>Amr</td><td>6100</td></tr>
<tr><td>333</td><td>Ehab</td><td>6300</td></tr>
<tr><td>444</td><td>Adel</td><td>7500</td></tr>
<tr><td>555</td><td>Mahmoud</td><td>8000</td></tr>
<tr><td>666</td><td>Eslam</td><td>9000</td></tr>
<tr><td>777</td><td>Khaled</td><td>8400</td></tr>
</table>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Ehab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khaled	8400

كود! " جدول 4 " قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
body{ background:#eee; }
table{ border-collapse: collapse; width:100%; }
th{
height:25px; text-align:left; vertical-align: top;
background-color: #7b8000; color: #e8eabf;
}
th, td{ padding:10px; border:1px solid #fff; }
tr:nth-child(even){ background-color: #d6d979; }
tr:nth-child(odd){ background-color: #f0f2b4; }
</style>
</head>
<body>
<table>
<tr><th>Number</th><th>Name</th><th>Salary</th></tr>
<tr><td>111</td><td>Ahmed</td><td>6000</td></tr>
<tr><td>222</td><td>Amr</td><td>6100</td></tr>
<tr><td>333</td><td>Ehab</td><td>6300</td></tr>
<tr><td>444</td><td>Adel</td><td>7500</td></tr>
<tr><td>555</td><td>Mahmoud</td><td>8000</td></tr>
<tr><td>666</td><td>Eslam</td><td>9000</td></tr>
<tr><td>777</td><td>Khaled</td><td>8400</td></tr>
</table>
</body>
</html>
```


وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

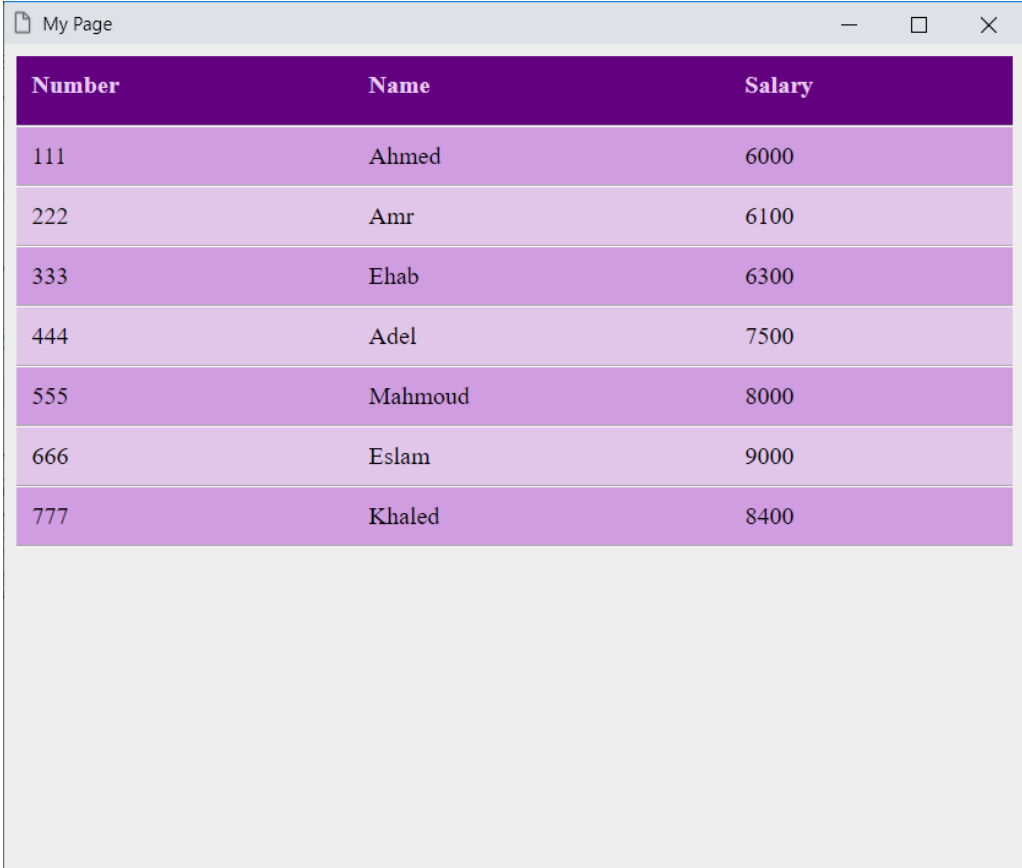


Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Ehab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khaled	8400

كود! " جدول 5 " قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
body{ background:#eee; }
table{ border-collapse: collapse; width:100%; }
th{
height:25px; text-align:left; vertical-align: top;
background-color: #620080; color: #e8cbf0;
}
th, td{ padding:10px; border-bottom:2px groove #fff; }
tr:nth-child(even){ background-color: #d09de0; }
tr:nth-child(odd){ background-color: #e1c6ea; }
</style>
</head>
<body>
<table>
<tr><th>Number</th><th>Name</th><th>Salary</th></tr>
<tr><td>111</td><td>Ahmed</td><td>6000</td></tr>
<tr><td>222</td><td>Amr</td><td>6100</td></tr>
<tr><td>333</td><td>Ehab</td><td>6300</td></tr>
<tr><td>444</td><td>Adel</td><td>7500</td></tr>
<tr><td>555</td><td>Mahmoud</td><td>8000</td></tr>
<tr><td>666</td><td>Eslam</td><td>9000</td></tr>
<tr><td>777</td><td>Khaled</td><td>8400</td></tr>
</table>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Ehab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khaled	8400

كود! " جدول 6 " قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
body{ background:#eee; }
table{ border-collapse: collapse; width:100%; }
th{
height:25px; text-align:left; vertical-align: top;
background:linear-gradient(navy, blue); color: #fff;
}
th, td{ padding:10px; border-top:3px ridge #fff; }
tr:nth-child(even){ background:linear-gradient(#7d7dd6,
#b6b6e8); }
tr:nth-child(odd){ background:linear-gradient(#b6b6e8,
#7d7dd6); }
</style>
</head>
<body>
<table>
<tr><th>Number</th><th>Name</th><th>Salary</th></tr>
<tr><td>111</td><td>Ahmed</td><td>6000</td></tr>
<tr><td>222</td><td>Amr</td><td>6100</td></tr>
<tr><td>333</td><td>Ehab</td><td>6300</td></tr>
<tr><td>444</td><td>Adel</td><td>7500</td></tr>
<tr><td>555</td><td>Mahmoud</td><td>8000</td></tr>
<tr><td>666</td><td>Eslam</td><td>9000</td></tr>
<tr><td>777</td><td>Khaled</td><td>8400</td></tr>
</table>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Ehab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khaled	8400

كود! " جدول 7 " قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
body{ background:#eee; }
table{ border-collapse: collapse; width:100%; }
th{
height:25px; text-align:left; vertical-align: top;
background:linear-gradient(to right, navy, blue); color: #fff;
}
th, td{
padding:10px; border-right:3px ridge #fff;
}
tr:nth-child(even){
background:linear-gradient(to right, #7d7dd6, #b6b6e8);
}
tr:nth-child(odd){
background:linear-gradient(to right, #b6b6e8, #7d7dd6);
}
</style>
</head>
<body>
<table>
<tr><th>Number</th><th>Name</th><th>Salary</th></tr>
<tr><td>111</td><td>Ahmed</td><td>6000</td></tr>
<tr><td>222</td><td>Amr</td><td>6100</td></tr>
<tr><td>333</td><td>Ehab</td><td>6300</td></tr>
<tr><td>444</td><td>Adel</td><td>7500</td></tr>
<tr><td>555</td><td>Mahmoud</td><td>8000</td></tr>
<tr><td>666</td><td>Eslam</td><td>9000</td></tr>
<tr><td>777</td><td>Khaled</td><td>8400</td></tr>
</table>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Ehab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khaled	8400

كود! " جدول 8 " قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
body{ background:#eee; }
table{ border-collapse: collapse; width:100%; }
th{
height:25px; text-align:left; vertical-align: top;
background:linear-gradient(darkred, red, darkred);
color: #fff;
}
th, td{
padding:10px; border-right:3px ridge #fff;
}
td:nth-child(even){
background:linear-gradient(to right, red, gold);
}
td:nth-child(odd){
background:linear-gradient(to left, red, gold);
}
</style>
</head>
<body>
<table>
<tr><th>Number</th><th>Name</th><th>Salary</th></tr>
<tr><td>111</td><td>Ahmed</td><td>6000</td></tr>
<tr><td>222</td><td>Amr</td><td>6100</td></tr>
<tr><td>333</td><td>Ehab</td><td>6300</td></tr>
<tr><td>444</td><td>Adel</td><td>7500</td></tr>
<tr><td>555</td><td>Mahmoud</td><td>8000</td></tr>
<tr><td>666</td><td>Eslam</td><td>9000</td></tr>
<tr><td>777</td><td>Khaled</td><td>8400</td></tr>
</table>
</body>
</html>
```


وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Ehab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khaled	8400

كود! " جدول 9 " قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
body{ background:#eee; }
table{ border-collapse: collapse; width:100%; }
th{
height:25px; text-align:left; vertical-align: top;
background:linear-gradient(navy, blue, navy); color: #fff;
}
th, td{ padding:10px; }
th:not(:last-child), td:not(:last-child){
border-right:3px ridge #fff;
}
td:nth-child(even){
background:linear-gradient(to right, #7d7dd6, #b6b6e8);
}
td:nth-child(odd){
background:linear-gradient(to left, #7d7dd6, #b6b6e8);
}
</style>
</head>
<body>
<table>
<tr><th>Number</th><th>Name</th><th>Salary</th></tr>
<tr><td>111</td><td>Ahmed</td><td>6000</td></tr>
<tr><td>222</td><td>Amr</td><td>6100</td></tr>
<tr><td>333</td><td>Ehab</td><td>6300</td></tr>
<tr><td>444</td><td>Adel</td><td>7500</td></tr>
<tr><td>555</td><td>Mahmoud</td><td>8000</td></tr>
<tr><td>666</td><td>Eslam</td><td>9000</td></tr>
<tr><td>777</td><td>Khaled</td><td>8400</td></tr>
</table>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Ehab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khaled	8400

كود! " جدول 10 " قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
body{ background:#eee; }
table{ border-collapse: collapse; width:100%; }
th{
height:25px; text-align:left; vertical-align: top;
background:radial-gradient(blue, navy); color: #fff;
}
th, td{ padding:10px; }
th:not(:last-child), td:not(:last-child){
border-right:3px ridge #fff;
}
td:nth-child(even){
background:radial-gradient(#7d7dd6, #b6b6e8);
}
td:nth-child(odd){
background:radial-gradient(#7d7dd6, #b6b6e8);
}
</style>
</head>
<body>
<table>
<tr><th>Number</th><th>Name</th><th>Salary</th></tr>
<tr><td>111</td><td>Ahmed</td><td>6000</td></tr>
<tr><td>222</td><td>Amr</td><td>6100</td></tr>
<tr><td>333</td><td>Ehab</td><td>6300</td></tr>
<tr><td>444</td><td>Adel</td><td>7500</td></tr>
<tr><td>555</td><td>Mahmoud</td><td>8000</td></tr>
<tr><td>666</td><td>Eslam</td><td>9000</td></tr>
<tr><td>777</td><td>Khaled</td><td>8400</td></tr>
</table>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

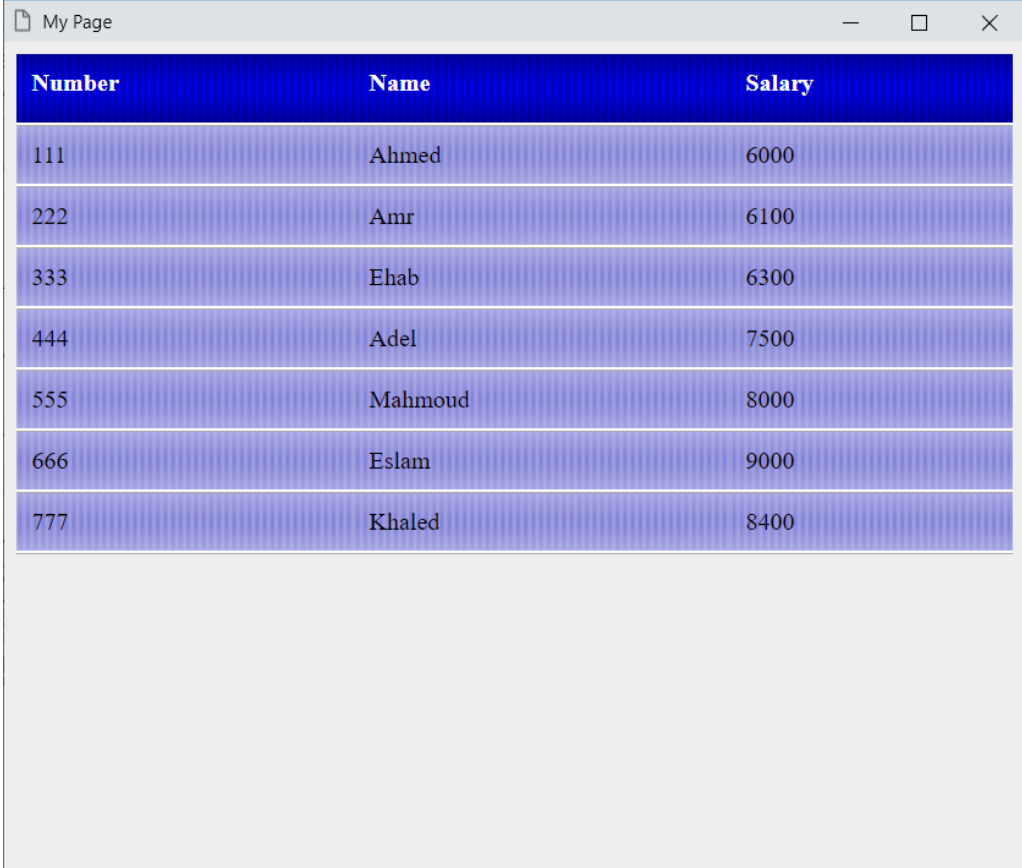


Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Ehab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khaled	8400

كود! " جدول 11 " قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
body{ background:#eee; }
table{ border-collapse: collapse; width:100%; }
th{
height:25px; text-align:left; vertical-align: top;
background:radial-gradient(blue, navy) left/5px;
color: #fff;
}
th, td{
padding:10px; border-bottom:3px ridge #fff;
}
td:nth-child(even){
background:radial-gradient(#7d7dd6, #b6b6e8) left/5px;
}
td:nth-child(odd){
background:radial-gradient(#7d7dd6, #b6b6e8) left/5px;
}
</style>
</head>
<body>
<table>
<tr><th>Number</th><th>Name</th><th>Salary</th></tr>
<tr><td>111</td><td>Ahmed</td><td>6000</td></tr>
<tr><td>222</td><td>Amr</td><td>6100</td></tr>
<tr><td>333</td><td>Ehab</td><td>6300</td></tr>
<tr><td>444</td><td>Adel</td><td>7500</td></tr>
<tr><td>555</td><td>Mahmoud</td><td>8000</td></tr>
<tr><td>666</td><td>Eslam</td><td>9000</td></tr>
<tr><td>777</td><td>Khaled</td><td>8400</td></tr>
</table>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

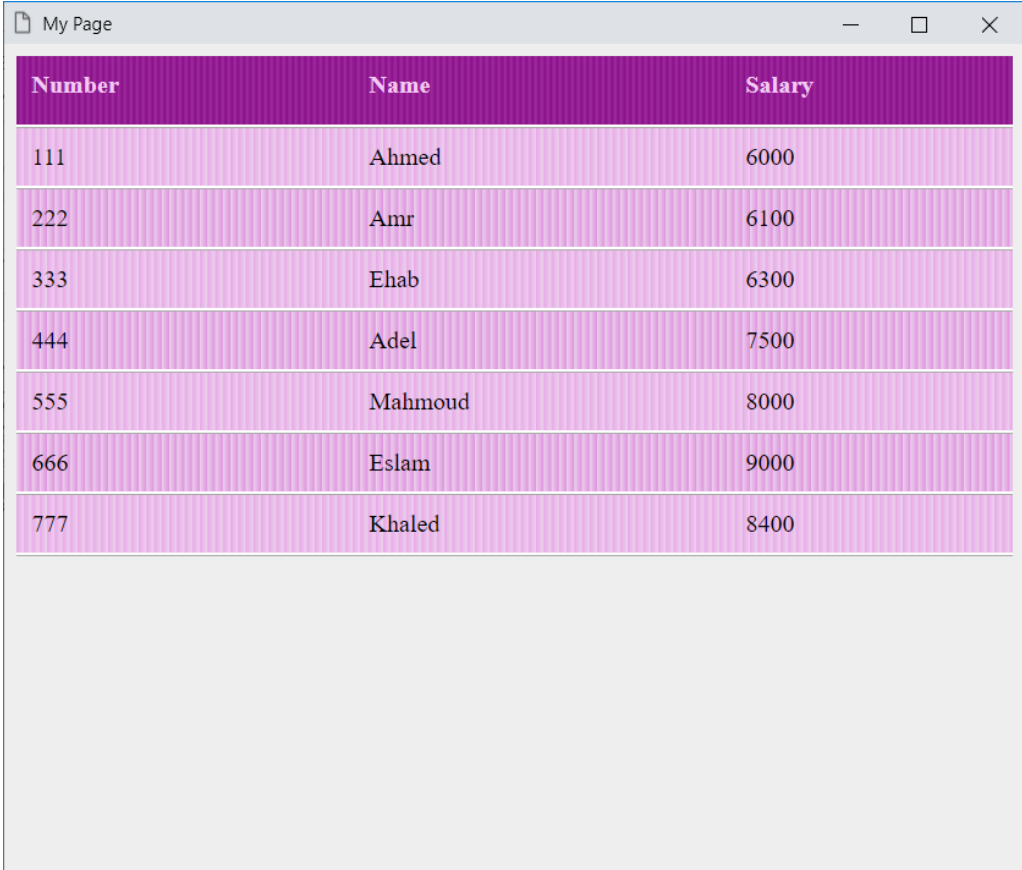


Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Ehab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khaled	8400

كود! " جدول 12 " قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
body{ background:#eee; }
table{ border-collapse: collapse; width:100%; }
th{
height:25px; text-align:left; vertical-align: top;
background:linear-gradient(to right,#800080, #a83ca8) left/1px;
color: #f4caf4;
}
th, td{
padding:10px; border-bottom:3px ridge #fff;
}
tr:nth-child(even){
background:linear-gradient(to right, #e8a8e8, #edd2ed) left/5px;
}
tr:nth-child(odd){
background:linear-gradient(to left, #de93de, #edd2ed) left/5px;
}
</style>
</head>
<body>
<table>
<tr><th>Number</th><th>Name</th><th>Salary</th></tr>
<tr><td>111</td><td>Ahmed</td><td>6000</td></tr>
<tr><td>222</td><td>Amr</td><td>6100</td></tr>
<tr><td>333</td><td>Ehab</td><td>6300</td></tr>
<tr><td>444</td><td>Adel</td><td>7500</td></tr>
<tr><td>555</td><td>Mahmoud</td><td>8000</td></tr>
<tr><td>666</td><td>Eslam</td><td>9000</td></tr>
<tr><td>777</td><td>Khaled</td><td>8400</td></tr>
</table>
</body>
</html>
```


وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

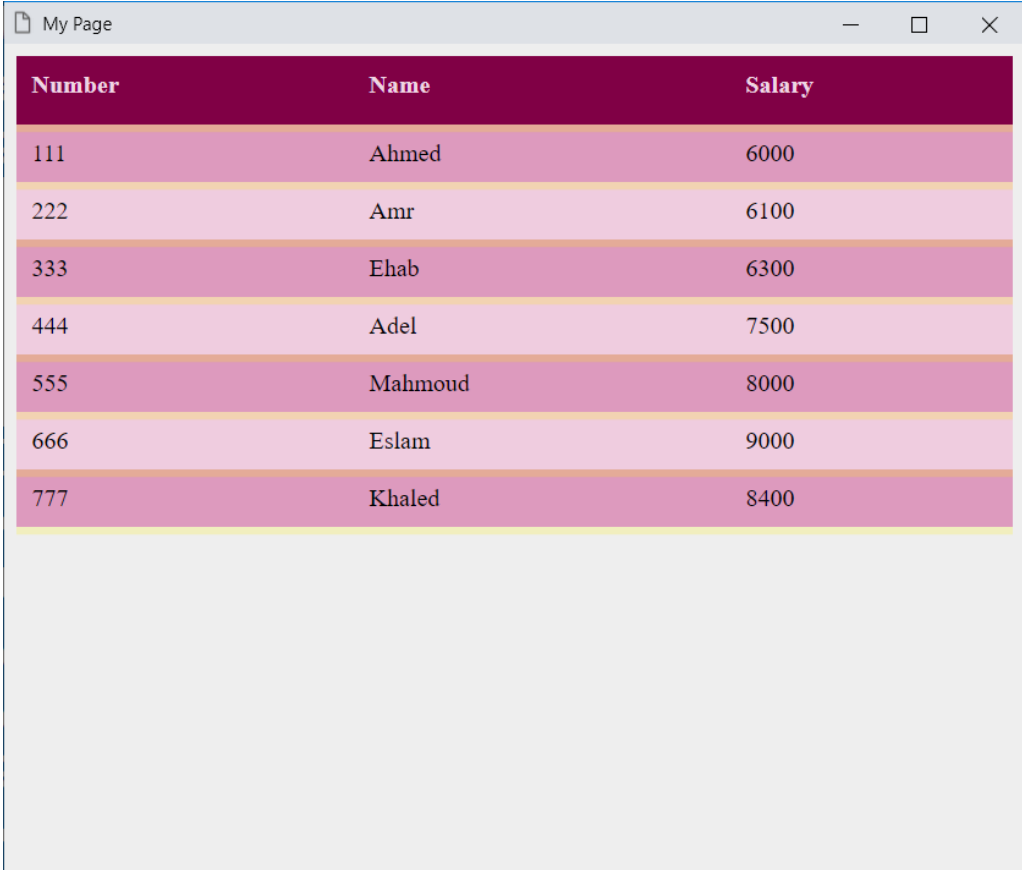


Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Ehab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khaled	8400

كود! " جدول 13 " قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
body{ background:#eee; }
table{ border-collapse: collapse; width:100%; }
th{
height:25px; text-align:left; vertical-align: top;
background-color: #800045; color: #f0d6e4;
}
th, td{ padding:10px; }
tr:nth-child(even){ background-color: #dd9abe; }
tr:nth-child(odd){ background-color: #efccdf; }
td, th{ box-shadow:0 5px rgba(255,242,0,0.2); }
</style>
</head>
<body>
<table>
<tr><th>Number</th><th>Name</th><th>Salary</th></tr>
<tr><td>111</td><td>Ahmed</td><td>6000</td></tr>
<tr><td>222</td><td>Amr</td><td>6100</td></tr>
<tr><td>333</td><td>Ehab</td><td>6300</td></tr>
<tr><td>444</td><td>Adel</td><td>7500</td></tr>
<tr><td>555</td><td>Mahmoud</td><td>8000</td></tr>
<tr><td>666</td><td>Eslam</td><td>9000</td></tr>
<tr><td>777</td><td>Khaled</td><td>8400</td></tr>
</table>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

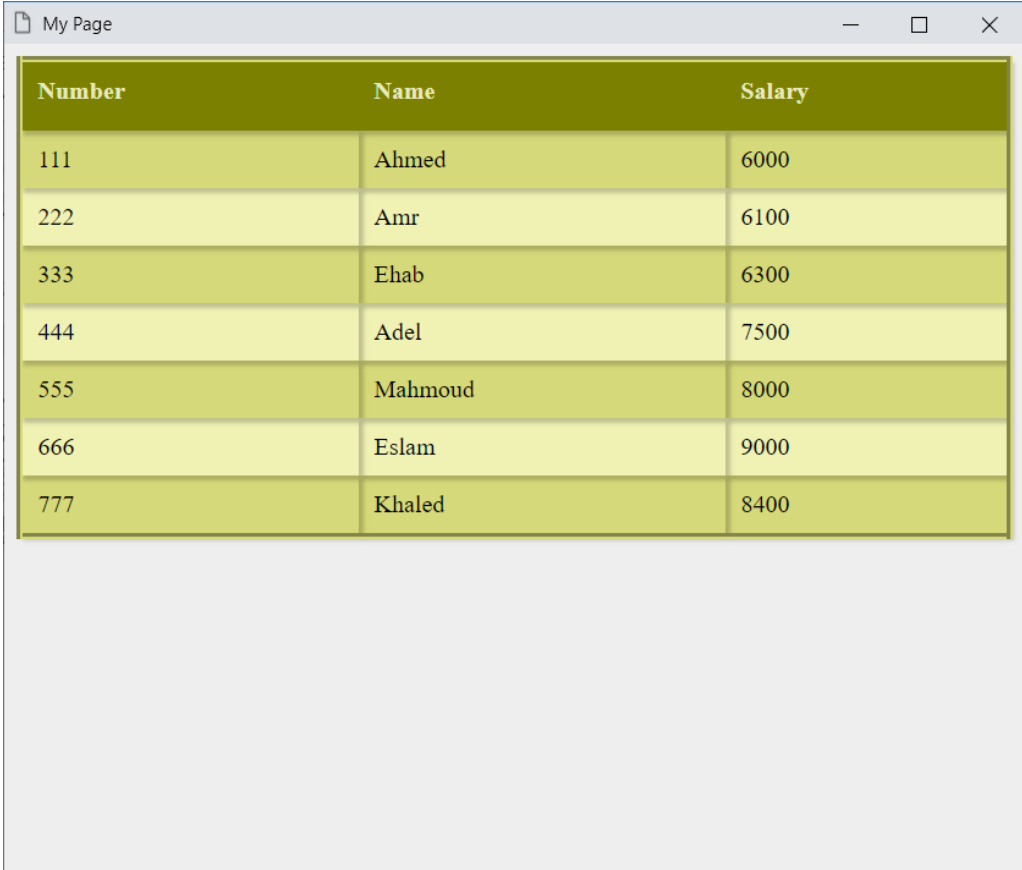


Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Ehab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khaled	8400

كود! " جدول 14 " قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
body{ background:#eee; }
table{
border-collapse:collapse; width:100%;
border:4px groove #d6d979;
}
th{
height:25px; text-align:left; vertical-align: top;
background-color: #7b8000; color: #e8eabf;
}
th, td{ padding:10px; }
tr:nth-child(even){ background-color: #d6d979; }
tr:nth-child(odd){ background-color: #f0f2b4; }
td, th{ box-shadow:3px 3px 3px rgba(0,0,0,0.2); }
</style>
</head>
<body>
<table>
<tr><th>Number</th><th>Name</th><th>Salary</th></tr>
<tr><td>111</td><td>Ahmed</td><td>6000</td></tr>
<tr><td>222</td><td>Amr</td><td>6100</td></tr>
<tr><td>333</td><td>Ehab</td><td>6300</td></tr>
<tr><td>444</td><td>Ade1</td><td>7500</td></tr>
<tr><td>555</td><td>Mahmoud</td><td>8000</td></tr>
<tr><td>666</td><td>Eslam</td><td>9000</td></tr>
<tr><td>777</td><td>Khaled</td><td>8400</td></tr>
</table>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



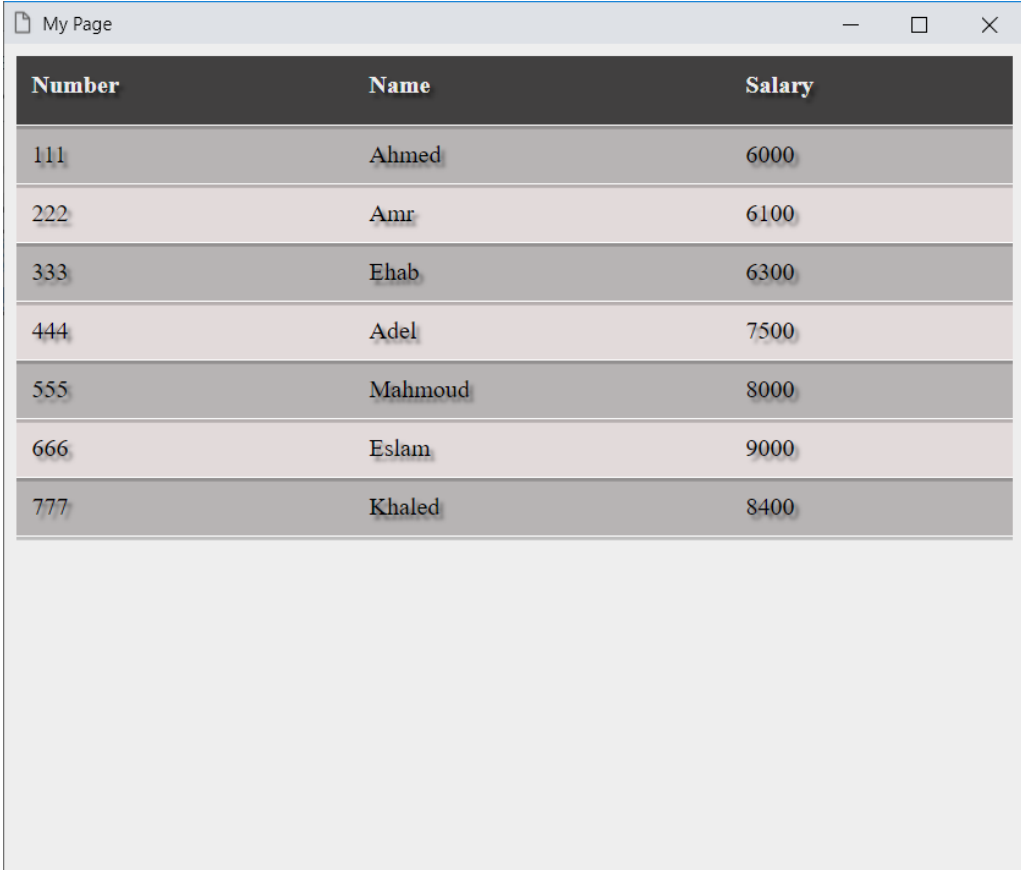
The screenshot shows a web browser window with the title 'My Page'. Inside the window, there is a table with three columns: 'Number', 'Name', and 'Salary'. The table contains seven rows of data.

Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Ehab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khaled	8400

كود! " جدول 15 " قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
body{ background:#eee; }
table{ border-collapse:collapse; width:100%; }
th{
height:25px; text-align:left; vertical-align: top;
background-color: #414040; color: #edecec;
}
th, td{ padding:10px; border-bottom:1px solid #fff; }
tr:nth-child(even){ background-color: #b7b4b4; }
tr:nth-child(odd){ background-color: #e2dada; }
td, th{
box-shadow:0 3px rgba(0,0,0,0.2);
text-shadow:3px 3px 3px rgba(0,0,0,0.5);
}
</style>
</head>
<body>
<table>
<tr><th>Number</th><th>Name</th><th>Salary</th></tr>
<tr><td>111</td><td>Ahmed</td><td>6000</td></tr>
<tr><td>222</td><td>Amr</td><td>6100</td></tr>
<tr><td>333</td><td>Ehab</td><td>6300</td></tr>
<tr><td>444</td><td>Adel</td><td>7500</td></tr>
<tr><td>555</td><td>Mahmoud</td><td>8000</td></tr>
<tr><td>666</td><td>Eslam</td><td>9000</td></tr>
<tr><td>777</td><td>Khaled</td><td>8400</td></tr>
</table>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

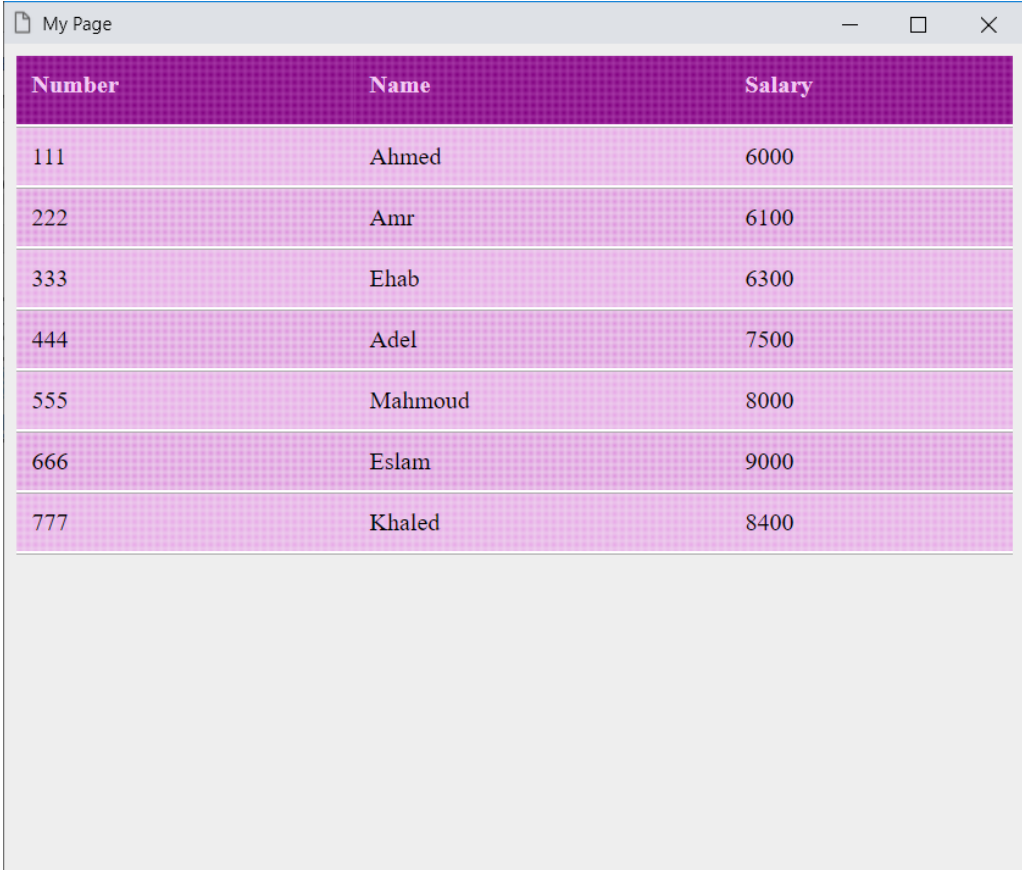


Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Ehab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khaled	8400

كود! " جدول 16 " قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
body{ background:#eee; }
table{ border-collapse: collapse; width:100%; }
th{
height:25px; text-align:left; vertical-align: top;
background:radial-gradient(#800080, #a83ca8) left/5px 5px;
color: #f4caf4;
}
th, td{
padding:10px; border-bottom:3px ridge #fff;
}
tr:nth-child(even){
background:radial-gradient(#e8a8e8, #edd2ed) left/5px 5px;
}
tr:nth-child(odd){
background:radial-gradient(#de93de, #edd2ed) left/5px 5px;
}
</style>
</head>
<body>
<table>
<tr><th>Number</th><th>Name</th><th>Salary</th></tr>
<tr><td>111</td><td>Ahmed</td><td>6000</td></tr>
<tr><td>222</td><td>Amr</td><td>6100</td></tr>
<tr><td>333</td><td>Ehab</td><td>6300</td></tr>
<tr><td>444</td><td>Adel</td><td>7500</td></tr>
<tr><td>555</td><td>Mahmoud</td><td>8000</td></tr>
<tr><td>666</td><td>Eslam</td><td>9000</td></tr>
<tr><td>777</td><td>Khaled</td><td>8400</td></tr>
</table>
</body>
</html>
```


وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

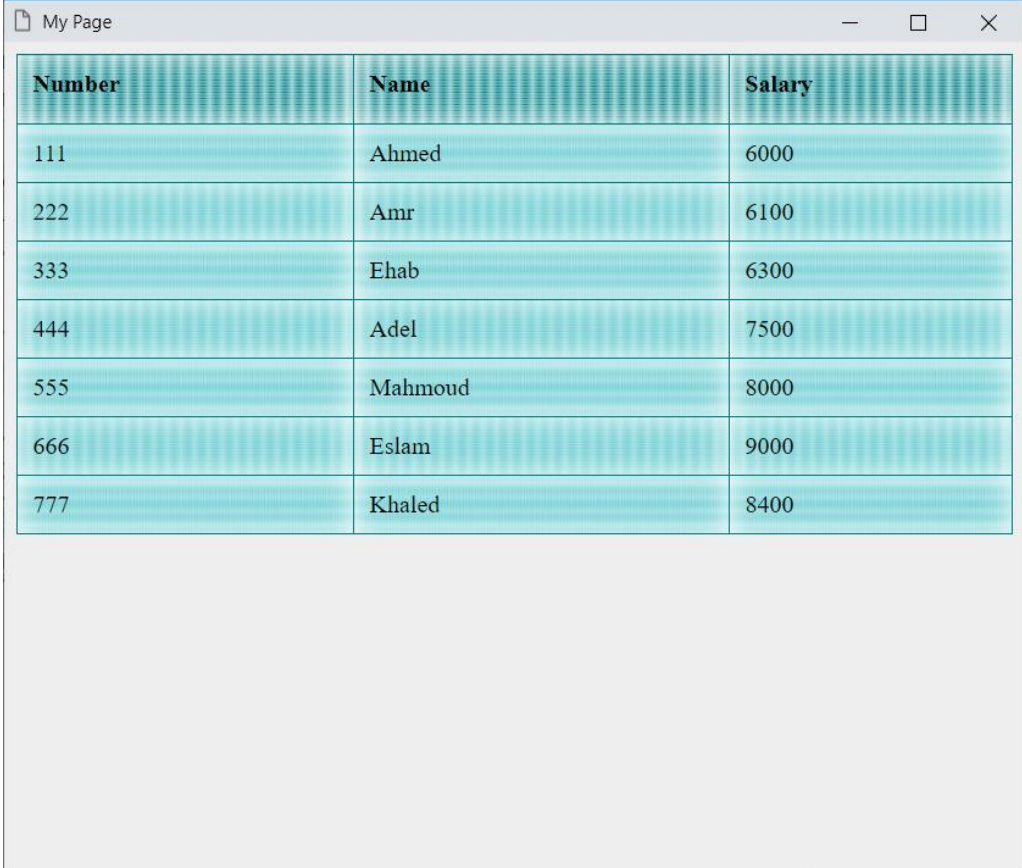


Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Ehab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khaled	8400

كود! " جدول 17 " قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
body{ background:#eee; }
table{ border-collapse: collapse; width:100%; }
th{
height:25px; text-align:left; vertical-align: top;
background:radial-gradient(#007b80, #c7edef) left/9px 2px;
}
th, td{ padding:10px; }
tr:nth-child(even){
background:radial-gradient(#6bcace, #b0e7ea) left/2px 9px;
}
tr:nth-child(odd){
background:radial-gradient(#6bcace, #b0e7ea) left/9px 2px;
}
td, th{
box-shadow:inset 0 0 30px white;
border:1px solid #007b80;
}
</style>
</head>
<body>
<table>
<tr><th>Number</th><th>Name</th><th>Salary</th></tr>
<tr><td>111</td><td>Ahmed</td><td>6000</td></tr>
<tr><td>222</td><td>Amr</td><td>6100</td></tr>
<tr><td>333</td><td>Ehab</td><td>6300</td></tr>
<tr><td>444</td><td>Adel</td><td>7500</td></tr>
<tr><td>555</td><td>Mahmoud</td><td>8000</td></tr>
<tr><td>666</td><td>Eslam</td><td>9000</td></tr>
<tr><td>777</td><td>Khaled</td><td>8400</td></tr>
</table>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



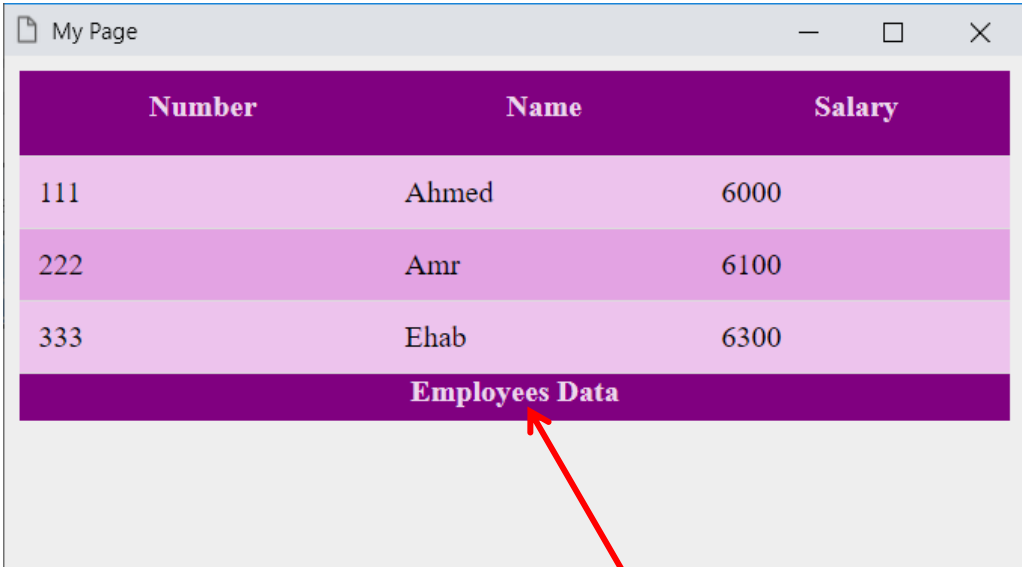
Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Ehab	6300
444	Adel	7500
555	Mahmoud	8000
666	Eslam	9000
777	Khaled	8400

مكان عنوان الجدول caption-side

الخاصية caption-side تأخذ قيمة افتراضية، وهي top، أي وضع العنوان الرئيسي للجدول بالأعلى، ولو أخذت القيمة bottom، سيكون عنوان الجدول بالأسفل، وسنجد **بالكود!** التالي:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
body{ background:#eee; }
table{
border-collapse: collapse; width:100%;
caption-side:bottom;
}
th, caption{
height:25px; vertical-align: top;
background-color: #800080; color: #edd8ed;
}
th, td{ padding:10px; border-bottom:1px solid #ddd; }
tr:nth-child(even){ background-color: #edc3ed; }
tr:nth-child(odd){ background-color: #e3a3e3; }
</style>
</head>
<body>
<table>
<caption><b>Employees Data</b></caption>
<tr><th>Number</th><th>Name</th><th>Salary</th></tr>
<tr><td>111</td><td>Ahmed</td><td>6000</td></tr>
<tr><td>222</td><td>Amr</td><td>6100</td></tr>
<tr><td>333</td><td>Ehab</td><td>6300</td></tr>
</table>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



Number	Name	Salary
111	Ahmed	6000
222	Amr	6100
333	Ehab	6300

Employees Data

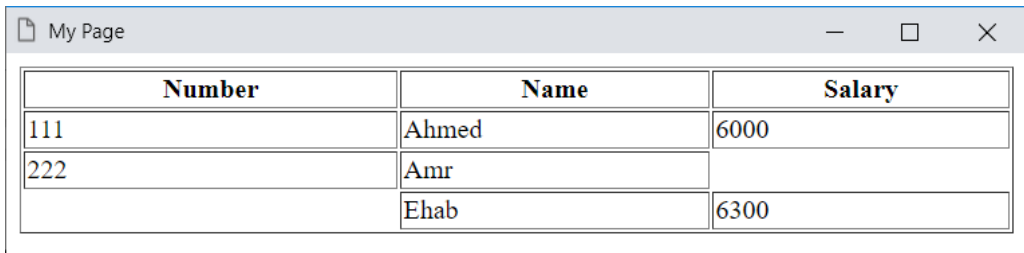
لاحظ أن عنوان الجدول أصبح
bottom بالقيمة

الخلايا الفارغة empty-cells

الخاصية empty-cells تأخذ القيمة show بشكل افتراضي، ولو أخذت القيمة hide، ستخفي أي خلية فارغة، وجرب بالكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
table{
width:100%; empty-cells:hide;
}
</style>
</head>
<body>
<table border="1">
<tr><th>Number</th><th>Name</th><th>Salary</th></tr>
<tr><td>111</td><td>Ahmed</td><td>6000</td></tr>
<tr><td>222</td><td>Amr</td><td></td></tr>
<tr><td></td><td>Ehab</td><td>6300</td></tr>
</table>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



Number	Name	Salary
111	Ahmed	6000
222	Amr	
	Ehab	6300

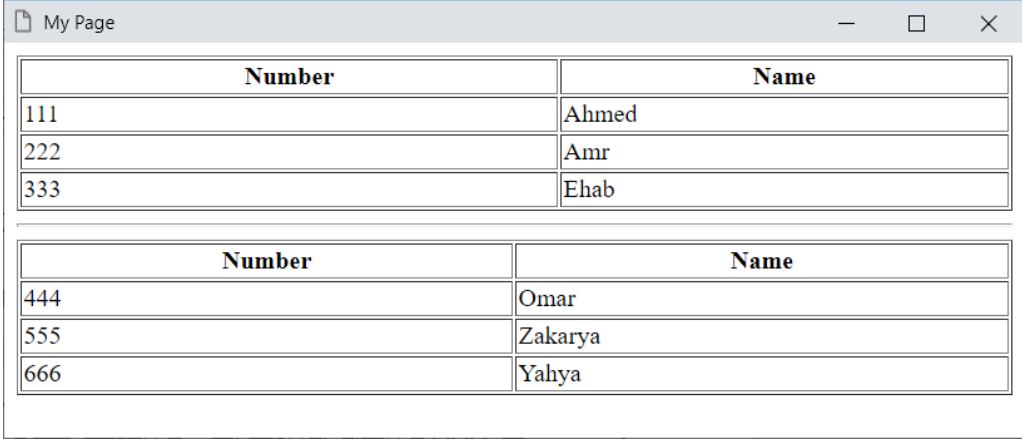
لاحظ أن الخلايا الفارغة، غير موجودة، وتم اظهار ال border للجدول ليكون هذا واضحاً.

تنسيق الجدول table-layout

خاصية table-layout، لتنسيق الجدول بطريقة معينة، وتأخذ القيمة auto لينسق تلقائياً، او fixed، ليكوه ثابت على الأعمدة.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Page</title>
    <style>
      #tbl1{ width:100%; table-layout:auto; }
      #tbl2{ width:100%; table-layout:fixed; }
    </style>
  </head>
  <body>
    <table id="tbl1" border="1">
      <tr><th>Number</th><th>Name</th></tr>
      <tr><td>111</td><td>Ahmed</td></tr>
      <tr><td>222</td><td>Amr</td></tr>
      <tr><td>333</td><td>Ehab</td></tr>
    </table>
    <hr>
    <table id="tbl2" border="1">
      <tr><th>Number</th><th>Name</th></tr>
      <tr><td>444</td><td>Omar</td></tr>
      <tr><td>555</td><td>Zakariya</td></tr>
      <tr><td>666</td><td>Yahya</td></tr>
    </table>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



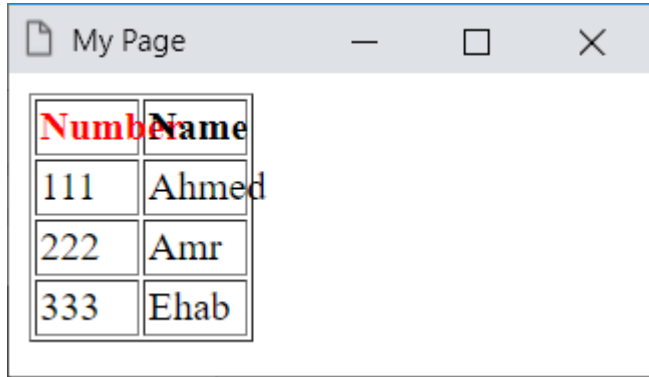
Number	Name
111	Ahmed
222	Amr
333	Ehab

Number	Name
444	Omar
555	Zakarya
666	Yahya

لاحظ أنه تم عمل العرض 100% ليتضح لنا الفرق، ويمكنه توضيح القيمة fixed بطريقة أخرى، بالكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
#tbl1{ width:90px; table-layout:fixed; }
</style>
</head>
<body>
<table id="tbl1" border="1">
<tr><th style="color:red">Number</th><th>Name</th></tr>
<tr><td>111</td><td>Ahmed</td></tr>
<tr><td>222</td><td>Amr</td></tr>
<tr><td>333</td><td>Ehab</td></tr>
</table>
</body>
</html>
```


وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



Number	Name
111	Ahmed
222	Amr
333	Ehab

لاحظ أنه مع القيمة fixed، وتصغير عرض الجدول، خرج الكلام عن الخلايا، ولو تم التبديل بالقيمة auto لما حدث ذلك.

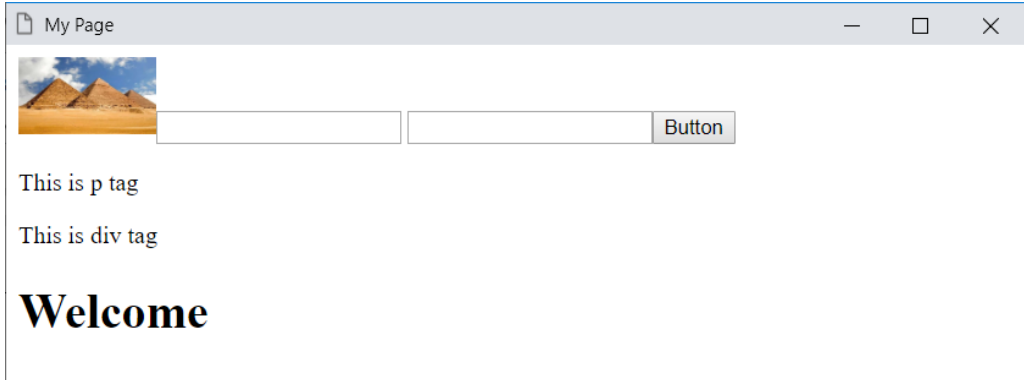
الموضحة Position

مفهوم الموضحة

انتبه! وركز جيداً لأن هذا الموضوع **هام جداً** أنكه مما تتخيل.
إذا تم كتابة **كود!** مثل الكود التالي:

```
<!DOCTYPE html>
<html>
<head>
  <title>My Page</title>
</head>
<body>
  <input>
  <input><button>Button</button><p>This is p tag</p>
  <div>This is div tag</div><h1>Welcome</h1>
</body>
</html>
```

سوف يتم انشاء العناصر كما نعرف مثل التالي:



ولو اردت التحكم في التغيير، فسوف تغير مثلاً من inline الى block أو من block الى inline، ويمكنك وضع رمز المسافة، أو التحكم في التنسيق بالجدول، أو تغيير الحشو والهامش، ولكه الموضوع هنا مختلف تماماً، فرجاء **انتبه!** وركز جيداً فيما يلي.


هل سألت نفسك ذات مرة، ما إذا كنت تريد انشاء العناصر كما يلي:




انتبه! ولاحظ أن كل شيء تغير، وكل العناصر خرجت عن طبيعتها المعروفة، فأصبح هناك وسوم على بعضها، ووسوم تنقلت مع أماكنها، وكأنك أمسكتها بيدك وحركتها كما تريد، وأيضاً **انتبه!** للعناصر، حيث يوجد عنصر في مكان مع عنصر آخر في نفس المكان، فإيهما يظهر فوق الآخر ليغطي عليه، والسؤال الآن هو كيف نعمل مثل هذه الأشياء بفهم؟ وبفضل الله الجواب هو أنك ستفعلها بسهولة إن شاء الله، ولكن **انتبه!** وركز جيداً لتأسس أولاً بفهم، ثم تطبيق.

الموضع الطبيعي والتغيد

يتم وضع العناصر ان كانت على inline، مثل الشكل التالي:



ويتم وضع العناصر ان كانت على block، مثل الشكل التالي:



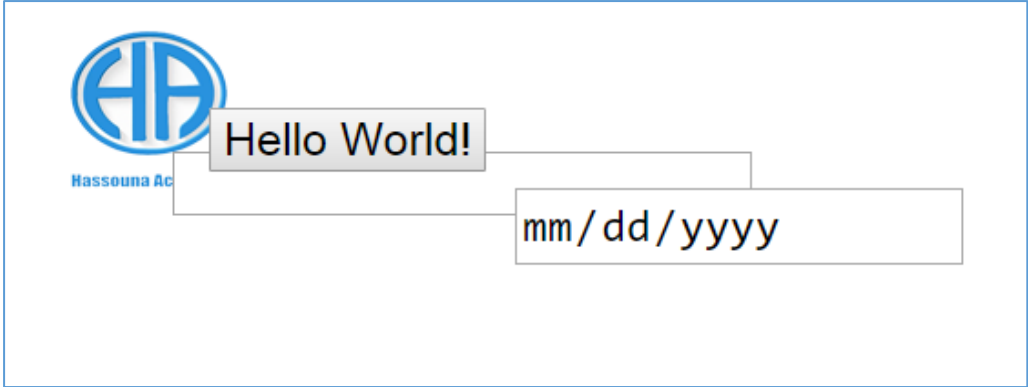
Hassouna Academy

والآن!، ماذا لو أردنا وضعهم كل عنصر في مكان مختلف، مثل التالي:

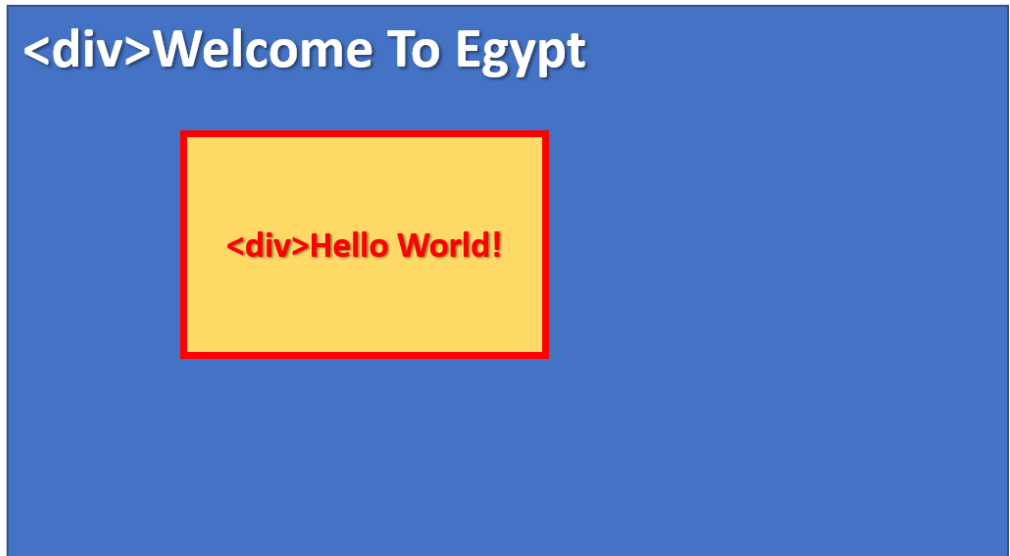


Hassouna Academy

وماذا لو أردنا وضعهم في شكك متداخل مع تحديد ايهم يظهر فوق الآخر،
 مثل الشكل التالي:



وماذا لو أردنا وضع حاوية داخل حاوية أخرى مع ضبط مكانها بناء
 على الحاوية الأم، مثل الشكل التالي:

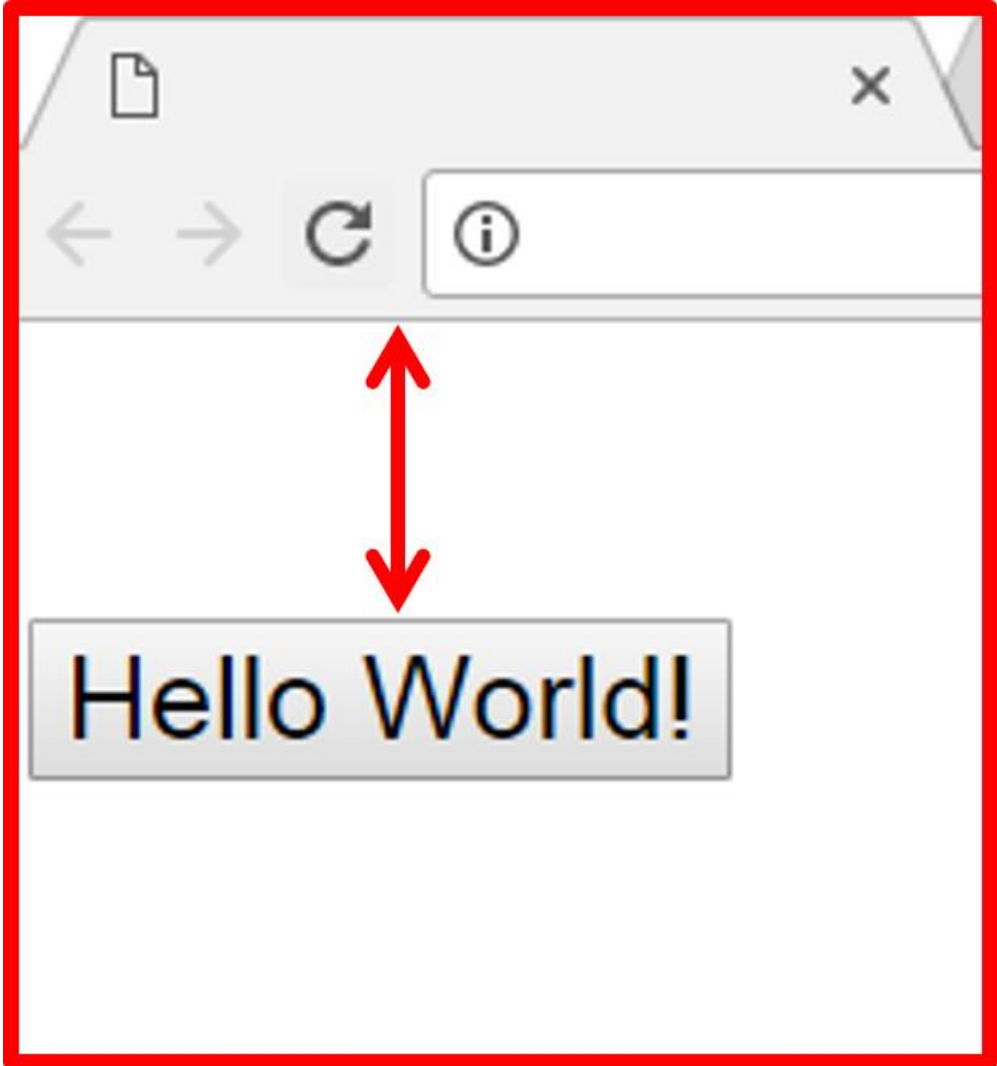


بإذن الله فيما يلي سوف نتفهم هذا وأكثر بفضل الله.

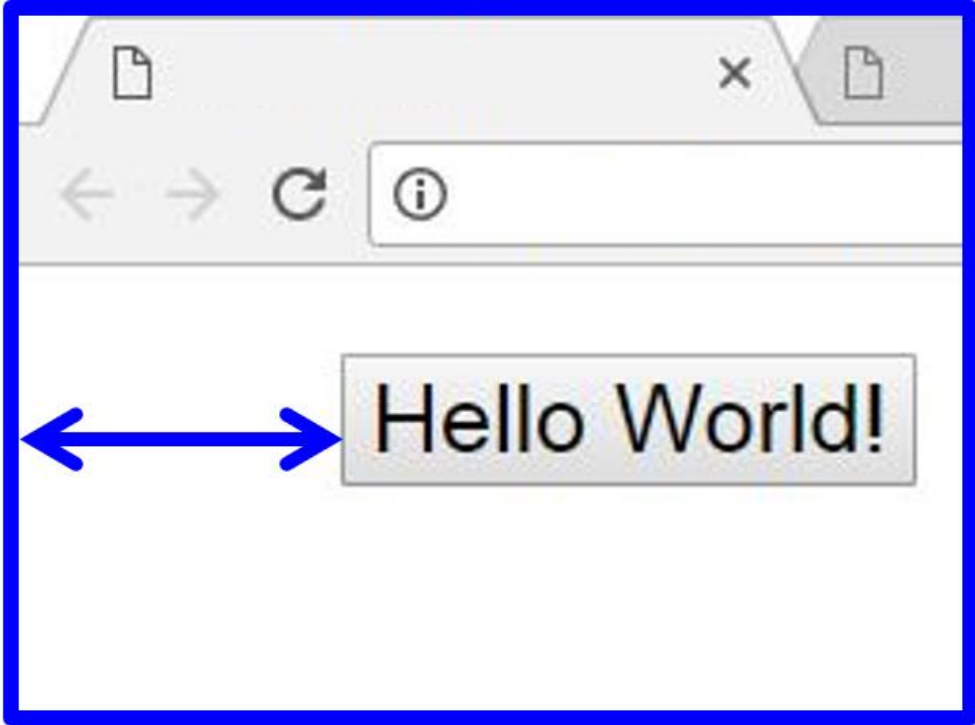
أبعاد الحواف

لابد أن تعرف الأبعاد عه الحواف جيداً، قبل الشروع في الموقع.

البعد عه الحافة العليا هو top، وسوف نعبه عنه ونميزه باللون الأحمر، والشكل التالي يوضحه ويعبه عنه:



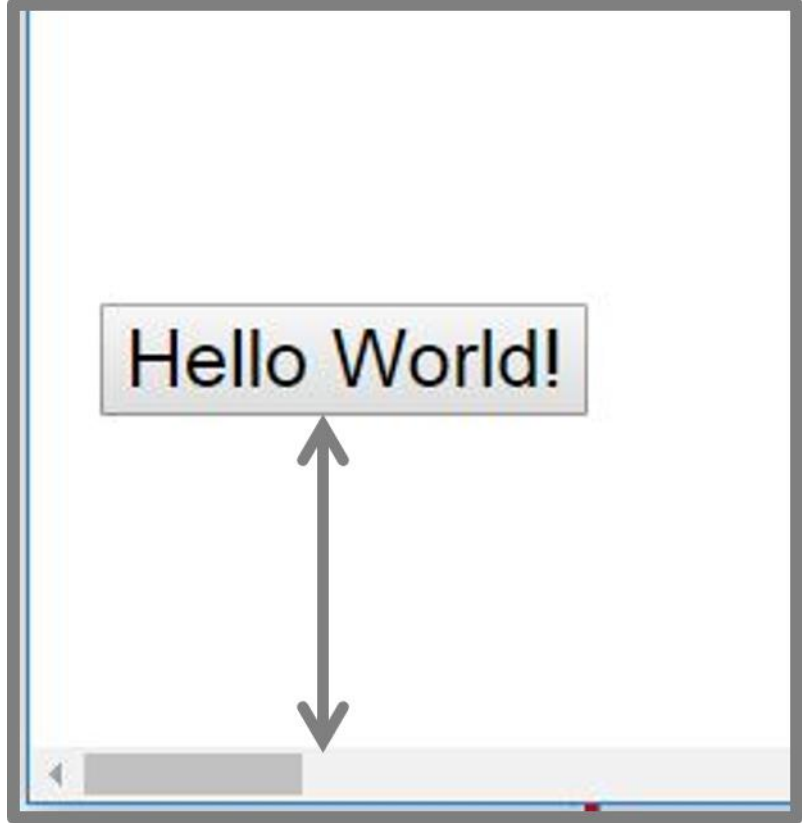
البعد عن الحافة اليسرى هو left، وسوف نعبئ عنه باللون الأزرق،
والشكل التالي يوضحه ويعبر عنه:



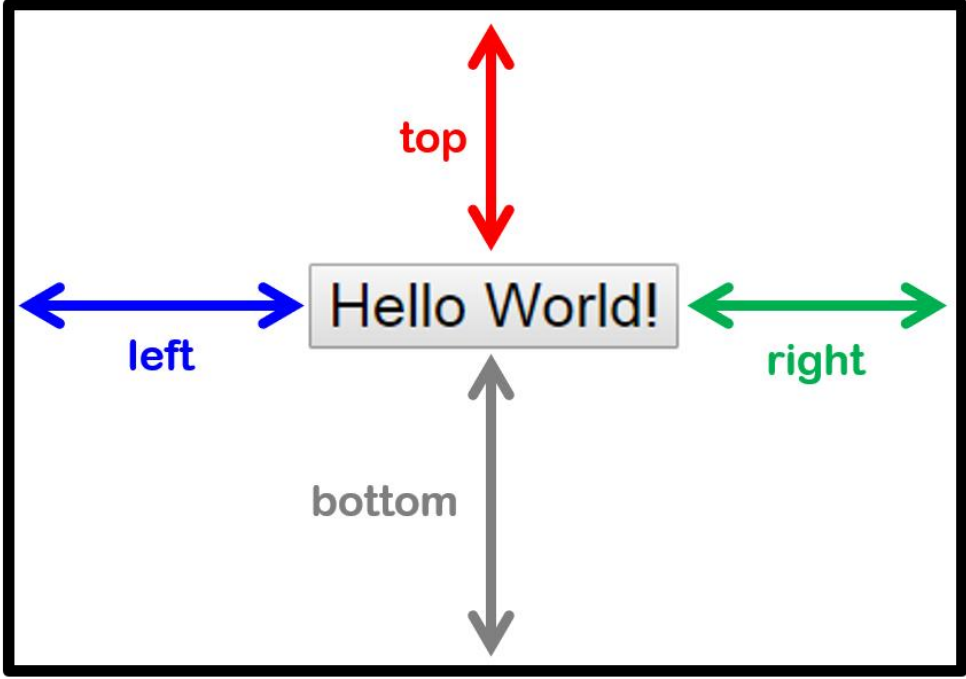
البءء عءء الءافة الئمى هو right، وسوف نءءء عءء باللون الأخضر،
والشكل الآلى يؤءءءه وبعءء عءء:



البعء عن الحافة السفلى هو bottom، وسوف نميزه باللون الرمادي،
والشكل التالي يوضحه ويعبر عنه:



إذاً، الأرباع ابعاد الرئيسية هي (**top**, **left**, **right**, **bottom**)،
والصورة التالية تعبر عنهم:



وكذلك هذه خواص في لغة CSS، وكل خاصية منهم تأخذ قيمة تعبر فيها
عن المسافة بين الحافة والعنصر.

أنواع الأوضاع

العناصر تأخذ قيمة الوضع لتحل مكان على صفحة الويب، والأوضاع على صفحة الويب باختصار، كما يلي:

الوضع static (الطبيعي)

هو الوضع السائد الافتراضي لأي عنصر ولا يتأثر بـ top, left, right, bottom

الوضع relative (مره مع العناصر)

هو وضع نسبي، حيث يتم تحديد أبعاده مع بعد أقرب عنصر له

الوضع absolute (حر في أي مكان)

هو الوضع الحر، أي يتم وضعه في أي مكان بشكل مستقل

الوضع fixed (حر ولاصق بثبات)

هو وضع لاصق، وهو مثل absolute ولكنه أضيف على ذلك أنه لا يختلف أبداً حتى لو تم تحريك شريط التمرير للصفحة

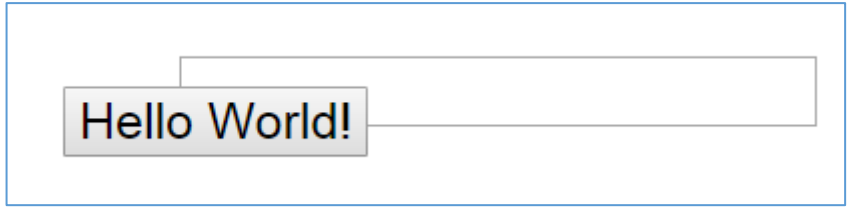
الوضع sticky (مره ولاصق متغير)

هو وضع مره relative إذا لم تم عليه، ويلصق بثبات مثل fixed إذا مررت عليه، ثم يظل ثابتاً طالما أنت تحته.

ظهور عنصر على آخر

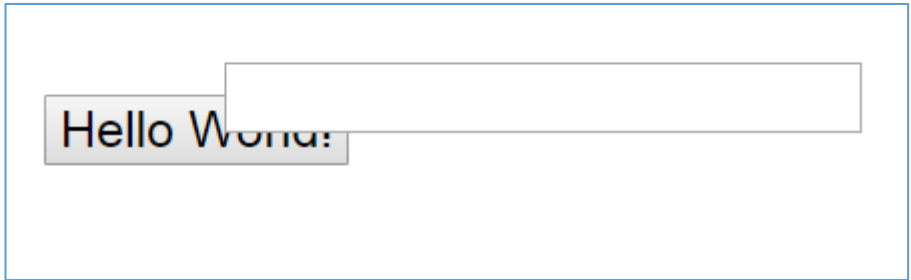
عندما تقوم بإنشاء عنصرا على صفحة الويب الخاصة بك، ويكونان مشتركان في نفس الموضع، وفي مكان واحد، فلا بد أن يظهر أحدهما على الآخر، مثل أن تغط أنت وصاحبك في صف طابور، فإذا كنت أنت في الأمام، فسوف تظهر ويختفي صاحبك، وإذا كان صاحبك هو الذي في الأمام، ستختفي ويظهر صاحبك، ولننتفهم أكثر بشكل عملي، تابع ما يلي:

انظر الى الشكل التالي:



ولاحظ أن الزر ظاهر على مربع النص.

ثم انظر الى الشكل التالي:

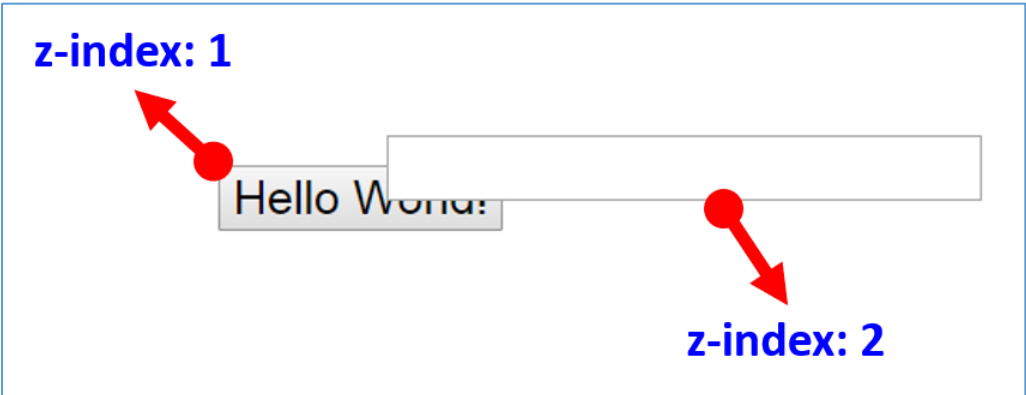


ولاحظ أن مربع النص هو الذي ظهر على الزر.

والآن هيا بنا لتتعرف كيفية ظهور العناصر على بعضها، وهو أمر سهل
عنه طريق خاصة باسم z-index، وهي التي تتحكم في ظهور عنصر
على الآخر حسب القيمة التي تأخذها تلك الخاصية، فالعنصر الذي يأخذها
بقيمة أكبر من قيم z-index للعناصر الأخرى يفوز بالظهور، ولنتفهم
أكثر، لاحظ وانظر الى الشكل التالي:



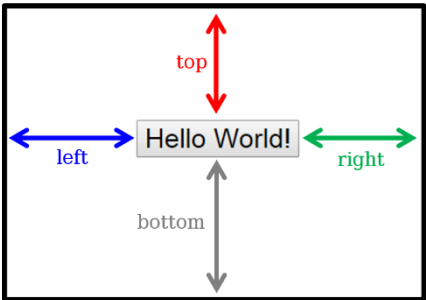
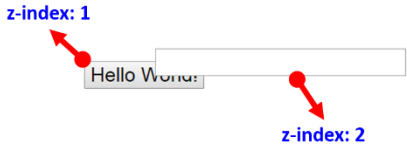

وانظر أيضاً الى الشكل التالي:



ثم لاحظ أنه عند وجود قيمة أكبر للخاصية z-index يتم اظهار
العنصر على عنصر ذا قيمة أقل منه.

ملخص مفهوم الموضع

بفضل الله وعونه تحدثنا حتى الآن عن، مفهوم الموضع، وطبيعة الموضع، وأبعاد الجواف، وأنواع المواضع، وظهور العناصر على بعض، ولتلاخيص كل هذا بفضل الله ركز على الشكل التالي:

الأوضاع	
الابعاد (top, left, right, bottom)	الوضع static هو الوضع العائنه الافتراضي لأي عنصر، ولا يتأثر بـ top, left, right, bottom
	الوضع relative هو وضع نسبي، ويمكنه تحديد ابعاد العنصر إلى أقرب عنصر له
	الوضع absolute هو الوضع الحر، أي يتم وضعه في أي مكان بشكل مستقل
	الوضع fixed هو وضع لاصق، وهو مثل absolute ولكنه اضعف على ذلك انه لا يتغير أبداً
	الوضع sticky هو وضع هربه relative إذا لم نمر عليه، ويلصق ببيانات مثل fixed إذا هربت عليه، ثم يظهر ثابتاً طالما أنت تحتها.
الظهور	
	

وبفضل الله عند التركيز جيداً في الشكل السابق، سوف يثبت في ذهنك دائماً بفضل الله مفاهيم الأوضاع التي تجعلك متمكناً بفضل الله في موضوع الـ position، لأنه كما ذكرنا من قبل هام جداً.

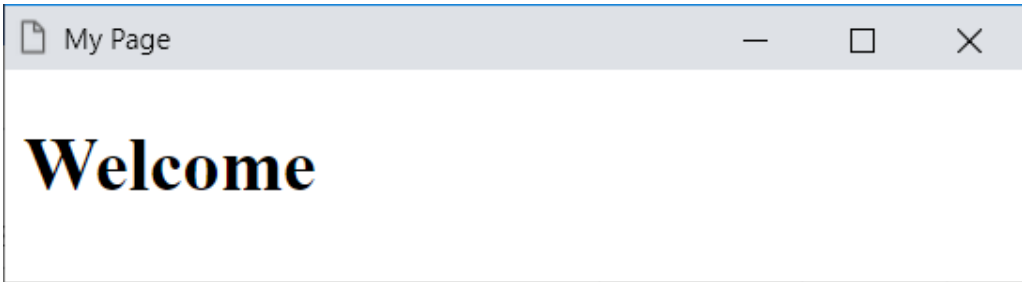
انتبه! فإن هذه الصفحة ملخص لموضوع الـ position، نعم هي كذلك بفضل الله، ولكنها لا تغنيك عن ترك ما تم شرحه بفضل الله، أو ترك ما سوف يتم شرحه إن شاء الله.

الوضع static

هذا هو الطبيعي أو الافتراضي، وكما ذكرنا من قبل أنه لا يتأثر بأبعاد الحواف، ولملاحظة ذلك، تابع المثال التالي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <title>My Page</title>
  <style>
    h1{
      position:static;
      top:150px; left:150px;
    }
  </style>
</head>
<body>
  <h1>Welcome</h1>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



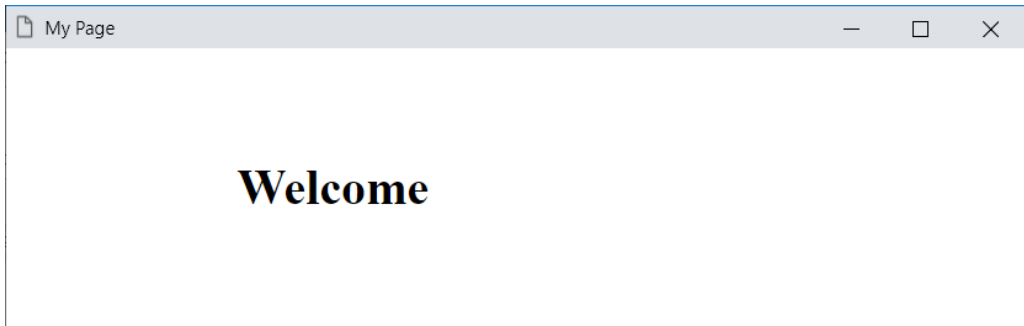
لاحظ أن العنصر h1 لم يتحرك من مكانه، ولو أنك أزلت تنسيقات لغة ال CSS، له تجد هناك أي فرق، لأن static هو الافتراضي.

الوضعية absolute

هذا الوضعية يمكنه تشبيعه كما يقال بالمصري بـ "الفتوة"، أو "القومندو"، أو "الكبير"، لأنه يستطيع أن يدخل أو يذهب أي مكان بدون إذن، ولا يهتم بأي عنصر أبداً، وسنعب عنه بما يلي إن شاء الله.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Page</title>
    <style>
      h1{
        position:absolute;
        top:50px; left:150px;
      }
    </style>
  </head>
  <body>
    <h1>Welcome</h1>
  </body>
</html>
```

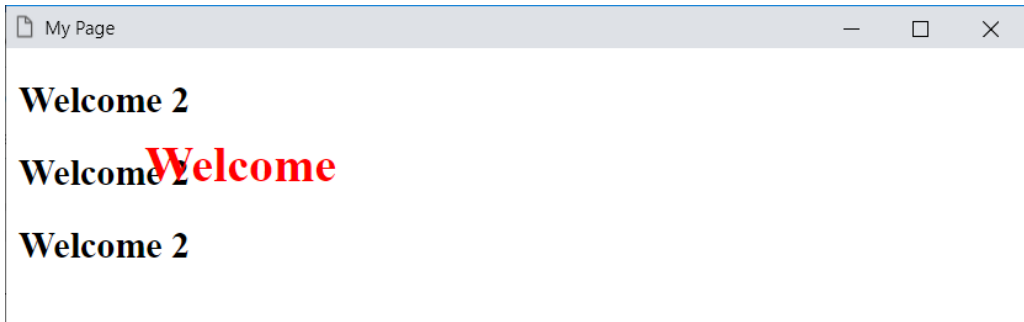
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه في المثال السابق، تم تحريك الـ h1 بمقدار 50px من الحافة العليا، وتحرك بمقدار 150px من الحافة اليسرى.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
h1{
position:absolute;
top:35px; left:90px; color:red;
}
</style>
</head>
<body>
<h1>Welcome</h1>
<h2>Welcome 2</h2><h2>Welcome 2</h2><h2>Welcome 2</h2>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



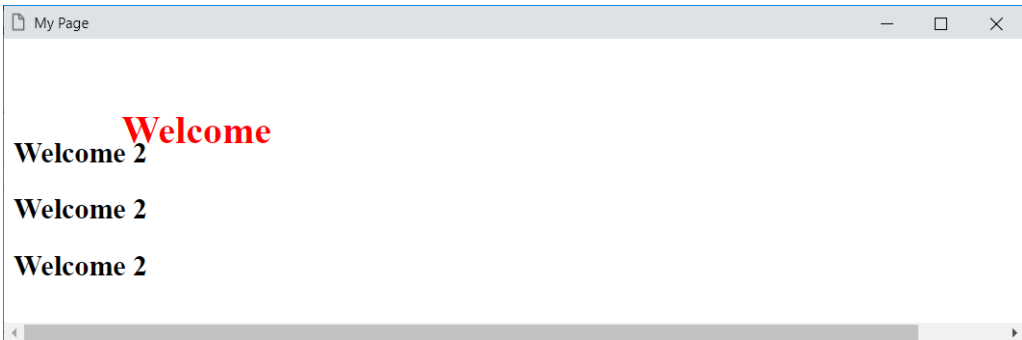
لاحظ أن حركة الـ h1، لا تداعي أي عنصر آخر، وتم وضعها في المكان المحدد.

الوضع relative

هو يستجيب لك، حيث يتم وضعه في مكانه كما تريد، ولكنه يراعي العناصر القريبة منه، وسوف نكتب نفس المثال التالي، مع التغيير الى relative. **كود!** قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
h1{
position:relative;
top:35px; left:90px; color:red;
}
</style>
</head>
<body>
<h1>Welcome</h1>
<h2>Welcome 2</h2><h2>Welcome 2</h2><h2>Welcome 2</h2>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

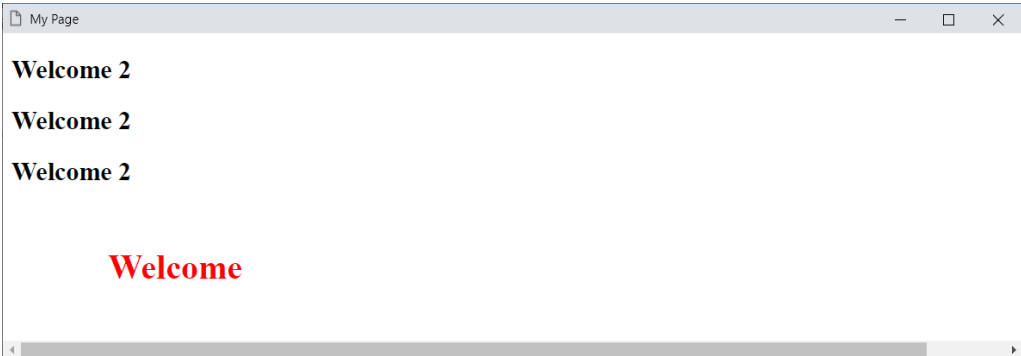


لاحظ أنه يراعي ما بعده من عناصر.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
h1{
  position:relative;
  top:35px; left:90px; color:red;
}
</style>
</head>
<body>
<h2>Welcome 2</h2><h2>Welcome 2</h2><h2>Welcome 2</h2>
<h1>Welcome</h1>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه يراعي ما قبله من عناصر.

الوضع fixed

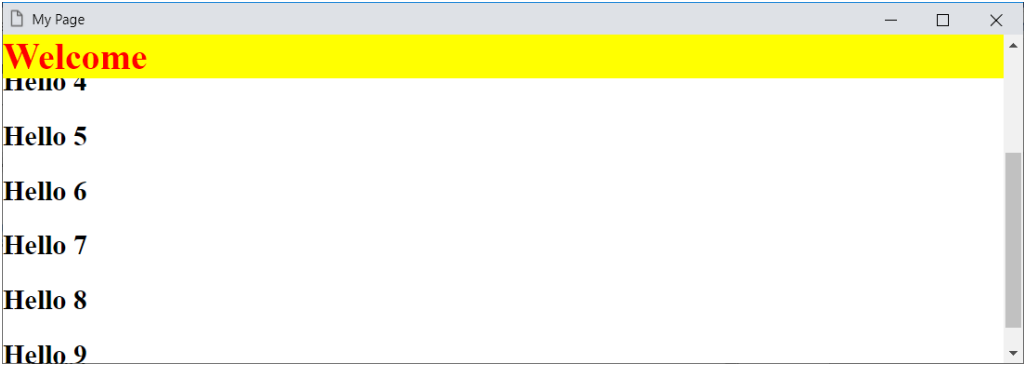
كل الأوضاع السابقة، تكون موجودة في مكان ما في صفحة الويب، ولكنه إذا كان هناك عناصر أخرى، وتم تحريك ال scroll عن هذه الأوضاع، فسوف تختفي، أما الوضع fixed، فهو وضع لاصق، حتى ولو كان هناك عناصر، وللتعرف عليه، ركن فيما يلي.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
body{ padding-top:25px; margin:0; }
h1{
margin:0; width:100%;
position:fixed;
top:0px; left:0px;
background:yellow;
color:red;
}
</style>
</head>
<body>
<h1>Welcome</h1>
<h2>Hello 1</h2><h2>Hello 2</h2><h2>Hello 3</h2>
<h2>Hello 4</h2><h2>Hello 5</h2><h2>Hello 6</h2>
<h2>Hello 7</h2><h2>Hello 8</h2><h2>Hello 9</h2>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



والآن انتبه! وحرك شريط التمرير، لتدري مثل الشكل التالي:

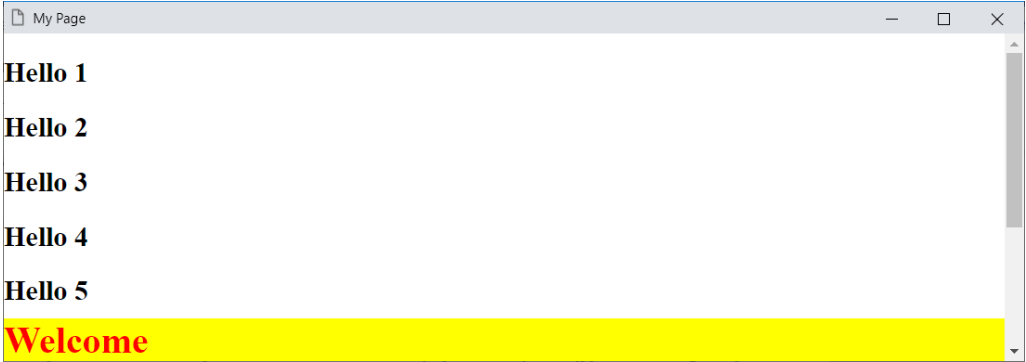


نعم، انه ثابت تماماً، وكأنه معلق، ولكنه رجاء انتبه! ولاحظ أنه ثابت من الأعلى، مع هامشك صفر، ولاحظ في الكود أنه أخذ العرض بالكامل، ولاحظ في الكود أن الـ `body` يأخذ حشو من الأعلى بمقدار `25px`، وذلك حتى لا يختفي أول عنصر وراء الـ `h1` الثابت، ولاحظ في الكود أن الـ `h1` الثابت يأخذ القيمة صفر من اليسار ومن الأعلى.

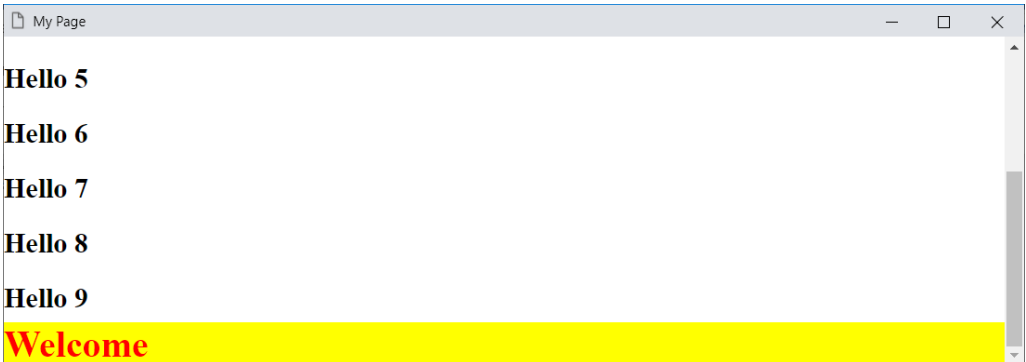
والآن **انتبه!** لأننا سوف نجعله لاصق من الأسفل، كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <title>My Page</title>
  <style>
    body{ padding-bottom:25px; margin:0; }
    h1{
      margin:0; width:100%;
      position:fixed;
      bottom:0px; left:0px;
      background:yellow;
      color:red;
    }
  </style>
</head>
<body>
  <h1>Welcome</h1>
  <h2>Hello 1</h2><h2>Hello 2</h2><h2>Hello 3</h2>
  <h2>Hello 4</h2><h2>Hello 5</h2><h2>Hello 6</h2>
  <h2>Hello 7</h2><h2>Hello 8</h2><h2>Hello 9</h2>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



والآن **انتبه!** وحرك شريط التمرير، لتدري مثل الشكل التالي:

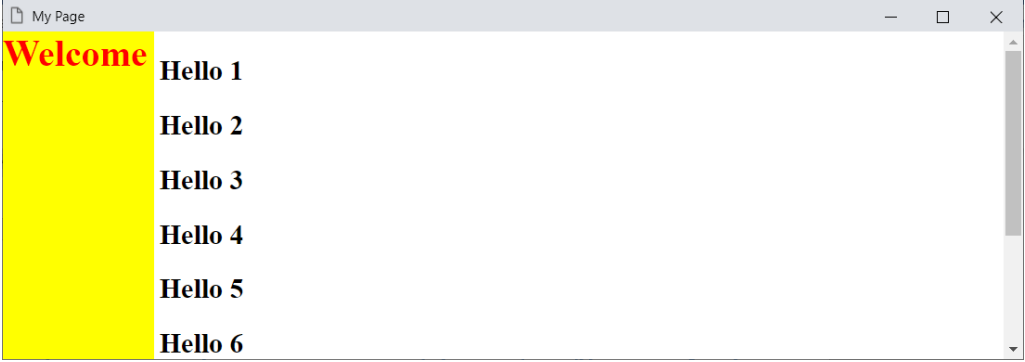


نعم، انه ثابت تماماً مع الأسفل، مع هامش صفير، ولاحظ في الكود انه أخذ العرض بالكامل، ولاحظ في الكود أن الـ `body` يأخذ حشو مع الأسفل بمقدار `25px`، وذلك حتى لا يختفي آخر عنصر في الصفحة وراء وسم الـ `h1` الثابت، ولاحظ في الكود أن الـ `h1` الثابت يأخذ القيمة صفير مع اليسار ومع الأسفل.

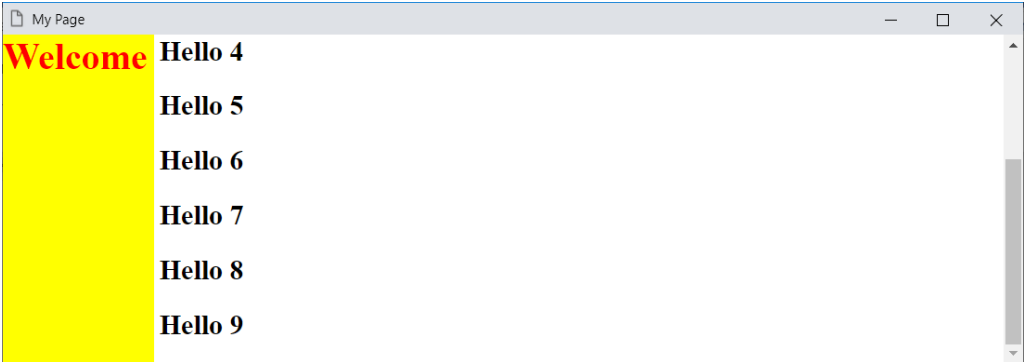
والآن **انتبه!** لأننا سوف نجعله لاصق مع اليسار، كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <title>My Page</title>
  <style>
    body{ padding-left:135px; margin:0; }
    h1{
      margin:0; width:130px; height:100%;
      position:fixed;
      bottom:0px; left:0px;
      background:yellow;
      color:red;
    }
  </style>
</head>
<body>
  <h1>Welcome</h1>
  <h2>Hello 1</h2><h2>Hello 2</h2><h2>Hello 3</h2>
  <h2>Hello 4</h2><h2>Hello 5</h2><h2>Hello 6</h2>
  <h2>Hello 7</h2><h2>Hello 8</h2><h2>Hello 9</h2>
</body>
</html>
```


وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



والآن انتبه! وحرك شريط التمرير، لتدري مثل الشكل التالي:

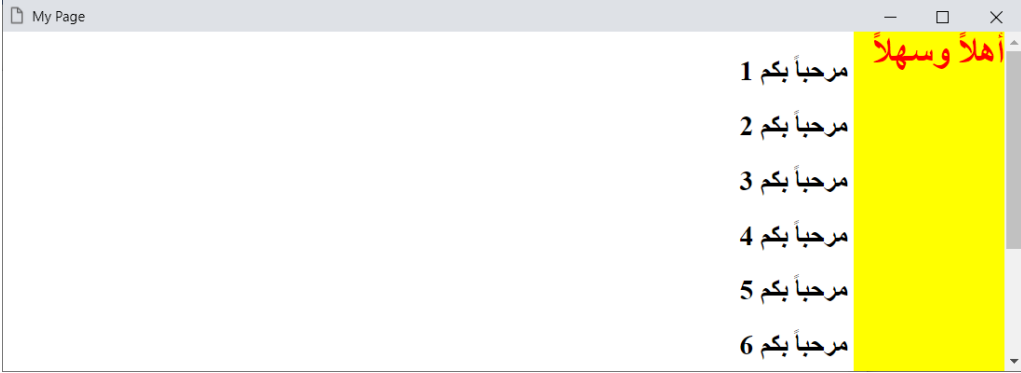


نعم، انه ثابت تماماً مع اليسار، مع هامش صفر، ولاحظ في الكود انه أخذ العرض بمقدار يظهر الكلمة Welcome المكتوبة في الوسم `h1`، ولاحظ في الكود أن الوسم `body` يأخذ حشو مع اليسار بمقدار عرض الوسم `h1 + 5px` ليصبح `135px`، وذلك حتى لا تختفي جميع العناصر وراء عرض الوسم `h1` الثابت، ولاحظ في الكود أن الوسم `h1` الثابت يأخذ القيمة صفر مع اليسار ومع الأسفل، وأيضاً أخذ الطول بالكامل.

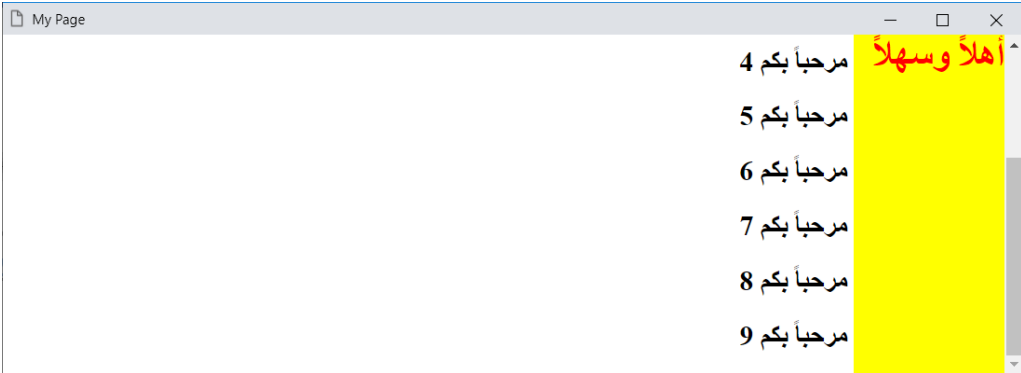
والآن **انتبه!** لأننا سوف نجعله لاصقاً من اليمين، كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Page</title>
    <style>
      body{          padding-right:135px;          margin:0;
direction:rtl; }
      h1{
        margin:0; width:130px; height:100%;
        position:fixed;
        bottom:0px; right:0px;
        background:yellow;
        color:red;
      }
    </style>
  </head>
  <body>
    <h1>أهلاً وسهلاً</h1>
    <h2>1 مرحباً بكم</h2><h2>2 مرحباً بكم</h2>
    <h2>3 مرحباً بكم</h2><h2>4 مرحباً بكم</h2>
    <h2>5 مرحباً بكم</h2><h2>6 مرحباً بكم</h2>
    <h2>7 مرحباً بكم</h2><h2>8 مرحباً بكم</h2>
    <h2>9 مرحباً بكم</h2>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



والآن انتبه! وحرك شريط التمرير، لتدي مثل الشكل التالي:



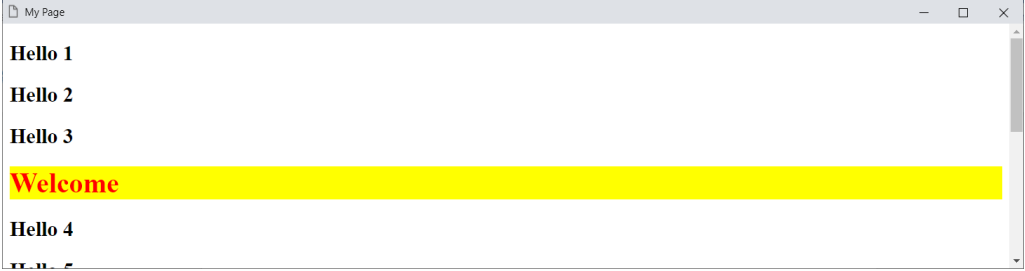
نعم، انه ثابت تماماً مع اليمين، مع هامش صفر، ولاحظ في الكود انه أخذ العرض بمقدار يظهر الكلمة أهلاً وسهلاً المكتوبة في الوسم h1، ولاحظ في الكود أن الوسم body يأخذ حشو مع اليمين بمقدار عرض الوسم h1 + 5px ليصبح 135px، وذلك حتى لا تختفي جميع العناصر وراء عرض الوسم h1 الثابت، ولاحظ أن الوسم h1 الثابت يأخذ القيمة صفر مع اليسار ومع الأسفل، وأيضاً أخذ الطول بالكامل، ولاحظ أن اتجاه الوسم body يأخذ rtl ليناسب اللغة العربية.

الوضعية sticky

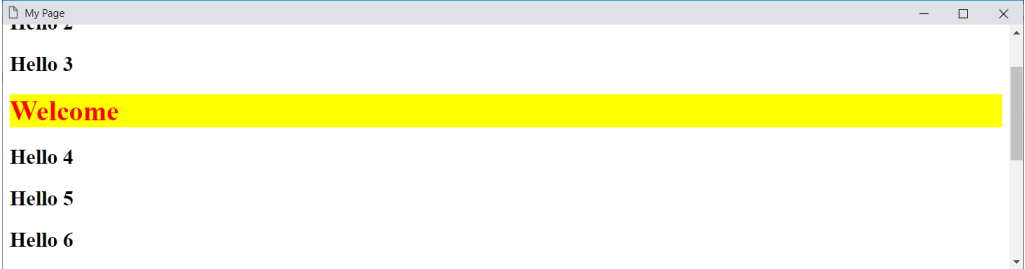
هو وضعية من relative إذا لم تم عليه، ويلصق بثبات مثل fixed إذا مرت عليه، ثم يظل ثابتاً طالما أنت تحته.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Page</title>
    <style>
      h1{
        position:sticky; top:5px;
        margin-bottom:20px;
        background:yellow; color:red;
      }
    </style>
  </head>
  <body>
    <h2>Hello 1</h2><h2>Hello 2</h2><h2>Hello 3</h2>
    <h1>Welcome</h1>
    <h2>Hello 4</h2><h2>Hello 5</h2><h2>Hello 6</h2>
    <h2>Hello 7</h2><h2>Hello 8</h2><h2>Hello 9</h2>
    <h2>Hello 10</h2><h2>Hello 11</h2><h2>Hello 12</h2>
  </body>
</html>
```

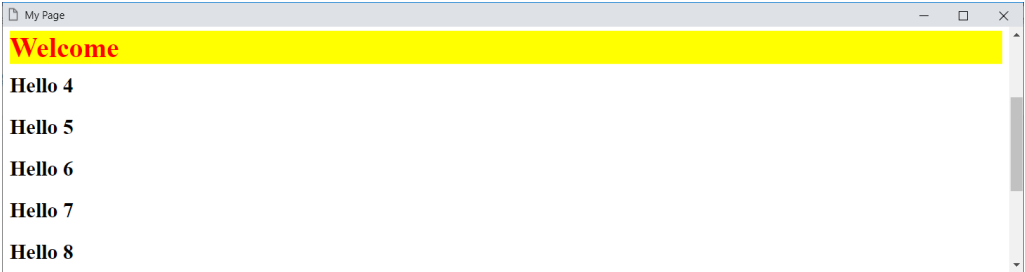
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



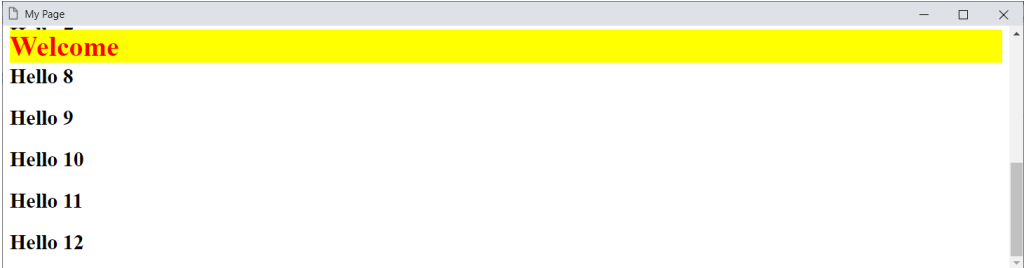
والآن حرك شريط التمرير قليلاً، لترى مثل الشكل التالي:



والآن حركه مره أخرى قليلاً، لترى مثل الشكل التالي:



والآن سوف يثبت، إذا نزل مره أخرى، مثل الشكل التالي:



ومع المثال السابق، لاحظ أن الوسم h1 المقصود بالقيمة sticky، يوجد عناصر قبل منه، ويوجد عناصر بعد منه، وهو يراعيها وكأنه relative، وبعد مرور شريط التمرير على الوسم h1، ثم النزول أسفل منه، أصبح ثابتاً وكأنه fixed، وهذا هو الوضع sticky عملياً على صفحة الويب.

التنظيم Float

معني float

الكلمة float تعني عائمة، وتُخيل معي مدرب السباحة، وهو يقول لمتدريه السباحة أن يعموا ناحية اليمين، فيكون اتجاههم بالكامل نحو اليمين، أو يقول لهم عوموا ناحية اليسار، فيكون اتجاههم بالكامل نحو اليسار، وباختصار فإن هذا المعني يمثل ال float، حيث يتم وضع العناصر نحو اليمين، أو نحو اليسار، وللتجارب تابع ما يلي.

عندما تنج العنصر نحو اليسار أو نحو اليمين، وكأنتك وضعت وسم span وبقواره وسم span آخر، وهكذا، أو كأنتك وضعت وسم input وبقواره وسم input آخر، وهكذا، ولك **انتبه!** فهذا لا تقصد ال display، ولك فقط نحاول التشبيه لك لتتفهم الأمر.

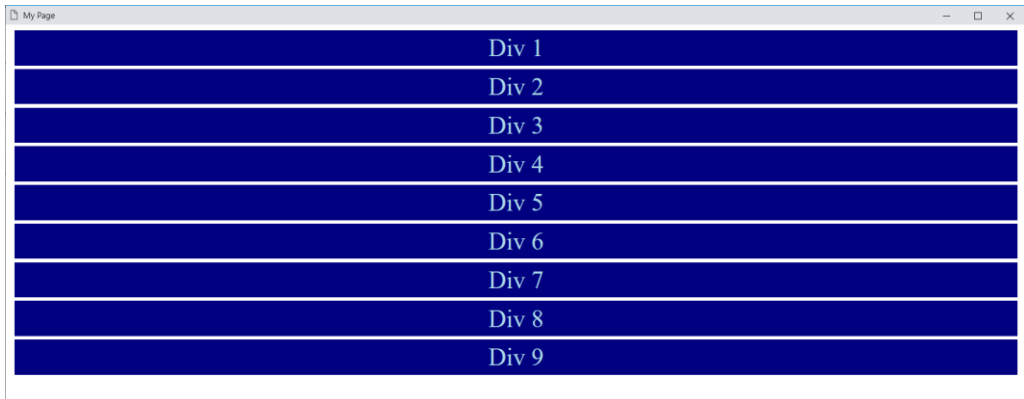
يتم التحكم في ال float عن طريق الخاصية float، والقيمة الافتراضية لها هي none، أي لا يوجد float في الأساس.

اتجاه العناصر نحو اليسار float left

والآن سوف نبدأ بداية عادية، كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
div{
background:navy; color:lightblue; font-size:35px;
margin:5px; padding:5px; text-align:center;
}
</style>
</head>
<body>
<div>Div 1</div><div>Div 2</div><div>Div 3</div>
<div>Div 4</div><div>Div 5</div><div>Div 6</div>
<div>Div 7</div><div>Div 8</div><div>Div 9</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

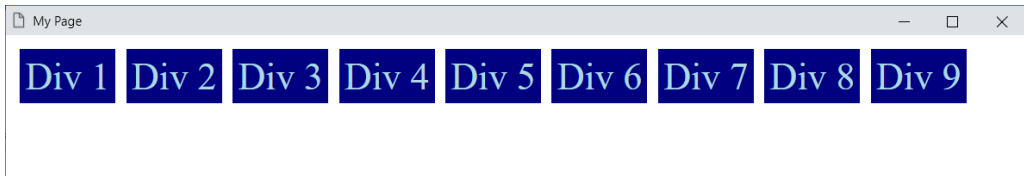


وكما نلاحظ وجود 9 وسوم div، مع تنسيقات عادية، ومعروفة.

والآن انتبه! جيداً، لأننا سوف نعيد نفس المثال، مع زيادة بسيطة جداً.
كود! قم بكتابة الكود التالي ثم احفظه:

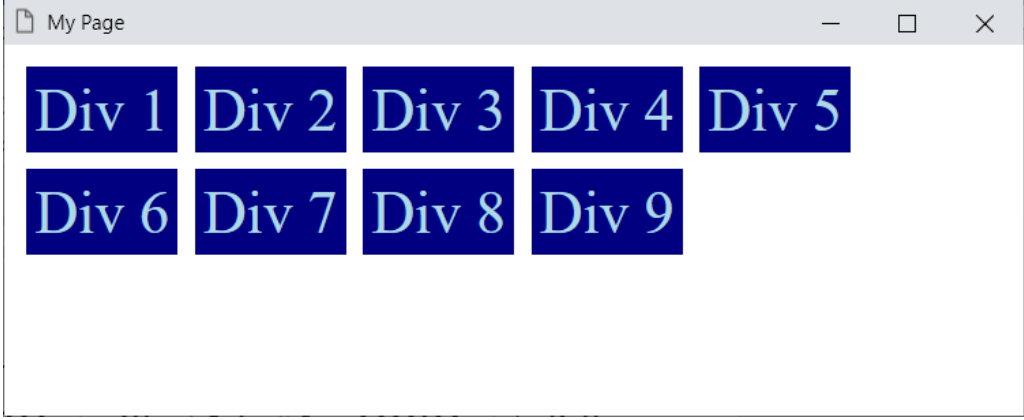
```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
div{
background:navy; color:lightblue; font-size:35px;
margin:5px; padding:5px; text-align:center;
float:left;
}
</style>
</head>
<body>
<div>Div 1</div><div>Div 2</div><div>Div 3</div>
<div>Div 4</div><div>Div 5</div><div>Div 6</div>
<div>Div 7</div><div>Div 8</div><div>Div 9</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

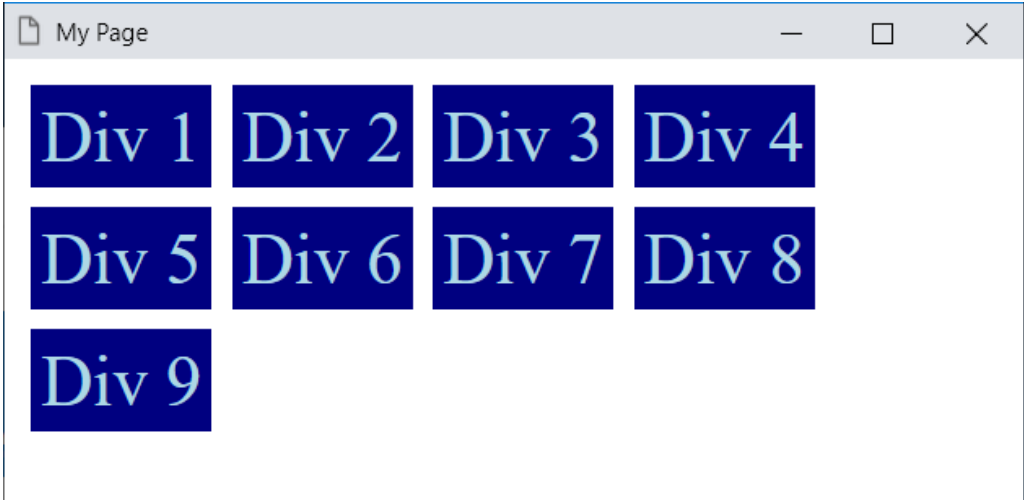


لاحظ أن وضع float:left غير تغيير واضح وكل على جميع الوسوم div، كما هو في الشكل السابق، مع العلم أننا لم نغير الخاصية display، ولم نغير الخاصية width، ولم نغير أي شيء، إلا الخاصية float، حيث أخذت القيمة left.

والآن قم بتصغير نافذة المتصفح، لترى مثل الشكل التالي:



والآن مرة أخرى قم بتصغير نافذة المتصفح، لترى مثل الشكل التالي:



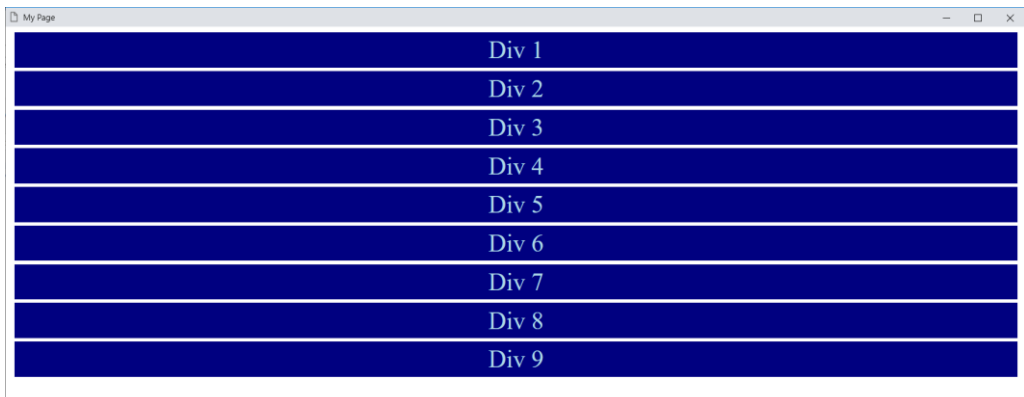
وهذا لتلاحظ المرونة، مع تغيرات الحجم.

float right اتجاه العناصر نحو اليمين

والآن سوف نبدأ بداية عادية، كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
div{
background:navy; color:lightblue; font-size:35px;
margin:5px; padding:5px; text-align:center;
}
</style>
</head>
<body>
<div>Div 1</div><div>Div 2</div><div>Div 3</div>
<div>Div 4</div><div>Div 5</div><div>Div 6</div>
<div>Div 7</div><div>Div 8</div><div>Div 9</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

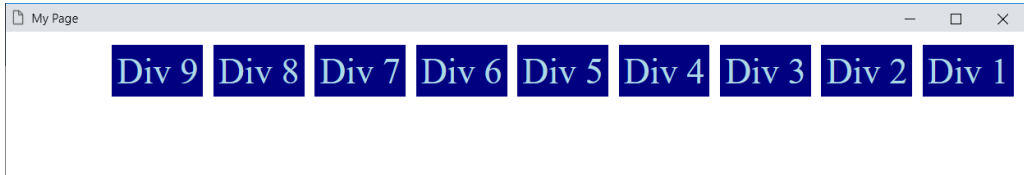


وكما نلاحظ وجود 9 وسوم div، مع تنسيقات عادية، ومعروفة.

والآن **انتبه!** جيداً ، لأننا سوف نعيد نفس المثال، مع زيادة بسيطة جداً .
كود! قم بكتابة الكود التالي ثم احفظه:

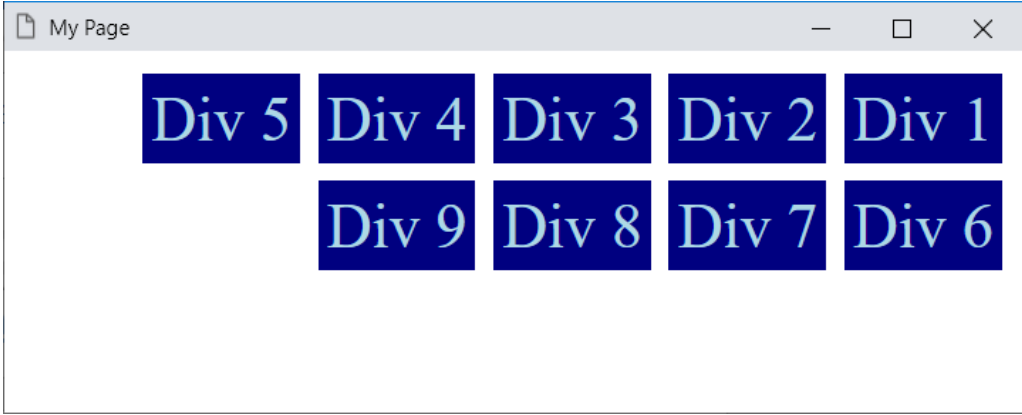
```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
div{
background:navy; color:lightblue; font-size:35px;
margin:5px; padding:5px; text-align:center;
float:right;
}
</style>
</head>
<body>
<div>Div 1</div><div>Div 2</div><div>Div 3</div>
<div>Div 4</div><div>Div 5</div><div>Div 6</div>
<div>Div 7</div><div>Div 8</div><div>Div 9</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

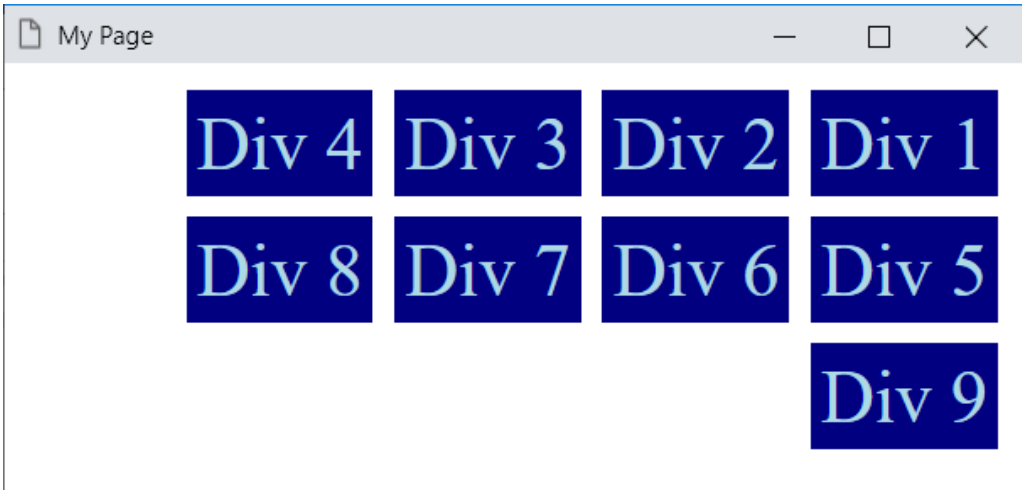


لاحظ أن وضع `float:right` غير تغيير واحد وكل على جميع الوسوم `div`، كما هو في الشكل السابق، مع العلم أننا لم نغير الخاصية `display`، ولم نغير الخاصية `width`، ولم نغير الخاصية `direction`، فقط الخاصية `float`، حيث أخذت القيمة `right`.

والآن قم بتصغير نافذة المتصفح، لترى مثل الشكل التالي:



والآن مرة أخرى قم بتصغير نافذة المتصفح، لترى مثل الشكل التالي:



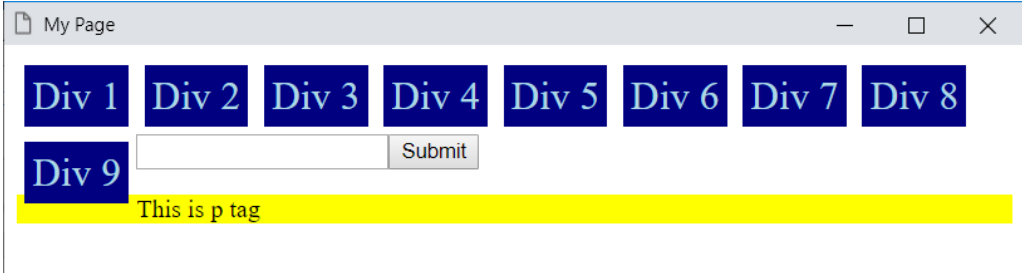
وهذا لتلاحظ المرونة، مع تغيرات الحجم، وتلاحظ أنها تتناسب مع اللغة العربية، لأنها بدأت من اليمين.

ال float مع العناصر الأخرى

مع موضوع ال float، يوجد شيء، ينبغي الانتباه إليه جيداً، وسوف نوضح هذا الشيء، فيما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
p{ background:yellow; }
div{
background:navy; color:lightblue;
font-size:25px;
margin:5px; padding:5px; text-align:center;
float:left;
}
</style>
</head>
<body>
<div>Div 1</div><div>Div 2</div><div>Div 3</div>
<div>Div 4</div><div>Div 5</div><div>Div 6</div>
<div>Div 7</div><div>Div 8</div><div>Div 9</div>
<input><input type="submit">
<p>This is p tag</p>
</body>
</html>
```

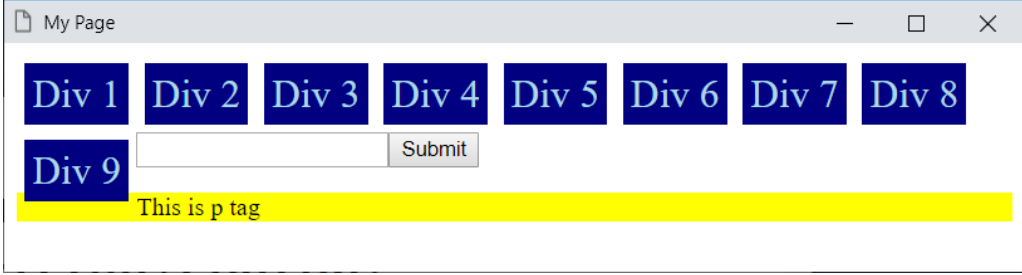
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ وجود وسوم div أولاً، ثم وسميه input، ثم وسم p، ولكن
 الوضع على الصفحة غير مرغوب فيه، فهل وسم br سوف يحل؟
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
p{ background:yellow; }
div{
background:navy; color:lightblue; font-size:25px;
margin:5px; padding:5px; text-align:center;
float:left;
}
</style>
</head>
<body>
<div>Div 1</div><div>Div 2</div><div>Div 3</div>
<div>Div 4</div><div>Div 5</div><div>Div 6</div>
<div>Div 7</div><div>Div 8</div><div>Div 9</div> <br>
<input><input type="submit"><p>This is p tag</p>
</body>
</html>
```

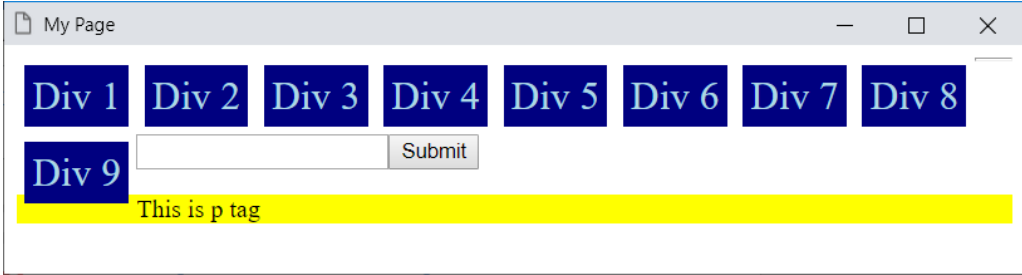
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



وللأسف، النتيجة كما هي، فهل وسم hr سوف يحل؟:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <title>My Page</title>
  <style>
    p{ background:yellow; }
    div{
      background:navy; color:lightblue; font-size:25px;
      margin:5px; padding:5px; text-align:center;
      float:left;
    }
  </style>
</head>
<body>
  <div>Div 1</div><div>Div 2</div><div>Div 3</div>
  <div>Div 4</div><div>Div 5</div><div>Div 6</div>
  <div>Div 7</div><div>Div 8</div><div>Div 9</div> <hr>
  <input><input type="submit"><p>This is p tag</p>
</body>
</html>
```

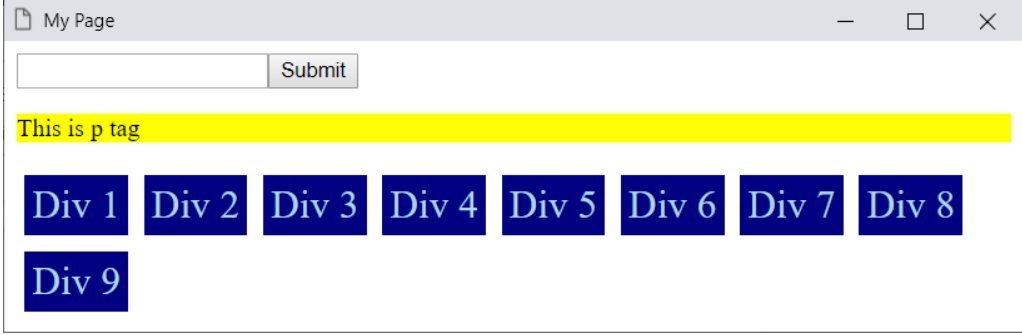

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



للأسف الومر hr لم يحد، ماذا لو وضعناهم قبل وسوم الـ div؟
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <title>My Page</title>
  <style>
    p{ background:yellow; }
    div{
      background:navy; color:lightblue; font-size:25px;
      margin:5px; padding:5px; text-align:center;
      float:left;
    }
  </style>
</head>
<body>
  <input type="submit"><p>This is p tag</p>
  <div>Div 1</div><div>Div 2</div><div>Div 3</div>
  <div>Div 4</div><div>Div 5</div><div>Div 6</div>
  <div>Div 7</div><div>Div 8</div><div>Div 9</div>
</body>
</html>
```

وبءء ءءءءء الصفءة ءظهر كما بالشكل ءءاءى:



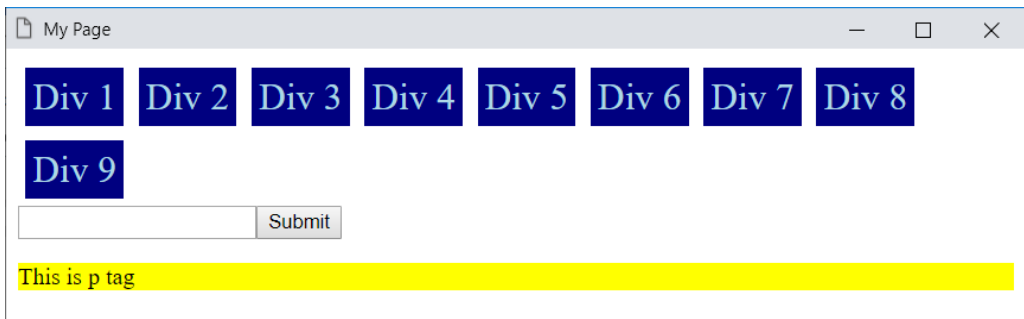
والآن ىءبءء لنا، أنه ما قبل ال float، لا ىكون فءه مشكلء، ولكء مشكلءة ما بعء ال float، لاءالء قائمء، والءل إن شاء الله سىكون فء ءرس عزء ال float ب clear القاءم، فرءاء مءابءة الءروس.

عزل ال float ب clear

يمكنك عزل ال float مع اليسار كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
p{ background:yellow; }
div{
background:navy; color:lightblue; font-size:25px;
margin:5px; padding:5px; text-align:center;
float:left;
}
</style>
</head>
<body>
<div>Div 1</div><div>Div 2</div><div>Div 3</div>
<div>Div 4</div><div>Div 5</div><div>Div 6</div>
<div>Div 7</div><div>Div 8</div><div>Div 9</div>
<p style="clear:left;"></p>
<input><input type="submit"><p>This is p tag</p>
</body>
</html>
```

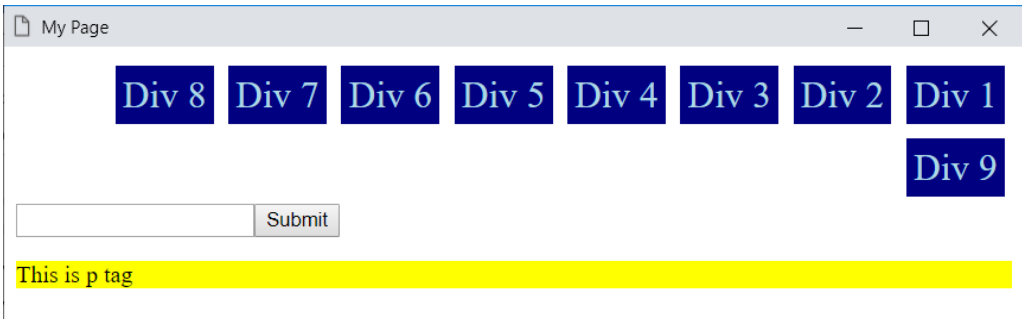
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ من الشكل السابق، أنه تم وضع وسم p فارغ، به خاصية clear
تم ضبطها على القيمة left، لعزل ال float:left، ولكه هذا العزل
لا يصلح مع اليمين، فيملك العزل من اليمين، بالكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
p{ background:yellow; }
div{
background:navy; color:lightblue; font-size:25px;
margin:5px; padding:5px; text-align:center;
float:right;
}
</style>
</head>
<body>
<div>Div 1</div><div>Div 2</div><div>Div 3</div>
<div>Div 4</div><div>Div 5</div><div>Div 6</div>
<div>Div 7</div><div>Div 8</div><div>Div 9</div>
<p style="clear:right;"></p>
<input><input type="submit"><p>This is p tag</p>
</body>
</html>
```

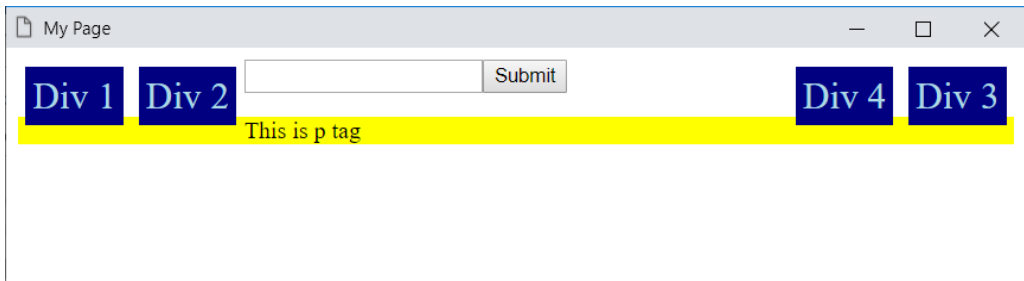
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ من الشكل السابق، أنه تم وضع وسم p فارغ، به خاصية clear تم ضبطها على القيمة right، لعزل ال float:right، ولكنه في بعض الأحيان نريد عمل صفحة مثل الكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
p{ background:yellow; }
div{
background:navy; color:lightblue; font-size:25px;
margin:5px; padding:5px; text-align:center;
}
.d1, .d2{ float:left; }
.d3, .d4{ float:right; }
</style>
</head>
<body>
<div class="d1">Div 1</div><div class="d2">Div 2</div>
<div class="d3">Div 3</div><div class="d4">Div 4</div>
<input><input type="submit"><p>This is p tag</p>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

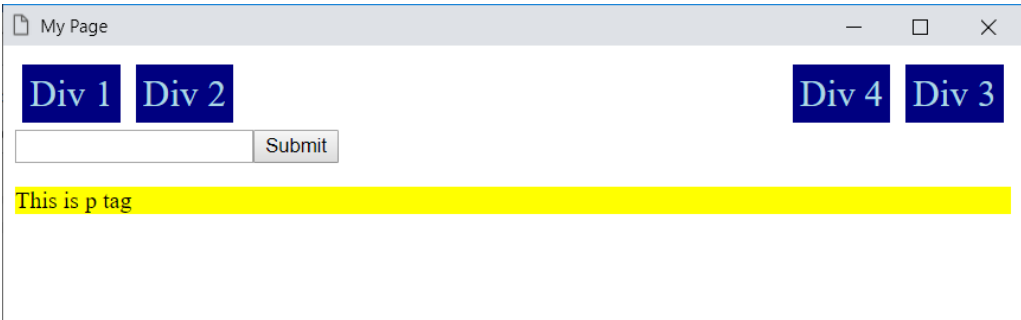


لاحظ أنه يوجد float نحو اليمين، ونحو اليسار.

انتبه! فيمكنك أن تقوم بعمل أي float أي كان يمينه أو يساره، كما يلي.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
p{ background:yellow; }
div{
background:navy; color:lightblue; font-size:25px;
margin:5px; padding:5px; text-align:center;
}
.d1, .d2{ float:left; }
.d3, .d4{ float:right; }
</style>
</head>
<body>
<div class="d1">Div 1</div><div class="d2">Div 2</div>
<div class="d3">Div 3</div><div class="d4">Div 4</div>
<p style="clear:both;"></p>
<input><input type="submit"><p>This is p tag</p>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم استخدام clear:both لعزل الكود.

الصدوق box

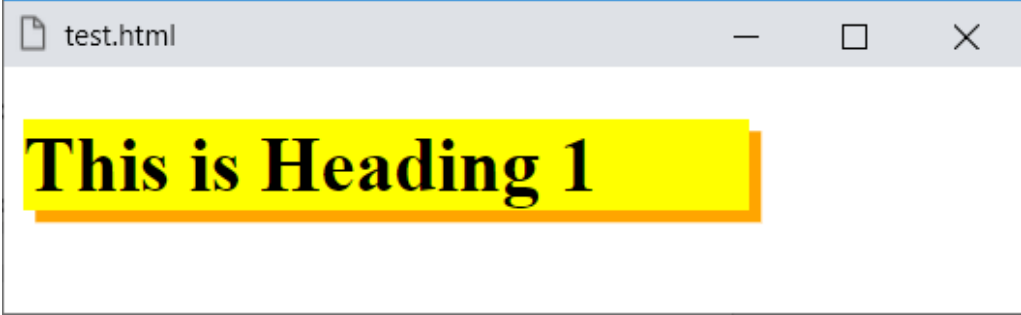
ظل الصدوق box-shadow

من قبل قمننا بعمل ظل للنص، والآن نريد انشاء ظل للعناصر نفسها، وسوف تكون بنفس الطريقة، التي تعاملنا معها من قبل، أثناء تجربة ظل النص، في بعده وتغيره.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      div, h1{
        background-color:yellow;
        width:300px;
        box-shadow:5px 5px orange;
      }
    </style>
  </head>
  <body>
    <h1>This is Heading 1</h1>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

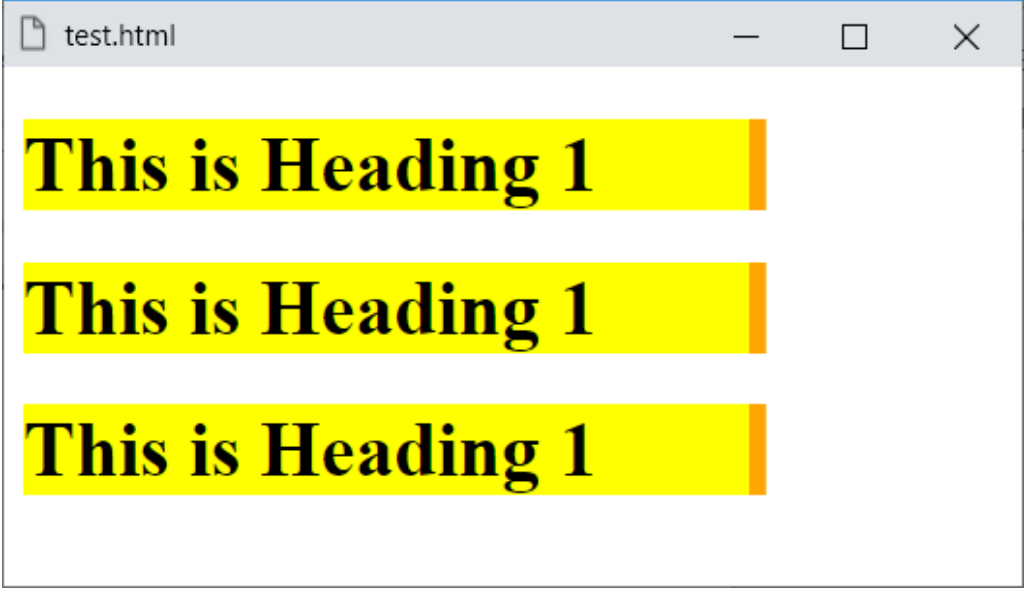


لاحظ وجود ظل باللون البرتقالي، والتحرك أفقياً بمقدار 5px، ثم التحريك رأسياً بمقدار 5px.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
div, h1{
background-color:yellow; width:300px;
box-shadow:7px 0px orange;
}
</style>
</head>
<body>
<h1>This is Heading 1</h1>
<h1>This is Heading 1</h1>
<h1>This is Heading 1</h1>
</body>
</html>
```


وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ وجود ظل برتقالي، ويتحرك أفقياً بمقدار 7px ولا يتحرك رأسياً، فيظهر بهذا الشكل الجميل.

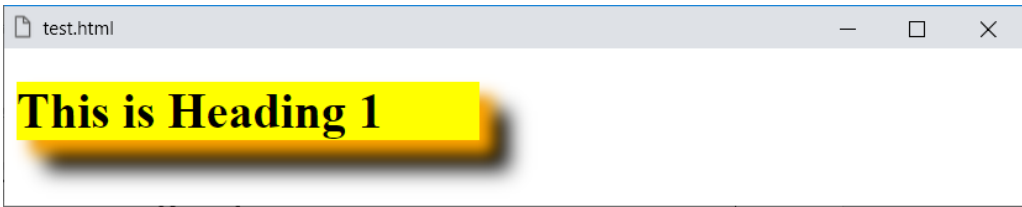
انتبه! فإنه يمكن ضبط التشويش، كما فعلنا مع ظل النص من قبل، وأيضاً يمكن ومنح أكثر من ظل بشكل مستقل لكل واحد.

والآن سوف نقوم بعمل مثال به فطال، الأول برتقالي، والثاني أسود، مع الاختلاف بين كل منهم، في الحركة والتشويش واللون، وذلك كما يلي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
div, h1{
background-color:yellow;
width:300px;
box-shadow:9px 9px 15px orange,
19px 19px 20px black;
}
</style>
</head>
<body>
<h1>This is Heading 1</h1>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

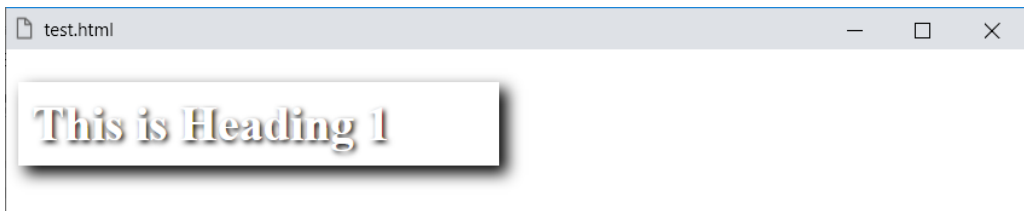


لاحظ أن الظل البرتقالي تحرك أفقياً ورأسياً بمقدار 9px، وتشوش بمقدار 15px، أما الظل الأسود تحرك أفقياً ورأسياً بمقدار 19px وتم التشويش عليه بمقدار 20px، كما هو واضح في الصورة.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    h1{
      color:#fff;
      width:300px; padding:9px;
      text-shadow:1.5px 1.5px 5px black;
      box-shadow:5px 5px 15px black;
    }
  </style>
</head>
<body>
  <h1>This is Heading 1</h1>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

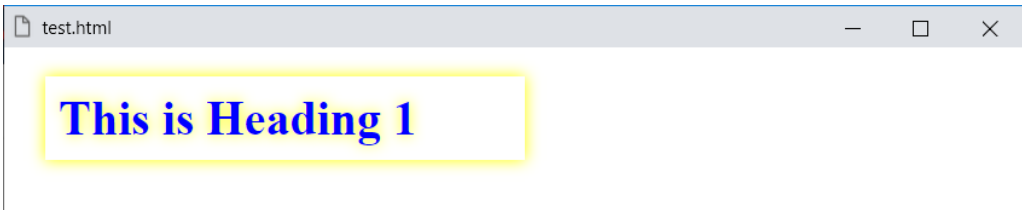


انتبه! ولاحظ أنه لو تم حذف الخاصية text-shadow والخاصية box-shadow له يكون أي شيء ظاهر على الصفحة، لأن الـ div لونه أبيض، والنص لونه أبيض، وكذلك الصفحة الأساسية، ولكن مع الظلال، أصبح شكله جذاب وجميل.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1{
color:#00f; margin:19px;
width:300px; padding:9px;
text-shadow:0 0 15px yellow;
box-shadow:0 0 15px yellow;
}
</style>
</head>
<body>
<h1>This is Heading 1</h1>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

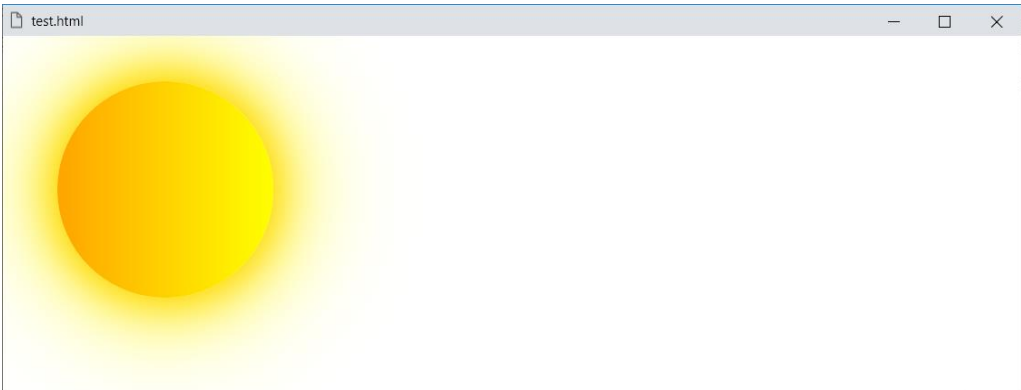


لاحظ أنه مع عدم التحريك، وزيادة التشويش، تم عمل استنارة حول العنصر، وحول النص المكتوب.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
div{
width:150px; height:150px; margin:40px; padding:20px;
background:linear-gradient(to right,orange,yellow);
border-radius:100px;
box-shadow:0 0 55px gold,
           0 0 100px gold,
           0 0 200px yellow;
}
</style>
</head>
<body>
<div></div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

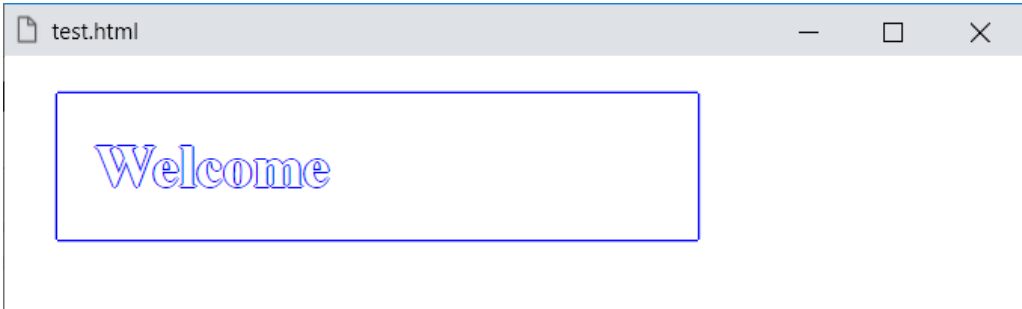


لاحظ أنه تم رسم شمس، عن طريق radius الإطار، والتدرج اللوني، والظل المتعدد، مع التشويش الزائد عليه.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1{
width:300px; color:#fff;
margin:20px; padding:20px;
box-shadow:0 1px blue, 1px 0 blue,
-1px 0 blue, 0 -1px blue;
text-shadow:0 1px blue, 1px 0 blue,
-1px 0 blue, 0 -1px blue;
}
</style>
</head>
<body>
<h1>Welcome</h1>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



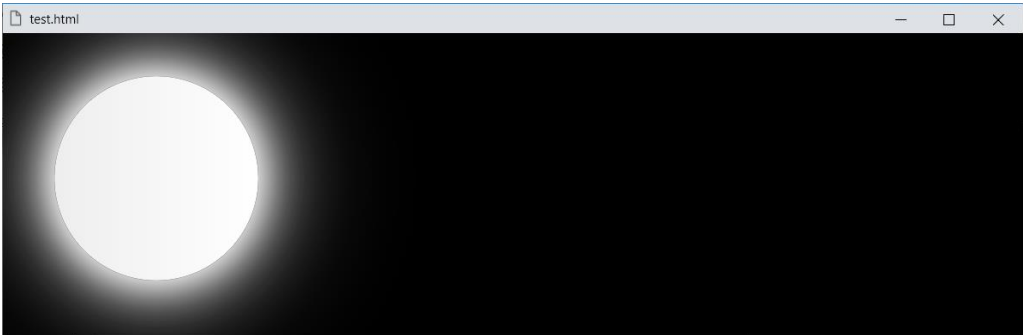
انتبه! فإن هذا ليس إطار، بل إنه ظل، وتم إنشاؤه بواسطة ثمانية من الظلال، أربعة للوسم `h1` بالخاصية `box-shadow`، وأربعة للنص بالخاصية `text-shadow`.

انتبه! فإنه يتم بأربعة خطوات متتالية، للوصول إلى الإحاطة بالعنصر أو النص من الاتجاهات الأربعة (اليمين، اليسار، الأسفل، الأعلى)، أولاً: يتم التحريك الرأسى إلى الأسفل بالموجب بـ `1px`، ثانياً: يتم التحريك الأفقى إلى اليمين بالموجب بـ `1px`، ثالثاً: يتم التحريك الرأسى إلى الأعلى بالسالب بالقيمة `-1px`، رابعاً: يتم التحريك الأفقى إلى اليسار بالسالب بالقيمة `-1px`، وبذلك يتم الإحاطة بالاتجاهات الأربعة.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
body{ background-color:#000; }
div{
width:150px; height:150px; margin:40px; padding:20px;
background:linear-gradient(to right,#eee,white);
border:0.20px ridge #eee;
border-radius:100px;
box-shadow:0 0 55px white,
           0 0 100px white,
           0 0 200px #eee;
}
</style>
</head>
<body>
<div></div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

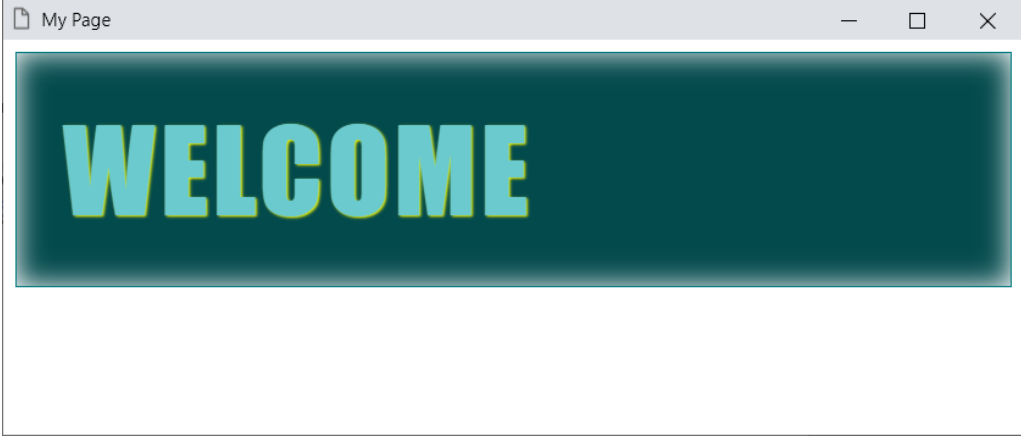


لاحظ أنه تم رسم قمر، عن طريق radius الإطار، والتدرج اللوني، والظل المتعدد، مع التشويش الزائد عليه، وإطار خفيف.

انتبه! فإنه يمكن قلب الظل الى الداخل بدلاً من الخارج عن طريق القيمة inset كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Page</title>
    <style>
      div{
        background:#034a4d; color:#6bcace;
        padding:30px; letter-spacing:3px;
        font-family:impact; font-size:75px;
        box-shadow:inset 0 0 30px white;
        text-shadow:1px 1px 2px yellow;
        border:1px solid #007b80;
      }
    </style>
  </head>
  <body>
    <div>WELCOME</div>
  </body>
</html>
```

وبءء ءشءعل الصءفة ءظهء كما بالشكل ءءلى:



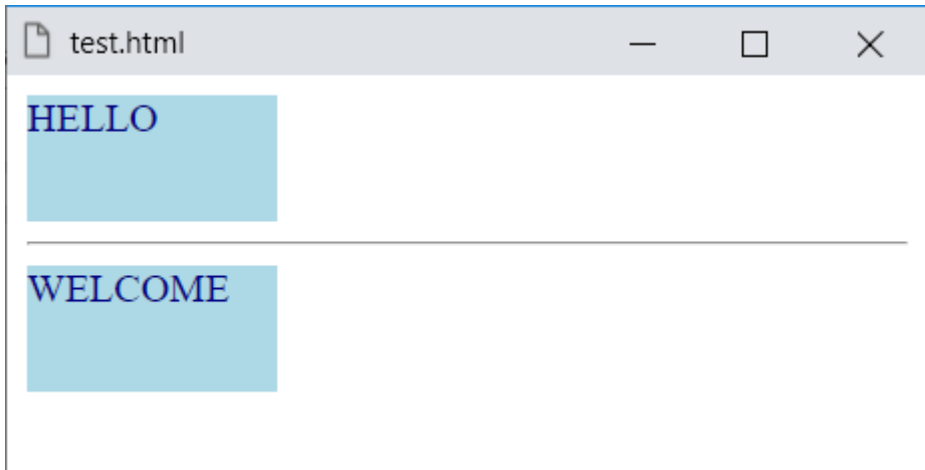
لاءظ أنه ءم عمل ظل ءاءلى باللون الأبيض عه طءرق القفمة `.inset`.

تجسيم الصندوق box-sizing

عند انشاء عنصر على صفحة الويب، وليك مثلاً الوسم div، ثم تم ضبط العرض width، والطول height، والحشو padding، والإطار border، لذلك الوسم، فإن العرض سوف يكبر تلقائياً، وللتجربة أولاً ننشئ وسم div لهم نفس الخصائص **بالكود!** التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{ background:lightblue; color:navy; width:100px; height:50px; }
</style>
</head>
<body>
  <div id="d1">HELLO</div><hr>
  <div id="d2">WELCOME</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

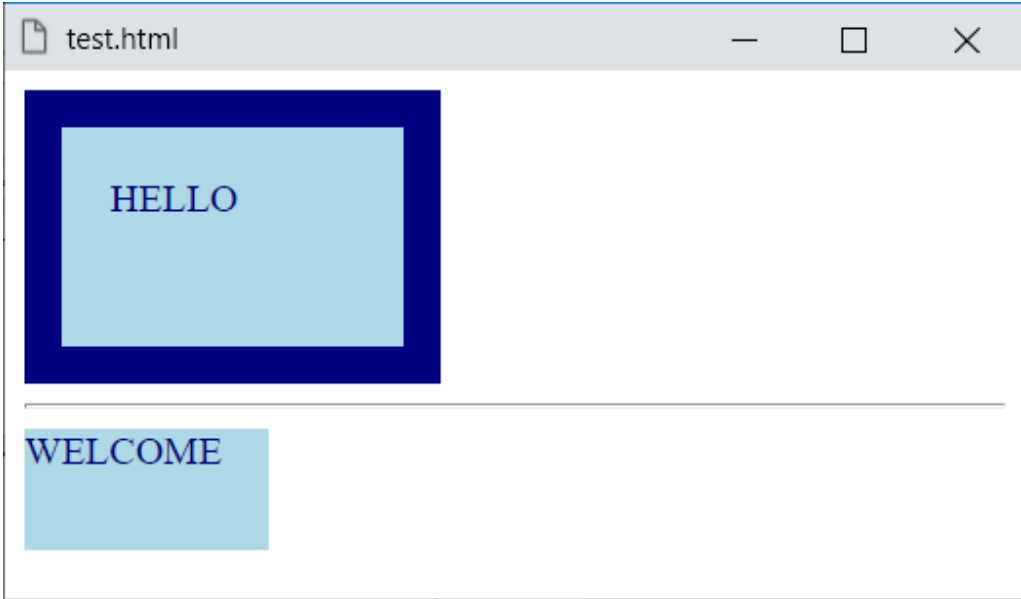


لاحظ أنهم مثل بعضهم، والآن **انتبه!** فسوف نغير للعنصر d1 الإطار والحشو فقط، دون المساس بالعنصر d2، بالمثال التالي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{ background:lightblue; color:navy; width:100px; height:50px; }
  #d1{ padding:20px; border:15px solid navy; }
</style>
</head>
<body>
  <div id="d1">HELLO</div><hr>
  <div id="d2">WELCOME</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

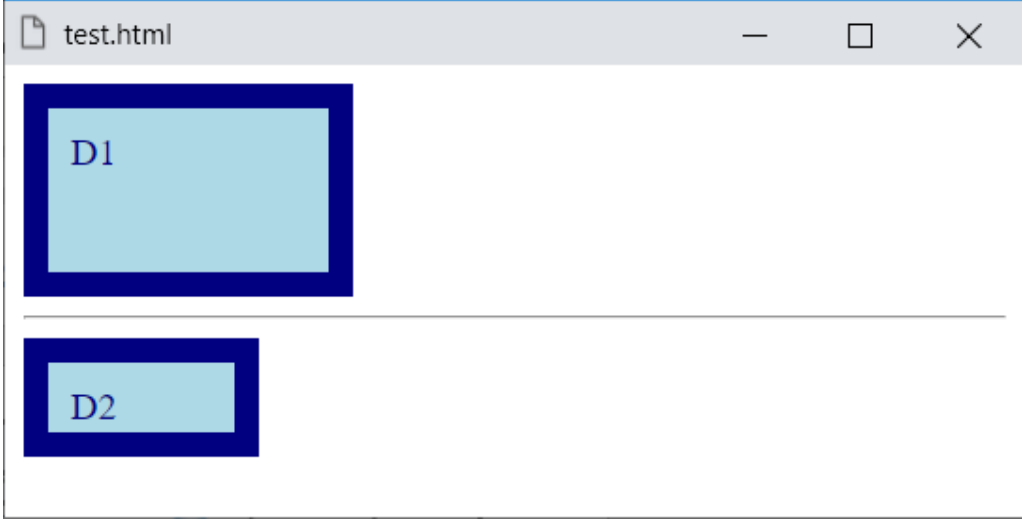


لاحظ في الكود أن العرض والطول لكك منعما لم يتغير، فقط للوسم d1، تم تحديد الإطار والحشو، فزاد حجم العنصر.

والآن يأتي دور الخاصية `box-sizing`، والتي تأخذ قيمة افتراضية `content-box`، أي يزيد الحشو والإطار، والقيمة الأخرى لها هي `border-box`، أي يختصب قيم الحشو والإطار من الحجم الأصلي، وسنبينهم إن شاء الله **بالكود!** التالي:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      div{
        background:lightblue; color:navy;
        border:10px solid navy;
        width:100px; height:50px;
        padding:10px;
      }
      #d1{ box-sizing:content-box; }
      #d2{ box-sizing:border-box; }
    </style>
  </head>
  <body>
    <div id="d1">D1</div><hr>
    <div id="d2">D2</div>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

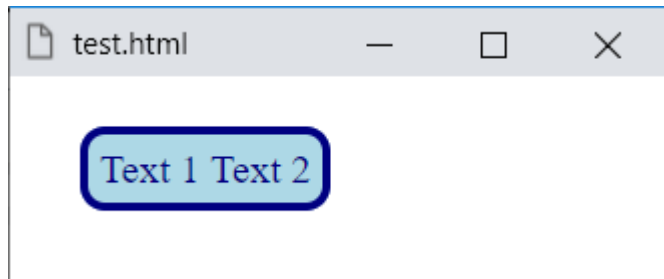


لاحظ أن هناك وسميه `div`، لهم نفس الخلفية، ونفس اللون، ونفس الإطار، ونفس العرض، ونفس الطول، ونفس الحشو، ولكن مع القيمة `border-box`، تغير الوسم `d2`.

ديكور السطر للصندوق `box-decoration-break` للتوضيح قم بكتابة الكود! التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    body{ padding:20px; }
    span{
      background:lightblue; color:navy;
      border:3px solid navy;
      border-radius:10px;
      width:200px; height:50px;
      padding:5px;
    }
  </style>
</head>
<body>
  <span id="s1">Text 1 Text 2</span>
</body>
</html>
```

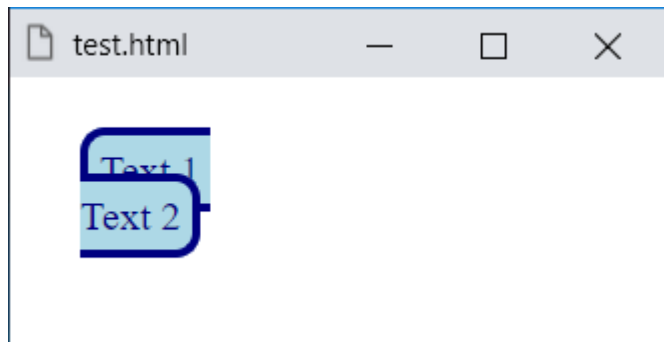
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



والآن نجرّب ونضع الوسم br داخل الوسم span، بالكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
body{ padding:20px; }
span{
background:lightblue; color:navy;
border:3px solid navy;
border-radius:10px;
width:200px; height:50px;
padding:5px;
}
</style>
</head>
<body>
<span id="s1">Text 1 <br> Text 2</span>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

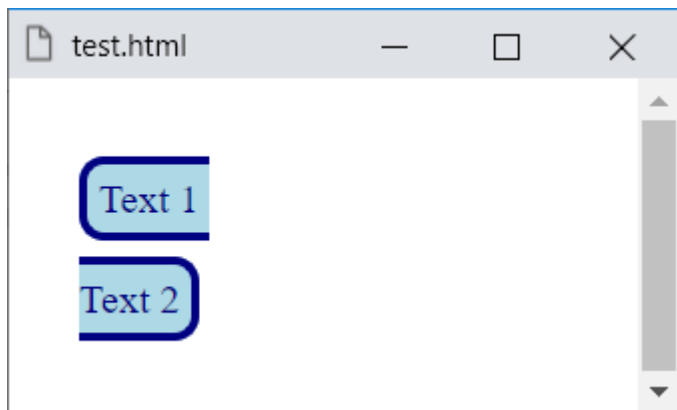


لاحظ أنه تم تقطيع النصوص مع الـ .break

والآن سوف نقوم بزيادة الـ `line-height`، لإظهار كل سطر وانحناء:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
body{ padding:20px; }
span{
background:lightblue; color:navy;
border:3px solid navy;
border-radius:10px;
width:200px; height:50px;
padding:5px; line-height:2.5;
}
</style>
</head>
<body>
<span id="s1">Text 1 <br> Text 2</span>
</body>
</html>
```

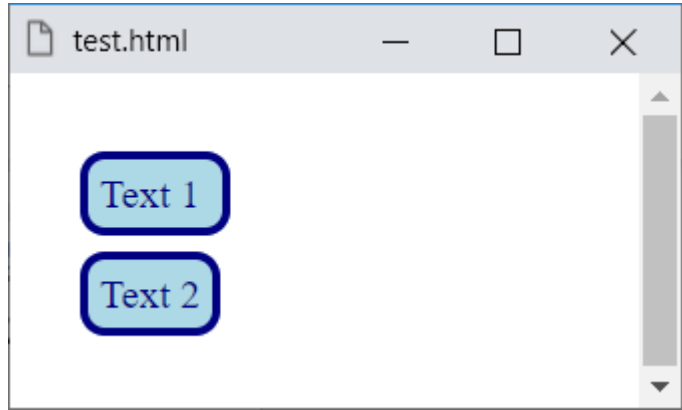
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



رجاء **انتبه!**، فإنه هنا تم التقطيع بيه ال break لأن القيمة الافتراضية للخاصية box-decoration-break هي slice، أي قطع، ولكنه يوجد قيمة clone لإلغاء التقطيع كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
body{ padding:20px; }
span{
background:lightblue; color:navy;
border:3px solid navy;
border-radius:10px;
width:200px; height:50px;
padding:5px; line-height:2.5;
-webkit-box-decoration-break: clone;
box-decoration-break: clone;
}
</style>
</head>
<body>
<span id="s1">Text 1 <br> Text 2</span>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

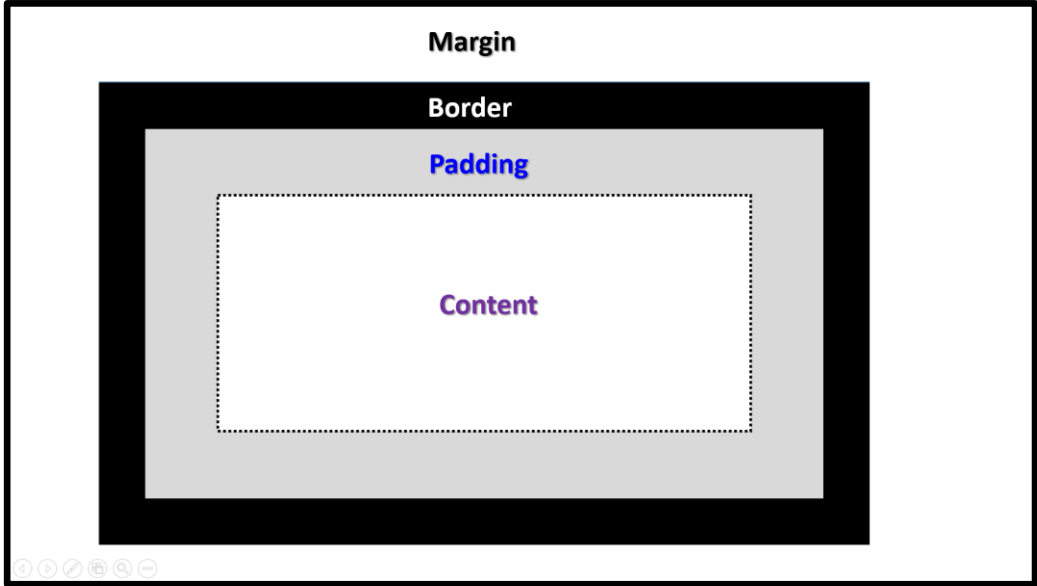


لاحظ أنه تم استخدام بادئة المتصفح -webkit- في الكود.

لاحظ أن القيمة clone، قامت بإلغاء التقطيع بين كل .break.

الصدوق المنظم Box Model

فكرة الصدوق المنظم، أو نموذج المحتوى، وهو ال box model، هي فكرة بسيطة وجميلة في نفس الوقت، حيث تستخدم بعض ما تعلمناه بفضل الله، لتصنع مربع تضع فيه محتوى بشكل أجمل من الشكل العادي، حيث نستخدم، الهامش margin، والإطار border، والحشو padding، والإطار الخارجي outline، والظل shadow، في إنشاء هذا الصدوق الجميل، والشكل التالي يعبر عنه ال box model:

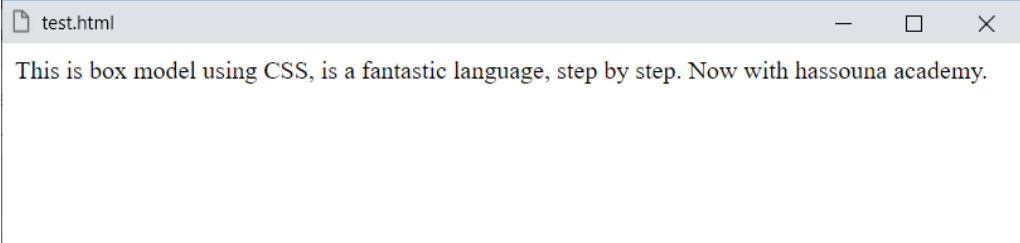


كما تلاحظ في الشكل السابق، حدود ال margin وال border وال padding وال content، وأثناء التطبيق إن شاء الله سوف نضيف إضافات أخرى، على مجموعة من المراحل كما يلي.

أولاً: وضع المحتوى داخل div بدون أي تنسيق
سوف نبدأ بالكوود! التالي، فرجاء اكتبه ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
<div>
  This is box model using CSS, is a fantastic language,
  step by step. Now with hassouna academy.
</div>
</body>
</html>
```

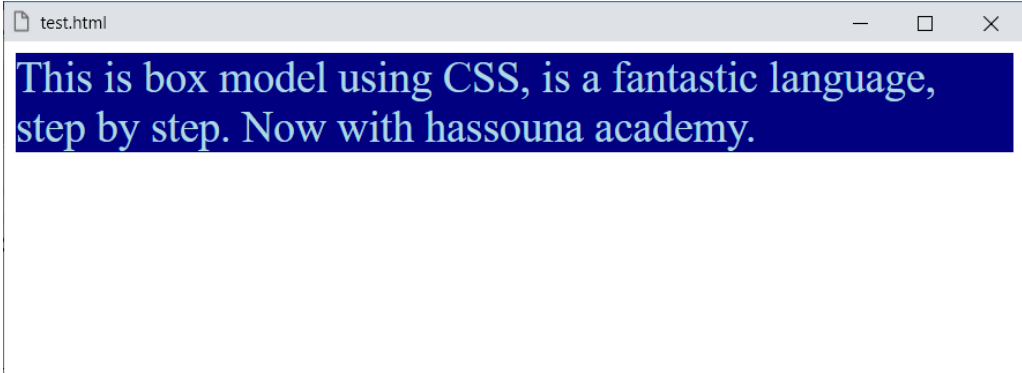
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



ثانياً: لوه النص والخلفية وحجم الخط
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{
    background:navy; color:lightblue; font-size:30px;
  }
</style>
</head>
<body>
<div>
  This is box model using CSS, is a fantastic language,
  step by step. Now with hassouna academy.
</div>
</body>
</html>
```

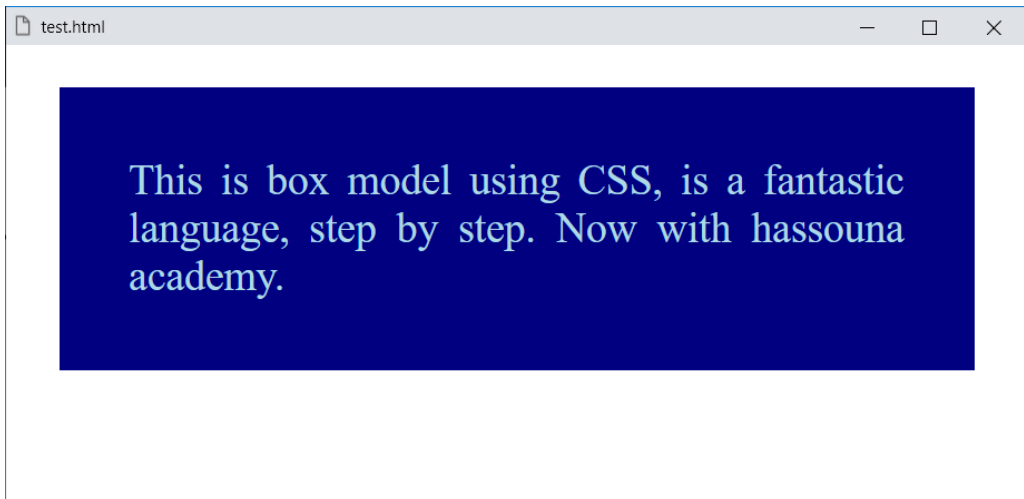
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



ثالثاً : العاشق والحشو ومحاذاة النص
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{
    background:navy; color:lightblue; font-size:30px;
    margin:30px; padding:50px; text-align:justify;
  }
</style>
</head>
<body>
<div>
  This is box model using CSS, is a fantastic language,
  step by step. Now with hassouna academy.
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

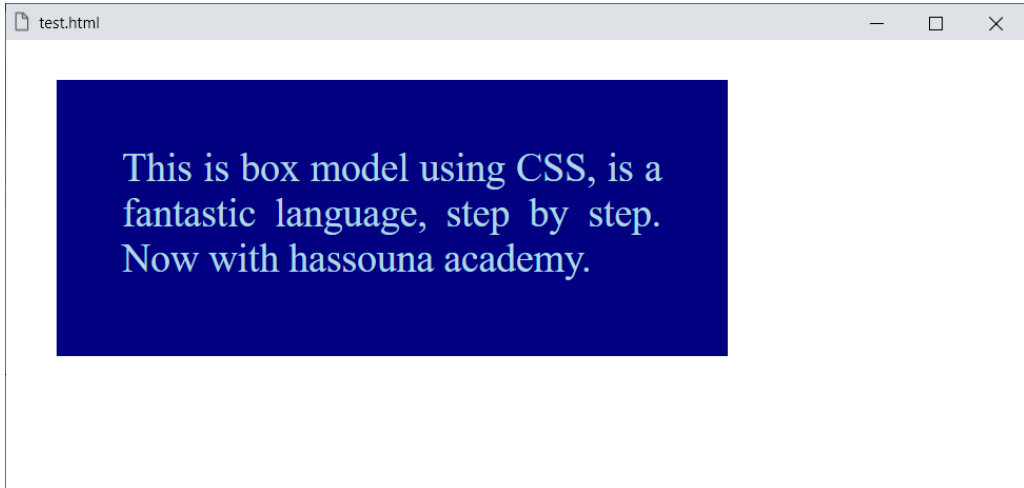


رابعاً: الطول والعرض

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{
    background:navy; color:lightblue; font-size:30px;
    margin:30px; padding:50px; text-align:justify;
    width:410px; height:110px;
  }
</style>
</head>
<body>
<div>
  This is box model using CSS, is a fantastic language,
  step by step. Now with hassouna academy.
</div>
</body>
</html>
```

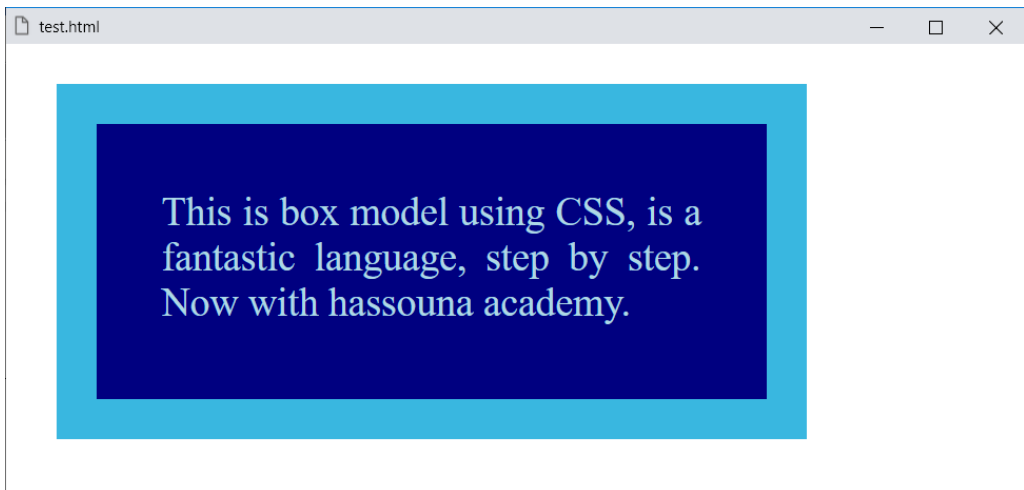
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



خامساً: الإطار
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
div{
background:navy; color:lightblue; font-size:30px;
margin:30px; padding:50px; text-align:justify;
width:410px; height:110px;
border:30px solid #39b7e0;
}
</style>
</head>
<body>
<div>
This is box model using CSS, is a fantastic language,
step by step. Now with hassouna academy.
</div>
</body>
</html>
```

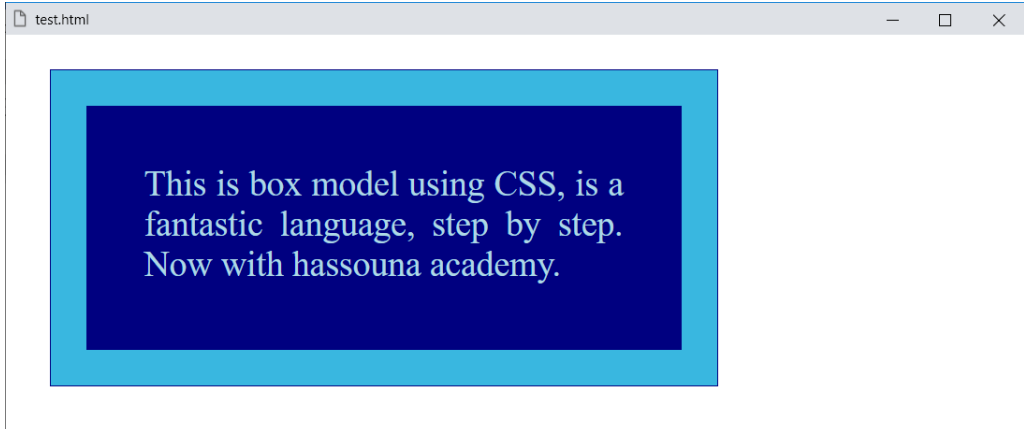
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



سادساً: الإطار الخارجي
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
div{
background:navy; color:lightblue; font-size:30px;
margin:30px; padding:50px; text-align:justify;
width:410px; height:110px;
border:30px solid #39b7e0;
outline:1px solid navy;
}
</style>
</head>
<body>
<div>
This is box model using CSS, is a fantastic language,
step by step. Now with hassouna academy.
</div>
</body>
</html>
```

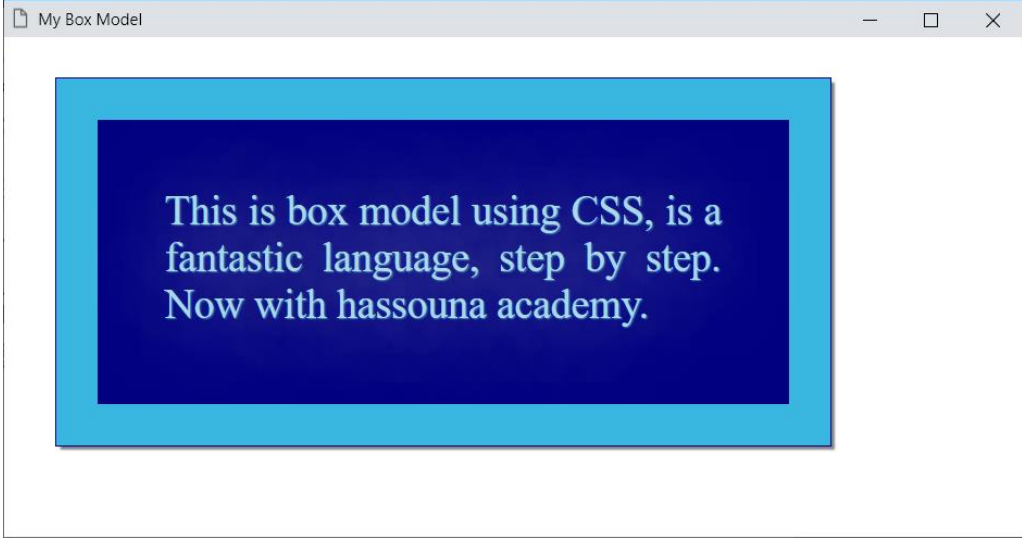
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



سابقاً: تظليل الصندوق والنص
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>
  My Box Model
</title>
<style>
  div{
    background:navy; color:lightblue; font-size:30px;
    margin:30px; padding:50px; text-align:justify;
    width:410px; height:110px;
    border:30px solid #39b7e0;
    outline:1px solid navy;
    box-shadow:3px 3px 1px gray;
    text-shadow:0 0 120px #fff,
                1px 1px 0.5px rgba(57,183,224,0.7);
  }
</style>
</head>
<body>
<div>
  This is box model using CSS, is a fantastic language,
  step by step. Now with hassouna academy.
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



وكما هو واضح أمامنا بفضل الله شكل جميل، بدلاً من العمل العشوائي،
تم ترتيب التنسيق لون وخلفية وحجم **بالكود!** التالي:

```
background:navy; color:lightblue; font-size:30px;
```

ثم هامش وحشو ومحاذاة **بالكود!** التالي:

```
margin:30px; padding:50px; text-align:justify;
```

ثم طول وعرض **بالكود!** التالي:

```
width:410px; height:110px;
```

ثم إطار **بالكود!** التالي:

```
border:30px solid #39b7e0;
```

ثم إطار خارجي **بالكود!** التالي:

```
outline:1px solid navy;
```

ثم الظلال **بالكود!** التالي:

```
box-shadow:3px 3px 1px gray;  
text-shadow:0 0 120px #fff,  
1px 1px 0.5px rgba(57,183,224,0.7);
```

مراحل تطور الصندوق

الشكل التالي يعبر عن مراحل تطور ال box model :

1
This is box model using CSS, is a fantastic language, step by step. Now with hassouna academy.

2
This is box model using CSS, is a fantastic language, step by step. Now with hassouna academy.

3
This is box model using CSS, is a fantastic language, step by step. Now with hassouna academy.

4
This is box model using CSS, is a fantastic language, step by step. Now with hassouna academy.

5
This is box model using CSS, is a fantastic language, step by step. Now with hassouna academy.

6
This is box model using CSS, is a fantastic language, step by step. Now with hassouna academy.

7
This is box model using CSS, is a fantastic language, step by step. Now with hassouna academy.

انظر الى رقم 1 ثم رقم 7، ثم انظر الى مرحلة بعد مرحلة، وستعرف أن اللغة CSS لها دور كبير في تجميل تصميّات الويب بفضل الله.

المرونة (Flex Box) Flexibility

موضوع المرونة في لغة CSS، شيق جداً وفي غاية الروعة، حيث يجعل مصمم الويب، يصنع حاوية، ثم يضع بها بعض العناصر، ثم ينسق الحاوية وعناصرها كيفما شاء بفضل الله، وسيلكون التفصيل إن شاء الله كما يلي.

انتبه! لأن في هذا الموضوع الخصائص تكمل بعضها، ولكل واحدة تأثير.

الظهور المره `display flex`

سوف نقوم بعمل حاوية عادية، بها بعض العناصر، **بالكود!** التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:90%; height:200px;
  }
  div.child1{ background:#0e0; }
  div.child2{ background:#e00; }
  div.child3{ background:#00e; }
</style>
</head>
<body>
<div class="parent">
  <div class="child1">Div 1</div>
  <div class="child2">Div 2</div>
  <div class="child3">Div 3</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



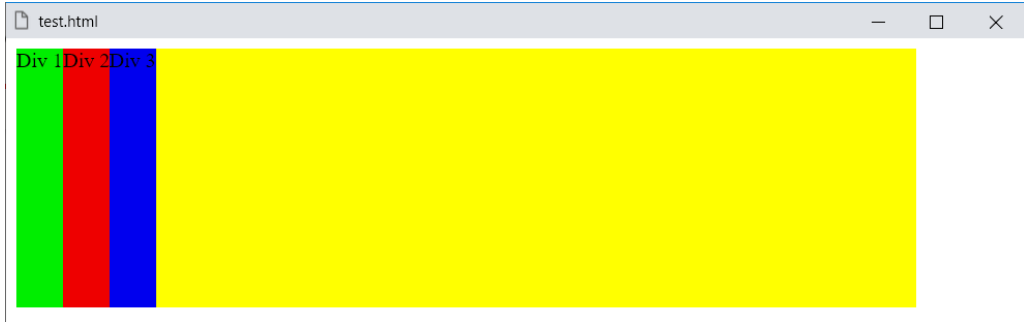
لاحظ أن الشكل بسيط جداً، فهو مجرد وسم div حاوية باللون الأصفر وتأخذ كلاس باسم parent، وثلاث وسم div فرعية بداخل الأصل، فالأول أخضر بكلاس child1، والثاني أحمر بكلاس child2، والثالث أزرق بكلاس child3، مع العلم أن الـ div الأصل يأخذ عرض بالنسبة 90% وطول بمقدار 200px.

والآن سوف نستخدم الخاصية display واعطاؤها القيمة flex لجعل الحاوية parent، مرنة، وسوف نلاحظ الفرق ما بين الشكل السابق، والتمويه التالي، ولكنه **انتبه!** أن الـ div الأصلي يأخذ عرض 90%، ويأخذ طول 200px، لأن هذا سيؤثر على التمويه التالي كما سنوضح إن شاء الله.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:90%; height:200px;
    display:flex;
  }
  div.child1{ background:#0e0; }
  div.child2{ background:#e00; }
  div.child3{ background:#00e; }
</style>
</head>
<body>
<div class="parent">
  <div class="child1">Div 1</div>
  <div class="child2">Div 2</div>
  <div class="child3">Div 3</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

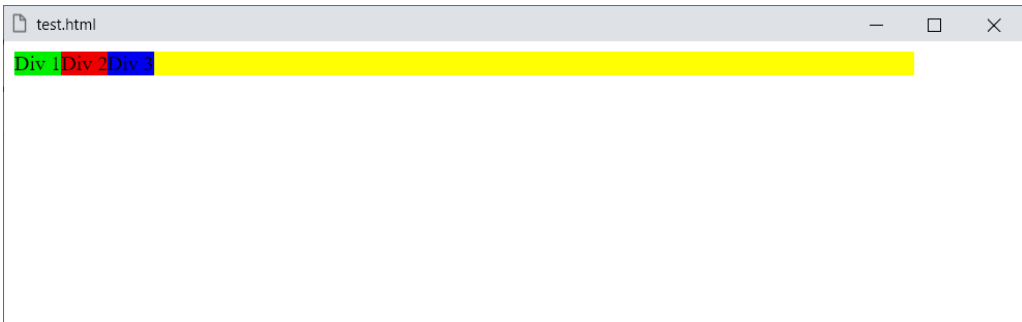


لاحظ أن كل div فرعي، موجود بالحاوية parent، أخذ عرض بقدر ما يحتويه، وأخذ طول بكامل طول ال parent.

والآن سنجرب، بدون الخاصية height، بالكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:90%;
    display:flex;
  }
  div.child1{ background:#0e0; }
  div.child2{ background:#e00; }
  div.child3{ background:#00e; }
</style>
</head>
<body>
<div class="parent">
  <div class="child1">Div 1</div>
  <div class="child2">Div 2</div>
  <div class="child3">Div 3</div>
</div>
</body>
</html>
```

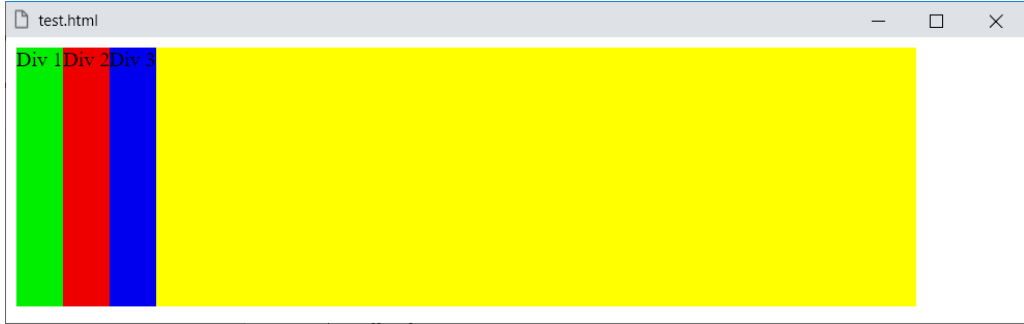
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



والآن بدون الطول لعنصر ال parent، أخذت كل ال child طول بقدر ما فيهما فقط، ومع التمارين القادمة سنتفهم أكثر إن شاء الله.

مءاذاة العناصء align-items

هذه الخاصفة، سوف ءءل لك اللءز، فف ءعل الوسوم div ءأءه الطول بالءامل، الءف ءم عمله مء قفء، مءل الشءل ءالف:



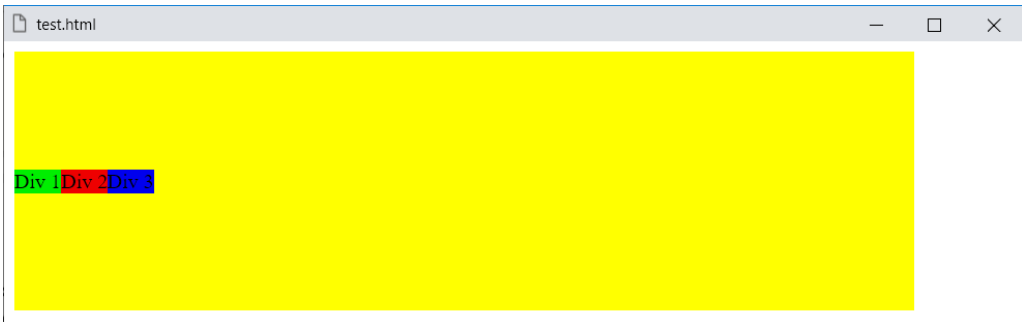
وكما هو واءءل مء الصورة، أه الوسوم div، أءءه طول، بءو أه نءصمه لها، وءلك لأن الخاصفة align-items ءأءه قفمة افءراضفة وهف stretch، فءمءء كل ال ءءل ال parent.

ءءاء **انءبه!** لأن الخاصفة align-items ءءب لنا ءءلعب مء العناصء مء ءاءفة الرأسفة، ومعنف هذا أنه لو ءءءت لك أننا سوف نبءأ مء الأول، فإن هذا فعنف مء أعلف، ولو ءءءت لك أننا سوف نبءأ مء الآخر، فإن هذا فعنف مء الأسفل، ولو ءءءت لك أننا سوف نءءع العناصء فف الوسط، فإن هذا فعنف ما ببه الأعلى والأسفل، فلك هذا معنف رأسفاً، ءءف لا ءءلءل علفك الأمور فرءاء **انءبه!** ءءفاً.

يمكنك جعل العناصر في الوسط رأسياً ، بالكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:90%; height:200px;
    display:flex; align-items:center;
  }
  div.child1{ background:#0e0; }
  div.child2{ background:#e00; }
  div.child3{ background:#00e; }
</style>
</head>
<body>
<div class="parent">
  <div class="child1">Div 1</div>
  <div class="child2">Div 2</div>
  <div class="child3">Div 3</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

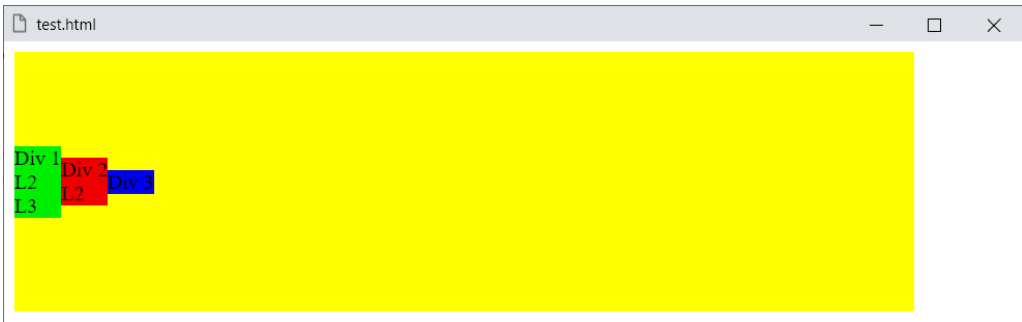


لاحظ أن القيمة center للخاصية align-items جعلت العناصر في المنتصف رأسياً .

انتبه! ولا تتعجل في الحكم على الأمور، فإن الخاصية `align-items`،
تعطي طول على قدر ما بداخل العناصر، ما لم تستخدم القيمة `stretch`.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:90%; height:200px;
    display:flex; align-items:center;
  }
  div.child1{ background:#0e0; }
  div.child2{ background:#e00; }
  div.child3{ background:#00e; }
</style>
</head>
<body>
<div class="parent">
  <div class="child1">Div 1<br>L2<br>L3</div>
  <div class="child2">Div 2<br>L2</div>
  <div class="child3">Div 3</div>
</div>
</body>
</html>
```

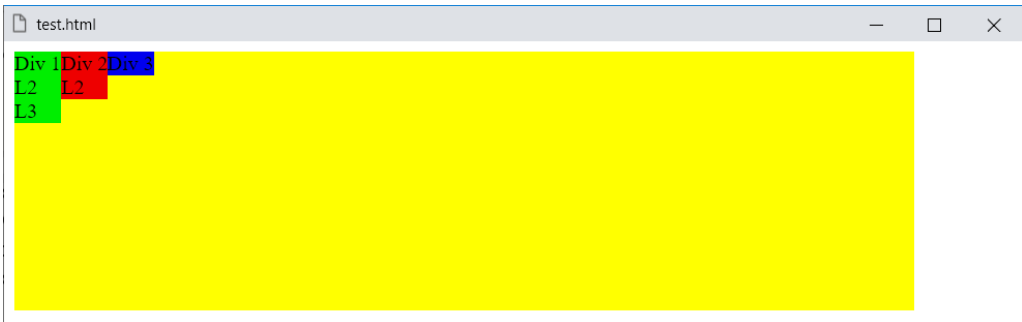
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



يمكنك جعل العناصر في الأول رأسياً ، بالكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:90%; height:200px;
    display:flex; align-items:flex-start;
  }
  div.child1{ background:#0e0; }
  div.child2{ background:#e00; }
  div.child3{ background:#00e; }
</style>
</head>
<body>
<div class="parent">
  <div class="child1">Div 1<br>L2<br>L3</div>
  <div class="child2">Div 2<br>L2</div>
  <div class="child3">Div 3</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

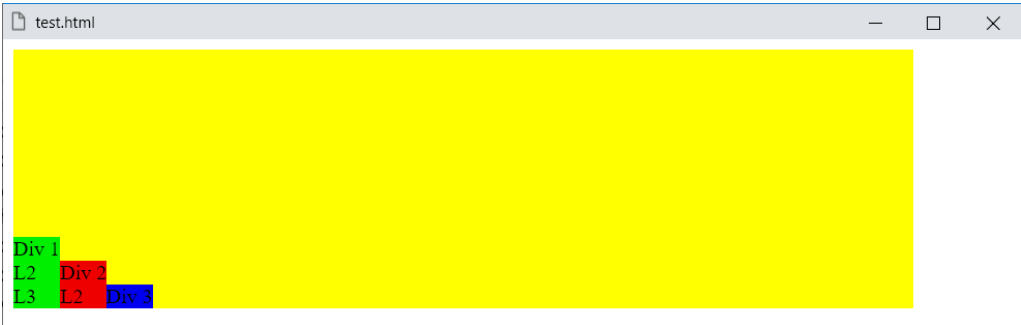


لاحظ أن القيمة flex-start للخاصية align-items جعلت العناصر في الأول رأسياً .

يملك جعل العناصر في الآخر رأسياً ، بالكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:90%; height:200px;
    display:flex; align-items:flex-end;
  }
  div.child1{ background:#0e0; }
  div.child2{ background:#e00; }
  div.child3{ background:#00e; }
</style>
</head>
<body>
<div class="parent">
  <div class="child1">Div 1<br>L2<br>L3</div>
  <div class="child2">Div 2<br>L2</div>
  <div class="child3">Div 3</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

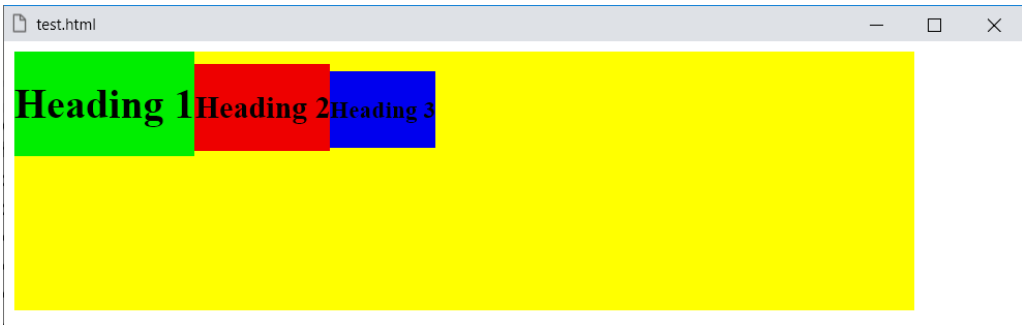


لاحظ أن القيمة flex-end للخاصية align-items جعلت العناصر في الآخر رأسياً .

يمكنك جعل العناصر على خط واحد، حتى لو اختلفت، بالكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:90%; height:200px;
    display:flex; align-items:baseline;
  }
  div.child1{ background:#0e0; }
  div.child2{ background:#e00; }
  div.child3{ background:#00e; }
</style>
</head>
<body>
<div class="parent">
  <div class="child1"><h1>Heading 1</h1></div>
  <div class="child2"><h2>Heading 2</h2></div>
  <div class="child3"><h3>Heading 3</h3></div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



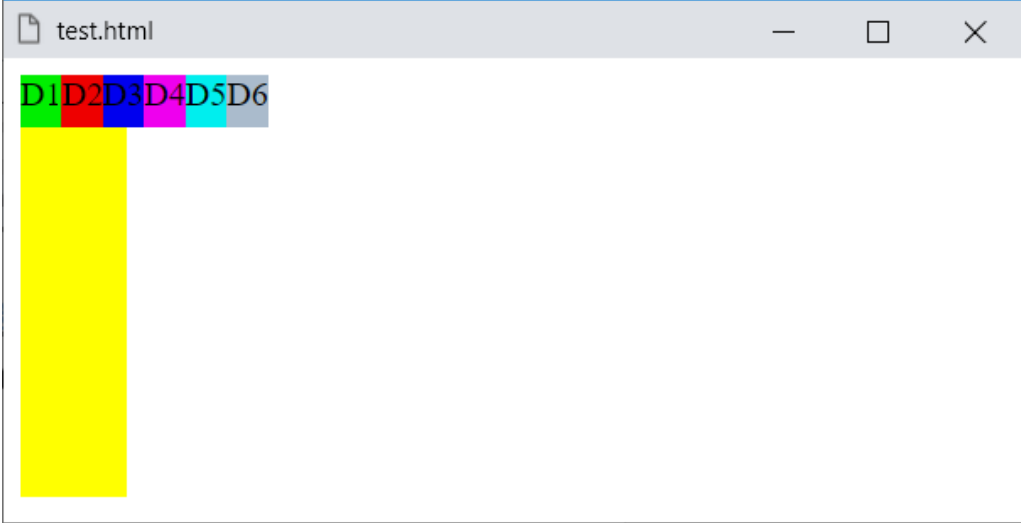
لاحظ أن القيمة baseline للخاصية align-items جعلت العناصر على سطر واحد، حتى مع اختلاف أحجامها.

التفاف المحتوى flex-wrap

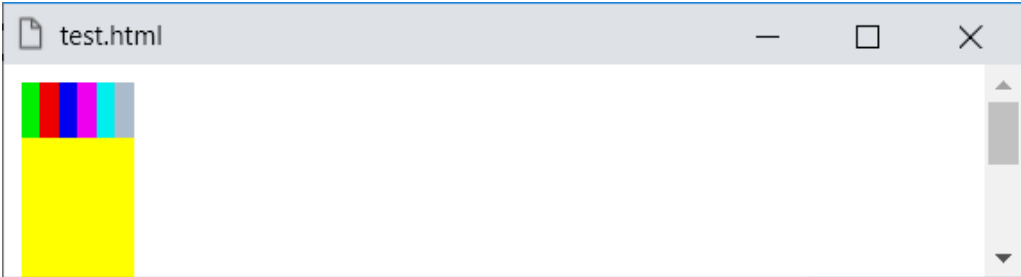
بالتأكيد كلنا يعرف التفاف النص، وهو أن النص يلتف ويكمل في السطر التالي، في حال أن عرض السطر الحالي تم ملئه بالكامل، وكذلك الأمر في المحتوى، مثلاً، هيا نجرب **كود!** بيده لنا كما يلي:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:50px; height:200px;
    display:flex;
  }
  div#d1{ background:#0e0; width:50px; height:25px; }
  div#d2{ background:#e00; width:50px; height:25px; }
  div#d3{ background:#00e; width:50px; height:25px; }
  div#d4{ background:#e0e; width:50px; height:25px; }
  div#d5{ background:#0ee; width:50px; height:25px; }
  div#d6{ background:#abc; width:50px; height:25px; }
</style>
</head>
<body>
<div class="parent">
  <div id="d1">D1</div> <div id="d2">D2</div>
  <div id="d3">D3</div> <div id="d4">D4</div>
  <div id="d5">D5</div> <div id="d6">D6</div>
</div>
</body>
</html>
```


وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن ال child خرجت عن ال parent، ولم يتم الأخذ بعرض كل من ال child، وهو 50px، حيث ان عرض ال parent أيضاً 50px، ويجوزي وسميه ونصف تقريباً من ال div ال child، ولم يلتفوا، لأن الخاصية flex-wrap تأخذ قيمة افتراضية لها، فيتم توزيع العرض على العناصر الموجودة، ما دامت هناك سعة لذلك، وهنا خرج ال child عن ال parent، لأن به نصوص أجبرته على ذلك، ولو تم حذف النصوص لاختلف الأمر، مثل الشكل التالي:

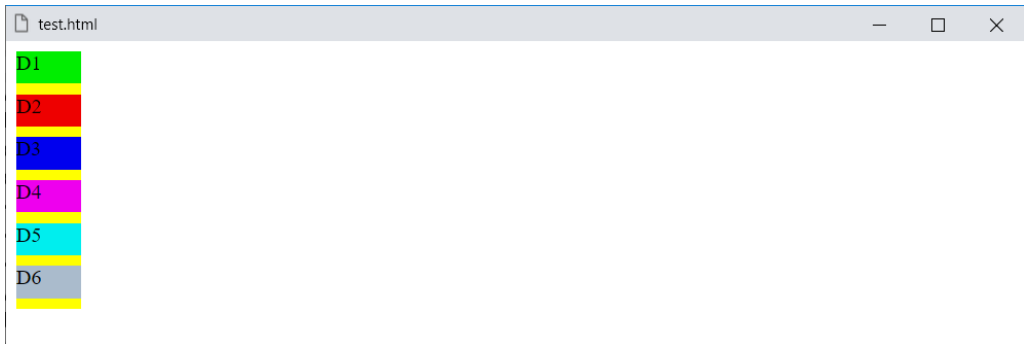


ولكن هناك قيم أخرى للتنسيق، كما سنتعلم إن شاء الله.

والآن سوف نعيد النص كما كان، لنجرب قيمة wrap، بالكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:50px; height:200px;
    display:flex; flex-wrap:wrap;
  }
  div#d1{ background:#0e0; width:50px; height:25px; }
  div#d2{ background:#e00; width:50px; height:25px; }
  div#d3{ background:#00e; width:50px; height:25px; }
  div#d4{ background:#e0e; width:50px; height:25px; }
  div#d5{ background:#0ee; width:50px; height:25px; }
  div#d6{ background:#abc; width:50px; height:25px; }
</style>
</head>
<body>
<div class="parent">
  <div id="d1">D1</div> <div id="d2">D2</div>
  <div id="d3">D3</div> <div id="d4">D4</div>
  <div id="d5">D5</div> <div id="d6">D6</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

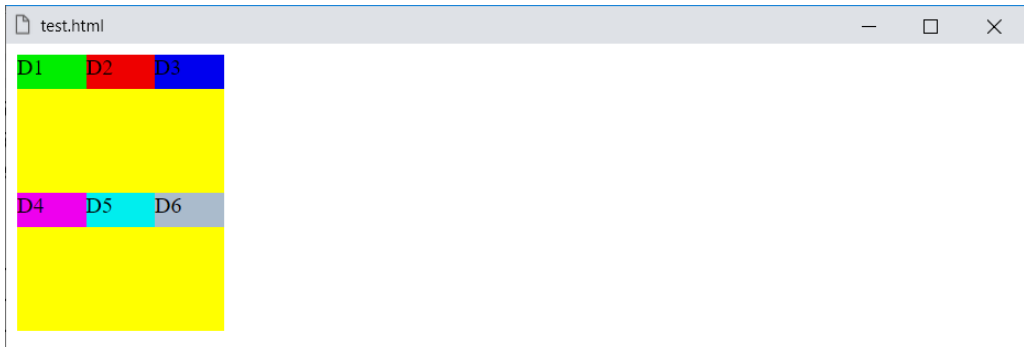


لاحظ أن كل div، أخذت سطر.

والآن نُبذ العرض بالكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:150px; height:200px;
    display:flex; flex-wrap:wrap;
  }
  div#d1{ background:#0e0; width:50px; height:25px; }
  div#d2{ background:#e00; width:50px; height:25px; }
  div#d3{ background:#00e; width:50px; height:25px; }
  div#d4{ background:#e0e; width:50px; height:25px; }
  div#d5{ background:#0ee; width:50px; height:25px; }
  div#d6{ background:#abc; width:50px; height:25px; }
</style>
</head>
<body>
<div class="parent">
  <div id="d1">D1</div> <div id="d2">D2</div>
  <div id="d3">D3</div> <div id="d4">D4</div>
  <div id="d5">D5</div> <div id="d6">D6</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

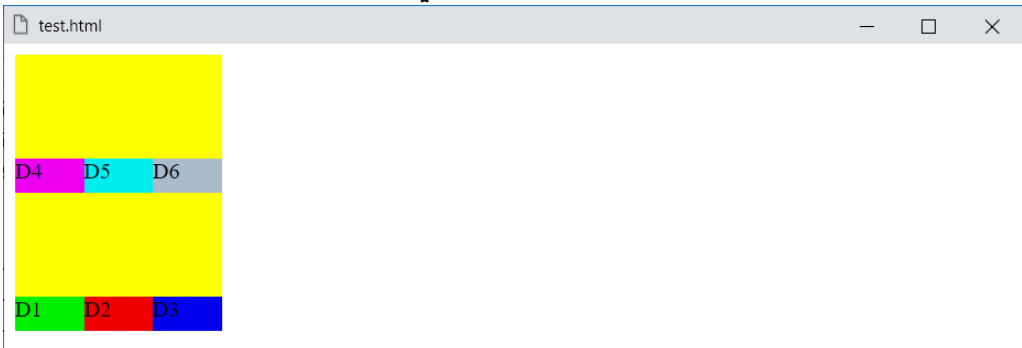


لاحظ أن كل ثلاثة div، تأخذ سطر.

والآن نجرّب التفاف عكسي بالكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#fff0; width:150px; height:200px;
    display:flex; flex-wrap:wrap-reverse;
  }
  div#d1{ background:#0e0; width:50px; height:25px; }
  div#d2{ background:#e00; width:50px; height:25px; }
  div#d3{ background:#00e; width:50px; height:25px; }
  div#d4{ background:#e0e; width:50px; height:25px; }
  div#d5{ background:#0ee; width:50px; height:25px; }
  div#d6{ background:#abc; width:50px; height:25px; }
</style>
</head>
<body>
<div class="parent">
  <div id="d1">D1</div> <div id="d2">D2</div>
  <div id="d3">D3</div> <div id="d4">D4</div>
  <div id="d5">D5</div> <div id="d6">D6</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ عمل التفاف عكسي، وكأن البداية منه الأسفل.

flex-direction اتجاه المرونة

لاحظ فيما سبق أن العناصر يتم تنظيمها في شكل صف row، والقيمة row هي القيمة الافتراضية لخاصية flex-direction. والآن ندرج التنظيم بالصف العكسي بالقيمة row-reverse، فيعكس ويبدأ من آخر عنصر الى أول عنصر.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:300px; height:200px;
    display:flex; flex-direction:row-reverse;
  }
  div#d1{ background:#0e0; width:50px; height:25px; }
  div#d2{ background:#e00; width:50px; height:25px; }
  div#d3{ background:#00e; width:50px; height:25px; }
  div#d4{ background:#e0e; width:50px; height:25px; }
  div#d5{ background:#0ee; width:50px; height:25px; }
  div#d6{ background:#abc; width:50px; height:25px; }
</style>
</head>
<body>
<div class="parent">
  <div id="d1">D1</div> <div id="d2">D2</div>
  <div id="d3">D3</div> <div id="d4">D4</div>
  <div id="d5">D5</div> <div id="d6">D6</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



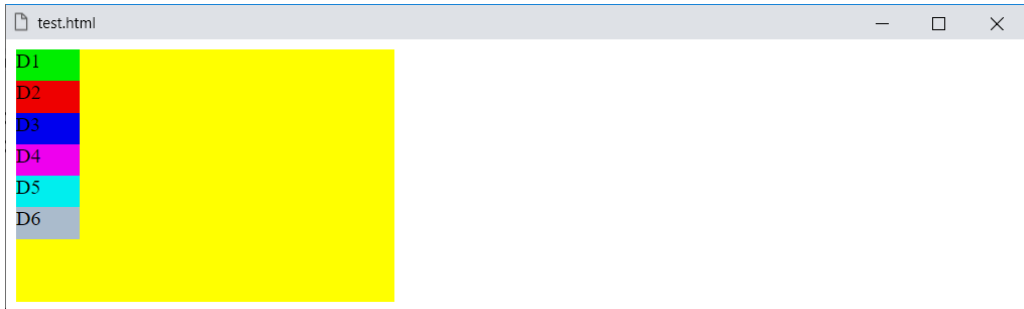
فيما سبق تم تمارينه كان الأمر الطبيعي، ومع القيمة row الافتراضية يبدأ من D1، ثم D2، ثم D3، ثم D4، ثم D5، ثم D6، ولكنه **انتبه!** فالبدء D6، ثم D5، ثم D4، ثم D3، ثم D2، ثم D1، وهذا هو تأثير القيمة row-reverse، حيث تم التنظيم كصف معكوس.

انتبه! فإننا هنا لا نتحدث عن dir="rtl" في html، ولا نتحدث عن direction:rtl في css، ولكنه الأمر مختلف، فإذا حددت rtl للصفحة، سيكون طبيعة الصفحة من اليمين، فيعكس هو إلى اليسار، فرجاء **انتبه!** حتى لا تختلط عليك الأمور.

والآن تنظيم عمودي بالقيمة column، بالكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:300px; height:200px;
    display:flex; flex-direction:column;
  }
  div#d1{ background:#0e0; width:50px; height:25px; }
  div#d2{ background:#e00; width:50px; height:25px; }
  div#d3{ background:#00e; width:50px; height:25px; }
  div#d4{ background:#e0e; width:50px; height:25px; }
  div#d5{ background:#0ee; width:50px; height:25px; }
  div#d6{ background:#abc; width:50px; height:25px; }
</style>
</head>
<body>
<div class="parent">
  <div id="d1">D1</div> <div id="d2">D2</div>
  <div id="d3">D3</div> <div id="d4">D4</div>
  <div id="d5">D5</div> <div id="d6">D6</div>
</div>
</body>
</html>
```

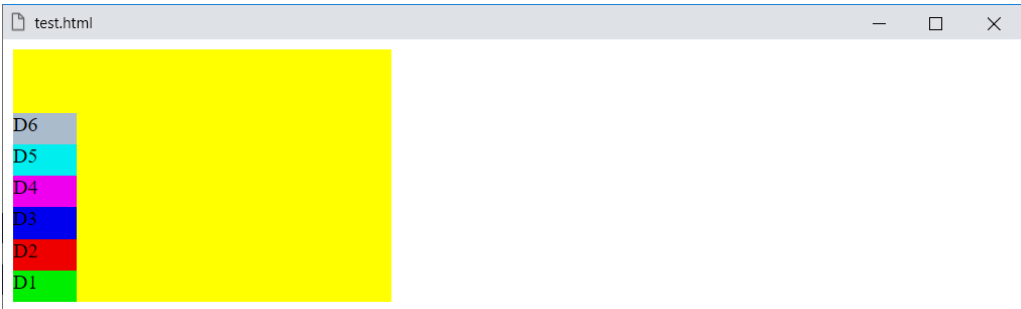
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



والعمودي العكسي من أسفل بـ column-reverse، بالكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:300px; height:200px;
    display:flex; flex-direction:column-reverse;
  }
  div#d1{ background:#0e0; width:50px; height:25px; }
  div#d2{ background:#e00; width:50px; height:25px; }
  div#d3{ background:#00e; width:50px; height:25px; }
  div#d4{ background:#e0e; width:50px; height:25px; }
  div#d5{ background:#0ee; width:50px; height:25px; }
  div#d6{ background:#abc; width:50px; height:25px; }
</style>
</head>
<body>
<div class="parent">
  <div id="d1">D1</div> <div id="d2">D2</div>
  <div id="d3">D3</div> <div id="d4">D4</div>
  <div id="d5">D5</div> <div id="d6">D6</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

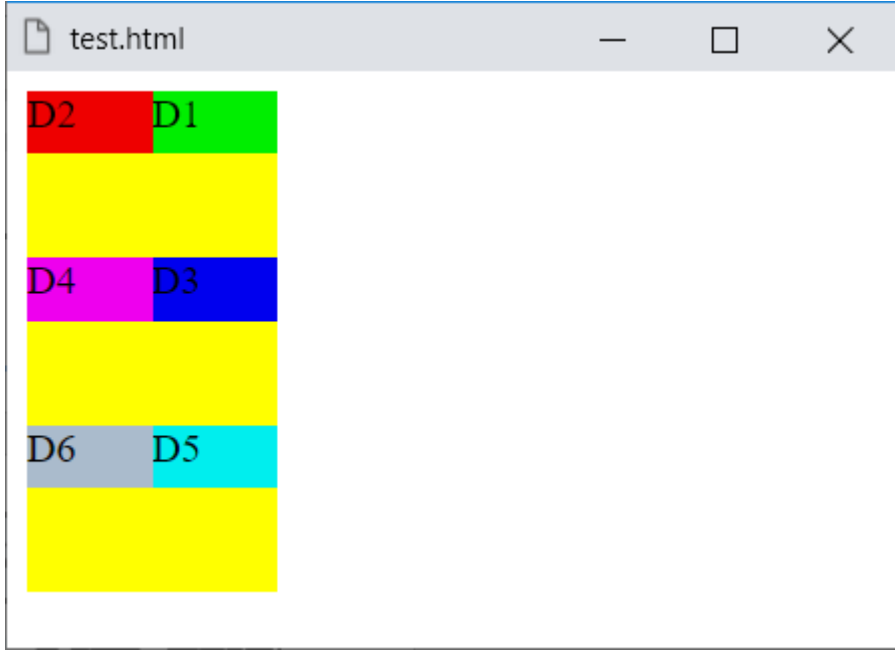


تدفق المرونة flex-flow

هي خاصية تجمعة بين خاصيتيه للتسعيد، والخاصيتيه التي تجمعه بينهم هم flex-direction، و flex-wrap.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:100px; height:200px;
    display:flex; flex-flow:row-reverse wrap;
  }
  div#d1{ background:#0e0; width:50px; height:25px; }
  div#d2{ background:#e00; width:50px; height:25px; }
  div#d3{ background:#00e; width:50px; height:25px; }
  div#d4{ background:#e0e; width:50px; height:25px; }
  div#d5{ background:#0ee; width:50px; height:25px; }
  div#d6{ background:#abc; width:50px; height:25px; }
</style>
</head>
<body>
<div class="parent">
  <div id="d1">D1</div> <div id="d2">D2</div>
  <div id="d3">D3</div> <div id="d4">D4</div>
  <div id="d5">D5</div> <div id="d6">D6</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



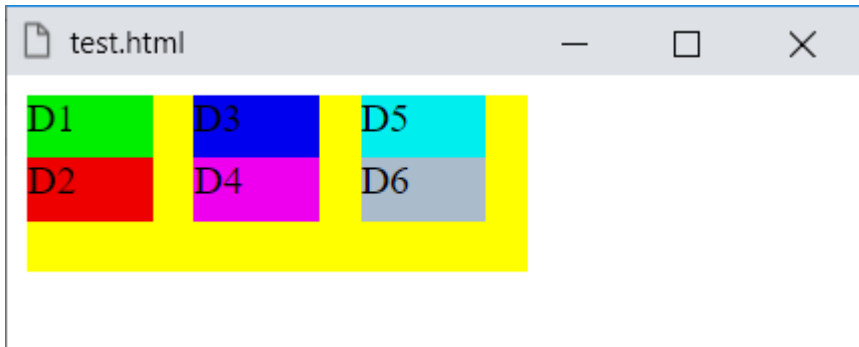
وكما تلاحظ أنه مع طريق الخاصية flex-flow، يمكنه ضبط اتجاه المرونة flex-direction، والالتفاف flex-wrap، في مكان واحد للتسهيل والاختصار.

وإذا تم تجربة القيمة column مع القيمة wrap، سيكون الوضع مختلف، ولكه لتدري التأثير لابد من تقليل الطول لتكرار الأعمدة، وزيادة العرض لتوسيع المسافة، كما سنجرب إن شاء الله في المثال التالي.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
div.parent{
background:#fff0; width:200px; height:70px;
display:flex; flex-flow:column wrap;
}
div#d1{ background:#0e0; width:50px; height:25px; }
div#d2{ background:#e00; width:50px; height:25px; }
div#d3{ background:#00e; width:50px; height:25px; }
div#d4{ background:#e0e; width:50px; height:25px; }
div#d5{ background:#0ee; width:50px; height:25px; }
div#d6{ background:#abc; width:50px; height:25px; }
</style>
</head>
<body>
<div class="parent">
<div id="d1">D1</div> <div id="d2">D2</div>
<div id="d3">D3</div> <div id="d4">D4</div>
<div id="d5">D5</div> <div id="d6">D6</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



محاذاة المحتوى align-content

المحاذاة هنا تعطي مدونة من ناحية أخرى، ولدي التأثير قم بتقليل العرض، ونهبط خاصية flex-wrap على القيمة wrap، ليحدث التفاف، وتدي التأثيرات.

كود! قم بكتابة الكود التالي ثم احفظه:

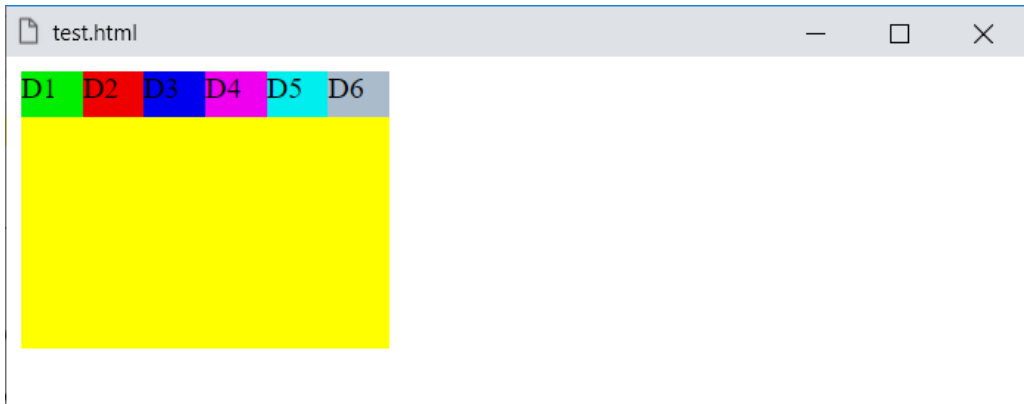
```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:200px; height:150px;
    display:flex; flex-wrap:wrap;
    align-content:center;
  }
  div#d1{ background:#0e0; width:50px; height:25px; }
  div#d2{ background:#e00; width:50px; height:25px; }
  div#d3{ background:#00e; width:50px; height:25px; }
  div#d4{ background:#e0e; width:50px; height:25px; }
  div#d5{ background:#0ee; width:50px; height:25px; }
  div#d6{ background:#abc; width:50px; height:25px; }
</style>
</head>
<body>
<div class="parent">
  <div id="d1">D1</div> <div id="d2">D2</div>
  <div id="d3">D3</div> <div id="d4">D4</div>
  <div id="d5">D5</div> <div id="d6">D6</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه نتج عن العرض 200px، صفان، الأول به أربع عناصر، والثاني به عنصران، فأصبحوا قريبين مع انهم في المنتصف رأسياً، وذلك حدث بإعطاء القيمة center للخاصية align-content.

انتبه! أنه لو تم الغاء ال wrap، مع بقاء القيمة center كما هي، سيكون مثل الشكل التالي:



يمكننا البداية مع الآخر بالقيمة flex-end للخاصية align-content، كما يلي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:200px; height:150px;
    display:flex; flex-wrap:wrap;
    align-content:flex-end;
  }
  div#d1{ background:#0e0; width:50px; height:25px; }
  div#d2{ background:#e00; width:50px; height:25px; }
  div#d3{ background:#00e; width:50px; height:25px; }
  div#d4{ background:#e0e; width:50px; height:25px; }
  div#d5{ background:#0ee; width:50px; height:25px; }
  div#d6{ background:#abc; width:50px; height:25px; }
</style>
</head>
<body>
<div class="parent">
  <div id="d1">D1</div> <div id="d2">D2</div>
  <div id="d3">D3</div> <div id="d4">D4</div>
  <div id="d5">D5</div> <div id="d6">D6</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن البداية هنا من الأسفل، وكأننا أخذنا العناصر من الأعلى كما هي، ثم وضعناهم في الأسفل.

يمكننا البداية مع الأول بالقيمة flex-start الخاصة بـ align-content، كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:200px; height:150px;
    display:flex; flex-wrap:wrap;
    align-content:flex-start;
  }
  div#d1{ background:#0e0; width:50px; height:25px; }
  div#d2{ background:#e00; width:50px; height:25px; }
  div#d3{ background:#00e; width:50px; height:25px; }
  div#d4{ background:#e0e; width:50px; height:25px; }
  div#d5{ background:#0ee; width:50px; height:25px; }
  div#d6{ background:#abc; width:50px; height:25px; }
</style>
</head>
<body>
<div class="parent">
  <div id="d1">D1</div> <div id="d2">D2</div>
  <div id="d3">D3</div> <div id="d4">D4</div>
  <div id="d5">D5</div> <div id="d6">D6</div>
</div>
</body>
</html>
```


وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن البداية هنا من الأول.

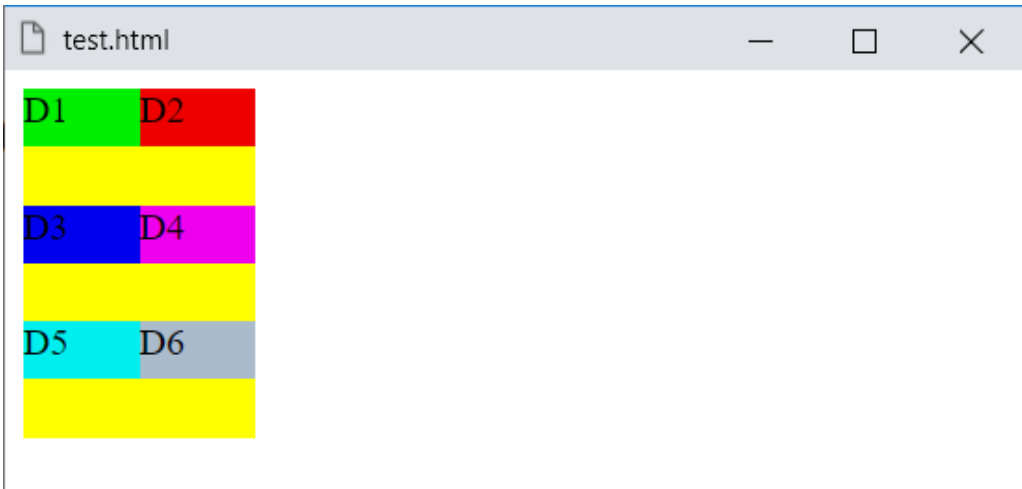
يملك توزيع العناصر بمسافات حسب الطول، بالقيمة stretch، وهي القيمة الافتراضية للخاصية align-content.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:200px; height:150px;
    display:flex; flex-wrap:wrap;
    align-content:stretch;
  }
  div#d1{ background:#0e0; width:50px; height:25px; }
  div#d2{ background:#e00; width:50px; height:25px; }
  div#d3{ background:#00e; width:50px; height:25px; }
  div#d4{ background:#e0e; width:50px; height:25px; }
  div#d5{ background:#0ee; width:50px; height:25px; }
  div#d6{ background:#abc; width:50px; height:25px; }
</style>
</head>
<body>
<div class="parent">
  <div id="d1">D1</div> <div id="d2">D2</div>
  <div id="d3">D3</div> <div id="d4">D4</div>
  <div id="d5">D5</div> <div id="d6">D6</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



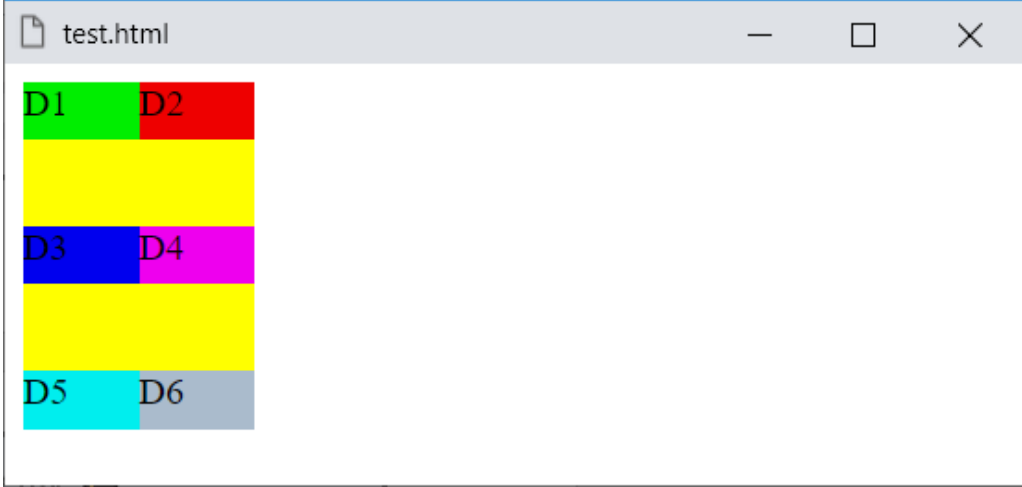
لاحظ أنه تم توزيع العناصر على الطول، وإذا تم تقليل العرض للحاوية
 ال parent الى 100px، سترى مثل الشكل التالي:



يملك عمل مسافات متساوية بين العناصر عن طريق ضبط الخاصية align-content على القيمة space-between، وسوف نجرّب بالكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:100px; height:150px;
    display:flex; flex-wrap:wrap;
    align-content:space-between;
  }
  div#d1{ background:#0e0; width:50px; height:25px; }
  div#d2{ background:#e00; width:50px; height:25px; }
  div#d3{ background:#00e; width:50px; height:25px; }
  div#d4{ background:#e0e; width:50px; height:25px; }
  div#d5{ background:#0ee; width:50px; height:25px; }
  div#d6{ background:#abc; width:50px; height:25px; }
</style>
</head>
<body>
<div class="parent">
  <div id="d1">D1</div> <div id="d2">D2</div>
  <div id="d3">D3</div> <div id="d4">D4</div>
  <div id="d5">D5</div> <div id="d6">D6</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم توزيع مسافات منسقة ومنظمة ومتساوية بين كل صف والصف الآخر.

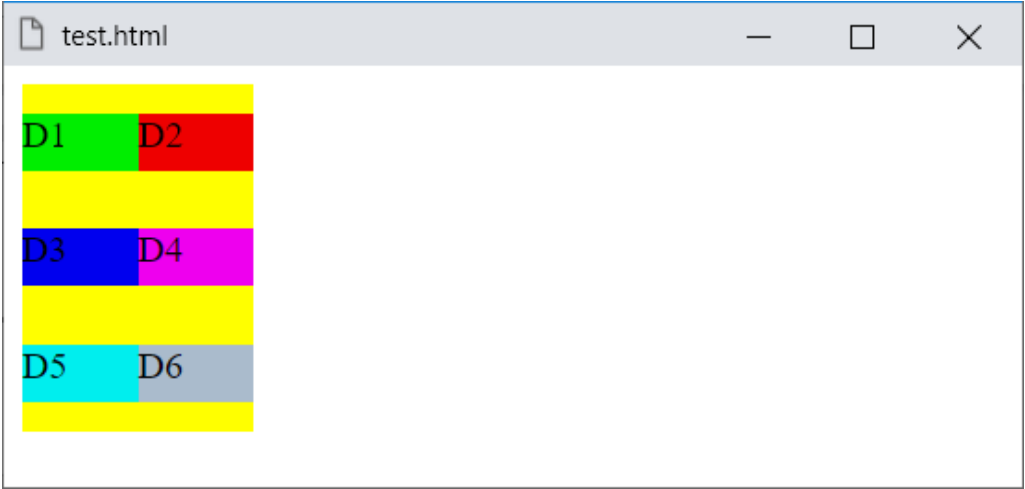
لاحظ أنه لا يوجد مسافة قبل أول صف.

لاحظ أنه لا يوجد مسافة بعد آخر صف.

يملك توزيع مسافات متساوي مع حول العناصر، بحيث يكون هناك مسافة قبل العنصر، ونفس المسافة بعده، ثم مسافة قبل العنصر الذي يليه، ثم مسافة بعده، وهكذا، ومعني ذلك أنك ستدري، مسافة، ثم عنصر، ثم مسافتيه، ثم عنصر، ثم مسافتيه، ثم عنصر، وهكذا حتى آخر عنصر، ثم مسافة واحدة بعده، وسنجد **بالكود!** التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
div.parent{
background:#ff0; width:100px; height:150px;
display:flex; flex-wrap:wrap;
align-content:space-around;
}
div#d1{ background:#0e0; width:50px; height:25px; }
div#d2{ background:#e00; width:50px; height:25px; }
div#d3{ background:#00e; width:50px; height:25px; }
div#d4{ background:#e0e; width:50px; height:25px; }
div#d5{ background:#0ee; width:50px; height:25px; }
div#d6{ background:#abc; width:50px; height:25px; }
</style>
</head>
<body>
<div class="parent">
<div id="d1">D1</div> <div id="d2">D2</div>
<div id="d3">D3</div> <div id="d4">D4</div>
<div id="d5">D5</div> <div id="d6">D6</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



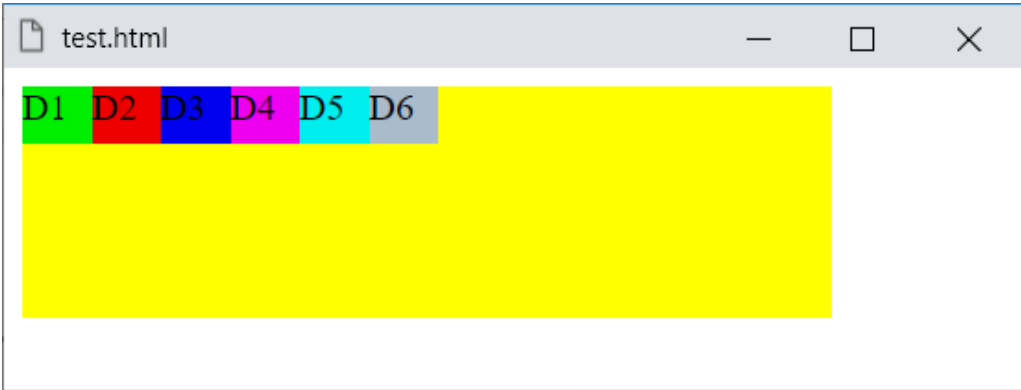
لاحظ أن هناك مسافة قبل أول عنصر، ومسافة بعد آخر عنصر، وباقي العناصر بينهم مسافتهم.

تنظيم المحتوى justify-content

خاصية justify-content هامة جداً لتنظيم المحتوى وهي تفعل نفس تنسيقات الخاصية align-content ولكنه أفقياً، والقيمة الافتراضية لها هي flex-start، وسنجردها **باللورد!** التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:350px; height:100px;
    display:flex;
    justify-content:flex-start;
  }
  div#d1{ background:#0e0; width:30px; height:25px; }
  div#d2{ background:#e00; width:30px; height:25px; }
  div#d3{ background:#00e; width:30px; height:25px; }
  div#d4{ background:#e0e; width:30px; height:25px; }
  div#d5{ background:#0ee; width:30px; height:25px; }
  div#d6{ background:#abc; width:30px; height:25px; }
</style>
</head>
<body>
<div class="parent">
  <div id="d1">D1</div> <div id="d2">D2</div>
  <div id="d3">D3</div> <div id="d4">D4</div>
  <div id="d5">D5</div> <div id="d6">D6</div>
</div>
</body>
</html>
```


وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

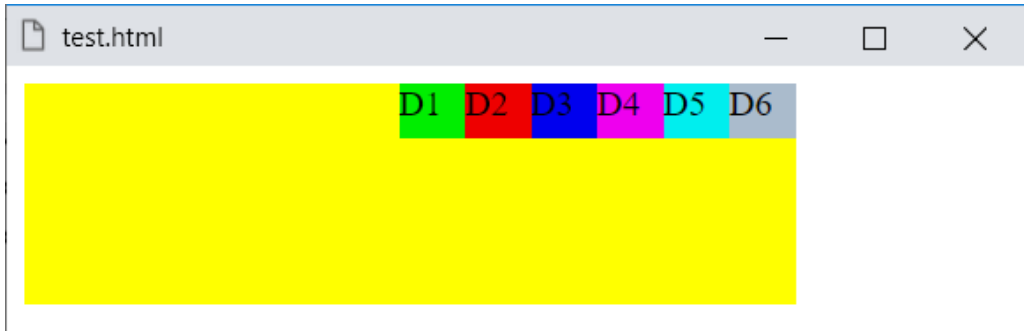


انتبه! لأن القيمة `flex-start` هنا المعني بها أفقياً وليس رأسياً، أي أنه بدأ من اليسار لأن الصفحة هنا تبدأ من اليسار، ولو اردنا ان يكون في النهاية فيعني هذا اليمين.

سنجرب القيمة flex-end، بالكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:350px; height:100px;
    display:flex; justify-content:flex-end;
  }
  div#d1{ background:#0e0; width:30px; height:25px; }
  div#d2{ background:#e00; width:30px; height:25px; }
  div#d3{ background:#00e; width:30px; height:25px; }
  div#d4{ background:#e0e; width:30px; height:25px; }
  div#d5{ background:#0ee; width:30px; height:25px; }
  div#d6{ background:#abc; width:30px; height:25px; }
</style>
</head>
<body>
<div class="parent">
  <div id="d1">D1</div> <div id="d2">D2</div>
  <div id="d3">D3</div> <div id="d4">D4</div>
  <div id="d5">D5</div> <div id="d6">D6</div>
</div>
</body>
</html>
```

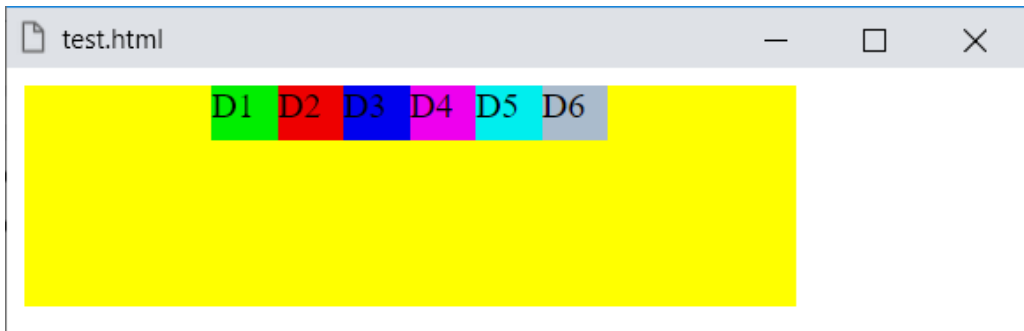
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



سنجرب القيمة center ، بالكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:350px; height:100px;
    display:flex; justify-content:center;
  }
  div#d1{ background:#0e0; width:30px; height:25px; }
  div#d2{ background:#e00; width:30px; height:25px; }
  div#d3{ background:#00e; width:30px; height:25px; }
  div#d4{ background:#e0e; width:30px; height:25px; }
  div#d5{ background:#0ee; width:30px; height:25px; }
  div#d6{ background:#abc; width:30px; height:25px; }
</style>
</head>
<body>
<div class="parent">
  <div id="d1">D1</div> <div id="d2">D2</div>
  <div id="d3">D3</div> <div id="d4">D4</div>
  <div id="d5">D5</div> <div id="d6">D6</div>
</div>
</body>
</html>
```

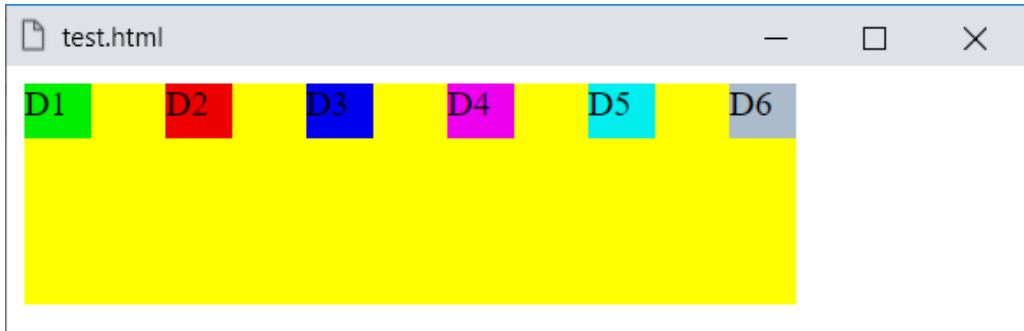
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



سنجرب القيمة space-between، بالكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:350px; height:100px;
    display:flex; justify-content:space-between;
  }
  div#d1{ background:#0e0; width:30px; height:25px; }
  div#d2{ background:#e00; width:30px; height:25px; }
  div#d3{ background:#00e; width:30px; height:25px; }
  div#d4{ background:#e0e; width:30px; height:25px; }
  div#d5{ background:#0ee; width:30px; height:25px; }
  div#d6{ background:#abc; width:30px; height:25px; }
</style>
</head>
<body>
<div class="parent">
  <div id="d1">D1</div> <div id="d2">D2</div>
  <div id="d3">D3</div> <div id="d4">D4</div>
  <div id="d5">D5</div> <div id="d6">D6</div>
</div>
</body>
</html>
```

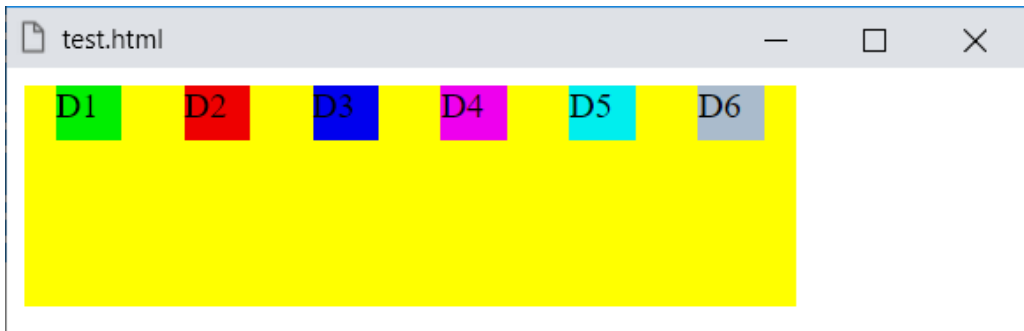
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



سنجرب القيمة space-around، بالكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:350px; height:100px;
    display:flex; justify-content:space-around;
  }
  div#d1{ background:#0e0; width:30px; height:25px; }
  div#d2{ background:#e00; width:30px; height:25px; }
  div#d3{ background:#00e; width:30px; height:25px; }
  div#d4{ background:#e0e; width:30px; height:25px; }
  div#d5{ background:#0ee; width:30px; height:25px; }
  div#d6{ background:#abc; width:30px; height:25px; }
</style>
</head>
<body>
<div class="parent">
  <div id="d1">D1</div> <div id="d2">D2</div>
  <div id="d3">D3</div> <div id="d4">D4</div>
  <div id="d5">D5</div> <div id="d6">D6</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



محاذاة العنصر نفسه align-self

هذه الخاصية تمنحك مرونة قوية للعناصر ال child، لك عنصر على حدي، والقيمة الافتراضية لها هي auto، أي تلقائي، وتأخذ القيم flex-start، flex-end، stretch، و center، و baseline، وهذه المفترض أنك فهمت كل تلك القيم، مما سبق، والموضوع مع الخاصية align-self، مرونة للعنصر أكثر مما تتخيل، "فمثلاً إذا كنت تعمل على الاتجاه row، فإن عملك يحسب رأسياً"، "وإذا كنت تعمل على الاتجاه column، فإن عملك يحسب أفقياً"، وبالطبع المقصود بالاتجاه هنا هو ال parent.

إذا تم ضبط خاصية flex-direction على القيمة row لل parent، ثم تم ضبط خاصية align-self على القيمة center لل child، سيكون في المنتصف رأسياً.

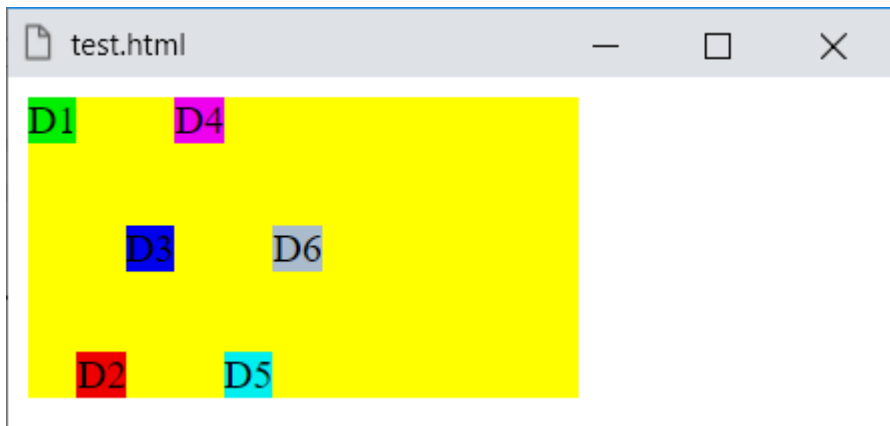
إذا تم ضبط خاصية flex-direction على القيمة column لل parent، ثم تم ضبط خاصية align-self على القيمة center لل child، سيكون في المنتصف أفقياً.

وهكذا، للبداية، والنهائية، وباقي القيم التي تم ذكرها أنفاً، وفيما يلي مثاليه للتوضيح في الحالتين، ولاحظ أنت بنفسك، بشرط أن تكون قمت بمذاكرة ما سبق جيداً.

المثال الأول بقيمة row، كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
div.parent{
background:#fff; width:220px; height:120px;
display:flex; align-items:flex-start; flex-direction:row;
}
div#d1{ background:#0e0; align-self:flex-start; }
div#d2{ background:#e00; align-self:flex-end; }
div#d3{ background:#00e; align-self:center; }
div#d4{ background:#e0e; align-self:flex-start; }
div#d5{ background:#0ee; align-self:flex-end; }
div#d6{ background:#abc; align-self:center; }
</style>
</head>
<body>
<div class="parent">
<div id="d1">D1</div> <div id="d2">D2</div>
<div id="d3">D3</div> <div id="d4">D4</div>
<div id="d5">D5</div> <div id="d6">D6</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



المثال الثاني بقيمة column، كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
div.parent{
background:#fff; width:220px; height:120px;
display:flex; align-items:flex-start; flex-direction:column;
}
div#d1{ background:#0e0; align-self:flex-start; }
div#d2{ background:#e00; align-self:flex-end; }
div#d3{ background:#00e; align-self:center; }
div#d4{ background:#e0e; align-self:flex-start; }
div#d5{ background:#0ee; align-self:flex-end; }
div#d6{ background:#abc; align-self:center; }
</style>
</head>
<body>
<div class="parent">
<div id="d1">D1</div> <div id="d2">D2</div>
<div id="d3">D3</div> <div id="d4">D4</div>
<div id="d5">D5</div> <div id="d6">D6</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

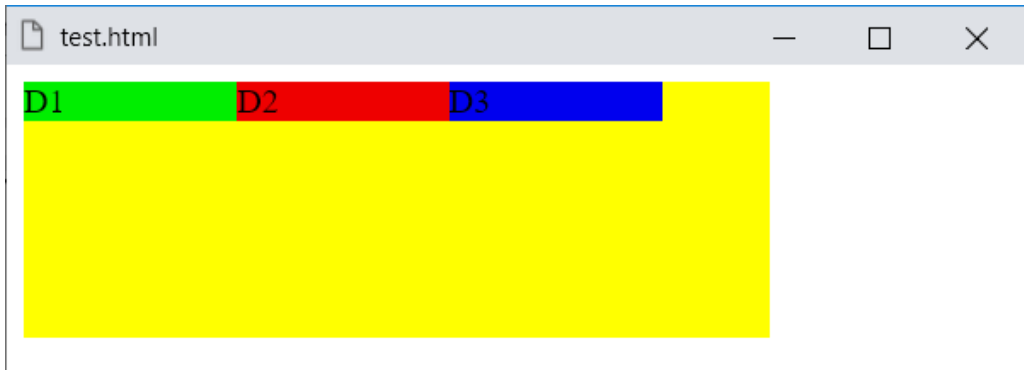


أساس للمرونة flex-basis

مع الخاصية flex-basis، يمكن وضع أساس لعنصر child، بمقدار معين، وللتوضيح اكتب الكود! التالي، ثم أحفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
div.parent{
background:#ff0; width:350px; height:120px;
display:flex; align-items:flex-start;
}
div#d1{ background:#0e0; width:100px; }
div#d2{ background:#e00; width:100px; }
div#d3{ background:#00e; width:100px; }
</style>
</head>
<body>
<div class="parent">
<div id="d1">D1</div><div id="d2">D2</div>
<div id="d3">D3</div>
</div>
</body>
</html>
```

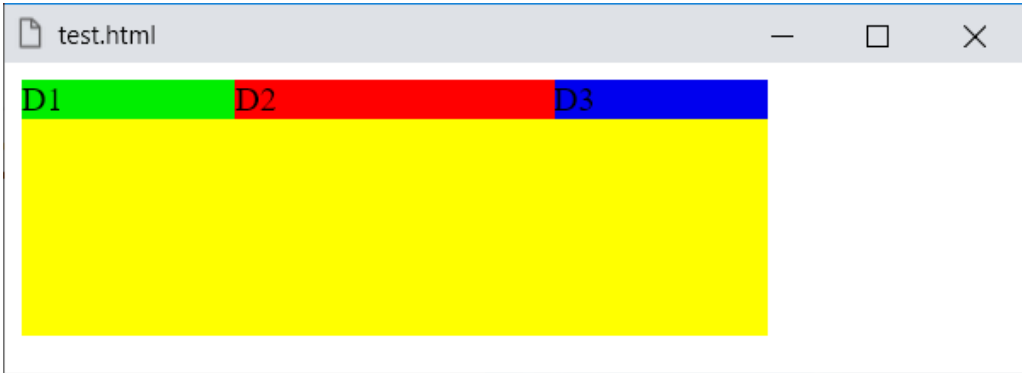
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه في التمرين السابق، تم ضبط عرض ال parent على 350px، وتم إعطاء عرض لكل child بمقدار 100px، وهم ثلاثة وسوم div، أي $300 = 100 + 100 + 100$ ، ويتبقى 50px في ال parent، كما نلاحظ في الشكل السابق، ويمكن أن نميز واحدة منهم بزيادة معينة عن طريق الخاصية flex-basis، وسوف نميز العنصر الثاني d2، بحيث يأخذ عرض بمقدار 150px، فيحوز على ال 50px المتبقية، ويكونوا منه نصيبه، بالكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:350px; height:120px;
    display:flex; align-items:flex-start;
  }
  div#d1{background:#0e0; width:100px; }
  div#d2{background:red; width:100px; flex-basis:150px; }
  div#d3{background:#00e; width:100px; }
</style>
</head>
<body>
<div class="parent">
  <div id="d1">D1</div><div id="d2">D2</div>
  <div id="d3">D3</div>
</div>
</body>
</html>
```

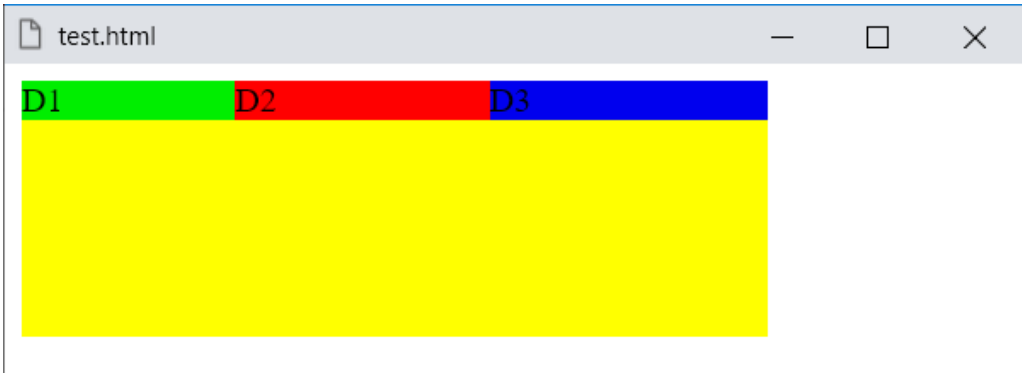
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



ولو غيرنا كود العنصر d2 والعنصر d3 فقط، ليكُون كما يلي:

```
div#d2{background:red; width:100px; flex-basis:120px; }
div#d3{background:#00e; width:100px; flex-basis:130px; }
```

سيكُون الناتج مثل الشكل التالي:



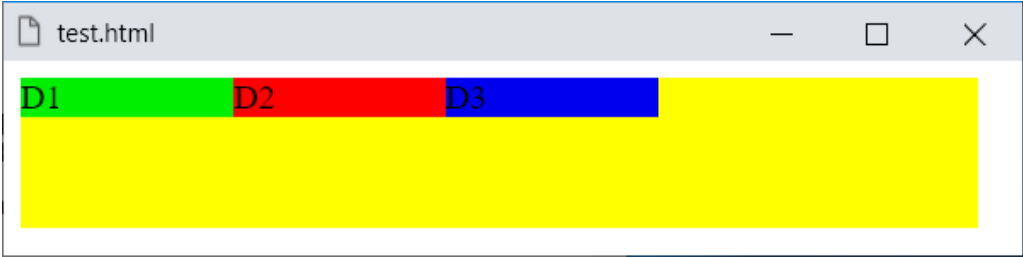
وهكذا، تساعد خاصية flex-basis، في تخصيص أكتد، ومرونة أكتد بفضل الله.

ازدياد العناصر flex-grow

نمو بعض العناصر، مع بقاء العناصر الأخرى، يتم التحكم فيه عن طريق الخاصية flex-grow، والقيمة الافتراضية لها هي صفر، أي لا نمو لها، وتأخذ قيمة رقمية، مثلاً 1 أو 2 أو 3، وهكذا، والعنصر صاحب القيمة الأعلى يكون له الحق الأكبر في النمو والازدياد، والآه سوف نجرب إن شاء الله، بعض الأمثلة في هذا الموضوع، وبعد التدريب عليها بفضل الله سوف تفهم خاصية الازدياد flex-grow جيداً.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:450px; height:70px;
    display:flex; align-items:flex-start;
  }
  div#d1{background:#0e0; width:100px; flex-grow:0; }
  div#d2{background:red; width:100px; flex-grow:0; }
  div#d3{background:#00e; width:100px; flex-grow:0; }
</style>
</head>
<body>
<div class="parent">
  <div id="d1">D1</div><div id="d2">D2</div>
  <div id="d3">D3</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ في الكود السابق أعطينا كل العناصر الـ child، القيمة صفر للخاصية flex-grow، والصفر هي القيمة الافتراضية، كما ذكرنا، ولذلك يتضح أن كل شيء كما هو، بدون أي اختلاف.

لاحظ أن الـ div الذي يحمل id بالقيمة d1، تم ضبط العرض width له على القيمة 100px، وثبت عليها ولم يتغير في الصفحة.

لاحظ أن الـ div الذي يحمل id بالقيمة d2، تم ضبط العرض width له على القيمة 100px، وثبت عليها ولم يتغير في الصفحة.

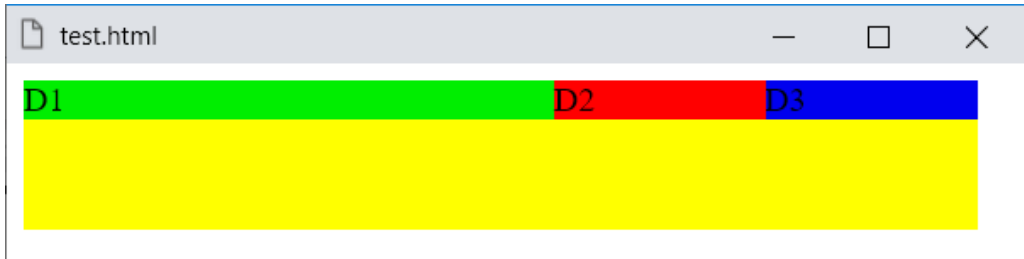
لاحظ أن الـ div الذي يحمل id بالقيمة d3، تم ضبط العرض width له على القيمة 100px، وثبت عليها ولم يتغير في الصفحة.

والآن، **انتبه!** لأننا سوف نبدأ، في التجارب على الخاصية flex-grow، بالزيادة، كما يلي إن شاء الله.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:450px; height:70px;
    display:flex; align-items:flex-start;
  }
  div#d1{background:#0e0; width:100px; flex-grow:1; }
  div#d2{background:red; width:100px; flex-grow:0; }
  div#d3{background:#00e; width:100px; flex-grow:0; }
</style>
</head>
<body>
<div class="parent">
  <div id="d1">D1</div><div id="d2">D2</div>
  <div id="d3">D3</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

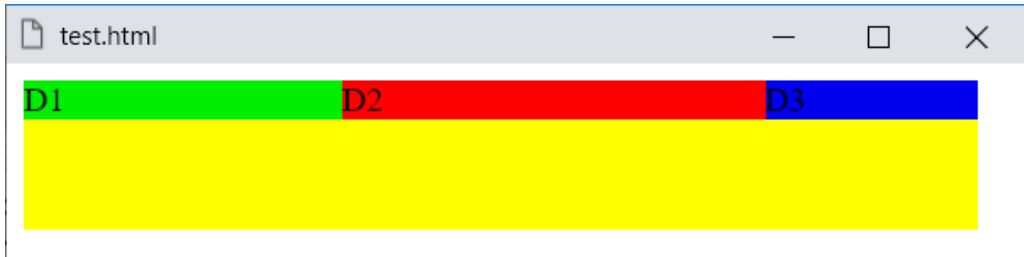


لاحظ أن العنصر d1 أصبح أكثر نمواً من غيره، لأنه تم إعطاء القيمة 1 لخاصية flex-grow له.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
div.parent{
background:#ff0; width:450px; height:70px;
display:flex; align-items:flex-start;
}
div#d1{background:#0e0; width:100px; flex-grow:1; }
div#d2{background:red; width:100px; flex-grow:2; }
div#d3{background:#00e; width:100px; flex-grow:0; }
</style>
</head>
<body>
<div class="parent">
<div id="d1">D1</div><div id="d2">D2</div>
<div id="d3">D3</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

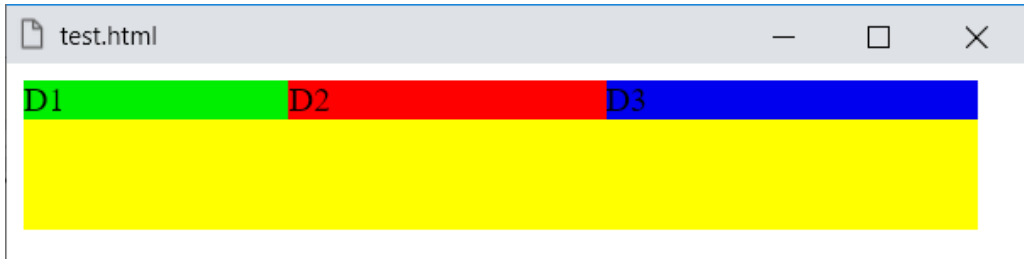


لاحظ أن العنصر d2 أخذ نمو بقيمة 2، فهو أكثر نمو، ثم العنصر d2 لأن قيمة نموه 1، فنموه أقل من نمو d2.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:450px; height:70px;
    display:flex; align-items:flex-start;
  }
  div#d1{background:#0e0; width:100px; flex-grow:1; }
  div#d2{background:red; width:100px; flex-grow:2; }
  div#d3{background:#00e; width:100px; flex-grow:3; }
</style>
</head>
<body>
<div class="parent">
  <div id="d1">D1</div><div id="d2">D2</div>
  <div id="d3">D3</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



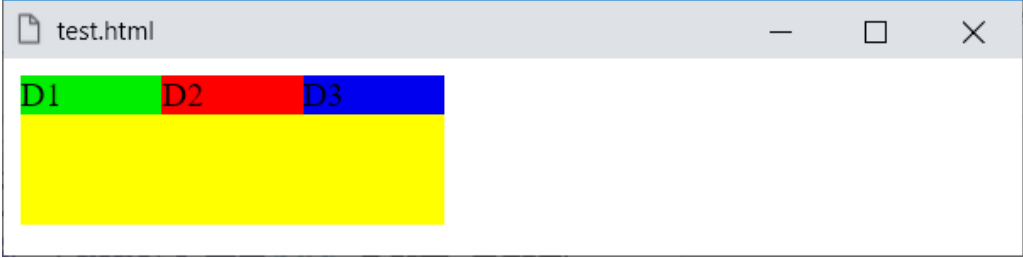
لاحظ أن العنصر d3 أخذ نمو بقيمة 3، فهو أكثر نمو، ثم العنصر d2 لأن قيمة نموه 2، ثم العنصر d1 لأن قيمة نموه 1، فنموه أقل من نمو d2 و d3.

انكماش العناصر flex-shrink

الخاصية flex-shrink، عكس الخاصية flex-grow، لأنها تقلص العناصر وتجعلها تنكّمت، والقيمة الافتراضية لها هي 1، أي أنها تنكّمت افتراضياً، لماذا؟، لأنك لو حددت عرض الـ parent مثلاً بمقدار 200px، ثم وضعت بداخله ثلاثة child، وكل واحد منهم أخذ عرض مثلاً بمقدار 100px، أي أن اجماليهم الثلاثة هو 300px، واجمالي الحاوية هو 200px، أي أن هناك عجز 100px كاملة، ولكي تلاحظ أنها تنكّمت تلقائياً لتناسب العرض المتاح، لأنك تعمل بالـ flex، أي المرونة، والكود! التالي يوضح ذلك:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:200px; height:70px;
    display:flex; align-items:flex-start;
  }
  div#d1{background:#0e0; width:100px; }
  div#d2{background:red; width:100px; }
  div#d3{background:#00e; width:100px; }
</style>
</head>
<body>
<div class="parent">
  <div id="d1">D1</div><div id="d2">D2</div>
  <div id="d3">D3</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

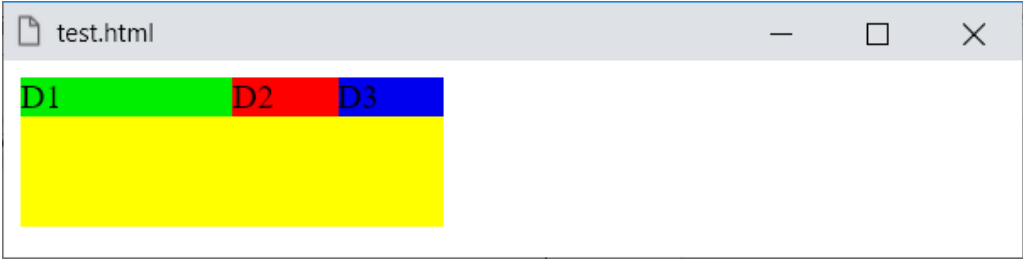


لاحظ أن العناصر child، خضعت لتأقلم مع عرض ال parent، وذلك يدل على أن القيمة الافتراضية هي 1، أي قم بالإنكماش.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:200px; height:70px;
    display:flex; align-items:flex-start;
  }
  div#d1{background:#0e0; width:100px; flex-shrink:0; }
  div#d2{background:red; width:100px; }
  div#d3{background:#00e; width:100px; }
</style>
</head>
<body>
<div class="parent">
  <div id="d1">D1</div><div id="d2">D2</div>
  <div id="d3">D3</div>
</div>
</body>
</html>
```

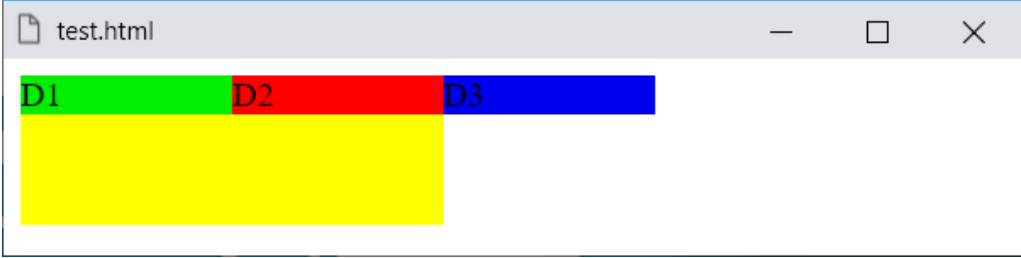
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



وهنا تم ضبط الخاصية flex-shrink للعنصر d1 على القيمة 0، ومعناها وكأنني أقول له لا تنكس، أي لا تصغر أبداً، وهنا أخذ العرض على حساب العنصرين الآخرين d2 و d3، فتم تصغيرهم أكثر، وسنجرّب تجربة أخرى بعد أن نكتب الكود! التالي ثم نحفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:200px; height:70px;
    display:flex; align-items:flex-start;
  }
  div#d1{background:#0e0; width:100px; flex-shrink:0; }
  div#d2{background:red; width:100px; flex-shrink:0; }
  div#d3{background:#00e; width:100px; flex-shrink:0; }
</style>
</head>
<body>
<div class="parent">
  <div id="d1">D1</div><div id="d2">D2</div>
  <div id="d3">D3</div>
</div>
</body>
</html>
```

وبءء ءشءعل الصءفة ءظهء كما بالشكل ءءالئ:



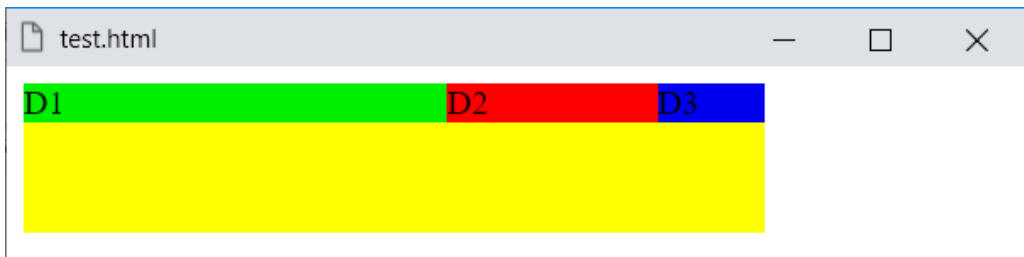
وهنا ءم ءبء الخاصفة flex-shrink لءمفة العناصء child بالءمفة 0، أئ عءء الءءماشء، فأصءءا ءارء الحاوءة parent الأساسفة.

الخاصية flex

يملك جملة قيم flex-grow، flex-shrink، و flex-basis في خاصية واحدة بالترتيب وهي خاصية flex، مثل الكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
div.parent{
background:#ff0; width:350px; height:70px;
display:flex; align-items:flex-start;
}
div#d1{background:#0e0; width:100px; flex:1 0 auto; }
div#d2{background:red; width:100px; flex:0 1 auto; }
div#d3{background:#00e; width:100px; flex:0 1 50px; }
</style>
</head>
<body>
<div class="parent">
<div id="d1">D1</div><div id="d2">D2</div>
<div id="d3">D3</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



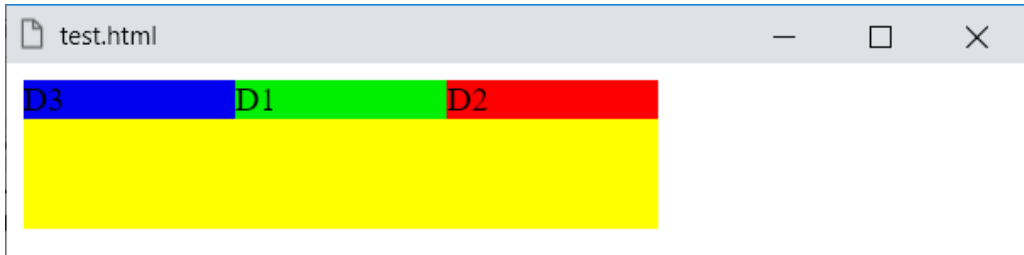
انتبه!، فلا بد من فهم الخواص المذكورة أولاً، لفهم الخاصية flex.

ترتيب العناصر order

بالخاصية order، يمكنك تغيير أماكن العناصر بسهولة، **بالكود!** التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:300px; height:70px;
    display:flex; align-items:flex-start;
  }
  div#d1{background:#0e0; width:100px; order:2; }
  div#d2{background:red; width:100px; order:3; }
  div#d3{background:#00e; width:100px; order:1; }
</style>
</head>
<body>
<div class="parent">
  <div id="d1">D1</div><div id="d2">D2</div>
  <div id="d3">D3</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



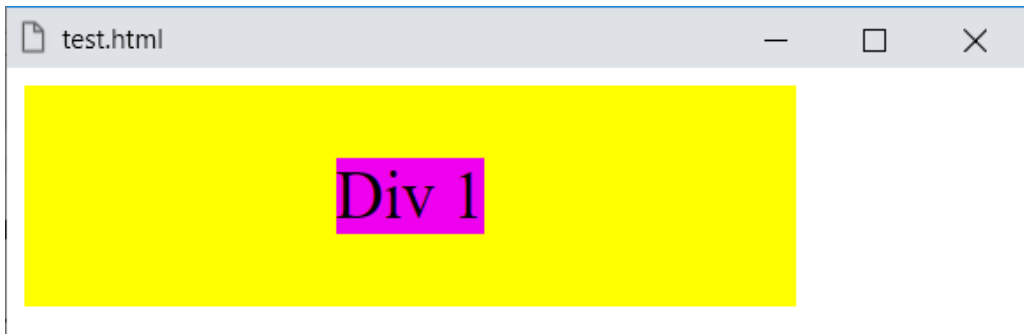
تم وضع d1، ثم d2، ثم d3، ولكنه **انتبه!** أنه بالخاصية order تم تبديل الأماكن بكل مرونة، بغير رقمية، والأولية للرقم الأقل.

Centralize Element توسط العنصر

الآن بفضل الله يمكنك توسيط العنصر في المركز تماماً بسهولة، عن طريق الخاصية `align-items`، والخاصية `justify-content`.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:350px; height:100px;
    display:flex;
    align-items:center; justify-content:center;
  }
  div#d1{ background:#e0e; font-size:30px; }
</style>
</head>
<body>
<div class="parent">
  <div id="d1">Div 1</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



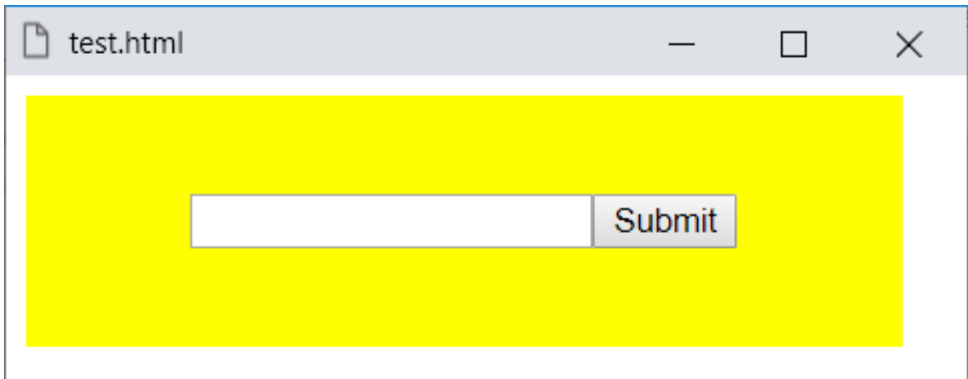
توسيط العناصر Centralize Elements

يملك توسيط عناصر بنفس الطريقة السابقة.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.parent{
    background:#ff0; width:350px; height:100px;
    display:flex;
    align-items:center; justify-content:center;
  }
</style>
</head>
<body>
<div class="parent">
  <input><input type="submit">
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



عرض صور منظمة حسب الرغبة
كود! قم بكتابة الكود التالي ثم احفظه:

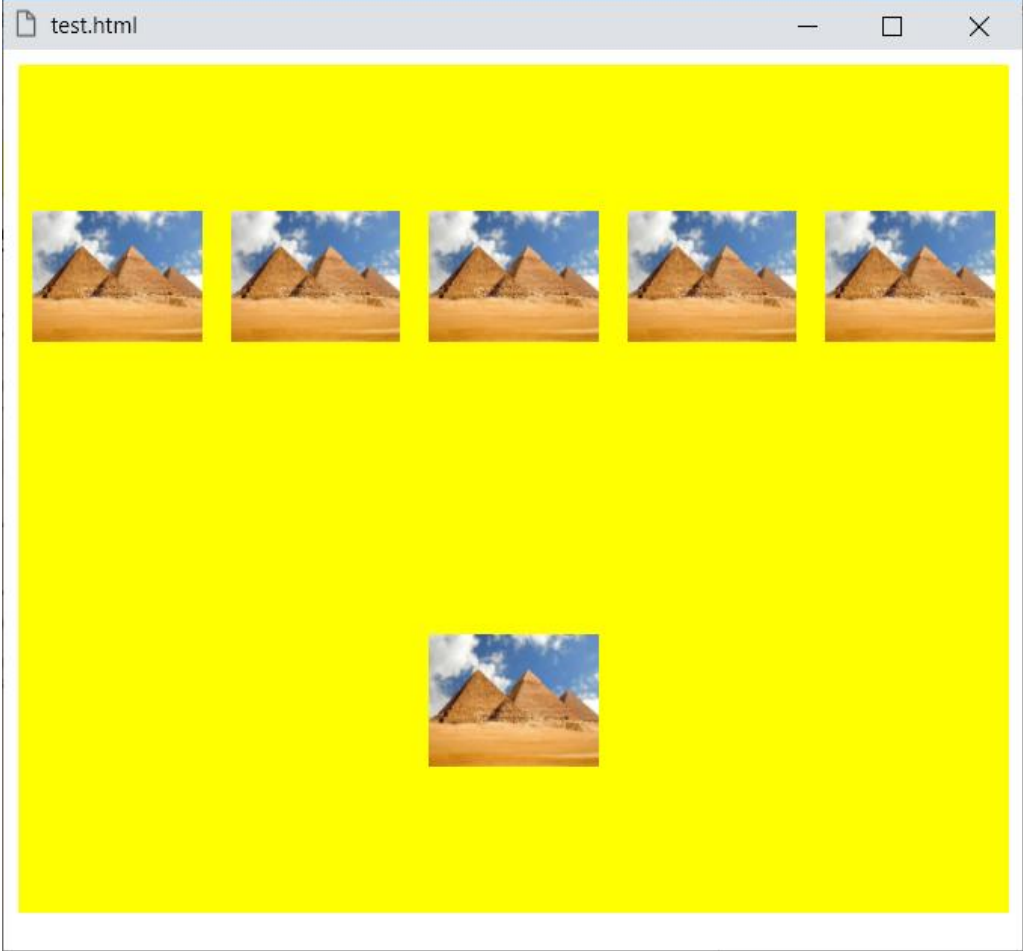
```
<!DOCTYPE html>
<html>
<head>
<style>
img{ width:90px; height:70px; }
div.parent{
background:#fff; width:100%; height:450px;
display:flex; flex-flow:row wrap;
align-items:flex-start;
justify-content:space-around;
align-content:space-around;
}
</style>
</head>
<body>
<div class="parent">
 
 
 
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم تجميع عرض نافذة المتصفح، فكان 6 صور بصف واحد.

والآن قلل عرض نافذة المتصفح قليلاً ، لترى مثل الشكل التالي:

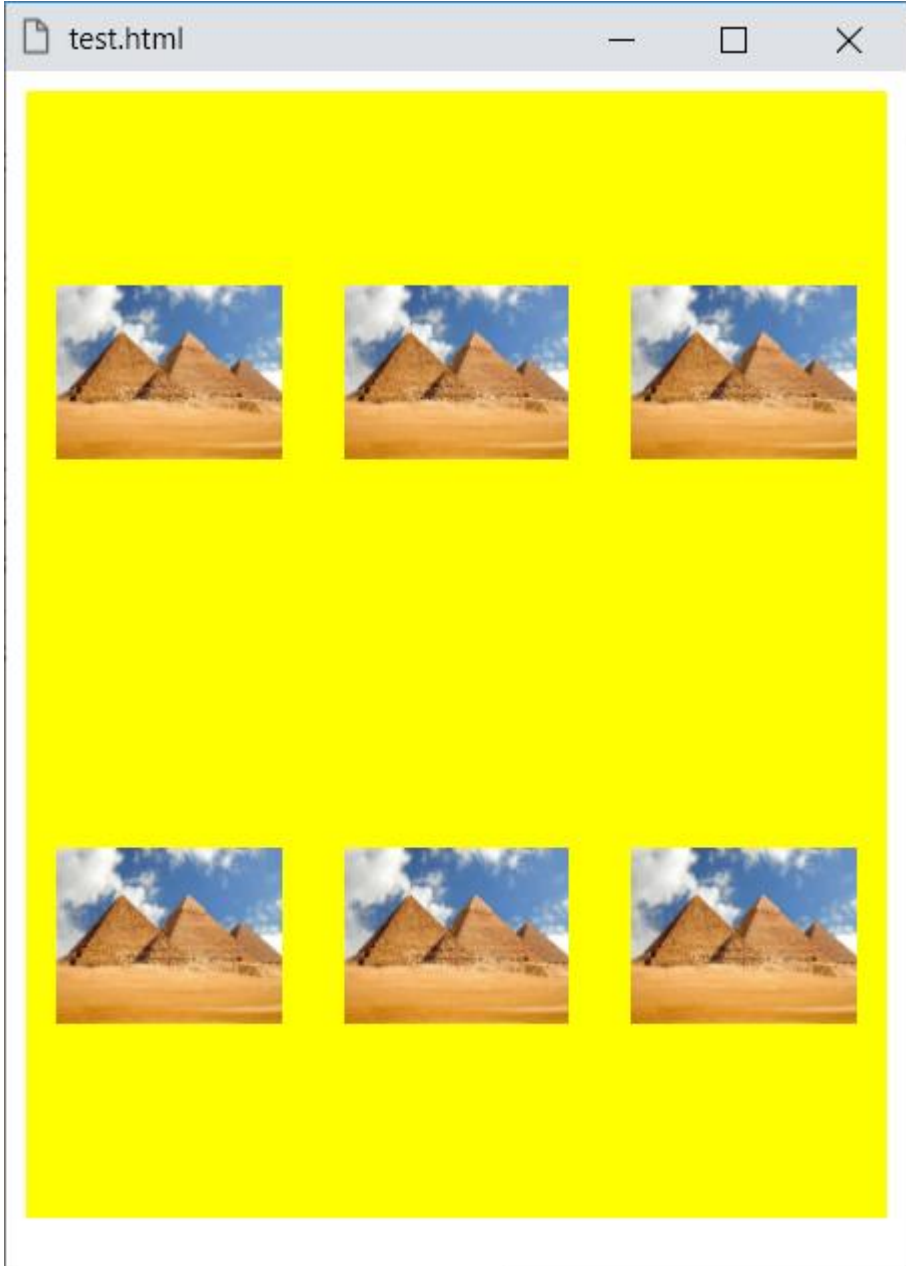


لاحظ أن هناك 5 صور، ثم صورة بالأسفل.

لاحظ أن المسافات يتم توزيعها، بناء على عرض وطول الحاوية الأصلية.

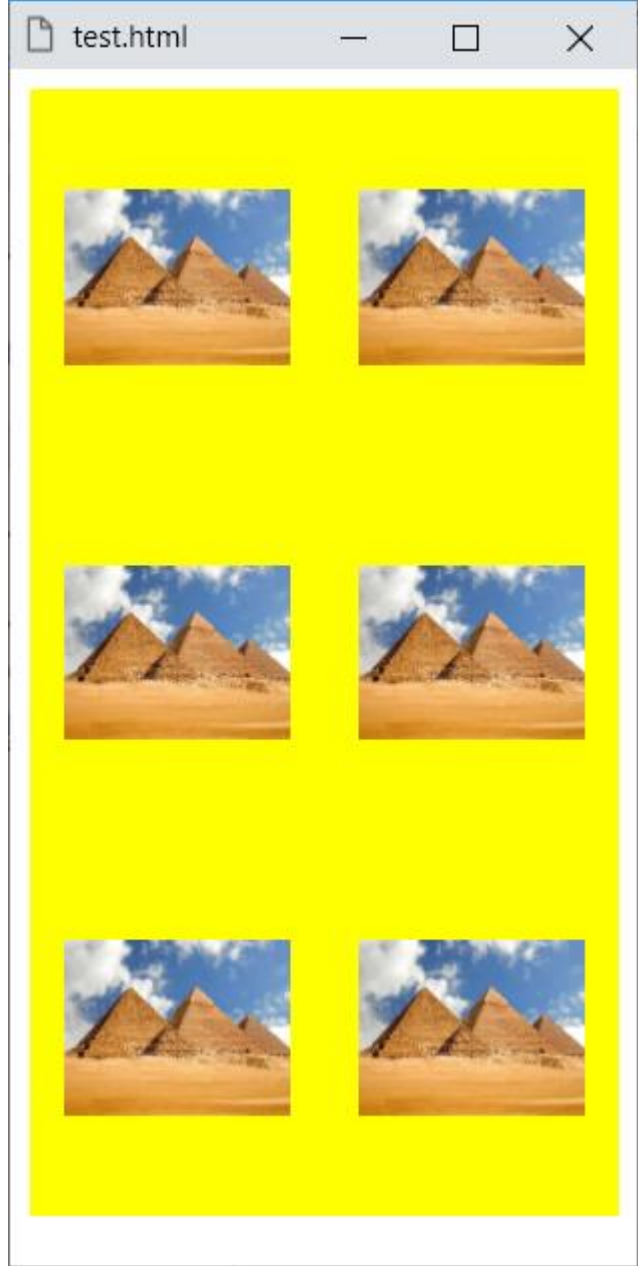
لا تنسى أنه مع الممكّ تغيير التصميم، بما تعلمناه من قبل بفضل الله.

والآن قلل عرضه نافذة المتصفح قليلاً، لترى مثل الشكل التالي:



لاحظ أنه تم توزيع الصور على المساحة المتاحة، فكان 3 ثم 3 أخرى.

والآن قلا عرءه نافءة المءصفا قلاباً ، لءرء مءك الشكك الءالء:



وهكذا ، وبذلك أءبء معك صءءوء مءن ءءءره كما ءرءء بفضل الله.

المؤشر والعلامة Cursor and Caret

علامة الإدخال (الإقحام) في الكتابة caret-color
عند الكتابة في صندوق الإدخال يكون مثل الشكل التالي تارة:

Hassouna Academy|

وبالشكل التالي تارة أخرى:

Hassouna Academy

ولو تحركت بالأسهم، ستظهر العلامة بمكان مختلف مثل الشكل التالي:

Hassouna|Academy

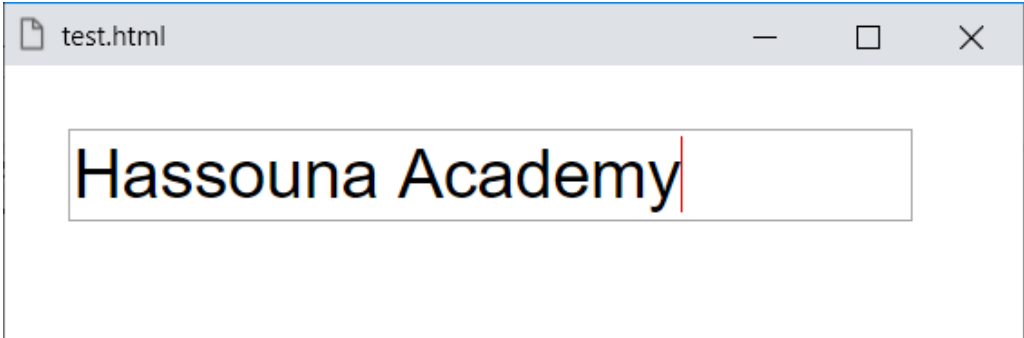
وهذه هي علامة الإدخال أو الإقحام في الكتابة

ويمكن تغيير هذه العلامة عن طريق الخاصية caret-color، والتي
تأخذ قيمة افتراضية لها وهي auto، ويمكنه لون ليتغير لون العلامة
باللون المحدد.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{ padding:20px; }
  input{
    font-size:30px; outline:none;
    caret-color:red;
  }
</style>
</head>
<body>
  <input value="Hassouna Academy">
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

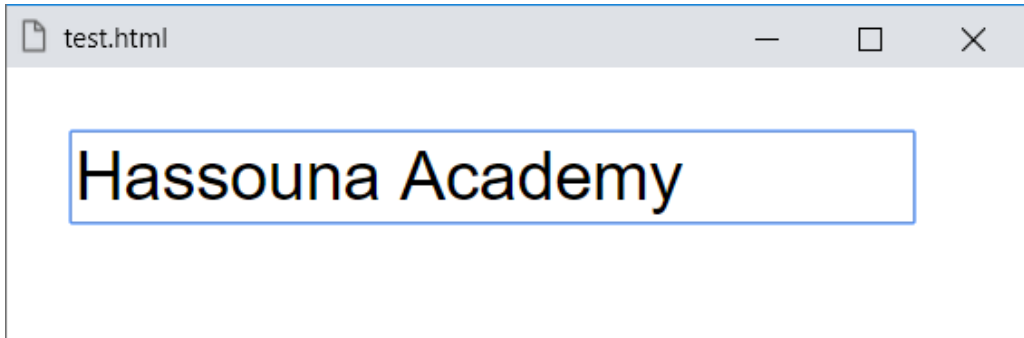


لاحظ أن العلامة أصبحت باللون الأحمر.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body{ padding:20px; }
      input{
        font-size:30px;
        caret-color:transparent;
      }
    </style>
  </head>
  <body>
    <input value="Hassouna Academy">
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

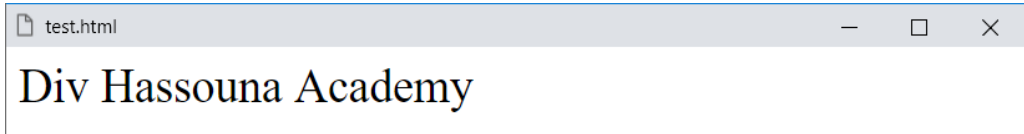


لاحظ أن العلامة تم إخفائها لأنها أخذت لون شفاف، فلم تعد ظاهرة أثناء التعامل مع مربع النص.

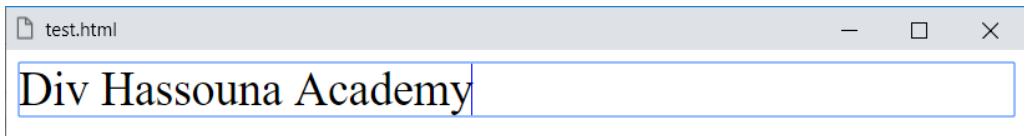
إذا كنت تريد أن تدي علامة الإدخال في عنصر مغلق، مثل الـ `p`، أو الـ `div`، استخدم `contenteditable`، بالكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{ padding:20px; }
  div{
    font-size:30px; caret-color:blue;
  }
</style>
</head>
<body>
  <div contenteditable>Div Hassouna Academy</div>
</body>
</html>
```

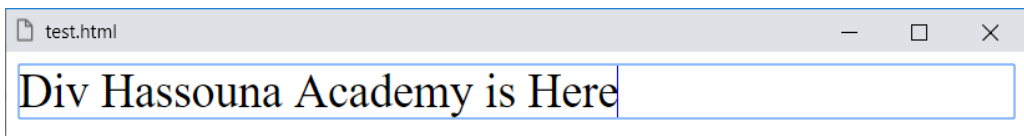
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:




انتبه!، فهو يبدأ طبيعياً، ولكنه لو ضغطت عليه بالماوس سيكوّن كالتالي:






























ولو عدت عليه سيقبل التعديل، مثل الشكل التالي:










CURSOR المؤشر

معروف أن المؤشر هو شكل الماوس والافتراضي له هو  ويمكن تغيير هذا الشكل عن طريق الخاصية CURSOR، وكل قيمة يتم تخصيصها للخاصية CURSOR لها شكل معين، والجدول التالي يوضح قيم المؤشر:

Category	CSS value	Show	Description
General	auto		The UA will determine the cursor to display based on the current context. E.g., equivalent to text when hovering text.
	default		The platform-dependent default cursor. Typically, an arrow.
	none		No cursor is rendered.
Links & status	context-menu		A context menu is available.
	help		Help information is available.
	pointer		The cursor is a pointer that indicates a link. Typically, an image of a pointing hand.
	progress		The program is busy in the background, but the user can still interact with the interface (in contrast to wait).
	wait		The program is busy, and the user can't interact with the interface (in contrast to progress). Sometimes an image of an hourglass or a watch.
Selection	cell		The table cell or set of cells can be selected.
	crosshair		Cross cursor, often used to indicate selection in a bitmap.
	text		The text can be selected. Typically, the shape of an I-beam.
	vertical-text		The vertical text can be selected. Typically, the shape of a sideways I-beam.

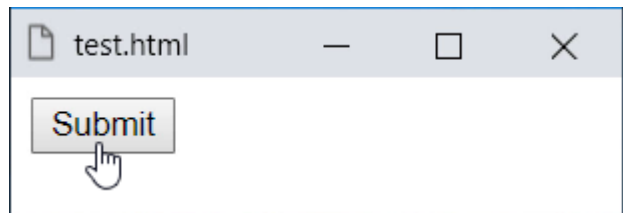
Drag & drop	alias		An alias or shortcut is to be created.
	copy		Something is to be copied.
	move		Something is to be moved.
	no-drop		An item may not be dropped at the current location. {{bug("275173")}}: On Windows and Mac OS X, no-drop is the same as not-allowed.
	not-allowed		The requested action will not be carried out.
	grab		Something can be grabbed (dragged to be moved).
	grabbing		Something is being grabbed (dragged to be moved).
Resizing & scrolling	all-scroll		Something can be scrolled in any direction (panned). {{bug("275174")}}: On Windows, all-scroll is the same as move.
	col-resize		The item/column can be resized horizontally. Often rendered as arrows pointing left and right with a vertical bar separating them.
	row-resize		The item/row can be resized vertically. Often rendered as arrows pointing up and down with a horizontal bar separating them.
	n-resize		Some edge is to be moved. For example, the se-resize cursor is used when the movement starts from the <i>south-east</i> corner of the box. In some environments, an equivalent bidirectional resize cursor is shown. For example, n-resize and s-resize are the same as ns-resize.
	e-resize		
	s-resize		
	w-resize		
	ne-resize		
	nw-resize		
se-resize			

	sw-resize		
	ew-resize		Bidirectional resize cursor.
	ns-resize		
	nesw-resize		
	nwse-resize		
Zooming	zoom-in		Something can be zoomed (magnified) in or out.
	zoom-out		

لاحظ أن كل قيمة في الجدول السابق أمامها شكل المؤشر التي تؤدي إليه على الصفحة أو العنصر، ويمكنك التجربة **بالكود!** التالي:

```
<!DOCTYPE html>
<html>
  <head></head>
  <body>
    <input type="submit" style="cursor:pointer">
  </body>
</html>
```

وبعد تشغيل الصفحة قم بلمس الزر بالماوس لترى مثل الشكل التالي:



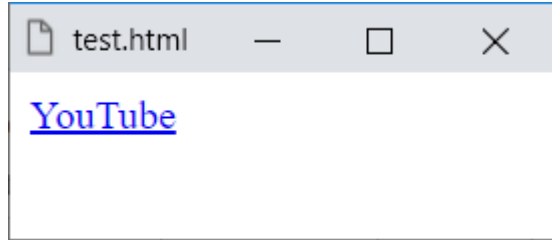
لاحظ أن القيمة pointer جعلت شكل الماوس على الزر مثل اليد وكأنه رابط، وبنفس الطريقة تستطيع التغيير للشكل حسب الاحتياج.

حذف المؤشر pointer-events

يمكنك إلغاء حذف المؤشر، حيث لا يظهر يد hand، ولا يذهب إلى أي مكان، حتى ولو تم الضغط عليه، لا يؤثر.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      a{ pointer-events: none; }
    </style>
  </head>
  <body>
    <a href="https://www.youtube.com">YouTube</a>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



والآن إذا تم الضغط على هذا الرابط له يعمل.

التقطيع clipping

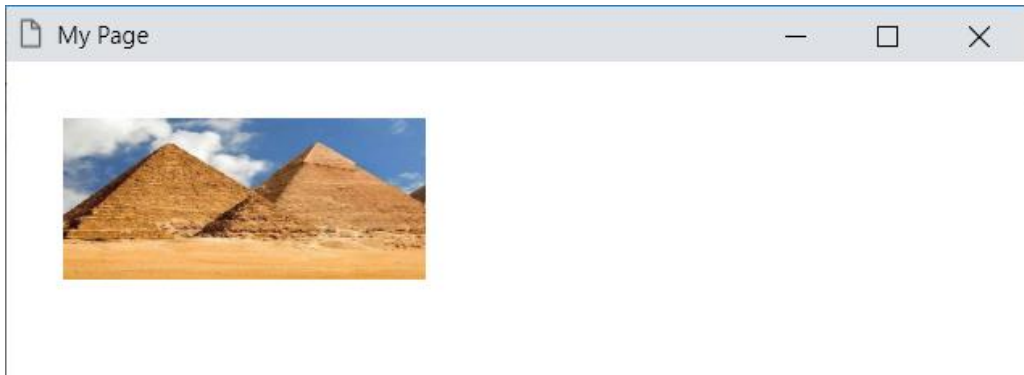
القصاص clip

يمكنك قص قطعة محددة من الصورة، أو العنصر، باستخدام الخاصية clip، والتي تأخذ قيمة افتراضية auto، ويمكن إعطاؤها الدالة rect، فتأخذ top، ثم right، ثم bottom، ثم left، ليتم استخلاص القطعة المراد بقاؤها، وسنجرّب صورة **بالكود!** التالي:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
#i1{
position:absolute;
clip:rect(20px,200px,100px,20px);
}
</style>
</head>
<body>

</body>
</html>
```

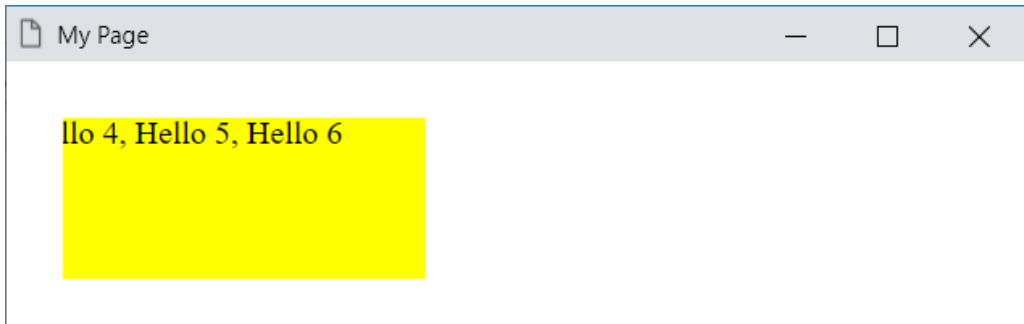
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم ضبط الخاصية position على القيمة absolute، وتم قطع 20px من الأعلى، ثم اظهار حتى 200px من اليمين، ثم اظهار حتى 100px من الأسفل، ثم قطع 20px من اليسار، ويمكن أيضاً التعامل مع العناصر، وسنجد صورة **بالكود!** التالي:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
#d1{
width:250px; height:120px;
background:yellow; position:absolute;
clip:rect(20px,200px,100px,20px);
}
</style>
</head>
<body>
<div id="d1">
Hello 1, Hello 2, Hello 3 <br>
Hello 4, Hello 5, Hello 6
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن الـ div لا يظهر بالكامل.

مسار القص clip-path

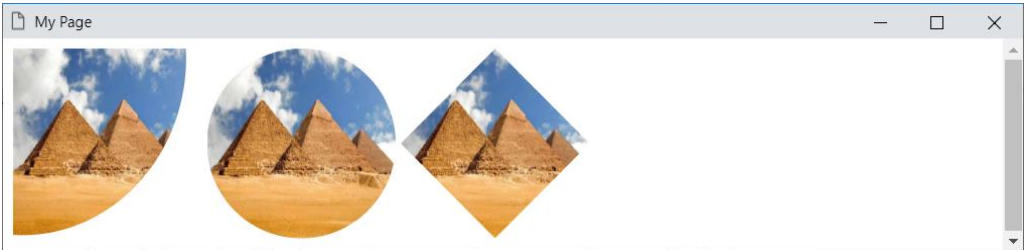
خاصية clip-path تمكنك من القص بطريقة مسار يتم اعطاؤه لها.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
#i1{ clip-path:ellipse(130px 140px at 5% 5%); }
#i2{ clip-path:circle(50%); }
#i3{ clip-path:polygon(50% 0, 100% 50%, 50% 100%, 0 50%); }
</style>
</head>
<body>



</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم قطع شكل بيضاوي، ودائرة، ومضلع، ولكنه **انتبه!** فله يتم الشرح أو التفصيل لهذا الموضوع، في هذا الكتاب، حيث إن التفصيل في مواضيع الرسم يخص الـ SVG، وإن شاء الله يمكنك البحث عن كتاب صخرة الأساس في SVG، لتفصيلات أكثر مع الرسم بصفحة الويب.

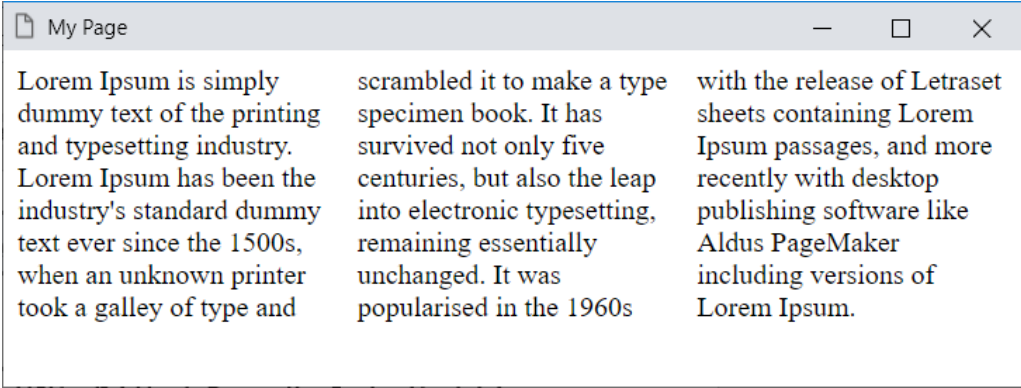
الأعمءة المءعءة Multi Columns

ءءء الأعمءة column-count

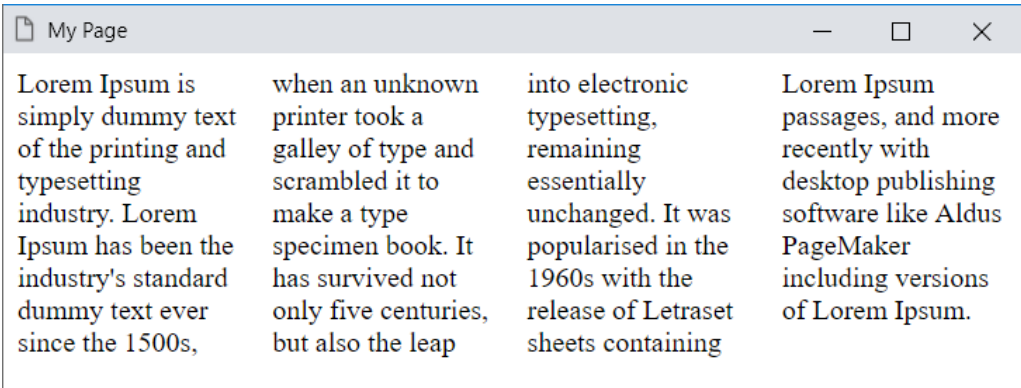
الءاصفة column-count قفمءءا الافءءاءفة هف auto، أو ءأء ءءء، لفاء ءقسفم النص الءاص ءك الف مءموءة مء الأعمءة، مءء ءنظفم الءراءء والمءلاء، وسوف ءءب وسم ءفء ءءء ءءء ءفء ءفء للءءربة، وسنوضء **بالءوء!** ءءالف:

```
<!DOCTYPE html>
<html>
<head>
  <title>My Page</title>
  <style>
    ءفء{
      column-count:3;
    }
  </style>
</head>
<body>
  <ءفء>
    Lorem Ipsum is simply dummy text of the printing and typesetting industry.
    Lorem Ipsum has been the industry's standard dummy text ever since the 1500s,
    when an unknown printer took a galley of type and scrambled it to make a type specimen book.
    It has survived not only five centuries, but also the leap into electronic typesetting,
    remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset
    sheets containing Lorem Ipsum passages, and more recently with desktop publishing software
    like Aldus PageMaker including versions of Lorem Ipsum.
  </ءفء>
</body>
</html>
```


وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن النص تحول الى ثلاث أعمدة، ومع القيمة 4، سيكون كالتالي:



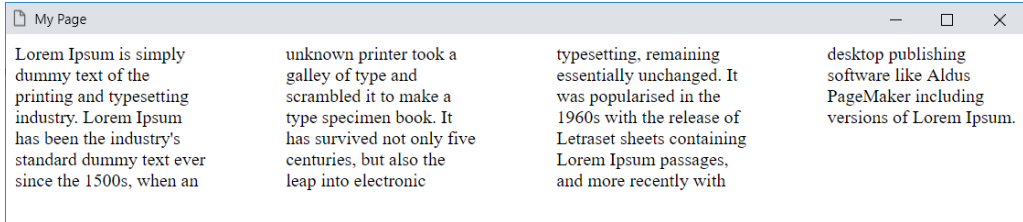
وهكذا، وحسب الاحتياج لتقسيم النص الخاص بك بفضل الله بكل سهولة.

الفارق بين الأعمدة column-gap

يمكّن وضع مسافات فارقة بين كل عمود وآخر، عن طريق الخاصية column-gap، وسنجرّب بالكوڤ! التالي:

```
<!DOCTYPE html>
<html>
<head>
  <title>My Page</title>
  <style>
    div{
      column-count:4; column-gap:70px;
    }
  </style>
</head>
<body>
  <div>
    Lorem Ipsum is simply dummy text of the printing and typesetting industry.
    Lorem Ipsum has been the industry's standard dummy text ever since the 1500s,
    when an unknown printer took a galley of type and scrambled it to make a type specimen book.
    It has survived not only five centuries, but also the leap into electronic typesetting,
    remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset
    sheets containing Lorem Ipsum passages, and more recently with desktop publishing software
    like Aldus PageMaker including versions of Lorem Ipsum.
  </div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



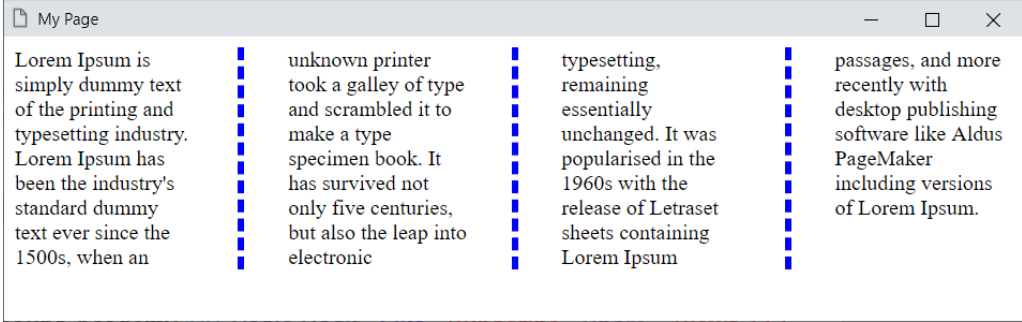
لاحظ أنه تم وضع فارق بين كل عمود وآخر بمقدار 70px.

column-rule الأعمدة

يمكنك وضع فواصل مثل الإطار border، بين كل عمود، والعمود الآخر،
عن طريق الخاصية **column-rule-style** لنمط الفاصل،
والخاصية **column-rule-width** لسمك الفاصل، والخاصية
column-rule-color للون الفاصل، وسنجد **باللورد!** التالي:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
div{
column-count:4; column-gap:70px;
column-rule-style:dashed;
column-rule-width:5px;
column-rule-color:blue;
}
</style>
</head>
<body>
<div>
Lorem Ipsum is simply dummy text of the printing and typesetting industry.
Lorem Ipsum has been the industry's standard dummy text ever since the 1500s,
when an unknown printer took a galley of type and scrambled it to make a type specimen book.
It has survived not only five centuries, but also the leap into electronic typesetting,
remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset
sheets containing Lorem Ipsum passages, and more recently with desktop publishing software
like Aldus PageMaker including versions of Lorem Ipsum.
</div>
</body>
</html>
```

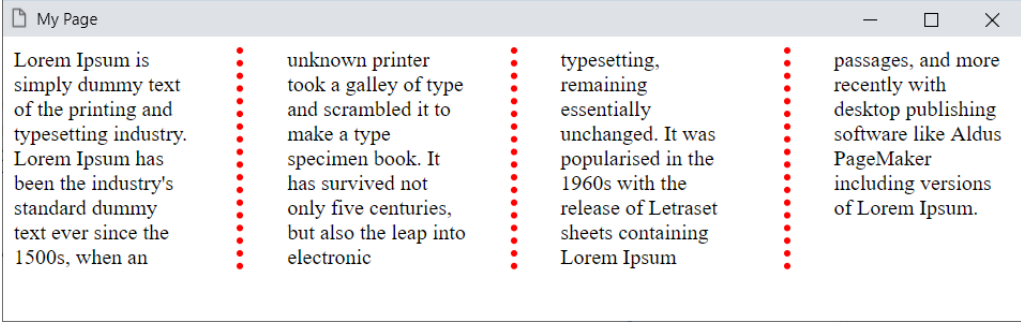
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



ويمك اختصارهم في خاصية واحدة، وهي `column-rule`.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
div{
column-count:4; column-gap:70px;
column-rule:5px dotted red;
}
</style>
</head>
<body>
<div>
Lorem Ipsum is simply dummy text of the printing and typesetting industry.
Lorem Ipsum has been the industry's standard dummy text ever since the 1500s,
when an unknown printer took a galley of type and scrambled it to make a type specimen book.
It has survived not only five centuries, but also the leap into electronic typesetting,
remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset
sheets containing Lorem Ipsum passages, and more recently with desktop publishing software
like Aldus PageMaker including versions of Lorem Ipsum.
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



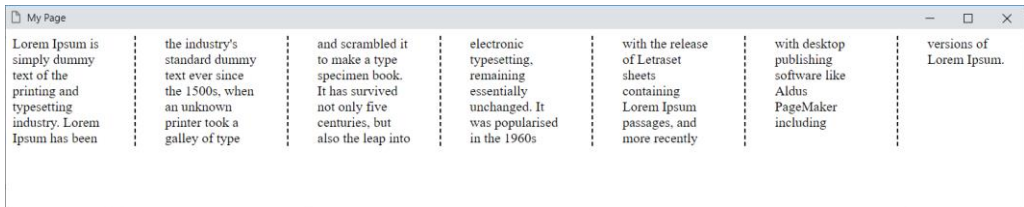
انتبه! فكل ما تعلمته في دروس الإطار border، سيساعدك هنا كثيراً
بفضل الله.

عرض العمود column-width

بالخاصية column-width يمكن تجاهل column-count،
وتحديد عرض العمود، وسنجرّب **بالكود!** التالي:

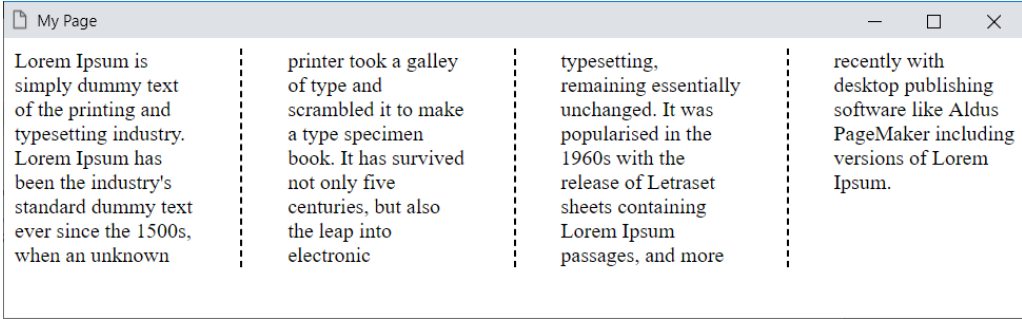
```
<!DOCTYPE html>
<html>
<head>
  <title>My Page</title>
  <style>
    div{
      column-gap:70px;
      column-rule:2px dashed;
      column-width:100px;
    }
  </style>
</head>
<body>
  <div>
    Lorem Ipsum is simply dummy text of the printing and typesetting industry.
    Lorem Ipsum has been the industry's standard dummy text ever since the 1500s,
    when an unknown printer took a galley of type and scrambled it to make a type specimen book.
    It has survived not only five centuries, but also the leap into electronic typesetting,
    remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset
    sheets containing Lorem Ipsum passages, and more recently with desktop publishing software
    like Aldus PageMaker including versions of Lorem Ipsum.
  </div>
</body>
</html>
```

وبعد تشغيل الصفحة بعرض نافذة كبير تظهر كما بالشكل التالي:

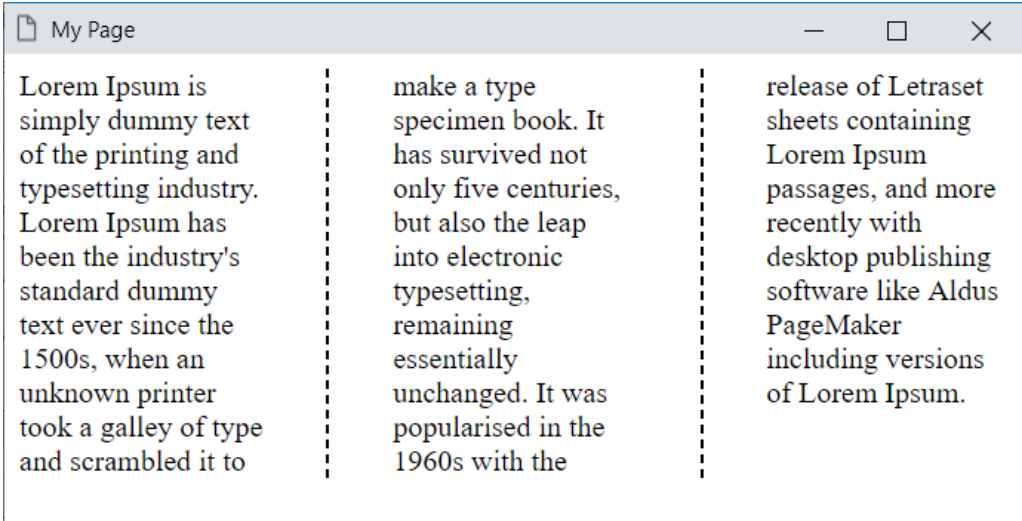


لاحظ أنه تم التقسيم حسب العرض الموجود، لكل عمود 100px، كما
تم ضبط الخاصية column-width.

والآن قم بتصغير نافذة المتصفح، لترى مثل الشكل التالي:



والآن قم بتصغير نافذة المتصفح مرة أخرى، لترى مثل الشكل التالي:



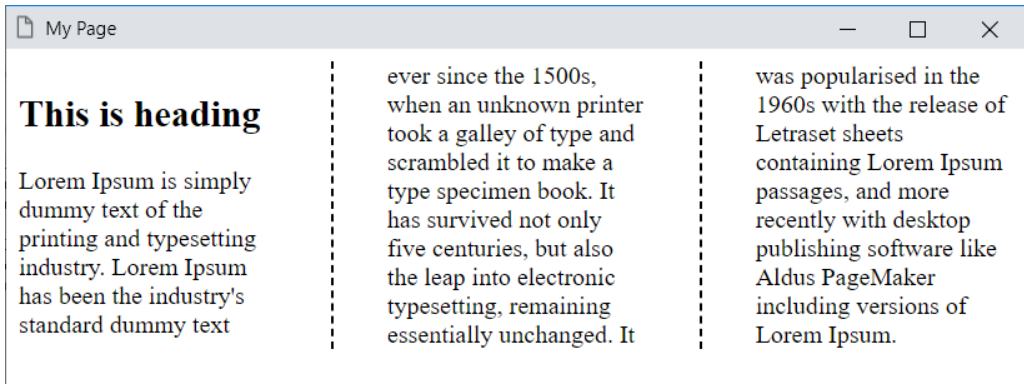
ولو قلت نافذة المتصفح عن الـ 100px، له يتم أي تقسيم.

عنوان مءمء للأعمءة column-span

بمءك وءءء عنوان بشكء عاءء، وسءءءب ذلك بالءوء! الءالء:

```
<!DOCTYPE html>
<html>
<head>
  <title>My Page</title>
  <style>
    #d1{ column-count:3; column-gap:70px; column-rule:2px dashed; }
  </style>
</head>
<body>
  <div id="d1">
    <h2>This is heading</h2>
    Lorem Ipsum is simply dummy text of the printing and typesetting industry.
    Lorem Ipsum has been the industry's standard dummy text ever since the 1500s,
    when an unknown printer took a galley of type and scrambled it to make a type specimen book.
    It has survived not only five centuries, but also the leap into electronic typesetting,
    remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset
    sheets containing Lorem Ipsum passages, and more recently with desktop publishing software
    like Aldus PageMaker including versions of Lorem Ipsum.
  </div>
</body>
</html>
```

وبءء ءشءءل الصءفة ءظهء كما بالشكء الءالء:

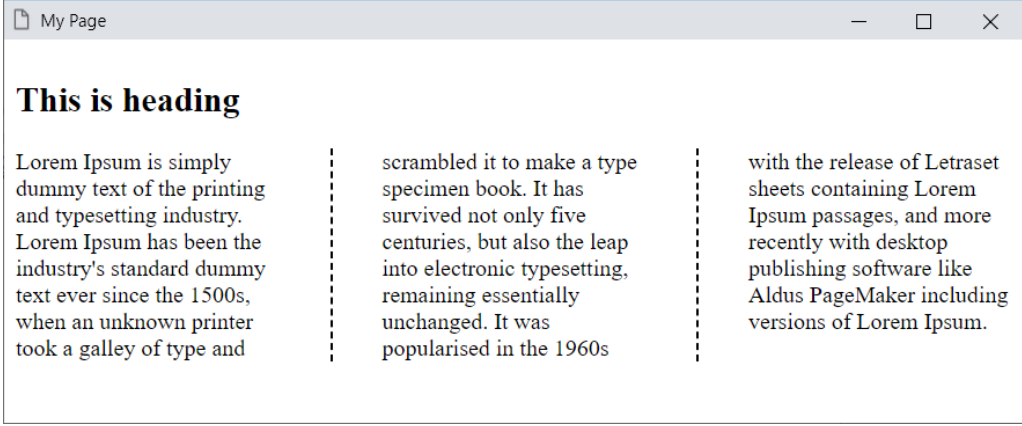


لاءظ أن العنوان فظهر مءمء لعموء واءء فقط.

ولجعل العنوان يمتد لجميع الأعمدة الموجودة، نخصص قيمة للخاصية `column-span`، والتي تأخذ قيمة افتراضية وهي `none`، أي لا شيء، ويمكن ضبطها على قيمة `all`، لتمتد على جميع الأعمدة الموجودة.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
#d1{
  column-count:3; column-gap:70px;
  column-rule:2px dashed;
}
#d1 h2{
  column-span:all;
}
</style>
</head>
<body>
<div id="d1">
<h2>This is heading</h2>
Lorem Ipsum is simply dummy text of the printing and typesetting industry.
Lorem Ipsum has been the industry's standard dummy text ever since the 1500s,
when an unknown printer took a galley of type and scrambled it to make a type specimen book.
It has survived not only five centuries, but also the leap into electronic typesetting,
remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset
sheets containing Lorem Ipsum passages, and more recently with desktop publishing software
like Aldus PageMaker including versions of Lorem Ipsum.
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن التنسيق المقصود يخص الـ h2 الموجودة في العنصر d1، ولو كنت تريد تنسيق عنوان واحد فقط، يمكنك إعطاؤه id، أو class، ثم التنسيق عليه.

لاحظ أن العنوان تمتد فوق جميع الأعمدة، بعد تخصيص القيمة all للخاصية column-span، للعنوان.

تعبئة الأعمدة column-fill

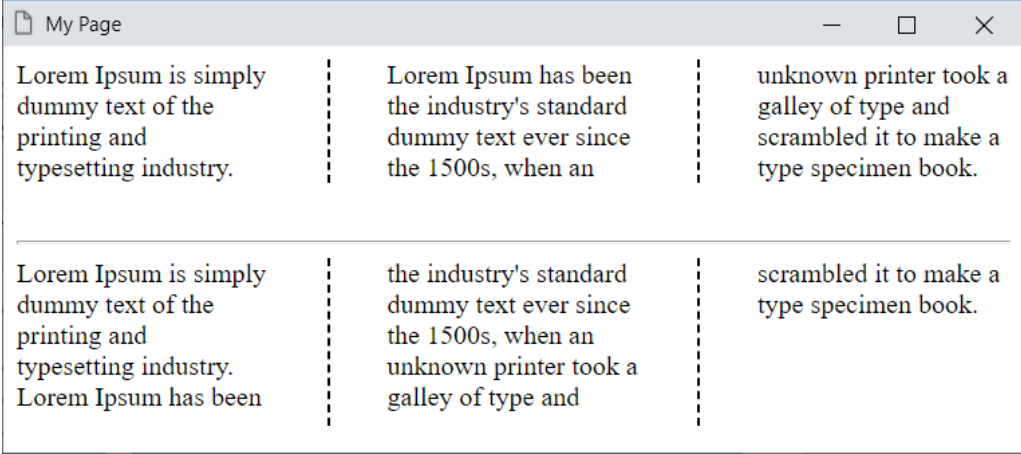
الخاصية column-fill، القيمة الافتراضية لها هي **balance**، حيث يملأ كل عمود بنفس المقدار من المحتوى النصي، ولكه له يسمح للأعمدة أن تكون أطول من الطول height، لذلك قد تكون الأعمدة أقصر من الطول height حيث يقوم المتصفح بتوزيع المحتوى بشكل أفقي بالتساوي، وهذه الخاصية تأخذ قيمة أخرى وهي القيمة **auto**، حيث يملأ كل عمود حتى يصل إلى الطول height، ويفعل ذلك حتى ينتهي من المحتوى النصي الموجود، لذلك ليس بالضرورة أن تملأ هذه القيمة جميع الأعمدة ولا تملأها بالتساوي، وللتجربة لابد من تحديد الطول height، وعمل وسمي div مختلف في هذه الخاصية لتلاحظ التأثير، وتلاحظ الفرق بين أعمدة القيمة balance، وأعمدة القيمة auto، على طول height، وليك مثلاً بمقدار 100px، للتجربة، وأيضاً سيتم عمل فوارق gap، وفواصل rule بين الأعمدة، لتكون واضحة بشكل كبير، وسوف يتم تجربة كل ما تم ذكره بفضل الله، في المثال التالي.

انتبه! لأنه سوف يتم وضع نفس المحتوى النصي في كل من الـ div، الذي سوف يتم عملهم.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
div{
column-count:3; column-gap:70px;
column-rule:2px dashed; height:100px;
}
#d1{ column-fill:balance; }
#d2{ column-fill:auto; }
</style>
</head>
<body>
<div id="d1">
Lorem Ipsum is simply dummy text of the printing and typesetting industry.
Lorem Ipsum has been the industry's standard dummy text ever since the 1500s,
when an unknown printer took a galley of type and scrambled it
to make a type specimen book.
</div>
<hr>
<div id="d2">
Lorem Ipsum is simply dummy text of the printing and typesetting industry.
Lorem Ipsum has been the industry's standard dummy text ever since the 1500s,
when an unknown printer took a galley of type and scrambled it
to make a type specimen book.
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



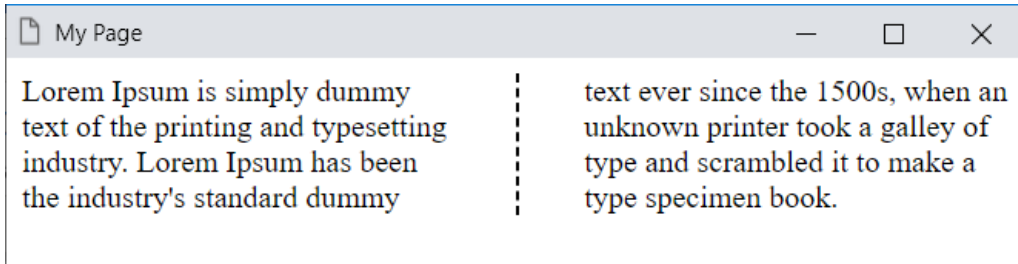
لاحظ أن الـ `div` الذي تم ضبطه على القيمة `balance` كعمود فيه يحتوي على أربع أسطر، بينما الذي تم ضبطه على القيمة `auto` كعمود فيه يحتوي على خمسة أسطر، مع العلم أن الـ `div` تم ضبطه على طول `height` واحد بمقدار `100px`، لتتم مقارنتهم.

خاصية الأعمدة columns

هي خاصية مختصرة لخاصية العرض column-width، ثم خاصية العدد column-count، وسنجرّب **بالكود!** التالي:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
div{
column-rule:2px dashed; column-gap:70px;
columns:100px 2;
}
</style>
</head>
<body>
<div>
Lorem Ipsum is simply dummy text of the printing and typesetting industry.
Lorem Ipsum has been the industry's standard dummy text ever since the 1500s,
when an unknown printer took a galley of type and scrambled it
to make a type specimen book.
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم تحديد العرض والعدد بالخاصية columns فقط.

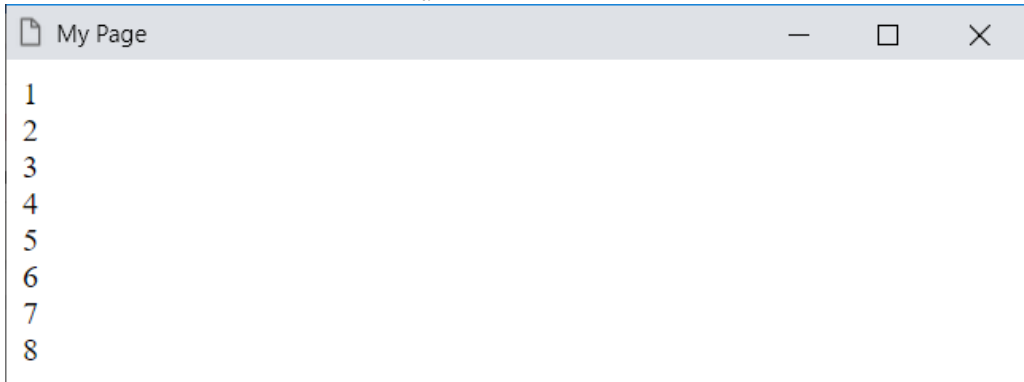
النظام الشبكي Grid Layout

أكثر شيءٍ يملك تشبيه النظام الشبكي به، هو الجدول، حيث به أعمدة، وصفوف، وخلايا، ويملك الدمج وتوزيع العناصر، ويملك إعطاء نسب مئوية، لجعل العناصر قابلة للتجسيم عند تغيير العرض، وهنا مع نظام الـ grid يكون الوندج في غاية الروعة، قم بالبداية العادية، كما يلي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Page</title>
  </head>
  <body>
    <div>
      <div>1</div><div>2</div>
      <div>3</div><div>4</div>
      <div>5</div><div>6</div>
      <div>7</div><div>8</div>
    </div>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



قالب الأعمدة grid-template-columns

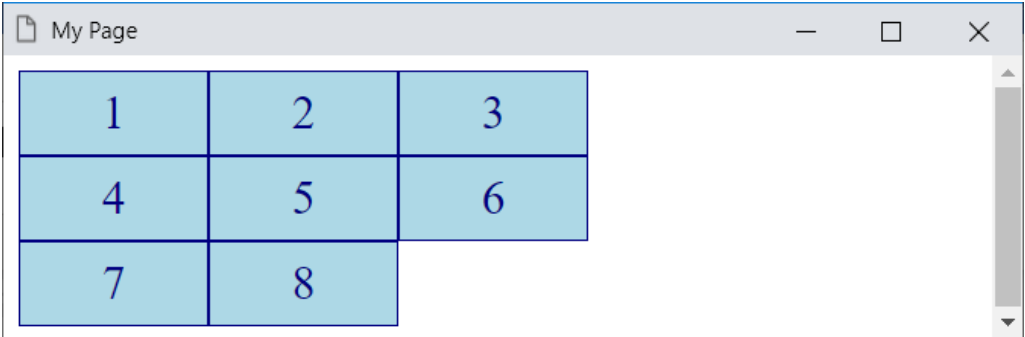
أولاً ، لابد من تحديد القيمة grid للخاصية display، ثم تحديد القالب الذي سوف يحدد الأعمدة، وسنتفهم أكثر إن شاء الله مع التجربة. والآن سنبدأ في التغيير، بالكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
div.grid{
display:grid;
grid-template-columns:50px 50px;
}
div.grid>.item{
background:lightblue; color:navy;
padding:7px; font-size:25px; text-align:center;
border:1px solid navy;
}
</style>
</head>
<body>
<div class="grid">
<div class="item">1</div><div class="item">2</div>
<div class="item">3</div><div class="item">4</div>
<div class="item">5</div><div class="item">6</div>
<div class="item">7</div><div class="item">8</div>
</div>
</body>
</html>
```


وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



والآن **انتبه!** معي جيداً، فلا أريدك أن تدرك على تنسيقات العناصر الداخلية `item` في الوقت الحالي، فلها معرفة، وتم شرحها من قبل، بل أريدك فقط أن تدرك على `grid-template-columns:50px 50px;` لأنه بهذا الكود تم تكوين عموديه، وكل منهما بعرض `50px` كما هو واضح بالشكل، ولو كتبت القيم مثل `100px 100px 100px`، سيكون مثل الشكل التالي:

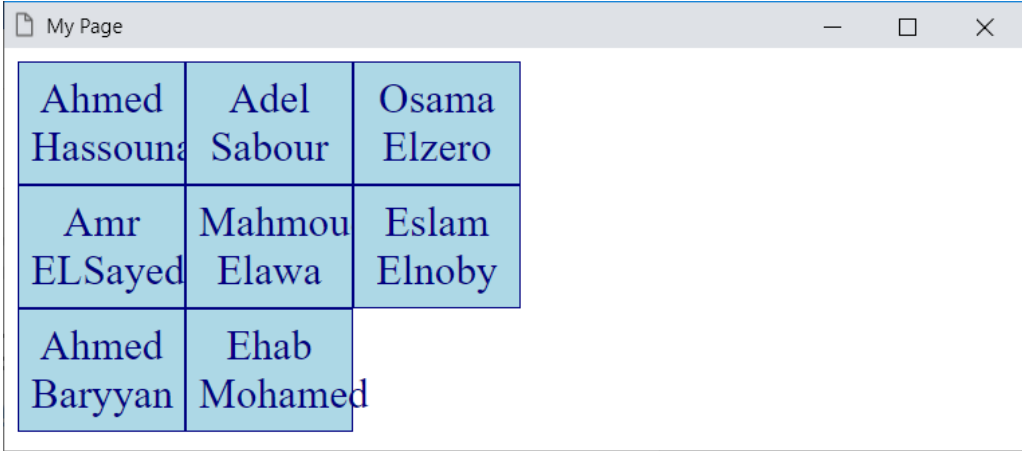


حيث تم تكوين ثلاث أعمدة، وعرض كل منهما `100px`، وهكذا، حيث يمكنك وضع ما تريد في القالب.

كود! قم بكتابة الكود التالي ثم احفظه:

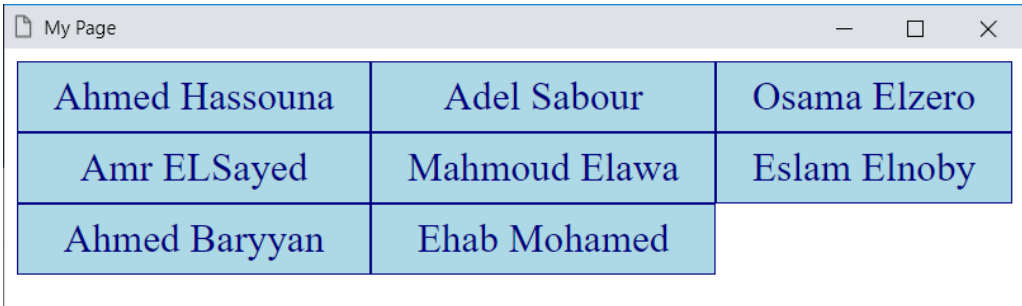
```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
div.grid{
display:grid;
grid-template-columns:100px 100px 100px;
}
div.grid>.item{
background:lightblue; color:navy;
padding:7px; font-size:25px; text-align:center;
border:1px solid navy;
}
</style>
</head>
<body>
<div class="grid">
<div class="item">Ahmed Hassouna</div>
<div class="item">Adel Sabour</div>
<div class="item">Osama Elzero</div>
<div class="item">Amr ELSayed</div>
<div class="item">Mahmoud Elawa</div>
<div class="item">Eslam Elnoby</div>
<div class="item">Ahmed Baryyan</div>
<div class="item">Ehab Mohamed</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



Ahmed Hassouna	Adel Sabour	Osama Elzero
Amr ELSayed	Mahmou Elawa	Eslam Elnoby
Ahmed Baryyan	Ehab Mohamed	

لاحظ أن المحتوى غير ظاهر بالكامل، لأن خاصية قالب الأعمدة أخذت القيم $100px\ 100px\ 100px$ ، فكان المحتوى في بعض الخلايا يحتاج إلى عرض أكثر من $100px$ ، وإذا تغيرت إلى $auto\ auto\ auto$ ، سيختلف الأمر مثل الشكل التالي:



Ahmed Hassouna	Adel Sabour	Osama Elzero
Amr ELSayed	Mahmoud Elawa	Eslam Elnoby
Ahmed Baryyan	Ehab Mohamed	

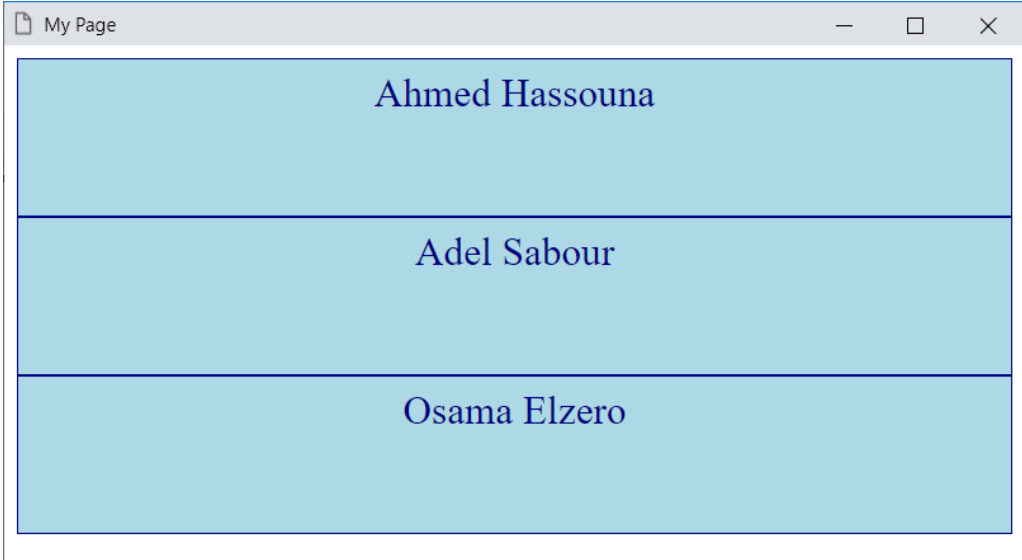
لاحظ أن دور القيمة $auto$ هام جداً في تنسيق خلايا ال $grid$.

قالب الصفوف grid-template-rows

أولاً ، لابد من تحديد القيمة grid للخاصية display، ثم تحديد القالب الذي سوف يحدد الصفوف، وسنتفهم أكثر إن شاء الله مع التجربة.
والآن سنجرب **بالكود!** التالي:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
div.grid{
display:grid;
grid-template-rows:100px 100px 100px;
}
div.grid>.item{
background:lightblue; color:navy;
padding:7px; font-size:25px; text-align:center;
border:1px solid navy;
}
</style>
</head>
<body>
<div class="grid">
<div class="item">Ahmed Hassouna</div>
<div class="item">Adel Sabour</div>
<div class="item">Osama Elzero</div>
</div>
</body>
</html>
```

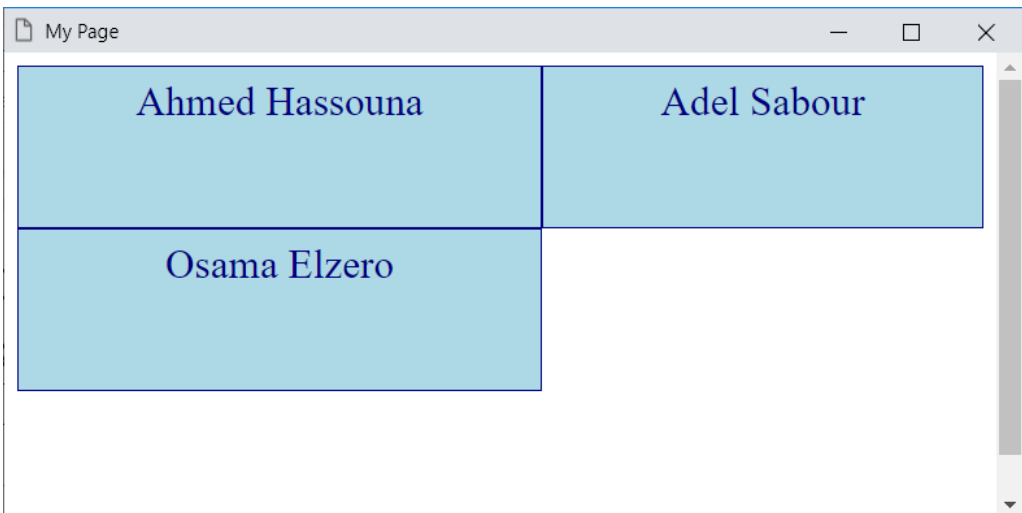
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ، تكون ثلاث صفوف بمقدار 100px لطول كل صف، بدون أعمدة، وهذا لا يمنع استخدام الصفوف والأعمدة، ولو تم زيادة الكود التالي:

```
grid-template-columns:auto auto;
```

سيكون مثل الشكل التالي:



قالب مناطق الخلايا grid-template-areas

إذا أحببت أن أقرب لك المعنى، فسوف أقول لك، اعتبر هذا الموضوع مثل فكرة دمج الخلايا، حيث خاصية grid-template-areas، تحدد مناطق بعض الخلايا، ولكنه أولاً لابد أن يتم إعطاء قيمة باسم معية لخاصية grid-area للعنصر الـ item المراد تحديده منطقتيه، ثم بعد ذلك يتم ضبط خاصية grid-template-areas للـ grid.

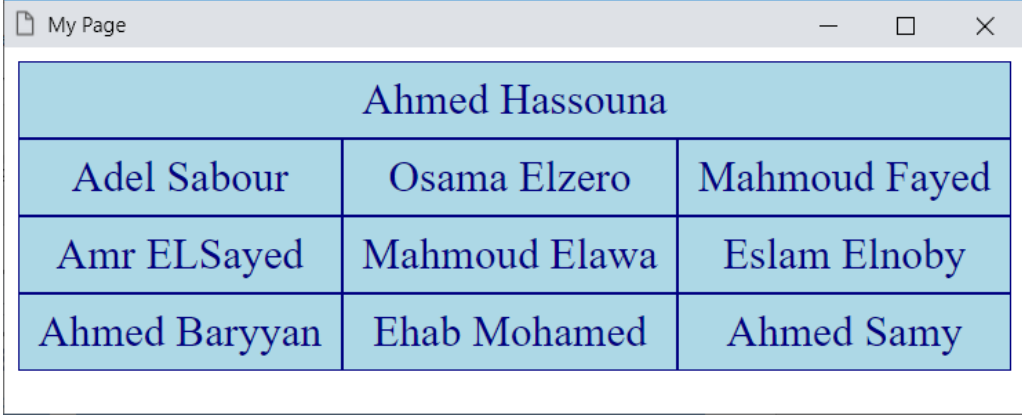
قبل البدء في كتابة الكود في موضوع مناطق الخلايا، نريد أن نتعرف بالتفصيل على كيفية كتابة كودها أولاً، وسوف يتم سرد بعض النقاط بترتيب متعمد للفهم، وهي كما يلي:

- إنشاء اسم جديد للعنصر المراد فقط بالخاصية grid-area، والموجود بداخل الـ grid
- في عنصر الـ grid نفسه، يتم كتابة هذا الاسم مرتين ليأخذ مكان خليتين، أو ثلاثة ليأخذ مكان ثلاث خلايا، أو أربعة ليأخذ مكان أربع خلايا، أو خمسة ليأخذ مكان خمس خلايا، أو ستة ليأخذ مكان ستة خلايا، وهكذا، وبك كل مرة ومرة مسافة، وتتم كل هذه الكتابة في الخاصية grid-template-areas ما بين رمزيه سنجد كون
- لوجود خلايا بجوار الخلية المقصود دمجها، يتم أخذ مسافة بعد تكمارات الاسم ثم الرمز دون، ولوجود خلية أخرى مسافة ثم دون، وهكذا

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <title>My Page</title>
  <style>
    #i1{ grid-area:item1; }
    div.grid{
      display:grid;
      grid-template-columns:auto auto auto;
      grid-template-areas: 'item1 item1 item1';
    }
    div.grid>.item{
      background:lightblue; color:navy;
      padding:7px; font-size:25px; text-align:center;
      border:1px solid navy;
    }
  </style>
</head>
<body>
  <div class="grid">
    <div class="item" id="i1">Ahmed Hassouna</div>
    <div class="item">Adel Sabour</div>
    <div class="item">Osama Elzero</div>
    <div class="item">Mahmoud Fayed</div>
    <div class="item">Amr ELSayed</div>
    <div class="item">Mahmoud Elawa</div>
    <div class="item">Eslam Elnoby</div>
    <div class="item">Ahmed Baryyan</div>
    <div class="item">Ehab Mohamed</div>
    <div class="item">Ahmed Samy</div>
  </div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



Ahmed Hassouna		
Adel Sabour	Osama Elzero	Mahmoud Fayed
Amr ELSayed	Mahmoud Elawa	Eslam Elnoby
Ahmed Baryyan	Ehab Mohamed	Ahmed Samy

لاحظ أنه تم تسمية العنصر المراد، بسطر الكود التالي:

```
#i1{ grid-area:item1; }
```

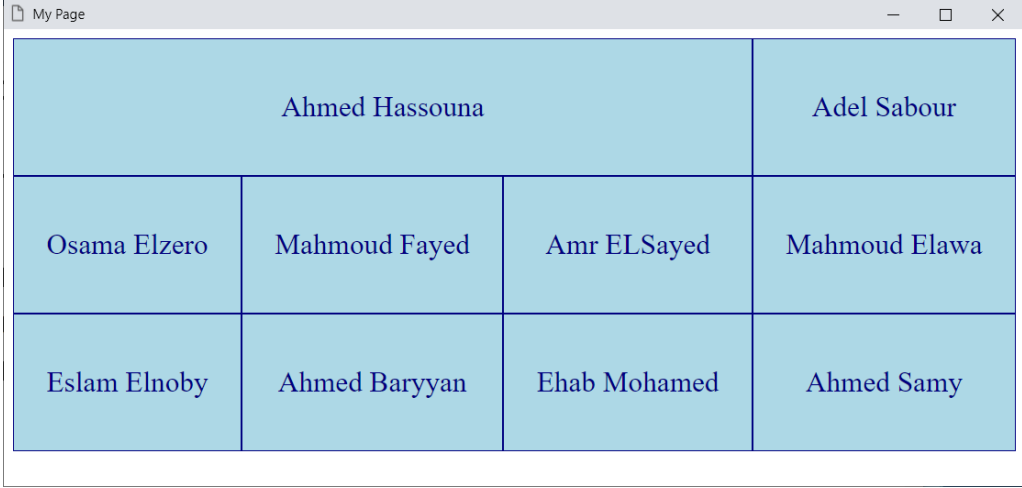
لاحظ أنه تم تكرار الاسم 3 مرات ليدهج تحته ثلاث خلايا بالسطر التالي:

```
grid-template-areas: 'item1 item1 item1';
```


كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
#i1{ grid-area:item1; }
div.grid{
display:grid; grid-template-rows:120px 120px 120px;
grid-template-columns:auto auto auto auto;
grid-template-areas: 'item1 item1 item1 .';
}
div.grid>.item{
background:lightblue; color:navy;
padding:7px; font-size:25px; text-align:center;
border:1px solid navy; display:flex;
align-items:center; justify-content:center;
}
</style>
</head>
<body>
<div class="grid">
<div class="item" id="i1">Ahmed Hassouna</div>
<div class="item">Adel Sabour</div>
<div class="item">Osama Elzero</div>
<div class="item">Mahmoud Fayed</div>
<div class="item">Amr ELSayed</div>
<div class="item">Mahmoud Elawa</div>
<div class="item">Eslam Elnoby</div>
<div class="item">Ahmed Baryyan</div>
<div class="item">Ehab Mohamed</div>
<div class="item">Ahmed Samy</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



Ahmed Hassouna			Adel Sabour
Osama Elzero	Mahmoud Fayed	Amr ELSayed	Mahmoud Elawa
Eslam Elnoby	Ahmed Baryyan	Ehab Mohamed	Ahmed Samy

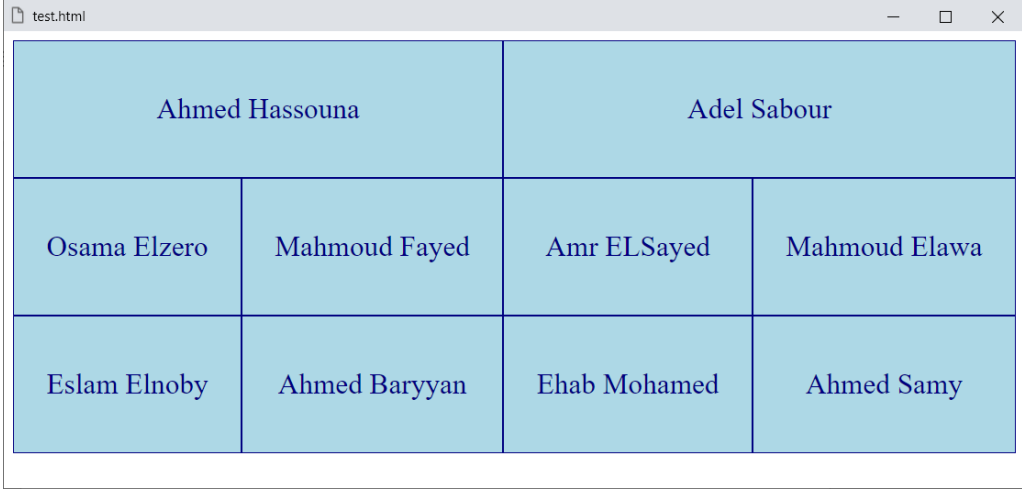
لاحظ أنه تم كتابة اسم العنصر المراد دمجه ثلاث مرات ليصبح ثلاث خلايا، ثم بعد التكرارات الثلاثة هذه، تم كتابة مسافة ثم بعدها الرمز دون بشكل مباشر، وهذا ليكون هناك خلية موجودة بين الخلية التي دمجت ثلاث خلايا.

لاحظ أنه تم تحديد طول كل صف بمقدار 120px، ولاحظ أنه تم استخدام ال flex لجعل المحتوى في المنتصف.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
#i1{ grid-area:item1; }
#i2{ grid-area:item2; }
div.grid{
display:grid; grid-template-rows:120px 120px 120px;
grid-template-columns:auto auto auto auto;
grid-template-areas: 'item1 item1 item2 item2';
}
div.grid>.item{
background:lightblue; color:navy;
padding:7px; font-size:25px; text-align:center;
border:1px solid navy; display:flex;
align-items:center; justify-content:center;
}
</style>
</head>
<body>
<div class="grid">
<div class="item" id="i1">Ahmed Hassouna</div>
<div class="item" id="i2">Adel Sabour</div>
<div class="item">Osama Elzero</div>
<div class="item">Mahmoud Fayed</div>
<div class="item">Amr ELSayed</div>
<div class="item">Mahmoud Elawa</div>
<div class="item">Eslam Elnoby</div>
<div class="item">Ahmed Baryyan</div>
<div class="item">Ehab Mohamed</div>
<div class="item">Ahmed Samy</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



Ahmed Hassouna		Adel Sabour	
Osama Elzero	Mahmoud Fayed	Amr ELSayed	Mahmoud Elawa
Eslam Elnoby	Ahmed Baryyan	Ehab Mohamed	Ahmed Samy

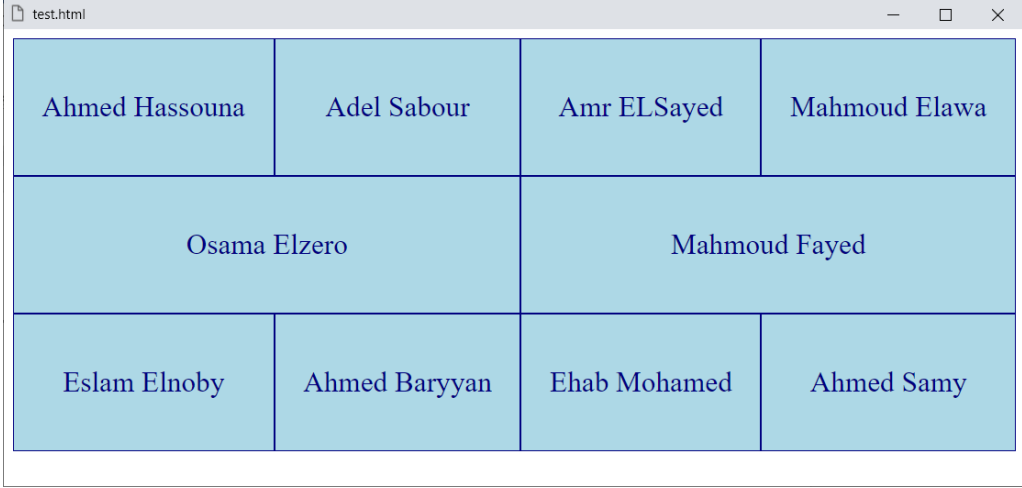
لا حظ هنا أنه تم تسمية عنصريه وهم item1 و item2، ثم تم وضع القيم 'item1 item1 item2 item2' داخل الخاصية grid-template-areas، فتم دمج خليتيه لـ item1، وتم دمج خليتيه لـ item2.

انتبه! جيداً لأن المثال التالي إن شاء الله، سيجعلك تبدأ في الفهم أكثر بفضل الله، حيث سوف يتم تسمية جميع العناصر items، ثم يتم وضع ترتيب ظهورهم في الـ grid.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
#i1{ grid-area:item1; } #i2{ grid-area:item2; }
#i3{ grid-area:item3; } #i4{ grid-area:item4; }
#i5{ grid-area:item5; } #i6{ grid-area:item6; }
div.grid{
display:grid; grid-template-rows:120px 120px 120px;
grid-template-columns:auto auto auto auto;
grid-template-areas: 'item1 item2 item5 item6'
                    'item3 item3 item4 item4';
}
div.grid>.item{
background:lightblue; color:navy;
padding:7px; font-size:25px; text-align:center;
border:1px solid navy; display:flex;
align-items:center; justify-content:center;
}
</style>
</head>
<body>
<div class="grid">
<div class="item" id="i1">Ahmed Hassouna</div>
<div class="item" id="i2">Adel Sabour</div>
<div class="item" id="i3">Osama Elzero</div>
<div class="item" id="i4">Mahmoud Fayed</div>
<div class="item" id="i5">Amr ELSayed</div>
<div class="item" id="i6">Mahmoud Elawa</div>
<div class="item" id="i7">Eslam Elnoby</div>
<div class="item" id="i8">Ahmed Baryyan</div>
<div class="item" id="i9">Ehab Mohamed</div>
<div class="item" id="i10">Ahmed Samy</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



Ahmed Hassouna	Adel Sabour	Amr ELSayed	Mahmoud Elawa
Osama Elzero		Mahmoud Fayed	
Eslam Elnoby	Ahmed Baryyan	Ehab Mohamed	Ahmed Samy

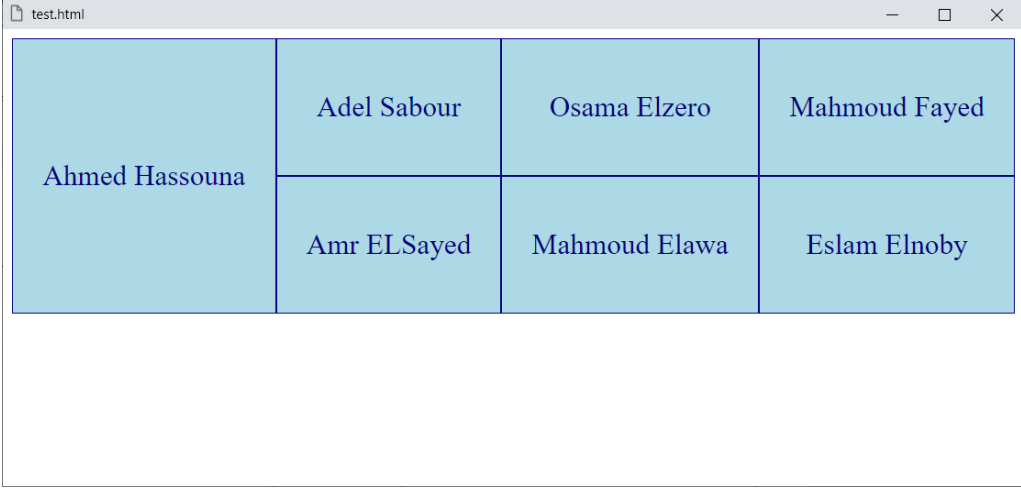
وهنا يظهر لك أنه مع المتاح أن تقسم ال grid كما تحب، حيث تكتب مجموعة عناصر بين سبجل كوت، ثم مسافة، ثم مجموعة عناصر أخرى بين سبجل كوت، وهكذا حتى تبني ال grid الخاص بك.

لاحظ أنه تم وضع أربع عناصر بالترتيب الذي نريده ليمثلوا أول صف، وهم item1 و item2 و item5 و item6، ثم جعلنا العنصر item3 يأخذ خليتيه، ثم جعلنا العنصر item4 يأخذ خليتيه، ثم الذي لم نكتبه ولم نسميه ب grid-area، سيتم ترتيبهم تلقائياً، ولكنه **انتبه!** جيداً، فإذا قمت بتسمية عنصر قم بكتابته داخل الترتيب.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
#i1{ grid-area:item1; } #i2{ grid-area:item2; }
#i3{ grid-area:item3; } #i4{ grid-area:item4; }
#i5{ grid-area:item5; } #i6{ grid-area:item6; }
#i7{ grid-area:item7; }
div.grid{
display:grid; grid-template-rows:120px 120px 120px;
grid-template-columns:auto auto auto auto;
grid-template-areas: 'item1 item2 item3 item4'
                    'item1 item5 item6 item7';
}
div.grid>.item{
background:lightblue; color:navy;
padding:7px; font-size:25px; text-align:center;
border:1px solid navy; display:flex;
align-items:center; justify-content:center;
}
</style>
</head>
<body>
<div class="grid">
<div class="item" id="i1">Ahmed Hassouna</div>
<div class="item" id="i2">Adel Sabour</div>
<div class="item" id="i3">Osama Elzero</div>
<div class="item" id="i4">Mahmoud Fayed</div>
<div class="item" id="i5">Amr ELSayed</div>
<div class="item" id="i6">Mahmoud Elawa</div>
<div class="item" id="i7">Eslam Elnoby</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



Ahmed Hassouna	Adel Sabour	Osama Elzero	Mahmoud Fayed
Ahmed Hassouna	Amr ELSayed	Mahmoud Elawa	Eslam Elnoby

لاحظ أنه تم كتابة العنصر item1 في السلسلة الأولى في البداية، ثم تم كتابته في السلسلة الثانية في البداية، فتم دمجهم طولياً.

ال قالب grid-template

هو مختصر لخواص القالب، فلخاصية grid-template-rows، ثم الرمز /، ثم الخاصية grid-template-columns، أو كتابة الخاصية grid-template-areas، فيتم عمل القالب. **كود!** قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.grid{
    display:grid;
    grid-template:120px 120px 120px / auto auto auto;
  }
  div.grid>.item{
    background:lightblue; color:navy;
    padding:7px; font-size:25px; text-align:center;
    border:1px solid navy; display:flex;
    align-items:center; justify-content:center;
  }
</style>
</head>
<body>
<div class="grid">
  <div class="item" id="i1">Ahmed Hassouna</div>
  <div class="item" id="i2">Adel Sabour</div>
  <div class="item" id="i3">Osama Elzero</div>
  <div class="item" id="i4">Mahmoud Fayed</div>
  <div class="item" id="i5">Amr ELSayed</div>
  <div class="item" id="i6">Mahmoud Elawa</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



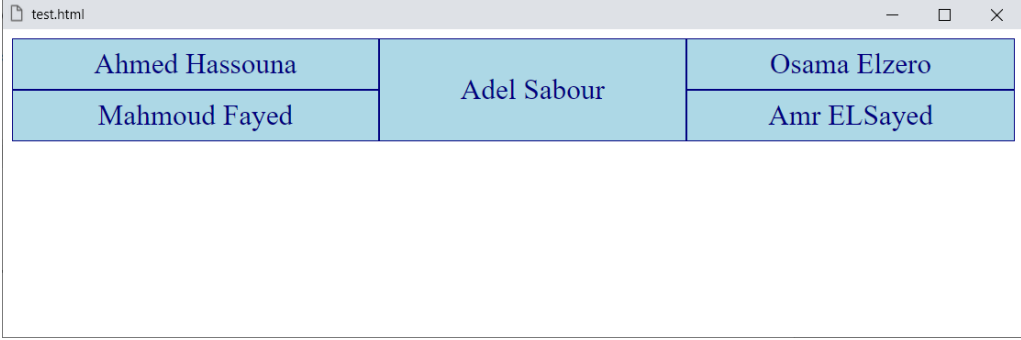
Ahmed Hassouna	Adel Sabour	Osama Elzero
Mahmoud Fayed	Amr ELSayed	Mahmoud Elawa

لاحظ أنه تم كتابة قالب الصفوف، ثم قالب الأعمدة في خاصية واحدة.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
#i1{ grid-area:item1; } #i2{ grid-area:item2; }
#i3{ grid-area:item3; } #i4{ grid-area:item4; }
#i5{ grid-area:item5; }
div.grid{
display:grid;
grid-template:'item1 item2 item3'
               'item4 item2 item5';
}
div.grid>.item{
background:lightblue; color:navy;
padding:7px; font-size:25px; text-align:center;
border:1px solid navy; display:flex;
align-items:center; justify-content:center;
}
</style>
</head>
<body>
<div class="grid">
<div class="item" id="i1">Ahmed Hassouna</div>
<div class="item" id="i2">Adel Sabour</div>
<div class="item" id="i3">Osama Elzero</div>
<div class="item" id="i4">Mahmoud Fayed</div>
<div class="item" id="i5">Amr ELSayed</div>
</div>
</body>
</html>
```

وبعد تشغفل الصفءة تظهر كما بالشكل التالف:



Ahmed Hassouna	Adel Sabour	Osama Elzero
Mahmoud Fayed		Amr ELSayed

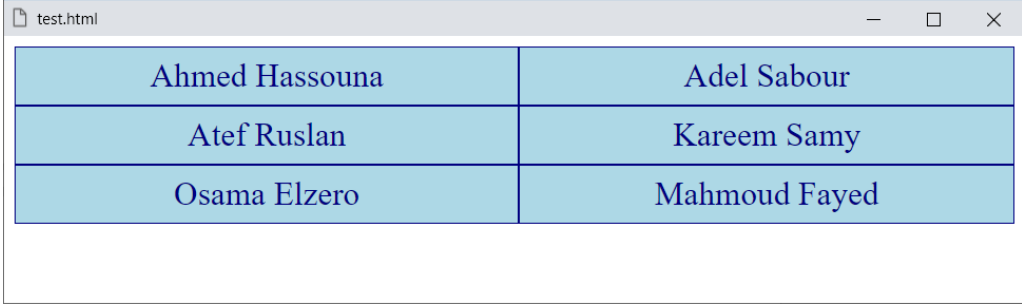
لاحظ أنه تم كتابة القفم الخاصة بقالب مناطق الخلايا في الخاصة
.grid-template، بدلاً من .grid-template-areas.

مناطق الخلايا grid-area

يمكنك تحديد أماكن الخلايا بالأرقام، بدلاً من العمل بالـ template،
وسنجد بـ بالكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
#i1{ grid-area: 1/1 } #i2{ grid-area: 1/2 }
#i3{ grid-area: 2/1 } #i4{ grid-area: 2/2 }
#i5{ grid-area: 3/1 } #i6{ grid-area: 3/2 }
div.grid{
display:grid;
}
div.grid>.item{
background:lightblue; color:navy;
padding:7px; font-size:25px; text-align:center;
border:1px solid navy; display:flex;
align-items:center; justify-content:center;
}
</style>
</head>
<body>
<div class="grid">
<div class="item" id="i1">Ahmed Hassouna</div>
<div class="item" id="i2">Adel Sabour</div>
<div class="item" id="i3">Atef Ruslan</div>
<div class="item" id="i4">Kareem Samy</div>
<div class="item" id="i5">Osama Elzero</div>
<div class="item" id="i6">Mahmoud Fayed</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



Ahmed Hassouna	Adel Sabour
Atef Ruslan	Kareem Samy
Osama Elzero	Mahmoud Fayed

لاحظ أننا لم نحدد للعنصر الرئيسي إلا `display:grid` فقط.

- لاحظ أنه تم تحديد العنصر i1 ليكون في الصف الأول والعمود الأول.
- لاحظ أنه تم تحديد العنصر i2 ليكون في الصف الأول والعمود الثاني.
- لاحظ أنه تم تحديد العنصر i3 ليكون في الصف الثاني والعمود الأول.
- لاحظ أنه تم تحديد العنصر i4 ليكون في الصف الثاني والعمود الثاني.
- لاحظ أنه تم تحديد العنصر i5 ليكون في الصف الثالث والعمود الأول.
- لاحظ أنه تم تحديد العنصر i6 ليكون في الصف الثالث والعمود الثاني.

فجوات بين الخلايا grid-gap

يمكن عمل فجوات عمودية، بالخاصية `grid-column-gap`،
 وسنجد بالكوود! التالي:

```
<!DOCTYPE html>
<html>
<head>
  <title>My Page</title>
  <style>
    div.grid{
      display:grid; background:#b11eb1; padding:10px;
      grid-template-columns:auto auto auto;
      grid-column-gap:10px;
    }
    div.grid>.item{
      background:rgba(255,255,255,0.7);
      padding:20px 0; font-size:25px; text-align:center;
    }
  </style>
</head>
<body>
  <div class="grid">
    <div class="item">Ahmed Hassouna</div>
    <div class="item">Adel Sabour</div>
    <div class="item">Osama Elzero</div>
    <div class="item">Mahmoud Fayed</div>
    <div class="item">Amr ELSayed</div>
    <div class="item">Mahmoud Elawa</div>
    <div class="item">Eslam Elnoby</div>
    <div class="item">Ahmed Baryyan</div>
    <div class="item">Ehab Mohamed</div>
  </div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن حشو ال grid بمقدار 10px، وكذلك مقدار الفجوات، لتتناسب، وتجانس، وتتناسق.

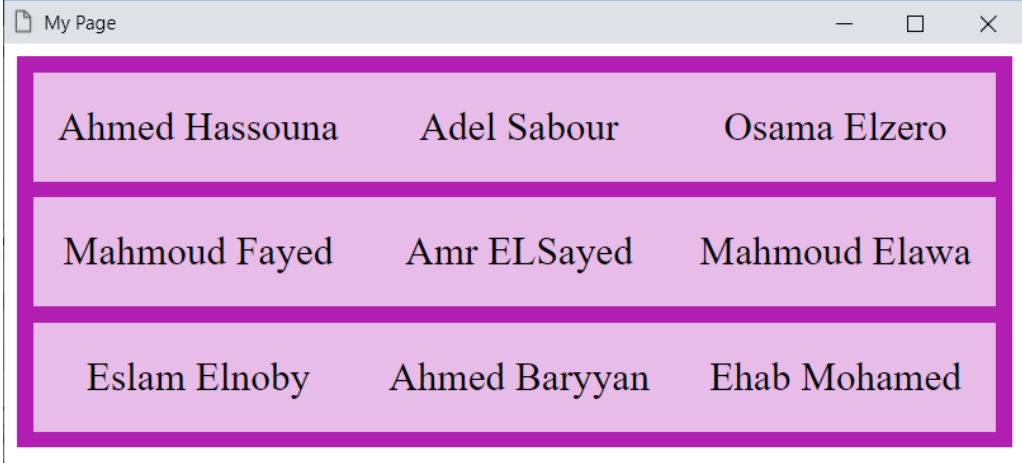
لاحظ أن لون خلفية ال items، أبيض بمقدار شفافية 0.7، وذلك ليكون نفس لون خلفية ال grid، ولكنه أفتح منه بقليل.

لاحظ أن حشو ال items، تم ضبطه من أعلي وأسفل فقط، حتى لا يضايق عرض النص بالخلية.

يمكنك عمل فجوات أفقية، بالخاصية `grid-row-gap`، وسنجد
بالكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
  div.grid{
    display:grid; background:#b11eb1; padding:10px;
    grid-template-columns:auto auto auto;
    grid-row-gap:10px;
  }
  div.grid>.item{
    background:rgba(255,255,255,0.7);
    padding:20px 0; font-size:25px; text-align:center;
  }
</style>
</head>
<body>
<div class="grid">
  <div class="item">Ahmed Hassouna</div>
  <div class="item">Adel Sabour</div>
  <div class="item">Osama Elzero</div>
  <div class="item">Mahmoud Fayed</div>
  <div class="item">Amr ELSayed</div>
  <div class="item">Mahmoud Elawa</div>
  <div class="item">Eslam Elnoby</div>
  <div class="item">Ahmed Baryyan</div>
  <div class="item">Ehab Mohamed</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



Ahmed Hassouna	Adel Sabour	Osama Elzero
Mahmoud Fayed	Amr ELSayed	Mahmoud Elawa
Eslam Elnoby	Ahmed Baryyan	Ehab Mohamed

لاحظ أنه تم استبدال الخاصية `grid-column-gap` بالخاصية `grid-row-gap`، فأصبحت الفجوات أو الفراغات أفقية بين الصفوف.

يمكنك عمل فجوات أفقية ورأسية، بالخاصية `grid-gap`، وسنجد
بالكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
  div.grid{
    display:grid; background:#b11eb1; padding:10px;
    grid-template-columns:auto auto auto;
    grid-gap:10px;
  }
  div.grid>.item{
    background:rgba(255,255,255,0.7);
    padding:20px 0; font-size:25px; text-align:center;
  }
</style>
</head>
<body>
<div class="grid">
  <div class="item">Ahmed Hassouna</div>
  <div class="item">Adel Sabour</div>
  <div class="item">Osama Elzero</div>
  <div class="item">Mahmoud Fayed</div>
  <div class="item">Amr ELSayed</div>
  <div class="item">Mahmoud Elawa</div>
  <div class="item">Eslam Elnoby</div>
  <div class="item">Ahmed Baryyan</div>
  <div class="item">Ehab Mohamed</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



Ahmed Hassouna	Adel Sabour	Osama Elzero
Mahmoud Fayed	Amr ELSayed	Mahmoud Elawa
Eslam Elnoby	Ahmed Baryyan	Ehab Mohamed

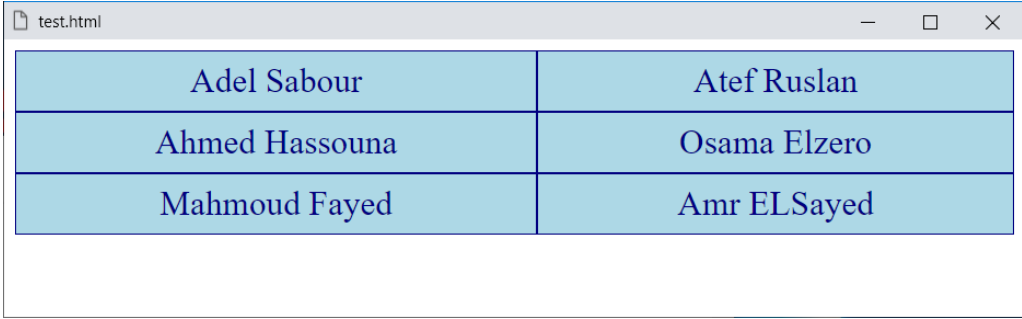
لاحظ أن الخاصية `grid-gap`، تعتبر اختصار `shorthand`، للخاصيتين `grid-column-gap` و `grid-row-gap`، في وقت واحد، لعمل فجوات بين الأعمدة، وكذلك فجوات بين الصفوف.

صف البداية grid-row-start

يمكنك أن نحدد لعنصر معين أن يبدأ من صف معين على حسب الرغبة، وذلك بالخاصية grid-row-start بإعطائها رقم الصف.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
#i1{ grid-row-start:2; }
div.grid{
display:grid;
grid-template-columns:auto auto;
}
div.grid>.item{
background:lightblue; color:navy;
padding:7px; font-size:25px; text-align:center;
border:1px solid navy; display:flex;
align-items:center; justify-content:center;
}
</style>
</head>
<body>
<div class="grid">
<div class="item" id="i1">Ahmed Hassouna</div>
<div class="item" id="i2">Adel Sabour</div>
<div class="item" id="i3">Atef Ruslan</div>
<div class="item" id="i4">Osama Elzero</div>
<div class="item" id="i5">Mahmoud Fayed</div>
<div class="item" id="i6">Amr ELSayed</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



Adel Sabour	Atef Ruslan
Ahmed Hassouna	Osama Elzero
Mahmoud Fayed	Amr ELSayed

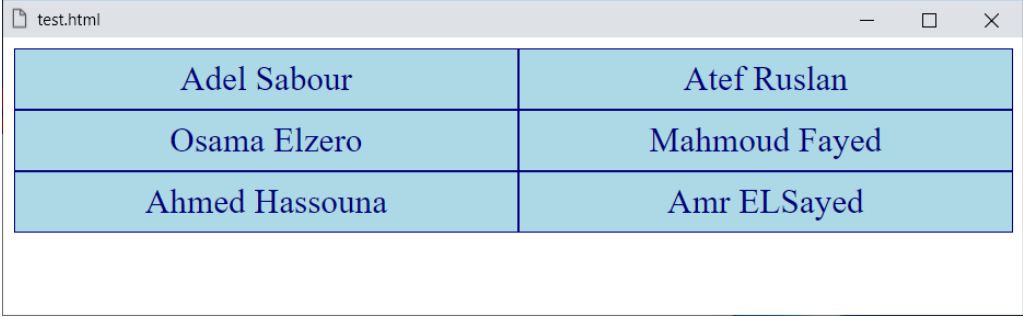
لاحظ أن العنصر i1 تم وضعه في الأول بطبيعته، ولكن بعد إعطاء القيمة 2 للخاصية grid-row-start، أصبح في أول الصف الثاني، حيث أخذت رقم الصف الذي سوف تبدأ فيه، وإذا لم تخصص قيمة لهذه الخاصية، فتأخذ قيمة افتراضية بـ auto.

صف النهاية grid-row-end

يمكّنك أن نحدد لعنصر معين أن ينتهي قبل صف معين على حسب الرغبة، وذلك بالخاصية grid-row-end بإعطائها رقم الصف الذي ينتهي قبله، وسنجدرب **بالكود!** التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
  #i1{ grid-row-end: 4; }
  div.grid{
    display:grid;
    grid-template-columns:auto auto;
  }
  div.grid>.item{
    background:lightblue; color:navy;
    padding:7px; font-size:25px; text-align:center;
    border:1px solid navy; display:flex;
    align-items:center; justify-content:center;
  }
</style>
</head>
<body>
<div class="grid">
  <div class="item" id="i1">Ahmed Hassouna</div>
  <div class="item" id="i2">Adel Sabour</div>
  <div class="item" id="i3">Atef Ruslan</div>
  <div class="item" id="i4">Osama Elzero</div>
  <div class="item" id="i5">Mahmoud Fayed</div>
  <div class="item" id="i6">Amr ELSayed</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



Adel Sabour	Atef Ruslan
Osama Elzero	Mahmoud Fayed
Ahmed Hassouna	Amr ELayed

وهنا تم إعطاء القيمة 4 للخاصية `grid-row-end`، لينتهي العنصر المقصود الى الصف الرابع، أي ما قبل الصف الرابع، وإذا لم تخصص قيمة لهذه الخاصية، فتأخذ قيمة افتراضية بـ `auto`.

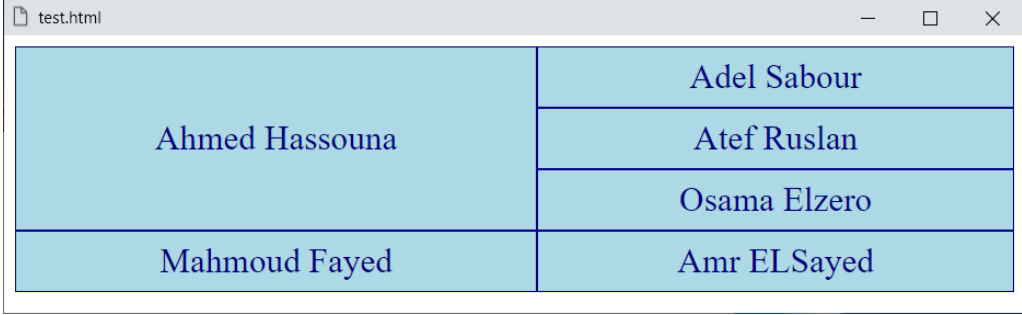
انتبه! أنه إذا أعطيت الخاصية القيمة مثلاً 2 سوف يكون محلها في الصف الأول، وإذا أعطيتها القيمة 3 سوف يكون محلها في الصف الثاني، وفي المثال السابق لما أعطيناها القيمة 4 كان محلها في الصف الثالث، وهكذا حتى تفعل ما ترغب به بفضل الله.

انتبه! فإنه يمكنك عمل تمدد `span` للخلايا بهذه الخاصية الرائعة، وذلك عن طريق إعطاؤها قيمة `span`، ثم مسافة، ثم رقم للتمد، مثلاً 2 لتتمدد في خليتين، أو 3 لتتمدد في ثلاث خلايا، وهكذا، وسنجرّب في المثال التالي إن شاء الله.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
#i1{ grid-row-end:span 3; }
div.grid{
display:grid;
grid-template-columns:auto auto;
}
div.grid>.item{
background:lightblue; color:navy;
padding:7px; font-size:25px; text-align:center;
border:1px solid navy; display:flex;
align-items:center; justify-content:center;
}
</style>
</head>
<body>
<div class="grid">
<div class="item" id="i1">Ahmed Hassouna</div>
<div class="item" id="i2">Adel Sabour</div>
<div class="item" id="i3">Atef Ruslan</div>
<div class="item" id="i4">Osama Elzero</div>
<div class="item" id="i5">Mahmoud Fayed</div>
<div class="item" id="i6">Amr ELSayed</div>
</div>
</body>
</html>
```

وبءء ءشءعل الصءفة ءظهر كما بالشكل ءءالؑ



Ahmed Hassouna	Adel Sabour
Ahmed Hassouna	Atef Ruslan
Ahmed Hassouna	Osama Elzero
Mahmoud Fayed	Amr ELSayed

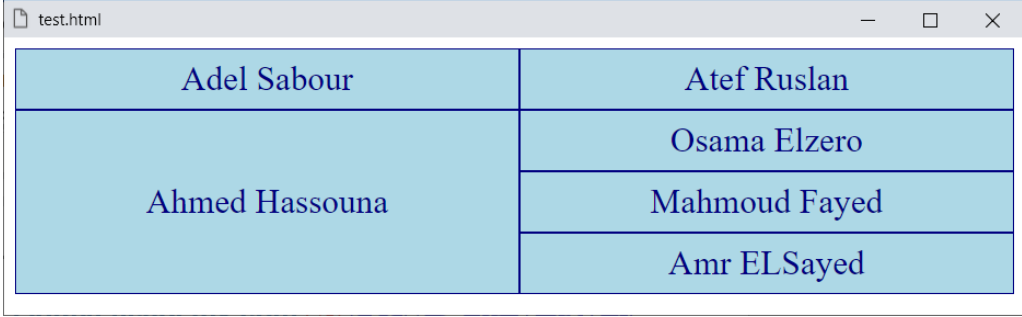
لاءظ أن العنصر i1 ءمءء فف ءلائ ءلافا؁ لأنه آءء القفمة 3 span؁ فءل مءل ءلائ ءلافا رأسفة ءاملة؁ أو فمءءنا القول إنه ءمء ءلائ صءفوف.

الصف grid-row

الخاصية grid-row، مختصر للخاصية grid-row-start و grid-row-end، وسنجرّب **بالكود!** التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
#i1{ grid-row:2 / span 3; }
div.grid{
display:grid;
grid-template-columns:auto auto;
}
div.grid>.item{
background:lightblue; color:navy;
padding:7px; font-size:25px; text-align:center;
border:1px solid navy; display:flex;
align-items:center; justify-content:center;
}
</style>
</head>
<body>
<div class="grid">
<div class="item" id="i1">Ahmed Hassouna</div>
<div class="item" id="i2">Adel Sabour</div>
<div class="item" id="i3">Atef Ruslan</div>
<div class="item" id="i4">Osama Elzero</div>
<div class="item" id="i5">Mahmoud Fayed</div>
<div class="item" id="i6">Amr ELSayed</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



Adel Sabour	Atef Ruslan
Ahmed Hassouna	Osama Elzero
	Mahmoud Fayed
	Amr ELSayed

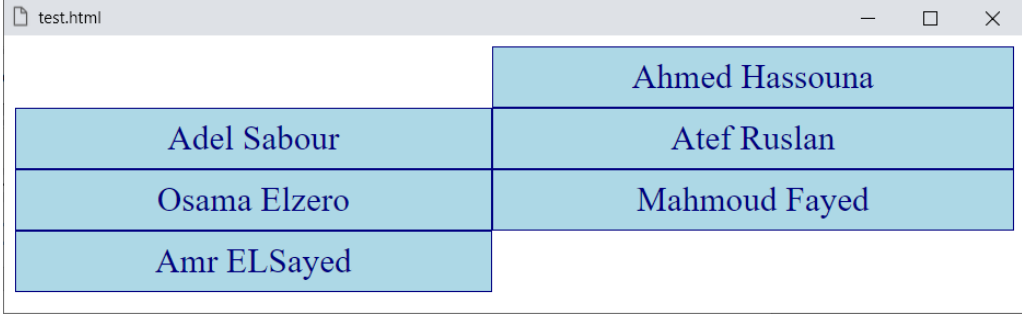
لاحظ أن الخاصية `grid-row`، أخذت قيمتيه وبينهم فاصل /، حيث كانت القيمة النهائية هي `3 / span 2`، ومعني ذلك، وكأننا نقول لهذا العنصر، ابدأ من الصف الثاني، وقم بدمج ثلاث خلايا.

عمود البداية grid-column-start

يملك تحديد رقم العمود الذي ستبدأ الخلية منه.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
#i1{ grid-column-start:2; }
div.grid{
display:grid;
grid-template-columns:auto auto;
}
div.grid>.item{
background:lightblue; color:navy;
padding:7px; font-size:25px; text-align:center;
border:1px solid navy; display:flex;
align-items:center; justify-content:center;
}
</style>
</head>
<body>
<div class="grid">
<div class="item" id="i1">Ahmed Hassouna</div>
<div class="item" id="i2">Adel Sabour</div>
<div class="item" id="i3">Atef Ruslan</div>
<div class="item" id="i4">Osama Elzero</div>
<div class="item" id="i5">Mahmoud Fayed</div>
<div class="item" id="i6">Amr ELSayed</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



The screenshot shows a browser window titled 'test.html' displaying a table with the following content:

	Ahmed Hassouna
Adel Sabour	Atef Ruslan
Osama Elzero	Mahmoud Fayed
Amr ELSayed	

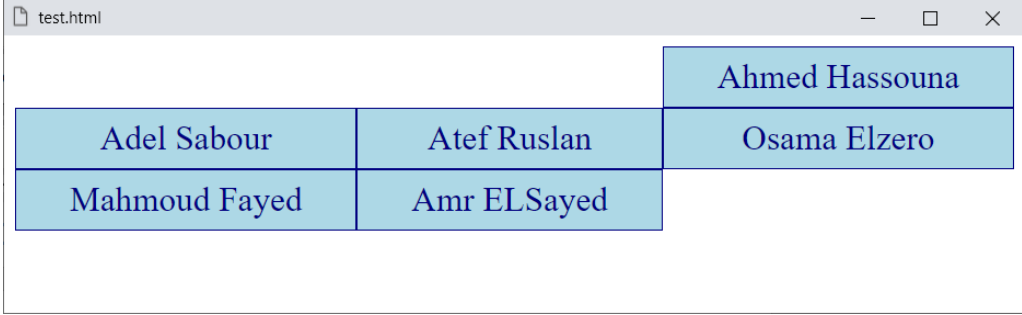
لاحظ أن العنصر المقصود بدأ من العمود الثاني.

عمود النهاية grid-column-end

يمكنك تحديد رقم العمود الذي يكون مكان العنصر قبله.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
#i1{ grid-column-end:4; }
div.grid{
display:grid;
grid-template-columns:auto auto auto;
}
div.grid>.item{
background:lightblue; color:navy;
padding:7px; font-size:25px; text-align:center;
border:1px solid navy; display:flex;
align-items:center; justify-content:center;
}
</style>
</head>
<body>
<div class="grid">
<div class="item" id="i1">Ahmed Hassouna</div>
<div class="item" id="i2">Adel Sabour</div>
<div class="item" id="i3">Atef Ruslan</div>
<div class="item" id="i4">Osama Elzero</div>
<div class="item" id="i5">Mahmoud Fayed</div>
<div class="item" id="i6">Amr ELSayed</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



		Ahmed Hassouna
Adel Sabour	Atef Ruslan	Osama Elzero
Mahmoud Fayed	Amr ELSayed	

لاحظ أن نهاية العنصر الى العمود 4، أي مكان هذا العنصر ما قبل العمود 4، فأصبح محلة أو مكانة في العمود الثالث.

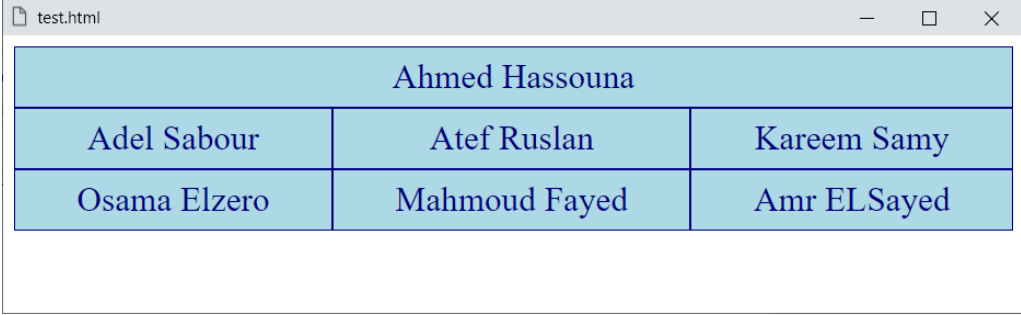
انتبه! أنه إذا أعطيت الخاصية القيمة مثلاً 2 سوف يكون محلها في العمود الأول، وإذا أعطيتها القيمة 3 سوف يكون محلها في العمود الثاني، وفي المثال السابق لما أعطيتها القيمة 4 كان محلها في العمود الثالث، وهكذا حتى تفعل ما ترغب به بفضل الله.

انتبه! فإنه يمكنك عمل تمدد span للخلايا بهذه الخاصية الرائعة، وذلك عن طريق إعطاؤها قيمة span، ثم مسافة، ثم رقم للتمد، مثلاً 2 لتتمدد في خليتيه، أو 3 لتتمدد في ثلاث خلايا، وهكذا، وسنجرّب في المثال التالي إن شاء الله.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
#i1{ grid-column-end:span 3; }
div.grid{
display:grid;
grid-template-columns:auto auto auto;
}
div.grid>.item{
background:lightblue; color:navy;
padding:7px; font-size:25px; text-align:center;
border:1px solid navy; display:flex;
align-items:center; justify-content:center;
}
</style>
</head>
<body>
<div class="grid">
<div class="item" id="i1">Ahmed Hassouna</div>
<div class="item" id="i2">Adel Sabour</div>
<div class="item" id="i3">Atef Ruslan</div>
<div class="item" id="i4">Kareem Samy</div>
<div class="item" id="i5">Osama Elzero</div>
<div class="item" id="i6">Mahmoud Fayed</div>
<div class="item" id="i7">Amr ELSayed</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



Ahmed Hassouna		
Adel Sabour	Atef Ruslan	Kareem Samy
Osama Elzero	Mahmoud Fayed	Amr ELSayed

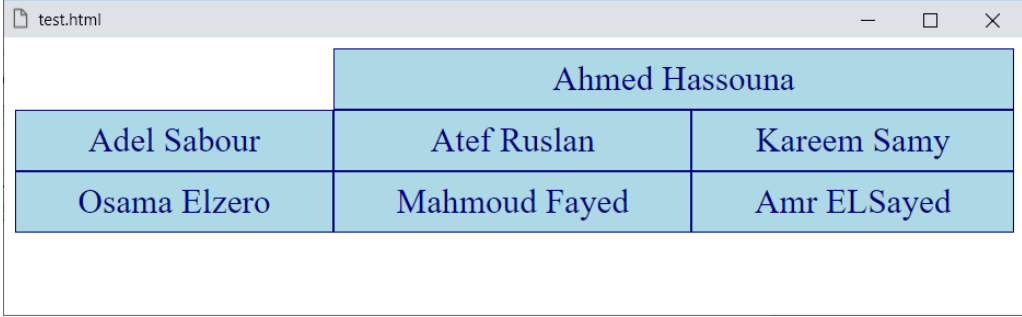
لاحظ أن العنصر `i1` تمتد في ثلاث خلايا، لأنه أخذ القيمة `span 3`،
فحل محل ثلاث خلايا أفقية كاملة، أو يمكننا القول إنه دمج ثلاث أعمدة.

العمود grid-column

خاصية العمود grid-column، تعتبر مختصر سريع للخاصية
grid-column-start و grid-column-end، وسنجد
بالتالي!

```
<!DOCTYPE html>
<html>
<head>
<style>
#i1{ grid-column:2 / span 2; }
div.grid{
display:grid;
grid-template-columns:auto auto auto;
}
div.grid>.item{
background:lightblue; color:navy;
padding:7px; font-size:25px; text-align:center;
border:1px solid navy; display:flex;
align-items:center; justify-content:center;
}
</style>
</head>
<body>
<div class="grid">
<div class="item" id="i1">Ahmed Hassouna</div>
<div class="item" id="i2">Adel Sabour</div>
<div class="item" id="i3">Atef Ruslan</div>
<div class="item" id="i4">Kareem Samy</div>
<div class="item" id="i5">Osama Elzero</div>
<div class="item" id="i6">Mahmoud Fayed</div>
<div class="item" id="i7">Amr ELSayed</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



Ahmed Hassouna		
Adel Sabour	Atef Ruslan	Kareem Samy
Osama Elzero	Mahmoud Fayed	Amr ELSayed

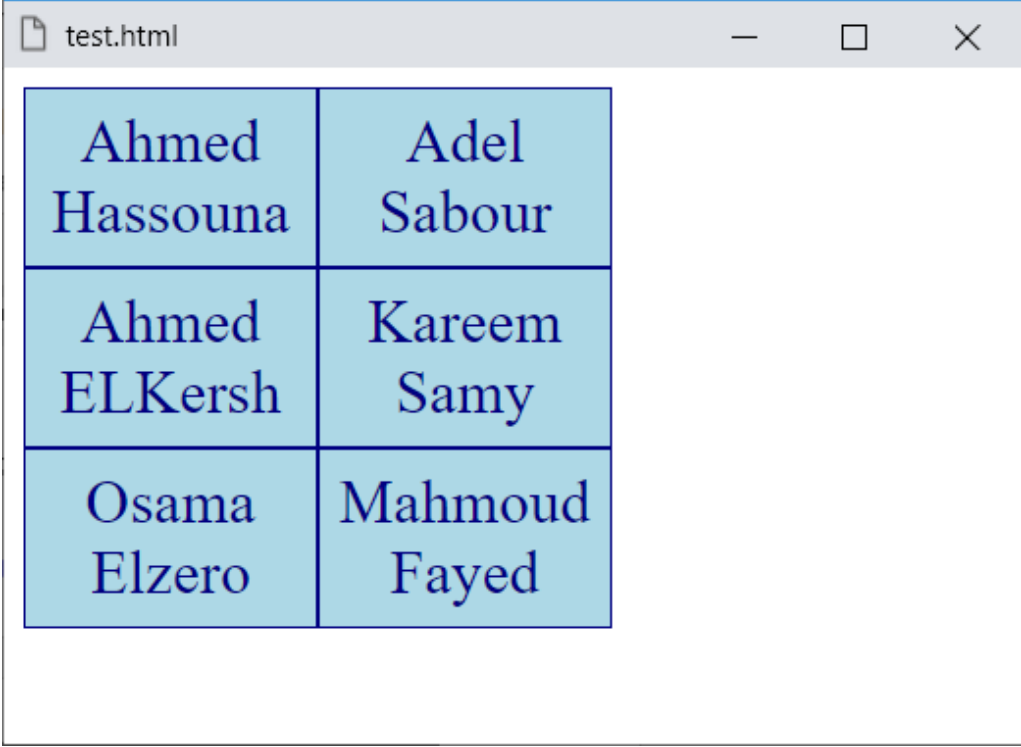
لاحظ أن الخاصية `grid-column`، أخذت قيمتيه وبينهم فاصلا /، حيث كانت القيمة النهائية هي `2 / span 2`، ومعني ذلك، وكأننا نقول لهذا العنصر، ابدأ من العمود الثاني، وقم بدمج خليتيه.

تلقائية العمود `grid-auto-columns`

يمكنك تقسيم المناطق عن طريق الخاصية `grid-area`، ثم إعطاء قيمة للخاصية `grid-auto-columns` لتكون هذه القيمة لعرض كل خلية، وسنجد إن شاء الله **بالكود!** التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
#i1{ grid-area: 1/1 } #i2{ grid-area: 1/2 }
#i3{ grid-area: 2/1 } #i4{ grid-area: 2/2 }
#i5{ grid-area: 3/1 } #i6{ grid-area: 3/2 }
div.grid{
display:grid;
grid-auto-columns:120px;
}
div.grid>.item{
background:lightblue; color:navy;
padding:7px; font-size:25px; text-align:center;
border:1px solid navy; display:flex;
align-items:center; justify-content:center;
}
</style>
</head>
<body>
<div class="grid">
<div class="item" id="i1">Ahmed Hassouna</div>
<div class="item" id="i2">Adel Sabour</div>
<div class="item" id="i3">Ahmed ELKersh</div>
<div class="item" id="i4">Kareem Samy</div>
<div class="item" id="i5">Osama Elzero</div>
<div class="item" id="i6">Mahmoud Fayed</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



Ahmed Hassouna	Adel Sabour
Ahmed ELKersh	Kareem Samy
Osama Elzero	Mahmoud Fayed

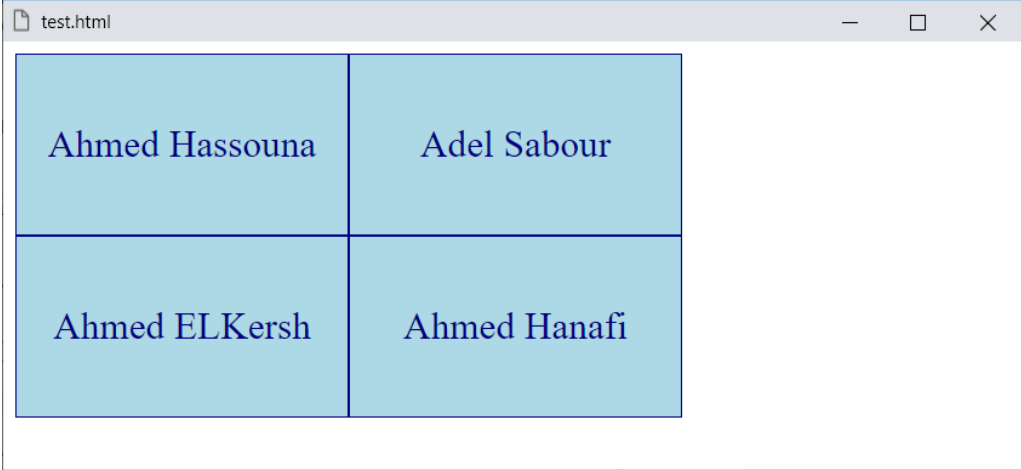
لاحظ أن عرض كل خلية أصبح 120px، ولكنه لا تنسى أنه تم تقسيم المناطق عن طريق الخاصية `grid-area`، ثم إعطاء قيمة 120px للخاصية `grid-auto-columns`.

تلقائية الصف grid-auto-rows

يمكنك ضبط تلقائية الصف مع طريق الخاصة grid-auto-rows.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
#i1{ grid-area: 1/1 } #i2{ grid-area: 1/2 }
#i3{ grid-area: 2/1 } #i4{ grid-area: 2/2 }
div.grid{
display:grid;
grid-auto-columns:220px;
grid-auto-rows:120px;
}
div.grid>.item{
background:lightblue; color:navy;
padding:7px; font-size:25px; text-align:center;
border:1px solid navy; display:flex;
align-items:center; justify-content:center;
}
</style>
</head>
<body>
<div class="grid">
<div class="item" id="i1">Ahmed Hassouna</div>
<div class="item" id="i2">Adel Sabour</div>
<div class="item" id="i3">Ahmed ELKersh</div>
<div class="item" id="i4">Ahmed Hanafi</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



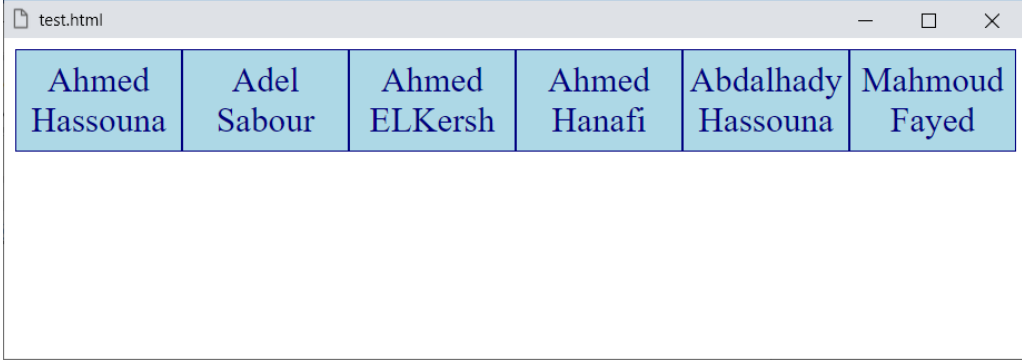
Ahmed Hassouna	Adel Sabour
Ahmed ELKersh	Ahmed Hanafi

لاحظ أنه الخاصية `grid-auto-rows` تم تخصيص القيمة لها، فأصبحت طول كل صف بمقدار `120px`.

تلقائية التدفق `grid-auto-flow`يمكّن التحكم في تدفق العناصر، وسنجد **باللورد!** التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div.grid{
    display:grid;
    grid-auto-flow:column;
    grid-auto-columns:120px;
  }
  div.grid>.item{
    background:lightblue; color:navy;
    padding:7px; font-size:25px; text-align:center;
    border:1px solid navy; display:flex;
    align-items:center; justify-content:center;
  }
</style>
</head>
<body>
<div class="grid">
  <div class="item" id="i1">Ahmed Hassouna</div>
  <div class="item" id="i2">Adel Sabour</div>
  <div class="item" id="i3">Ahmed ELKersh</div>
  <div class="item" id="i4">Ahmed Hanafi</div>
  <div class="item" id="i5">Abdalhady Hassouna</div>
  <div class="item" id="i6">Mahmoud Fayed</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



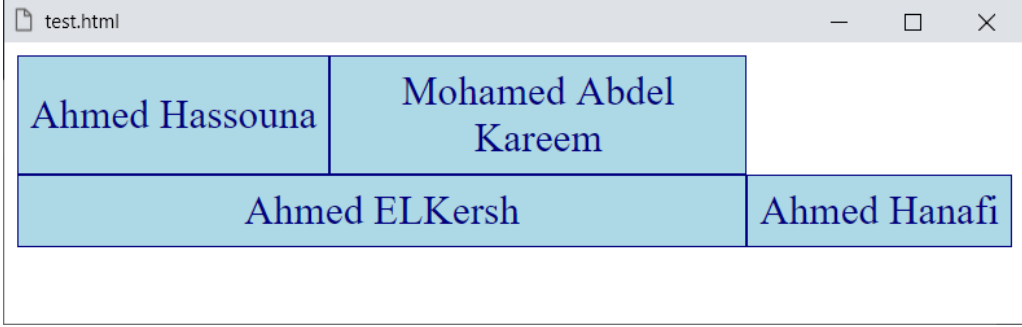
Ahmed Hassouna	Adel Sabour	Ahmed ELKersh	Ahmed Hanafi	Abdalhady Hassouna	Mahmoud Fayed
-------------------	----------------	------------------	-----------------	-----------------------	------------------

الخاصية `grid-auto-flow` تأخذ قيمة افتراضية بـ `row`، أي ان التدفق في الأساس يكون بترتيب صفوف، ولكنه عندما أعطيناها القيمة `column`، تم الترتيب كأعمدة، كما هو ظاهر في الشكل السابق.

يملكه وعند قيمة لعمل تدفق بكثافة بالقيمة dense، ولفهم ذلك سوف
نقوم بعمل المثال التالي أولاً بدون كثافة:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
#i3{ grid-column:auto / span 2; }
div.grid{
display:grid;
grid-template-columns:auto auto auto;
}
div.grid>.item{
background:lightblue; color:navy;
padding:7px; font-size:25px; text-align:center;
border:1px solid navy; display:flex;
align-items:center; justify-content:center;
}
</style>
</head>
<body>
<div class="grid">
<div class="item" id="i1">Ahmed Hassouna</div>
<div class="item" id="i2">Mohamed Abdel Kareem</div>
<div class="item" id="i3">Ahmed ELKersh</div>
<div class="item" id="i4">Ahmed Hanafi</div>
</div>
</body>
</html>
```

وءءء ءءءءء الصفءة ءظهء كما بالءءك الءالء:

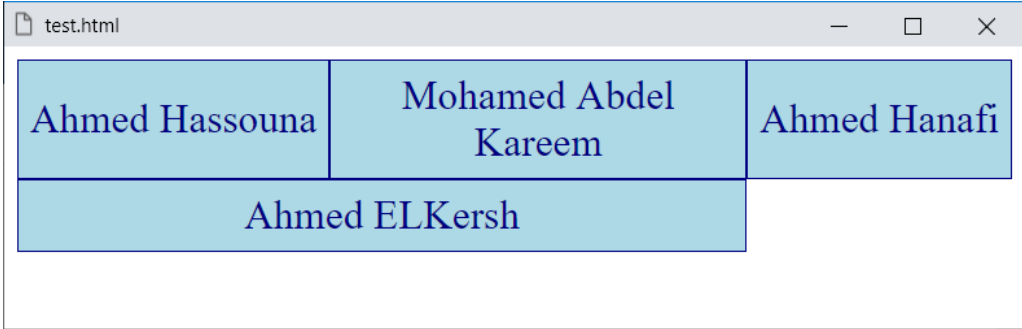


Ahmed Hassouna	Mohamed Abdel Kareem
Ahmed ELKersh	Ahmed Hanafi

لاءء أن الخلاء المءءوب ففاء "Ahmed Hanafi" ءمء مءاذاءها بالأسفل ءلقائباء مع الخلاء الءء ءمءءها، والآن اءءب الكوء الءالء:

grid-auto-flow:dense;

ءزفاءة لءءسفق ال `div.grid`، وسوف فظهء كالءءك الءالء:



Ahmed Hassouna	Mohamed Abdel Kareem	Ahmed Hanafi
Ahmed ELKersh		

فتة تجيم العناصر resize

هل تذكر عندما قمت بإنشاء الوسم textarea على صفحتك؟، إذا لم تذكر راجع كتاب صخرة الأساس في HTML، وإن تذكرت، فتذكر أنه هناك جزء يتم التجيم منه، كما يوضح الشكل التالي:



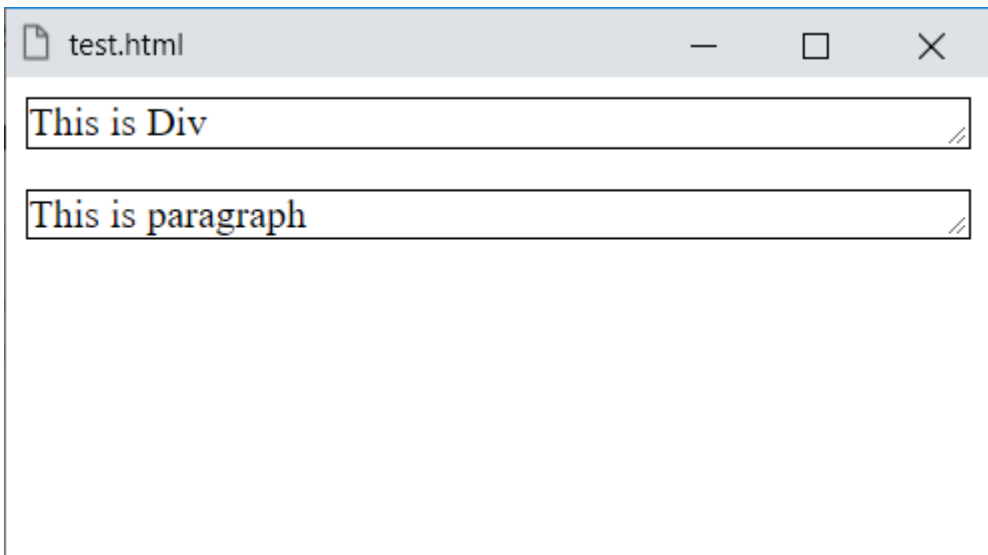
هذا هو المقبض المقصود، الذي يتم التجيم منه

وعد طرقت الخاصية resize نستطيع فعل هذا الشيء مع باقي العناصر الأخرى مثل الوسم div والوسم p، وإن شاء الله سوف نجرب في المثال التالي بفضل الله.

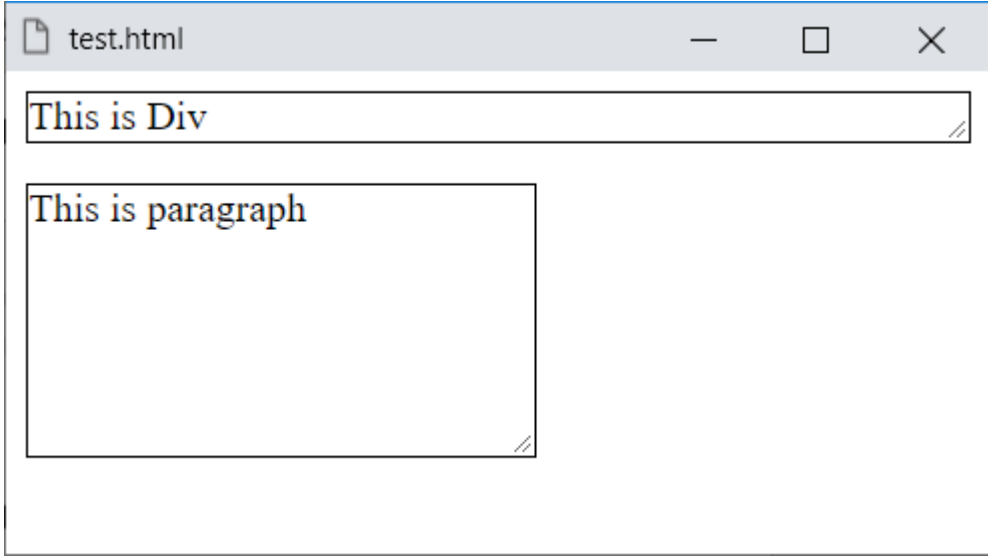
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    div , p{
      border:1px solid black;
      overflow:auto;
      resize: both;
    }
  </style>
</head>
<body>
  <div>This is Div</div>
  <p>This is paragraph</p>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ في الشكل السابق، وجود نفس مقبض التحكم الذي كان موجوداً في الـ textarea، والآن **انتبه!** لأننا سوف نذهب بالماوس الى هذا المقبض ثم نغير حجم الـ p بشكل عشوائي، ليكون كالشكل التالي:



وهكذا مع الـ div، ولاحظ أيضاً ضبط الخاصية overflow على القيمة auto، لتناسب مع التحجيم، ولاحظ أيضاً أن الخاصية resize تم ضبطها على القيمة both، أي يتغير رأسياً وأفقياً، كما يملك إعطاؤها القيمة horizontal لتكون أفقية فقط، ويملك إعطاؤها القيمة vertical لتكون رأسية فقط، أو القيمة none ليتم إلغاؤها، وnone هي الافتراضية، وهنا **انتبه!** جيداً، لأن هناك الاستفادة كبيرة، وهي أنه مع المملك إلغاء تحجيم الـ textarea، وإخراجه عن طبيعته، وذلك ب ضبط القيمة none لخاصية resize له.

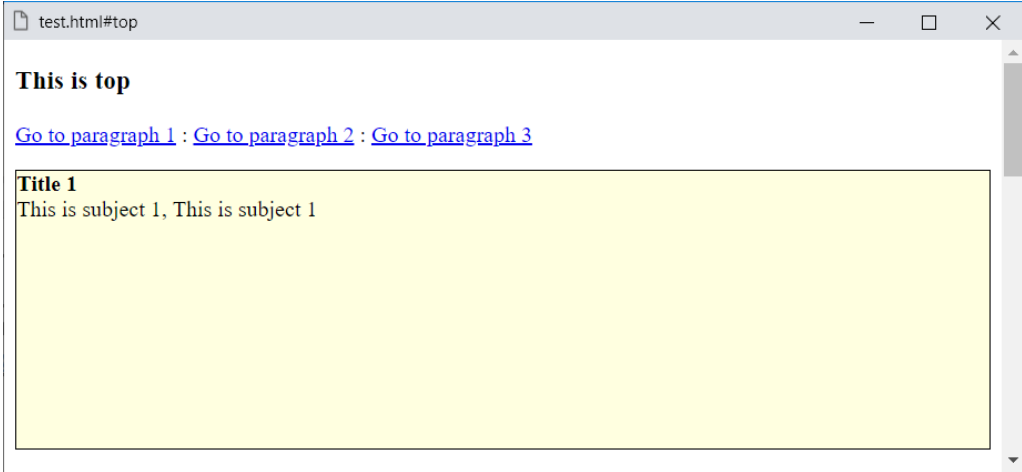
أسلوب التمرير scroll-behavior

والآن لفهم عمل هذه الخاصية، سوف نكتب مثال تم استخدامه في كتاب صخرة الأساس HTML، من قبل في صفحة رقم 118، كما يلي:

كود! تم بكتابة الكود التالي تم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  p{
    border:1px solid black; height:200px;
    background:lightyellow;
  }
</style>
</head>
<body>
<h3 id="top">This is top</h3>
<a href="#p1">Go to paragraph 1</a> :
<a href="#p2">Go to paragraph 2</a> :
<a href="#p3">Go to paragraph 3</a>
<p id="p1"><b>Title 1</b><br>
This is subject 1, This is subject 1</p>
<a href="#top">Go to top</a><hr>
<p id="p2"><b>Title 2</b><br>
This is subject 2, This is subject 2</p>
<a href="#top">Go to top</a><hr>
<p id="p3"><b>Title 3</b><br>
This is subject 3, This is subject 3</p>
<a href="#top">Go to top</a><hr>
<br><br><br><br><br><br><br><br><br>
</body>
</html>
```


وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



والآن مع الطبيعي عند الضغط على أي رابط، أنه ينزل فجأة على المكان المقصود في الصفحة، والآن قم بزيادة الكود التالي في الـ `style`:

```
html{ scroll-behavior: smooth; }
```

والآن احفظ واضغط مثلاً على الرابط "Go to paragraph 3" وسوف ترى التغيير في أسلوب التحريك وهو بالبطيء، وكأنك تمرره بنفسك.

الءءول والمنظور Transform, Perspective

ءاءل اللغة CSS الءمفة ءوءء ءاصفة باسم transform، ءملكه مصمم ومطور الوءب مه اءراء الءءولات والءءفرات الكئفة بكك سهولة، للوصول الء ءصمم أءءء ءازفة، أو ءصمم به ءفرات كئفة لو أءببء أن ءفعلاها بءوه اللغة CSS سوف ءسءرق وءء كئء وءءء كئء وءلكفة وءءم صور مهءر للإءءءء، مءك أن ءقلب عنصر مه أعلى إلى أسفل، أو مءك أن ءم عمفة شه للعنصر و كأنه مطاطف، أو مءك أن ءم عمفة اسءارة، وءءء ذلك مه الءءولات المءمفة للعناصر.

فءقسم الءءول transform الء قسمفه، وهما كما فلف:

- ءنائف الأبعاء 2D
فم الءءفر ففها بأبعاء ءنائفة عاءفة على صفءة الوءب، سواء كان ءءرفك أو اسءارة أو ءءء ذلك.
- ءلائف الأبعاء 3D
هف نفس عملفاء الـ 2D، ولكه فم الءءفر ففها بأبعاء ءلائفة وءلكه أءءء، وفءطلب فهم المنظور أولاً لاسءءارمه وءفعفله.

التحول ثنائي الأبعاد 2D

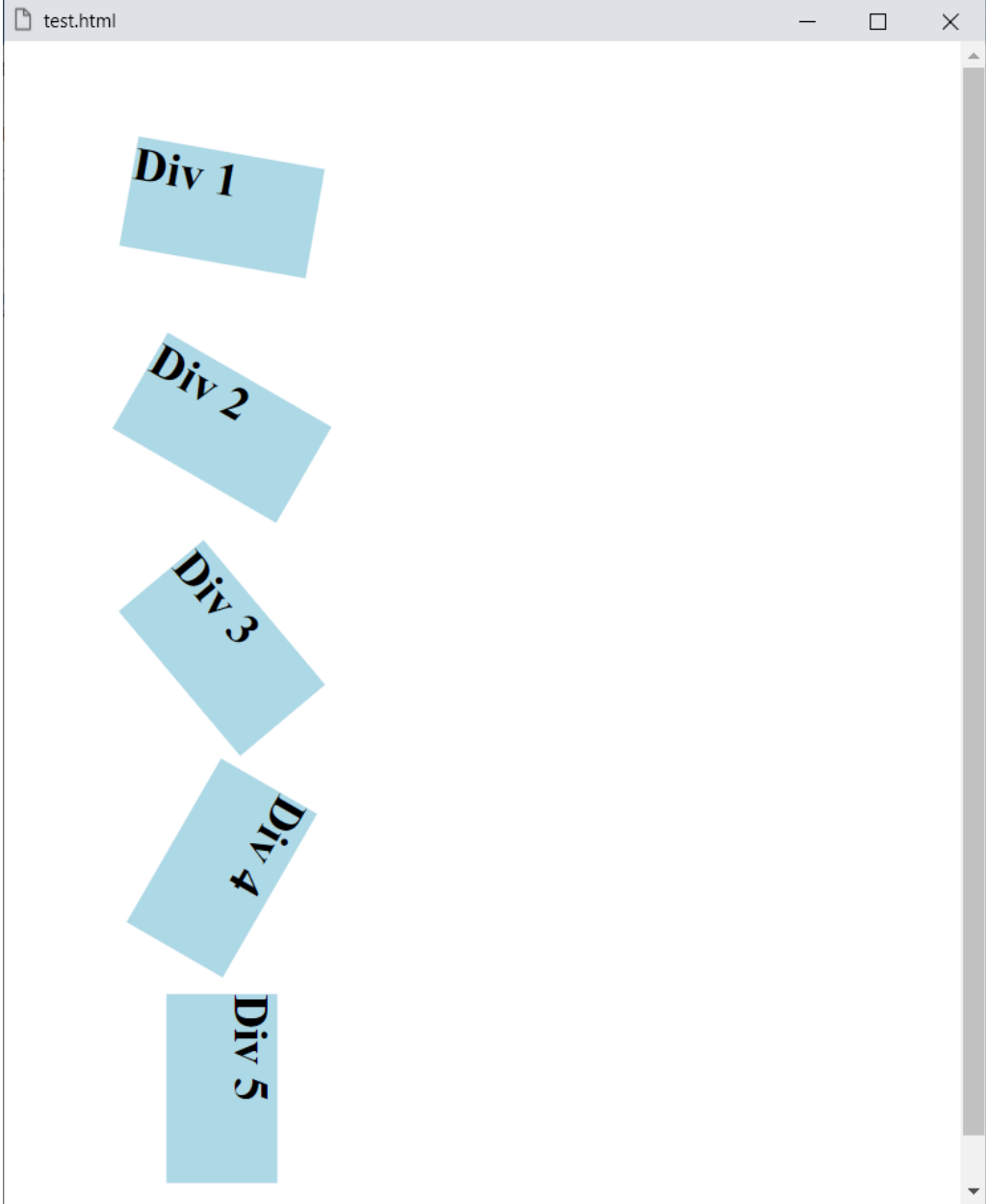
في هذا القسم يتم التغيير بشكل طبيعي، مثل الدوران والتحرك وغير ذلك.

تحول الاستدارة rotate

الخاصية transform تأخذ قيمة في الدالة rotate، وهذه القيمة هي عبارة عن الدرجة التي يتم تدوير العنصر على أساسها، مثلًا 120deg، أو 90deg، وسنجد **بالكود!** التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{
    background:lightblue; width:120px; height:70px;
    margin:70px; font-size:30px; font-weight:bold;
  }
  #d1{ transform: rotate(10deg) }
  #d2{ transform: rotate(30deg) }
  #d3{ transform: rotate(50deg) }
  #d4{ transform: rotate(120deg) }
  #d5{ transform: rotate(90deg) }
</style>
</head>
<body>
<div id="d1">Div 1</div>
<div id="d2">Div 2</div>
<div id="d3">Div 3</div>
<div id="d4">Div 4</div>
<div id="d5">Div 5</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



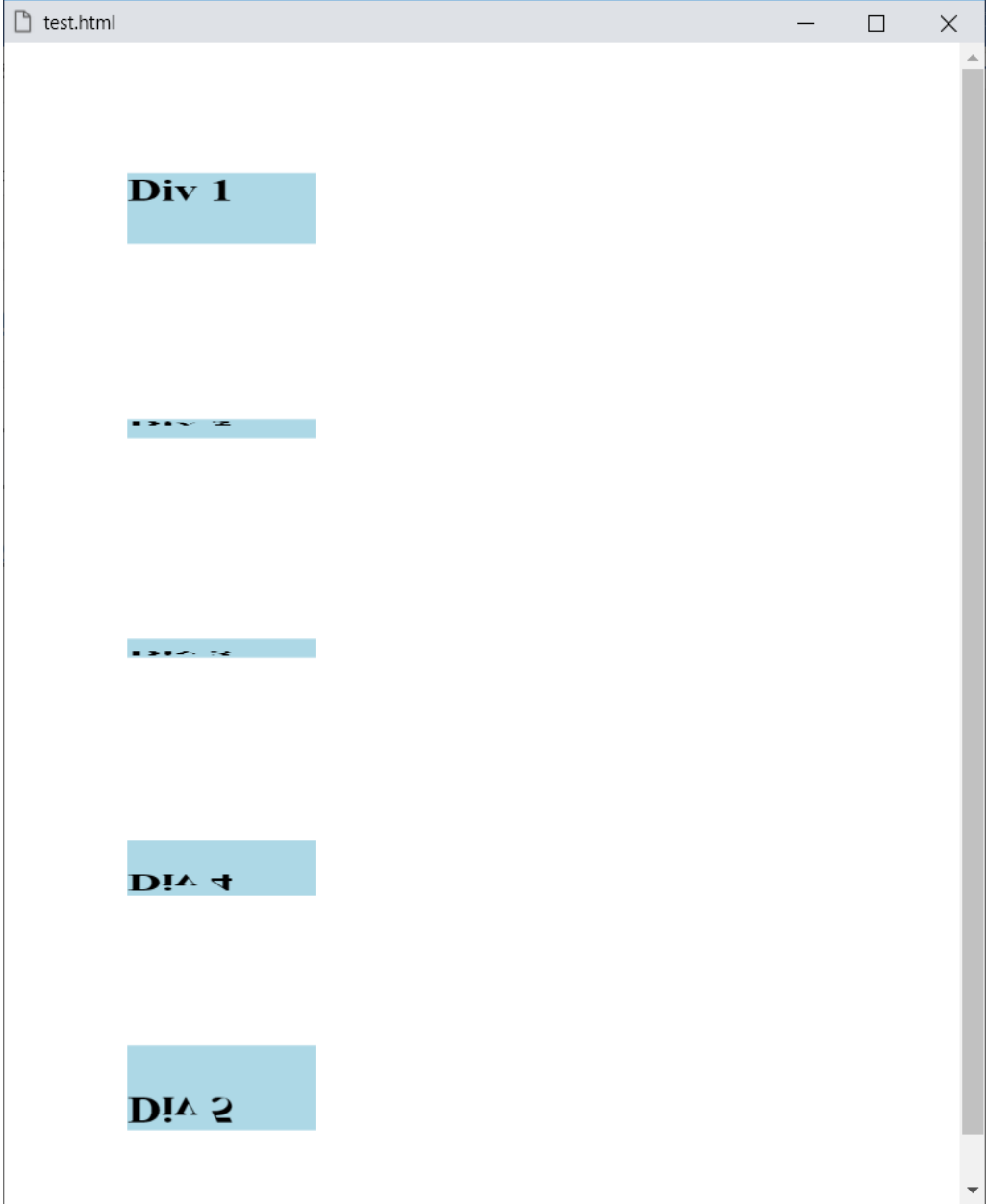
لاحظ أنك تشكل زاوية الاستدارة كما تريد بفضل الله.

استدارة أفقية rotateX

تخيل معي وأن العنصر ينكسح من أعلي وأسفل معاً، حتى ينقلب أفقياً،
 وسنعمل هذا بالدالة rotateX، وسوف نبيه ذلك **بالكود!** التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{
    background:lightblue; width:120px; height:70px;
    margin:70px; font-size:30px; font-weight:bold;
  }
  #d1{ transform: rotateX(50deg) }
  #d2{ transform: rotateX(80deg) }
  #d3{ transform: rotateX(100deg) }
  #d4{ transform: rotateX(120deg) }
  #d5{ transform: rotateX(140deg) }
</style>
</head>
<body>
  <div id="d1">Div 1</div>
  <div id="d2">Div 2</div>
  <div id="d3">Div 3</div>
  <div id="d4">Div 4</div>
  <div id="d5">Div 5</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



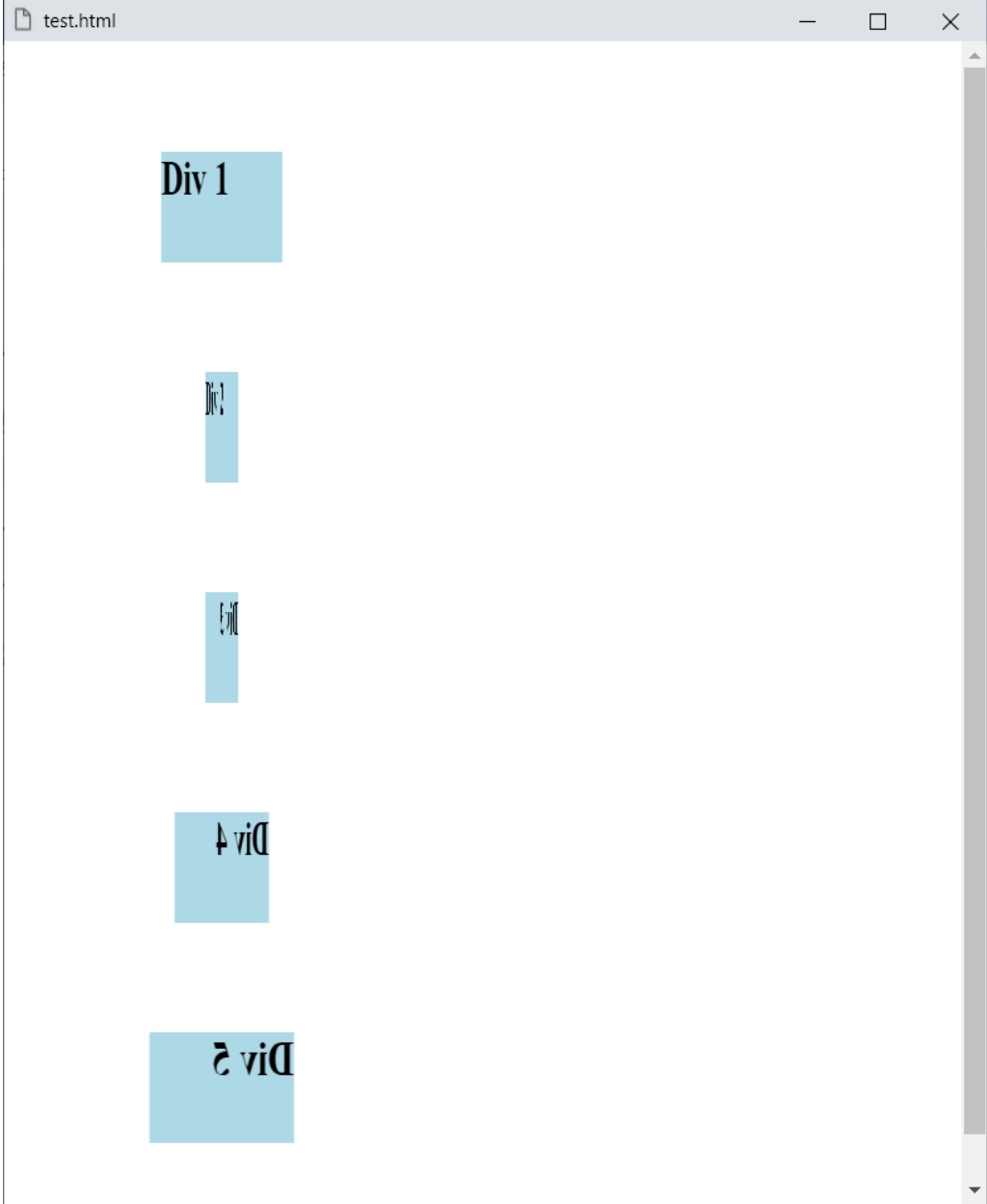
لاحظ أنه تم تقليصه، حتى تم عكسه أفقياً.

استدارة رأسية rotateY

تخيل معي وأن العنصر ينكمنش يميناً ويساراً معاً، حتى يتقلب رأسياً،
 وسنعمل هذا بالدالة rotateY، وسوف نبيه ذلك **بالكود!** التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{
    background:lightblue; width:120px; height:70px;
    margin:70px; font-size:30px; font-weight:bold;
  }
  #d1{ transform: rotateY(50deg) }
  #d2{ transform: rotateY(80deg) }
  #d3{ transform: rotateY(100deg) }
  #d4{ transform: rotateY(120deg) }
  #d5{ transform: rotateY(140deg) }
</style>
</head>
<body>
  <div id="d1">Div 1</div>
  <div id="d2">Div 2</div>
  <div id="d3">Div 3</div>
  <div id="d4">Div 4</div>
  <div id="d5">Div 5</div>
</body>
</html>
```

وبءء ءشءعل الصءفة ءظهء كما بالشكل ءءالئ:



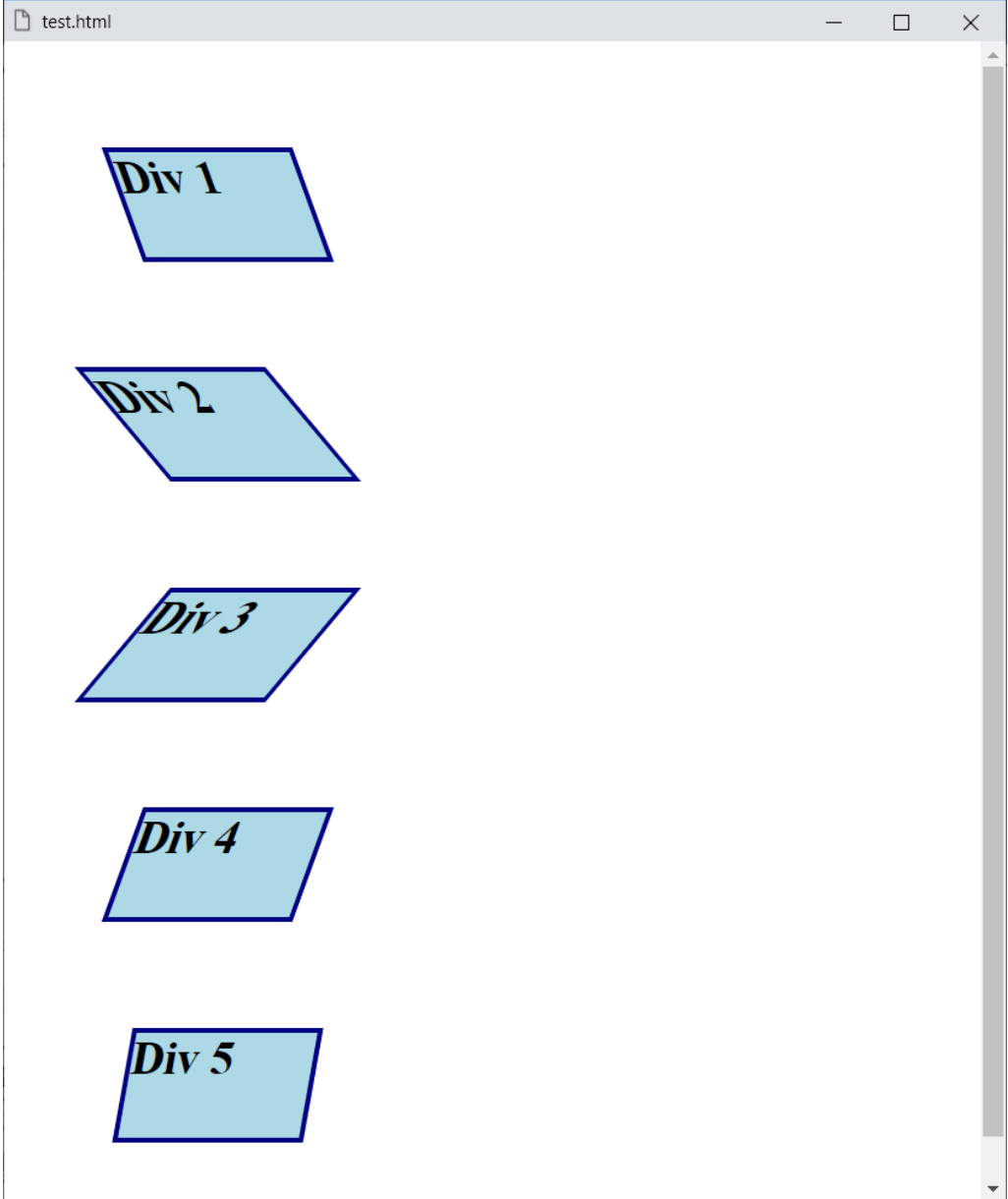
لاءظ أنه ءم ءقلبفه، ءءئ ءم ءكسه رأسباً.

الانحراف الأفقي skewX

تخيل معي لو أن الشكل المستطيل انحرف فأصبح الضلعين الطويلين متوازيين
يميناً أو يساراً، وسنستخدم هذا بالدالة skewX، وسوف نريك ذلك **بالكود!**
التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{
    background:lightblue; width:120px; height:70px;
    margin:70px; font-size:30px; font-weight:bold;
    border:3px solid navy;
  }
  #d1{ transform: skewX(20deg) }
  #d2{ transform: skewX(40deg) }
  #d3{ transform: skewX(140deg) }
  #d4{ transform: skewX(160deg) }
  #d5{ transform: skewX(170deg) }
</style>
</head>
<body>
<div id="d1">Div 1</div>
<div id="d2">Div 2</div>
<div id="d3">Div 3</div>
<div id="d4">Div 4</div>
<div id="d5">Div 5</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



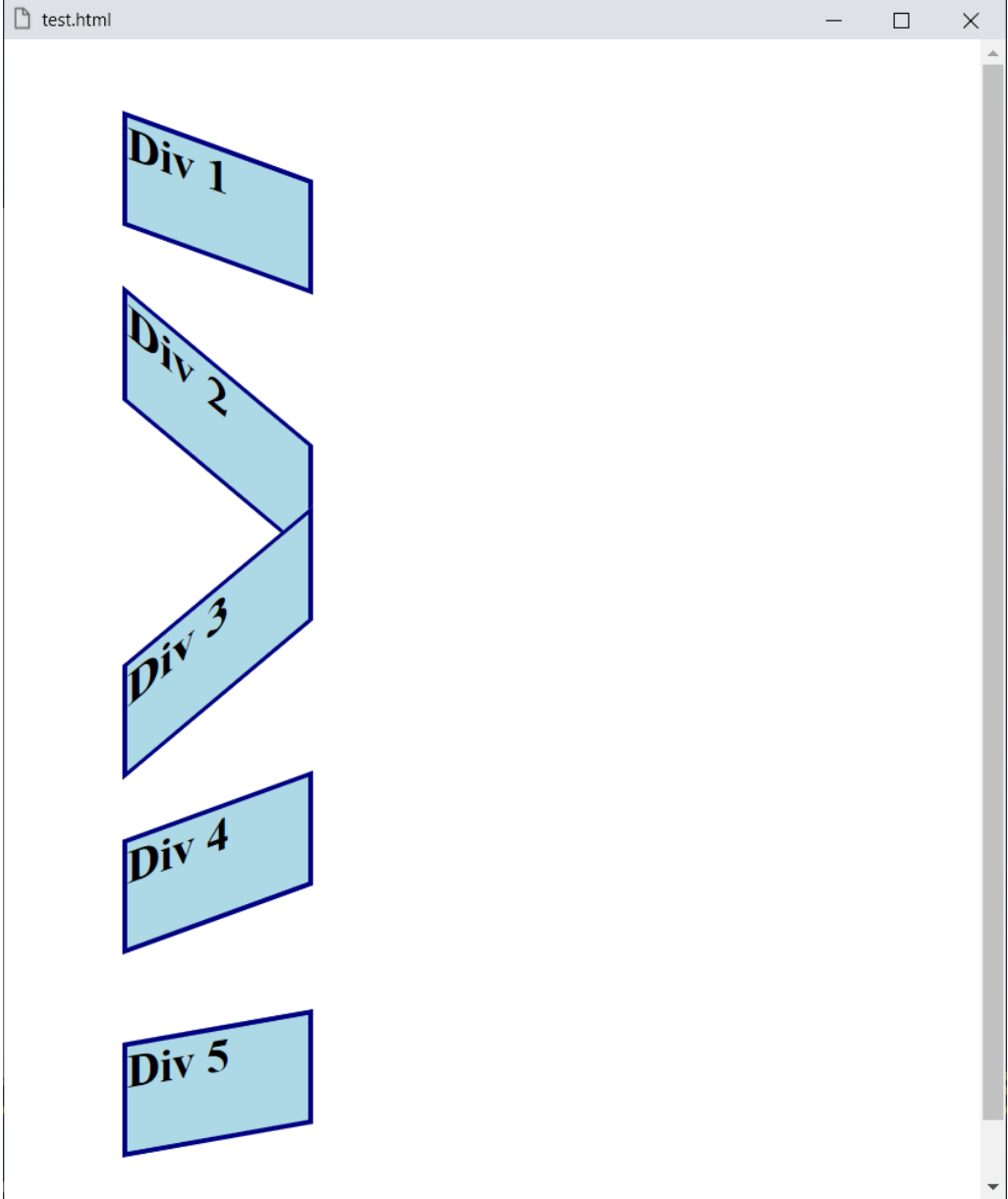
لاحظ أن الشكل ينحرف يميناً أو يساراً حسب الزاوية.

الانحراف الرأسى skewY

تخيل معي لو أن الشكل المستطيل انحرف فأصبح الضلعين العموديين مائلين أعلى أو أسفل، وسنعمل هذا بالدالة skewY، وسوف نبيّن ذلك بالكود التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{
    background:lightblue; width:120px; height:70px;
    margin:70px; font-size:30px; font-weight:bold;
    border:3px solid navy;
  }
  #d1{ transform: skewY(20deg) }
  #d2{ transform: skewY(40deg) }
  #d3{ transform: skewY(140deg) }
  #d4{ transform: skewY(160deg) }
  #d5{ transform: skewY(170deg) }
</style>
</head>
<body>
<div id="d1">Div 1</div>
<div id="d2">Div 2</div>
<div id="d3">Div 3</div>
<div id="d4">Div 4</div>
<div id="d5">Div 5</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



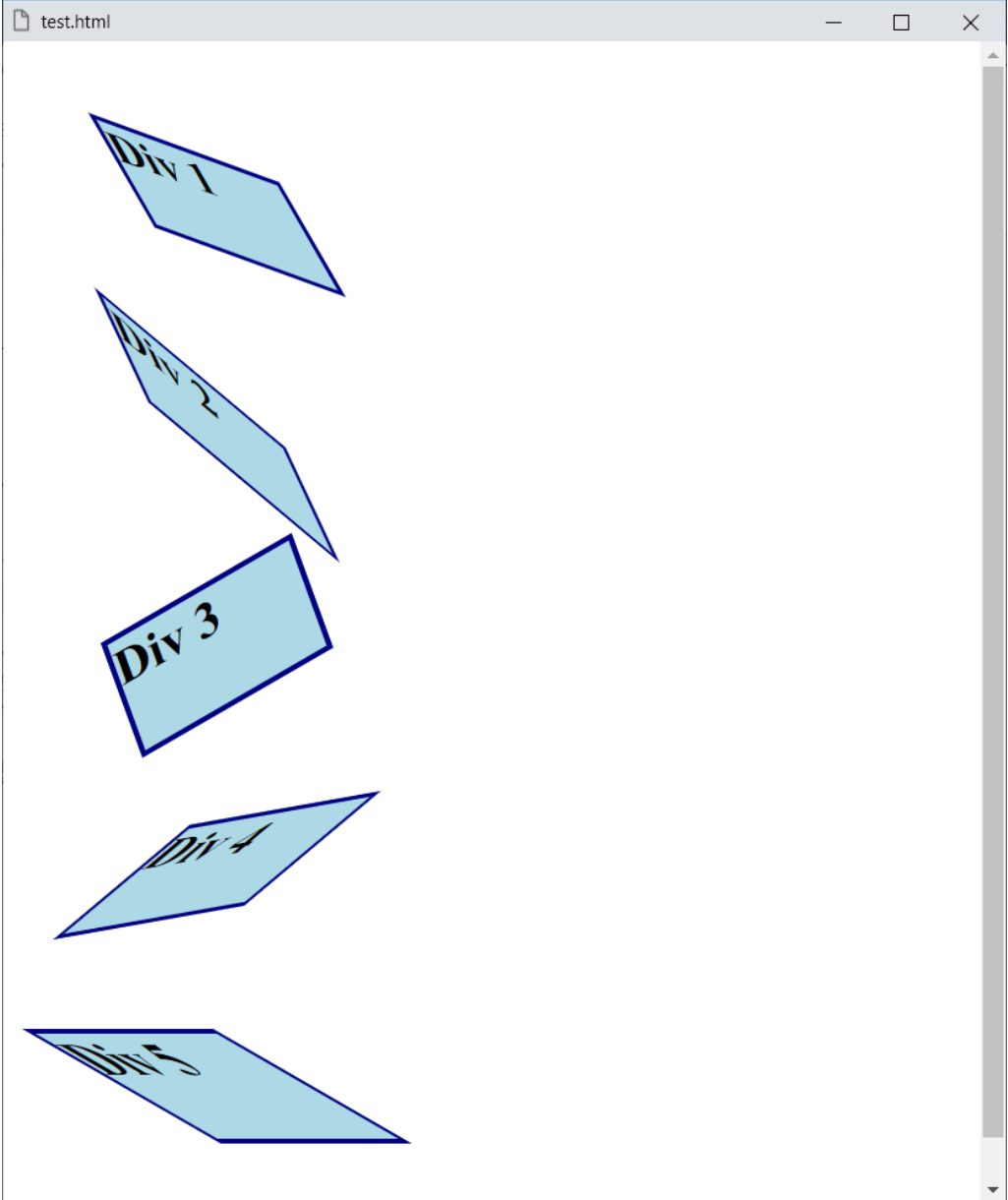
لاحظ أن الشكل ينحرف أعلى أو أسفل حسب الزاوية.

الانحراف الكلي skew

تخيل معي لو أن الشكل المستطيل انحرف فأصبح الضلعين الطويلين والضلعين العرضيين مائلين أعلى أو أسفل يميناً ويساراً، وسنعمل هذا بالدالة skew التي تأخذ زاوية للـ x ثم زاوية للـ y، وسوف نبين ذلك بالكوود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
div{
background:lightblue; width:120px; height:70px;
margin:70px; font-size:30px; font-weight:bold;
border:3px solid navy;
}
#d1{ transform: skew(30deg, 20deg) }
#d2{ transform: skew(25deg, 40deg) }
#d3{ transform: skew(20deg, 150deg) }
#d4{ transform: skew(130deg, 170deg) }
#d5{ transform: skew(240deg, 180deg) }
</style>
</head>
<body>
<div id="d1">Div 1</div>
<div id="d2">Div 2</div>
<div id="d3">Div 3</div>
<div id="d4">Div 4</div>
<div id="d5">Div 5</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



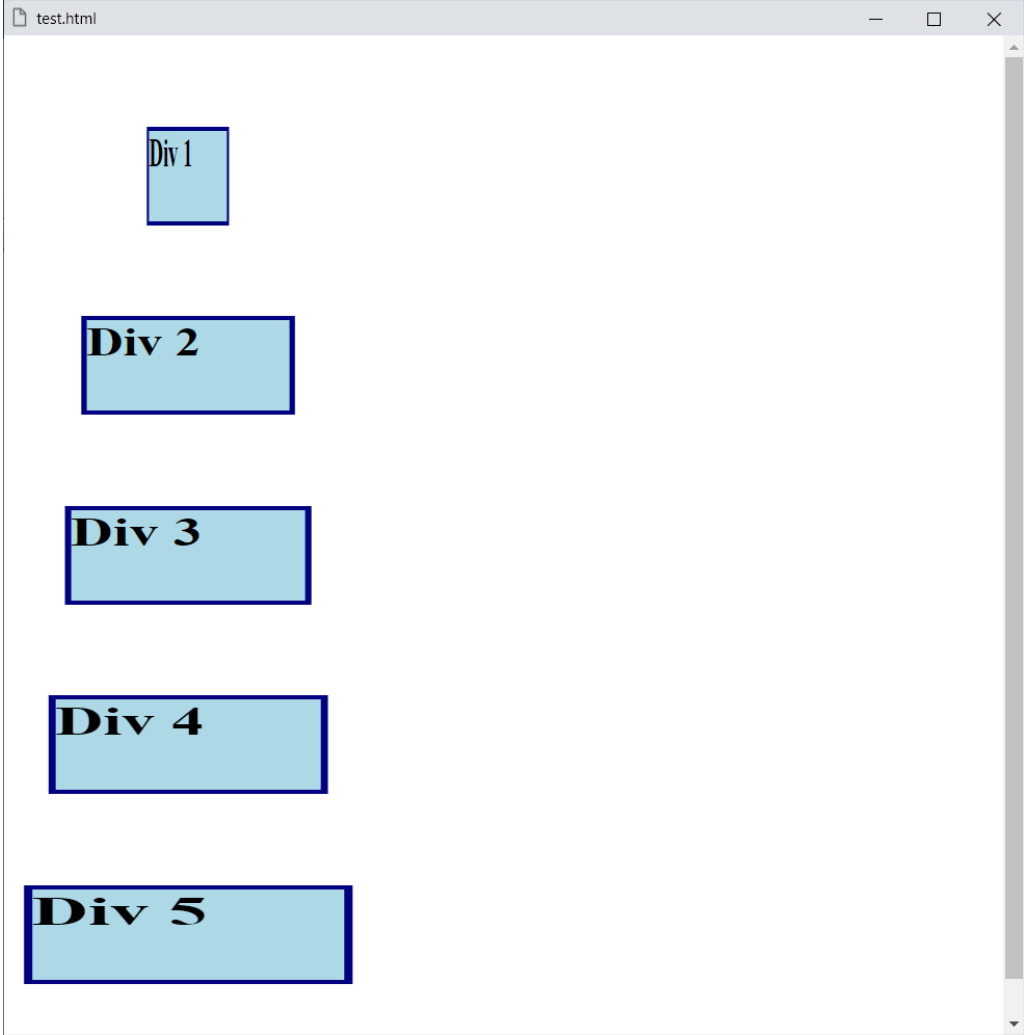
لاحظ أن هناك انحرافات من جميع الاتجاهات بناء على قيم الزوايا.

المقياس الأفقي scaleX

تخيل معي لو أنك أمسكت جسم مطاطي طوله 10سم، ثم قمت بشده بمقدار 20سم أخرى فأصبح ضعفي حجمه، فهكذا تفعل الدالة scaleX، سواء يكبر، أو يصغر، فالقيمة 1 هي الحجم الطبيعي، وتقل لتصغر أفقياً، وتزيد لتكبر أفقياً، وسوف نبين ذلك **بالكود!** التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
div{
background:lightblue; width:120px; height:70px;
margin:70px; font-size:30px; font-weight:bold;
border:3px solid navy;
}
#d1{ transform: scaleX(0.5) }
#d2{ transform: scaleX(1.3) }
#d3{ transform: scaleX(1.5) }
#d4{ transform: scaleX(1.7) }
#d5{ transform: scaleX(2) }
</style>
</head>
<body>
<div id="d1">Div 1</div>
<div id="d2">Div 2</div>
<div id="d3">Div 3</div>
<div id="d4">Div 4</div>
<div id="d5">Div 5</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



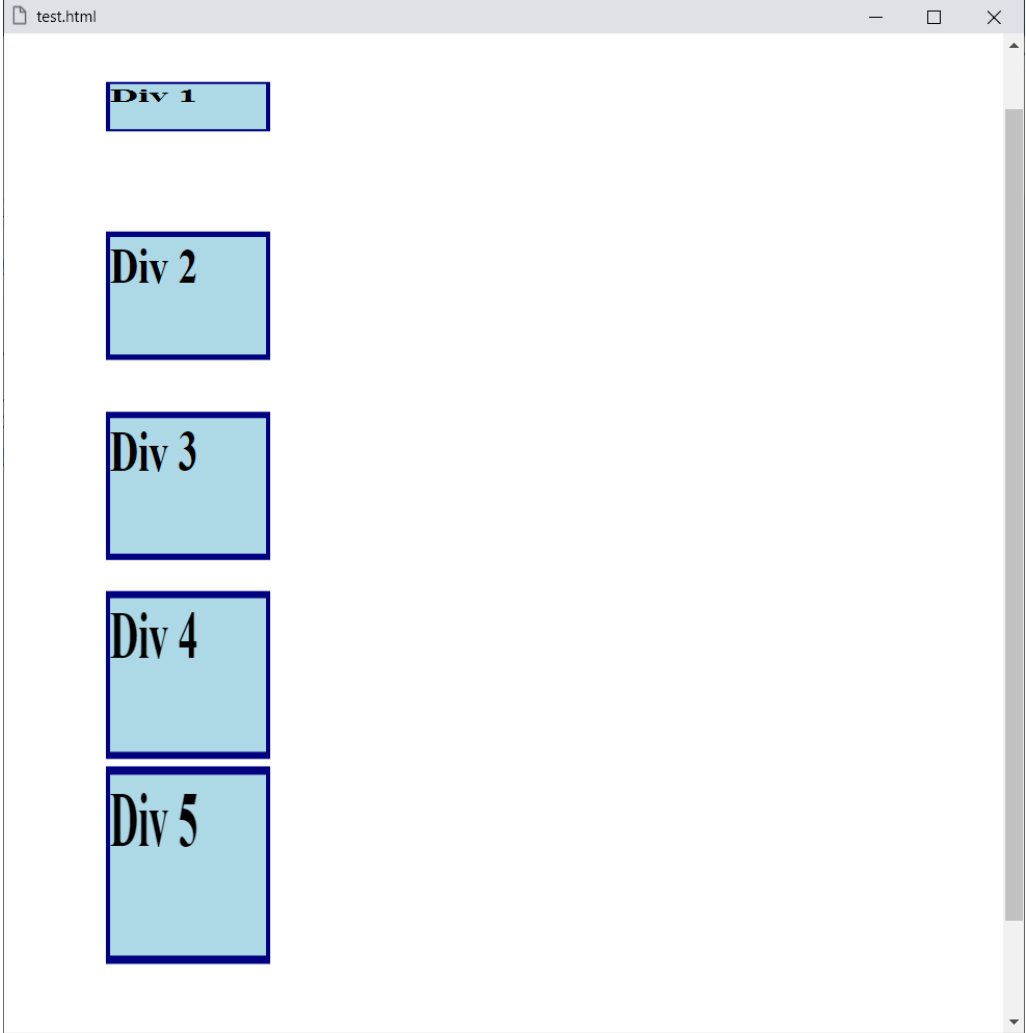
لاحظ أن ال scale تغير أفقياً، فتم التأيد على شكل العنصر العام، فمثلاً أنظر الى Div 5 وقارنه بيه الإطارين اعلي وأسفل، والإطارين يميناً وشمالاً، وسوف تلاحظ أن الأحجام مختلفة لأن ال scale تغير أفقياً بالدالة scaleX.

المقياس الرأسى scaleY

تخيل معي لو أنك أمسكت جسم مطاطي طوله 10سم، ثم قمت بشده بمقدار 20سم أخري فأصبح ضعيف حجه، فهكذا تفعل الدالة scaleY، سواء يكبر، أو يصغر، فالقيمة 1 هي الحجم الطبيعي، وتقل لتصغر رأسياً، وتزيد لتكبر رأسياً، وسوف نبين ذلك **بالكود!** التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
div{
background:lightblue; width:120px; height:70px;
margin:70px; font-size:30px; font-weight:bold;
border:3px solid navy;
}
#d1{ transform: scaleY(0.5) }
#d2{ transform: scaleY(1.3) }
#d3{ transform: scaleY(1.5) }
#d4{ transform: scaleY(1.7) }
#d5{ transform: scaleY(2) }
</style>
</head>
<body>
<div id="d1">Div 1</div>
<div id="d2">Div 2</div>
<div id="d3">Div 3</div>
<div id="d4">Div 4</div>
<div id="d5">Div 5</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



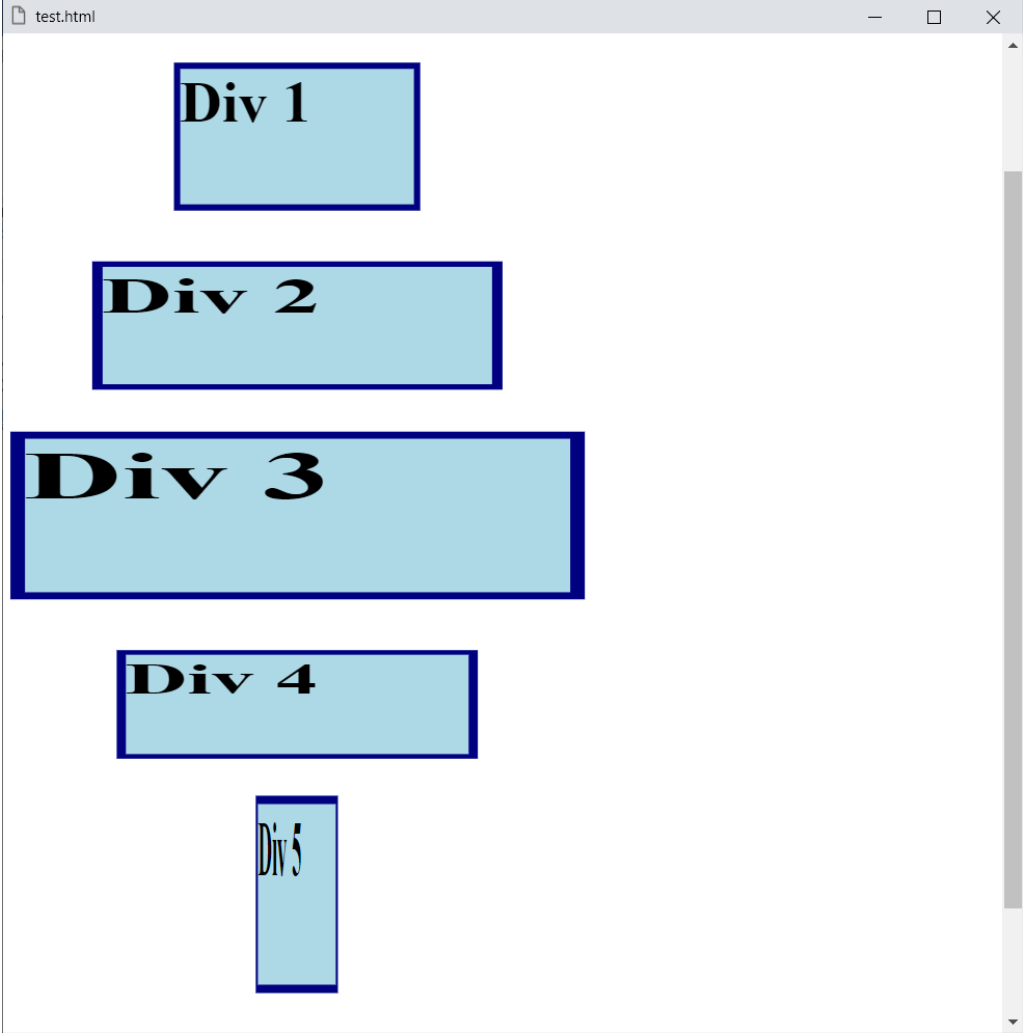
لاحظ أن الـ scale تغير رأسياً، فتم التأيد على شكل العنصر العام،
فمثلاً أنظر إلى Div 5 وقارن بين الإطارين اعلي وأسفل، والإطارين
يميناً وشمالاً، وسوف تلاحظ أن الأحجام مختلفة لأن الـ scale تغير
رأسياً بالدالة scaleY.

المقياس الكلي scale

الخاصية scale تجمع بين الخاصيتين scaleX و scaleY، في نفس الوقت، حيث تأخذ قيمة الـ x ، ثم قيمة الـ y ، فيتم تعديل المقياس. **كود!** قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{ padding:85px; }
  div{
    background:lightblue; width:120px; height:70px;
    margin:70px; font-size:30px; font-weight:bold;
    border:3px solid navy;
  }
  #d1{ transform: scale(1.5, 1.5) }
  #d2{ transform: scale(2.5, 1.3) }
  #d3{ transform: scale(3.5, 1.7) }
  #d4{ transform: scale(2.2, 1.1) }
  #d5{ transform: scale(0.5, 2) }
</style>
</head>
<body>
<div id="d1">Div 1</div>
<div id="d2">Div 2</div>
<div id="d3">Div 3</div>
<div id="d4">Div 4</div>
<div id="d5">Div 5</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن الـ scale تغير أفقياً ورأسياً في نفس الوقت.

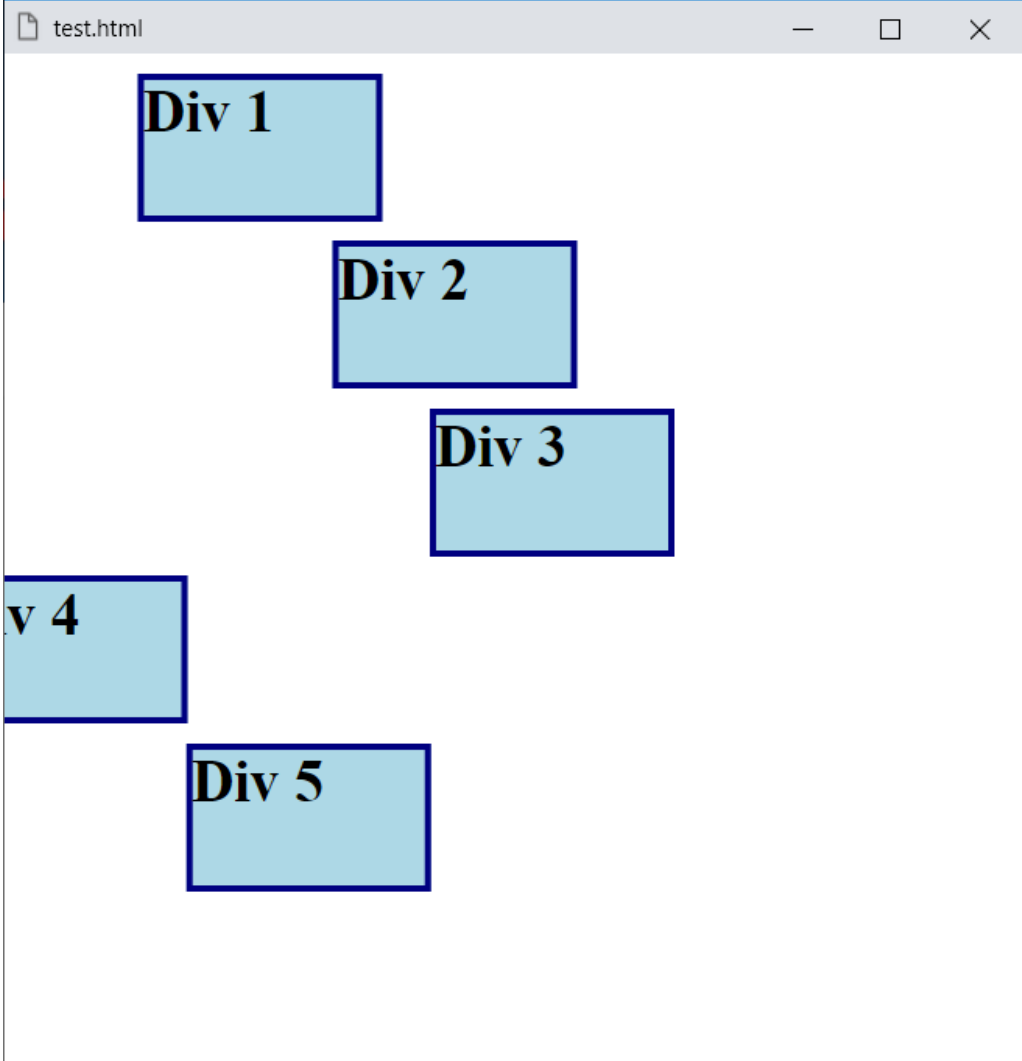
انتبه!، لو تم تخصيص القيمة 1- فسوف يتم انعكاس الصورة.

تدريك أفقي translateX

يمكنك تدريك العنصر أفقياً، نحو اليمين كلما زادت القيمة، أو نحو اليسار كلما قلت القيمة، وسوف نبيه ذلك بالكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{
    background:lightblue; width:120px; height:70px;
    margin:10px; font-size:30px; font-weight:bold;
    border:3px solid navy;
  }
  #d1{ transform: translateX(50px); }
  #d2{ transform: translateX(150px); }
  #d3{ transform: translateX(200px); }
  #d4{ transform: translateX(-50px); }
  #d5{ transform: translateX(75px); }
</style>
</head>
<body>
<div id="d1">Div 1</div>
<div id="d2">Div 2</div>
<div id="d3">Div 3</div>
<div id="d4">Div 4</div>
<div id="d5">Div 5</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



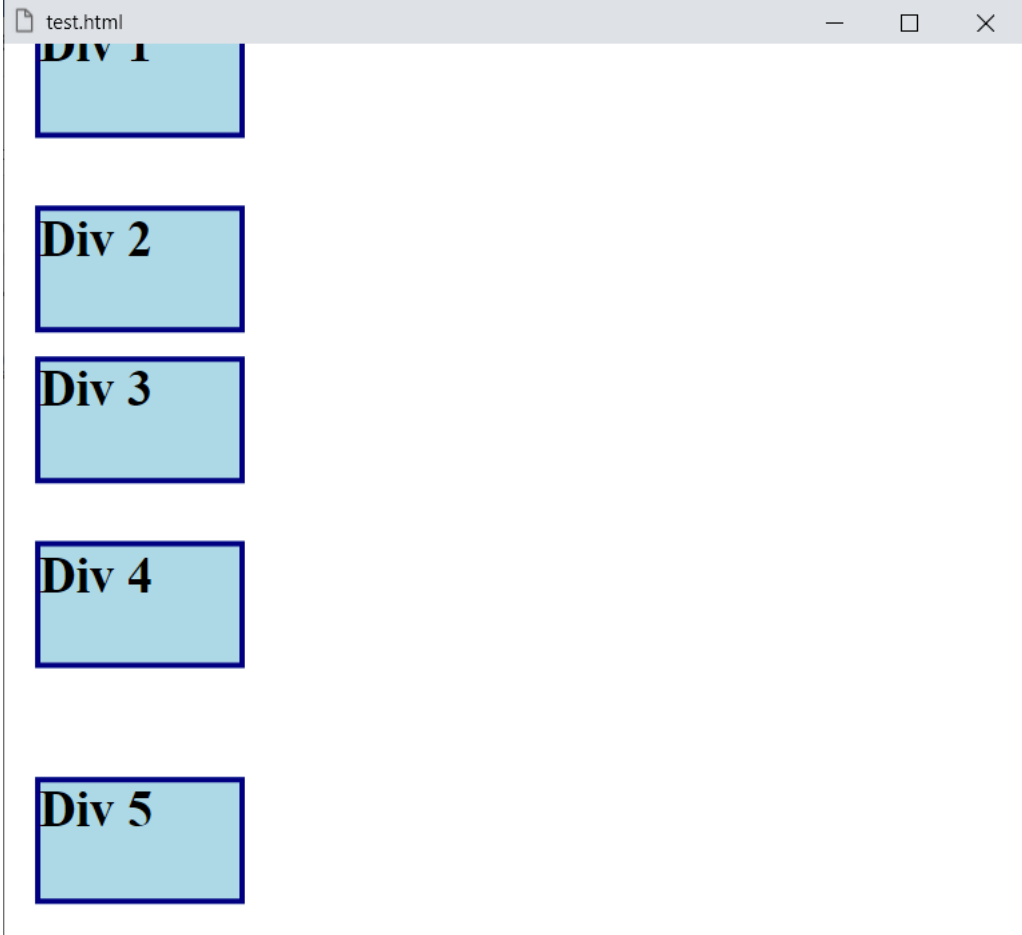
لاحظ أنه كلما زادت القيمة الموجودة في الدالة `translateX` تترشح العنصر إلى اليمين، وكلما قلت تترشح إلى اليسار، مع العلم أن اتجاه الصفحة هو `Left to Right` بشكل افتراضي، ولاحظ أن الذي أخذ قيمة سالبة خرج عن الصفحة.

تدريك أفقي translateY

يمكنك تدريك العنصر رأسياً، الى الأعلى كلما زادت القيمة، أو الى الأسفل كلما قلت القيمة، وسوف نبيد ذلك بالكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{
    background:lightblue; width:120px; height:70px;
    margin:10px; font-size:30px; font-weight:bold;
    border:3px solid navy;
  }
  #d1{ transform: translateY(-30px); }
  #d2{ transform: translateY(1px); }
  #d3{ transform: translateY(5px); }
  #d4{ transform: translateY(30px); }
  #d5{ transform: translateY(85px); }
</style>
</head>
<body>
<div id="d1">Div 1</div>
<div id="d2">Div 2</div>
<div id="d3">Div 3</div>
<div id="d4">Div 4</div>
<div id="d5">Div 5</div>
</body>
</html>
```

وبءء ءشءعل الصءفة ءظهر ءما بالشءك ءءل:



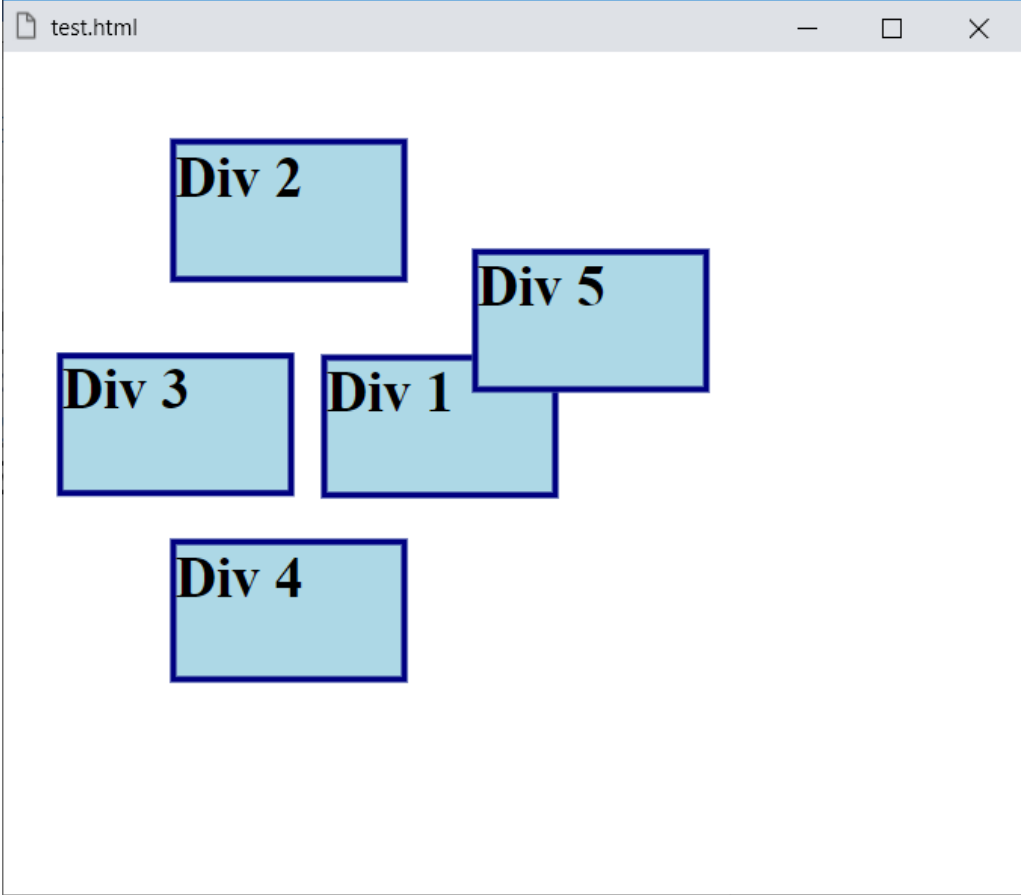
لاءظ أنه ءلما زاءء القفمة الموءوءة فف الءالة `translateY` ءرءءء العنصر الف الأعلى، وءلما قلت ءرءءء الف الأسفل، ولاءظ أن الءف آءء قفمة سالبة آءء عء الصءفة، ولاءظ أفضاً أن العنصر يأءء قفمءه عء موءوءعه، أفف لا يأءء القفمة مءل `top` و `bottom` و `right` و `left`، وءلك فءءرء بشءك ءبففعف سواء ب `translateX` أو `translateY` عء موءوءعه.

تدريك كمي translate

يمكّنك تدريك العنصر أفقياً ورأسياً، أي، يميناً ويساراً، وأعلى وأسفل، في نفس الوقت، وسوف نبيّن ذلك **بالكود!** التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{
    background:lightblue; width:120px; height:70px;
    margin:10px; font-size:30px; font-weight:bold;
    border:3px solid navy;
  }
  #d1{ transform: translate(150px, 150px); }
  #d2{ transform: translate(70px, -50px); }
  #d3{ transform: translate(10px, -23px); }
  #d4{ transform: translate(70px, -10px); }
  #d5{ transform: translate(230px, -250px); }
</style>
</head>
<body>
<div id="d1">Div 1</div>
<div id="d2">Div 2</div>
<div id="d3">Div 3</div>
<div id="d4">Div 4</div>
<div id="d5">Div 5</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم تغيير كلي لأماكن العناصر، ولاحظ أن الدالة translate تأخذ الـ x ثم الـ y .

مفهوم توزيعة التحول الكلي matrix

بالدالة matrix يمكنه عمل ال skew، وعمل ال scale، كما يمكنه عمل ال translate، بل وحتى ال rotate، أي أنك تستطيع عمل تحويلات كاملة الأركان بهذه الدالة المتميزة.

الدالة matrix تتطلب 6 من القيم للعمل، وذلك بالترتيب التالي:

- التمدد الأفقي scaleX
- الانحراف الرأسى skewY
- الانحراف الأفقي skewX
- التمدد الرأسى scaleY
- ترحيل أفقي translateX
- ترحيل رأسى translateY

لابد من فهم شيء هام جداً قبل البدء في matrix، وهو مثلاً عند العمل بالدالة scale واعطاؤها القيم 1 ثم 1 لك يتغير العنصر، أو مثلاً عند العمل بالدالة translate واعطاؤها 0 ثم 0 لك يتغير العنصر، وهكذا، وكذلك عند العمل بالدالة matrix مع احتياج عدم التغيير في العنصر تأخذ القيم 1 ثم 0 ثم 0 ثم 1 ثم 0، فيكون الكود مثل التالي:

```
transform: matrix(1, 0, 0, 1, 0, 0);
```

رجاء انتبه! ثم انتبه! ثم انتبه! جيداً لما يلي:

- لا بد من حفظ الترتيب جيداً لما تأخذه الدالة matrix بشكل ممتاز scaleX, skewY, skewX, scaleY, translateX, translateY
- الدالة matrix لا يكتب فيها مقاييس مثل px و deg وغيرها
- مثلاً عند ضبط ال skew على 10deg، ونريد ما يعادلها في الدالة matrix، فإن القيمة 0.1758 تقريباً مثلها
- يجب التركيز جيداً، لأنه سوف يتم عمل كل تحول بشكل منفصل، ثم يتم استخدامهم جميعاً
- يجب التركيز جيداً، لأنه عند عمل أمثلة، سيتم العمل ب matrix وما يقابلها بما تعلمناه من قبل حتى الآن
- يجب ملاحظة الفوارق التي يتم كتابتها في الكود لإظهار التأثيرات، مثل عمل padding للوسم body، أو تزويد ال margin للعناصر، أو غير ذلك
- يجب مراعاة دراسة التحولات السابقة جيداً للتعامل فيما يلي بسهولة

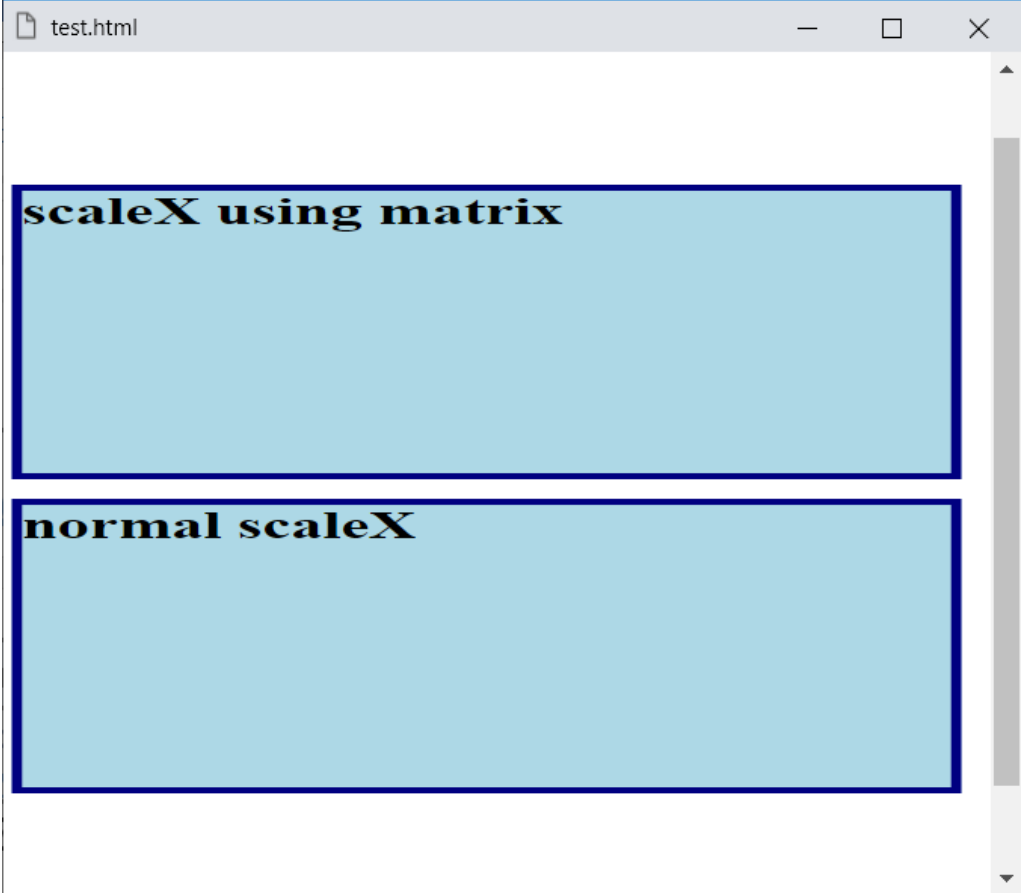
عمل ال scaleX بواسطة matrix

قبل البدء في هذا الموضوع يجب تعلم ال skewY, scaleX, translateY, translateX, scaleY, skewX, ويجب مراعاة ما في موضوع "مفهوم توزيعة التحول الكلي matrix".

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{ padding:90px; }
  div{
    background:lightblue; width:290px; height:150px;
    margin:10px; font-size:20px; font-weight:bold;
    border:3px solid navy;
  }
  #d1{ transform: matrix(1.7, 0, 0, 1, 0, 0); }
  #d2{ transform: scaleX(1.7); }
</style>
</head>
<body>
  <div id="d1">scaleX using matrix</div>
  <div id="d2">normal scaleX</div>
</body>
</html>
```

وءءء ءشءعل الصءفة ءظهء ءما بالشءك ءءلى:



لاءظ أن ما ءم عملة باءءءءام الءالة scaleX هو، هو ما ءم عملة
بالءالة .matrix.

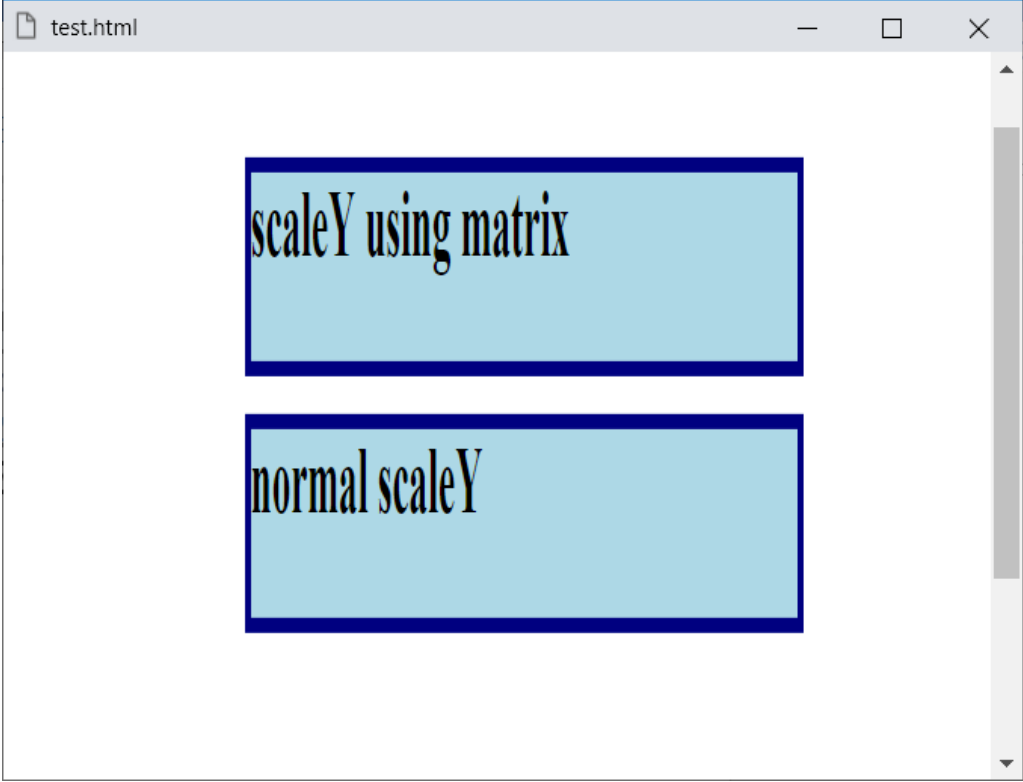
عمل ال scaleY بواسطة matrix

قبل البدء في هذا الموضوع يجب تعلم ال scaleX و skewY و translateX و translateY جيداً، ويجب مراعاة ما في موضوع "مفهوم توزيعة التحول الكلي matrix".

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body{ padding:30px; }
      div{
        background:lightblue; width:290px; height:40px;
        margin:90px; font-size:20px; font-weight:bold;
        border:3px solid navy;
      }
      #d1{ transform: matrix(1, 0, 0, 2.5, 0, 0); }
      #d2{ transform: scaleY(2.5); }
    </style>
  </head>
  <body>
    <div id="d1">scaleY using matrix</div>
    <div id="d2">normal scaleY</div>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن ما تم عملة باستخدام الالة scaleY هو، هو ما تم عملة بالالة matrix.

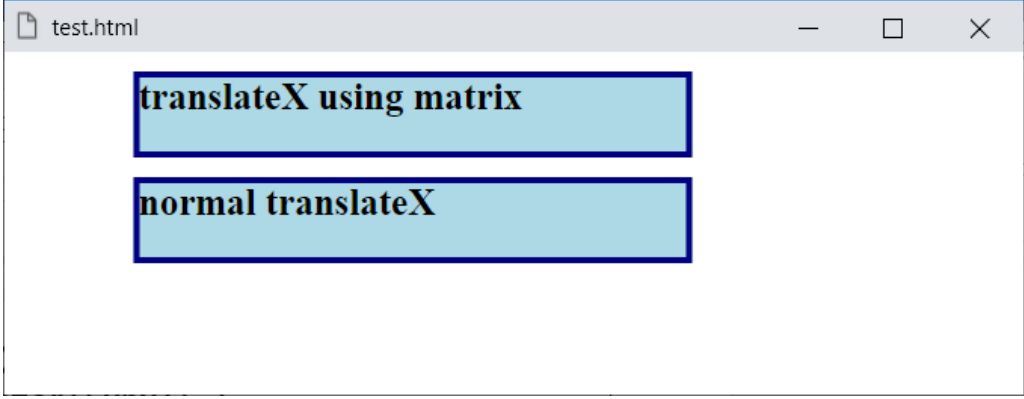
عمل ال `matrix` بواسطة `translateX`

قبل البدء في هذا الموضوع يجب تعلم ال `skewY`, `scaleX` و `skewX`, `scaleY`, `translateX`, `translateY` جيداً، ويجب مراعاة ما في موضوع "مفهوم توزيعة التحول الكلي `matrix`".

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{
    background:lightblue; width:290px; height:40px;
    margin:10px; font-size:20px; font-weight:bold;
    border:3px solid navy;
  }
  #d1{ transform: matrix(1, 0, 0, 1, 50, 0); }
  #d2{ transform: translateX(50px); }
</style>
</head>
<body>
  <div id="d1">translateX using matrix</div>
  <div id="d2">normal translateX</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن ما تم عملة باستخدام الڤالة `translateX` هو، هو ما تم عملة بالڤالة `matrix`، مع العلم أن العنصر يتحرك من موضعه الأصلي.

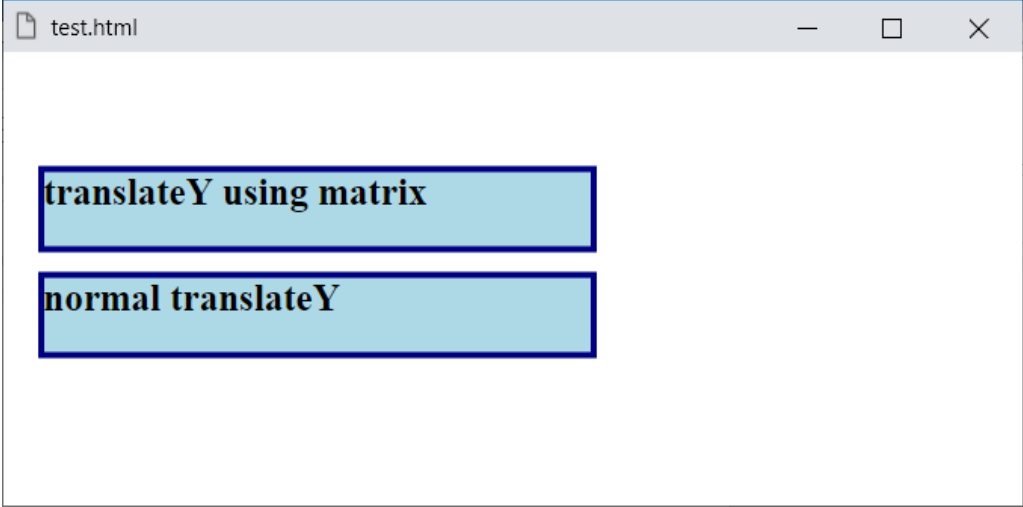
عمل ال `translateY` بواسطة `matrix`

قبل البدء في هذا الموضوع يجب تعلم ال `scaleX`، `skewY`، `skewX`، `scaleY`، `translateX`، `translateY` جيداً، ويجب مراعاة ما في موضوع "مفهوم توزيعة التحول الكلي `matrix`".

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{
    background:lightblue; width:290px; height:40px;
    margin:10px; font-size:20px; font-weight:bold;
    border:3px solid navy;
  }
  #d1{ transform: matrix(1, 0, 0, 1, 0, 50); }
  #d2{ transform: translateY(50px); }
</style>
</head>
<body>
  <div id="d1">translateY using matrix</div>
  <div id="d2">normal translateY</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن ما تم عملة باستخدام الدالة translateY هو، هو ما تم عملة بالدالة matrix، مع العلم أن العنصر يتحرك من موضعه الأصلي.

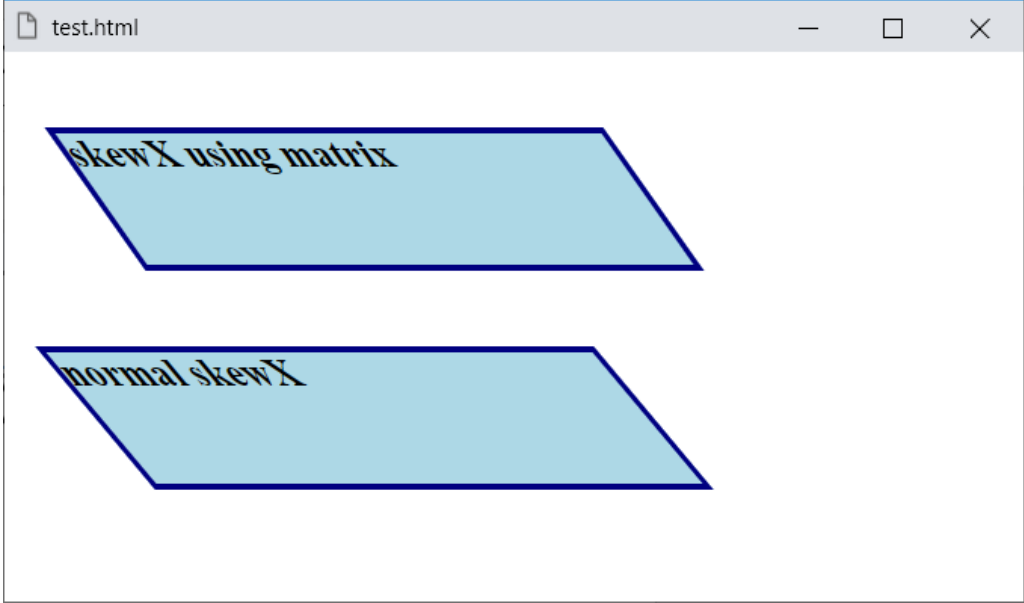
عمل ال skewX بواسطة matrix

قبل البدء في هذا الموضوع يجب تعلم ال skewY, scaleX, translateY, translateX, scaleY, skewX, ويجب مراعاة ما في موضوع "مفهوم توزيعة التحول الكلي matrix".

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
div{
background:lightblue; width:290px; height:70px;
margin:40px; font-size:20px; font-weight:bold;
border:3px solid navy;
}
#d1{ transform: matrix(1, 0, 0.7032, 1, 0, 0); }
#d2{ transform: skewX(40deg); }
</style>
</head>
<body>
<div id="d1">skewX using matrix</div>
<div id="d2">normal skewX</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن ما تم عملة باستخدام الدالة `skewX` هو، هو ما تم عملة بالدالة `matrix`.

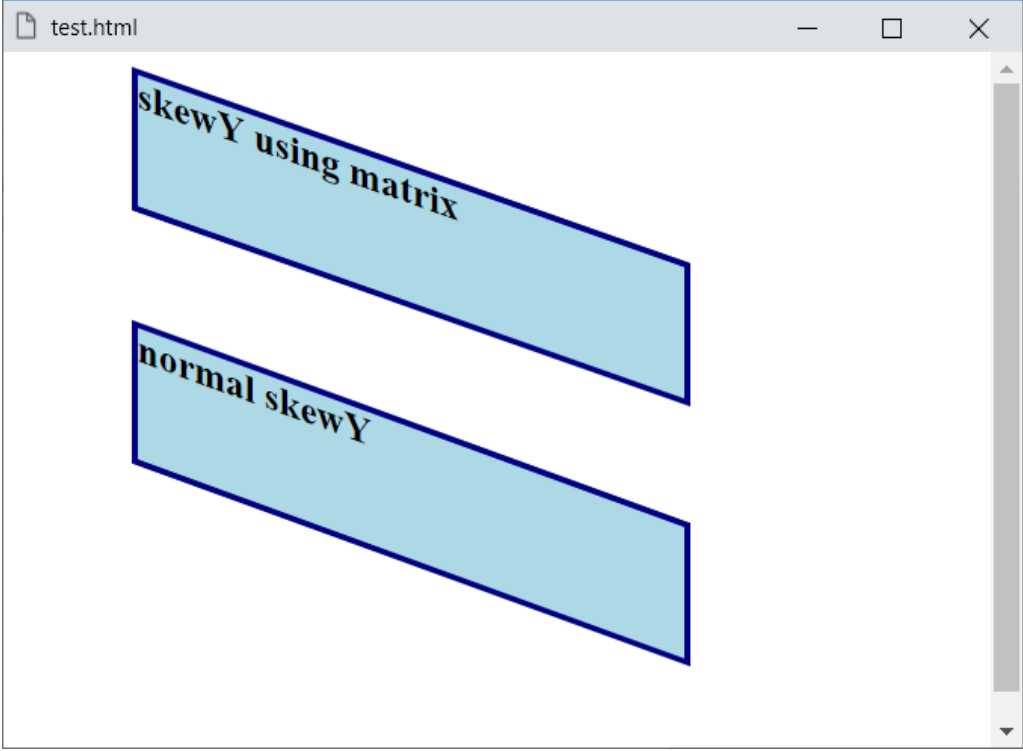
عمل ال skewY بواسطة matrix

قبل البدء في هذا الموضوع يجب تعلم ال skewX, scaleY, skewX, translateX, translateY جيداً، ويجب مراعاة ما في موضوع "مفهوم توزيعة التحول الكلي matrix".

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
div{
background:lightblue; width:290px; height:70px;
margin:60px; font-size:20px; font-weight:bold;
border:3px solid navy;
}
#d1{ transform: matrix(1, 0.3516, 0, 1, 0, 0); }
#d2{ transform: skewY(20deg); }
</style>
</head>
<body>
<div id="d1">skewY using matrix</div>
<div id="d2">normal skewY</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن ما تم عملة باستخدام الءالة skewY هو، هو ما تم عملة بالءالة .matrix.

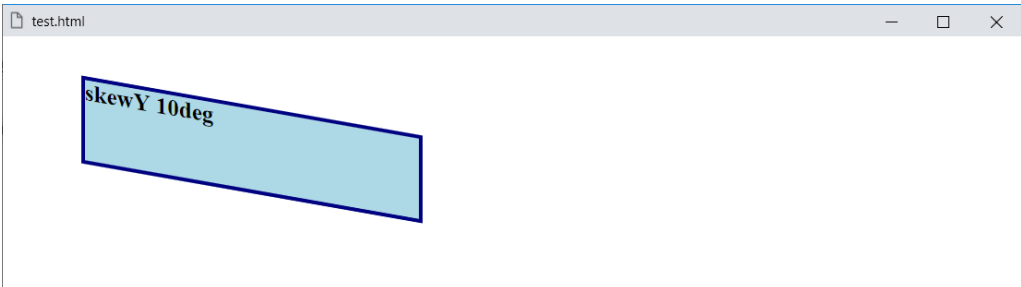
عمل ال rotate بواسطة matrix

قبل البدء في هذا الموضوع يجب تعلم ال skewY, scaleX, translateY, translateX, scaleY, skewX, ويجب مراعاة ما في موضوع "مفهوم توزيعة التحول الكلي matrix".

انتبه! وركز في المثال بالكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
div{
background:lightblue; width:290px; height:70px;
margin:60px; font-size:20px; font-weight:bold;
border:3px solid navy;
}
#d1{ transform: matrix(1, 0.1758, 0, 1, 0, 0); }
</style>
</head>
<body>
<div id="d1">skewY 10deg</div>
</body>
</html>
```

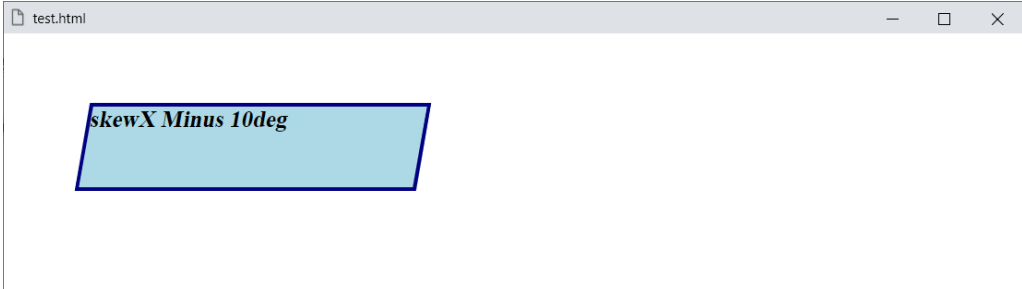
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



والآن **انتبه!** مرة أخرى، وركز في المثال **بالكود!** التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{
    background:lightblue; width:290px; height:70px;
    margin:60px; font-size:20px; font-weight:bold;
    border:3px solid navy;
  }
  #d1{ transform: matrix(1, 0, -0.1758, 1, 0, 0); }
</style>
</head>
<body>
  <div id="d1">skewX Minus 10deg</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

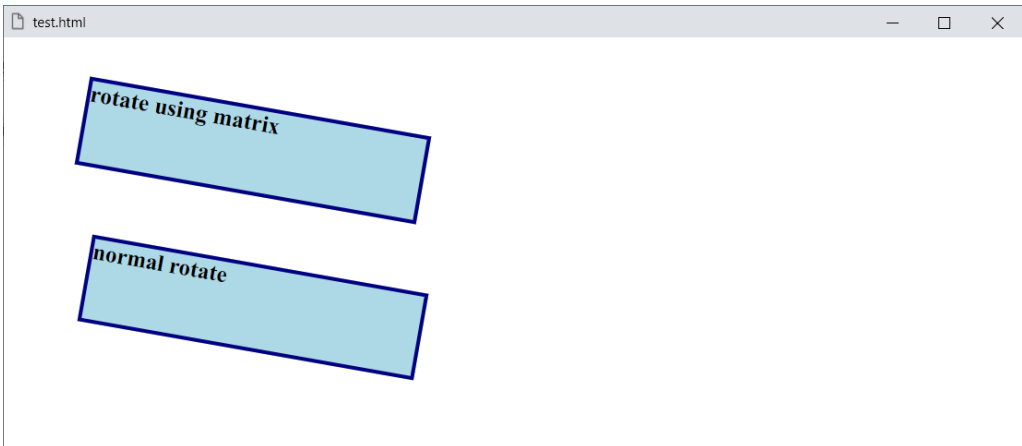


والآن **انتبه!**، واستنبط من المثالين السابقين أنه تم عمل نفس الزاوية للـ `skewX` والـ `skewY`، ولكه واحدة بالموجب وأخرى بالسالب، ومن هنا يمكن القول بأن القيمتين متعادلتيه، وأنهم مكملتيه لبعضهم، وأنهم عكس بعض، فيعود الشكل كما كان، ولكنه يصبح مستدير في ناحية، ولو تم تبديل الموجب والسالب، سوف يستدير في الناحية الأخرى.

والآن مع التوضيحات السابقة يمكنك التجربة بالمثال التالي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
div{
background:lightblue; width:290px; height:70px;
margin:60px; font-size:20px; font-weight:bold;
border:3px solid navy;
}
#d1{ transform: matrix(1, 0.1758, -0.1758, 1, 0, 0); }
#d2{ transform: rotate(10deg); }
</style>
</head>
<body>
<div id="d1">rotate using matrix</div>
<div id="d2">normal rotate</div>
</body>
</html>
```

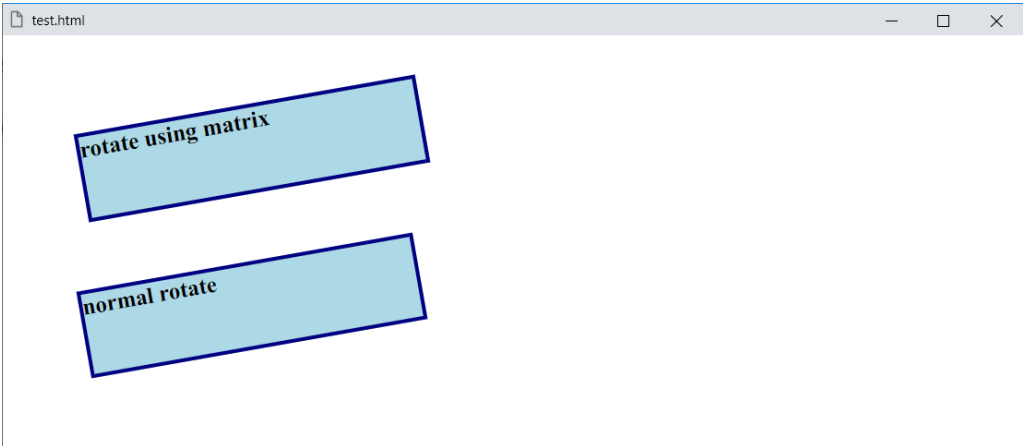
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



ويمكن أيضاً العمل باستدارة عكس الاستدارة السابقة بالمثال التالي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
div{
background:lightblue; width:290px; height:70px;
margin:60px; font-size:20px; font-weight:bold;
border:3px solid navy;
}
#d1{ transform: matrix(1, -0.1758, 0.1758, 1, 0, 0); }
#d2{ transform: rotate(-10deg); }
</style>
</head>
<body>
<div id="d1">rotate using matrix</div>
<div id="d2">normal rotate</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



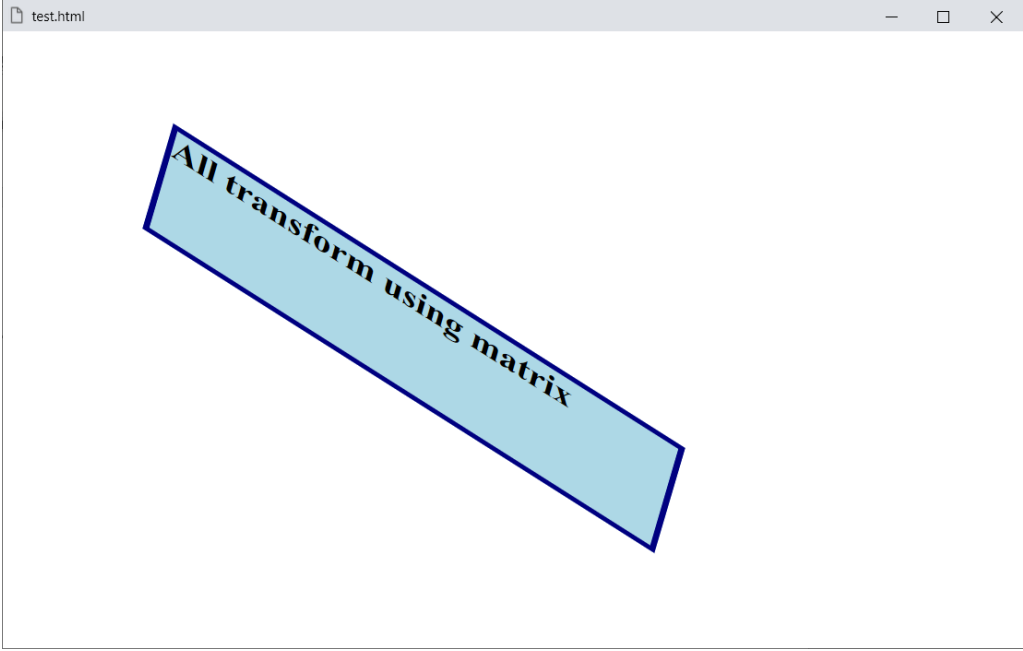
عمل جميع التحويلات بواسطة matrix

قبل البدء في هذا الموضوع يجب تعلم الـ skewY, scaleX, translateY, translateX, scaleY, skewX، ويجب مراعاة ما في موضوع "مفهوم توزيعية التحويلات الكلي matrix"، وأيضاً ما تم شرحه من قبل في الـ matrix.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{ padding:90px; }
  div{
    background:lightblue; width:290px; height:70px;
    margin:60px; font-size:20px; font-weight:bold;
    border:3px solid navy;
  }
  #d1{
    transform: matrix(1.5, 0.9516, -0.3516, 1.2, 50, 70);
  }
</style>
</head>
<body>
  <div id="d1">All transform using matrix</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



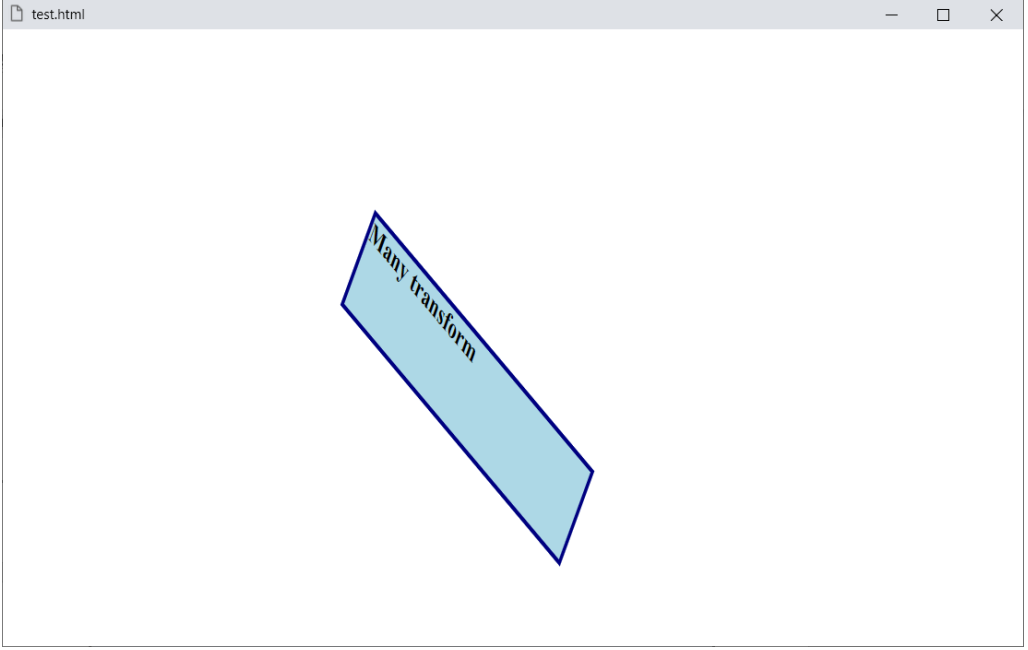
لاحظ أنه تم عمل `translate`, `rotate`, `skew`, `scale` في وقت واحد باستخدام الـ `matrix`.

أكثر من تحول Many Transform

يمكن كتابة أكثر من دالة تحول في نفس الخاصية transform، مع فاصلة بين كل تحول وآخر بمسافة، وسنبيه كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{ padding:90px; }
  div{
    background:lightblue; width:290px; height:70px;
    margin:60px; font-size:20px; font-weight:bold;
    border:3px solid navy;
  }
  #d1{
    transform: rotate(50deg) skew(30deg) translate(150px);
  }
</style>
</head>
<body>
  <div id="d1">Many transform</div>
</body>
</html>
```

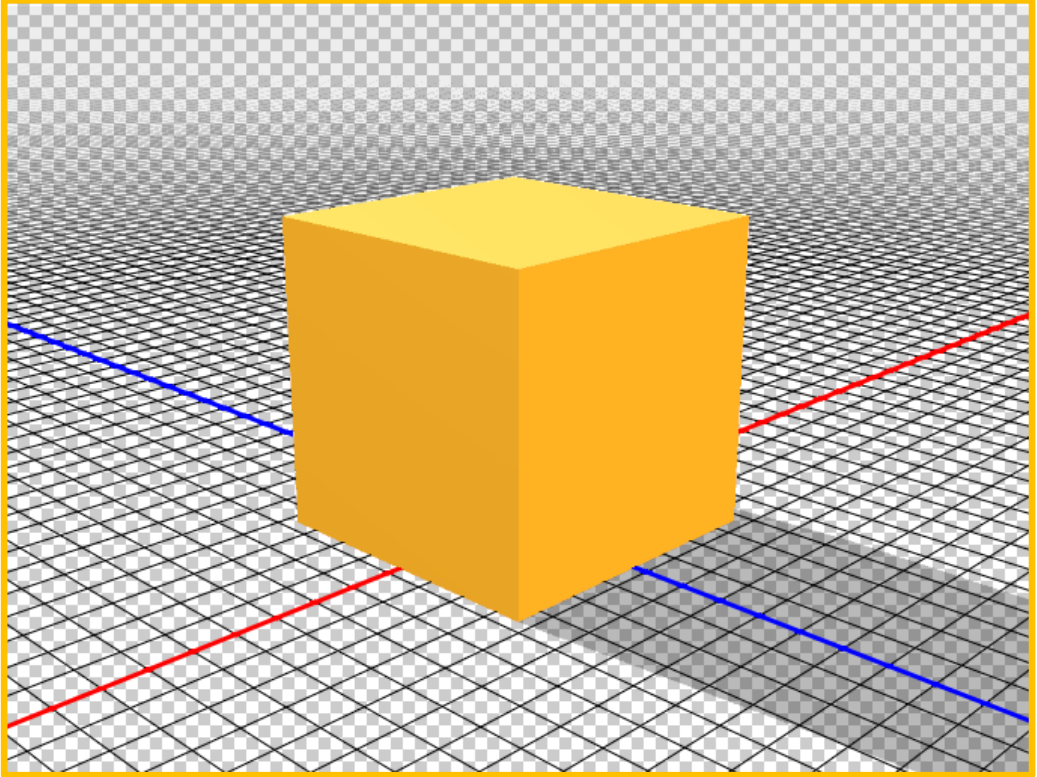
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تم عمل أكّده مع تأييد تحول في نفسه الوقت.

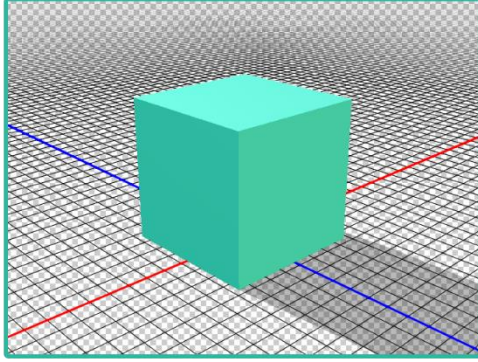
المنظور perspective والاستخدام الثلاثي الأبعاد 3D

خاصية المنظور perspective، يتم ضبطها لحاوية بها بعض العناصر، وعند ضبطها لعنصر معيه تؤثر على العناصر بداخل هذا العنصر، وتعمل لهم إمكانية العمل الثلاثي الأبعاد، وبمك أن تتخذ فكة المنظور عن طريق الشكل التالي:

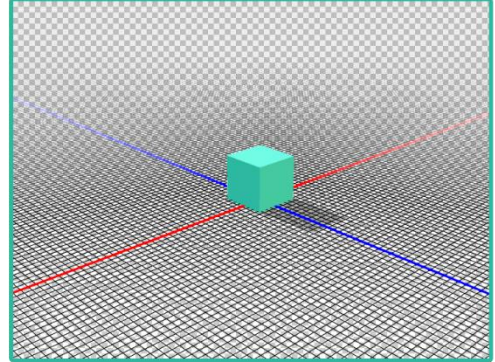


لاحظ أن الشكل الثلاثي الأبعاد الموجود في الصورة يجعلك تتخذ ما هو المنظور.

رجاء **انتبه!** فإن البعد للمنظور يختلف أيضاً، ولو تم عرضه عليك لظننت أنه zoom، ولكنه ليس كذلك، ويمكّنك أن نعبّر عنه كما يلي:

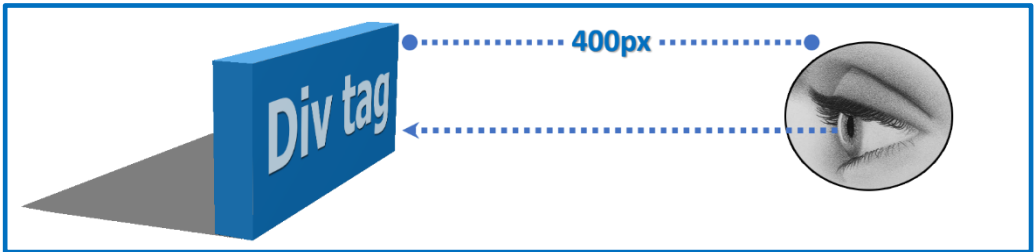


رؤية المنظور بمقدار صغير
(نراه منه مسافة قريبة)



رؤية المنظور بمقدار كبير
(نراه منه مسافة بعيدة)

وعند ضبط خاصية المنظور على مقدار، وليكن مثلاً 400px، فإن هذه القيمة تعتبر هي المسافة بين الرؤية والشكل، كما يوضح الشكل التالي:



وكما زاد مقدار المنظور، أصبح لديك تحكّم أكثر في الأبعاد، وفيما يلي سوف نقوم بعمل شكل ثلاثي الأبعاد عن طريق rotateX و rotateY ومعهم المنظور perspective، فرجاء **انتبه!** لما يلي جيداً.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{ padding:70px; }

  #main1, #main2{
    width:300px; height:200px; margin:10px;
    border:3px solid navy;
  }

  #d1, #d2{
    width:150px; height:120px; margin:50px;
    background:lightblue; border:1px solid #000;
    font-size:30px; transform: rotateX(50deg);
  }

  #main1{ perspective:200px; }

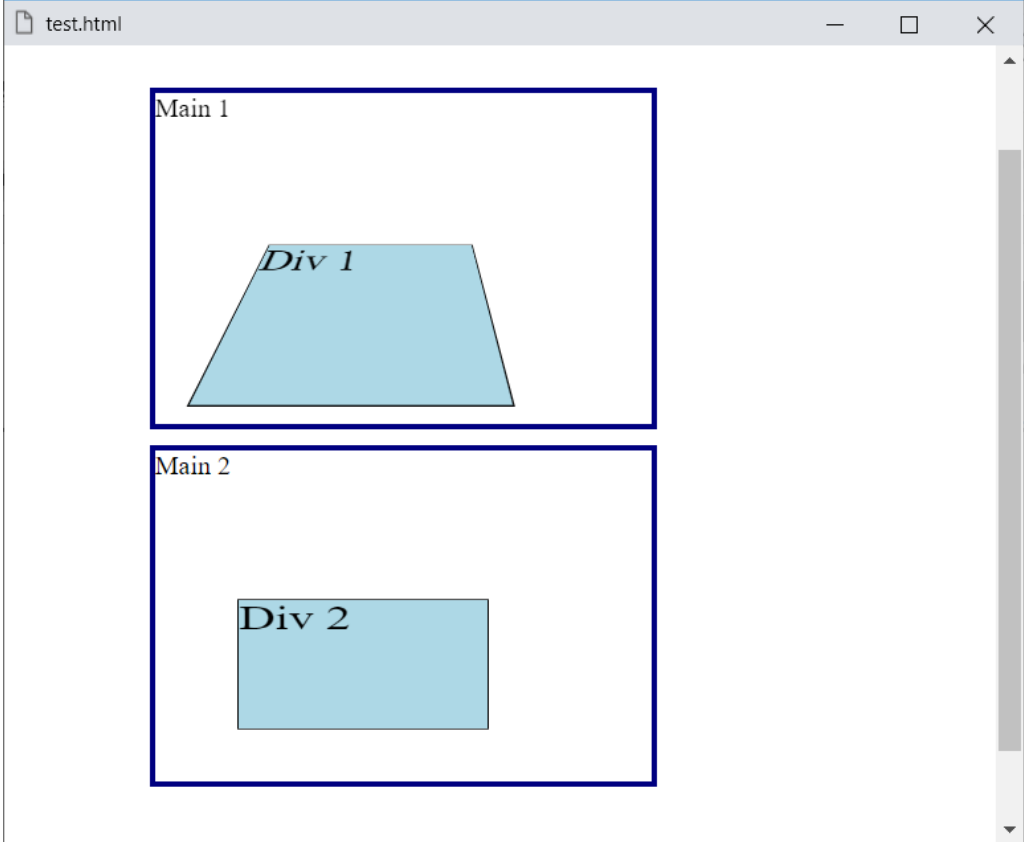
</style>
</head>
<body>

<div id="main1"> Main 1
  <div id="d1">Div 1</div>
</div>

<div id="main2"> Main 2
  <div id="d2">Div 2</div>
</div>

</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه الـ Div 1 والـ Div 2 أخذوا نفس التنسيق، ولكنه كل واحدة منهم في حاوية مختلفة، ولاحظ أنه الحاوية Main 1 تم ضبط خاصية perspective لها على القيمة 200px، فأصبح البعد لمنظورها بمقدار 200px، ولأن عنصر الـ Div 1 بداخلها أصبح الـ rotateX له مختلف تماماً عنه الـ Div 2.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>

body{ padding:70px; }

#main1, #main2{
width:300px; height:200px; margin:10px;
border:3px solid navy;
}

#d1, #d2{
width:150px; height:120px; margin:50px;
background:lightblue; border:1px solid #000;
font-size:30px; transform: rotateY(35deg);
}

#main1{ perspective:200px; }

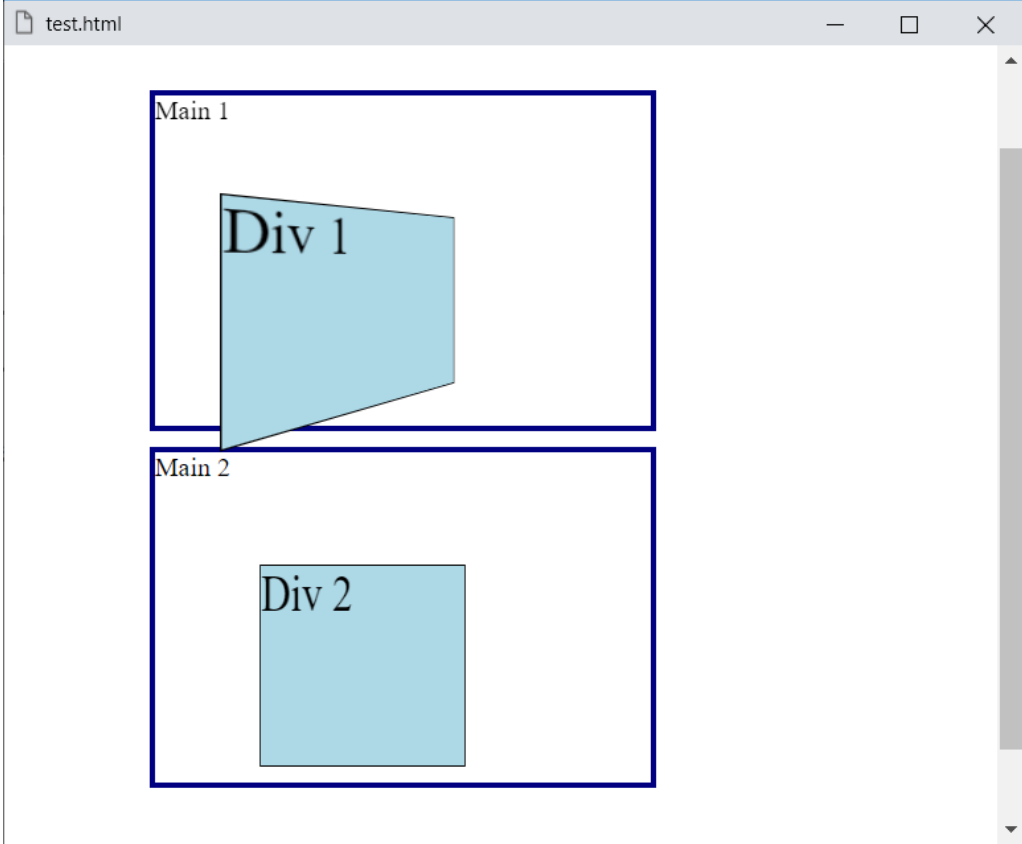
</style>
</head>
<body>

<div id="main1"> Main 1
<div id="d1">Div 1</div>
</div>

<div id="main2"> Main 2
<div id="d2">Div 2</div>
</div>

</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



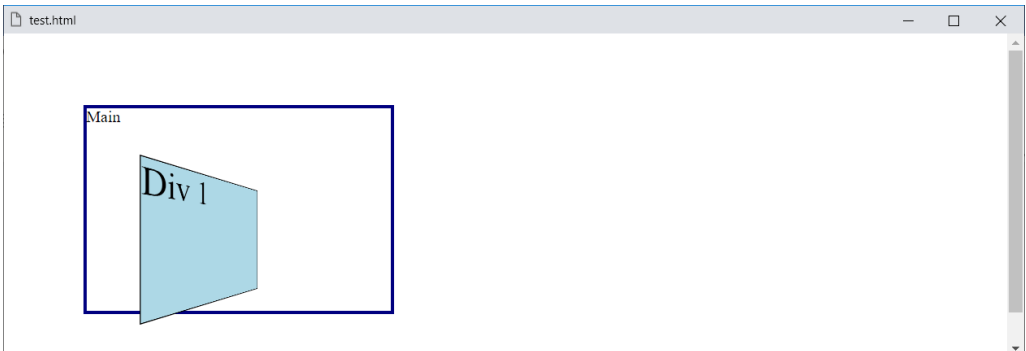
وفي هذا المثال لاحظ أن الـ Div 1 والـ Div 2 أيضاً أخذوا نفس النسبية، ولكل واحد منهم في حاوية مختلفة، ولاحظ أن الحاوية Main 1 تم ضبط خاصية perspective لها على القيمة 200px، فأصبح البعد لمنظورها بمقدار 200px، ولكل لأن عنصر الـ Div 1 بداخلها أصبح الـ rotateY له مختلف تماماً عن العنصر الـ Div 2.

عمل منظور بالدالة perspective

يمكن عمل المنظور باستخدام دالة perspective كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
.main{
width:300px; height:200px; margin:70px;
border:3px solid navy;
}
#d1{
width:150px; height:120px; margin:50px;
font-size:30px;
background:lightblue; border:1px solid #000;
transform: perspective(200px) rotateY(45deg);
}
</style>
</head>
<body>
<div class="main"> Main <div id="d1">Div 1</div> </div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

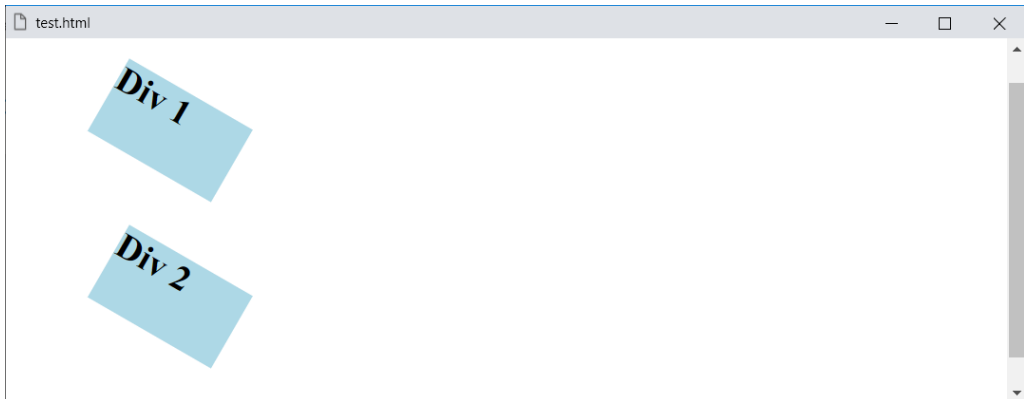


تحول الاستدارة باستخدام rotateZ

سوف يتم عمل وسميه div في المثال التالي، ولكنه سيتم الاستدارة عنه طريق rotate، وأخري عنه طريق rotateZ، وسنري أنهم مثل بعض، وسنجد **بالكود!** التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
div{
background:lightblue; width:120px; height:70px;
margin:70px; font-size:30px; font-weight:bold;
}
#d1{ transform: rotate(30deg) }
#d2{ transform: rotateZ(30deg) }
</style>
</head>
<body>
<div id="d1">Div 1</div>
<div id="d2">Div 2</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

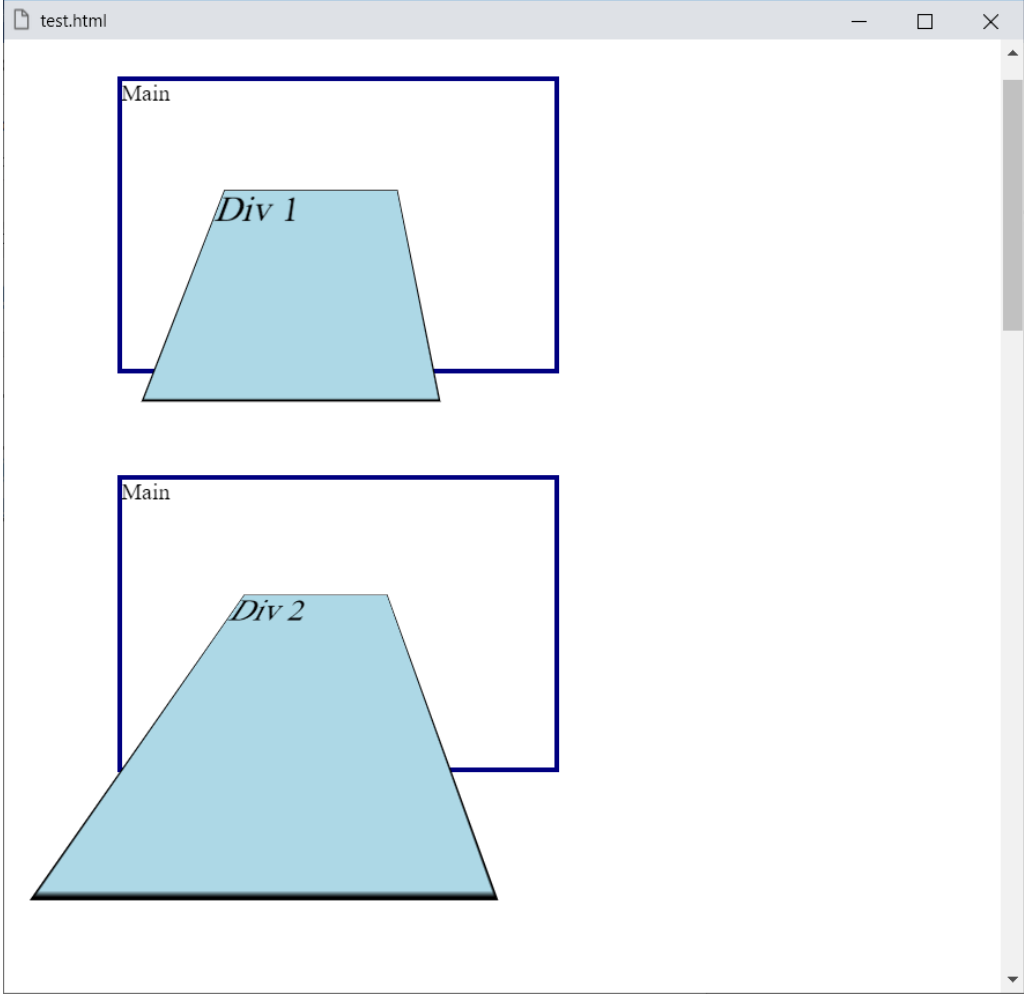


التمدد الجسم scaleZ

هذا التمدد يكون تمدد بالمنظور، أي أماما وخلفي، وستشعر أننا نقوم بعمل zoom، ولكنه تمدد مجسم للداخل والخارج، سوف يتم عمل مثال به وسمي div يحتويان على منظور perspective للعمل، وبداخل كل وسم منهم div أخرى، واحدة سيتم ضبط ال scaleZ لها على 5، والأخرى سيتم ضبط ال scaleZ لها على 10، لتلاحظ الفرق، ولكنه **انتبه!** فإن ال scaleZ ليظهر يحتاج الي عمل شيء، يجسمه، مثل rotateX أو rotateY، وسنجد **بالكود!** التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
.main{
width:300px; height:200px; margin:70px;
border:3px solid navy; perspective:200px;
}
#d1, #d2{
width:150px; height:120px; margin:50px;
font-size:30px;
background:lightblue; border:1px solid #000;
}
#d1{ transform: scaleZ(5) rotateX(10deg); }
#d2{ transform: scaleZ(10) rotateX(10deg); }
</style>
</head>
<body>
<div class="main"> Main <div id="d1">Div 1</div> </div>
<div class="main"> Main <div id="d2">Div 2</div> </div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



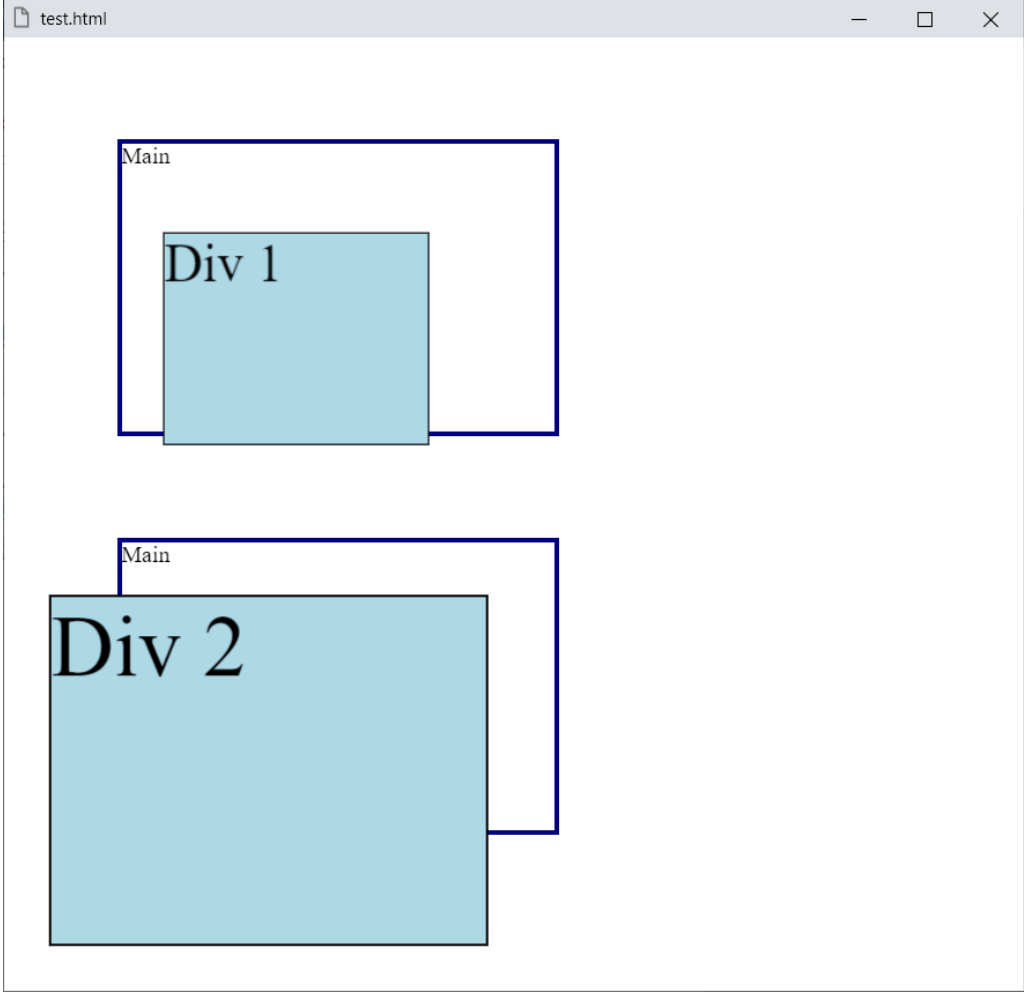
لاحظ أن الجزء الأكبر هو الذي في الأسفل، ولو تم عمله بدرجة بالسالب سيكُون العكس، ويصبح الجزء الأكبر هو الذي بالأعلى.

تدريك مجسم translateZ

ومع التدريك المجسم، سوف تشعر بال zoom فعلاً، لأنه يتحرك أيضاً
بالداخل والخارج، وسنجد بال **الكود!** التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
.main{
width:300px; height:200px; margin:70px;
border:3px solid navy; perspective:200px;
}
#d1, #d2{
width:150px; height:120px; margin:50px;
font-size:30px;
background:lightblue; border:1px solid #000;
}
#d1{ transform: translateZ(35px); }
#d2{ transform: translateZ(100px); }
</style>
</head>
<body>
<div class="main"> Main <div id="d1">Div 1</div> </div>
<div class="main"> Main <div id="d2">Div 2</div> </div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه كلما زادت قيمة `translateZ` كلما اقترب العنصر أكثر من العبد، وكلما قلت القيمة ابتعد العنصر عن العبد.

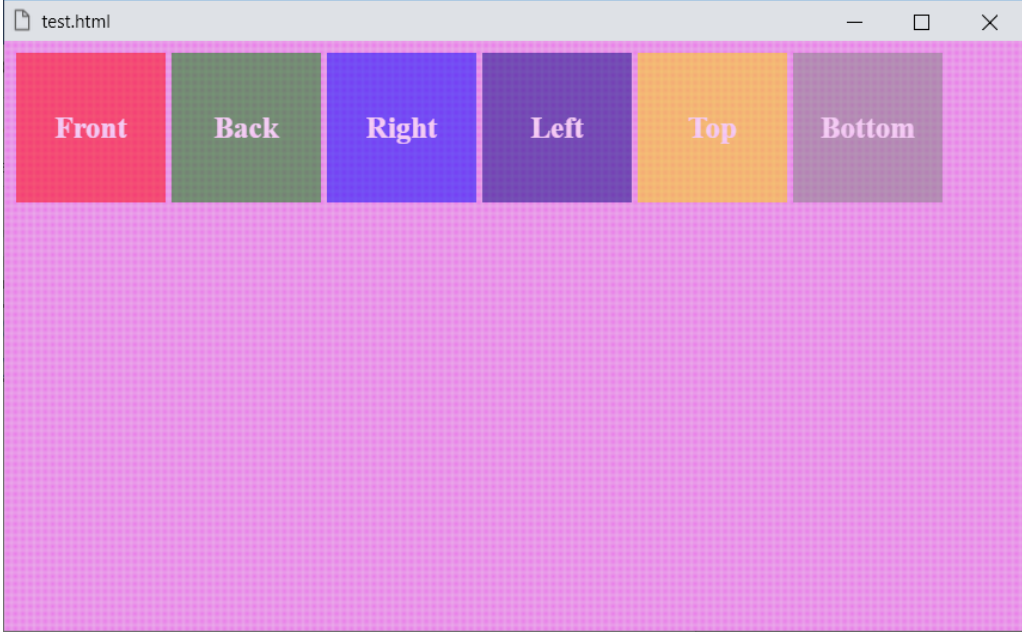
تكويد شكل مكعب ثلاثي الأبعاد Cube

بداية تكويد المكعب

المكعب يحتوي على 6 مربعات (أمامي، خلفي، علوي، سفلي، يميني، يساري)،
والآن نريد عملهم، مع حجم ولون وشفافية وظهور inline-block
وخط ومحاذاة، وسوف يتم عمل ذلك **بالكود!** التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
body{ background:radial-gradient(#ea7cea, #e8aee8) left/5px 5px; }
.cube div{
width:100px; height:100px; opacity:0.5; display:inline-block;
line-height:100px; font-size:20px; font-weight:bold;
text-align:center; color:#fff;
}
.cube .front{ background:red; }
.cube .back{ background:green; }
.cube .right{ background:blue; }
.cube .left{ background:navy; }
.cube .top{ background:gold; }
.cube .bottom{ background:gray; }
</style>
</head>
<body>
<div class="cube">
<div class="front">Front</div>
<div class="back">Back</div>
<div class="right">Right</div>
<div class="left">Left</div>
<div class="top">Top</div>
<div class="bottom">Bottom</div>
</div>
</body>
</html>
```

وبءء ءشءعل الصءفة ءظهء كما بالشءك ءءل:



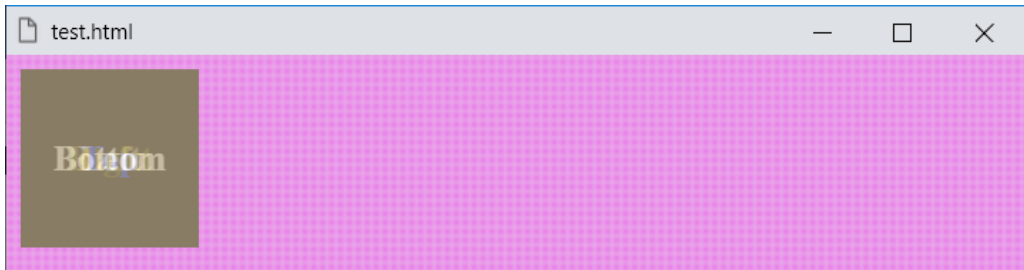
لاءظ أن شفافة العناصء، ولون ءلفية الءسم `body` بشءك مزءرف
أشفاء مءعمءة، للظهءر ففما بءء.

منظّم الوضوح المطلق absolute

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
body{ background:radial-gradient(#ea7cea, #e8aee8) left/5px 5px; }
.cube div{
width:100px; height:100px; opacity:0.5; display:inline-block;
line-height:100px; font-size:20px; font-weight:bold;
text-align:center; color:#fff; position:absolute;
}
.cube .front{ background:red; }
.cube .back{ background:green; }
.cube .right{ background:blue; }
.cube .left{ background:navy; }
.cube .top{ background:gold; }
.cube .bottom{ background:gray; }
</style>
</head>
<body>
<div class="cube">
<div class="front">Front</div>
<div class="back">Back</div>
<div class="right">Right</div>
<div class="left">Left</div>
<div class="top">Top</div>
<div class="bottom">Bottom</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

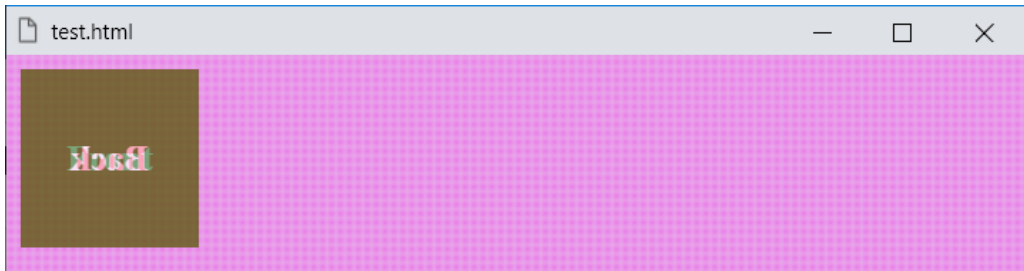


ضبط الزوايا المناسبة

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
body{ background:radial-gradient(#ea7cea, #e8aee8) left/5px 5px; }
.cube div{
width:100px; height:100px; opacity:0.5; display:inline-block;
line-height:100px; font-size:20px; font-weight:bold;
text-align:center; color:#fff; position:absolute;
}
.cube .front{ background:red; transform:rotateY(0deg); }
.cube .back{ background:green; transform:rotateY(180deg); }
.cube .right{ background:blue; transform:rotateY(90deg); }
.cube .left{ background:navy; transform:rotateY(-90deg); }
.cube .top{ background:gold; transform:rotateX(90deg); }
.cube .bottom{ background:gray; transform:rotateX(-90deg); }
</style>
</head>
<body>
<div class="cube">
<div class="front">Front</div>
<div class="back">Back</div>
<div class="right">Right</div>
<div class="left">Left</div>
<div class="top">Top</div>
<div class="bottom">Bottom</div>
</div>
</body>
</html>
```

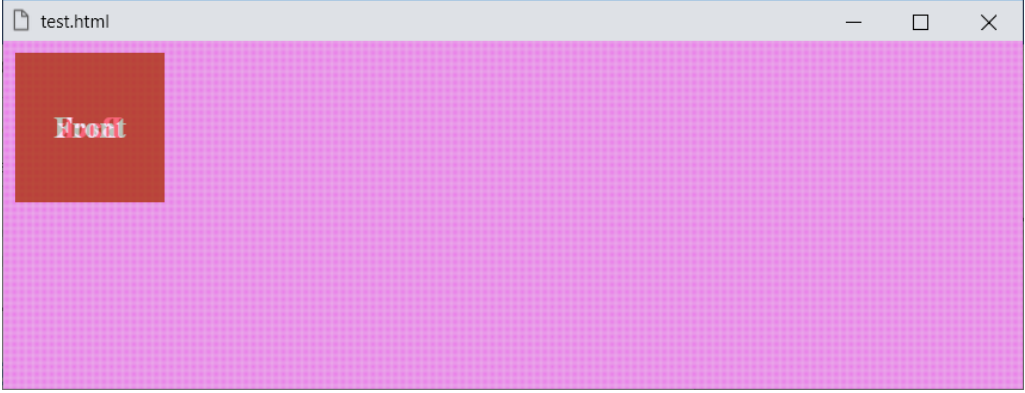
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



زيادة المشهد وضبط المنظور و transform-style على 3D كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
body{ background:radial-gradient(#ea7cea, #e8aee8) left/5px 5px; }
.view{ width:100px; height:100px; perspective:300px; }
.cube{ width:100px; height:100px; transform-style:preserve-3d; }
.cube div{
width:100px; height:100px; opacity:0.5; display:inline-block;
line-height:100px; font-size:20px; font-weight:bold;
text-align:center; color:#fff; position:absolute;
}
.cube .front{ background:red; transform:rotateY(0deg); }
.cube .back{ background:green; transform:rotateY(180deg); }
.cube .right{ background:blue; transform:rotateY(90deg); }
.cube .left{ background:navy; transform:rotateY(-90deg); }
.cube .top{ background:gold; transform:rotateX(90deg); }
.cube .bottom{ background:gray; transform:rotateX(-90deg); }
</style>
</head>
<body>
<div class="view">
<div class="cube">
<div class="front">Front</div>
<div class="back">Back</div>
<div class="right">Right</div>
<div class="left">Left</div>
<div class="top">Top</div>
<div class="bottom">Bottom</div>
</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه في الكود السابق تم زيادة وسم div بكلاس باسم view يحمل المكعب cube بداخلة، وتم ضبط المنظور له، كما تم ضبط الخاصية transform-style على القيمة preserve-3d والحجم للمكعب نفسه.

منبظ الإطار لمكونات المكعب الداخلية
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
body{ background:radial-gradient(#ea7cea, #e8aee8) left/5px 5px; }
.view{ width:100px; height:100px; perspective:300px; }
.cube{ width:100px; height:100px; transform-style:preserve-3d; }
.cube div{
width:100px; height:100px; opacity:0.5; display:inline-block;
line-height:100px; font-size:20px; font-weight:bold;
text-align:center; color:#fff; position:absolute;
border:2px solid black;
}
.cube .front{ background:red; transform:rotateY(0deg); }
.cube .back{ background:green; transform:rotateY(180deg); }
.cube .right{ background:blue; transform:rotateY(90deg); }
.cube .left{ background:navy; transform:rotateY(-90deg); }
.cube .top{ background:gold; transform:rotateX(90deg); }
.cube .bottom{ background:gray; transform:rotateX(-90deg); }
</style>
</head>
<body>
<div class="view">
<div class="cube">
<div class="front">Front</div>
<div class="back">Back</div>
<div class="right">Right</div>
<div class="left">Left</div>
<div class="top">Top</div>
<div class="bottom">Bottom</div>
</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

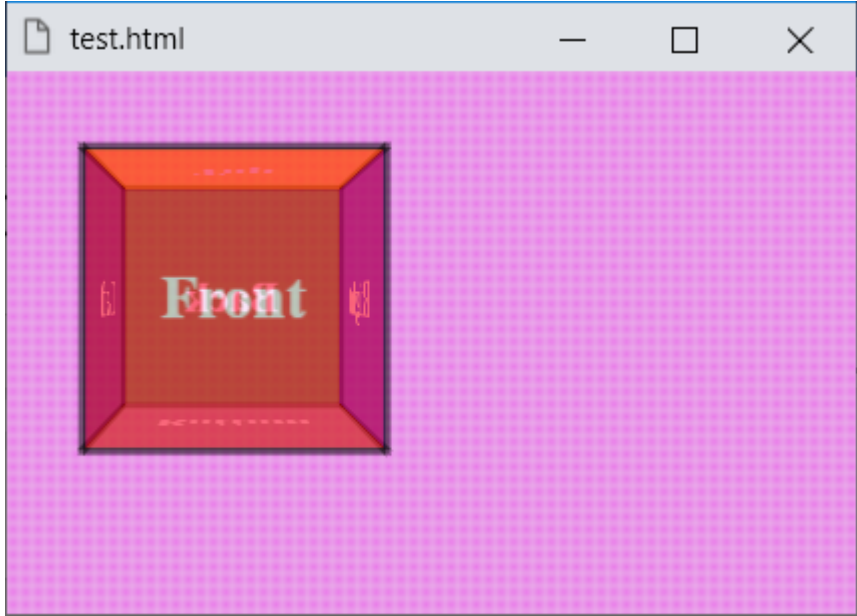


لاحظ أن الإطار ظهر بشكل متداخل.

عمل تحريك مجسم للـك بنصف قيمة كل مربع
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
body{ background:radial-gradient(#ea7cea, #e8aee8) left/5px 5px; }
.view{ width:100px; height:100px; perspective:300px; padding:30px; }
.cube{ width:100px; height:100px; transform-style:preserve-3d; }
.cube div{
width:100px; height:100px; opacity:0.5; display:inline-block;
line-height:100px; font-size:20px; font-weight:bold;
text-align:center; color:#fff; position:absolute;
border:2px solid black;
}
.cube .front{background:red;transform:rotateY(0deg) translateZ(50px);}
.cube .back{background:green;transform:rotateY(180deg) translateZ(50px);}
.cube .right{background:blue;transform:rotateY(90deg) translateZ(50px);}
.cube .left{background:navy;transform:rotateY(-90deg) translateZ(50px);}
.cube .top{background:gold;transform:rotateX(90deg) translateZ(50px);}
.cube .bottom{background:gray;transform:rotateX(-90deg) translateZ(50px);}
</style>
</head>
<body>
<div class="view">
<div class="cube">
<div class="front">Front</div>
<div class="back">Back</div>
<div class="right">Right</div>
<div class="left">Left</div>
<div class="top">Top</div>
<div class="bottom">Bottom</div>
</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

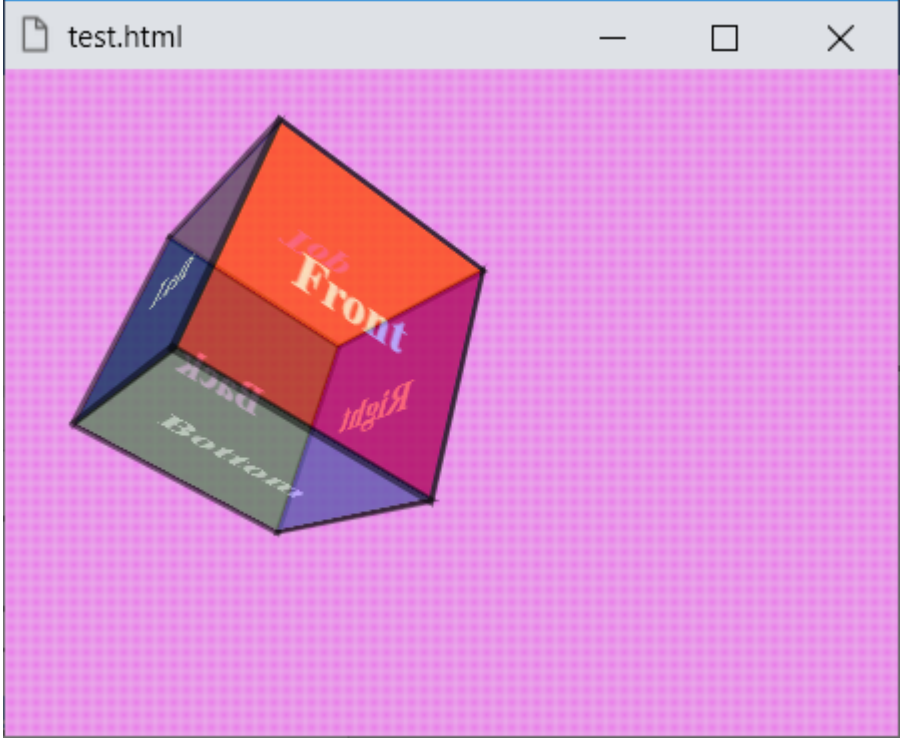


لاحظ أنه يظهر الآن وكأنه مكعب شفاف.

عمل اسطوانة ثلاثية الأبعاد rotate3d كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
body{ background:radial-gradient(#ea7cea, #e8aee8) left/5px 5px; }
.view{ width:100px; height:100px; perspective:300px; padding:50px; }
.cube{
width:100px; height:100px; transform-style:preserve-3d;
transform: rotate3d(1, 1, 1, 45deg);
}
.cube div{
width:100px; height:100px; opacity:0.5; display:inline-block;
line-height:100px; font-size:20px; font-weight:bold;
text-align:center; color:#fff; position:absolute;
border:2px solid black;
}
.cube .front{background:red;transform:rotateY(0deg) translateZ(50px);}
.cube .back{background:green;transform:rotateY(180deg) translateZ(50px);}
.cube .right{background:blue;transform:rotateY(90deg) translateZ(50px);}
.cube .left{background:navy;transform:rotateY(-90deg) translateZ(50px);}
.cube .top{background:gold;transform:rotateX(90deg) translateZ(50px);}
.cube .bottom{background:gray;transform:rotateX(-90deg) translateZ(50px);}
</style>
</head>
<body>
<div class="view">
<div class="cube">
<div class="front">Front</div>
<div class="back">Back</div>
<div class="right">Right</div>
<div class="left">Left</div>
<div class="top">Top</div>
<div class="bottom">Bottom</div>
</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

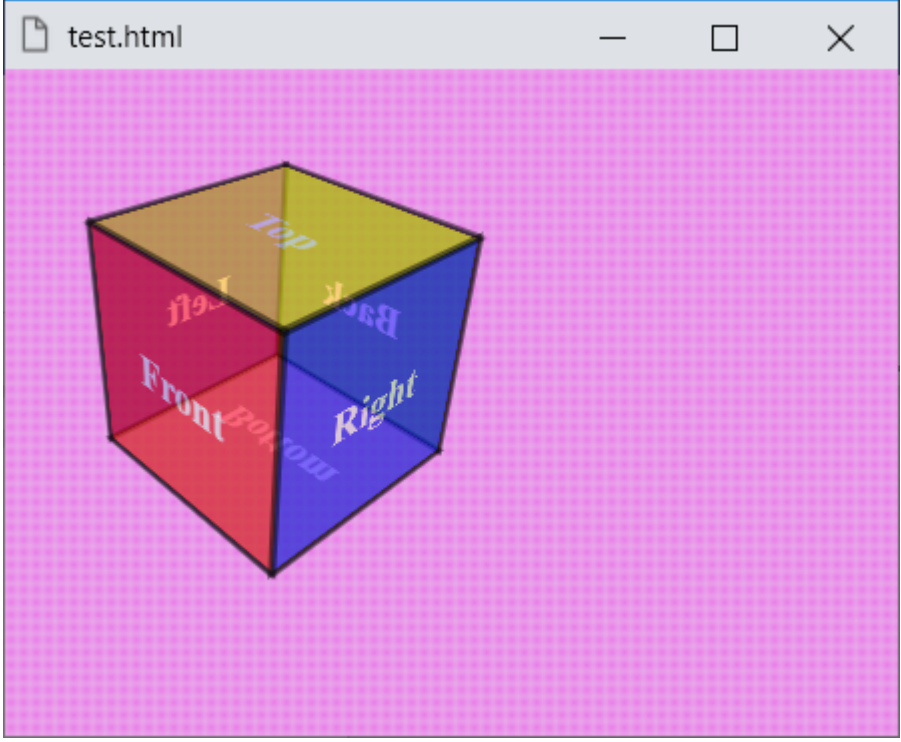


لاحظ أنه تم عمل rotate للمكعب، أي أنها استدارة ثلاثية الأبعاد باستخدام الدالة rotate3d، والدالة rotate3d تأخذ 4 قيم، وهم الـ x، والـ y، والـ z، والدرجة، فيتم عمل استدارة كاملة للمكعب كشكل واحد ثلاثي الأبعاد كما هو واضح في الشكل السابق.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
body{ background:radial-gradient(#ea7cea, #e8aee8) left/5px 5px; }
.view{ width:100px; height:100px; perspective:300px; padding:50px; }
.cube{
width:100px; height:100px; transform-style:preserve-3d;
transform: rotate3d(2, 3, -1, -55deg);
}
.cube div{
width:100px; height:100px; opacity:0.5; display:inline-block;
line-height:100px; font-size:20px; font-weight:bold;
text-align:center; color:#fff; position:absolute;
border:2px solid black;
}
.cube .front{background:red;transform:rotateY(0deg) translateZ(50px);}
.cube .back{background:green;transform:rotateY(180deg) translateZ(50px);}
.cube .right{background:blue;transform:rotateY(90deg) translateZ(50px);}
.cube .left{background:navy;transform:rotateY(-90deg) translateZ(50px);}
.cube .top{background:gold;transform:rotateX(90deg) translateZ(50px);}
.cube .bottom{background:gray;transform:rotateX(-90deg) translateZ(50px);}
</style>
</head>
<body>
<div class="view">
<div class="cube">
<div class="front">Front</div>
<div class="back">Back</div>
<div class="right">Right</div>
<div class="left">Left</div>
<div class="top">Top</div>
<div class="bottom">Bottom</div>
</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أنه تمت الرؤية منه اعلى.

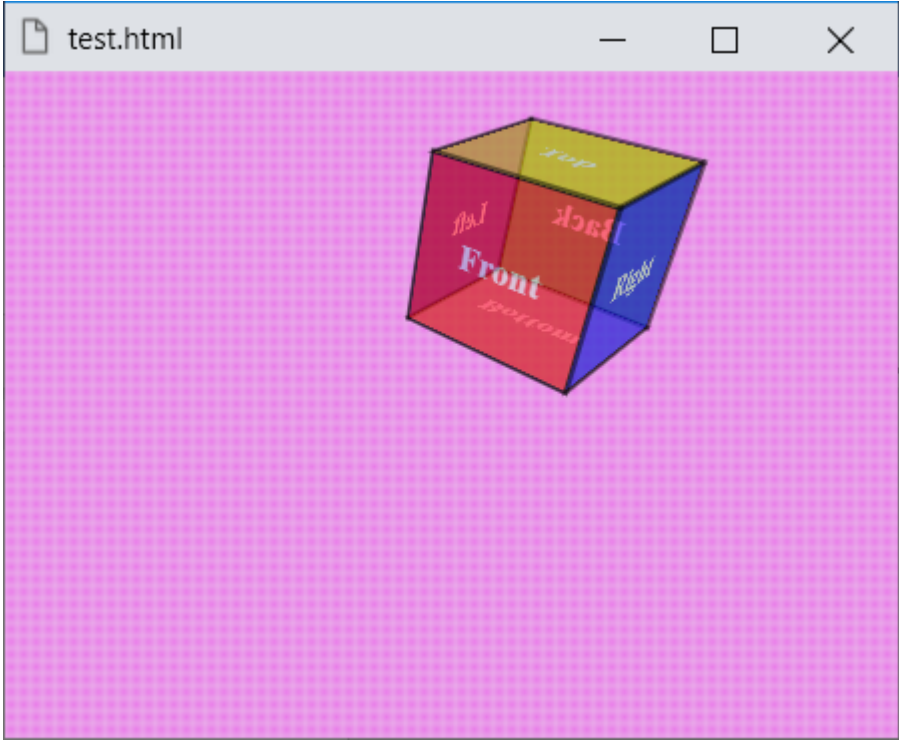
انتبه! فلما تم تغيير ال x ، وال y ، وال z ، والدرجة، تمت الاستدارة بشكل مختلف، ورؤية مختلفة أيضاً.

تحريك مجسم باستخدام translate3d

هذه الدالة تعيد اختصار للحركات X و Y و Z معاً.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
body{ background:radial-gradient(#ea7cea, #e8aee8) left/5px 5px; }
.view{ width:100px; height:100px; perspective:300px; padding:50px; }
.cube{
width:100px; height:100px; transform-style:preserve-3d;
transform: rotate3d(2, 3, -1, -55deg) translate3d(10px,10px,-200px) ;
}
.cube div{
width:100px; height:100px; opacity:0.5; display:inline-block;
line-height:100px; font-size:20px; font-weight:bold;
text-align:center; color:#fff; position:absolute;
border:2px solid black;
}
.cube .front{background:red;transform:rotateY(0deg) translateZ(50px);}
.cube .back{background:green;transform:rotateY(180deg) translateZ(50px);}
.cube .right{background:blue;transform:rotateY(90deg) translateZ(50px);}
.cube .left{background:navy;transform:rotateY(-90deg) translateZ(50px);}
.cube .top{background:gold;transform:rotateX(90deg) translateZ(50px);}
.cube .bottom{background:gray;transform:rotateX(-90deg) translateZ(50px);}
</style>
</head>
<body>
<div class="view">
<div class="cube">
<div class="front">Front</div>
<div class="back">Back</div>
<div class="right">Right</div>
<div class="left">Left</div>
<div class="top">Top</div>
<div class="bottom">Bottom</div>
</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



الخاصية `translate3d` تأخذ x ثم y ثم z ، ولاحظ أننا الآن نرى المكعب منه بعيد، لأن الـ z أخذت القيمة $-200px$ كما هو موضح في الكود السابق، وكلما زادت اقتربت، وكلما قلت ابتعدت.

تمدد مجسم باستخدام scale3d

هذه الدالة تعتبر اختصاراً للمتمدات X و Y و Z معاً.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
body{ background:radial-gradient(#ea7cea, #e8aee8) left/5px 5px; }
.view{ width:100px; height:100px; perspective:300px; padding:150px; }
.cube{
width:100px; height:100px; transform-style:preserve-3d;
transform: rotate3d(2, 3, -1, -55deg) scale3d(1,1,2.5) ;
}
.cube div{
width:100px; height:100px; opacity:0.5; display:inline-block;
line-height:100px; font-size:20px; font-weight:bold;
text-align:center; color:#fff; position:absolute;
border:2px solid yellow;
}
.cube .front{background:red;transform:rotateY(0deg) translateZ(50px);}
.cube .back{background:green;transform:rotateY(180deg) translateZ(50px);}
.cube .right{background:blue;transform:rotateY(90deg) translateZ(50px);}
.cube .left{background:navy;transform:rotateY(-90deg) translateZ(50px);}
.cube .top{background:gold;transform:rotateX(90deg) translateZ(50px);}
.cube .bottom{background:gray;transform:rotateX(-90deg) translateZ(50px);}
</style>
</head>
<body>
<div class="view">
<div class="cube">
<div class="front">Front</div>
<div class="back">Back</div>
<div class="right">Right</div>
<div class="left">Left</div>
<div class="top">Top</div>
<div class="bottom">Bottom</div>
</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



لاحظ أن التمدد المجسم، مع زيادة مه ناحية واحدة، جعله مستطيل ثلاثي الأبعاد، بدلاً مه المربع، مع العلم أن كل ناحية ستختلف في جهة الزيادة.

تحويلات الـ 3D بواسطة matrix3d

قبل البدء في هذا الموضوع يجب تتعلم أولاً كل ما تم شرحه في موضوع الـ transform حتى الآن.

انتبه! فإن الدالة matrix3d مجرد اختصار للدوال rotate3d(), rotateX(), rotateY(), rotateZ(), translate3d(), translateZ(), scale3d(), scaleZ(), perspective() فهي تأخذ 16 قيمة كما يلي:

- أربع قيم للدالة rotate3d 4
 - قيمة واحدة للدالة rotateX 1
 - قيمة واحدة للدالة rotateY 1
 - قيمة واحدة للدالة rotateZ 1
 - ثلاث قيم للدالة translate3d 3
 - قيمة واحدة للدالة translateZ 1
 - ثلاث قيم للدالة scale3d 3
 - قيمة واحدة للدالة scaleZ 1
 - قيمة واحدة للدالة perspective 1
- إذاً تكون $16 = 1 + 1 + 3 + 1 + 3 + 1 + 1 + 1 + 4$

انتبه! فإن هذه الدالة طويلة جداً، وسوف تحتاجها إذا استخدمت الجميع معاً، ولذلك فله يتم عمل أمثلة بها في هذا الكتاب.

الانتقال Transition

المقصود بالانتقال هنا، هو انتقال العنصر من وضع الى وضع آخر، مثل أن يتم تغيير العرض، أو أن يتم تغيير الطول، أو أن يتم تغيير لون الخلفية، أو مثل أن يتم زيادة او نقصان الحشو padding، الى آخر هذه الأمور التي تخص العنصر، ولكنه **انتبه!** فإن هناك بعض الخصائص تستجيب الى تأثيرات الانتقال أي أنها Animatable، وأخرى لا تستجيب، ولكنه **انتبه!** فإن الهدف من الانتقال transition هو التحكم أثناء الانتقال، فتم تأثيرات جميلة بهذا التحكم الرائع، ولنتعلم كيفية التحكم في الانتقال يجب معرفة ما يلي جيداً:

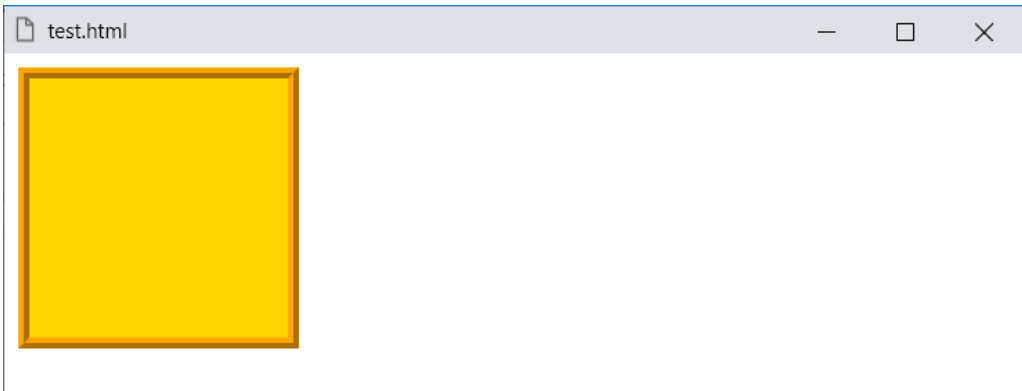
- 1- خاصية الانتقال transition-property، وهي تأخذ قيمة باسم الخاصية التي سوف يتم التحكم فيها مثل width او height او padding-right.
- 2- مدة الانتقال transition-duration، وهي تأخذ قيمة للتعبير عن الوقت المستغرق أثناء عملية الانتقال، وهذه القيمة رقم ثم حرف s للثانية، مثل 3s أي ثلاث ثواني، أو 0.5s أي نصف ثانية، أو 500ms أي نصف ثانية أيضاً لأن ms للميلي ثانية.
- 3- توظيف الوقت transition-timing-function، وهي تأخذ بعض القيم للتحكم في وقت الانتقال.
- 4- تأخير الانتقال transition-delay، وهي تأخذ قيمة للتأخير، مثل 1s او 3s، لتتأخر قبل الانتقال.

الخاصية والمدة والتأخير مع ال transition

والآن قبل تجربة أكواد ال transition، سنجرب ما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
#d1{
width:150px; height:150px; background:gold;
border:6px ridge orange;
}
#d1:hover{
width:300px;
}
</style>
</head>
<body>
<div id="d1"></div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



والآن عند لمس عنصر div سيكون مثل الشكل التالي:

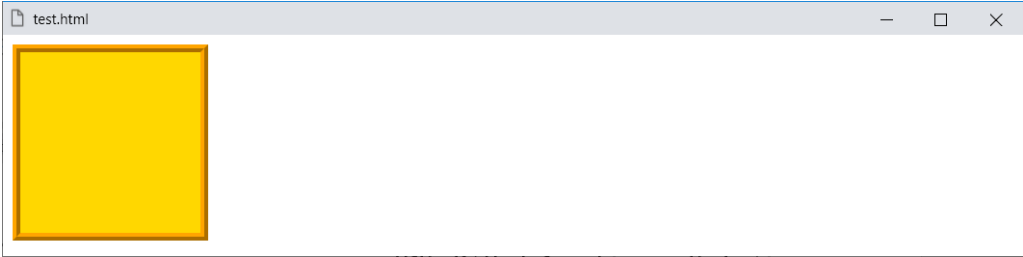


ولكن التغير في العرض width يكون لحظي، ولا يستغرق أي وقت.

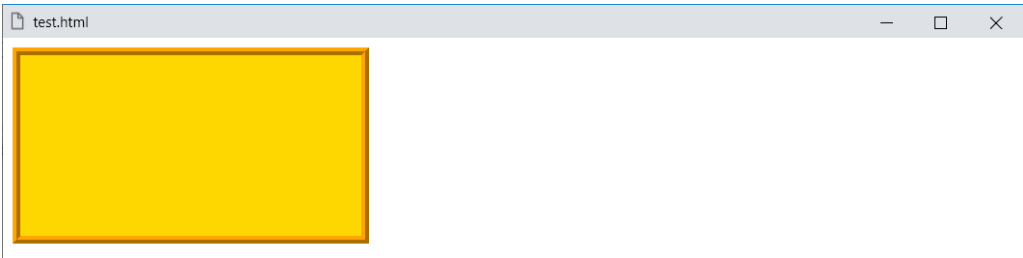
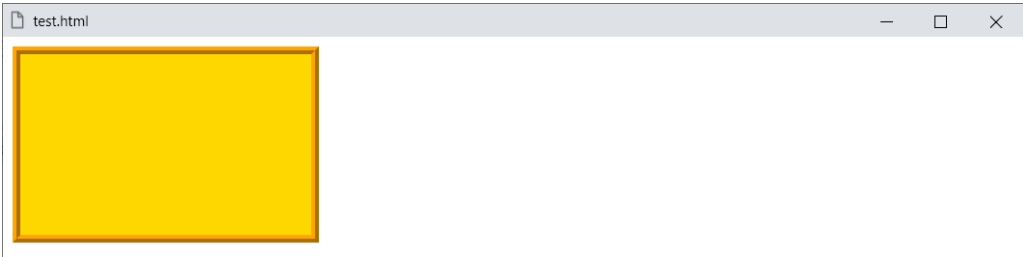
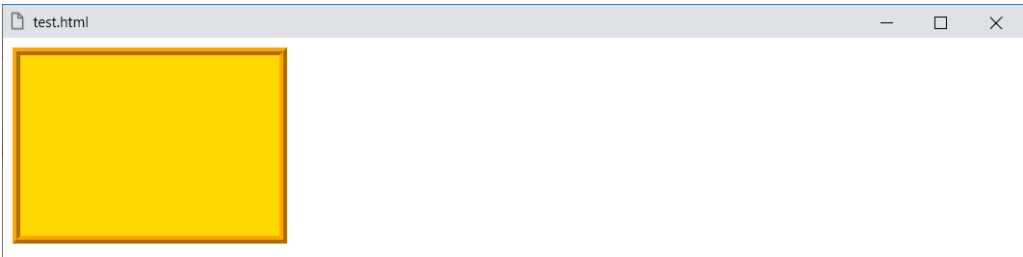
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
#d1{
width:150px; height:150px; background:gold;
border:6px ridge orange;
transition-property:width;
transition-duration:5s;
}
#d1:hover{
width:300px;
}
</style>
</head>
<body>
<div id="d1"></div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



والآن قم بلمس العنصر، وانتظر، ولاحظ الأشكال التالية:

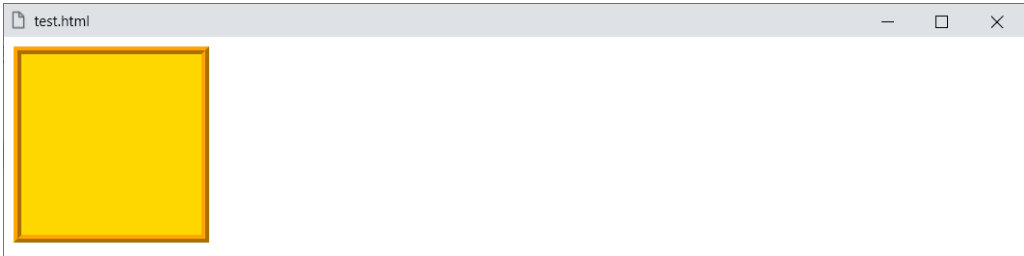


تم التغيير في خمس ثواني لزيادة العرض من 150px الى 300px، لأنه تم ضبط transition-property بـ width وتم ضبط transition-duration بـ 5s، داخل تنسيق الومسم الأصلي.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
#d1{
width:150px; height:150px; background:gold;
border:6px ridge orange;
transition-property:width;
transition-duration:5s;
transition-delay:2s;
}
#d1:hover{
width:300px;
}
</style>
</head>
<body>
<div id="d1"></div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

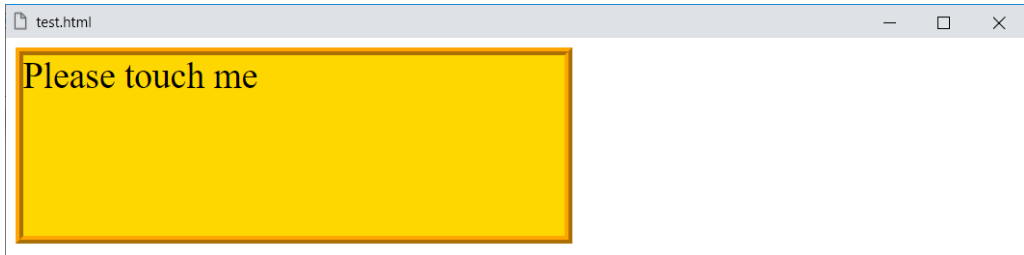


والآن قم بلمس العنصر div، وستلاحظ أنه سوف يتأخر ثانيته قبل التغير، ثم يتغير في خمس ثواني، والآن **انتبه!** فعند الابتعاد عنه ستلاحظ أنه سوف يتأخر ثانيته قبل التغير، ثم يتغير في خمس ثواني أيضاً.

كود! قم بكتابة الكود التالي ثم احفظه:

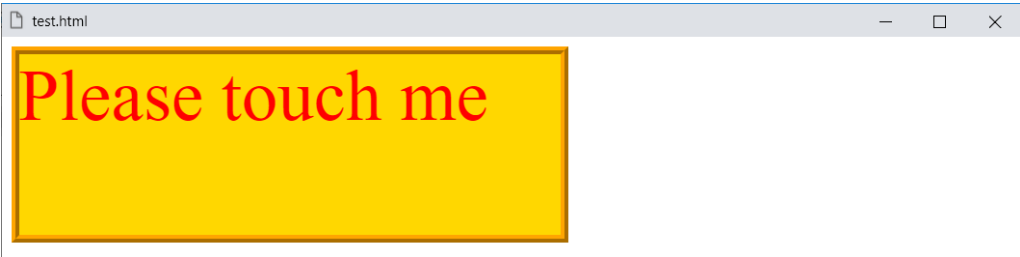
```
<!DOCTYPE html>
<html>
<head>
<style>
#d1{
width:450px; height:150px; background:gold;
border:6px ridge orange; font-size:30px;
transition-property:font-size , color;
transition-duration:5s;
}
#d1:hover{
font-size:60px;
color:red;
}
</style>
</head>
<body>
<div id="d1">Please touch me</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



والآن لاحظ أنه تم تأثير الانتقال على خاصيتيه مع طريق فاصلة، والخاصيتيه هما حجم الخط font-size ولونه النص color، حيث يتم تغيير الخط من 30px الى 60px، وتم تغيير اللون الى الأحمر.

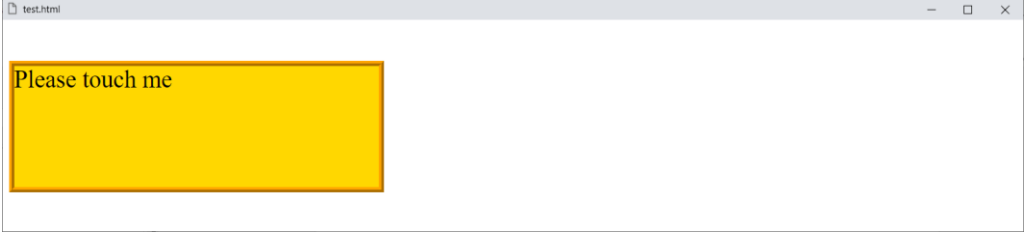
والآن قم بلمس العنصر div وانتظر ولاحظ الأشكال التالية:



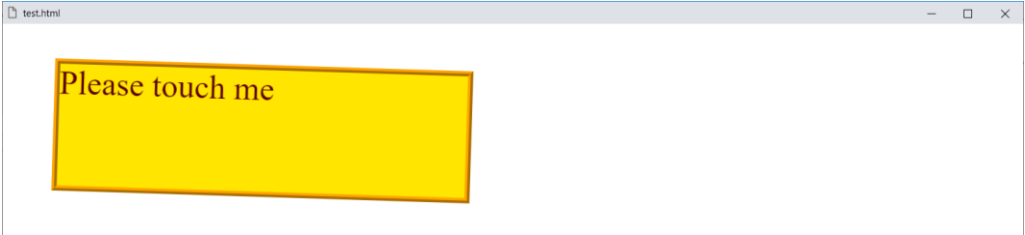
والآن، ماذا مع تغييرات في خصائص كثيرة؟!، والجواب سهل بفضل الله، وهو تعيين القيمة all للخاصية transition-property، وتعني كل الخصائص التي تدعم ال transition.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body{ margin-top:50px; }
      #d1{
        width:450px; height:150px; background:gold;
        border:6px ridge orange; font-size:30px;
        transition-property:all;
        transition-duration:5s;
      }
      #d1:hover{
        padding-right:150px; margin-left:150px;
        font-size:60px; transform:rotate(5deg);
        background-color:yellow; color:red;
      }
    </style>
  </head>
  <body>
    <div id="d1">Please touch me</div>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



والآن قم بلمس العنصر div وانتظر ولاحظ الأشكال التالية:



توظيف الوقت transition-timing-function

هذه الخاصية تستخدم لتوظيف الوقت، أي مثلاً عند بداية التحرك يكون سريع، ثم يكون بطيء، ثم عادي، أو مثل هذه الأمور، والقيمة الافتراضية لهذه الخاصية هي ease أي سهل أو عادي، وهو ما تم تجربته فيما سبق، وتأخذ أيضاً عدة قيم أخرى مثل ease-in و ease-out و ease-in-out و linear و بعض القيم الأخرى، ولو قمت بتجربة كل قيمة له تلاحظ الفرق إلا مع التدرج الشديد.

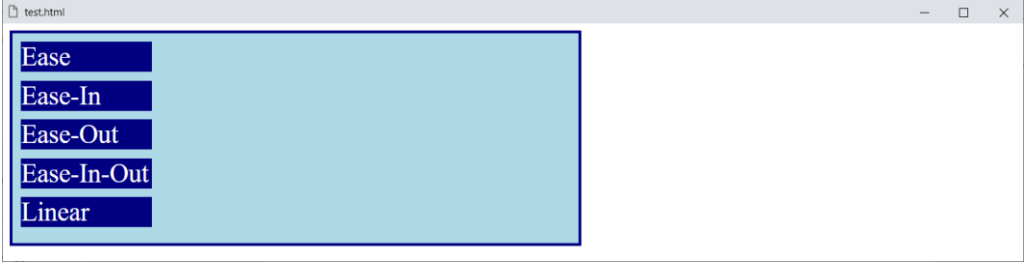
القيمة ease تجعل وقت الحركة يبدأ بطيئة، ثم يسرع ثم ينتهي ببطيء.
 القيمة ease-in تجعل وقت الحركة بطيء في البداية فقط.
 القيمة ease-out تجعل وقت الحركة بطيء في النهاية فقط.
 القيمة ease-in-out تجعل وقت الحركة بطيء في البداية والنهاية.
 القيمة linear تجعل وقت الحركة بنفس السرعة من البداية إلى النهاية.

والآن لملاحظة الفرق بين كل قيمة وأخرى، يجب الجمع بينهم والتأثير عليهم معاً في نفس الوقت، سوف يتم عمل مثال يجمع القيم ease و ease-in و ease-out و ease-in-out و linear لتلاحظ الفرق بين كل قيمة وأخرى عند التأثير مرة واحدة، والمثال إن شاء الله سيكون كما يلي.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{
    width:650px; height:240px; background:lightblue;
    border:3px solid navy; font-size:30px;
  }
  div p{
    width:150px; margin:10px; background:navy; color:#fff;
    transition-property:all;
    transition-duration:5s;
  }
  #p1{ transition-timing-function:ease; }
  #p2{ transition-timing-function:ease-in; }
  #p3{ transition-timing-function:ease-out; }
  #p4{ transition-timing-function:ease-in-out; }
  #p5{ transition-timing-function:linear; }
  div:hover p{ width:300px; }
</style>
</head>
<body>
<div>
  <p id="p1">Ease</p>
  <p id="p2">Ease-In</p>
  <p id="p3">Ease-Out</p>
  <p id="p4">Ease-In-Out</p>
  <p id="p5">Linear</p>
</div>
</body>
</html>
```

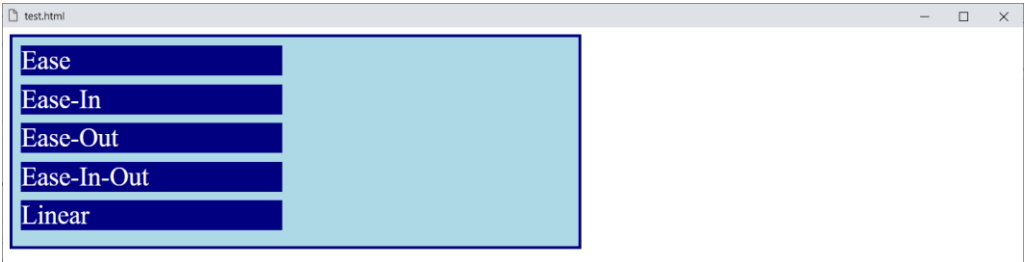
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



والآن للتجربة فقط قم بلمس الـ div ولاحظ كما يلي:



انتبه! ولاحظ اختلاف معدلات الوقت، مع ثبات قيمة الخاصية، وثبات المدة، لهم جميعاً، ولذلك في النهاية سيكون مثل الشكل التالي:

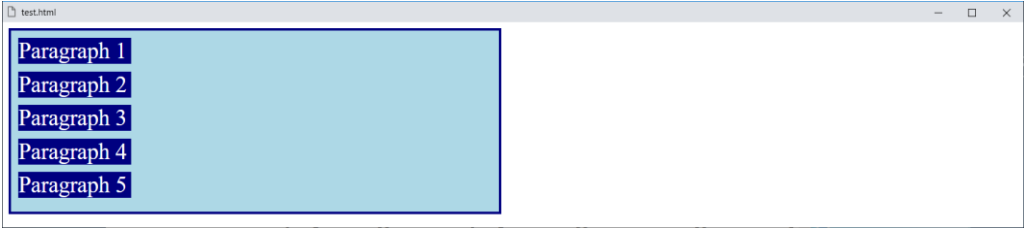


وكذلك الأمر عند ترك لمس الـ div.

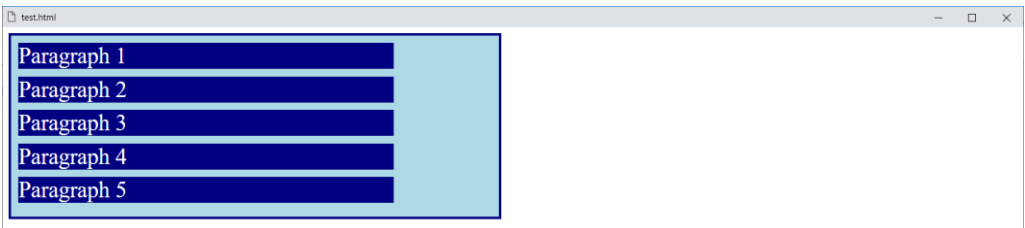
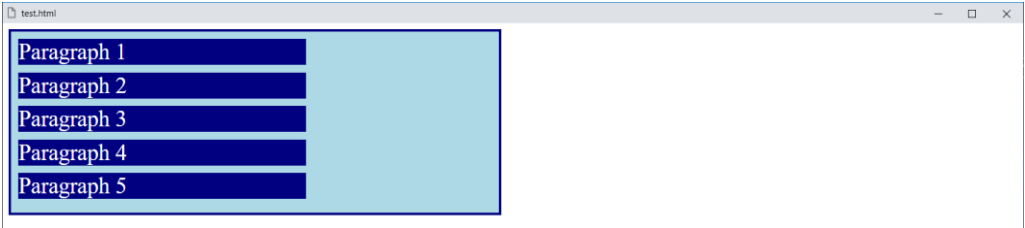
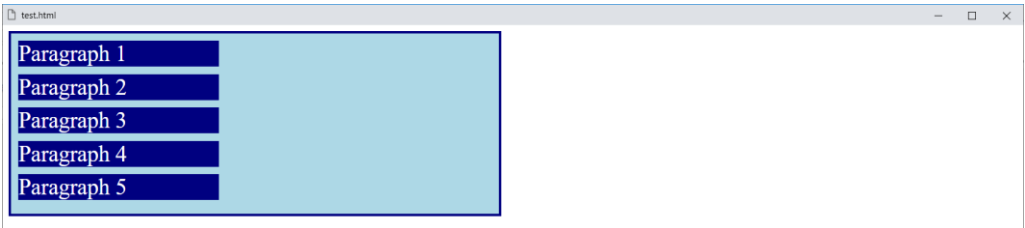
يملك استخدام القيمة steps لتقسيم الانتقال الى خطوات كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
div{
width:650px; height:240px; background:lightblue;
border:3px solid navy; font-size:30px;
}
div p{
width:150px; margin:10px; background:navy; color:#fff;
transition-property:all;
transition-duration:2500ms;
transition-timing-function:steps(3, end);
}
div:hover p{ width:500px; }
</style>
</head>
<body>
<div>
<p id="p1">Paragraph 1</p>
<p id="p2">Paragraph 2</p>
<p id="p3">Paragraph 3</p>
<p id="p4">Paragraph 4</p>
<p id="p5">Paragraph 5</p>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



وعند لمس الـ div ستلاحظ ثلاث مراحل كما يلي:

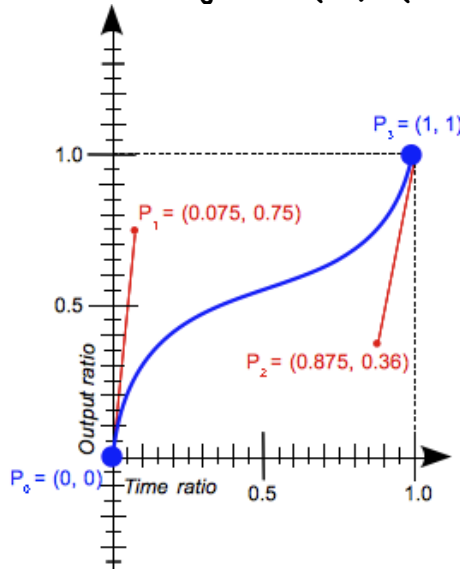


وهنا تم تقسيم وقت الانتقال على ثلاث مراحل، كما هو موضح بالأشكال السابقة، والكلمة end التي تم وضعها بعد 3 معناها أنك بعد اللمس لا يتحرك سريعاً بشكل مباشر، يمكنك عدم وضعها وستكون افتراضية، ولو تم استخدام start بدلاً منها سيتم التحرك في بداية اللمس سريعاً.

كما ان الخاصية transition-timing-function تأخذ
أيضاً القيمة step-start وهي تعادل steps(1,start)،
وتأخذ أيضاً القيمة step-end وهي تعادل steps(1,end).

استخدام cubic-bezier

اعلم ان الخاصية transition-timing-function تأخذ قيمة تجعلك تقوم بتأليف توظيف الوقت بنفسك، وهذه القيمة هي الدالة cubic-bezier وهي تأخذ منك أربع ارقام لتقوم بتعيينه نقطتيه، ثم على أساسهم يتم رسم curve لتتم عليه السرعات، ولكنه **انتبه!** جيداً، فلا بد من التركيز بشكل دقيق لفهم هذه العملية، لأنه في هذا الموضوع نحتاج أربع نقاط وهم p_0, p_1, p_2, p_3 وفي اللغة CSS تم تثبيت النقطة p_0 على القيمته $(0,0)$ لتعبر عن البداية، وتم تثبيت النقطة p_3 على القيمته $(1,1)$ لتعبر عن النهاية، ثم تم ترك النقطتيه p_1 و p_2 لك لتلاعب كما تحب مع السرعات، والشكل التالي يوضح الأربع نقاط والتعامل معهم لرسم المنحنى:

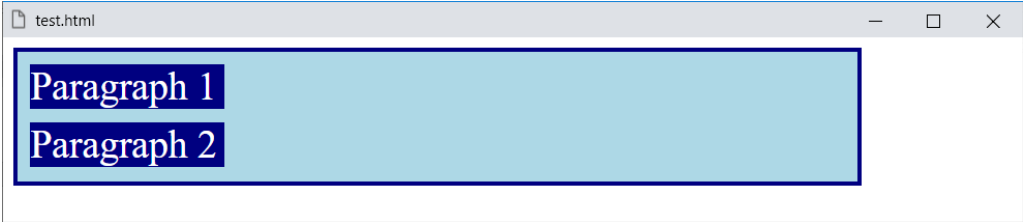


والدالة cubic-bezier تأخذ القيم (x_1, y_1, x_2, y_2) ، وعني ذلك أنها تأخذ النقطة p_1 ثم النقطة p_2 لترسم منحنى السرعة.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{
    width:650px; height:100px; background:lightblue;
    border:3px solid navy; font-size:30px;
  }
  div p{
    width:150px; margin:10px; background:navy; color:#fff;
    transition-property:all;
    transition-duration:2500ms;
  }
  div:hover p{ width:500px; }
  #p1{
    transition-timing-function:linear;
  }
  #p2{
    transition-timing-function:cubic-bezier(0,0,1,1);
  }
</style>
</head>
<body>
<div>
  <p id="p1">Paragraph 1</p>
  <p id="p2">Paragraph 2</p>
</div>
</body>
</html>
```

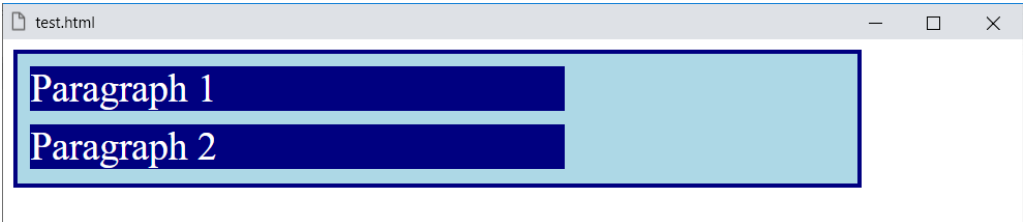
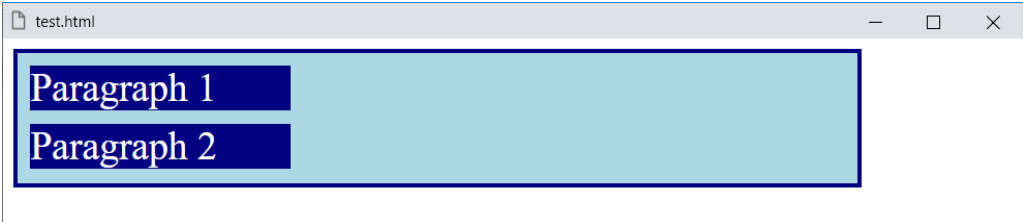
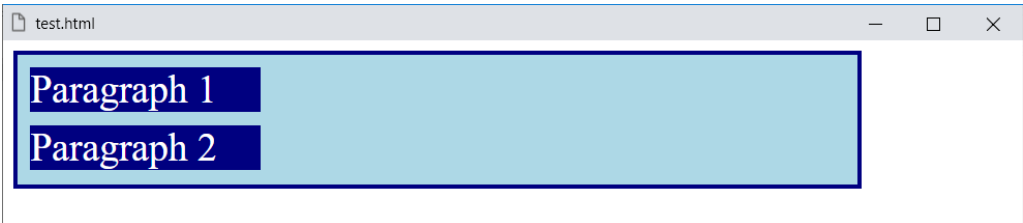

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



وهنا تم عمل فقرة على القيمة linear وأخري أخذت الكود التالي:

```
transition-timing-function:cubic-bezier(0,0,1,1);
```

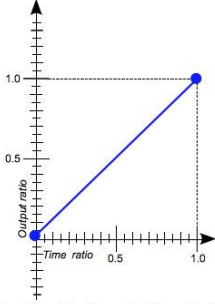
وهو يعادل أيضاً ال linear، وإذا تم لمس العنصر div سوف تلاحظ أنهم يمشون مع بعضهم البعض بنفس معدلات السرعة ولاحظ ما يلي:



بل وحتى في الرجوع، ستلاحظ أنهم مثل بعضهم أيضاً.

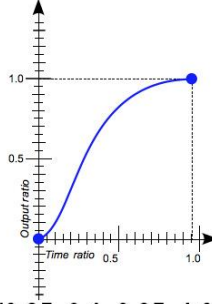
ولتفاصيل أكثر الشكل التالي يعبر عن الأنواع المختلفة:

Linear



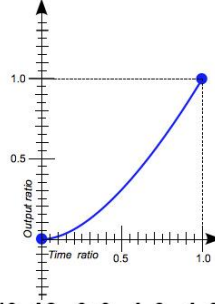
(0.0, 0.0, 1.0, 1.0)

Ease



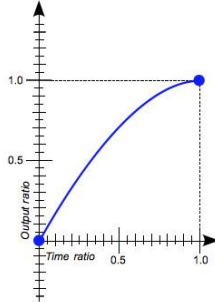
(0.25, 0.1, 0.25, 1.0)

Ease-in



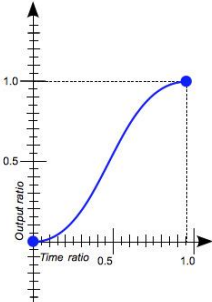
(0.42, 0.0, 1.0, 1.0)

Ease-out



(0.0, 0.0, 0.58, 1.0)

Ease-in-out



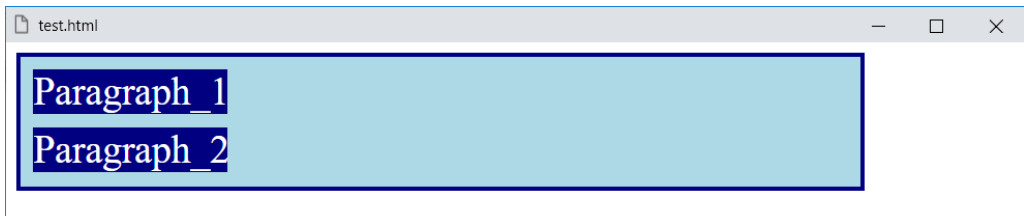
(0.42, 0.0, 0.58, 1.0)

انتبه! فإذا تم الزيادة عن 1 أو تم التقليل عن الـ 0 فسوف تخرج من الإطار الموضح بالصور وستتغير السرعة، وسنوضح ذلك بالمثال التالي:

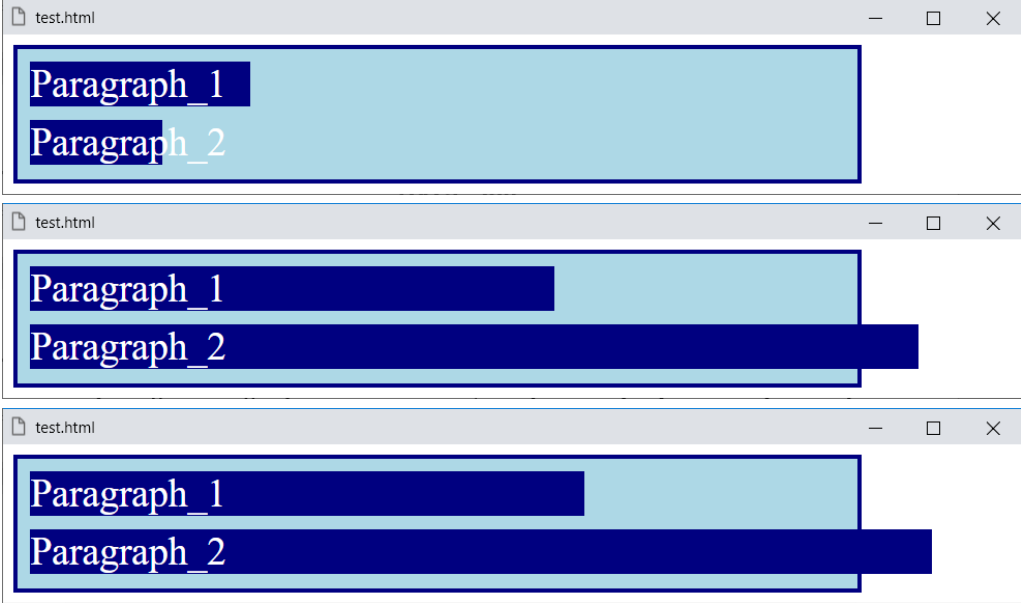
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
div{
width:650px; height:100px; background:lightblue;
border:3px solid navy; font-size:30px;
}
div p{
width:150px; margin:10px; background:navy; color:#fff;
transition-property:all;
transition-duration:5500ms;
}
div:hover p{ width:500px; }
#p1{
transition-timing-function:linear;
}
#p2{
transition-timing-function:cubic-bezier(0.2, -0.9,0.6,2.9);
}
</style>
</head>
<body>
<div>
<p id="p1">Paragraph_1</p>
<p id="p2">Paragraph_2</p>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



والآن قم بلمس الؤسم div وراقب الؤسم الفقرة p2 ولاحظ الأشكال التالية والتي تدل على انحراف معدلات السرعة:



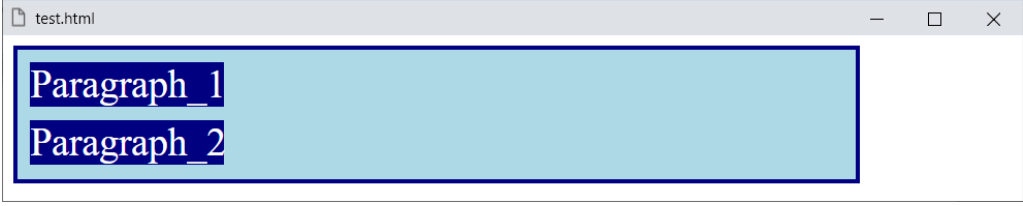
ومع هنا يتبين لنا أن موضوع ال cubic-bezier يجعلك تتحكم في معدلات السرعة كما تريد.

الخاصية transition

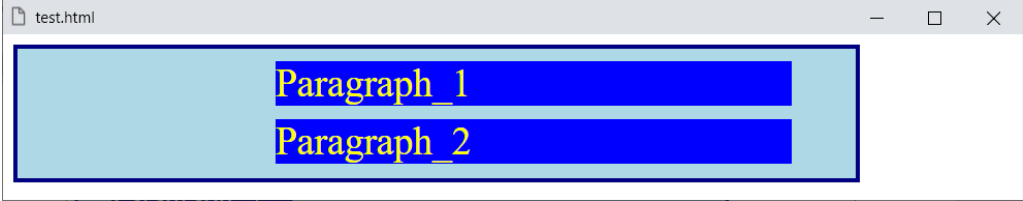
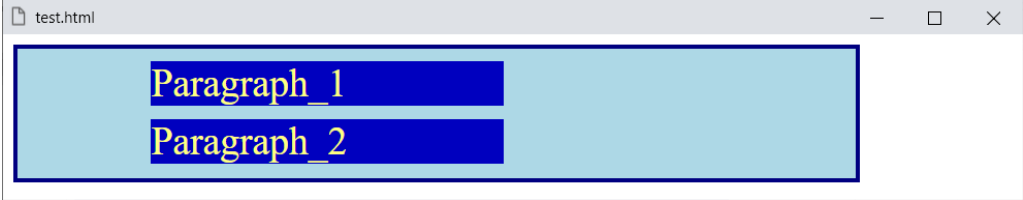
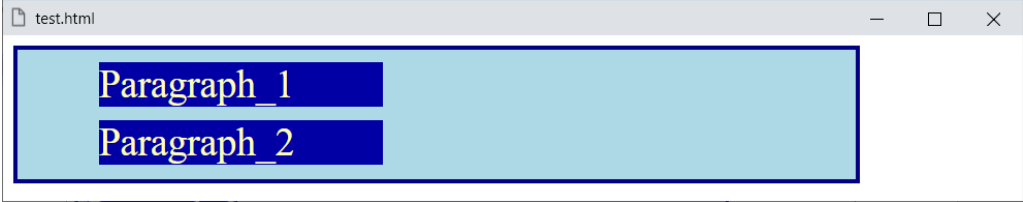
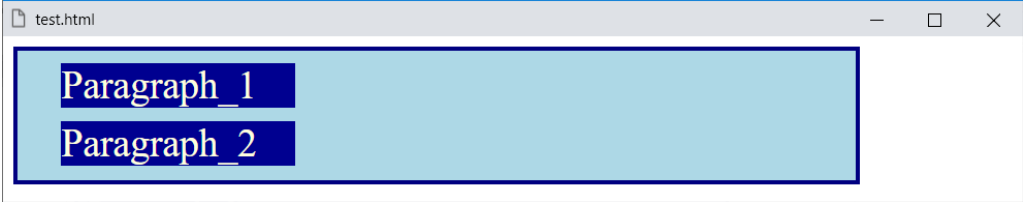
كل ما تم شرحه يمكنه إمكانية كاختصار داخل الخاصية transition، وسوف نجرب بفضل الله في المثال التالي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div{
    width:650px; height:100px; background:lightblue;
    border:3px solid navy; font-size:30px;
  }
  div p{
    width:150px; margin:10px; background:navy; color:#fff;
    transition:all 5s linear 0.5s;
  }
  div:hover p{
    width:400px; margin-left:200px;
    background:blue; color:yellow;
  }
</style>
</head>
<body>
<div>
  <p id="p1">Paragraph_1</p>
  <p id="p2">Paragraph_2</p>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



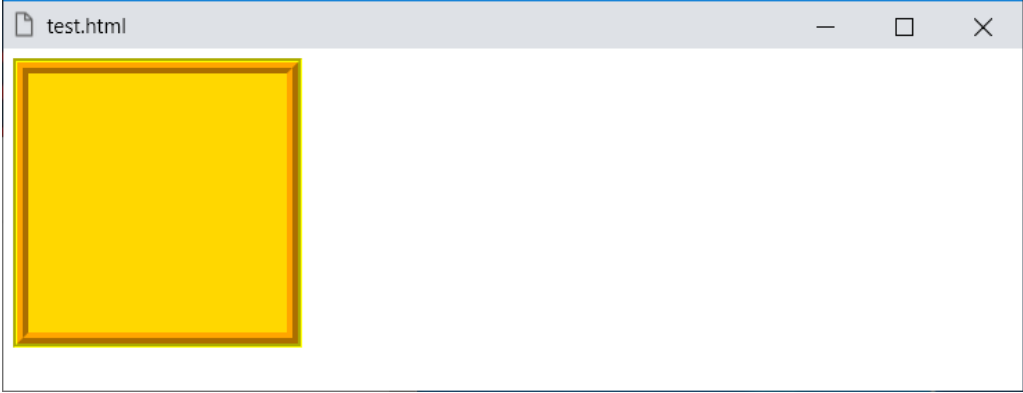
وعند لمس الؤسم div لاحظ الاختلافات كما بالأشكال التالية:



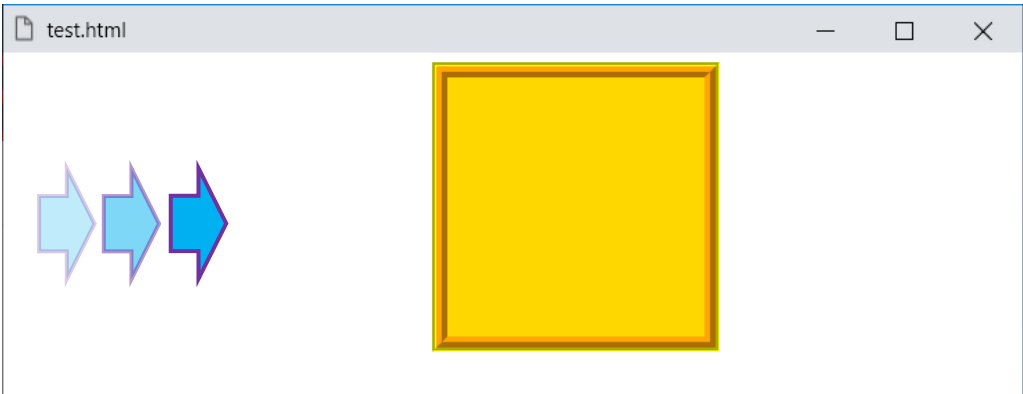
وهنا نلاحظ عمل اختصار shorthand قوي لعملية ال transition وكتابة الكك أمام خاصية transition.

الحرّكة Animation

الحرّكة في صفحة الويب تعدّ من الأمور الجميلة الشائعة، والتي تعطي مظهر جمالي لجزء من صفحة الويب أو لصفحة الويب بشكل عام، والحرّكة هي تغيير العنصر من وضع لوضع آخر، مثلاً أن ينتقل العنصر من مكان إلى مكان آخر، أو مثلاً أن يتغير لون العنصر من لون إلى لون آخر، أو مثلاً أن يتم تحوله، إلى آخر هذه الأمور، مثلاً انظر إلى الشكل التالي:



ثمّ سوف يتحرك العنصر كما بالشكل التالي:



وهنا لاحظ حركة العنصر من اليسار إلى اليمين.

صناعة مؤثر الحركة بالمحدد @keyframes

عند الاحتياج لعمل حركة animation، لابد أولاً من صنع تلك الحركة، وإعطائها اسم كما تريد، ثم النداء فيما بعد على تلك الحركة التي صنعناها وربتها بنفسك، وعند عمل هذه الحركة يتم استخدام المحدد @keyframes وسنوضحها بالكود التالي:

```
@keyframes movetoright{  
  from{ left:0; }  
  to { left:300px; }  
}
```

وكما نلاحظ أولاً نكتب المحدد @keyframes، ثم مسافة، ثم اسم الـ animation، ثم أقواس المجموعة { }، وبداخل تلك الأقواس تم عمل from بأقواسها مع تنسيقات CSS، وتم عمل to بأقواسها مع تنسيقات CSS، ومعني from أي ما بداخلها هو بداية الحركة، ومعني to أي ما بداخلها نهاية الحركة، ثم بعد تنفيذ تلك الحركة تعود مرة ثانية إلى البداية، وهنا يتضح لنا أن الحركة مقسمة إلى قسمين وهما from و to، ولاحظ أنه تم تسميته الـ animation بالاسم movetoright، وهو اسم اختياري حسب رغبتك أنت، وتبنيته كما تريد، كما أننا ننصح أن تكون التسمية متوافقة مع ما يقوم الـ animation بعمله، وهنا تم تسميته بهذا الاسم، لأنه يبدأ الـ left بصفر، ثم ينتهي الـ left بـ 300px، أي يتحرك إلى اليمين، وأيضاً **انتبه!** فانت تستطيع كتابة أكواد CSS كثيرة في الـ from والـ to.

وأيضاً يمكن تقسيم الـ animation الى أكثر من قسميه عن طريق النسب المئوية، مثل 10% و 50% و 70%، وهكذا بالترتيب على الأقسام التي تريدها، فمثلاً سنقسمها الى أربعة بالكود التالي:

```
@keyframes movetoright{
  0%{ left:0; }
  25%{ left:99px; }
  75%{ left:199px; }
  100%{ left:300px; }
}
```

وكما نلاحظ تم تقسيم الـ animation الى أربعة أقسام، وهنا البداية هي 0%، ثم هناك تغيرات عند 25%، ثم هناك تغيرات عند 75%، ثم هناك تغيرات عند مرحلة الاكتمال وهي 100%، ويمكن أيضاً تقسيم الـ animation الى ستة أجزاء بالكود التالي:

```
@keyframes movetoright{
  0%{ left:0; }
  20%{ left:60px; }
  40%{ left:120px; }
  60%{ left:180px; }
  80%{ left:240px; }
  100%{ left:300px; }
}
```

وكما نلاحظ أنه يمكن تقسيم الـ animation الى أي نسب تريدها.

تشغيل الحركة

بعد ان تعلمنا صناعة ال animation، يمكن الآن تشغيله، وتطبيقه على أحد العناصر، ولكه لابد من ضبط خاصية اسم ال animation، وهي الخاصية animation-name ليتم ضبطها على المؤثر الذي تم عمله مع قبل وهو movetoright، ويجب ضبط المدة الخاصة بال animation-duration بالخاصية animation-duration على مدة مثل 3s أو 5s أو غير ذلك، وسنجرّب بالمثال التالي:

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
#d1{
width:150px; height:150px; background:gold;
border:6px ridge orange; outline:3px groove yellow;
position:absolute; left:0;
animation-name:movetoright;
animation-duration:3s;
}

@keyframes movetoright{
from{ left:0; }
to{ left:300px; }
}
</style>
</head>
<body>
<div id="d1"></div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر وتتحرك وتعود كما بالأشكال التالية:



توظيف وقت الحركة animation-timing-function

هذه الخاصية لتوظيف وقت ال animation، والقيمة الافتراضية لهذه الخاصية هي ease أي سهل أو عادي، وتأخذ أيضاً عدة قيم أخرى، ولكه **انتبه!** فإن هذا الموضوع تم شرحه بالتفصيل في موضوع الانتقال transition.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
#d1{
width:150px; height:150px; background:gold;
border:6px ridge orange; outline:3px groove yellow;
position:absolute; left:0;
animation-name:movetoright;
animation-duration:3s;
animation-timing-function:linear;
}

@keyframes movetoright{
from{ left:0; }
to{ left:300px; }
}
</style>
</head>
<body>
<div id="d1"></div>
</body>
</html>
```

وبعد تشغيل الصفحة ستلاحظ أن الحركة تختلف وتكون linear.

تأخير الحركة animation-delay

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
#d1{
width:150px; height:150px; background:gold;
border:6px ridge orange; outline:3px groove yellow;
position:absolute; left:0;
animation-name:movetoright;
animation-duration:3s;
animation-timing-function:linear;
animation-iteration-count:3;
animation-delay:2s;
}

@keyframes movetoright{
from{ left:0; }
to{ left:300px; }
}
</style>
</head>
<body>
<div id="d1"></div>
</body>
</html>
```

وبعد تشغيل الصفحة ستلاحظ أن الحركة له تنفذ إلا بعد ثانيته.

عدد التكرارات animation-iteration-count

يمكّن التحكم في عدد مرات تكرر الـ animation عن طريق الخاصية animation-iteration-count والتي تأخذ رقم ليتم التكرار على أساسه، فمثلاً 3 للتكرار ثلاث مرات، أو 5 للتكرار خمسة مرات، وهكذا حسب الرغبة، وسنجد **بالكود!** التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
#d1{
width:150px; height:150px; background:gold;
border:6px ridge orange; outline:3px groove yellow;
position:absolute; left:0;
animation-name:movetoright;
animation-duration:3s;
animation-timing-function:linear;
animation-iteration-count:3;
}

@keyframes movetoright{
from{ left:0; }
to{ left:300px; }
}
</style>
</head>
<body>
<div id="d1"></div>
</body>
</html>
```

وبعد تشغيل الصفحة، ستلاحظ أن الحركة تكرر ثلاث مرات.

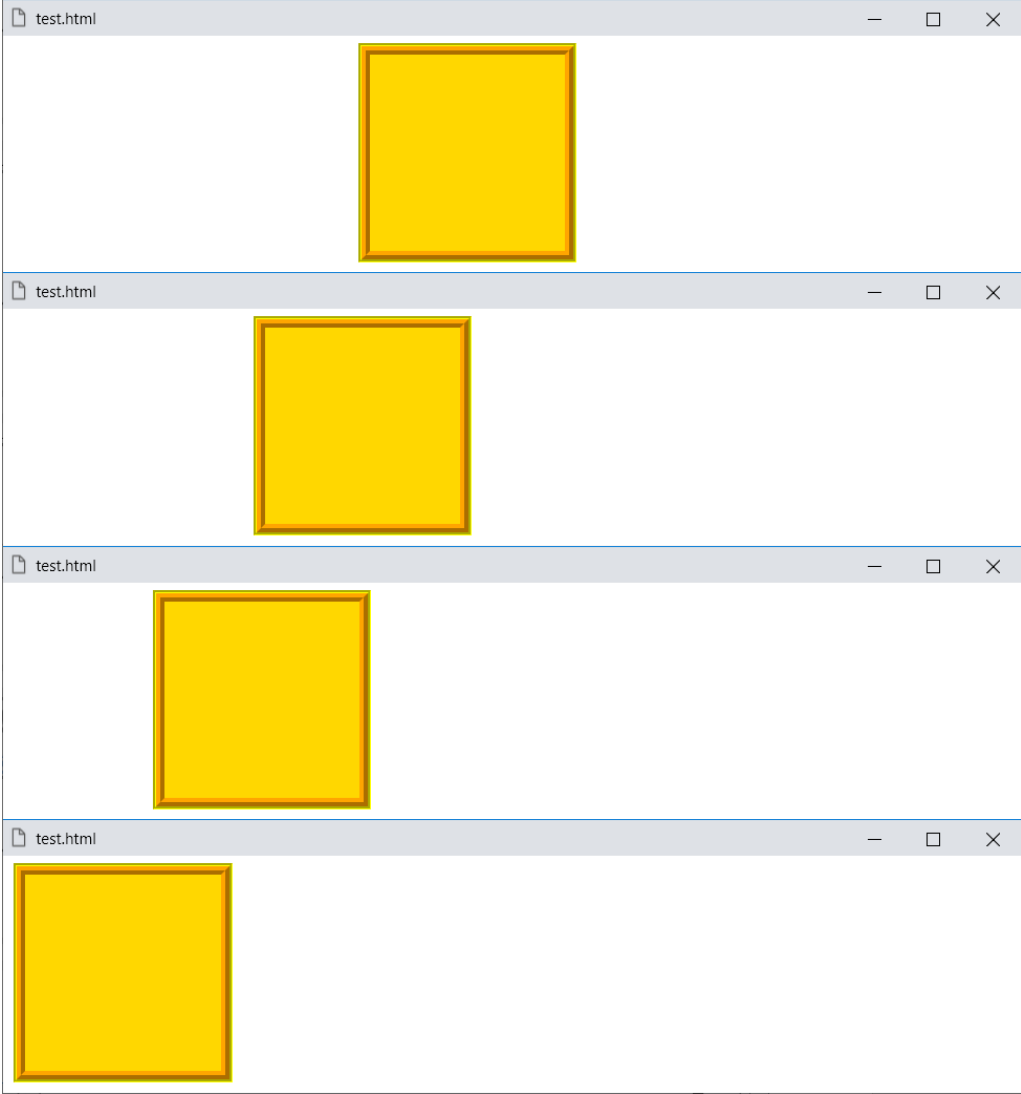
توجيه الحركة animation-direction

وهو طريق الخاصية animation-direction، يتم التحكم في آليه الحركة، فمع الطبيعي أن تبدأ، ثم تتم الحركة، ثم تنتهي مع حيث بدأت، وهنا الخاصية animation-direction تأخذ القيمة الافتراضية لها وهي normal، أما القيمة reverse تجعلها تعكس الحركة تماماً، وسنجد بالكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
#d1{
width:150px; height:150px; background:gold;
border:6px ridge orange; outline:3px groove yellow;
position:absolute; left:0;
animation-name:movetoright;
animation-duration:3s;
animation-timing-function:linear;
animation-iteration-count:6;
animation-delay:0.2s;
animation-direction:reverse;
}

@keyframes movetoright{
from{ left:0; }
to{ left:300px; }
}
</style>
</head>
<body>
<div id="d1"></div>
</body>
</html>
```

وبءء ءشءعل الصءءة ءنعكس الءءة ءما بالأشكال ءالفة:



وهنا ءم عكس الءءة ءاماف؁ فءلاً ءه أن ىسء العنءر ءه الىسار الى اللمفه؁ أءبءل ىبءأ ءه اللمفه؁ ءم ىءهب الى الىسار؁ وهنا لاءظ ءءوففر فى الوءء؁ ءءء لو ءم اءءفاء عمل ءرءفه عكسفه له ءبءل الوءء لءءطفءهءا؁ بل إن الءفءة reverse ءسهءل علفءك الأمر.

ويمكن توجيه الحركة بحيث تتم كما تم تخطيطها، ثم تعود بشكل عكسي،
وهكذا حسب ال animation-iteration-count الذي تم
ضبطه لها، وسنجرّب **بالكود!** التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
#d1{
width:150px; height:150px; background:gold;
border:6px ridge orange; outline:3px groove yellow;
position:absolute; left:0;
animation-name:movetoright;
animation-duration:3s;
animation-timing-function:linear;
animation-iteration-count:6;
animation-delay:0.2s;
animation-direction:alternate;
}

@keyframes movetoright{
from{ left:0; }
to{ left:300px; }
}
</style>
</head>
<body>
<div id="d1"></div>
</body>
</html>
```

وبعد تشغيل الصفحة ستلاحظ أنها تذهب 1، ثم تعود 2، ثم تذهب 3، ثم
تعود 4، ثم تذهب 5، ثم تعود 6، ولماذا؟! لأن التكرار ب 6، والعودة
محسوبة مع عدد التكرار.

ويمكن استخدام القيمة alternate-reverse، لتجمع بين ال reverse وال alternate، حيث تبدأ من العكس، ثم تستكمل بطريقة ال alternate، وسنجرّب **بالكود!** التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
#d1{
width:150px; height:150px; background:gold;
border:6px ridge orange; outline:3px groove yellow;
position:absolute; left:0;
animation-name:movetoright;
animation-duration:3s;
animation-timing-function:linear;
animation-iteration-count:6;
animation-delay:0.2s;
animation-direction:alternate-reverse;
}

@keyframes movetoright{
from{ left:0; }
to{ left:300px; }
}
</style>
</head>
<body>
<div id="d1"></div>
</body>
</html>
```

وبعد تشغيل الصفحة ستلاحظ أنها تعود 1، ثم تذهب 2، ثم تعود 3، ثم تذهب 4، ثم تعود 5، ثم تذهب 6، ثم تظهر فجأة بشكل عادي، لأن التكرار ب 6 فقط.

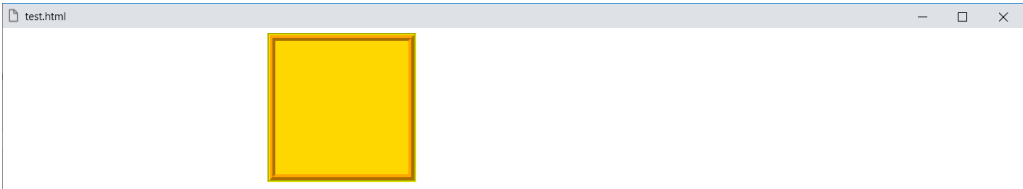
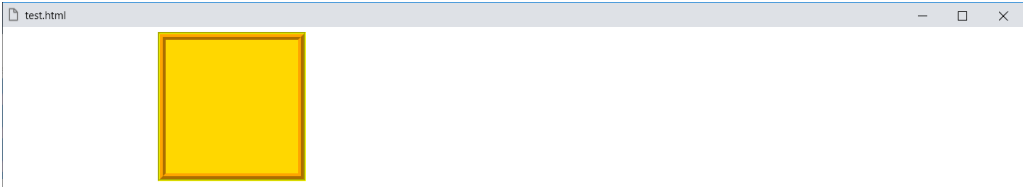
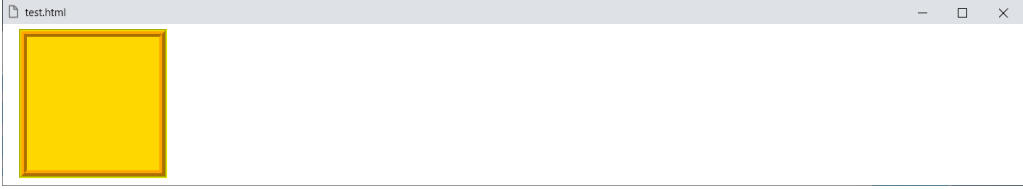
نمط تتبع الحركة animation-fill-mode

عن طريق الخاصية animation-fill-mode يمكن التحكم في نمط تتبع الـ animation، فمثلاً تابع الكود! التالي:

```
<!DOCTYPE html>
<html>
<head>
<style>
#d1{
width:150px; height:150px; background:gold;
border:6px ridge orange; outline:3px groove yellow;
position:absolute; left:0;
animation-name:movetoright;
animation-duration:3s;
animation-timing-function:linear;
animation-iteration-count:1;
animation-delay:0.2s;
animation-direction:normal;
animation-fill-mode:forwards;
}

@keyframes movetoright{
from{ left:0; }
to{ left:300px; }
}
</style>
</head>
<body>
<div id="d1"></div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالأشكال التالية:



وهنا لاحظ جيداً أن العنصر المتحرك ثبت على آخر وضعه أنتهي عليه الـ animation، وهذا ما فعلته القيمة forwards به، كما يوجد قيمة أخرى وهي backwards، وهي تجعل العنصر على التنسيق الأول from أو مثلاً 0% مع الاعتماد على اتجاه الحركة، حتى ولو تم عمل تأخير delay للـ animation، ويوجد أيضاً القيمة both وهي تجمع بين القيمتين forwards و backwards معاً في نفس الوقت.

تشغيل وإيقاف الحركة animation-play-state

الخاصية animation-play-state تأخذ قيمة افتراضية بـ running أي أن ال animation في وضع التشغيل، وهناك قيمة أخرى وهي paused أي إيقاف مؤقت لل animation، وكما نرى هي خاصية بسيطة وتجعلك تتحكم في ال animation سواء تريد منه أن يعمل أو تريد إيقافه.

خاصية التحريك animation

كما تعودنا من اللغة الرائعة CSS ان الخاصية الأم الرئيسية تجمع بين الخصائص الفرعية لها، وهنا الخاصية الأساسية animation يمكنه ان تأخذ قيم الخصائص name و duration و timing- و direction و iteration-count و delay و function و play-state و fill-mode لـ animation .
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
#d1{
width:150px; height:150px; background:gold;
border:6px ridge orange; outline:3px groove yellow;
position:absolute; left:0;
animation: movetoright 3s linear 3 0.5s alternate forwards running;
}

@keyframes movetoright{
from{ left:0; }
to{ left:300px; }
}
</style>
</head>
<body>
<div id="d1"></div>
</body>
</html>
```

وبعد تشغيل الصفحة ستلاحظ أنها تعمل بالاسم والمدة والتوظيف والتكرار والتأخير والتوجيه والتتبع والحالة في نفس الوقت بخاصية واحدة وهي الخاصية animation .

الحركة الانعائية Infinite animation

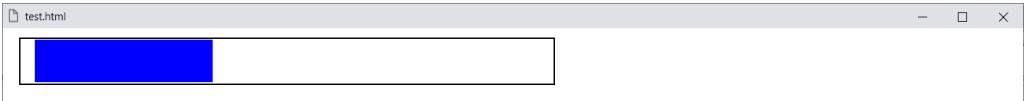
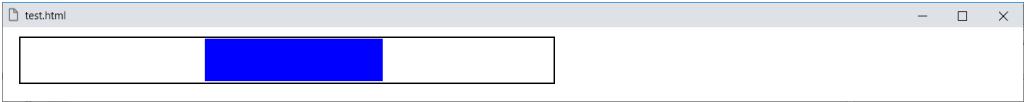
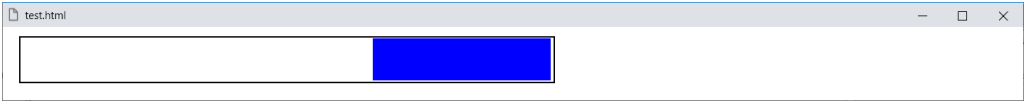
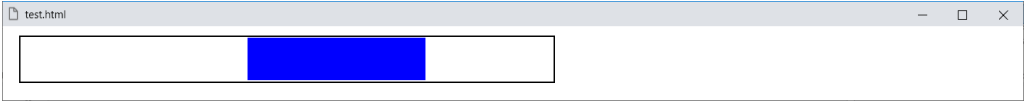
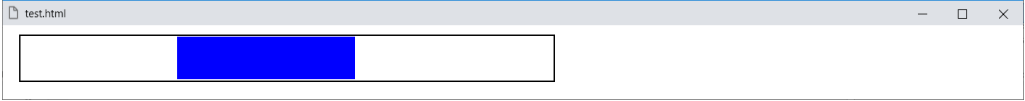
وبعد أن تعرفنا الـ animation بشكل جيد وتدريبنا عليه، يمكنه أن يقوم بعمل مثال بسيط لجعل الحركة لا نهائية باستخدام القيمة `iteration-count: infinite` للعملية التكرارية.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
#d1{
width:600px; height:50px; margin:10px;
border:1.5px ridge #000;
}

#d2{
width:200px; height:47px; background:#00f;
position:absolute; left:21px; top:13px;
animation: mymove 3s linear infinite alternate;
}
@keyframes mymove{
from{ left:21; }
to{ left:417px; }
}
</style>
</head>
<body>
<div id="d1"> <div id="d2"></div> </div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالأشكال التالية:



وهكذا ومثل الأشكال السابقة وبدون توقف يذهب يميناً ويساراً وكأن هناك شيء سوف ننتظره.

أمثلة على ال Animation

وبفضل الله وحده، بعد أن تعرفنا، وفهمنا، وتدريبنا كثيراً على موضوعات ال animation، سوف نقوم بعمل بعض الأمثلة إن شاء الله.

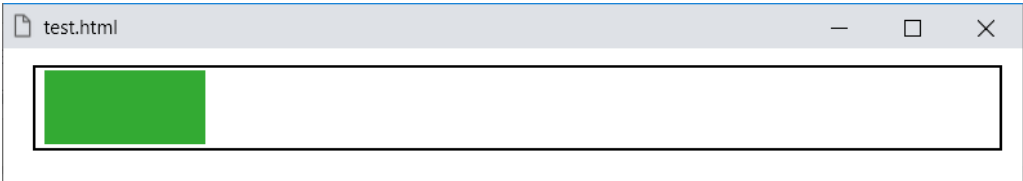
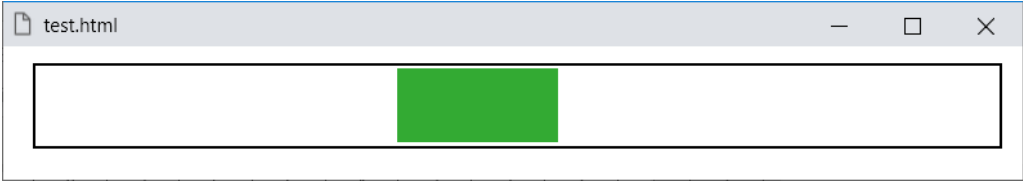
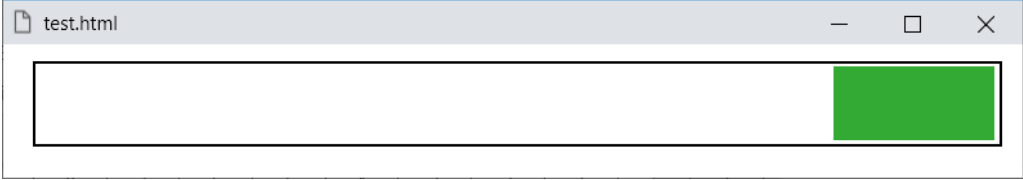
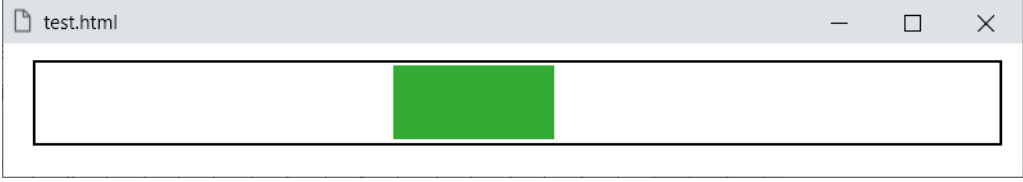
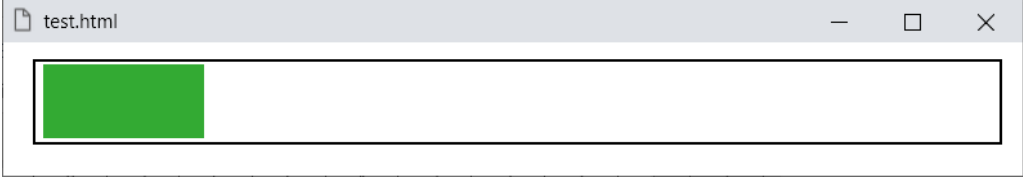
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
#d1{
width:600px; height:50px; margin:10px; border:1.5px ridge #000;
}

#d2{
width:100px; height:46px; margin:2px; background:#3a3;
animation: myaction 5s linear infinite alternate;
}

@keyframes myaction{
0%{ margin-left:0; }
10%{ margin-left:50px; }
20%{ margin-left:100px; }
30%{ margin-left:150px; }
40%{ margin-left:200px; }
50%{ margin-left:250px; }
60%{ margin-left:300px; }
70%{ margin-left:350px; }
80%{ margin-left:400px; }
90%{ margin-left:450px; }
100%{ margin-left:498px; }
}
</style>
</head>
<body>
<div id="d1"> <div id="d2"></div> </div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالأشكال التالية:



وهكذا، ولكنه **انتبه!**، فإن هذا المثال ليس مثل الذي تم تطبيقه في موضوع الحركة الانعائية، لأنه يعتمد ويتلاعب على العائمة اليساري، أما الآخر فكان يتلاعب مع الموضع left، وهنا أفضل من حيث الترابط، حيث لو قمت بوضعه في أي مكان في الصفحة سوف يتجاوب معك، ولا يكون مرتبط بموضع وضبط كثير.

كود! قم بكتابة الكود التالي ثم احفظه:

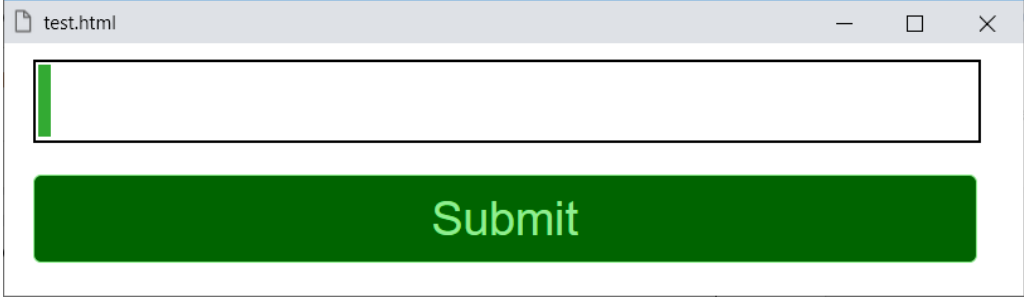
```
<!DOCTYPE html>
<html>
<head>
<style>
input[type=submit]{
background:darkgreen; color:lightgreen; margin:10px;
border:1px solid lightgreen; border-radius:5px;
padding:10px; width:600px; font-size:30px; cursor:pointer;
}

input[type=submit]:hover{ background:#181; }

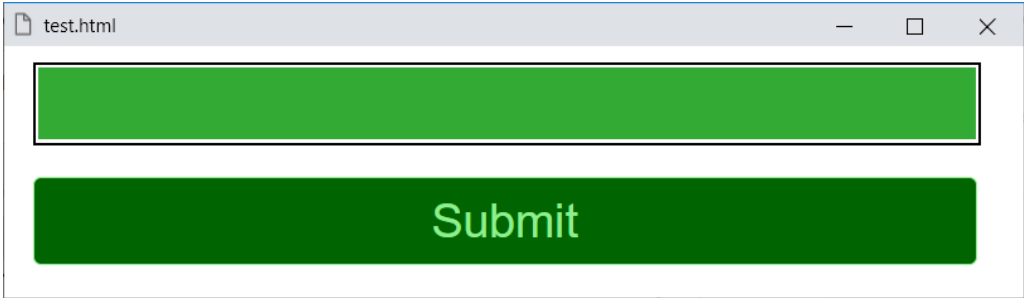
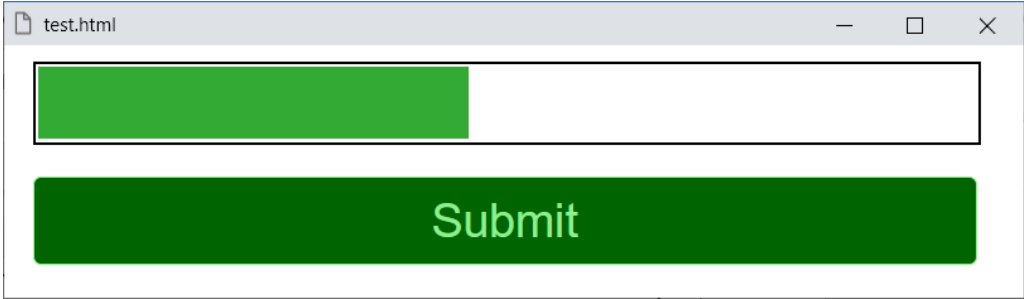
#d1{
width:600px; height:50px; margin:10px;
border:1.5px ridge #000;
}

#d2{
width:0; height:46px; margin:2px; background:#3a3;
animation: mymove 5s linear forwards;
}
@keyframes mymove{
0%{ width:0; }
10%{ width:60px; }
20%{ width:120px; }
30%{ width:180px; }
40%{ width:240px; }
50%{ width:300px; }
60%{ width:360px; }
70%{ width:420px; }
80%{ width:480px; }
90%{ width:540px; }
100%{ width:596px; }
}
</style>
</head>
<body>
<div id="d1"> <div id="d2"></div> </div>
<form> <input type="submit"> </form>
</body>
</html>
```

وبعد تشغيل الصفحة في البداية تظهر كما بالشكل التالي:



ثم نسيه كما بالأشكال التالية:



والزر submit الموجود على الصفحة، هو داخل الـ form، بحيث لو تم الضغط عليه بعد تحميل الصفحة، فتعمل من البداية مرة أخرى.

كود! قم بكتابة الكود التالي ثم احفظه:

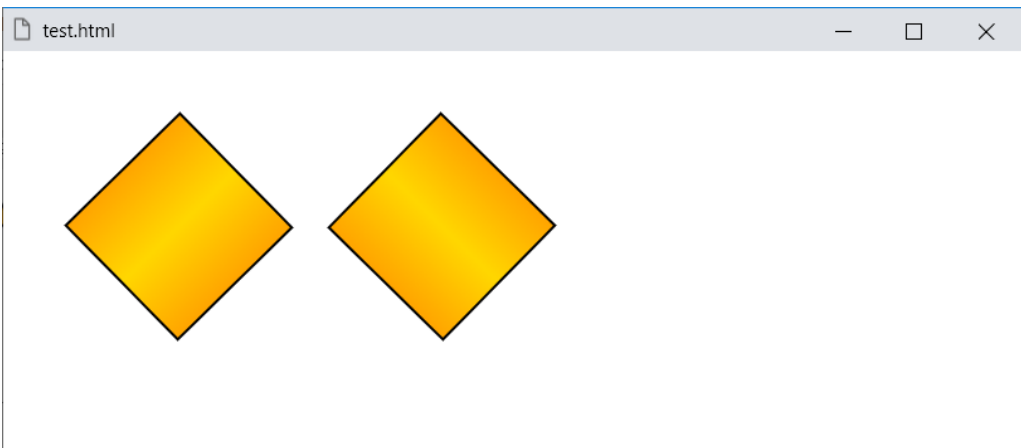
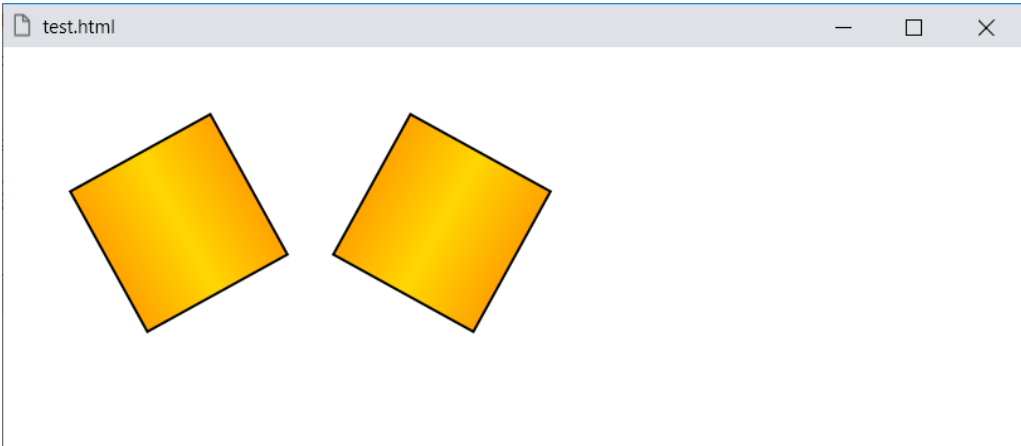
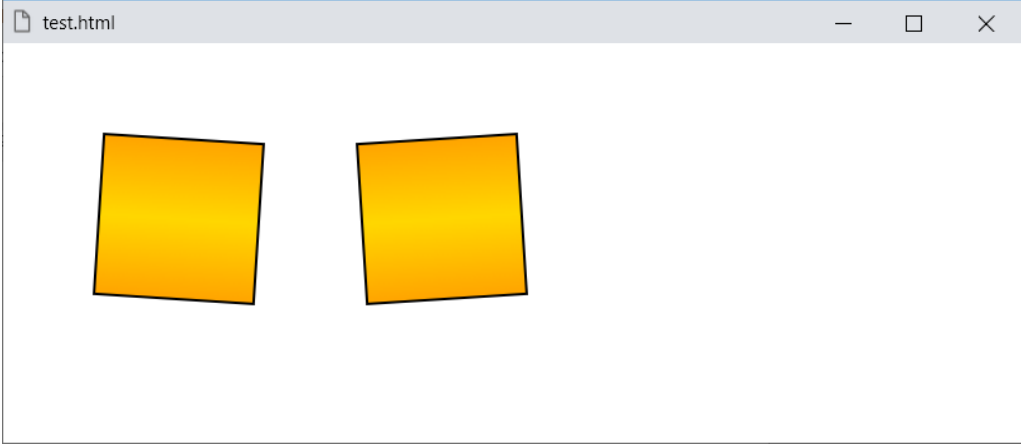
```
<!DOCTYPE html>
<html>
<head>
<style>
  body{ margin:30px; }

  #d1 , #d2{
    width:100px; height:100px; margin:30px;
    border:1.5px ridge #000; display:inline-block;
    background:linear-gradient(orange,gold,orange);
  }

  #d1{ animation: mymove 5s linear infinite; }
  #d2{ animation: mymove 5s linear infinite reverse; }

  @keyframes mymove{
    0%{ transform:rotate(0deg); }
    20%{ transform:rotate(72deg); }
    40%{ transform:rotate(144deg); }
    60%{ transform:rotate(216deg); }
    80%{ transform:rotate(288deg); }
    100%{ transform:rotate(360deg); }
  }
</style>
</head>
<body>
  <div id="d1"></div>
  <div id="d2"></div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالأشكال التالية:



رجاء انتبه! فهنا سوف نقف وقفة مع المثال السابق، ونفكر في فضل الله الأحد الصمد الذي لم يلد ولم يولد ولم يكن له كفواً أحد، فالله مع علينا أن خلقنا وفهمنا، فقال الله في القرآن فَفَهَّمْنَاهَا سُلَيْمَانَ، وهنا سنركز على فهم آلية الحركة، فهنا أردنا أن نجعل العنصر يستدير عن طريق ال rotate، وال rotate يستدير عن طريق الدرجة، والدرجة تتراوح من صفر الى 360، وفي ال animation قمنا بعمل خمس مراحل بعد الصفر وهم 20% و 40% و 60% و 80% و 100%، ولو تم قسمة ال 360 درجة عليهم سيكون $360 \div 5 = 72$ ، إذاً ليتم عمل توازن مع ال linear نزيد في كل مرحلة 72 درجة عن سابقتها، فكانت 0، ثم زادت فأصبحت 72، ثم 144، ثم 216، ثم 288، ثم أغلقت بـ 360، مع العلم أنه لو كان هناك 10 مراحل في ال animation لتمت القسمة على 10، لتكون الحركة متجانسة، وبنفس الأسلوب نستطيع عمل أمور كثيرة، سواء كنت تتعامل مع العرض أو الطول، أو أي خاصية تدعم ال animation.

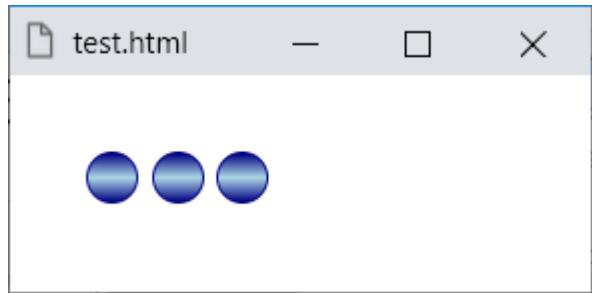
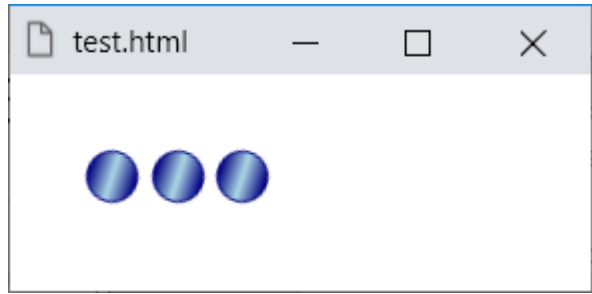
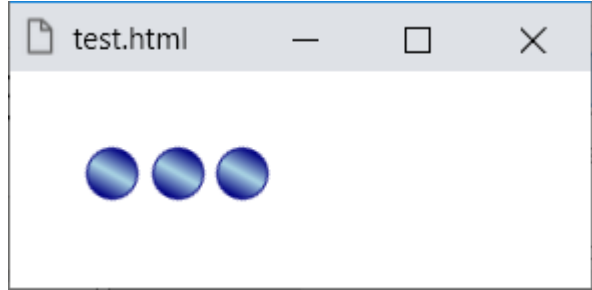
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{ margin:30px; }

  .circle{
    width:20px; height:20px; margin:0.5px;
    border-radius:15px;
    border:0.5px solid navy; display:inline-block;
    background:linear-gradient(navy,lightblue,navy);
    animation: mymove 5s cubic-bezier(0,.98,.98,.02) infinite;
  }

  @keyframes mymove{
    0%{ transform:rotate(0deg); }
    20%{ transform:rotate(72deg); }
    40%{ transform:rotate(144deg); }
    60%{ transform:rotate(216deg); }
    80%{ transform:rotate(288deg); }
    100%{ transform:rotate(360deg); }
  }
</style>
</head>
<body>
  <div class="circle"></div>
  <div class="circle"></div>
  <div class="circle"></div>
</body>
</html>
```


وبعد تشغيل الصفحة تظهر كما بالأشكال التالية:



وهكذا فنظف تسدير الي ما لا نهاية.

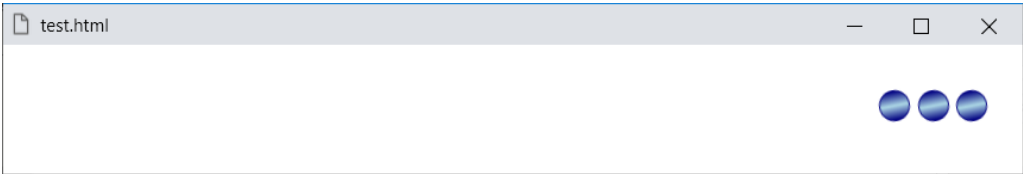
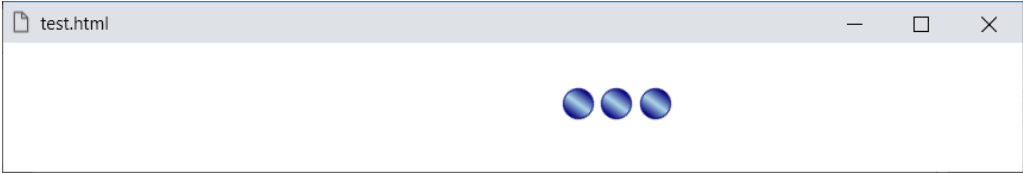
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{ margin:30px; }

  .circle{
    width:20px; height:20px; margin:0.5px;
    border-radius:15px; position:relative;
    border:0.5px solid navy; display:inline-block;
    background:linear-gradient(navy,lightblue,navy);
    animation: mymove 5s linear infinite alternate;
  }

  @keyframes mymove{
    0%{ transform:rotate(0deg); left:0%; }
    25%{ transform:rotate(90deg); left:23%; }
    50%{ transform:rotate(180deg); left:46%; }
    75%{ transform:rotate(270deg); left:69%; }
    100%{ transform:rotate(360deg); left:92%; }
  }
</style>
</head>
<body>
  <div class="circle"></div>
  <div class="circle"></div>
  <div class="circle"></div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالأشكال التالية:



وهكذا بفضل الله تستدير وتسير الى ما لا نهاية.

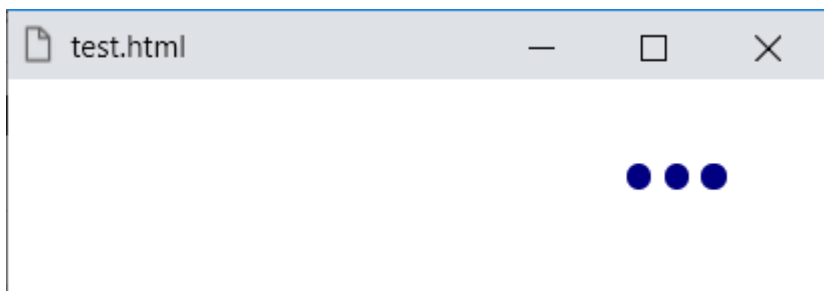
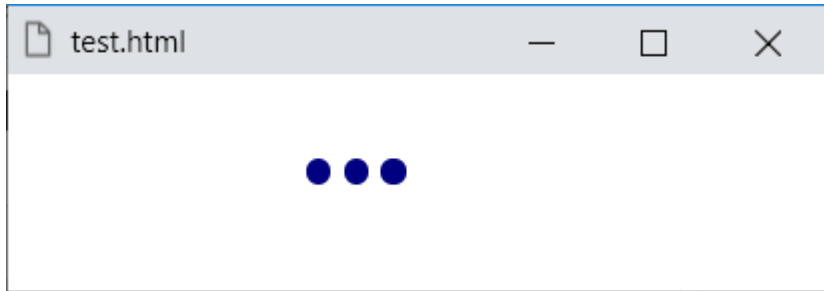
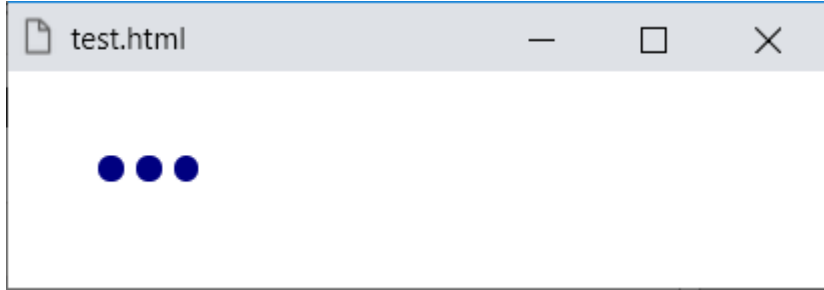
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{ margin:30px; }

  .circle{
    width:10px; height:10px; margin:0.5px;
    border-radius:15px; position:relative;
    background:navy; display:inline-block;
    animation: mymove 2s linear infinite alternate;
  }

  @keyframes mymove{
    0%{ left:0%; }
    25%{ left:23%; }
    50%{ left:46%; }
    75%{ left:69%; }
    100%{ left:92%; }
  }
</style>
</head>
<body>
  <div class="circle"></div>
  <div class="circle"></div>
  <div class="circle"></div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالأشكال التالية:



كود! قم بكتابة الكود التالي ثم احفظه:

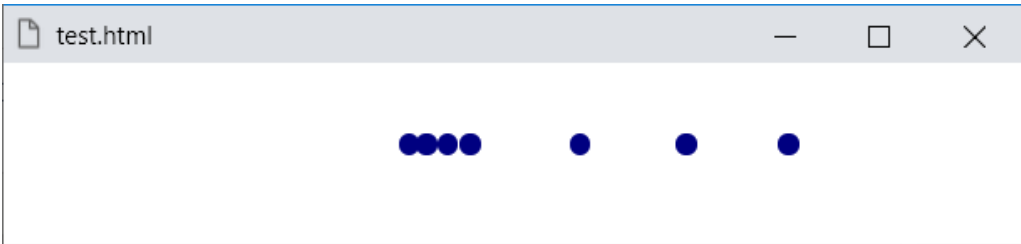
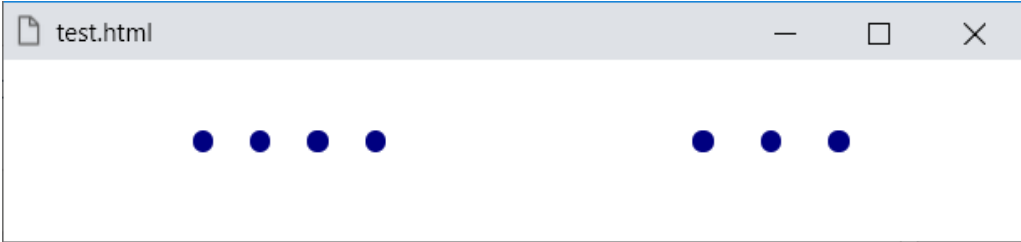
```
<!DOCTYPE html>
<html>
<head>
<style>
  body{ margin:30px; }

  .circle{
    width:10px; height:10px; margin:0.5px;
    border-radius:15px; position:relative;
    background:navy; display:inline-block;
  }

  #c1{ animation: mymove 2.10s linear infinite alternate; }
  #c2{ animation: mymove 2.15s linear infinite alternate-reverse; }
  #c3{ animation: mymove 2.20s linear infinite alternate; }
  #c4{ animation: mymove 2.25s linear infinite alternate-reverse; }
  #c5{ animation: mymove 2.30s linear infinite alternate; }
  #c6{ animation: mymove 2.35s linear infinite alternate-reverse; }
  #c7{ animation: mymove 2.40s linear infinite alternate; }

  @keyframes mymove{
    0%{ left:0%; }
    25%{ left:20%; }
    50%{ left:40%; }
    75%{ left:60%; }
    100%{ left:80%; }
  }
</style>
</head>
<body>
  <div class="circle" id="c1"></div>
  <div class="circle" id="c2"></div>
  <div class="circle" id="c3"></div>
  <div class="circle" id="c4"></div>
  <div class="circle" id="c5"></div>
  <div class="circle" id="c6"></div>
  <div class="circle" id="c7"></div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالأشكال التالية:



وهكذا يسببون في شكل متناثر الى ما لا نهاية.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{ margin:100px; }

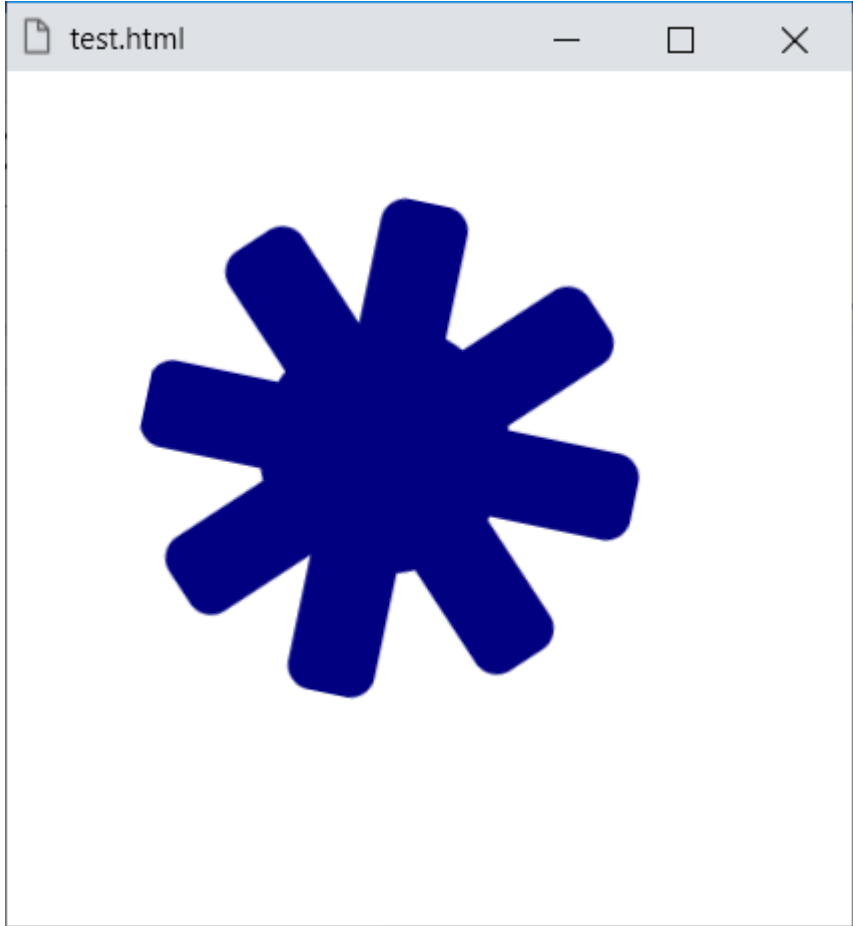
  #main{
    width:100px; height:100px; margin:0.5px;
    background:navy; display:inline-block; border-radius:55px;
  }

  .child{ background:navy; width:200px; height:35px; border-radius:10px; }
  #d1{ transform:rotate(90deg); margin-top:30px; margin-left:-50px; }
  #d2{ transform:rotate(180deg); margin-top:-30px; margin-left:-50px; }
  #d3{ transform:rotate(135deg); margin-top:-40px; margin-left:-50px; }
  #d4{ transform:rotate(-135deg); margin-top:-35px; margin-left:-50px; }

  #main{ animation: mymove 3s linear infinite; }

  @keyframes mymove{
    0%{ transform:rotate(0deg); }
    25%{ transform:rotate(90deg); }
    50%{ transform:rotate(180deg); }
    75%{ transform:rotate(270deg); }
    100%{ transform:rotate(360deg); }
  }
</style>
</head>
<body>
  <div id="main">
    <div id="d1" class="child"></div>
    <div id="d2" class="child"></div>
    <div id="d3" class="child"></div>
    <div id="d4" class="child"></div>
  </div>
</body>
</html>
```


وبعد تشغيل الصفحة ستلاحظ أن الشكل يسد يد وانظر الشكل التالي:



كود! قم بكتابة الكود التالي ثم احفظه:

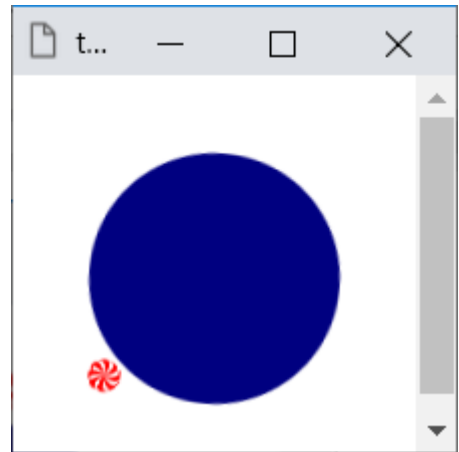
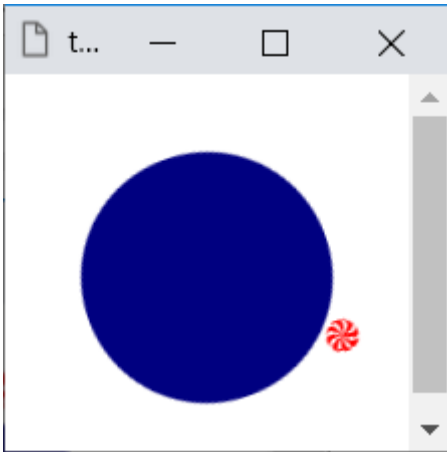
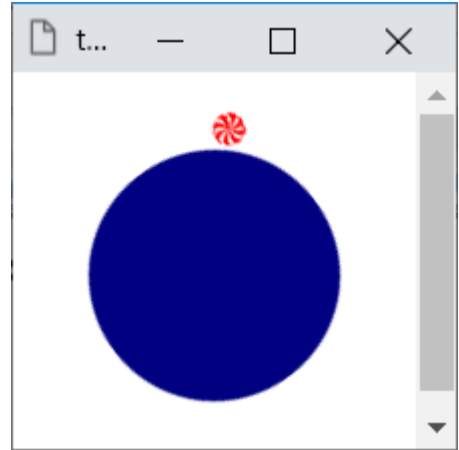
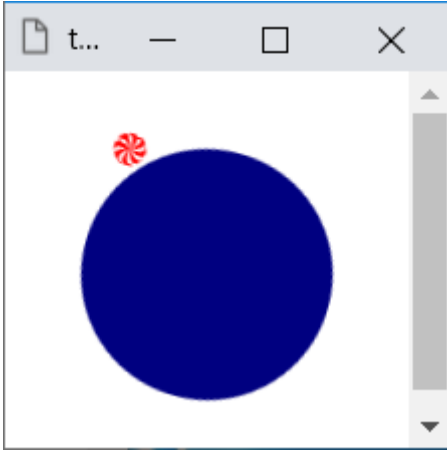
```
<!DOCTYPE html>
<html>
<head>
<style>
  body{ margin:30px; }

  #main{
    width:100px; height:100px; border-radius:55px;
    color:red; font-size:15px; font-weight:bold;
    background:navy; display:inline-block;
    animation: mymove 3s linear infinite;
  }

  @keyframes mymove{
    0%{ transform:rotate(0deg); }
    25%{ transform:rotate(90deg); }
    50%{ transform:rotate(180deg); }
    75%{ transform:rotate(270deg); }
    100%{ transform:rotate(360deg); }
  }

</style>
</head>
<body>
  <div id="main">
    &#1422;
  </div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالأشكال التالية:



وهكذا وهي تستدير، ولكه يعينى لك وكأن النص هو الذي يستدير حولها.

كود! قم بكتابة الكود التالي ثم احفظه:

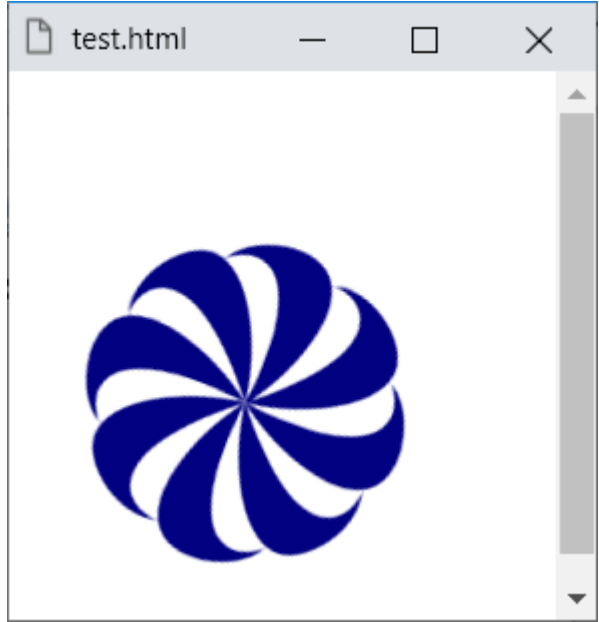
```
<!DOCTYPE html>
<html>
<head>
<style>
  body{ margin:30px; }

  #main{
    color:navy; font-size:150px; font-weight:bold;
    display:inline-block;
    animation: mymove 3s linear infinite;
  }

  @keyframes mymove{
    0%{ transform:rotate(0deg); }
    25%{ transform:rotate(90deg); }
    50%{ transform:rotate(180deg); }
    75%{ transform:rotate(270deg); }
    100%{ transform:rotate(360deg); }
  }

</style>
</head>
<body>
  <div id="main">
    &#1422;
  </div>
</body>
</html>
```

وبعد تشغيل الصفحة ستلاحظ دورانها وانظر الى الشكل التالي:



كود! قم بكتابة الكود التالي ثم احفظه:

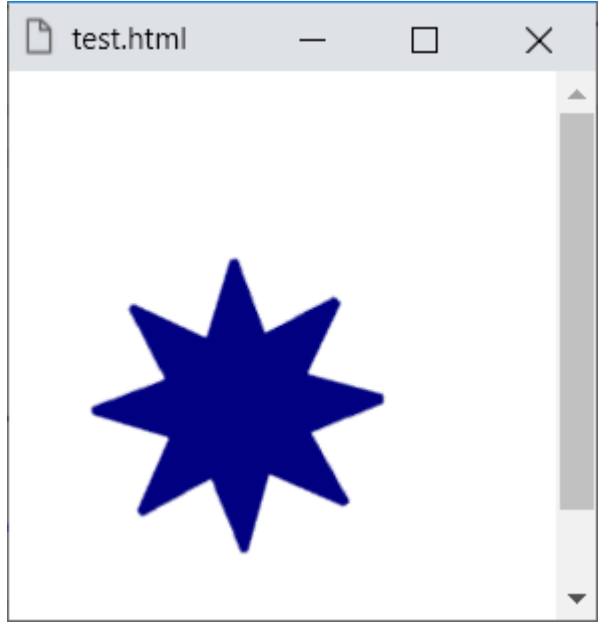
```
<!DOCTYPE html>
<html>
<head>
<style>
  body{ margin:30px; }

  #main{
    color:navy; font-size:150px; font-weight:bold;
    display:inline-block;
    animation: mymove 2s linear infinite alternate-reverse;
  }

  @keyframes mymove{
    0%{ transform:rotate(0deg); }
    25%{ transform:rotate(90deg); }
    50%{ transform:rotate(180deg); }
    75%{ transform:rotate(270deg); }
    100%{ transform:rotate(360deg); }
  }

</style>
</head>
<body>
  <div id="main">
    &#10040;
  </div>
</body>
</html>
```

وبعد تشغيل الصفحة ستلاحظ دورانها وانظر الى الشكل التالي:



كود! قم بكتابة الكود التالي ثم احفظه:

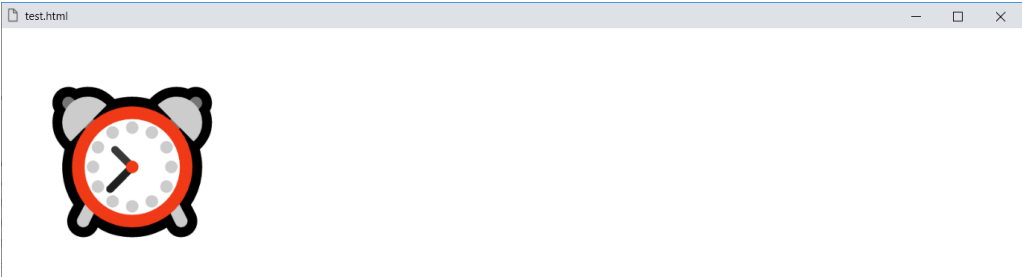
```
<!DOCTYPE html>
<html>
<head>
<style>
  body{ margin:30px; }

  #main{
    font-size:150px; font-weight:bold;
    display:inline-block;
    animation: mymove 50ms linear infinite alternate;
  }

  @keyframes mymove{
    from{ padding-left:0; }
    to{ padding-left:10px; }
  }

</style>
</head>
<body>
  <div id="main">
    &#9200;
  </div>
</body>
</html>
```

وبعد تشغيل الصفحة ستلاحظ أنها تهتز مثل المنبه وانظر الشكل التالي:

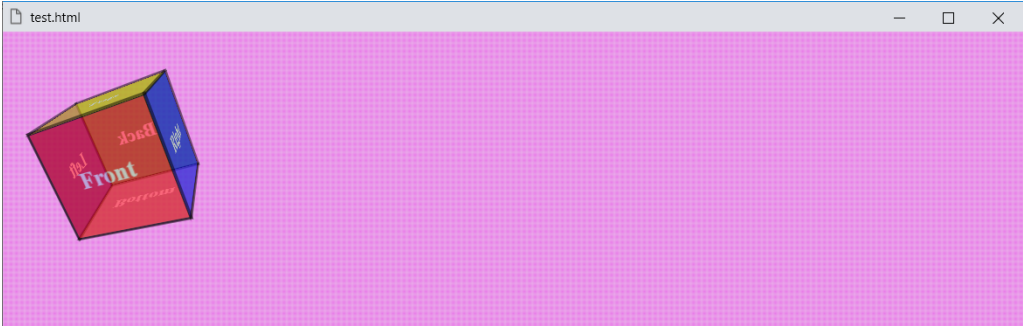
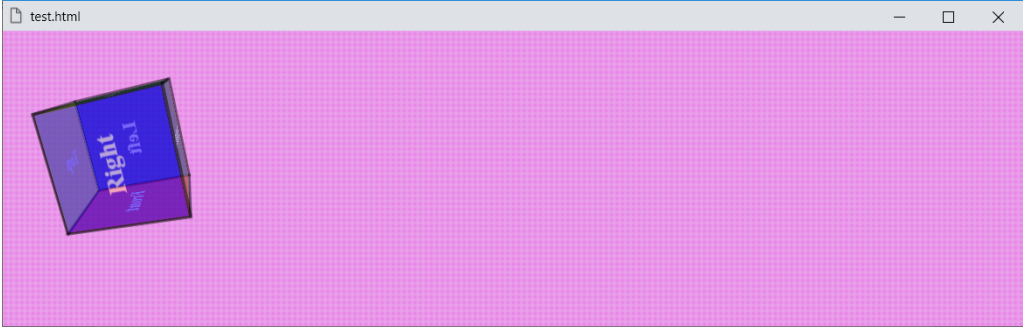
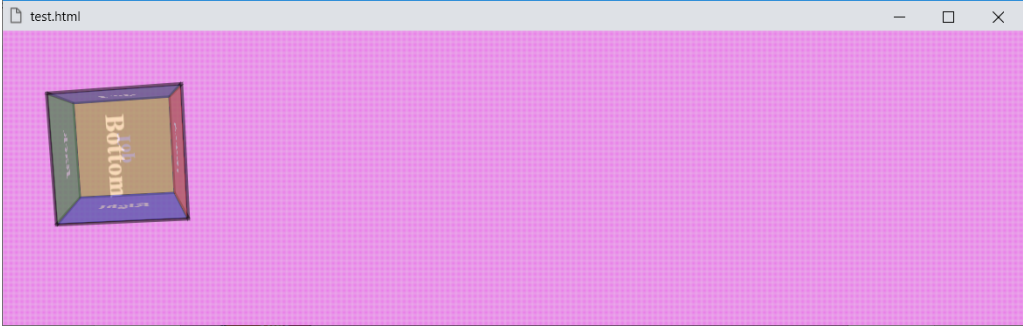


كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
body{ background:radial-gradient(#ea7cea, #e8aee8) left/5px 5px; }
.view{ width:100px; height:100px; perspective:300px; padding:50px; }
.cube{
width:100px; height:100px; transform-style:preserve-3d;
transform: rotate3d(1, 1, 1, 0deg);
animation: mymove 2s linear infinite;
}
.cube div{
width:100px; height:100px; opacity:0.5; display:inline-block;
line-height:100px; font-size:20px; font-weight:bold;
text-align:center; color:#fff; position:absolute;
border:2px solid black;
}
.cube .front{background:red;transform:rotateY(0deg) translateZ(50px);}
.cube .back{background:green;transform:rotateY(180deg) translateZ(50px);}
.cube .right{background:blue;transform:rotateY(90deg) translateZ(50px);}
.cube .left{background:navy;transform:rotateY(-90deg) translateZ(50px);}
.cube .top{background:gold;transform:rotateX(90deg) translateZ(50px);}
.cube .bottom{background:gray;transform:rotateX(-90deg) translateZ(50px);}

@keyframes mymove{
0%{ transform: rotate3d(1, 1, 1, 0deg); }
25%{ transform: rotate3d(1, 1, 1, 90deg); }
50%{ transform: rotate3d(1, 1, 1, 180deg); }
75%{ transform: rotate3d(1, 1, 1, 270deg); }
100%{ transform: rotate3d(1, 1, 1, 360deg); }
}
</style>
</head>
<body>
<div class="view">
<div class="cube">
<div class="front">Front</div>
<div class="back">Back</div>
<div class="right">Right</div>
<div class="left">Left</div>
<div class="top">Top</div>
<div class="bottom">Bottom</div>
</div>
</div>
</body>
</html>
```

وءءء ءءءءل الصفاءة ٭سءءءر ءءك 3d ءما بالأسءال الءالفة:

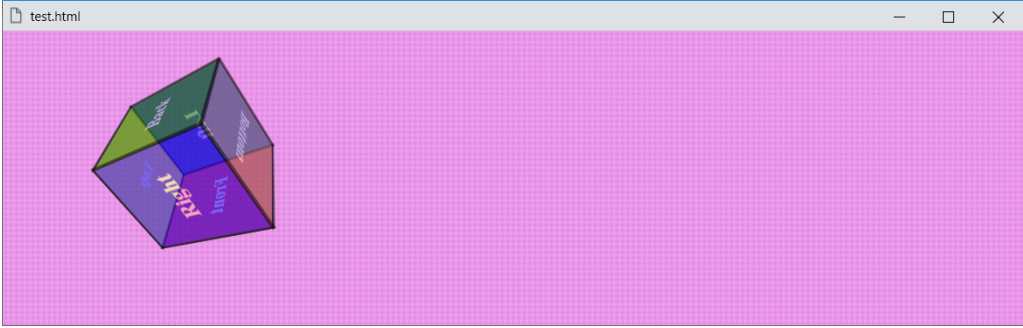


كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
body{ background:radial-gradient(#ea7cea, #e8aee8) left/5px 5px; }
.view{ width:100px; height:100px; perspective:300px; padding:50px; }
.cube{
width:100px; height:100px; transform-style:preserve-3d;
transform: rotate3d(1, 1, 1, 0deg); position:absolute;
animation: mymove 5s linear infinite alternate;
}
.cube div{
width:100px; height:100px; opacity:0.5; display:inline-block;
line-height:100px; font-size:20px; font-weight:bold;
text-align:center; color:#fff; position:absolute; left:10px;
border:2px solid black;
}
.cube .front{background:red;transform:rotateY(0deg) translateZ(50px);}
.cube .back{background:green;transform:rotateY(180deg) translateZ(50px);}
.cube .right{background:blue;transform:rotateY(90deg) translateZ(50px);}
.cube .left{background:navy;transform:rotateY(-90deg) translateZ(50px);}
.cube .top{background:gold;transform:rotateX(90deg) translateZ(50px);}
.cube .bottom{background:gray;transform:rotateX(-90deg) translateZ(50px);}

@keyframes mymove{
0%{ transform: rotate3d(1, 1, 1, 0deg); left:10px; }
25%{ transform: rotate3d(1, 1, 1, 90deg); left:60px; }
50%{ transform: rotate3d(1, 1, 1, 180deg); left:110px; }
75%{ transform: rotate3d(1, 1, 1, 270deg); left:160px; }
100%{ transform: rotate3d(1, 1, 1, 360deg); left:210px; }
}
</style>
</head>
<body>
<div class="view">
<div class="cube">
<div class="front">Front</div>
<div class="back">Back</div>
<div class="right">Right</div>
<div class="left">Left</div>
<div class="top">Top</div>
<div class="bottom">Bottom</div>
</div>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة يستدير ويتحرك شكل 3d كما بالأشكال التالية:



والآن انتبه!، فسوف نمنح صور لأرقام Digital على برنامج الفوتوشوب، ولو أردت تعلمه ابحث في يوتيوب بـ فوتوشوب حسونة أكاديمي، أو أدخل على قائمة التشغيل مع الرابط التالي:

<https://www.youtube.com/playlist?list=PLHIFW1KZRIfmtLWY0JQKyfR5RcuXdzfnU>

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{ margin:30px; }

  #main{
    width:250px; height:408px; padding:1px;
    box-shadow:2px 2px 60px yellow;
    background-image:url('base.jpg');
    animation: mymove 20s infinite alternate;
  }

  @keyframes mymove{
    0%{ box-shadow:2px 2px 60px yellow; background-image:url('base.jpg'); }
    10%{ box-shadow:2px 2px 60px aqua; background-image:url('0.jpg'); }
    20%{ box-shadow:2px 2px 60px green; background-image:url('1.jpg'); }
    30%{ box-shadow:2px 2px 60px red; background-image:url('2.jpg'); }
    40%{ box-shadow:2px 2px 60px blue; background-image:url('3.jpg'); }
    50%{ box-shadow:2px 2px 60px navy; background-image:url('4.jpg'); }
    60%{ box-shadow:2px 2px 60px gold; background-image:url('5.jpg'); }
    70%{ box-shadow:2px 2px 60px brown; background-image:url('6.jpg'); }
    80%{ box-shadow:2px 2px 60px darkred; background-image:url('7.jpg'); }
    90%{ box-shadow:2px 2px 60px darkgreen; background-image:url('8.jpg'); }
    100%{ box-shadow:2px 2px 60px gray; background-image:url('9.jpg'); }
  }

</style>
</head>
<body>
  <div id="main"></div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالأشكال التالية:



وهكذا يتم العد من 0 الى 9 ثم يرجع من 9 الى 0 الى ما لا نهاية.

والآن سوف نستخدم نفس آلية المثال السابق لصنع مثال آخر به خاتمه،
 فيتم العد من 1 الى 99 ثم يبدأ من جديد وهكذا.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{ margin:30px; }

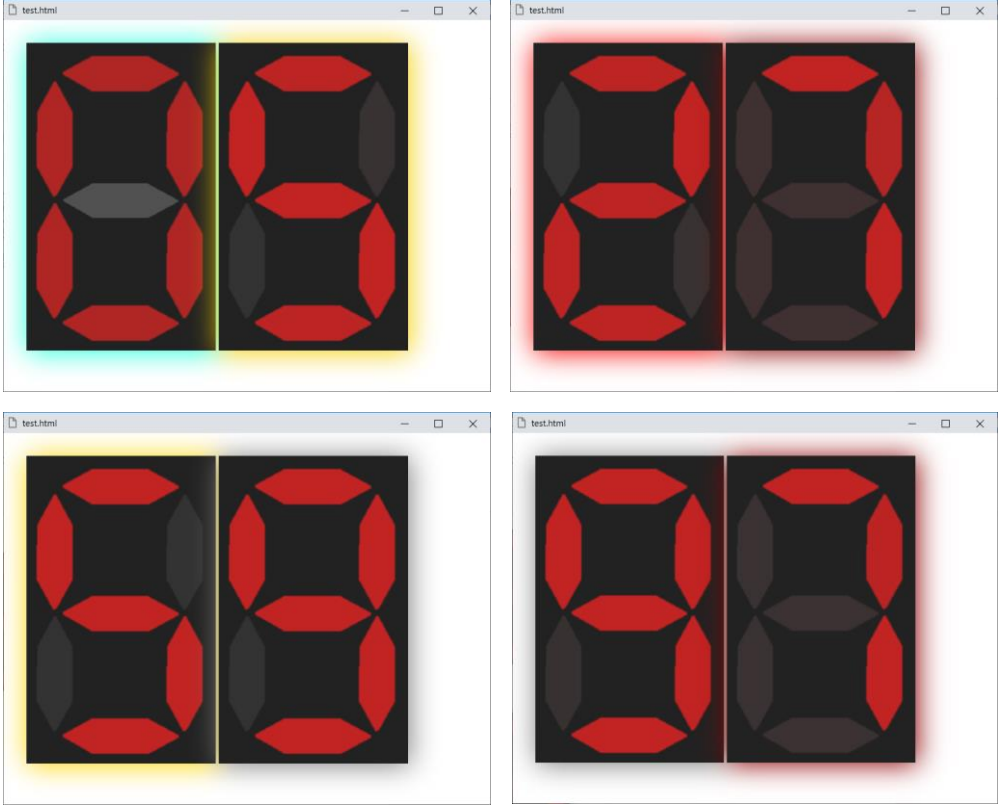
  #d1 , #d2{
    width:250px; height:408px; padding:1px;
    box-shadow:2px 2px 60px yellow; display:inline-block;
    background-image:url('base.jpg');
  }

  #d1{      animation: mymove 100s infinite; }
  #d2{      animation: mymove 10s infinite; }

  @keyframes mymove{
    0%{ box-shadow:2px 2px 60px yellow; background-image:url('base.jpg'); }
    10%{ box-shadow:2px 2px 60px aqua; background-image:url('0.jpg'); }
    20%{ box-shadow:2px 2px 60px green; background-image:url('1.jpg'); }
    30%{ box-shadow:2px 2px 60px red; background-image:url('2.jpg'); }
    40%{ box-shadow:2px 2px 60px blue; background-image:url('3.jpg'); }
    50%{ box-shadow:2px 2px 60px navy; background-image:url('4.jpg'); }
    60%{ box-shadow:2px 2px 60px gold; background-image:url('5.jpg'); }
    70%{ box-shadow:2px 2px 60px brown; background-image:url('6.jpg'); }
    80%{ box-shadow:2px 2px 60px darkred; background-image:url('7.jpg'); }
    90%{ box-shadow:2px 2px 60px darkgreen; background-image:url('8.jpg'); }
    100%{ box-shadow:2px 2px 60px gray; background-image:url('9.jpg'); }
  }

</style>
</head>
<body>
  <div id="d1"></div>
  <div id="d2"></div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالأشكال التالية:



وهكذا حتى يصل الى 99 ثم يبدأ منه جديد.

والآن سوف نصنع خانات يتم العد فيها من 1 الى 999 ثم يبدأ من جديد، وهكذا، وذلك كما يلي إن شاء الله.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{ margin:30px; }

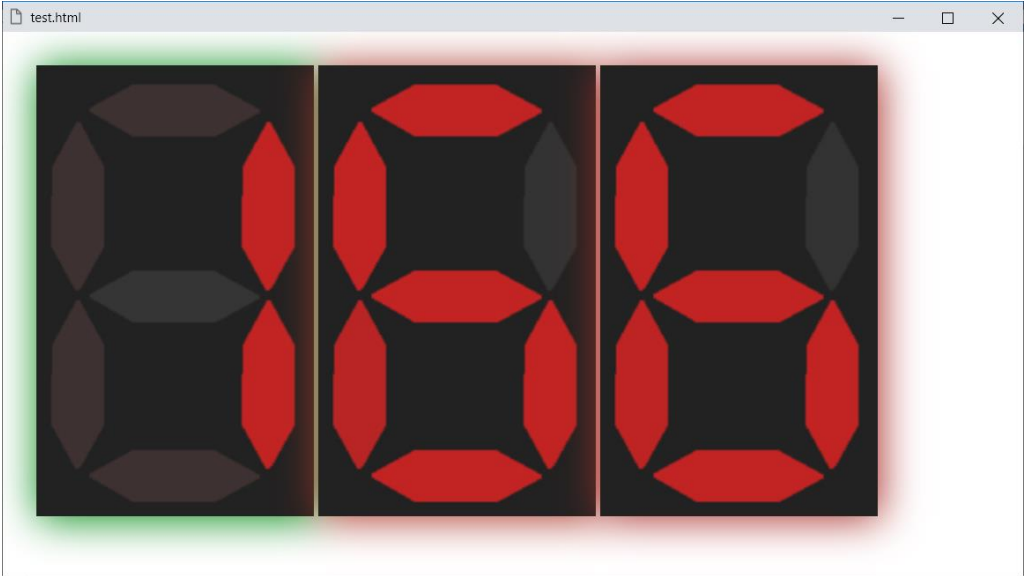
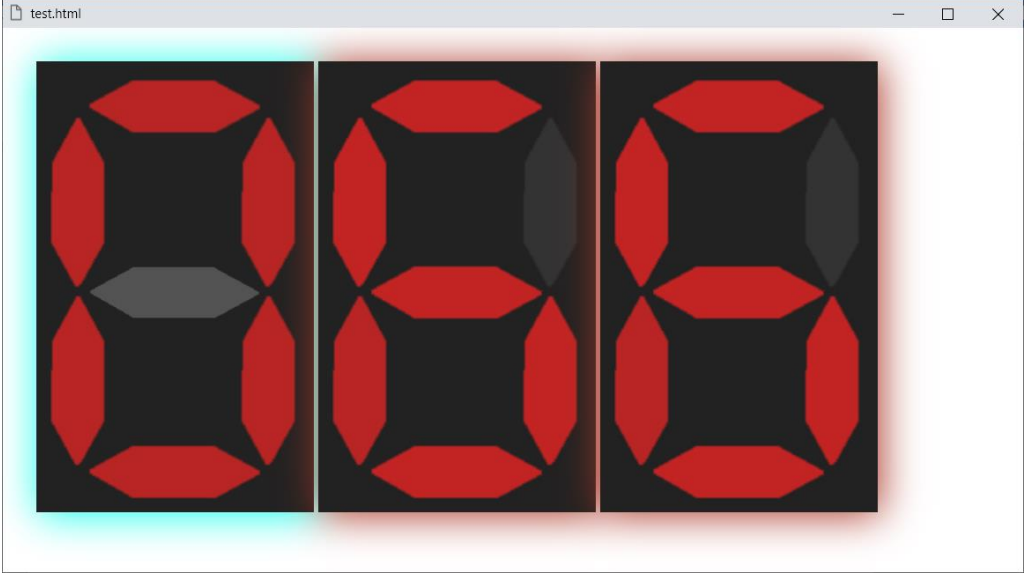
  #d1 , #d2 , #d3{
    width:250px; height:408px; padding:1px;
    box-shadow:2px 2px 60px yellow; display:inline-block;
    background-image:url('base.jpg');
  }

  #d1{      animation: mymove 1000s infinite; }
  #d2{      animation: mymove 100s infinite; }
  #d3{      animation: mymove 10s infinite; }

  @keyframes mymove{
    0%{ box-shadow:2px 2px 60px yellow; background-image:url('base.jpg'); }
    10%{ box-shadow:2px 2px 60px aqua; background-image:url('0.jpg'); }
    20%{ box-shadow:2px 2px 60px green; background-image:url('1.jpg'); }
    30%{ box-shadow:2px 2px 60px red; background-image:url('2.jpg'); }
    40%{ box-shadow:2px 2px 60px blue; background-image:url('3.jpg'); }
    50%{ box-shadow:2px 2px 60px navy; background-image:url('4.jpg'); }
    60%{ box-shadow:2px 2px 60px gold; background-image:url('5.jpg'); }
    70%{ box-shadow:2px 2px 60px brown; background-image:url('6.jpg'); }
    80%{ box-shadow:2px 2px 60px darkred; background-image:url('7.jpg'); }
    90%{ box-shadow:2px 2px 60px darkgreen; background-image:url('8.jpg'); }
    100%{ box-shadow:2px 2px 60px gray; background-image:url('9.jpg'); }
  }

</style>
</head>
<body>
  <div id="d1"></div>
  <div id="d2"></div>
  <div id="d3"></div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالأشكال التالية:



وهكذا حتى يصل الى 999 ثم يبدأ منه جديد.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  #myname:before{
    content:""; animation:myset 3s linear infinite;
  }

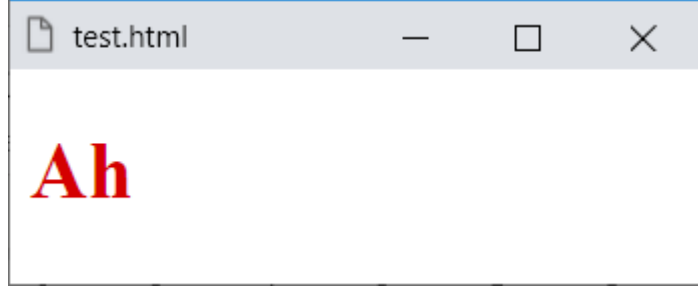
  @keyframes myset{
    0%{ content:""; color:#f00; }
    7%{ content:"A"; color:#000; }
    14%{ content:"Ah"; color:#f00; }
    21%{ content:"Ahm"; color:#000; }
    28%{ content:"Ahme"; color:#f00; }
    35%{ content:"Ahmed"; color:#000; }
    42%{ content:"Ahmed_"; color:#f00; }
    49%{ content:"Ahmed_H"; color:#000; }
    56%{ content:"Ahmed_Ha"; color:#f00; }
    63%{ content:"Ahmed_Has"; color:#000; }
    70%{ content:"Ahmed_Hass"; color:#f00; }
    77%{ content:"Ahmed_Hasso"; color:#000; }
    84%{ content:"Ahmed_Hassou"; color:#f00; }
    91%{ content:"Ahmed_Hassoun"; color:#000; }
    100%{ content:"Ahmed_Hassouna"; color:#f00; }
  }

</style>
</head>
<body>

  <h1 id="myname"><h1>

</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالأشكال التالية:



وهكذا يكتب الاسم بالتسلسل حتى يتم كتابته ثم يعيد من جديد.

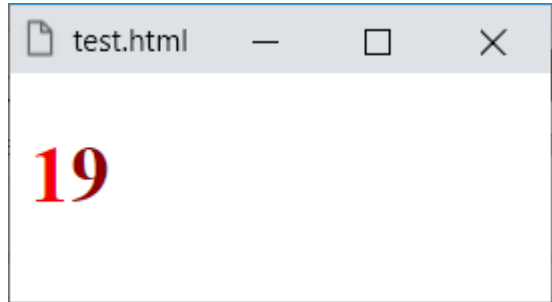
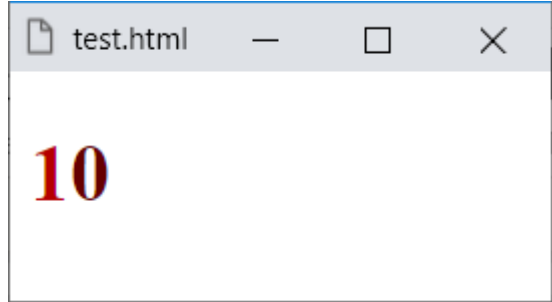
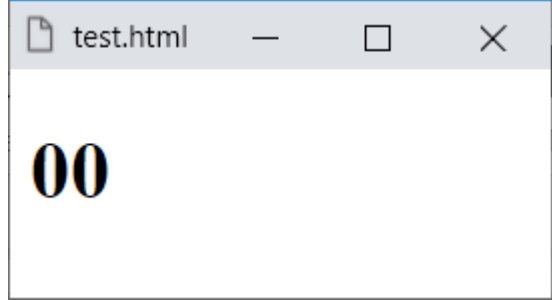
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  #mynum:before{
    content:"0";
    animation:myset 100s cubic-bezier(0,1.42,.83,.78) infinite;
  }

  #mynum:after{
    content:"0";
    animation:myset 10s infinite;
  }

  @keyframes myset{
    10%{ color:#000; content:"0"; }
    20%{ color:#f00; content:"1"; }
    30%{ color:#000; content:"2"; }
    40%{ color:#f00; content:"3"; }
    50%{ color:#000; content:"4"; }
    60%{ color:#f00; content:"5"; }
    70%{ color:#000; content:"6"; }
    80%{ color:#f00; content:"7"; }
    90%{ color:#000; content:"8"; }
    100%{color:#f00; content:"9"; }
  }
</style>
</head>
<body>
  <h1 id="mynum"><h1>
</body>
</html>
```

وءءء ءشءعل الصءفة ءظهء كما بالأشكال ءالفة:



وهءذا يظل بعء ءء 1 وءءى ىءءهى الى 99 ءم ىبءأ العء ءء البءاءة.

الأدوات Tools

عند العمل نحتاج كثيراً الى أدوات يتم وضعها على الشاشة، وفي هذا الموضوع سوف نقوم بعمل بعض الأدوات البسيطة.

والآن سوف نقوم بعمل ملف CSS باسم styles.css ثم يتم الإضافة فيه وتعديله واستخدامه، ورجاء **انتبه!**، فعندما يتم اخبارك بأن تضع أي كود داخل ملف styles.css ينبغي أن لا تنسى أننا سوف نستخدم كل ما فيه ان شاء الله، وأيضاً **انتبه!**، فله يتم الشرح كثيراً في هذا الموضوع، فالاعتماد على ما سبقه من شرح.

ورجاء **انتبه!** ثم **انتبه!** ثم **انتبه!**، فإن هذا الموضوع الغرض منه التدريب، والتعامل مع التجارب، وجعل الطالب أكثر جرئته مع الكود.

مربع النص Text Box

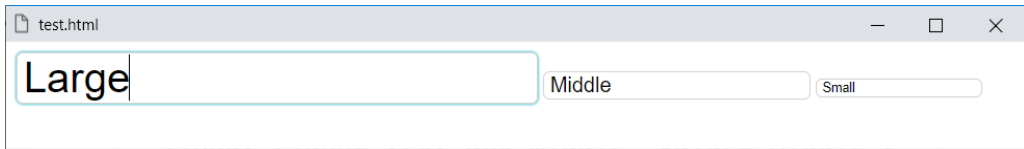
هذه الكود! التالي في ملف `styles.css` ثم احفظه:

```
.text-box-small ,  
.text-box-middle ,  
.text-box-large{  
    display:inline-block; border-radius:5px;  
    padding-left:5px; padding-right:5px;  
    border:0.5px solid lightgray;  
    transition:all 0.5s; outline:none;  
}  
  
.text-box-small:focus ,  
.text-box-middle:focus ,  
.text-box-large:focus{  
    box-shadow:0.5px 0.5px 2px #7ed4db,  
              -0.5px -0.5px 2px #7ed4db;  
}  
  
.text-box-small{  
    font-size:10px;  
}  
  
.text-box-middle{  
    font-size:16px;  
}  
  
.text-box-large{  
    font-size:32px;  
}
```


كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>
    <input class="text-box-large" value="Large">
    <input class="text-box-middle" value="Middle">
    <input class="text-box-small" value="Small">
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



ءنواء الشاشة Label

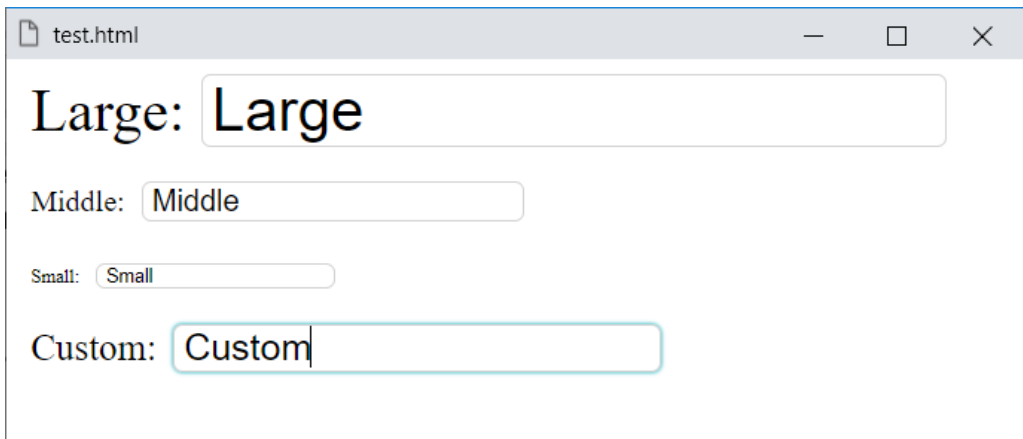
ءءء الكوء! الءالء فف مءف `styles.css` ءم اءفظه:

```
.label-small ,  
.label-middle ,  
.label-large{  
  display:inline-block;  
  padding-left:5px; padding-right:5px;  
}  
  
.label-small{  
  font-size:10px;  
}  
  
.label-middle{  
  font-size:16px;  
}  
  
.label-large{  
  font-size:32px;  
}
```

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
  <style>
    .cust{
      font-size:20px;
    }
  </style>
</head>
<body>
  <div class="label-large">Large:</div>
  <input class="text-box-large" value="Large">
  <br><br>
  <div class="label-middle">Middle:</div>
  <input class="text-box-middle" value="Middle">
  <br><br>
  <div class="label-small">Small:</div>
  <input class="text-box-small" value="Small">
  <br><br>
  <div class="label-small cust">Custom:</div>
  <input class="text-box-small cust" value="Custom">
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



الأزرار Buttons

هذه الكود! التالي في ملف `styles.css` ثم احفظه:

```
.button-small ,
.button-middle ,
.button-large{
    display:inline-block; border-radius:5px;
    padding:5px 20px; cursor:pointer;
    border:0.5px solid lightgray; outline:none;
    background-color:#efefef;
    transition:all 0.5s;
}

.button-small:hover ,
.button-middle:hover ,
.button-large:hover{
    box-shadow:0.5px 0.5px 2px #7ed4db,
                -0.5px -0.5px 2px #7ed4db;
}

.button-small{
    font-size:10px;
}

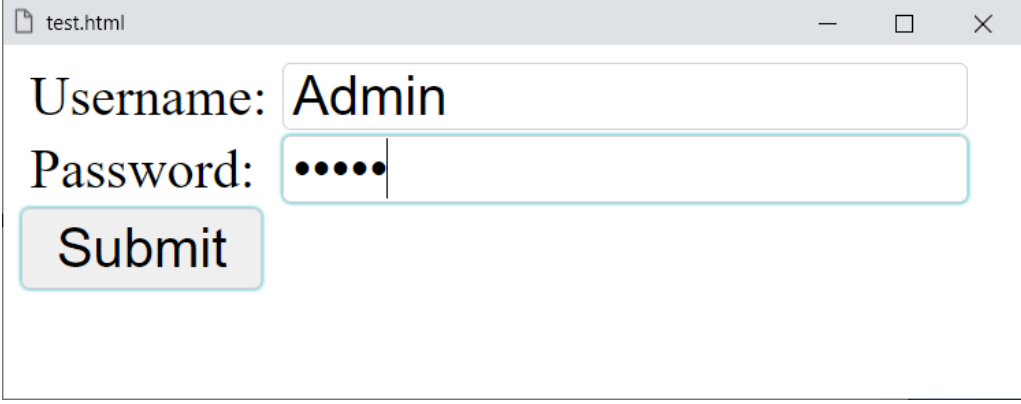
.button-middle{
    font-size:16px;
}

.button-large{
    font-size:32px;
}
```

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>
    <table>
      <tr>
        <td>
          <div class="label-large">Username:</div>
        </td>
        <td>
          <input class="text-box-large">
        </td>
      </tr>
      <tr>
        <td>
          <div class="label-large">Password:</div>
        </td>
        <td>
          <input type="password" class="text-box-large">
        </td>
      </tr>
      <tr>
        <td>
          <input type="submit" class="button-large">
        </td>
      </tr>
    </table>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



The screenshot shows a web browser window titled "test.html". Inside the window, there is a form with the following elements:

- A label "Username:" followed by a text input field containing the text "Admin".
- A label "Password:" followed by a password input field containing five dots ".....".
- A button labeled "Submit" below the password field.

اللوحة Panel

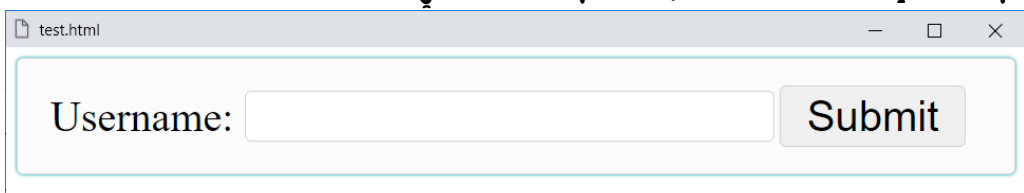
منع الكود! التالي في ملف `styles.css` ثم احفظه:

```
.panel{
  display:block; border-radius:5px;
  margin-top:7px; margin-bottom:7px;
  padding:20px; outline:none;
  border:0.5px solid lightgray;
  background-color:#fbfbfb;
  overflow:auto;
  box-shadow:0.5px 0.5px 2px #7ed4db,
             -0.5px -0.5px 2px #7ed4db;
  transition:all 0.5s;
}
```

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>
    <div class="panel">
      <div class="label-large">Username:</div>
      <input class="text-box-large">
      <input type="submit" class="button-large">
    </div>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



صندوق الاختيار Check Box

منع الكود! التالي في ملف `styles.css` ثم احفظه:

```
input[type=checkbox].checkbox + label{
  display:block;
  margin:0.3em;
}

input[type=checkbox].checkbox{
  display:none;
}

input[type=checkbox].checkbox + label:before{
  display:inline-block;
  content:"\2714";
  border:0.1em solid black;
  border-radius:0.2em;
  width:1em; height:1em;
  padding-left:0.2em;
  padding-bottom:0.4em;
  margin-right:0.2em;
  color:transparent;
  transition:all 0.5s;
}

input[type=checkbox].checkbox:checked + label:before{
  background-color:#aaa;
  color:#fff;
  box-shadow:0.5px 0.5px 2px #7ed4db,
             -0.5px -0.5px 2px #7ed4db;
}

input[type=checkbox].checkbox:not(:checked):hover + label:before{
  box-shadow:0.5px 0.5px 2px #7ed4db,
             -0.5px -0.5px 2px #7ed4db;
}
```


كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

  <div class="panel">
    <input class="checkbox" type="checkbox" id="java">
    <label for="java">JAVA</label>

    <input class="checkbox" type="checkbox" id="csharp">
    <label for="csharp">C Sharp</label>

    <input class="checkbox" type="checkbox" id="ring">
    <label for="ring">Ring</label>
  </div>

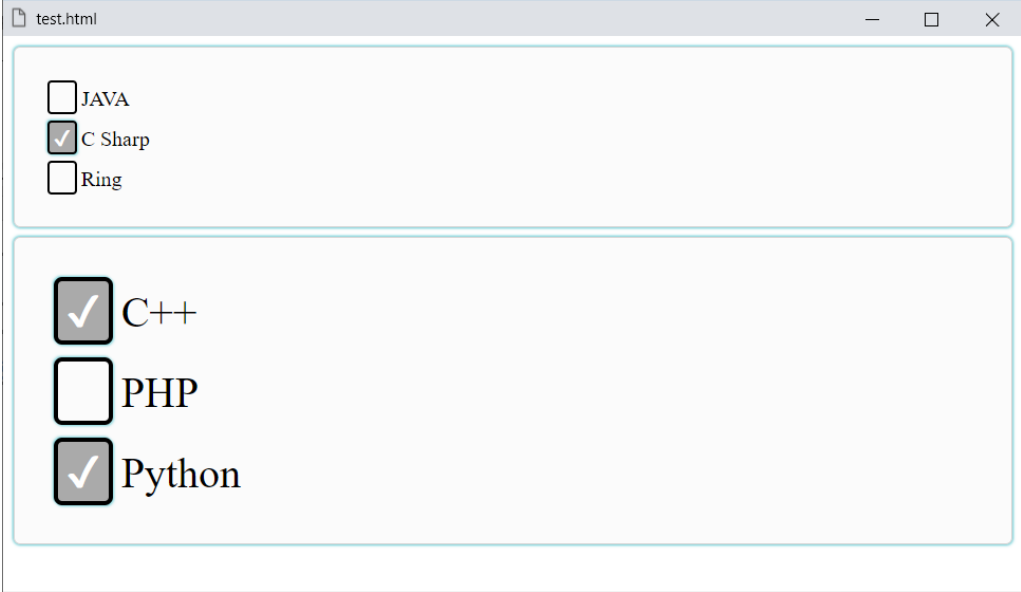
  <div class="panel" style="font-size:32px;">
    <input class="checkbox" type="checkbox" id="cpp">
    <label for="cpp">C++</label>

    <input class="checkbox" type="checkbox" id="php">
    <label for="php">PHP</label>

    <input class="checkbox" type="checkbox" id="python">
    <label for="python">Python</label>
  </div>

</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



The screenshot shows a web browser window titled "test.html". The page contains two sections of checkboxes for programming languages. The first section has three options: JAVA (unchecked), C Sharp (checked), and Ring (unchecked). The second section has three options: C++ (checked), PHP (unchecked), and Python (checked).

<input type="checkbox"/>	JAVA
<input checked="" type="checkbox"/>	C Sharp
<input type="checkbox"/>	Ring

<input checked="" type="checkbox"/>	C++
<input type="checkbox"/>	PHP
<input checked="" type="checkbox"/>	Python

صندوق الراديو Radio

منع الكود! التالي في ملف `styles.css` ثم احفظه:

```
input[type=radio].radio + label{
  display:block;
  margin:0.3em;
}

input[type=radio].radio{
  display:none;
}

input[type=radio].radio + label:before{
  display:inline-block;
  content:"\2b24";
  border:0.1em solid black;
  border-radius:0.9em;
  width:1em; height:1em;
  padding-left:0.3em;
  padding-right:0.1em;
  padding-bottom:0.4em;
  margin-right:0.3em;
  color:transparent;
  transition:all 0.5s;
}

input[type=radio].radio:checked + label:before{
  background-color:#aaa;
  color:#fff;
  box-shadow:0.9px 0.9px 5px #7ed4db,
            -0.9px -0.9px 5px #7ed4db;
}

input[type=radio].radio:not(:checked):hover + label:before{
  box-shadow:0.9px 0.9px 5px #7ed4db,
            -0.9px -0.9px 5px #7ed4db;
}
```

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

  <div class="panel">
    <input class="radio" type="radio" id="java" name="lang1">
    <label for="java">JAVA</label>

    <input class="radio" type="radio" id="csharp" name="lang1">
    <label for="csharp">C Sharp</label>

    <input class="radio" type="radio" id="ring" name="lang1">
    <label for="ring">Ring</label>
  </div>

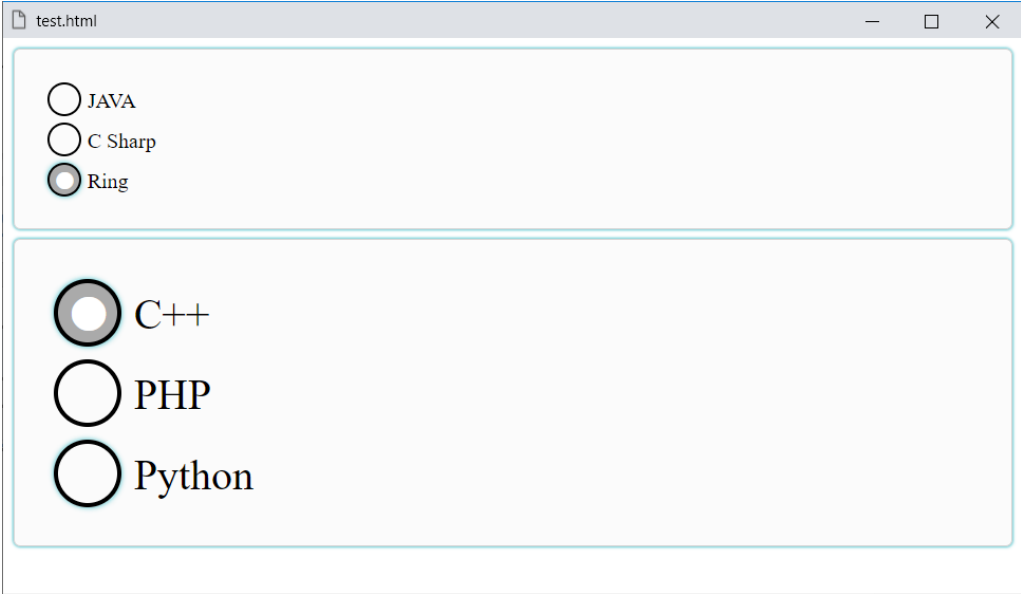
  <div class="panel" style="font-size:32px;">
    <input class="radio" type="radio" id="cpp" name="lang2">
    <label for="cpp">C++</label>

    <input class="radio" type="radio" id="php" name="lang2">
    <label for="php">PHP</label>

    <input class="radio" type="radio" id="python" name="lang2">
    <label for="python">Python</label>
  </div>

</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



The screenshot shows a browser window titled "test.html" with a form containing two sections of radio buttons. The first section has three options: JAVA, C Sharp, and Ring, with Ring selected. The second section has three options: C++, PHP, and Python, with C++ selected.

JAVA
 C Sharp
 Ring

C++
 PHP
 Python

شريط الاختيار Slider

منع الكود! التالي في ملف `styles.css` ثم احفظه:

```
.slider{
background: #eee;
-webkit-appearance: none;
height: 30px;
border:0.1px solid gray;
outline: none;
transition:all 0.5s;
}

.slider:hover{
box-shadow:0.5px 0.5px 2px #7ed4db,
-0.5px -0.5px 2px #7ed4db;
}

.slider::-webkit-slider-thumb{
-webkit-appearance: none;
width:30px; height:30px;
appearance:none;
background:#aaa;
cursor:pointer;
}

.slider::-moz-range-thumb {
width:30px; height:30px;
background:#aaa;
cursor:pointer;
}
```

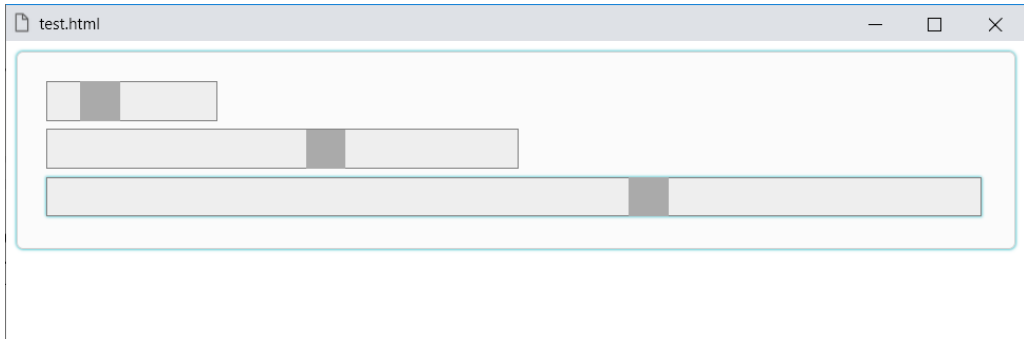
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

  <div class="panel">
    <input class="slider" type="range"><br>
    <input class="slider" type="range" style="width:50%"><br>
    <input class="slider" type="range" style="width:99%"><br>
  </div>

</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



صندوق الاختيار Combo Box

نص الكود! التالي في ملف `styles.css` ثم احفظه:

```
select.combobox{
  display:inline-block; border-radius:5px;
  padding:0.2em;
  border:0.5px solid lightgray;
  transition:all 0.5s; outline:none;
}

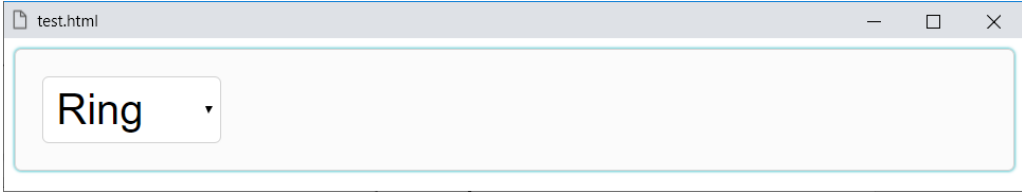
select.combobox > option{
  background:#efefef;
}

select.combobox:focus{
  box-shadow:0.5px 0.5px 2px #7ed4db,
             -0.5px -0.5px 2px #7ed4db;
}
```

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="panel" >
    <select class="combobox" style="font-size:32px;">
      <option>Ring</option>
      <option>Kotlin</option>
      <option>Python</option>
    </select>
  </div>
</body>
</html>
```


وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

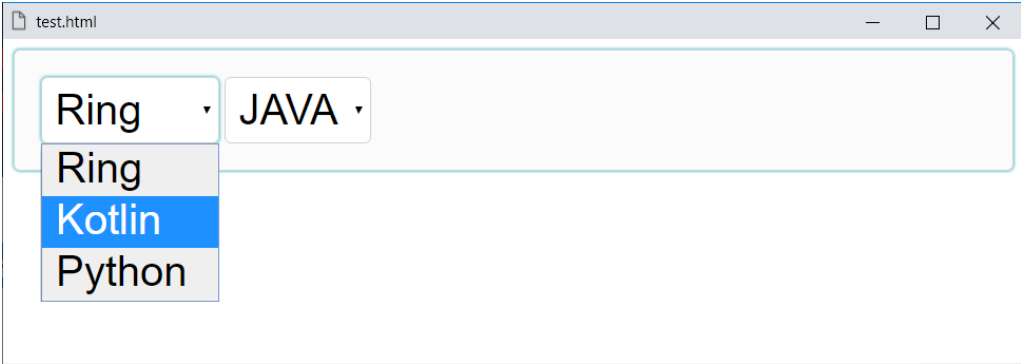
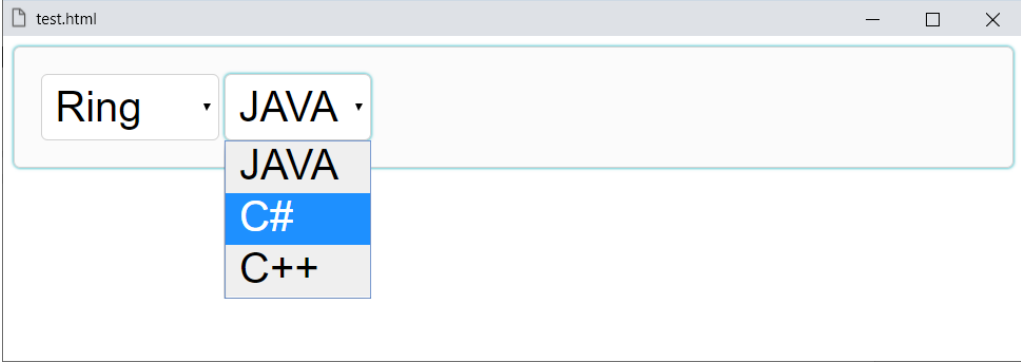


كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="panel" >
    <select class="combobox" style="font-size:32px;">
      <option>Ring</option>
      <option>Kotlin</option>
      <option>Python</option>
    </select>

    <select class="combobox" style="font-size:32px;">
      <option>JAVA</option>
      <option>C#</option>
      <option>C++</option>
    </select>
  </div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالأشكال التالية:



صندوق المفتاح Toggle

نسخ الكود! التالي في ملف `styles.css` ثم احفظه:

```
input[type=checkbox].toggle + label{
  display:block;
  margin:0.3em;
}

input[type=checkbox].toggle{
  display:none;
}

input[type=checkbox].toggle + label:before{
  width:2em; height:1em;
  text-align:left;
  display:inline-block;
  content:"\2b24";
  border:0.1em solid black;
  border-radius:1em;
  padding-left:0.2em;
  padding-right:0.2em;
  padding-bottom:0.4em;
  margin-right:0.2em;
  background-color:#efefef;
  color:#bbb;
  transition:all 0.5s;
}

input[type=checkbox].toggle:checked + label:before{
  text-align:right;
  background-color:#aaa;
  color:#fff;
  box-shadow:0.5px 0.5px 2px #7ed4db,
            -0.5px -0.5px 2px #7ed4db;
}

input[type=checkbox].toggle:not(:checked):hover + label:before{
  box-shadow:0.5px 0.5px 2px #7ed4db,
            -0.5px -0.5px 2px #7ed4db;
}
```

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

  <div class="panel">
    <input class="toggle" type="checkbox" id="java">
    <label for="java">JAVA</label>

    <input class="toggle" type="checkbox" id="csharp">
    <label for="csharp">C Sharp</label>

    <input class="toggle" type="checkbox" id="ring">
    <label for="ring">Ring</label>
  </div>

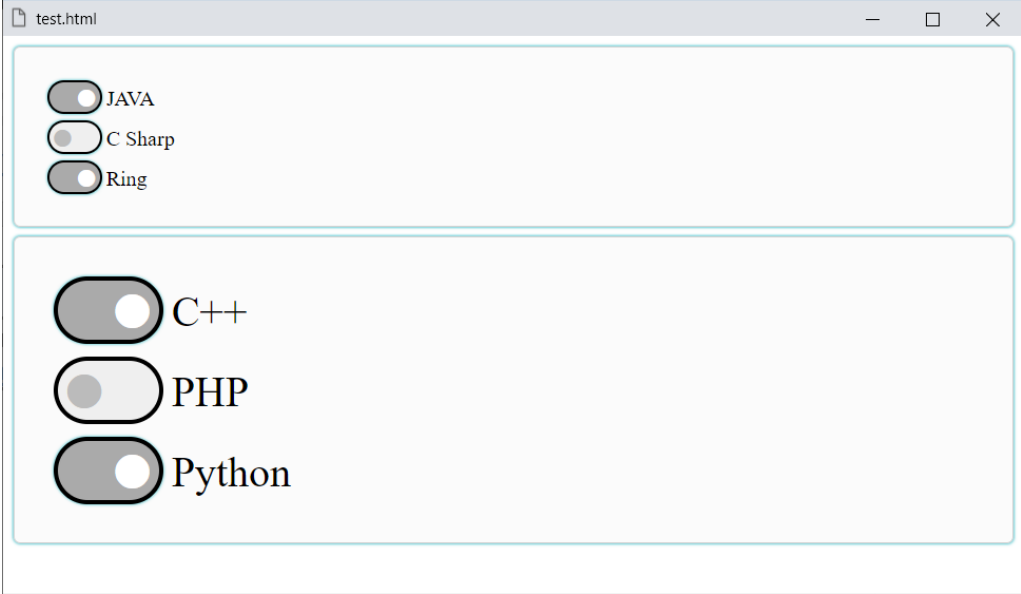
  <div class="panel" style="font-size:32px;">
    <input class="toggle" type="checkbox" id="cpp">
    <label for="cpp">C++</label>

    <input class="toggle" type="checkbox" id="php">
    <label for="php">PHP</label>

    <input class="toggle" type="checkbox" id="python">
    <label for="python">Python</label>
  </div>

</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



أداة العداد المخصص Custom Counter

منع الكود! التالي في ملف `styles.css` ثم احفظه:

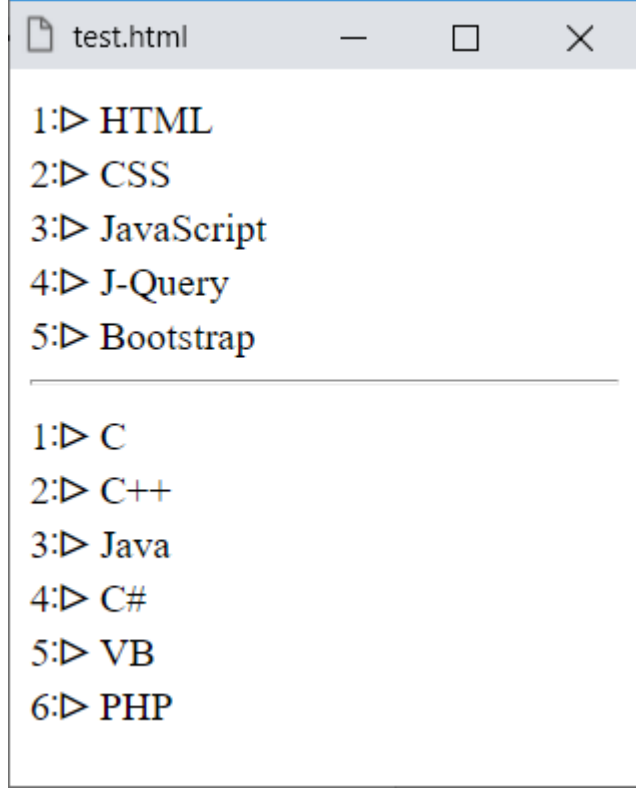
```
.counter-reset {  
  counter-reset:mycounter;  
}  
  
.counter .counter-item::before{  
  counter-increment:mycounter;  
  content:counter(mycounter) "\2236\1405 ";  
}
```

ومع الكود السابق نلاحظ أنه تم عمل عداد counter في الـ CSS باسم mycounter، وتم عمل كلاس counter-reset والذي يقوم بجعله يبدأ العد من البداية لاستخدامه عند الحاجة لذلك لبداية ترميز جديد، وتم استخدام counter-increment ليزيد مع كل عنصر، وتم استخدام counter وبداخل اقواسها العداد mycounter ليتم طباعته بالرقم الذي وصل اليه، ثم تم استخدام رموز مخصصة أمامه، لتكون فاصل بين الرقم وبين محتوى العنصر نفسه، وسوف نجرب في المثال التالي إن شاء الله.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>
    <div class="counter-reset counter">
      <div class="counter-item">HTML</div>
      <div class="counter-item">CSS</div>
      <div class="counter-item">JavaScript</div>
      <div class="counter-item">J-Query</div>
      <div class="counter-item">Bootstrap</div>
    </div>
    <hr>
    <div class="counter-reset counter">
      <div class="counter-item">C</div>
      <div class="counter-item">C++</div>
      <div class="counter-item">Java</div>
      <div class="counter-item">C#</div>
      <div class="counter-item">VB</div>
      <div class="counter-item">PHP</div>
    </div>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



styles.css أكواد الملف

```
/* Start Text Box */
.text-box-small ,
.text-box-middle ,
.text-box-large{
display:inline-block; border-radius:5px;
padding-left:5px; padding-right:5px;
border:0.5px solid lightgray;
transition:all 0.5s; outline:none;
}

.text-box-small:focus ,
.text-box-middle:focus ,
.text-box-large:focus{
box-shadow:0.5px 0.5px 2px #7ed4db,
-0.5px -0.5px 2px #7ed4db;
}

.text-box-small{
font-size:10px;
}

.text-box-middle{
font-size:16px;
}

.text-box-large{
font-size:32px;
}
/* End Text Box */

/* Start Label */
.label-small ,
.label-middle ,
.label-large{
display:inline-block;
padding-left:5px; padding-right:5px;
}

.label-small{
font-size:10px;
}
```

```
.label-middle{
  font-size:16px;
}

.label-large{
  font-size:32px;
}
/* End Label */

/* Start Button */
.button-small ,
.button-middle ,
.button-large{
  display:inline-block; border-radius:5px;
  padding:5px 20px; cursor:pointer;
  border:0.5px solid lightgray; outline:none;
  background-color:#efefef;
  transition:all 0.5s;
}

.button-small:hover ,
.button-middle:hover ,
.button-large:hover{
  box-shadow:0.5px 0.5px 2px #7ed4db,
            -0.5px -0.5px 2px #7ed4db;
}

.button-small{
  font-size:10px;
}

.button-middle{
  font-size:16px;
}

.button-large{
  font-size:32px;
}
/* End Button */
```

```
/* Start Panel */
.panel{
display:block; border-radius:5px;
margin-top:7px; margin-bottom:7px;
padding:20px; outline:none;
border:0.5px solid lightgray;
background-color:#fbfbfb;
overflow:auto;
box-shadow:0.5px 0.5px 2px #7ed4db,
           -0.5px -0.5px 2px #7ed4db;
transition:all 0.5s;
}
/* End Panel */

/* Start Check Box */
input[type=checkbox].checkbox + label{
display:block;
margin:0.3em;
}

input[type=checkbox].checkbox{
display:none;
}

input[type=checkbox].checkbox + label:before{
display:inline-block;
content:"\2714";
border:0.1em solid black;
border-radius:0.2em;
width:1em; height:1em;
padding-left:0.2em;
padding-bottom:0.4em;
margin-right:0.2em;
color:transparent;
transition:all 0.5s;
}

input[type=checkbox].checkbox:checked + label:before{
background-color:#aaa;
color:#fff;
box-shadow:0.5px 0.5px 2px #7ed4db,
           -0.5px -0.5px 2px #7ed4db;
}
}
```

```
input[type=checkbox].checkbox:not(:checked):hover + label:before{
  box-shadow:0.5px 0.5px 2px #7ed4db,
             -0.5px -0.5px 2px #7ed4db;
}
/* End Check Box */

/* Start Radio */
input[type=radio].radio + label{
  display:block;
  margin:0.3em;
}

input[type=radio].radio{
  display:none;
}

input[type=radio].radio + label:before{
  display:inline-block;
  content:"\2b24";
  border:0.1em solid black;
  border-radius:0.9em;
  width:1em; height:1em;
  padding-left:0.3em;
  padding-right:0.1em;
  padding-bottom:0.4em;
  margin-right:0.3em;
  color:transparent;
  transition:all 0.5s;
}

input[type=radio].radio:checked + label:before{
  background-color:#aaa;
  color:#fff;
  box-shadow:0.9px 0.9px 5px #7ed4db,
             -0.9px -0.9px 5px #7ed4db;
}

input[type=radio].radio:not(:checked):hover + label:before{
  box-shadow:0.9px 0.9px 5px #7ed4db,
             -0.9px -0.9px 5px #7ed4db;
}
/* End Radio */
```

```
/* Start Slider */
.slider{
  background: #eee;
  -webkit-appearance: none;
  height: 30px;
  border:0.1px solid gray;
  outline: none;
  transition:all 0.5s;
}

.slider:hover{
  box-shadow:0.5px 0.5px 2px #7ed4db,
            -0.5px -0.5px 2px #7ed4db;
}

.slider::-webkit-slider-thumb{
  -webkit-appearance: none;
  width:30px; height:30px;
  appearance:none;
  background:#aaa;
  cursor:pointer;
}

.slider::-moz-range-thumb {
  width:30px; height:30px;
  background:#aaa;
  cursor:pointer;
}
/* End Slider */

/* Start Combo Box */
select.combobox{
  display:inline-block; border-radius:5px;
  padding:0.2em;
  border:0.5px solid lightgray;
  transition:all 0.5s; outline:none;
}

select.combobox > option{
  background:#efefef;
}
```

```
select.combobox:focus{
  box-shadow:0.5px 0.5px 2px #7ed4db,
             -0.5px -0.5px 2px #7ed4db;
}
/* End Combo Box */

/* Start Toggle */
input[type=checkbox].toggle + label{
  display:block;
  margin:0.3em;
}

input[type=checkbox].toggle{
  display:none;
}

input[type=checkbox].toggle + label:before{
  width:2em; height:1em;
  text-align:left;
  display:inline-block;
  content:"\2b24";
  border:0.1em solid black;
  border-radius:1em;
  padding-left:0.2em;
  padding-right:0.2em;
  padding-bottom:0.4em;
  margin-right:0.2em;
  background-color:#efefef;
  color:#bbb;
  transition:all 0.5s;
}

input[type=checkbox].toggle:checked + label:before{
  text-align:right;
  background-color:#aaa;
  color:#fff;
  box-shadow:0.5px 0.5px 2px #7ed4db,
             -0.5px -0.5px 2px #7ed4db;
}

input[type=checkbox].toggle:not(:checked):hover + label:before{
  box-shadow:0.5px 0.5px 2px #7ed4db,
             -0.5px -0.5px 2px #7ed4db;
}
```

```
}
/* End Toggle */

/* Start Counter */
.counter-reset {
  counter-reset:mycounter;
}

.counter .counter-item::before{
  counter-increment:mycounter;
  content:counter(mycounter) "\2236\1405 ";
}
/* End Counter */
```

وفيما سبق هي أكواد الملف styles.css بالكامل حتى الآن.

شاشة الأدوات

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="panel" style="font-size:32px;">
    <div class="label-large">Full Name:</div>
    <input class="text-box-large"> <br>

    <input class="radio" type="radio" id="male" name="gender">
    <label style="display:inline-block" for="male">Male</label>
    <input class="radio" type="radio" id="female" name="gender">
    <label style="display:inline-block" for="female">Female</label> <br>

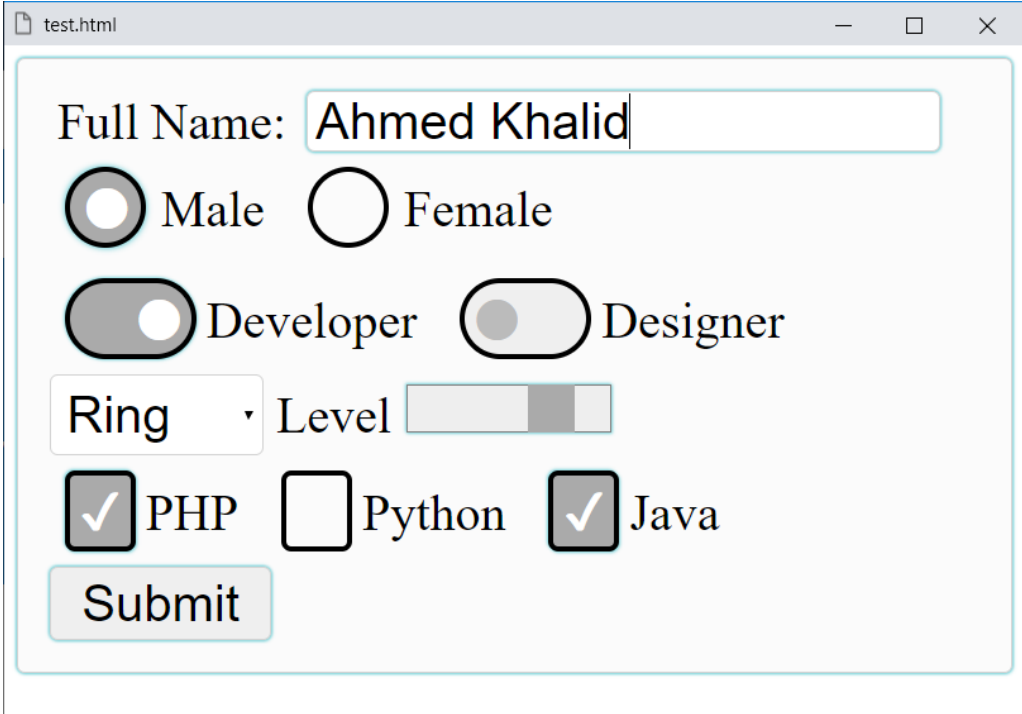
    <input class="toggle" type="checkbox" id="dev">
    <label style="display:inline-block" for="dev">Developer</label>
    <input class="toggle" type="checkbox" id="des">
    <label style="display:inline-block" for="des">Designer</label> <br>

    <select class="combobox" style="font-size:32px;">
      <option>Ring</option>
      <option>Kotlin</option>
      <option>Python</option>
    </select>
    <label>Level<label> <input type="range" class="slider"> <br>

    <input class="checkbox" type="checkbox" id="php">
    <label style="display:inline-block" for="php">PHP</label>
    <input class="checkbox" type="checkbox" id="python">
    <label style="display:inline-block" for="python">Python</label>
    <input class="checkbox" type="checkbox" id="java">
    <label style="display:inline-block" for="java">Java</label> <br>

    <input type="submit" class="button-large">
  </div>
</body>
</html>
```


وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



test.html

Full Name:

Male Female

Developer Designer

Ring Level

PHP Python Java

شريط تمرير الصفحة Scroll Bar

والآن سوف نقوم بعمل ملف CSS بالاسم `customscroll.css` ثم نكتب به الكود التالي ونحفظه:

```
::-webkit-scrollbar {
    width: 10px;
}

::-webkit-scrollbar-track {
    -webkit-box-shadow: inset 0 0 3px rgba(0,0,0,0.3);
}

::-webkit-scrollbar-thumb {
    background-color:#8880e3;
    -webkit-box-shadow: inset 0 0 3px rgba(0,0,0,0.5);
}

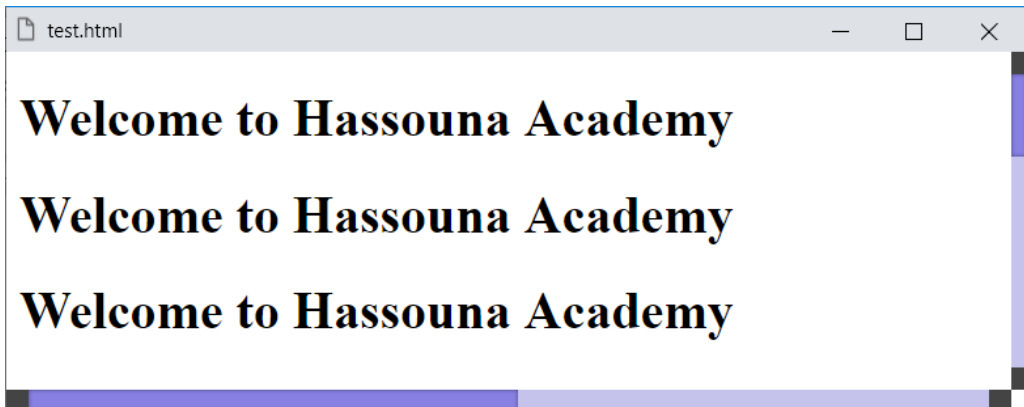
::-webkit-scrollbar-button {
    background: #444;
}

::-webkit-scrollbar-track-piece{
    background-color: #c5c2eb;
}
```

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="customscroll.css">
<style>
  h1{ width:200%; }
</style>
</head>
<body>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



ومع الشكل السابق نلاحظ تغير ال Scroll في الصفحة.

والآن سوف نقوم بعمل ملف CSS بالاسم `customscroll.css`
ثم نكتب به الكود! التالي ونحفظه:

```
::-webkit-scrollbar {
    width: 20px;
}

::-webkit-scrollbar-track {
    -webkit-box-shadow: inset 0 0 5px rgba(0,0,0,0.3);
}

::-webkit-scrollbar-thumb {
    background:linear-gradient(orange,red,orange);
    -webkit-box-shadow: inset 0 0 3px rgba(0,0,0,0.5);
    border-radius:10px;
}

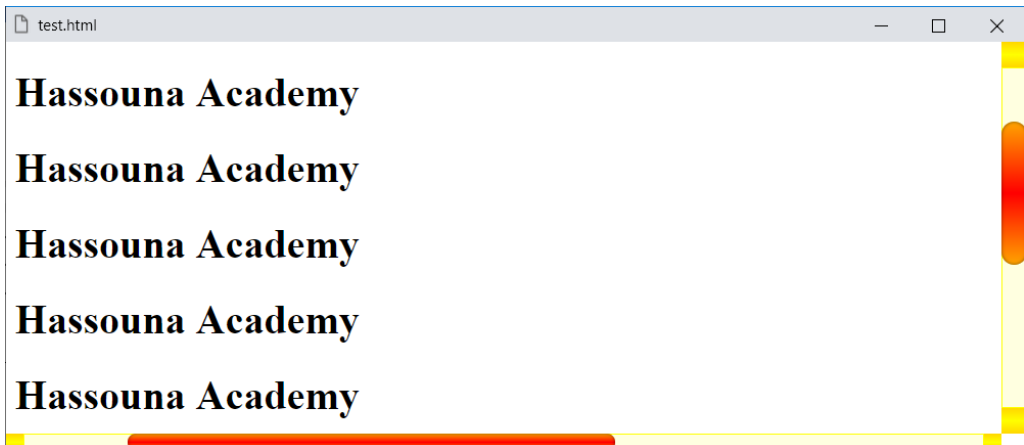
::-webkit-scrollbar-button {
    background:linear-gradient(gold,yellow,gold);
    height:20px;
}

::-webkit-scrollbar-track-piece{
    background-color: lightyellow;
    border:0.5px solid yellow;
}
```

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="customscroll12.css">
<style>
  h1{ width:200%; }
</style>
</head>
<body>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
</body>
</html>
```

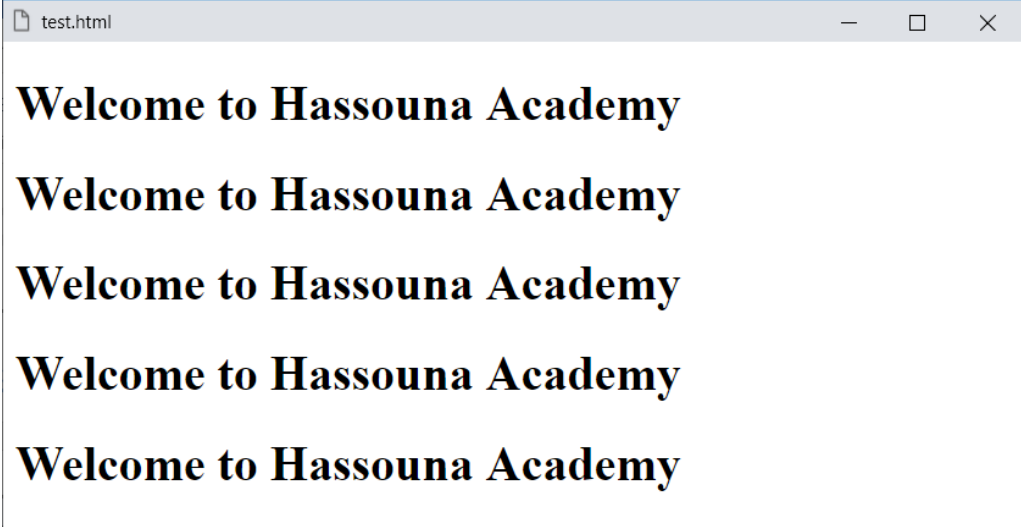
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



ويمكن إخفاء شريط التمرير للصفحة مع الاحتفاظ بالتنقل بشكل طبيعي،
سواء عن طريق دائرة الماوس mouse wheel، أو عن طريق
أسهم الكيبورد، وسنجد بالمثال التالي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body::-webkit-scrollbar {
    display:none;
  }
  h1{ width:200%; }
</style>
</head>
<body>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
<h1>Welcome to Hassouna Academy</h1>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



ولكن إذا قمت بالتحريك بزر عجلة الماوس أو اسهم الكيبورد، سوف تتجاوب معك بشكل طبيعي.

استعلامات الميديا Media Query

هذا الموضوع يعد من أهم موضوعات اللغة CSS، لأنه يضبط تنسيقان صفحة الويب من جميع الجهات، ويتم ضبطها أثناء العرض العادي، مثل أن نفتح صفحة الويب بشكل طبيعي، ويتم ضبطها من ناحية العرض أثناء الطباعة، مثل أن نقوم بعمل طباعة من المتصفح أو نضغط Ctrl+P، ويتم ضبطها من ناحية العرض على ال TV، ويتم ضبطها عند العمل على أي حجم، ويتم ضبطها بطريقة كثيرة، ثم بعد فهم تلك الأمور يكون موقعك Responsive وهذه هي كلمة السر في موضوع استعلامات ال Media Query، وال Responsive أي هو قابل للعرض على جميع الأحوال، سواء شاشات، أو لابتوب، أو TV، أو Mobile، ليكون موقعك بمثابة عرض جمالي على كل الشاشات مع اختلاف أحجامها.

البداية مع ال Media Query

يملك ضبط الميديا مع خلال النداء على ملف CSS به تنسيقات محددة.
والآن سوف نصنع ملف باسم [myprint.css](#) ليتم فيه وضع تنسيقات معينة تظهر فقط أثناء الطباعة.

ضع الكود! التالي في ملف [myprint.css](#) ثم احفظه:

```
div{
  font-size:30px;
  color:blue;
  border:3px dashed blue;
}
```

والآن سوف نصنع ملف باسم [myscreen.css](#) ليتم فيه وضع تنسيقات معينة تظهر فقط أثناء الطباعة.

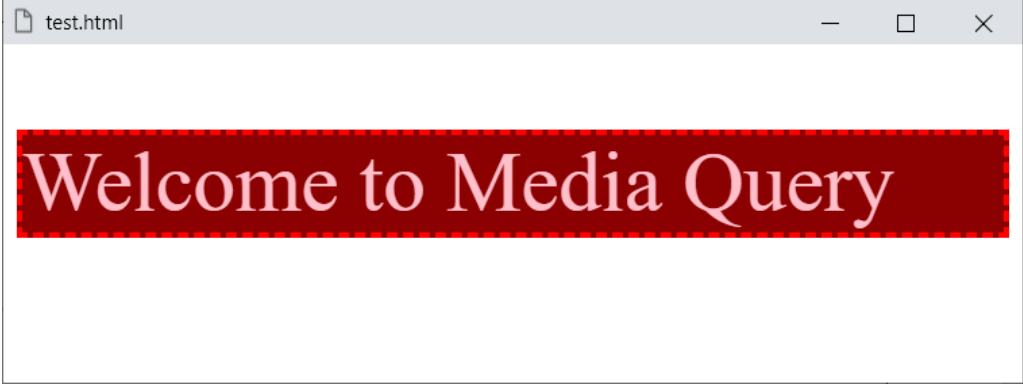
ضع الكود! التالي في ملف [myscreen.css](#) ثم احفظه:

```
div{
  font-size:50px;
  background-color:darkred;
  color:pink;
  border:3px dashed red;
}
```

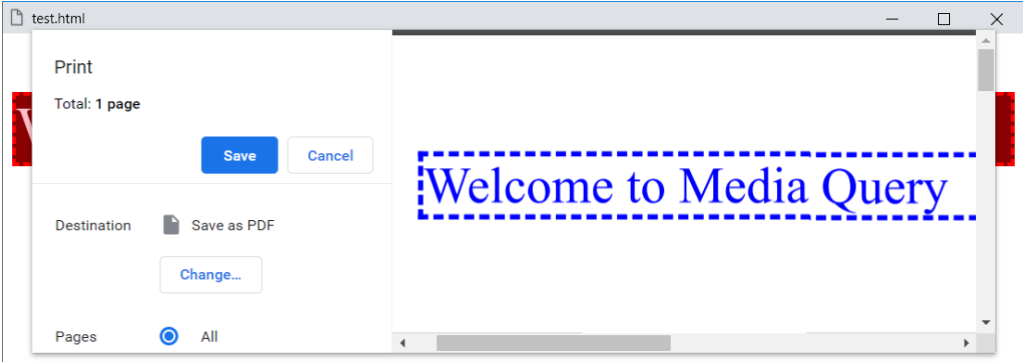
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="myscreen.css" media="screen">
  <link rel="stylesheet" href="myprint.css" media="print">
</head>
<body>
  <p>Welcome to Media Query</p>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



والآن قم بالضغط على Ctrl+P لتظهر لك شاشة الطباعة مختلفة تماماً عن الشاشة العادية كما بالشكل التالي:



ويتبين لنا من المثال السابق ما يلي:

- عند ضبط الخاصية media على القيمة screen للوسم link تم اعتماد التنسيق الى شاشة العرض فقط
- عند ضبط الخاصية media على القيمة print للوسم link تم اعتماد التنسيق الى وضع الطباعة فقط

انتبه! فإنه يمكن كتابة الـ media بطريقة أخرى كما يلي.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>

  @media screen{
    p{
      font-size:50px;
      background-color:darkred;
      color:pink;
      border:3px dashed red;
    }
  }

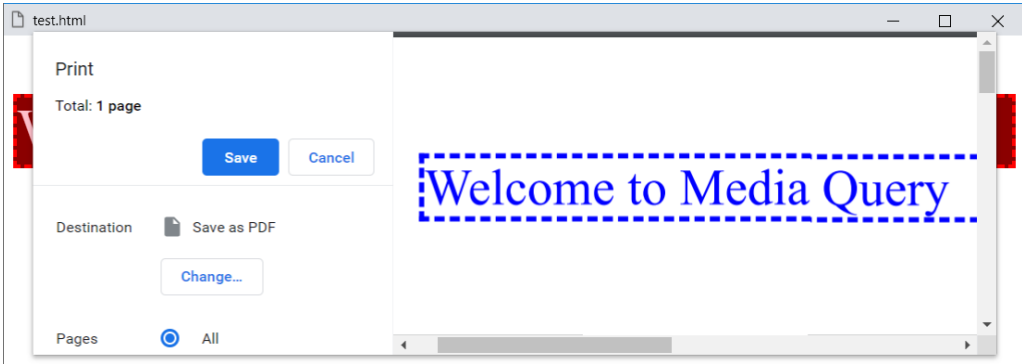
  @media print{
    p{
      font-size:30px;
      color:blue;
      border:3px dashed blue;
    }
  }

</style>
</head>
<body>
  <p>Welcome to Media Query</p>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



والآن قم بالضغط على Ctrl+P لتظهر لك شاشة الطباعة مختلفة تماماً عن الشاشة العادية كما بالشكل التالي:



والآن لاحظ أنه تم كتابة الـ media بطريقة مختلفة، عن طريق كتابة المحدد @media ثم مسافة ثم الكلمة screen للشاشة، أو الكلمة print للطباعة، ثم بين الأقواس { } تكتب ما تريد من تنسيقات CSS.

تنوع الاستخدامات مع ال Media Query

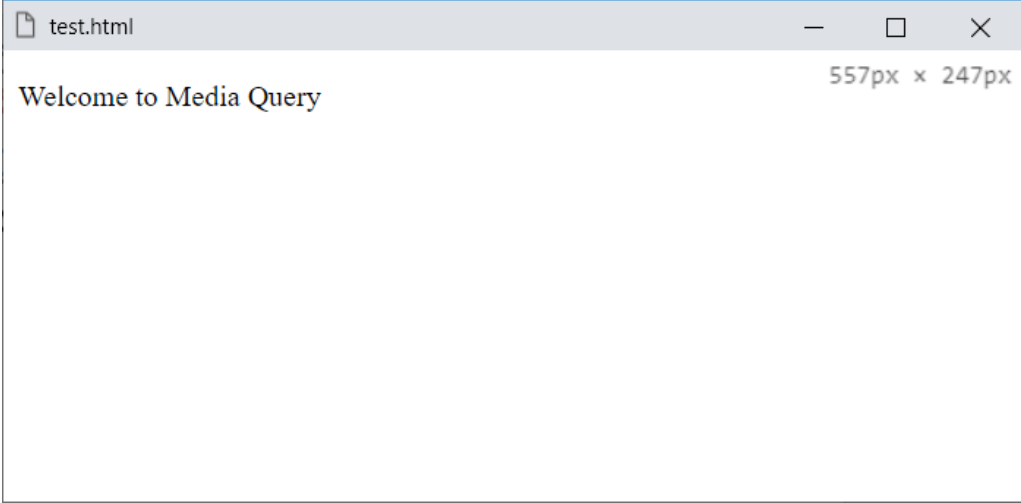
يمكنك أن تشرط على تنسيقك بأكثر من شرط، مثلاً أن يتم عرضة على الشاشة فقط، ولا ينفذ لعرض أكبر من 500 بكسل، كما يلي.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>

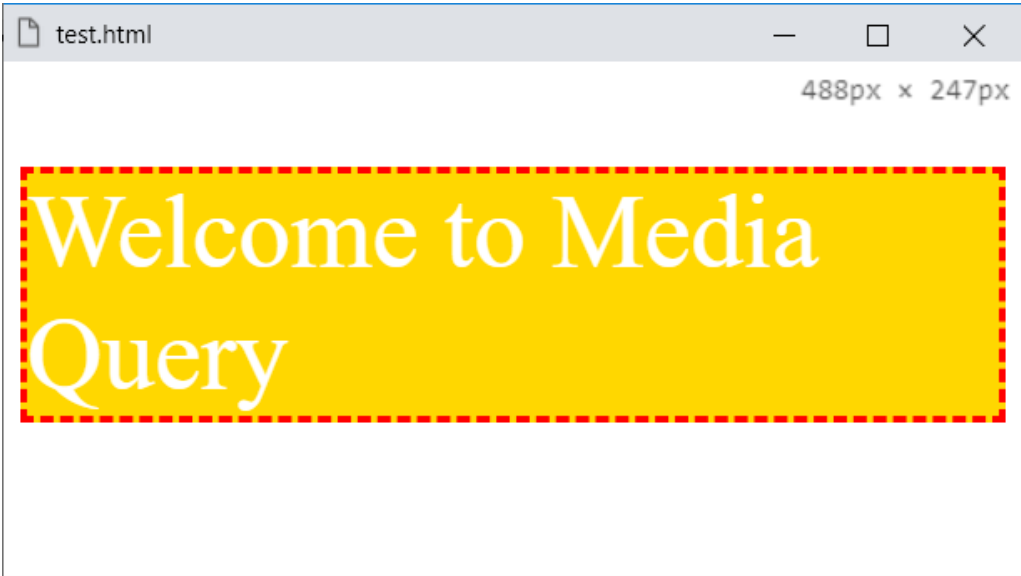
      @media only screen and (max-width:500px){
        p{
          font-size:50px;
          background-color:gold;
          color:#fff;
          border:3px dashed red;
        }
      }

    </style>
  </head>
  <body>
    <p>Welcome to Media Query</p>
  </body>
</html>
```

وبعد تشغيل الصفحة على عرض كبير تظهر كما بالشكل التالي:



لاحظ أن العرض السابق أقل من الـ 500px، ولذلك فلم يتم عليه التنسيق، وبعد تقليل العرض يطبق التنسيق، ولاحظ كما بالشكل التالي:



ولو تم عمل طباعة له يظهر أي تنسيق في الطباعة.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
  <style>

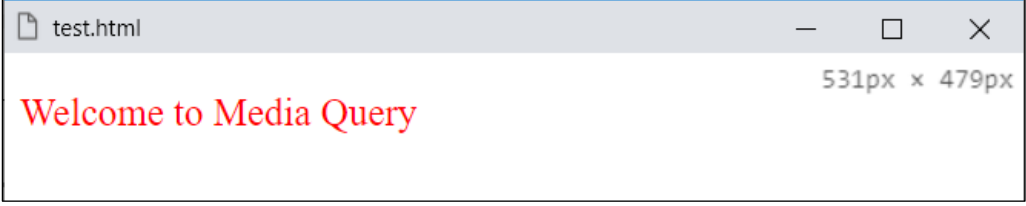
    #myimage{ display:none; }

    p{ color:red; font-size:20px; }

    @media only screen and (min-width:600px){
      p{
        font-size:50px;
        background-color:gold;
        color:#fff;
        border:3px dashed red;
      }
      #myimage{
        display:block;
      }
    }

  </style>
</head>
<body>
  <p>Welcome to Media Query</p>
  
</body>
</html>
```

وبعد تشغيل الصفحة على عرض صغير تظهر كما بالشكل التالي:



لاحظ أن العرض السابق أقل من الـ 600px، ولذلك فلم يتم عليه التنسيق الا التنسيق الأصلي له، وبعد زيادة العرض يطبق التنسيق، ولاحظ كما بالشكل التالي:



ولو تم عمل طباعة له يظهر أي تنسيق في الطباعة.

استخدام مع طبيعة الصفحة

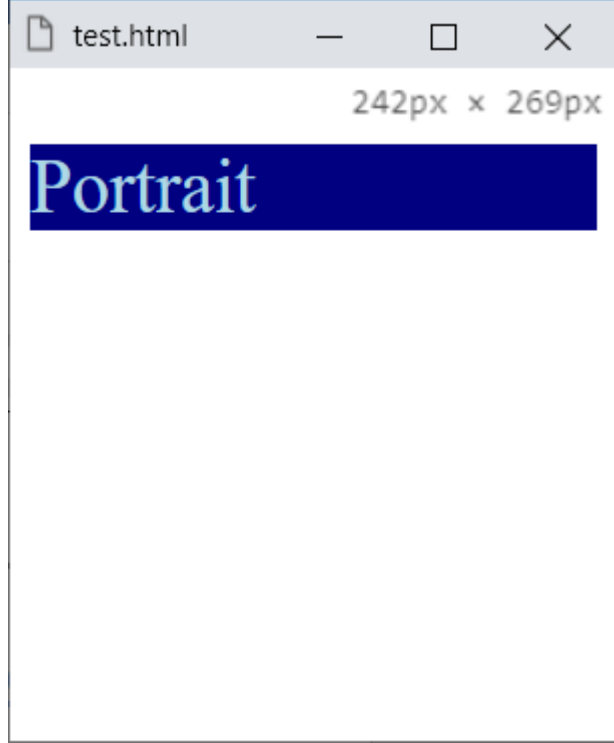
طبيعة الصفحة orientation، إما أن تكون landscape، وإما أن تكون portrait، وفيما يلي مثال يوضح ذلك.
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  p{
    display:none; font-size:30px;
    background:navy; color:lightblue;
  }

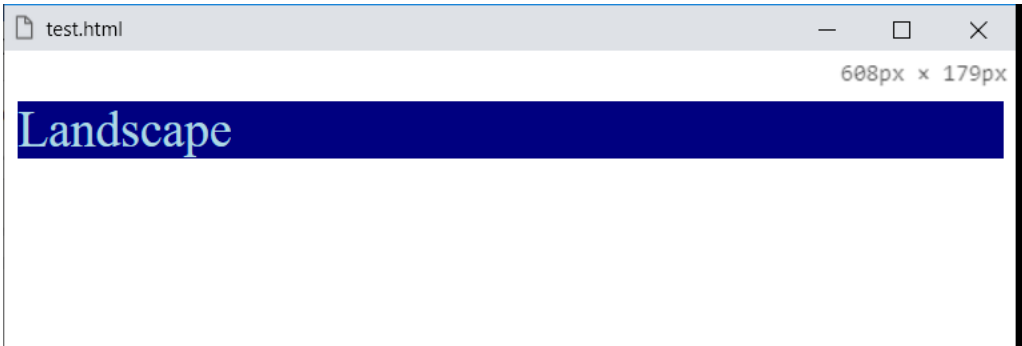
  @media only screen and (orientation:landscape){
    #p1{
      display:block;
    }
  }

  @media only screen and (orientation:portrait){
    #p2{
      display:block;
    }
  }
</style>
</head>
<body>
  <p id="p1">Landscape</p>
  <p id="p2">Portrait</p>
</body>
</html>
```

وبءء ءشءءل الصءفة على عرض وطول ٱتناسب مع ال **portrait** ءظهر
كما بالشءكل ءءالء:



وءءل الصءفة على عرض وطول ٱتناسب مع ال **landscape** ءظهر
كما بالشءكل ءءالء:



تنسيق يبه أقصى وأقل للعرض Min and Max كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  #p1{ font-size:30px; }

  @media (max-width:200px){
    #p1{ color:#aaa; }
  }

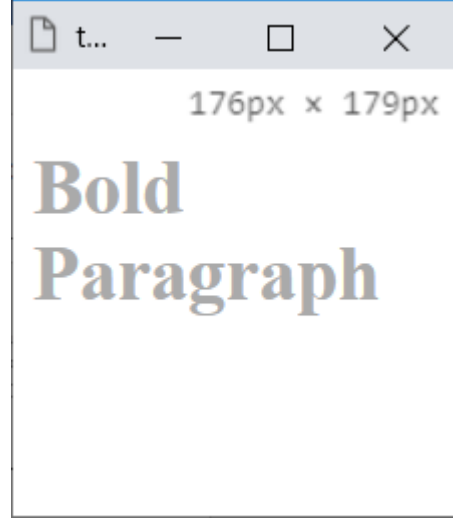
  @media (min-width:200px) and (max-width:400px){
    #p1{ color:#f00; }
  }

  @media (min-width:401px) and (max-width:600px){
    #p1{ color:#0f0; }
  }

  @media (min-width:601px) and (max-width:800px){
    #p1{ color:#00f; }
  }

  @media (min-width:801px){
    #p1{ color:#0ae; }
  }
</style>
</head>
<body>
  <p id="p1"><b>Bold Paragraph</b></p>
</body>
</html>
```

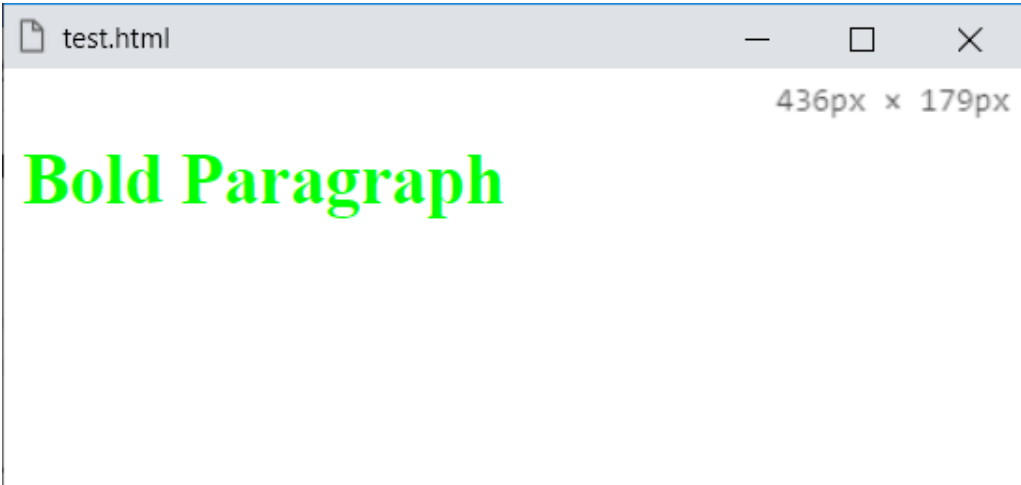
وبءء ءءءءل الصءفة بعرض لأقل مء 200px ءظهء ءما بالشءك ءءاءل:



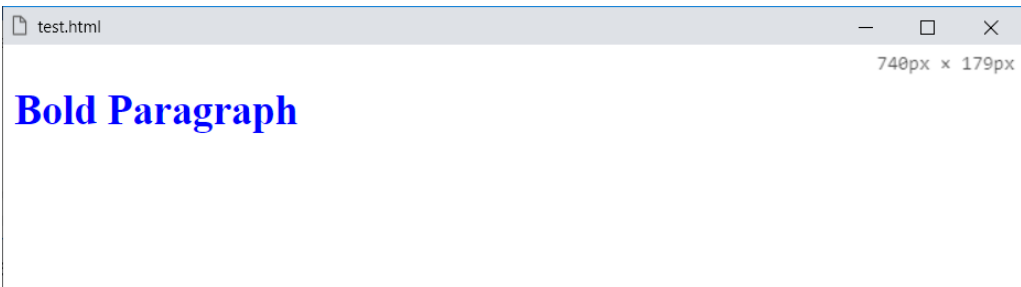
والآن سوف نجعل العرض ما بءء 200 و400 بءكسل لءكون ءما بءل:



والآن سوف نجعل العرض ما بينه 401 و600 بيكسل ليكون كما يلي:



والآن سوف نجعل العرض ما بينه 601 و800 بيكسل ليكون كما يلي:



والآن سوف نجعل العرض أكبر من 801 بيكسل ليكون كما يلي:



رجاء **انتبه!** لأن المثال السابق يلخص قصة ال Responsive بشكل كبير، لأنك تستطيع الآه أن تفعل أي تنسيقات أي كانت، سواء تخفي عناصر، أو تظهر عناصر، أو تحرك عناصر، أو تغير تنسيقات عناصر، وذلك لك حجم تريده بفضل الله.

والآن سوف نركز قليلاً مع الأكواد في المثال السابق، فإن الكود التالي:

```
@media (max-width:200px)
```

معناه أن التنسيقات يتم تطبيقها مع 200px للعرض كحد أقصى، أي ما أكبر منه ال 200px له يتم عليه تلك التنسيقات، والكود التالي:

```
@media (min-width:200px) and (max-width:400px)
```

معناه أن التنسيقات يتم تطبيقها مع عرض يتراوح من 200px كحد أدنى، و400px كحد أقصى، والكود التالي:

```
@media (min-width:401px) and (max-width:600px)
```

معناه أن التنسيقات يتم تطبيقها مع عرض يتراوح من 401px كحد أدنى، و600px كحد أقصى، والكود التالي:

```
@media (min-width:601px) and (max-width:800px)
```

معناه أن التنسيقات يتم تطبيقها مع عرض يتراوح من 601px كحد أدنى، و800px كحد أقصى، والكود التالي:

```
@media (min-width:801px)
```

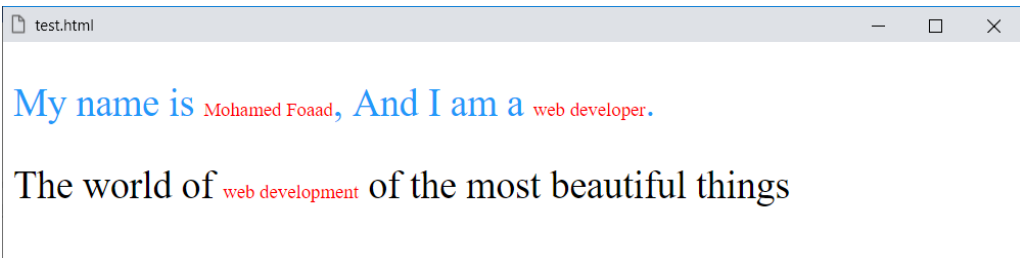
معناه أن التنسيقات يتم تطبيقها مع 801px للعرض كحد أدنى، أي ما أقل منه ال 801px له يتم عليه تلك التنسيقات.

الوراثة Inherit

موضوع الوراثة هو أن يرث عنصر child من عنصر parent.
والآن سوف نقوم بعمل مثال عادي بدون وراثة:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  p{ font-size:30px; }
  #p1{ color:#2199fa; }
  span{ font-size:15px; color:red; }
</style>
</head>
<body>
<p id="p1">
  My name is <span id="s1">Mohamed Foaad</span>,
  And I am a <span id="s2">web developer</span>.
</p>
<p id="p2">
  The world of <span id="s3">web development</span>
  of the most beautiful things
</p>
</body>
</html>
```

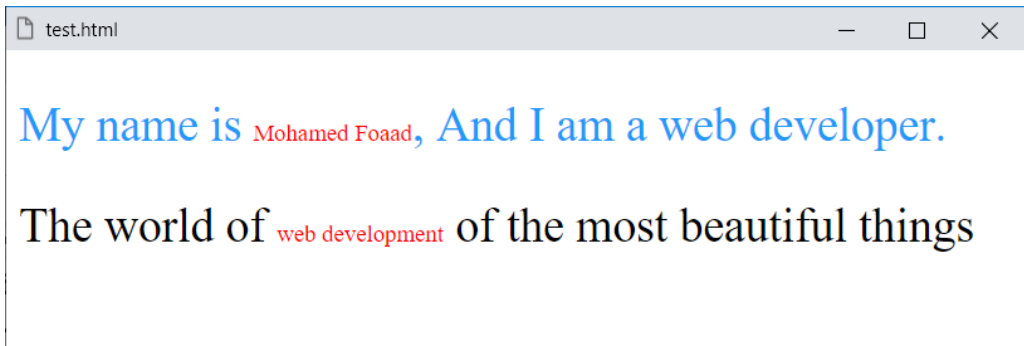
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



والآن سوف نطبق الوراثة علية، لتعرف تفعيلها وأهميتها كما يلي:
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
  p{ font-size:30px; }
  #p1{ color:#2199fa; }
  span{ font-size:15px; color:red; }
  #s2{ color:inherit; font-size:inherit; }
</style>
</head>
<body>
<p id="p1">
  My name is <span id="s1">Mohamed Foaad</span>,
  And I am a <span id="s2">web developer</span>.
</p>
<p id="p2">
  The world of <span id="s3">web development</span>
  of the most beautiful things
</p>
</body>
</html>
```

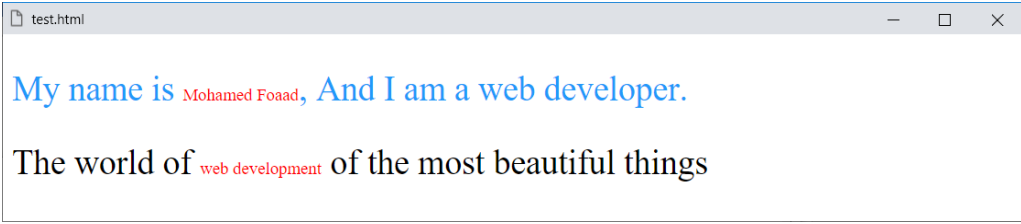
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



والآن لاحظ الاختلاف بين المثالية كما في الشكلية التالية:
هذا بدون وراثة



وهذا بالوراثة



ومع هنا نلاحظ أن الوسم span بداخل عنصر، وموجه له تنسيقاً مختلفة عنه، ولكنه بعد الوراثة، ورث من العنصر parent له بعض الخصائص.

ءرف المءفراء Variables

كلمة مءفء ءءفر الى شءء ىءم بناؤه، ءم ىءم ءءرفه او ءفظ قفمة كما نرفء باءءة، ءم ىءم ءءامل معه فف أف وقء فناءء القفمة ءف ىءفظ بها، ءف ىءم عمل مءفراء كءفءة وءبفء قفم مءءلفة بها، مءل الألوان او ءصوءه أو ءفء ذلك، ففءون مه السهل علفنا ءءفر فف قفمة مءفء واءء لىءم بهذا ءءفر ءفراء كءفءة فف الموقء، مءلاً ألوان الفءراء فف مءفء، والوان الأزراء فف مءفء، وهءذا.

وىءم ءرف المءفراء اءءل ال root: فف ءنسبءاء ال CSS، وءرف مءفء ىءم ءءابة الرمز ءاش مءفء -- ءم اسم المءفء ءم الرمز ءولون : ءم القفمة ءف نوء ان ىءم ءفظها فف ذلك المءفء للءاء علفها ففما بعء فف ءنسبءاء.

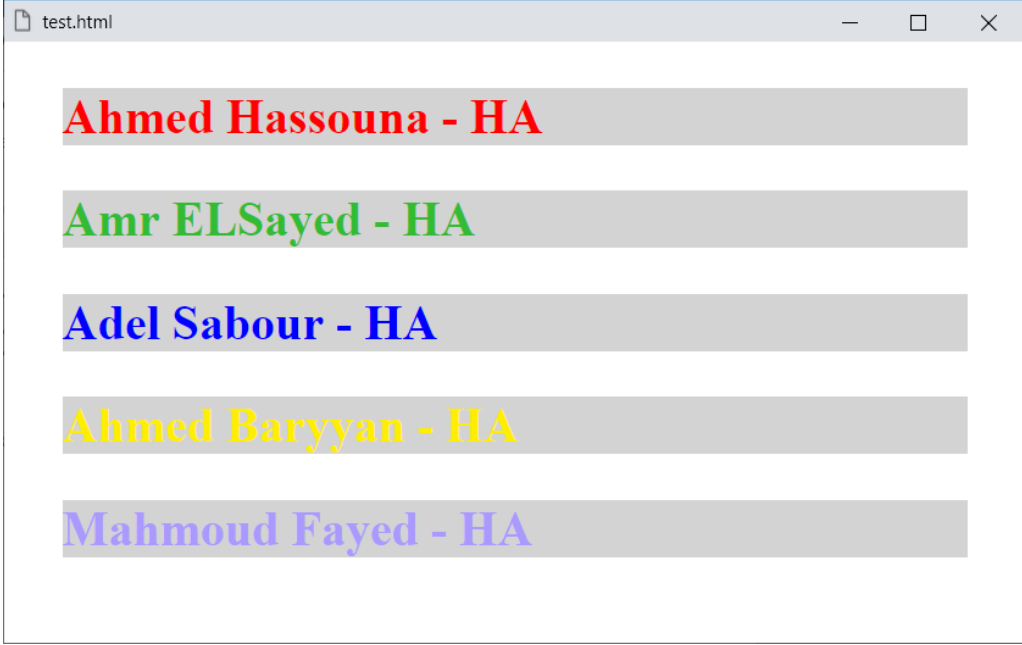
وللءاء على المءفراء للاسءفاءة مه القفم الموءوءة باءءلها، ىءم ءءابة الءالة var وقوسفها () ءم باءء القوسفء ءءب اسم المءفء كما هو. وففما فف سوف ىءم عمل مءال شامل للءامل مع المءفراء ان شاء الله.

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<style>
:root{
  --public-word:"HA";
  --margin-h1:30px;
  --bg-h1:lightgray;
  --name1-color:#f00;
  --name2-color:#3b3;
  --name3-color:#00f;
  --name4-color:#fe0;
  --name5-color:#a9f;
}

#name1{ color: var(--name1-color); }
#name2{ color: var(--name2-color); }
#name3{ color: var(--name3-color); }
#name4{ color: var(--name4-color); }
#name5{ color: var(--name5-color); }
h1{
  margin: var(--margin-h1);
  background: var(--bg-h1);
}
h1:after{
  content:" - " var(--public-word);
}
</style>
</head>
<body>
<h1 id="name1">Ahmed Hassouna</h1>
<h1 id="name2">Amr ELSayed</h1>
<h1 id="name3">Ade1 Sabour</h1>
<h1 id="name4">Ahmed Baryyan</h1>
<h1 id="name5">Mahmoud Fayed</h1>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



ونلاحظ في المثال السابق أنه تم تعريف المتغير `public-word` وبه القيمة "HA" وتم تعريف المتغير `margin-h1` وبه القيمة `30px` وتم تعريف المتغير `bg-h1` وبه القيمة `lightgray` وتم تعريف المتغير `name1-color` وبه القيمة `#f00` وتم تعريف المتغير `name2-color` وبه القيمة `#3b3` وتم تعريف المتغير `name3-color` وبه القيمة `#00f` ثم بعده مباشرة تم تعريف المتغير `name4-color` وبه القيمة `#fe0` وتم تعريف المتغير `name5-color` وبه القيمة `#a9f`، ونلاحظ أن كل هذه المتغيرات تم وضعها بقيمتها داخل الـ `root`: في تنسيقات الـ `css`، ثم تم استخدام كل هذه المتغيرات في أماكن مختلفة فيما بعد.

تمارين عملية practical examples

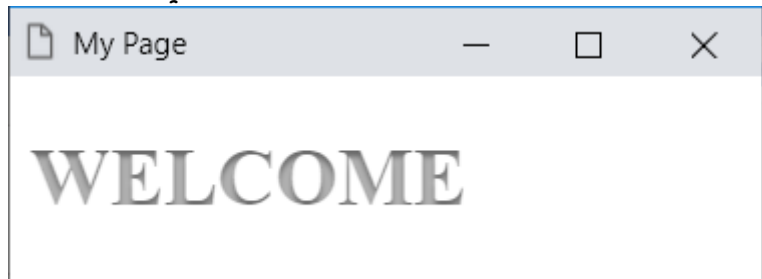
وبفضل الله وحده، بعد المذاكرة الشيقة لهذا الكتاب، سوف يتم وضع أمثلة
تدريب لما تعلمناه بفضل الله، ولكم **انتبه!** فلا يتم التركيز على الشكل
الجمالي في هذه الأمثلة، ولكم يتم التركيز على بعض الجوانب العملية.

تمريه نص مخفور

كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
<head>
<title>My Page</title>
<style>
h1{
background-color: #666; color: transparent;
text-shadow: 0px 2px 3px rgba(255,255,255,0.5);
-webkit-background-clip: text;
-moz-background-clip: text; background-clip: text;
}
</style>
</head>
<body> <h1>WELCOME</h1> </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



ءءفر شكل الروابط ءوء! قم بءءابة الكوء الءالب ثم اءفظه:

```
<html>
  <head>
    <title>My Page</title>
    <style>
      body{ padding:20px; }
      a:link , a:visited{
        text-decoration: none;
        background-color: navy;
        color:lightblue;
        padding:5px 20px;
        border:2px solid #3dbce5;
        border-radius: 15px;
      }

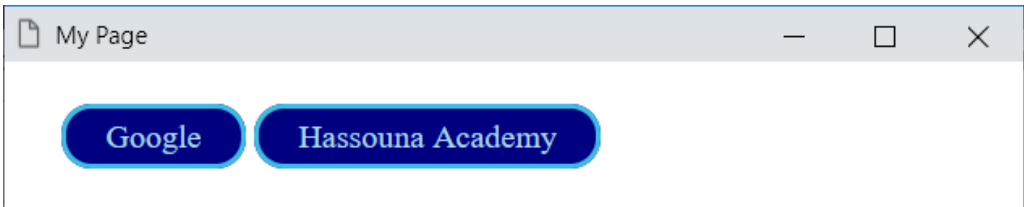
      a:hover , a:active{
        background-color:#4565ae;
      }
    </style>
  </head>

  <body>

  <a href="http://www.google.com">Google</a>
  <a href="https://www.hassouna-academy.com">Hassouna Academy</a>

  </body>
</html>
```

وبءء ءشءفل الصءفة ءظهر كما بالشكل الءالب:

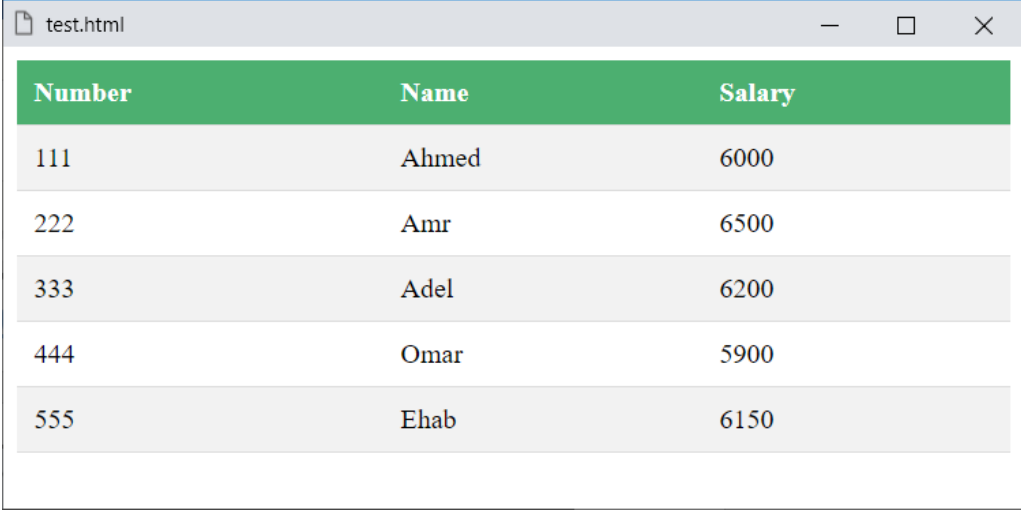


عمل جدول

كود! قم بكتابة الكود التالي ثم احفظه:

```
<html>
<head>
  <style>
    table{ border-collapse: collapse; width:100%; }
    th{
      height:35px;
      text-align:left;
      vertical-align: top;
      background-color:#4CAF70;
      color:#fff;
    }
    th, td{
      padding: 10px;
      border-bottom:1px solid #ddd;
    }
    tr:nth-child(even) {
      background-color: #f2f2f2;
    }
    tr:hover {background-color: #c2cbbd}
  </style>
</head>
<body>
  <table>
    <tr><th>Number</th><th>Name</th><th>Salary</th></tr>
    <tr><td>111</td><td>Ahmed</td><td>6000</td></tr>
    <tr><td>222</td><td>Amr</td><td>6500</td></tr>
    <tr><td>333</td><td>Adel</td><td>6200</td></tr>
    <tr><td>444</td><td>Omar</td><td>5900</td></tr>
    <tr><td>555</td><td>Ehab</td><td>6150</td></tr>
  </table>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:

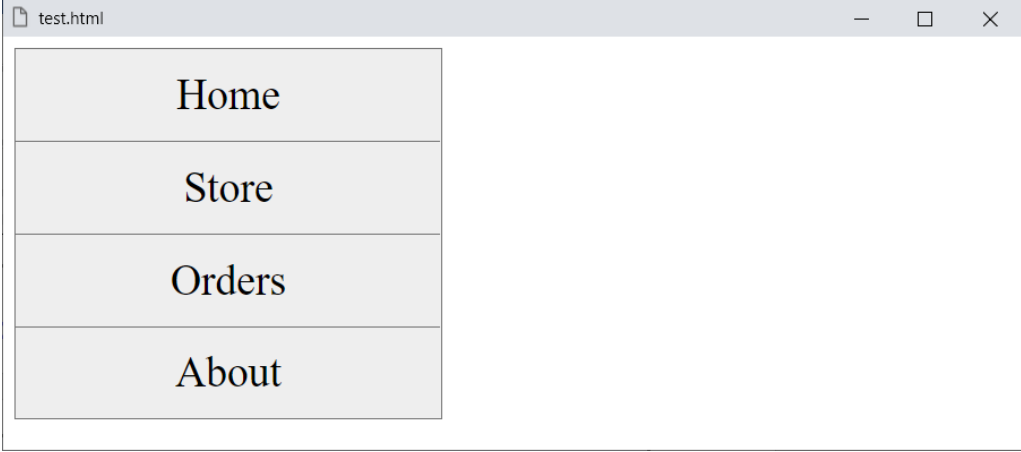


Number	Name	Salary
111	Ahmed	6000
222	Amr	6500
333	Adel	6200
444	Omar	5900
555	Ehab	6150

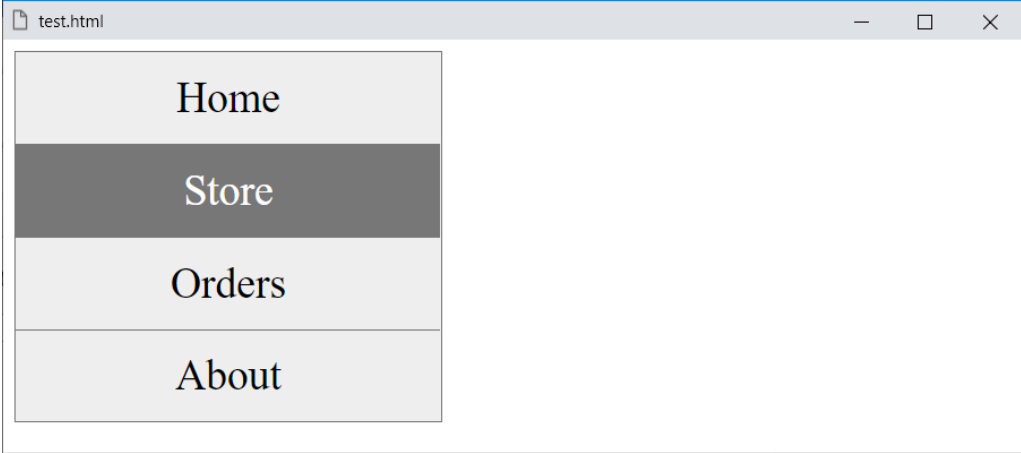
قائمة رأسية عادية
كود! قم بكتابة الكود التالي ثم احفظه:

```
<html>
<head>
  <style>
    *{ font-size: 30px; }
    ul{
      list-style-type:none;
      margin:0;
      padding:0;
      background: #eee;
      width:300px;
      border:1px solid #777;
    }
    li a{
      text-decoration: none;
      padding:15px;
      color:#000;
      display: block;
    }
    li{
      border-bottom:1px solid #777;
      text-align: center;
    }
    li:last-child{ border-bottom: none; }
    li a:hover{ background: #777; color:#fff; }
  </style>
</head>
<body>
  <ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">Store</a></li>
    <li><a href="#">Orders</a></li>
    <li><a href="#">About</a></li>
  </ul>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



ومع اللمس تكون مثل الشكل التالي:



قائمة جانبية والمحتوي على اليمين كود! قم بكتابة الكود التالي ثم احفظه:

```
<html>
<head>
  <style>
    body{ margin:0; }
    *{ font-size: 30px; }
    ul{
      list-style-type:none;
      margin:0; padding:0;
      background: #eee;
      width:30%; height:100%;
      border:1px solid #777;
      position: fixed; top:0;
      overflow:auto;
    }
    li a{
      text-decoration: none;
      padding:15px; color:#000;
      display: block;
    }
    li{
      border-bottom:1px solid #777;
      text-align: center;
    }
    li a:hover{
      background: #777;
      color:#fff;
    }
    #mypage{
      margin-left:30%;
      padding-left:15px;
    }
  </style>
</head>
```

```
<body>
  <ul>
    <li><a href="#home">Home</a></li>
    <li><a href="#store">Store</a></li>
    <li><a href="#orders">Orders</a></li>
    <li><a href="#about">About</a></li>
    <li><a href="#home">Home</a></li>
    <li><a href="#store">Store</a></li>
    <li><a href="#orders">Orders</a></li>
    <li><a href="#about">About</a></li>
    <li><a href="#home">Home</a></li>
    <li><a href="#store">Store</a></li>
    <li><a href="#orders">Orders</a></li>
    <li><a href="#about">About</a></li>
  </ul>
  <div id="mypage">
    <hr>
    <h1 id="start">Start</h1>
    <p>Welcome To My Website</p>

    <hr>
    <h1 id="home">Home</h1>
    <a href="#start">Back</a>
    <p>This is Home Part</p>

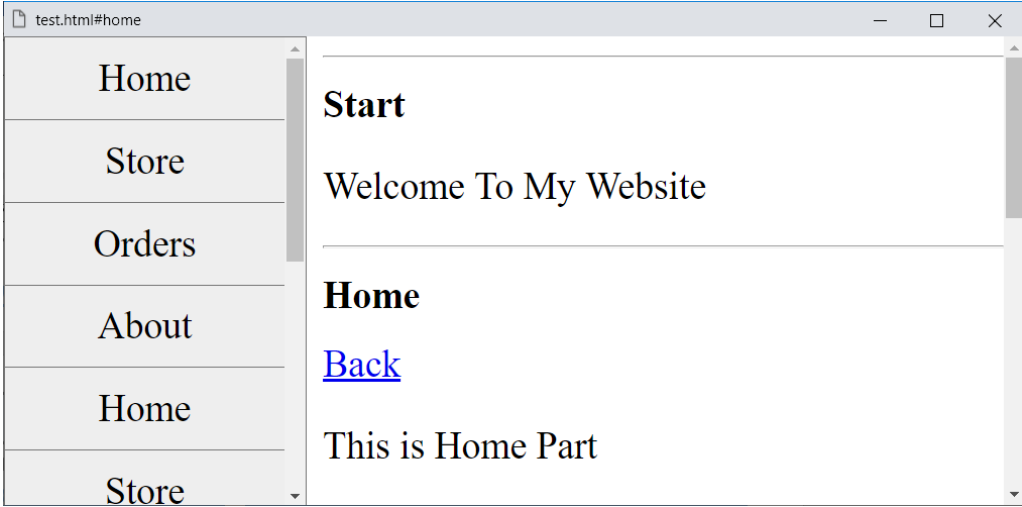
    <hr>

    <h1 id="store">Store</h1>
    <a href="#start">Back</a>
    <p>This is Store Part</p>

    <hr>
    <h1 id="orders">Orders</h1>
    <a href="#start">Back</a>
    <p>This is Orders Part</p>

    <hr>
    <h1 id="about">About</h1>
    <a href="#start">Back</a>
    <p>This is About Part</p>
  </div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



قائمة في الأسفل

كود! قم بكتابة الكود التالي ثم احفظه:

```
<html>
  <head>
    <style>
      *{font-size: 30px;}
      ul{
        list-style-type:none;
        margin:0;
        padding:0;
        background: #eee;
        width:100%;
        border:1px solid #777;
        display: inline-block;
        position: fixed;
        bottom:0;
        left:0;
      }

      body{
        padding-top:55px;
      }

      li a{
        text-decoration: none;
        padding:15px;
        color:#000;
        display: block;
      }

      li{
        /* border-right:1px solid #777; */
        text-align: center;
        float:left;
      }

      li:last-child{
        border-right: none;
        /* border-left:1px solid #777; */
        float: right;
      }
    </style>
  </head>
  <ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">About</a></li>
    <li><a href="#">Contact</a></li>
  </ul>
</html>
```

```
        li a:hover{
            background: #777;
            color:#fff;
        }

    </style>
</head>
<body>
<ul>
    <li><a href="#home">Home</a></li>
    <li><a href="#store">Store</a></li>
    <li><a href="#orders">Orders</a></li>
    <li><a href="#about">About</a></li>
</ul>

<hr>
<h1 id="start">Start</h1>
<p>Welcome To My Website</p>

<hr>
<h1 id="home">Home</h1>
<a href="#start">Back</a>
<p>This is Home Part</p>

<hr>

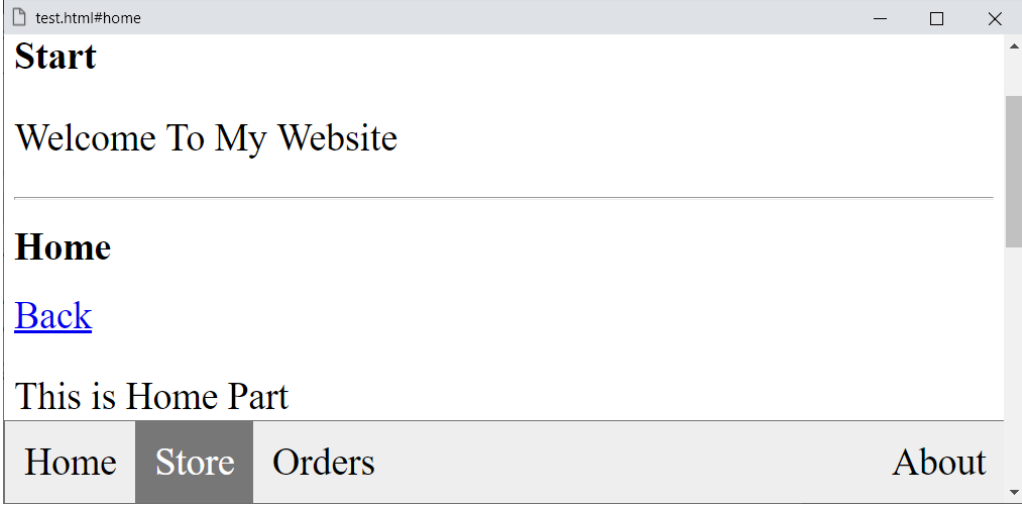
<h1 id="store">Store</h1>
<a href="#start">Back</a>
<p>This is Store Part</p>

<hr>
<h1 id="orders">Orders</h1>
<a href="#start">Back</a>
<p>This is Orders Part</p>

<hr>
<h1 id="about">About</h1>
<a href="#start">Back</a>
<p>This is About Part</p>

</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



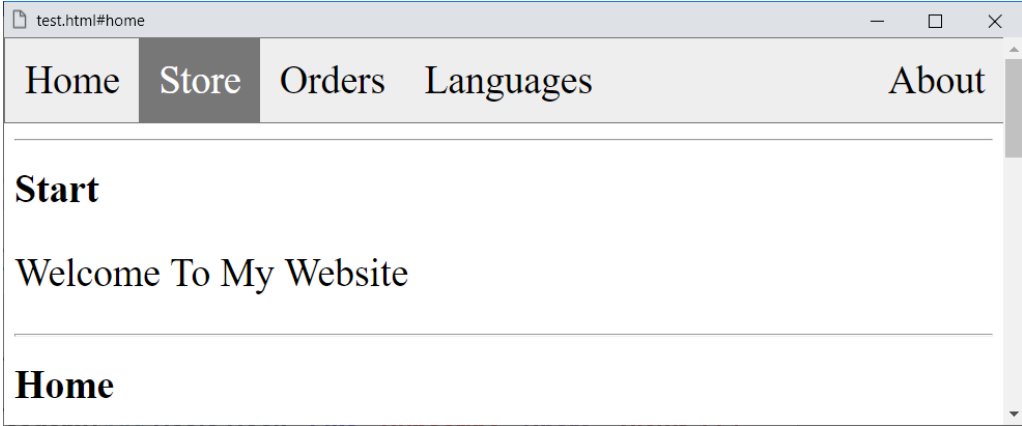
قائمة علوية منبقة

كود! قم بكتابة الكود التالي ثم احفظه:

```
<html>
  <head>
    <style>
      *{ font-size: 30px; }
      ul{
        list-style-type:none;
        margin:0;
        padding:0;
        background: #eee;
        width:100%;
        border:1px solid #777;
        display: inline-block;
        position: fixed;
        top:0;
        left:0;
      }
      body{
        padding-top:55px;
      }
      li a{
        text-decoration: none;
        padding:15px;
        color:#000;
        display: block;
      }
      li{
        /* border-right:1px solid #777; */
        text-align: center;
        float:left;
      }
      li:last-child{
        border-right: none;
        /* border-left:1px solid #777; */
        float: right;
      }
      li a:hover{ background: #777; color:#fff; }
      #lang{
        display: none;
        position: absolute;
```

```
background: #eee;
min-width: 160px;
border:1px solid #777;
}
#drop:hover #lang{ display: block; }
</style>
</head>
<body>
<ul>
  <li><a href="#home">Home</a></li>
  <li><a href="#store">Store</a></li>
  <li><a href="#orders">Orders</a></li>
  <li id="drop"><a href="#">Languages</a>
    <div id="lang">
      <a href="#">HTML</a>
      <a href="#">CSS</a>
      <a href="#">JS</a>
    </div>
  </li>
  <li><a href="#about">About</a></li>
</ul>
<hr>
<h1 id="start">Start</h1>
<p>Welcome To My Website</p>
<hr>
<h1 id="home">Home</h1>
<a href="#start">Back</a>
<p>This is Home Part</p>
<hr>
<h1 id="store">Store</h1>
<a href="#start">Back</a>
<p>This is Store Part</p>
<hr>
<h1 id="orders">Orders</h1>
<a href="#start">Back</a>
<p>This is Orders Part</p>
<hr>
<h1 id="about">About</h1>
<a href="#start">Back</a>
<p>This is About Part</p>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



ولكن لاحظ عند لمس القائمة Languages تظهر كما بالشكل التالي:

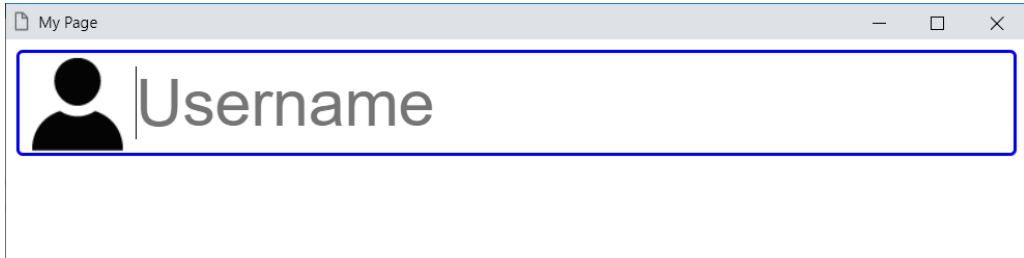


ولو تراكها تختلف مرة أخرى.

مربع نص بأيقونة مع صورة
كود! قم بكتابة الكود التالي ثم احفظه:

```
<html>
  <head>
    <title>My Page</title>
    <style>
      #txtUser{
        width:100%;
        height:auto;
        padding: 10px;
        background: #fff; outline:none;
        border:2px solid blue;
        border-radius:5px;
        font-size:50px;
        background-image:url(user.png);
        background-repeat: no-repeat;
        background-position: 1px;
        padding-left:90px;
      }
    </style>
  </head>
  <body>
    <input id="txtUser" placeholder="Username" />
  </body>
</html>
```

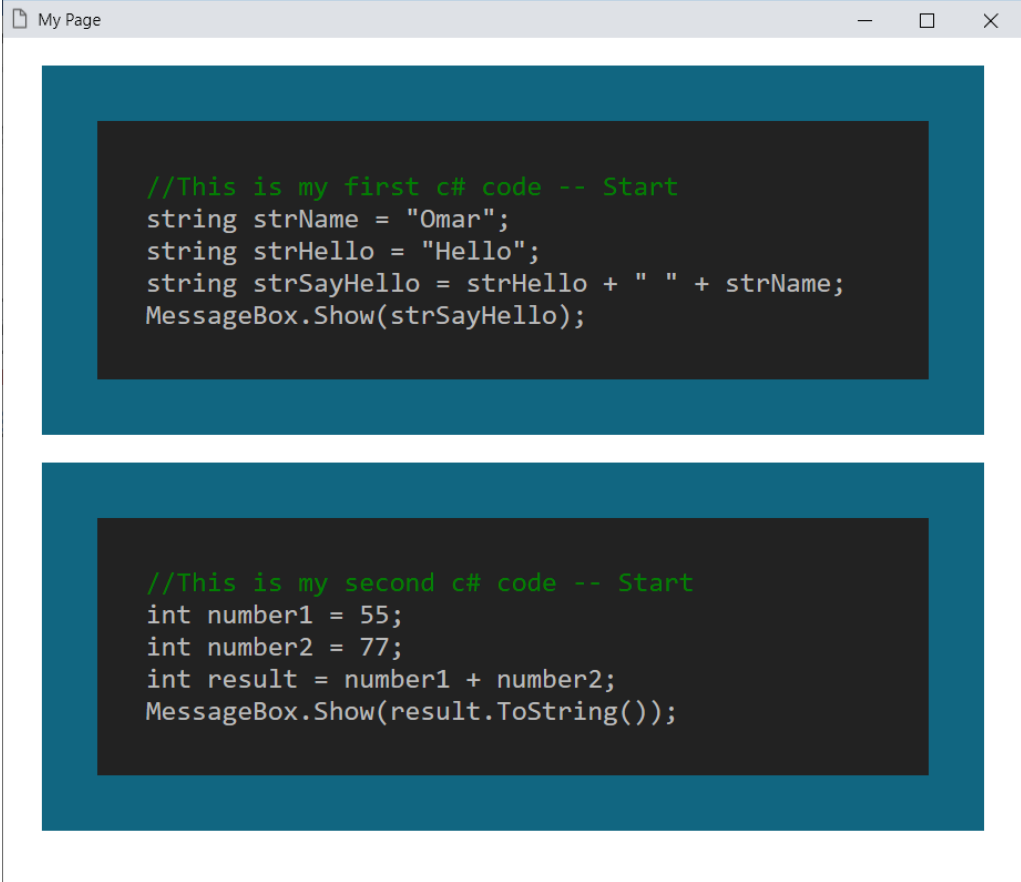
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



تنسيق أكواد برمجية بـ CSS كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      .my{
        font-family: "consolas";
        font-size:20px;
        color:#bbb;
        background:#222;
        padding:35px;
        margin:20px;
        border:40px solid #116681;
      }
      .my:first-line{
        color:green;
      }
    </style>
  </head>
  <body>
    <div class="my">
      //This is my first c# code -- Start<br>
      string strName      = "Omar";<br>
      string strHello     = "Hello";<br>
      string strSayHello = strHello + " " + strName;<br>
      MessageBox.Show(strSayHello);<br>
    </div>
    <div class="my">
      //This is my second c# code -- Start<br>
      int number1 = 55;<br>
      int number2 = 77;<br>
      int result  = number1 + number2;<br>
      MessageBox.Show(result.ToString());<br>
    </div>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



```
//This is my first c# code -- Start
string strName = "Omar";
string strHello = "Hello";
string strSayHello = strHello + " " + strName;
MessageBox.Show(strSayHello);
```

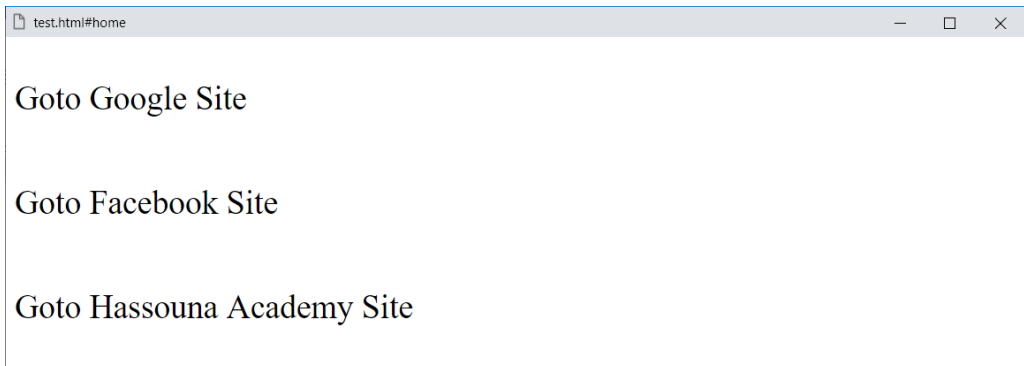
```
//This is my second c# code -- Start
int number1 = 55;
int number2 = 77;
int result = number1 + number2;
MessageBox.Show(result.ToString());
```

الروابط المخفية

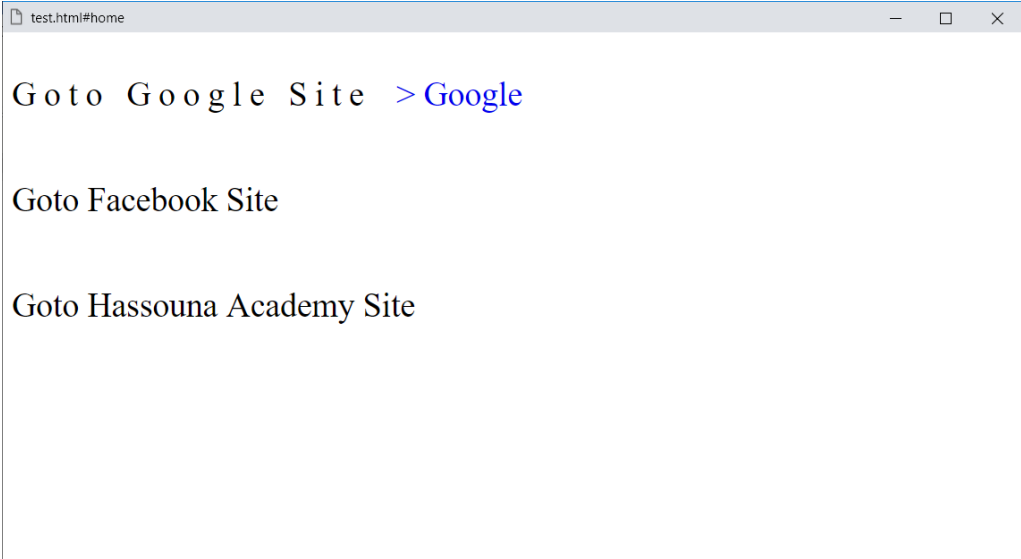
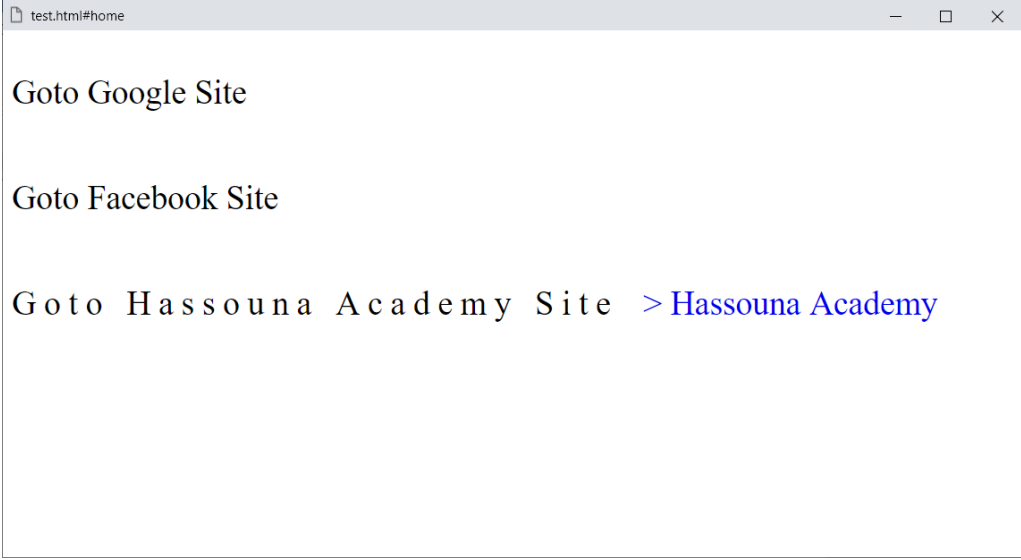
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      a{ opacity:0; }
      p, a{
        display:inline-block;
        text-decoration:none;
        font-size:30px;
        transition:all 1.2s;
        letter-spacing:0;
      }
      p:hover{ letter-spacing:7px; }
      p:hover a{ opacity:1; }
    </style>
  </head>
  <body>
    <p>Goto Google Site
    <a href="https://www.google.com" target="_blank">&nbsp;</a></p><br>
    Google</a></p><br>
    <p>Goto Facebook Site
    <a href="https://www.facebook.com" target="_blank">&nbsp;</a></p><br>
    Facebook</a></p><br>
    <p>Goto Hassouna Academy Site
    <a href="https://www.hassouna-academy.com" target="_blank">&nbsp;</a></p><br>
    Hassouna Academy</a></p><br>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



ولكن عند لمس رابط منهم يظهر كما بالأشكال التالية:



جدول بشریہ تمریر وتلمیحات **کود!** قم بكتابة الكود التالي ثم احفظه:

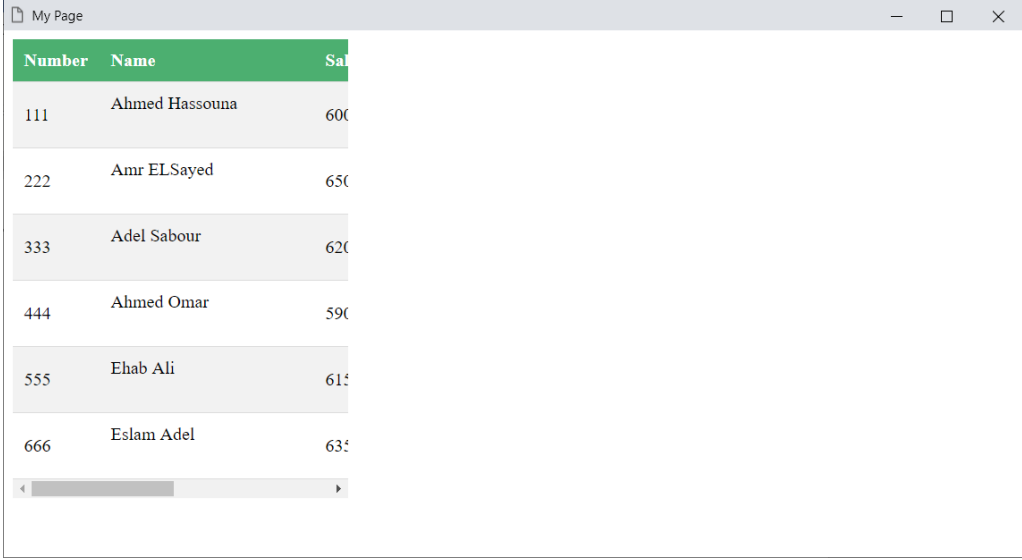
```
<html>
  <head>
    <title>My Page</title>
  <style>
    table{
      border-collapse: collapse;
      width:100%;
    }
    th{
      height:35px;
      text-align:left;
      vertical-align: top;
      background-color:#4CAF70;
      color:#fff;
    }
    th, td{
      padding: 10px;
      border-bottom:1px solid #ddd;
    }
    tr:nth-child(even) {
      background-color: #f2f2f2;
    }
    tr:hover {background-color: #c2cbbd}

    #data{overflow: auto;width:300px}
    .info{
      opacity: 0;
      background: yellow;
      position: relative;
      left: -50px;
      width:170px;
      border:1px solid blue;
      border-radius:10px;
      transition: all 1s;
    }
    tr:hover .info{
      opacity: 1;
    }
  </style>
```

```
</head>
<body>
  <div id="data">
    <table>
      <tr>
        <th>Number</th>
        <th >Name</th>
        <th>Salary</th>
        <th>City</th>
        <th>Country</th>
        <th>Offer</th>
        <th>Department</th>
      </tr>
      <tr>
        <td>111</td>
        <td>Ahmed Hassouna
        <div class="info">This is row1 from table</div>
        </td>
        <td>6000</td>
        <td>Cairo</td>
        <td>Egypt</td>
        <td>6000</td>
        <td>Information Systems</td>
      </tr>
      <tr>
        <td>222</td>
        <td>Amr ELSayed
        <div class="info">This is row2 from table</div>
        </td>
        <td>6500</td>
        <td>Giza</td>
        <td>Egypt</td>
        <td>6000</td>
        <td>Information Systems</td>
      </tr>
      <tr>
        <td>333</td>
        <td>Adel Sabour
        <div class="info">This is row3 from table</div>
        </td>
        <td>6200</td>
        <td>Cairo</td>
        <td>Egypt</td>
```

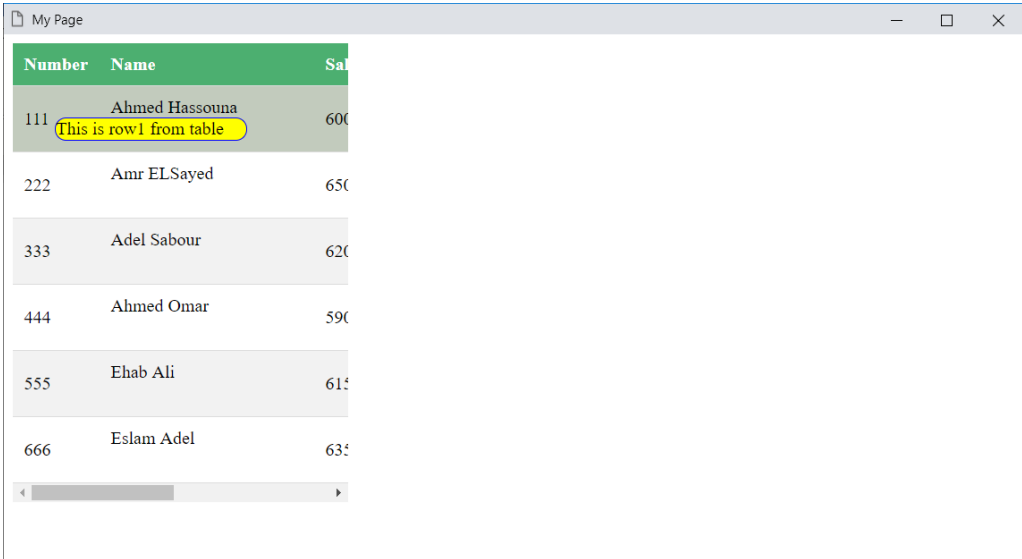
```
<td>6000</td>
<td>Information Systems</td>
</tr>
<tr>
<td>444</td>
<td>Ahmed Omar
<div class="info">This is row4 from table</div>
</td>
<td>5900</td>
<td>Alex</td>
<td>Egypt</td>
<td>6000</td>
<td>Information Systems</td>
</tr>
<tr>
<td>555</td>
<td>Ehab Ali
<div class="info">This is row5 from table</div>
</td>
<td>6150</td>
<td>Cairo</td>
<td>Egypt</td>
<td>6000</td>
<td>Information Systems</td>
</tr>
<tr>
<td>666</td>
<td>Eslam Adel
<div class="info">This is row6 from table</div>
</td>
<td>6350</td>
<td>Giza</td>
<td>Egypt</td>
<td>6000</td>
<td>Information Systems</td>
</tr>
</table>
</div>
</body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



Number	Name	Salary
111	Ahmed Hassouna	600
222	Amr ELSayed	650
333	Adel Sabour	620
444	Ahmed Omar	590
555	Ehab Ali	610
666	Eslam Adel	630

ولكن عند لمس الصف يظهر تلميح خاص به، كما يلي:



Number	Name	Salary
111	Ahmed Hassouna	600
222	Amr ELSayed	650
333	Adel Sabour	620
444	Ahmed Omar	590
555	Ehab Ali	610
666	Eslam Adel	630

مربع يتحرك ويتلوه
كود! قم بكتابة الكود التالي ثم احفظه:

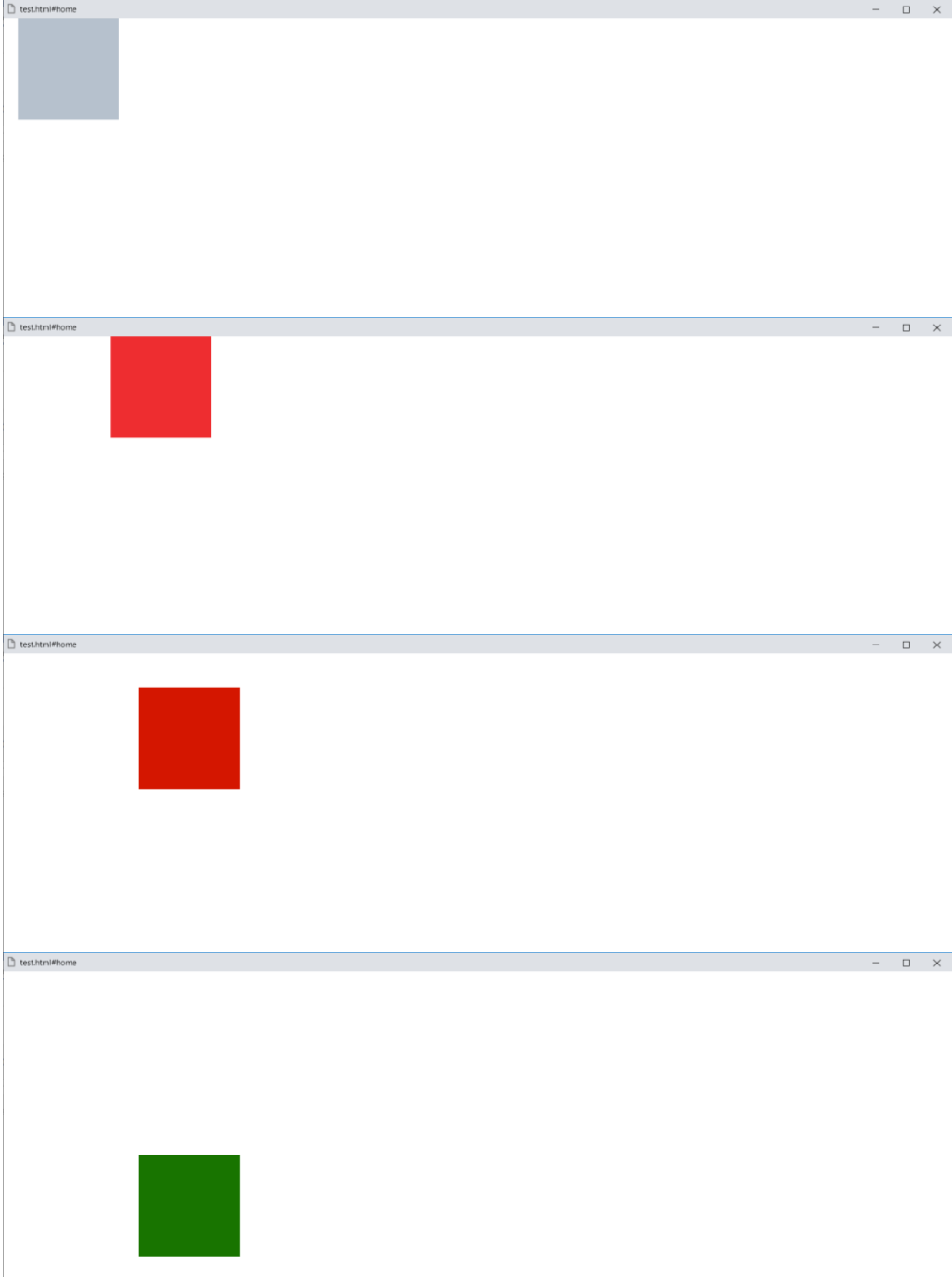
```
<!DOCTYPE html>
<html>
  <head>
    <style>
      #d1{
        background: lightblue;
        width:150px;
        height:150px;
        position:absolute;
        animation-name: my3;
        animation-duration: 3s;
        animation-iteration-count: 3;
      }

      @keyframes my3{
        0%{top:0;left:0;}
        25%{top:0;left:200px;background: red}
        75%{top:300px;left:200px;background: green}
        90%{top:300px;left:0;background: blue}
        100%{top:0;left:0;}
      }

      @keyframes my2{
        0%{ background: lightblue; }
        75%{ background: red; }
        80%{ background: green; }
        90%{ background: blue; }
      }

      @keyframes my1{
        from{ width:150px; }
        to{ width:300px; }
      }
    </style>
  </head>
  <body>
    <div id="d1"></div>
  </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالأشكال التالية:



صفحة المنتجات مع استخدام صور
كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Product</title>
    <style>
      img{ margin:10px; padding:10px; transition:all 1s; }
      .header{
        font-weight:bold; text-align:center; transition:all 1s;
      }
      #nodis{ display:none; }
      @media (max-width:299px){
        #all{ display:none; }
        #nodis{ display:block; }
      }
      @media (min-width: 300px) and (max-width: 450px){
        img{
          width:90%; height:200px;
          background-color:#84eff4;
        }
        .header{ font-size: 25px; color:#fff; }
        body{ background-color: #072845; }
      }

      @media (min-width: 450px) and (max-width: 600px){
        img{
          width:40%; height:180px;
          background-color:#070750;
        }
        .header{ font-size: 35px; color:#000; }
        body{ background-color: #58a7eb; }
      }

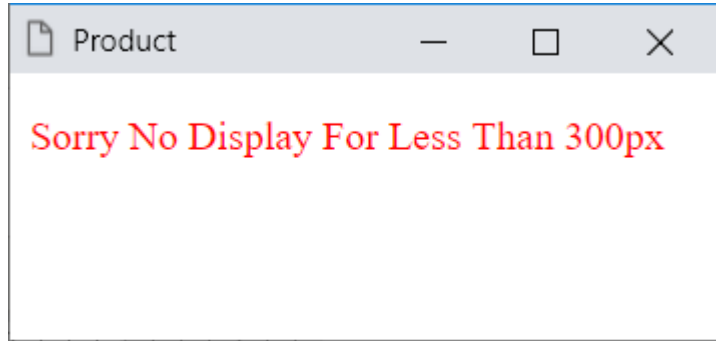
      @media (min-width: 600px) and (max-width: 1000px){
        img{
          width:25%; height:160px;
          background-color:#f484f0;
        }
        .header{ font-size: 45px; color:#fff; }
        body{ background-color: #550b16; }
      }
    </style>
  </head>
  <body>
    <div class="header">
      <img alt="Product image" data-bbox="100 194 902 915" />
    </div>
  </body>
</html>
```

```
}

@media (min-width: 1001px) and (max-width: 1200px){
  img{
    width:20%; height:140px;
    background-color:#144808;
  }
  .header{ font-size: 55px; color:#000; }
  body{ background-color: #95ef80; }
}

@media (min-width: 1201px){
  img{
    width:15%; height:120px;
    background-color:#eff481;
  }
  .header{ font-size: 55px; color:#fff; }
  body{ background-color: #86900f; }
}
</style>
</head>
<body>
  <p style="color:red" id="nodis">
    Sorry No Display For Less Than 300px
  </p>
  <div id="all">
    <p class="header">Welcome To Product Page</p>
    
    
    
    
    
    
    
    
    
    
    
    
  </div>
</body>
</html>
```


وبعد تشغيل الصفحة بعرض اقل من 300px تظهر كما بالشكل التالي:



وبعد التكبير أكثر تكون مثل الشكل التالي:



وبءء التءبء أءءء ءءءء الشءءء ءءءء:



وبعد التأكيد أنك ترون مثل الشكل التالي:



لغة CSS من البداية إلى الاحتراف - من اكاڤمية حسونة ٥ بفضل الله

أكاڤمية حسونة ٥ صخرة الأساس - ادخل واشترك وشاهد وشارك + ليصلك كل جديد
www.hassouna-academy.com www.youtube.com/user/HassounaAcademy

وبعد التكبيرة أنكه تكون مثل الشكل التالي:



صفحة رقم 955

ادخل وسجل حساب على الموقع الآن www.hassouna-academy.com

وبعد التأكيد أنك ترون مثل الشكل التالي:



مربع يتقلب ويرتعش كود! قم بكتابة الكود التالي ثم احفظه:

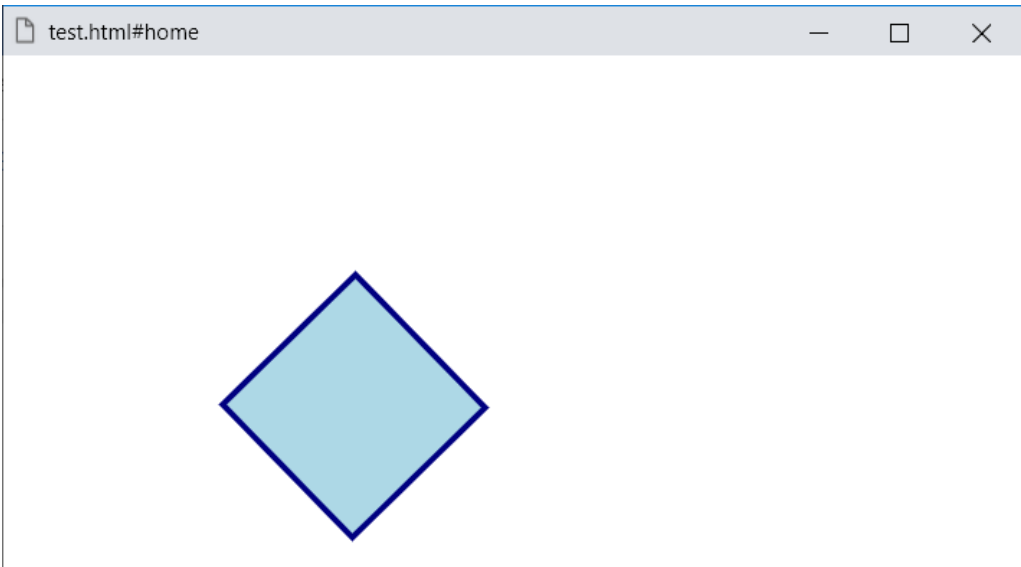
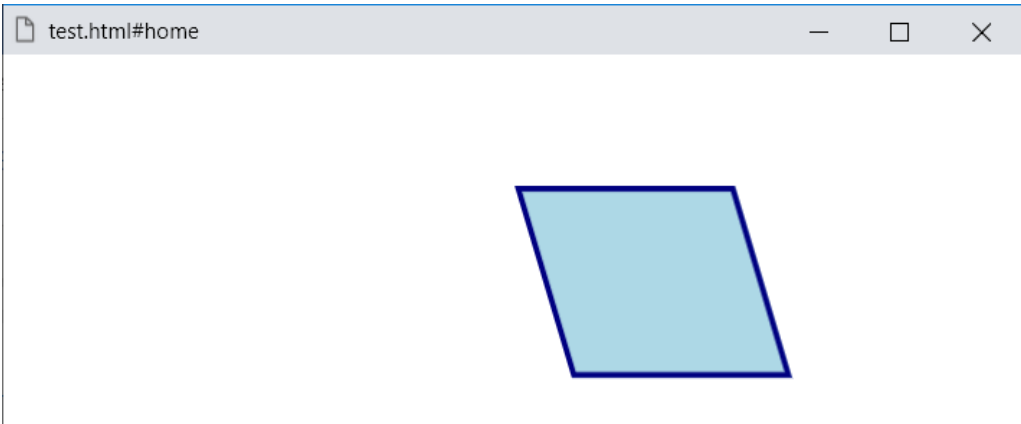
```
<!DOCTYPE html>
<html>
  <head>
    <style>
      #d1{
        width:100px; height:100px;
        background-color: lightblue; border:3px solid navy;
        position: absolute; top:10px; left:10px;
        animation-name:my1; animation-duration:9s;
      }
      @keyframes my1{
        0%   {left:10px ; top:10px ;}
        5%   {left:300px; top:10px ;}

        10%   {width:150px;}
        15%   {width:100px;}
        20%   {width:150px;}
        25%   {width:100px;}
        30%   {width:150px;}
        35%   {width:100px;}
        40%   {width:150px;}
        45%   {width:100px;}

        50%   {transform: skewX(30deg)}
        52%   {transform: skewX(0)}
        55%   {transform: skewX(30deg)}
        57%   {transform: skewX(0)}
        60%   {transform: skewX(30deg);}
        62%   {transform: skewX(0)}
        67%   {transform: skewX(30deg)}
        70%   {transform: skewX(0)}
        72%   {left:300px; top:300px ;}

        75%   {transform:rotate(360deg)}
      }
    </style>
  </head>
  <body> <div id="d1"></div> </body>
</html>
```

وبعد تشغيل الصفحة تظهر كما بالأشكال التالية:



تنويج روابط وازرار ومربعات نص كود! قم بكتابة الكود التالي ثم احفظه:

```
<!DOCTYPE html>

<html>
  <head>
<style>
body{
  background-color: #fff;
}

a{
  text-decoration:none;
  background:navy;
  color:lightblue;
  border:3px solid lightblue;
  border-radius:7px;
  padding:10px;
  margin:1px;
  font-size:20px;
  display:inline-block;
  transition:all 0.5s;
}

a:hover{
  background:lightblue;
  color:navy;
  border:3px solid navy;
  font-size:25px;
  margin-left:20px;
  margin-right:20px;
}

input[type="text"]{
  font-size:25px;
  width:300px;
  height:39px;
  background: whitesmoke;
  border:1px solid blue;
  border-radius:7px;
  outline:none;
  transition:all 0.5s;
}
```



```
input[type="text"]:focus{
  background: lightblue;
  border:2px solid yellow;
  width:320px;
}

input[type="button"]{
  font-size:25px;
  width:200px;
  padding:3px;
  background: darkred;
  color:pink;
  border:1px solid orange;
  border-radius:7px;
  outline:none;
  transition:all 0.5s;
}

input[type="button"]:hover{
  font-size:27px;
  width:210px;
  padding:5px;
  background: pink;
  color:darkred;
  border:1px solid gold;
}

#lang{
  display:none;
}

#pro{
  text-decoration:none;
  background:#6b046f;
  color:#ddade6;
  border:3px solid lightblue;
  border-radius:7px;
  padding:10px;
  margin:1px;
  font-size:20px;
  display:inline-block;
  transition:all 1.5s;
}

#pro:hover{
  padding:30px;
}
```

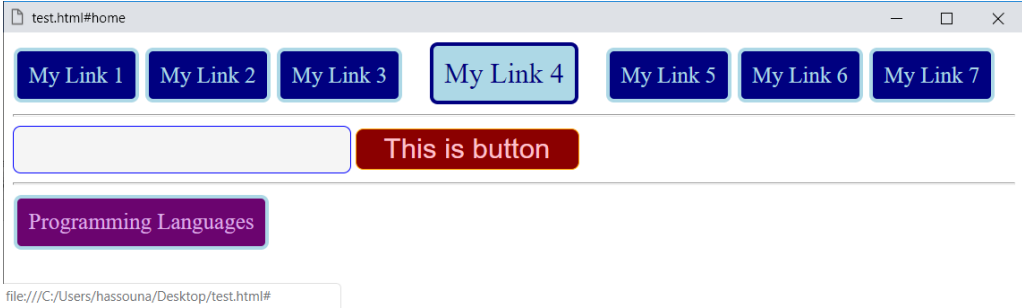
```
#pro a{
  margin-top:20px;
}

#pro:hover #lang{
  display:block;
}
</style>
</head>
<body>
  <a href="#">My Link 1</a>
  <a href="#">My Link 2</a>
  <a href="#">My Link 3</a>
  <a href="#">My Link 4</a>
  <a href="#">My Link 5</a>
  <a href="#">My Link 6</a>
  <a href="#">My Link 7</a>
  <hr>
  <input type="text" >&nbsp;<input type="button" value="This is button" >
  <hr>
  <div href="#" id="pro">Programming Languages
  <div id="lang">
    <a href="#">Java</a>
    <a href="#">Ring</a>
    <a href="#">Python</a>
    <a href="#">C Sharp</a>
    <a href="#">PHP</a>
    <a href="#">VB</a>
    <a href="#">Go</a>
  </div>
  </div>
</body>
</html>
```

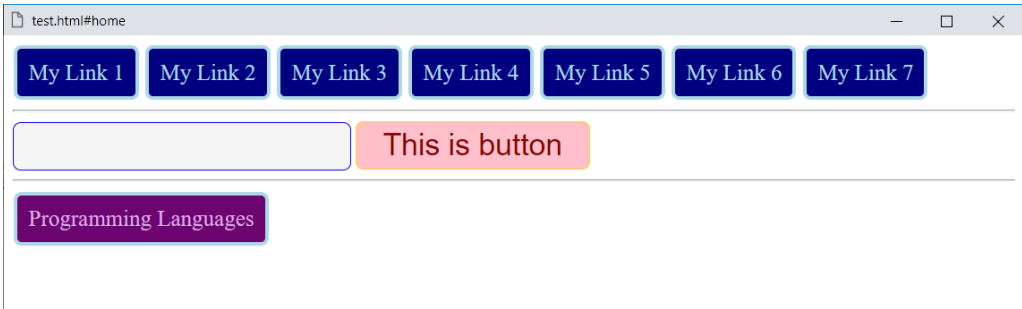
وبعد تشغيل الصفحة تظهر كما بالشكل التالي:



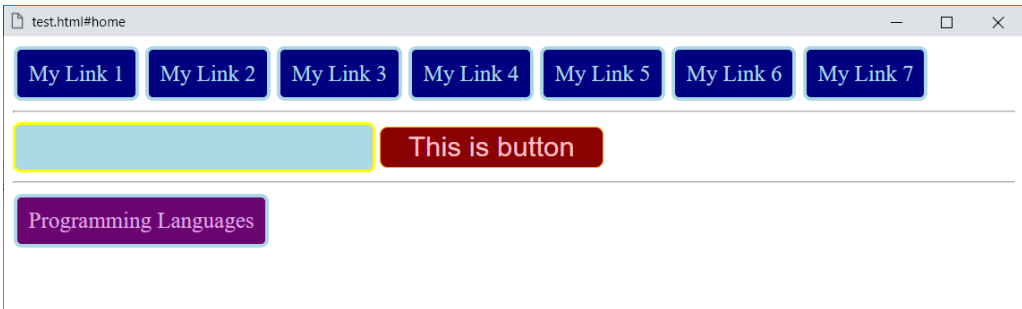
ولكن عند لمس الروابط تكون مثل ما يلي:



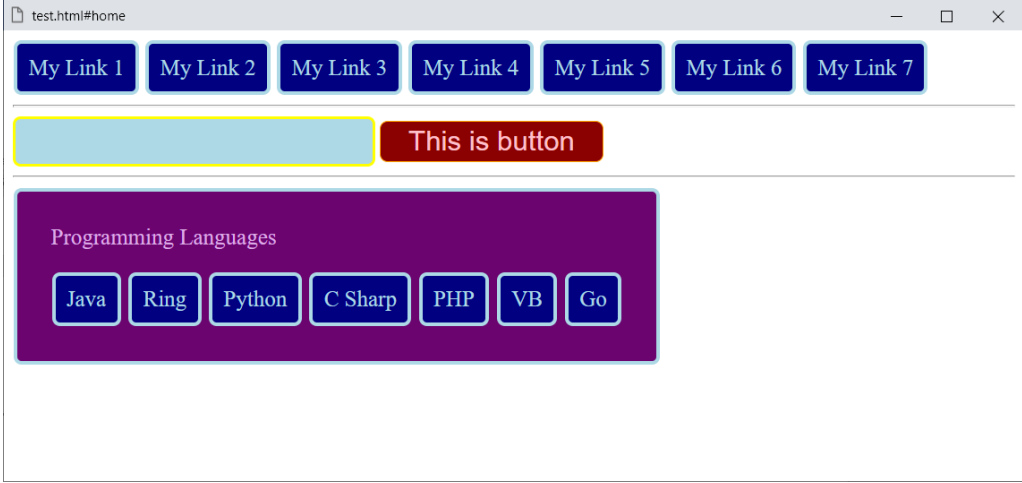
ولمس الزر يكون مثل ما يلي:



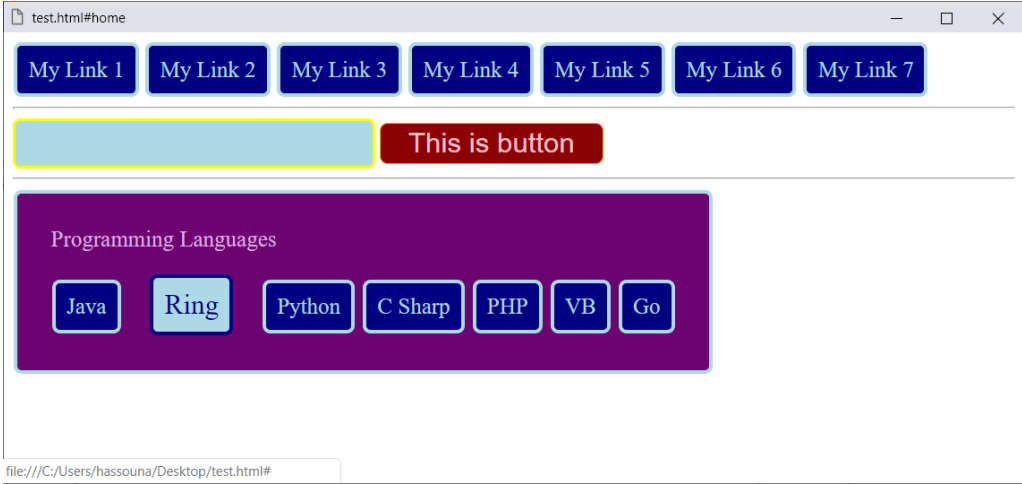
والدخول الى مربع النص يكون مثل ما يلي:



ولمسك لغات البرمجة يفتح مثل التالي:



ولمسحهم يكون مثل ما يلي:



كنوز وهدايا

اليك كنوز وهدايا، ووصيتي لك، أن تنشرها في كل مكان من الروابط الأصلية، وأيضاً وصيتي لك بأن تشاهدها وتنصح الجميع بمشاهدتها من اليوتيوب من أكاديمية حسونة، لتنجح هذه الكورسات بفضل الله (ﷻ).

شرح في قصص نجاح

<https://www.youtube.com/playlist?list=PLHIFW1KZRIfvUNrnmYsLPbU2t4oG0P-8>

شرح في الكمبيوتر بشكل عام

https://www.youtube.com/playlist?list=PLHIFW1KZRIfm_dFhGnawAln0OE1aSwoSN

شرح عن مقدمة في الحاسبات

https://www.youtube.com/playlist?list=PLHIFW1KZRIfmt3FFa1D6y-UjTshUC_zXM

شرح في أنظمة ماك ولينكس

<https://www.youtube.com/playlist?list=PLHIFW1KZRIfmTj0gddZTgcPdqDGP1qscF>

شرح في الويندوز Windows من الأساس وبالتفصيل

<https://www.youtube.com/playlist?list=PLHIFW1KZRIfKu8K8Dz603sxLa3eQ-g6G>

شرح في برنامج وورد Word من الأساس

https://www.youtube.com/playlist?list=PLHIFW1KZRIfk_q1_xWEJ8g1_nbPY_Y75A

ءءء فف برنامء اءسل Excel مء الأساس

<https://www.youtube.com/playlist?list=PLHIFW1KZRIfmeaZUramzmRwJVLQThBiuu>

ءءء فف برنامء الفوءوشوب Photoshop

<https://www.youtube.com/playlist?list=PLHIFW1KZRIfntLWyOJQKyfR5RcuXdzfnU>

ءءء فف البرامء الباءزة

https://www.youtube.com/playlist?list=PLHIFW1KZRIfmCe0esfmzTZTQ_AfoLOH1M

ءءء فف الءوارزمفاء وءرائء الءءق + أمءلة

<https://www.youtube.com/playlist?list=PLHIFW1KZRIfmOvX3-91zteoSP-b0VS44o>

ءءء آءر فف الءوارزمفاء وءرائء الءءق

<https://www.youtube.com/playlist?list=PLHIFW1KZRIflwIO8IiPxNkSSb-ekzNqDs>

ءءء فف ءءلل البفاءاء

https://www.youtube.com/playlist?list=PLHIFW1KZRIfkDoNSo0b00s_0Zn67DFAti

برنامء المبرمءفاء

<https://www.youtube.com/playlist?list=PLHIFW1KZRIfnLcjFtaI905vt1DKdSVM4K>

برنامء أساس مبرمء

https://www.youtube.com/playlist?list=PLHIFW1KZRIfkb-fDWH140jJowQX_Rztgk

ءءء فف لا اعرف شفاءء عء الكمبوءر وأرفء أن اءون مبرمء

<https://www.youtube.com/playlist?list=PLHIFW1KZRIfnKiRiVeW9H2k4TnwXZbaoc8>

شرح في منهج الصف الثاني الاعدادي

<https://www.youtube.com/playlist?list=PLHIfW1KZRIfmKjYjX1FJcSx1Q0JnHanpQ>

شرح في منهج الصف الثالث الاعدادي

https://www.youtube.com/playlist?list=PLHIfW1KZRIfmM2mbNfA_4jTI3vyK4iCOE

شرح في كورس قديم عن الويب HTML

https://www.youtube.com/playlist?list=PLHIfW1KZRIfmPHW37aNxMFosXD_p1jQz0

شرح في كورس جديد عن الويب HTML

<https://www.youtube.com/playlist?list=PLHIfW1KZRIfnXKjZ9UcT5BVt9x19B1ItX>

شرح في كورس عن الويب CSS

<https://www.youtube.com/playlist?list=PLHIfW1KZRIf106xnz4jpmzzSuo5B63aJf>

شرح في كورس عن الويب JavaScript

<https://www.youtube.com/playlist?list=PLHIfW1KZRIfmiYYwt29BG7N9zzsd-jLRQ>

شرح في دررشة برهجة عملي

https://www.youtube.com/playlist?list=PLHIfW1KZRIf10CJzVbN-DHGUJ_Y2DpPim

شرح في الويب جي كويري j-query

<https://www.youtube.com/playlist?list=PLHIfW1KZRIf1130bMFi02Ry7oRu3MJVRG>

شرح في الوب بوت ستراب bootstrap

<https://www.youtube.com/playlist?list=PLHIFW1KZRIfmzBcYnuLNrmdLxYCDzJgQi>

شرح في جاسوه JSON

<https://www.youtube.com/playlist?list=PLHIFW1KZRIf1A5jVQbAAHVizSxoe0Atgq>

شرح في انجولار Angular

<https://www.youtube.com/playlist?list=PLHIFW1KZRIf1btKBrCp5U2UdbeV1c7F-5>

شرح في الوب asp.net وسي شارب ومع الخدمات الأساسية

<https://www.youtube.com/playlist?list=PLHIFW1KZRIf1kDPH1OLTkv1daEp4pvIga>

شرح في الوب ومع AJAX

<https://www.youtube.com/playlist?list=PLHIFW1KZRIfmYd8YFztSmCzc3hcZXVua9>

شرح في قواعد البيانات MySQL

<https://www.youtube.com/playlist?list=PLHIFW1KZRIf1O4Gp9vMtUgry-xAOK17i>

شرح في الوب ولغة البرمجة PHP

<https://www.youtube.com/playlist?list=PLHIFW1KZRIfmPDQahhTDDd1DJ4PwQ13pC>

شرح في الوب ومع دبلومة الوب الكاملة

<https://www.youtube.com/playlist?list=PLHIFW1KZRIf1YXyXHNiG-hyyVaz7-9ujn>

شرح في الوب ومع سلسلة تعلم الفيديو الواحد

<https://www.youtube.com/playlist?list=PLHIFW1KZRIfnPUohWz2vPjm0obEjzZJD8>

شرح في الويب ومع صناعة آلة حاسبة بسيطة

<https://www.youtube.com/playlist?list=PLHIfW1KZRIfkYSPySS9Czt3Dxx2rsvqcN>

شرح في البرمجة واساسيات السي شارب C#

<https://www.youtube.com/playlist?list=PLHIfW1KZRIfm8nQAOJF5u2aV43tMRAAmr>

شرح في شاشات السي شارب C#

<https://www.youtube.com/playlist?list=PLHIfW1KZRIfnbNoGB0NdoRdl1q9fdo6uM>

شرح في البرمجة كائنية التوجه OOP وسي شارب C#

<https://www.youtube.com/playlist?list=PLHIfW1KZRIf16UP-PlUli03pokSc4af2S>

شرح في الويب مع سي شارب و ASP.NET C#

<https://www.youtube.com/playlist?list=PLHIfW1KZRIfmi8jwSbqiQuVfXeXIBk-bw>

شرح في عالم واسع مع قواعد البيانات DB مع C#

<https://www.youtube.com/playlist?list=PLHIfW1KZRIf1Aus00vgdVEzLUBCx8ooH>

شرح في أمثلة عملية مع قواعد البيانات و C#

<https://www.youtube.com/playlist?list=PLHIfW1KZRIfmYdNXeeabjt4mxSX5RWjzo>

شرح في أسرار قواعد البيانات Database

<https://www.youtube.com/playlist?list=PLHIfW1KZRIfnKHFH1uUwdpew8h-A81AI>

شده في ال LINQ مع C#

<https://www.youtube.com/playlist?list=PLHIFW1KZRIfnW4RD1n5tzW6htvNhnkr7t>

شده في عمل برنامج المطاعم والأكلان

<https://www.youtube.com/playlist?list=PLHIFW1KZRIfnK119p0w2oEDXc0AEpIfd5>

شده في عمل برنامج المبيعات الشامل

<https://www.youtube.com/playlist?list=PLHIFW1KZRIfm2VgnbTfoxoUS1xoag0UHX>

شده في دورة السي شارب العملاقة بفضل الله

<https://www.youtube.com/playlist?list=PLHIFW1KZRIfkDF2xTIB5kX8gdthmLTufx>

شده في قواعد البيانات الشامل والسي شارب C#

https://www.youtube.com/playlist?list=PLHIFW1KZRIfmg0C_60N0IRFWrifni6CTg

شده في ربط MySQL مع C#

<https://www.youtube.com/playlist?list=PLHIFW1KZRIfkQ7fgtqD4b4VuhxV6RzNu->

شده في ربط Oracle مع C#

<https://www.youtube.com/playlist?list=PLHIFW1KZRIf10vnBDg28R0DW9HBkyHvc->

شده في عمل متصفح بالسي شارب

<https://www.youtube.com/playlist?list=PLHIFW1KZRIf17ToU9EDuRVYiki9C9wxnJ>

شده في كورس سي شارب مختصر

<https://www.youtube.com/playlist?list=PLHIFW1KZRIfmIhZWCyXnY1xz8ALlfPRAn>

شرح في الويب ومع الدورة التنفيذية في سي شارب و ASP.NET

<https://www.youtube.com/playlist?list=PLHIfw1KZRIfn6kI3NyvzTDM-wdw9EX7D6>

شرح في البرمجة ومع اساسيات فيجوال بيسك VB.NET

https://www.youtube.com/playlist?list=PLHIfw1KZRIfkVIMpDSgFTJsR_I01dH9i

شرح في شاشات الفيجوال بيسك VB.NET

<https://www.youtube.com/playlist?list=PLHIfw1KZRIfn7Pz7MnzABWryuHRP9ab70>

شرح في دورة الفيجوال بيسك العملاقة ياذن الله

<https://www.youtube.com/playlist?list=PLHIfw1KZRIfmLGHfsQ1G5cPv1xokzBsbT>

شرح في فيجوال بيسك وسكول سيرفر مع التقارير الكاملة

<https://www.youtube.com/playlist?list=PLHIfw1KZRIfkKhf3ha77qccAWYmvgM7LR>

شرح في ربط VB.NET مع Oracle

https://www.youtube.com/playlist?list=PLHIfw1KZRIf1AkiMFHY-0Jc_XXnpLW000

شرح في ربط VB.NET مع Access

<https://www.youtube.com/playlist?list=PLHIfw1KZRIfkdbHPdTbpcYZzPQtPUeR7I>

شرح في الفيجوال بيسك القديم VB 6

<https://www.youtube.com/playlist?list=PLHIfw1KZRIfk5L0D0QbV8DyUwfoAhQyieg>

شده في البرمجة مع أساسيات لغة الجافا JAVA

<https://www.youtube.com/playlist?list=PLHIFW1KZRIf1flvzYY7B1d5CJjXpbIriU>

شده في بداية البرمجة الكائنية مع جافا JAVA

https://www.youtube.com/playlist?list=PLHIFW1KZRIfmb25mVAsCJy_Ehgu5rbbQx

شده في البرمجة الكائنية مع جافا JAVA

https://www.youtube.com/playlist?list=PLHIFW1KZRIfnT2i7Ba4F2nrWb_ENXjyTs

شده في جافا مع قواعد البيانات

<https://www.youtube.com/playlist?list=PLHIFW1KZRIfnKQjOoZc540z3fmGe-s1LU>

شده في جافا مع اندرويد Android

<https://www.youtube.com/playlist?list=PLHIFW1KZRIfnO6WRvLbWok4ZZww4k3T1u>

شده في شاشات الجافا

<https://www.youtube.com/playlist?list=PLHIFW1KZRIf12XxZwryDuP05ZLkRhL-CS>

شده في دورة الجافا العملاقة بفضل الله

https://www.youtube.com/playlist?list=PLHIFW1KZRIfn9BnepQuzwiM_ZPIwUDawL

شده في أمثلة مع جافا

https://www.youtube.com/playlist?list=PLHIFW1KZRIfnUMvey6UAveK5msn8L_u1S

شده في التعامل مع ملف ال jar للجافا

<https://www.youtube.com/playlist?list=PLHIFW1KZRIfnDa6squg40jDjYfx5YC8Zx>

شرح في البرمجة مع لغة البرمجة كوتل Kotlin

<https://www.youtube.com/playlist?list=PLHIfW1KZRIf15UHtM6DRFAVn3PBuJ7NGD>

شرح في كلاس مع لغة كوتل Kotlin

<https://www.youtube.com/playlist?list=PLHIfW1KZRIf1KgeC1VLxHJXFqIsV0Y>

شرح في كوتل مع اندرويد Android

<https://www.youtube.com/playlist?list=PLHIfW1KZRIfmbpu6cVCxo98uYk0Hfxah>

شرح في البرمجة ومع اساسيات البايثون Python

<https://www.youtube.com/playlist?list=PLHIfW1KZRIfmm35tXSbxsFDouRJJAPP9Y2>

شرح في البرمجة الكائنية OOP مع البايثون

<https://www.youtube.com/playlist?list=PLHIfW1KZRIfmHvfmFZZ0XuzZYu0osG9k1>

شرح في البرمجة مع لغة البايثون Python اساسيات و OOP

<https://www.youtube.com/playlist?list=PLHIfW1KZRIfnVNMUcSLFFyDzuLDheF99n>

شرح في شاشات البايثون Python

<https://www.youtube.com/playlist?list=PLHIfW1KZRIfmMhy8GltKHW5UJZPnGJ0K>

شرح في دورة البايثون العملاقة بفضل الله

<https://www.youtube.com/playlist?list=PLHIfW1KZRIfnM9y0sQRwjVz2-IwvnEJep>

شءء فف البرمءة مع لغة الرنءء Ring

<https://www.youtube.com/playlist?list=PLHIFW1KZRIf16KzfLziF1650MmThnQ0jt>

شءء فف البرمءة الءائفة OOP ورنءء Ring

<https://www.youtube.com/playlist?list=PLHIFW1KZRIf15TA4nTIIVWsfDyVwIBP6y>

شءء فف ءءاشاء رنءء Ring

<https://www.youtube.com/playlist?list=PLHIFW1KZRIfmE8gCtJ0pIOGmBz3bc Nrs>

شءء فف البرمءة مع لغة السف بلس بلس ++ C

<https://www.youtube.com/playlist?list=PLHIFW1KZRIfkz N9aNWXRByEJOudCxSRh>

شءء مءنوء وعام فف عالم البرمءة

<https://www.youtube.com/playlist?list=PLHIFW1KZRIf16aI0bGbzWZYGHc2ctcW7>

شءء فف قواعء الءبائاء SQL Server

<https://www.youtube.com/playlist?list=PLHIFW1KZRIf1vUe-YfDpIqqnU70BopQ3q>

شءء فف قواعء الءبائاء ولغة السكول SQL

<https://www.youtube.com/playlist?list=PLHIFW1KZRIfmNd2URuqV05Itboj LQvrx>

شءء فف قواعء الءبائاء مع أكسس Access

<https://www.youtube.com/playlist?list=PLHIFW1KZRIfnKpy1msMUPqVYdpu3n0dkY>

شءء فف قواعء بئائاء الوسائء المءعءة Multimedia DB

<https://www.youtube.com/playlist?list=PLHIFW1KZRIfmMYvE8mnEH4VT-4v21XV6p>

شرح في عمل برنامجك مع الأداة Magic DB

<https://www.youtube.com/playlist?list=PLHIFW1KZRIfnkb0hIry8E0LbA0f1haXT0>

شرح في برنامج امتحان الطلاب

<https://www.youtube.com/playlist?list=PLHIFW1KZRIfnd5v0u8hxwTKXebJK90pgQ>

تسالي مبرمجيه

https://www.youtube.com/playlist?list=PLHIFW1KZRIf1zc_du0yEhaCCPZRAuHv

أكاديمية حسونة

https://www.youtube.com/playlist?list=PLHIFW1KZRIfmxKFdzlyAna7sYE_L2hI0o

لقاءات وأحداث Events

https://www.youtube.com/playlist?list=PLHIFW1KZRIfmFsmWKbc_rf6qAPbVykmJ


شكراً خاصاً

الشكراً أولاً وأخيراً وابتداءً وانتهاءً لله عز وجل، ونشكراً كل من:


- شكراً لأبي وأمي اللذان رباني صغيراً.
- شكراً لك أصدقائي واصحابي واحبابي في كل مكان.
- شكراً لك من عرف قدر العلم.
- شكراً لك طالب أو طالبة أو اخ أو صاحب أو صديق أو حبيب ينشر هذا العمل في كل مكان ليفيد الجميع بإذن الله.
- شكراً للأستاذ احمد باريان علي دعمه في نشر العلم في كل مكان.
- شكراً للمهندس أحمد إبراهيم سالم (Ahmed Ibrahim)
- شكراً للدكتور عادل عبد الصبور (Adel Sabour).
- شكراً للمهندس محمود سمير فايد (Mahmoud Fayed).
- شكراً للمهندس أسامة محمد (Osama Elzero).
- شكراً للقائمي علي موقع developer.mozilla.org
- شكراً للقائمي علي موقع wikipedia.org
- شكراً للقائمي علي موقع w3.org

نشكرك على قراءتك

أولاً: نتقدم لك بخالص الشكر 🤝 على قراءتك هذا الكتاب، ونتمنى لك الاستفادة الكاملة من محتواه.

ثانياً: لا تنسى أن تسجل لنفسك حساب  في أكاديمية حسونة لتنضم إلينا وتشارك معنا كل محتويات الموقع، سجل من الرابط التالي:

<https://www.hassouna-academy.com/register>

ثالثاً: يسر أكاديمية حسونة أن تخبرها برأيك عن هذا الكتاب من خلال تقديم كلمة شكرك  من الرابط التالي:


<https://www.hassouna-academy.com/thanks>

المراجع References

الءءول الءالف فءءوئ على المراجع الءف تم الرجوع إليها أثناء تألف هءا الءاب، سواء كانت مراجع ءعلفمفة مرئفة أو مقروءة، أو معلوماةفة.

المراجع	الراءب
أكاڤمفة الزفرو	http://elzero.org/
v نظم ءءنولوجفا المعلوماء	http://www.7ist.net/
أكاڤمفة حسونة	https://www.hassouna-academy.com/
وفاكفبفءفا	https://www.wikipedia.org/
منظمة الوب العالمفة	https://www.w3.org
mozilla	https://developer.mozilla.org

الخاتمة The End

والآن، أتوجه الى **أحبابي** بكلمة من القلب ، خالصة إن شاء الله، فإني
معي في اختراعات الإنسان، فإني في تطويرات وصناعات الإنسان، فإني في
الإنسان وقدراته، وقل سبحان الله الذي خلق الإنسان، فإني في قول الله عز
الإنسان في القرآن الكريم "لَقَدْ خَلَقْنَا الْإِنْسَانَ فِي أَحْسَنِ تَقْوِيمٍ"، فإني في
قدرة الله الواحد الأحد، الفرد الصمد، الذي لم يلد ولم يولد، ولم يكن له
كفولاً أحداً، فإني في الله العظيم، الذي خلق السماوات والأرض، فإني في رب
العزة، فإني في رب الكون، وقل، سبحان الله، وقل، سبحان الملك العظيم،
وقل سبحان ذي العزة والجبروت، سبحان ذو الملك والملاوت، سبحان الله
العظيم، سبحان الله الكريم، سبحان الله، عدد خلقه، ورضا نفسه، وزنة
عرشه، ومداد كلماته، وآخر دعوانا "الحمد لله رب العالمين".