

# المختصر المفيد في CSS

اعداد : أحمد إبراهيم

PHP

C#

VB

jQuery

JavaScript

HTML

AJAX

JAVA

ASP

C++

CSS



## فهرس الكتاب

3	..... المقدمة
4	..... البداية مع CSS
5	..... الفصل الأول ( مظهر النصوص والخطوط )
5	..... أولاً :- خصائص مظهر النصوص ( text )
11	..... ثانياً :- خصائص الخطوط
13	..... الفصل الثاني ( الحدود والهوامش والحشو )
20	..... الفصل الثالث ( مظهر القوائم والخلفيات )
20	..... أولاً :- خصائص الخلفيات
25	..... ثانياً :- خصائص مظهر القوائم
27	..... الفصل الرابع ( خصائص متنوعة )
34	..... الفصل الخامس ( التحويلات الهندسية وخصائص الحركة )
37	..... الفصل السادس ( الوسوم الزائفة )
45	..... ملاحظات عامة

## المقدمة

### كيف فكرت بانشاء هذا الكتاب ؟

فكرت بتعلم تصميم مواقع الإنترنت وأنا لم يكن لي أي خلفية مسبقة في البرمجة ولست صاحب أختصاص في هذا المجال وانا مجرد هاوي أعشق البرمجة ولازلت أتعلم الى حد هذه اللحظة عندما بدأت أتعلم البرمجة ( ولا ازال ) أحببتها بشكل جنوني وأصبحت البرمجة تأخذ كل وقتي حيث أصبحت أمضي في اليوم من 10 الى 15 ساعة امام شاشة اللابتوب وفي بعض الأحيان استمر لأكثر من 30 ساعة بشكل متواصل , في البداية أكبر مشكلة واجهتني هي أنه لا توجد كتب ضخمة وتشمل كل جوانب لغة البرمجة والمحتوى العربي ضعيف للغاية والمشكلة الأكبر هي أنني لا أجيد اللغة الإنجليزية بشكل جيد لذا تقريباً قرأت أكثر الكتب العربية التي كتبت في هذه اللغة وبما أن أكثرها هو عبارة عن مجموعة صفحات ليس إلا وأغلب المعلومات التي فيه هي فقط أساسيات اللغة وعند قراءة كتاب تجد إن كل مافيه تقريباً هو موجود في كتاب آخر مع إضافة بسيطة لذا فكرت بأن أجمع كل معلومة جديدة أتعلمها في هذه اللغة وادونها بشكل مختصر ليسهل علي الوصول لها عند نسيانها لذا أنشأت مرجع بسيط خاص بي لهذه اللغة , وبعد فترة أصبح فيه مجموعة لا بأس بها من المعلومات ففكرت أن أنشرها بشكل كتاب ليستفاد منها كل من يحب تعلم لغة الـ CSS , لذا قد تجدون في هذا الكتاب الكثير من الإختصار في شرح الأوامر ولم أضع أمثلة مطوله بل فقط أمثلة بسيطة لتصل الفكرة .

وبنفس الطريقة أنشأت مراجع خاصة بي تشمل عدة لغات وتقنيات منها مرجع للغة HTML ومرجع للغة PHP ومرجع للغة JavaScript وسأنتشر هذه الكتب قريباً لتكون متاحة للجميع .

ملاحظة/ هذه هي الطبعة الأولى من الكتاب نشرت بتاريخ 2014/4/20 ومن المحتمل أن أضيف على هذا المرجع الصغير بعض الإضافات وأجري عليه بعض التعديلات وأنشره كطبعه ثانية ( في المستقبل ) .

ارحب بكل أسئلتكم واستفساراتكم أيضاً كل اقتراحاتكم لأنه كما قلت سابقاً إنني لازلت في بداية طريقي في البرمجة ولا زلت اتعلم شيء جديد كل يوم .

يمكنكم التواصل معي على العنوان التالي :- [ahmedcia17@gmail.com](mailto:ahmedcia17@gmail.com)

ملاحظة/ يفترض بالقارئ أن يعرف كيفية التعامل مع لغة html قبل أن يقرأ هذا الكتاب .

أحمد إبراهيم

## البداية مع CSS

CSS تعني بالانجليزية ( Cascading Style Sheets ) ومعناها بالعربية ( صفحات الأنماط الانسيابية ) وهي ليست لغة برمجة وإنما تساعد على تنسيق وترتيب صفحة html لأفضل شكل .

### طرق كتابة اكواد CSS

توجد ثلاثة طرق لكتابة اكواد CSS :-

**1-** كتابة التنسيق في القسم **style** في داخ أي وسم من وسوم html ( في داخل صفحة html , مثال /

```
<p style="font-size:20pt; font-family:Tahoma; > the text </p>
```

**2-** كتابة كود CSS في الجزء **<head>** في صفحة html , وذلك بكتابة وسم البداية

```
<style type="text/css"> ووسم النهاية </style> وبينهم نضع التنسيق , مثال /
```

```
<html> <head>  
<style type="text/css">  
p { font-size:20pt; color:#000; }  
</style> </head>  
<body>  
<p> the text </p>  
</body> </html>
```

**3-** كتابة الكود في ملف خارجي عن صفحة html :- وهذه أفضل طريقة إذا كانت لدينا عدة صفحات ( صفحات كثيرة ) حيث فرضاً لو أردت تعديل لون الجدول لن نتعب وتغير جميع الصفحات ! بل ستقوم بتعديل ملف CSS فقط ! أليس هذا رائعاً ؟  
لكتابة CSS في صفحة مستقلة ومن ثم استدعائها :-

أولاً :- عند كتابة صفحة CSS لا نكتب الوسوم بين وسمي النهاية والبداية الخاصة ب CSS  
<style type="text/css" >-----</style> وتكتب وسوم الهوتميل مباشرة

```
p { color:red;
      background-color:yellow; }
h1 { color:black; }
```

ثانياً :- عند استدعاء صفحة CSS يجب أن يكتب كود الاستدعاء بين وسمي <head> وكود الاستدعاء هو :-

```
<link rel="stylesheet" type="text/css" href=" اسم الملف .css " />
```

## الفصل الأول ( مظهر النصوص والخطوط )

أولاً :- خصائص مظهر النصوص ( text )

### • Text-transform:

هذه الخاصية تستخدم لتغيير حالة الأحرف ما بين صغيرة او كبيرة , وتأخذ احد هذه القيم :-

1. Uppercase لجعل الحروف كبيرة .
2. lowercase لجعل الحروف صغيرة .
3. capitalize لجعل الحرف الاول فقط من كل كلمة حرفاً كبيراً .

مثال/

```
p { text-transform:uppercase; }
```

### • text-indent:

تستخدم هذه الخاصية لترك مسافة فاصلة في بداية كل فقرة , وهي تأخذ قيمة رقمية مثلاً :-

20px

مثال/

```
p { text-indent:10px; }
```

\* ويمكن اعطاء قيمة رقمية كبيرة بالسالب ليختفي النص الظاهر من على العنصر وهذا يستخدم غالباً مع الأزرار والروابط , مثال /

```
a{ text-indent:-9999px; }
```

### • text-decoration:

وهي تضع خط تحت الوصلة التشعبية او النص المكتوب وتحدد مكان الخط في الاسفل او الاعلى او المنتصف او عدم ظهور اي خط , وهي تأخذ احد الخواص التالية :-

- |                 |                            |
|-----------------|----------------------------|
| 1. underline    | تجعل خط اسفل النص .        |
| 2. overline     | تجعل خط فوق النص .         |
| 3. line-through | تجعل الخط في منتصف النص .  |
| 4. none         | تخفي الخط ولا تجعله ظاهر . |

مثال/

`a { text-decoration:none; }`

### • text-align:

وهي تستخدم لمحاذاة النص وتحديد مكانه وكذلك لمساواة اسطر الفقرة , وهي تأخذ احد القيم التالية :-

- |            |                                 |
|------------|---------------------------------|
| 1. left    | لمحاذاة النص إلى جهة اليسار .   |
| 2. right   | لمحاذاة النص إلى جهة اليمين .   |
| 3. top     | لمحاذاة النص إلى الأعلى .       |
| 4. bottom  | لمحاذاة النص إلى الأسفل .       |
| 5. center  | لمحاذاة النص إلى المنتصف .      |
| 6. justify | تجعل جميع اسطر الفقرة متساوية . |

مثال/

`p { text-align:left; }`

### • direction

تستخدم هذه الخاصية لتحديد اتجاه النص من اليمين إلى اليسار او بالعكس , وتأخذ القيمتين التاليتين :-

1. rtl من اليمين إلى اليسار .
2. ltr من اليسار إلى اليمين .

مثال/

```
body { direction:rtl; }
```

### • letter-spacing:

تستخدم هذه الخاصية لتبعد بين الحروف في الكلمة الواحدة وتأخذ قيمة رقمية .

مثال/

```
p { letter-spacing:10px; }
```

### • word-spacing:

تستخدم هذه الخاصية للتباعد بين الكلمات ( بين كلمة وكلمة أخرى ) وهي تأخذ قيمة رقمية .

مثال/

```
p { word-spacing:10px; }
```

### • white-space

في الـ html اذا كتبنا نص وضغطنا enter من لوحة المفاتيح كتبنا نص آخر فإنه سيظهر في الموقع وكأنه نص واحد أي وكأننا لم نضغط على enter ( وطبعاً حتى لو ضغطنا أكثر من مرة ) وإذا أردنا أن يظهر في الموقع سطر جديد كلما نضغط على enter نستخدم هذه الخاصية ونسند لها القيمة pre , ولاحظ أن في هذه القيمة سيحسب الفراغات وكل المسافات المتروكة , وإذا أردناه أن يحسب فقط الأسطر الجديدة ولا يحسب الفراغات نضع القيمة

pre-line



مثال

`white-space:pre-line;`

### ● line-height:

تستخدم هذه الخاصية لتحديد التباعد بين اسطر الكتابة , بمعنى لو كان لدينا نص كبير يمتد لعدة اسطر يمكننا التحكم بتباعد هذه الأسطر من خلال هذه الخاصية , مع ملاحظة أن هذه الخاصية تطبق إذا كان هذا النص الكبير في فقرة واحدة ( وسم html واحد ) وليس في أكثر من فقرة , وهي تأخذ قيمة رقمية .

مثال/

`p { line-height:30px; }`

### ● text-shadow:

هذه خاصية جديدة في **CSS3** , وهي تستخدم لإظهار ظل للنصوص وتأخذ ثلاثة قيم رقمية وقيمة لونية , حيث إن القيمة الأولى تمثل بعد الظل عن النص بشكل أفقي والقيمة الثانية تمثل بعد الظل عن النص بشكل عمودي والقيمة الثالثة تمثل شدة وضوح الظل أما الأخيرة وهي القيمة اللونية فهي تمثل لون الظل .

مثال/

`h2 { text-shadow:10px 10px 5px #333; }`

❖ ويمكن أن نضيف أكثر من ظل لنفس النص وذلك بكتابة القيم الأربعة أسفل قيم خاصية الظل الأولى مع استبدال الفارزة المنقوطة في نهاية الخاصية الأولى إلى فارزة فقط .

مثال/

`text-shadow:7px 5px 3px #333,`

`10px 10px 3px #fff;`

وبنفس الطريقة يمكن إضافة ظل ثالث أو رابع أو أكثر كما نشاء .

- هناك مجموعة جديدة من الخصائص تستخدم في CSS3 للتعديل على النصوص ويجب أن نسبقها بالبائنة الخاصة بكل متصفح لتعرض في المتصفحات المختلفة , وهي موضحة في الجدول في الأسفل :-

الخاصية	القيم الممكنة	الشرح
hanging-punctuation	none / last / allow-end / first / force-end	تحدد موضع علامة الترقيم خارج حدود العنصر <p>
punctuation-trim	start / end / allow-end / adjacent / none	تزيل علامة الترقيم الافتتاحية من بداية كل سطر من اسطر <p>
text-justify	auto / inter-word / inter-ideograph / kashida / distribute / inter-cluster / trim /	تحدد طريقة تحديد المحاذاة عندما تكون قيمة الخاصية ( text-align ) تساوي ( justify )
text-outline	قيمة بالبيكسل تعبر عن العرض وقيمة لونية تعبر عن اللون	تحدد عرض ولون الحدود الخارجية للنص
text-overflow	clip / ellipsis / string	تحدد ما الذي يتم عمله عندما يصبح طول النص أكبر من النص الحاوي له
text-wrap	none / normal / unrestricted / suppress	تحدد طريقة تحقق التقاف النصوص
word-break	normal / break-all / hyphenate	تحدد طريقة تقسيم الكلمات الطويلة
word-warp	normal / word-break	تقوم بتحقيق التقاف النصوص عبر تقسيم الكلمات الطويلة بناءً على الطريقة المحددة في الخاصية word-break

## ثانياً :- خصائص الخطوط

### • font-style:

تستخدم هذه الخاصية للتحكم بالخطوط وتأخذ احد القيم التالية :-

1. **italic** تجعل الخط مائل .
2. **normal** تجعل الخط عادي وهي القيمة الافتراضية .

مثال/

```
p { font-style:italic; }
```

### • font-size:

وهي تستخدم لتحديد حجم الخط وتأخذ قيمة رقمية مثلاً 10pt

مثال/

```
p { font-size:13pt; }
```

### • font-family:

وهي تستخدم لتحديد نوع الخطوتكون قيمتها عبارة عن اسم الخط مثلاً **andalus** , ويمكن أن تأخذ قيم أخرى بالإضافة إلى اسم الخط وهي اسم عائلة من العوائل الثلاثة وهي :-

1. **sans-serif** يفضل استعمالها مع اللغة الانجليزية .
2. **monospace** ينصح باستعمالها مع الشيفرات الحاسوبية مثل شيفرات لغات البرمجة .
3. **serif**

مثال/

```
body { font-family:tahoma; }
```

### • font-weight:

وهي تستخدم لتحديد سماكة الخط , ونأخذ احد القيم التالية :-

1. **bold** تجعل الخط سميك .
2. **normal** ليكون النص طبيعي ( القيمة الافتراضية ) .

مثال/

```
p { font-weight:bold; }
```

### • font:

تمكننا هذه الخاصية من تطبيق مجموعة من تأثيرات الخط بشكل مختصر حيث إنها ستأخذ عدة قيم وكل قيمة تمثل تأثير خاصة معينة .

مثال/

```
p { font:italic bold 15px Tahoma; }
```

مع ملاحظة أن هذه القيم يتم كتابتها بالترتيب وأي قيمة لا نريدها يمكن أن لا نكتبها , حيث إن القيمة الأولى تمثل شكل الخط والثانية تمثل سمك الخط والثالثة تمثل حجم الخط والرابعة تمثل نوع الخط .

### • color:

تستخدم هذه الخاصية لتحديد لون الخط وهي تأخذ قيمة لونية .

مثال/

```
p { color:#00ffcc; }
```

## الفصل الثاني

### ( الحدود والهوامش والحشو )

#### • border-collapse

هذه الخاصية تستخدم مع الجداول فقط , وهي تحدد هل سيتم طي حدود الخلايا واعتبار وجود حد واحد بين الخليتين ( القيمة المسئولة عن ذلك هي collapse ) أم سيتم اعتبار حد كل خلية مستقلاً عن الآخر ( مسافة بين حد كل خلية وأخرى ) .

مثال/

```
table { border-collapse:collapse; }
```

#### • border-style

وهي تستخدم لوضع إطار حول فقرة معينة أو نص معين , وتأخذ احد القيم التالية لتحديد نوع هذا الاطار :-

1. outset تستخدم للحدود ثلاثية الابعاد .
2. inset تستخدم للحدود ثلاثية الابعاد .
3. groove تستخدم للحدود ثلاثية الابعاد .
4. double تستخدم للحدود المضاعفة .
5. solid تستخدم للحدود العادية .
6. ridge تستخدم للحدود ثلاثية الابعاد .
7. dashed تستخدم للحدود المتقطعة .
8. dotted تستخدم للخطوط المنقطعة .

مثال/

```
div { border-style:groove; }
```

ويمكن ان نعطي لكل جهة من الاطار نوع اطار مختلف وذلك بكتابة الاتجاه ما بين كلمة **border** و **style** هكذا

**border-style-الاتجاه** , مثال/

**div { border-top-style:solid; }**

ويمكن ان نكتب كل الاتجاهات بشكل مختصر وذلك عن طريق اسناد اربع قيم للخاصية حيث ان القيمة الاولى تمثل الجهة العلوية والثانية تمثل اليمين والثالثة الاسفل والرابعة اليسار ( باتجاه عقرب الساعة ) ونفصل بين كل قيمة واخرى بفاصلة واحدة , مثال/

**div { border-style:solid dotted ridge double; }**

• **border-width:**

وهي تستخدم لتحديد سمك الاطار وتأخذ قيمة رقمية مثلاً 2px او تاخذ احد هذه القيم :-

- |                      |                  |
|----------------------|------------------|
| للإطار الضعيف .      | <b>thin</b> .1   |
| للإطار متوسط السمك . | <b>Medium</b> .2 |
| للإطار السميك .      | <b>thick</b> .3  |

مثال/

**p { border-width:1px; }**

ويمكن ان نعطي لكل جهة من الاطار سمك معين وذلك بكتابة الاتجاه ما بين كلمة **border** و **width** هكذا

**border-width-الاتجاه** , مثال/

**p { border-left-width:2px; }**

ويمكن ان نكتب كل الاتجاهات بشكل مختصر وذلك عن طريق اسناد اربع قيم للخاصية حيث ان القيمة الاولى تمثل الجهة العلوية والثانية تمثل اليمين والثالثة الاسفل والرابعة اليسار ( باتجاه عقرب الساعة ) ونفصل بين كل قيمة واخرى بفاصلة واحدة , مثال/

**p { border-width:1px 2px 6px 1px; }**

### • border-color

وهي تستخدم لتحديد لون الاطار وتأخذ قيمة لونية , مثال/

```
p { border-color:#e3e; }
```

ويمكن ان نعطي لكل جهة من الاطار لون معين وذلك بكتابة الاتجاه ما بين كلمة **border** و **color** هكذا :-

**border-الاتجاه-color** , مثال/

```
p { border-bottom-color:red; }
```

ويمكن ان نكتب كل الاتجاهات بشكل مختصر وذلك عن طريق اسناد اربع قيم للخاصية حيث ان القيمة الاولى تمثل الجهة العلوية والثانية تمثل اليمين والثالثة الاسفل والرابعة اليسار ( باتجاه عقرب الساعة ) ونفصل بين كل قيمة واخرى بفاصلة واحدة , مثال/

```
p { border-color:#e3e #333 #fff red; }
```

### • border:

تستعمل هذه الخاصية لتحديد سمك ونوع ولون حدود الإطار , وهي عبارة عن اختصار للخواص السابقة حيث يمكن أن نجمع في هذه الخاصية الثلاث خواص السابقة وهي ( **border-width** و **border-style** و **border-color** ) حيث تكون قيمتها عبارة عن ثلاثة قيم كل قيمة تمثل خاصية لكن يجب الانتباه إلى الترتيب في وضع القيم حيث أن القيمة الأولى تمثل سمك الإطار والثانية تمثل نوع الإطار والثالثة تمثل لون الإطار , ونفصل بين كل قيمة وأخرى بفاصلة واحدة .

مثال/

```
p { border:1px solid #333; }
```

### ● outline:

هذه الخاصية تمثل الحدود الخارجية للإطار حيث يمكن مثلاً أن نعطي لفقرة معينة إطار لكنه يكون ذو لونين حيث إن اللون الأول يمثل الإطار ( border ) واللون الثاني يمثله ما هو خارج الإطار ( outline ) وهذه الخاصية تأخذ قيم الخاصية ( border ) نفسها .

مثال/

```
p { border:2px solid red;
  outline:5px ridge #3ff; }
```

❖ ويمكن أن نكتب الخاصية ( outline ) لوحدها أي ليس من الضروري أن نستخدمها هي والخاصية ( border ) معاً في نفس الفقرة .

### ● padding:

تستخدم هذه الخاصية لتحديد المسافة المحيطة بالنص او الصورة أو المسافة الفاصلة بين النص والإطار وهي تأخذ قيمة رقمية مثلاً 5em , وفي هذه الحالة سوف يترك المسافة المحيطة بالعنصر من جميع الاتجاهات لكن ويمكن أن نحدد المسافة من جهة واحدة او اكثر وذلك باضافة الاتجاه بعد الخاصية padding هكذا :- الاتجاه-padding .

مثال/

```
p { padding-left:4em;
  padding-top:3em; }
```

ويمكن أن نضيف قيم الاتجاهات الأربعة إلى الخاصية padding: وذلك بإعطائها أربع قيم حيث إن القيمة الأولى تمثل الجهة العلوية والثانية تمثل اليمين والثالثة تمثل الأسفل والرابعة تمثل اليسار ( باتجاه عقارب الساعة ) حيث سوف نترك بين كل قيمة وأخرى بفاصلة واحدة

مثال/

```
p { padding:5px 3px 8px 3px; }
```



### • margin:

تستخدم هذه الخاصية لتحديد المسافة بين حدود الفقرة وما يحيط بها من الخارج ( من فقرات أخرى أو حدود صفحة ) وهي تأخذ قيمة رقمية , مثال/

```
p { margin:20px; }
```

لكن في هذه الحالة سوف تضيف المسافة الفارغة من جميع الاتجاهات لكن إذا أردنا ترك مسافة من جهة واحدة فقط سوف نكتب الاتجاه بعد الخاصية هكذا :- ( **margin-right:** )  
مثال/

```
p { margin-top:20px;  
margin-left:8px; }
```

وكذلك يمكننا أن نضيف مسافة لكل جهة بشكل مختلف عن طريق إضافة أربع قيم ( تمثل الاتجاهات الأربعة ) إلى الخاصية **margin:** حيث إن القيمة الأولى تمثل الجهة العلوية والثانية تمثل اليمين والثالثة تمثل الأسفل والرابعة تمثل اليسار ( باتجاه عقارب الساعة ) حيث سوف نترك فاصلة واحدة بين كل قيمة وأخرى .  
مثال/

```
p { margin:5px 10px 0 4px; }
```

❖ إذا أردنا أن نحدد مكان فقرة في منتصف الصفحة يمكن أن نفعل ذلك من خلال إسناد القيمة ( **auto** ) و ( **0** ) إلى الخاصية ( **margin:** ) .

مثال/

```
div { margin:0 auto; }
```

### • border-radius:

وهي تستعمل لتقويس زاوية الإطار وتأخذ قيمة رقمية مثلاً **20px** , مع ملاحظة أن هذه الخاصية جديدة في **CSS3** .

مثال/

```
div { border-radius:20px; }
```

ويمكن أن نضع اتجاه لتتقوس زاوية معينة فقط , مثال/

```
div { border-top-right-radius:20px; }
```

### • border-image:url( );

هذه الخاصية جديدة في CSS3 وهي مسؤولة عن إنشاء إطار عنصر بالاعتماد على صورة صغيرة تمثل الإطار والصيغة العامة لهذه الخاصية هي :-

```
border-image:source slice width outside repeat;
```

Source تمثل مصدر الصورة بين قوسين مسبوقة ب url

outside & slice قيمة تمثل جزء الصورة الذي سيتم تكراره او تمده لصنع الإطار ( رقم )

width عرض الإطار ( رقم )

repeat نضع مكانها احد هذين القيمتين :- 1-repeat لتكرار جزء من الصورة .

2-stretch لتمدد جزء من الصورة .

ويجب أن نسبقها بالبادئة الخاصة بكل متصفح لتعرض في المتصفحات المختلفة .

مثال/

```
div { border-image:url(name.gif) 30 30 20 stretch ; }
```

### • box-shadow

هذه خاصية جديدة في CSS3 , وهذه الخاصية تستخدم لصنع ظل حول إطار معين ( فقرة او نص او قسم معين ) ويمكننا التحكم بحجم ولون الظل , وهذه الخاصية تأخذ ثلاث قيم رقمية وواحدة لونية حيث إن القيمة الأولى تحدد سمك الظل إذا أعطيناها قيمة سالبة سيكون الظل من جهة اليسار وإذا أعطيناها قيمة موجبة سيكون الظل من جهة اليمين , القيمة الثانية تحدد سمك الظل وإذا أعطيناها قيمة سالبة سيكون الظل في الأعلى أما إذا أعطيناها قيمة موجبة سيكون الظل من الأسفل , القيمة الثالثة تمثل شدة وضوح الظل ولا يمكن أن تأخذ قيمة سالبة وأما القيمة الرابعة فهي تحدد لون الظل .

مثال/

```
p { box-shadow:10px 10px 5px #888; }
```

❖ ويمكن أن نضيف أكثر من ظل لنفس النص وذلك بكتابة القيم الأربعة أسفل قيم خاصية الظل الأولى مع استبدال الفارزة المنقوطة في نهاية الخاصية الأولى إلى فارزة فقط .

مثال/

**box-shadow:7px 5px 3px #333,**

**10px 10px 3px #fff;**

وبنفس الطريقة يمكن إضافة ظل ثالث أو رابع أو أكثر كما نشاء .

## الفصل الثالث

### ( مظهر القوائم والخلفيات )

أولاً :- خصائص الخلفيات

• background-color:

وهي تستخدم لتحديد لون الخلفية , وتأخذ قيمة لونية مثلاً #333 .

مثال/

```
h1 { background-color:red; }
```

• background-image:url( );

تستخدم لتجعل الخلفية عبارة عن صورة حيث نكتب بين قوسيه اسم وامتداد الصورة وان كانت في مجلد آخر نضيف اسم المجلد قبل الصورة .

مثال/

```
p { background-image:url("الامتداد.اسم الصورة"); }
```

• background-repeat:

وهي تستعمل للتحكم بتكرار الصورة في الخلفية , وتأخذ احد هذه القيم :-

1. repeat-x لتكرار الصورة بشكل أفقي .
2. repeat-y لتكرار الصورة بشكل عمودي .
3. repeat لتكرار الصورة عمودياً وأفقياً .
4. no-repeat لعدم تكرار الصورة .

مثال/

```
div { background-repeat:no-repeat; }
```

### ● background-position:

وهي تستخدم لتحديد موقع الصورة في الخلفية إن كانت لا تملأ الشاشة , وتكون قيمتها عبارة عن اتجاه :-

- |           |                                 |
|-----------|---------------------------------|
| 1. left   | لمحاذاة الصورة إلى جهة اليسار . |
| 2. right  | لمحاذاة الصورة إلى جهة اليمين . |
| 3. top    | لمحاذاة الصورة إلى الأعلى .     |
| 4. bottom | لمحاذاة الصورة إلى الأسفل .     |
| 5. center | لمحاذاة الصورة إلى المنتصف .    |

ويمكننا أن نستخدم اتجاهين لتحديد موقع الصورة بالضبط مثلاً :-

top left او center center او أي اتجاهين آخرين .

مثال/

```
p { background-position:bottom right; }
```

\* ويمكن اعطائها قيم رقمية بالبيكسل حيث سيكون رقمان الأول يمثل البعد الأفقي والثاني يمثل البعد عمودي الذي سيبدأ منه العرض ( ظهور الصورة ) ويمكن أن تكون الأرقام بالسالب .

مثال/

```
p { background-position:bottom: 10px 50px; }
```

ويمكن ان نضيف قيمة لجهة واحدة فقط background-position-x تمثل البعد الأفقي background-position-y تمثل البعد العمودي .

مثال/

```
background-position-x:30px;
background-position-y:-44px;
```

### ● background:

تستخدم هذه الخاصية لعمل اختصار بدلا من كتابة مجموعة من الخواص للخلفية نجعلها كلها في هذه الخاصية حيث سنعطيها عدة قيم القيمة الأولى هي قيمة الخاصية **background-color:** والقيمة الثانية هي قيمة الخاصية **background-image:** والقيمة الثالثة هي قيمة الخاصية **background-repeat** والقيمة الرابعة هي قيمة الخاصية **background-position:** , مع ملاحظة أن هذه القيم تكتب بالترتيب وأي قيمة لا نحتاجها يمكن أن لا نكتبها .

مثال/

```
div { background:#333 url('image.gif') repeat-x top left; }
```

\* وكذلك يمكن اضافة بعض القيم الأخرى مثلا transparent وهي تستخدم لتجعل الخلفية شفافة او يمكن ان نضيف القيمة fixed لتجعل الخلفية ثابتة .

مثال/

```
div { background: transparent url('image.gif') repeat-x fixed; }
```

### ● background-size:

هذه الخاصية تسمح بتحديد أبعاد الصورة في الخلفية وهي خاصية جديدة في **CSS3** ويجب أن نسبقها بالبادئة الخاصة بكل متصفح لتعرض في المتصفحات المختلفة , وهي تأخذ قيم رقمية بالبيكسل حيث الرقم الأول يشير إلى العرض والثاني يشير إلى ارتفاع الصورة .

مثال/

```
body { background-size:80px 60px; }
```

### ● background-origin:

هذه الخاصية جديدة في **CSS3** ويجب أن نسبقها بالبادئة الخاصة بكل متصفح لتعرض في المتصفحات المختلفة , وهذه الخاصية تحدد احد أجزاء نموذج الصندوق ليتم تطبيق الخلفية على أساسه , وهي تأخذ احد هذه القيم :-

- 1- **context-box** لتطبيق الخلفية على المستطيل المحدد بالمحتوى .
- 2- **padding-box** لتطبيق الخلفية على المستطيل المحدد بالمحتوى وبالإزاحة الداخلية معاً .
- 3- **border-box** لتطبيق الخلفية على المستطيل المحدد بالمحتوى وبالإزاحة الداخلية وبالحدود معاً .

### ● إضافة خلفيتان

من التحسينات الجديدة التي أضيفت إلى خاصية الخلفية في **CSS3** هو السماح بإضافة أكثر من صورة في نفس الوقت كخلفية لنفس العنصر وذلك عبر الفصل بين كل خلفية أخرى بفاصلة .

مثال/

```
body { background-image:url(name.gif) , url(name.gif); }
```

### ● linear-gradient( )

هذه الخاصية جديدة في **CSS3** ويجب أن نسبقها بالبادئة الخاصة بكل متصفح لتعرض في المتصفحات المختلفة , وهذه الخاصية مسؤولة عن عمل تدرج لوني بشكل خطي , حيث أنها تكتب كقيمة للخاصية ( **background:** ) لأنها تمثل لون الخلفية , وسوف نضع في داخل قوسها مجموعة قيم حيث إن القيمة الأولى تمثل الجهة التي ستبدأ منها الألوان ( **top** ) ( **bottom** , **right** , **left** ) ويمكن أن لا نكتب هذه القيمة لتبدأ بشكل افتراضي من الأعلى , القيمة الثانية تمثل اللون الأول ويمكن أن نكتب بجانبها نسبة مئوية أو رقم بالبيكسل يمثل مساحة تمدد هذا اللون والقيمة الثالثة تمثل اللون الثاني وأيضا يمكن أن نكتب بجانبها نسبة مئوية أو رقم بالبيكسل يمثل مساحة تمدد هذا اللون وهكذا يمكن إضافة قيم أخرى كلها تمثل ألوان إضافية مع ملاحظة أنه يفصل بين قيمة وأخرى بفارزة .

مثال/

```
div { background:-webkit-linear-gradient( top , black 50% , #f33 , #ccc 20px ); }
```

## • ( ) radial-gradient

هذه الخاصية جديدة في CSS3 ويجب أن نسبقها بالبائدة الخاصة بكل متصفح لتعرض في المتصفحات المختلفة , وهذه الخاصية مسؤولة عن عمل تدرج لوني بأشكال مختلفة وهي تكتب كقيمة للخاصية ( background ) لأنها تمثل لون الخلفية , ونضع بين قوسيه مجموعة من القيم حيث إن القيمة الأولى تمثل موقع هذا التدرج وتكون بشكل نسبتين مؤيتين حيث إن النسبة الأولى تمثل الإحداثي السيني والنسبة الثانية تمثل الإحداثي الصادي ويمكن أن لا نضع هذه النسب لتأخذ بشكل افتراضي النسبتين ( 50% 50% ) والقيمة الثانية تمثل شكل التدرج اللوني مثلاً ( circle , closest-corner , farthest-corner , ellipse ) والقيمة الثالثة تمثل اللون الأول والقيمة الرابعة تمثل اللون الثاني ويمكن أن نضيف قيم أخرى تمثل الألوان , مع ملاحظة أنه يفصل بين كل قيمة وأخرى بفارزة .

مثال/

```
div { background:-webkit-radial-gradient( 20% 50% , farthest-corner , green , white ); }
```



## ثانياً :- خصائص مظهر القوائم

### • list-style-type

تستخدم لتحديد نوع الرمز ( الترقيم ) أمام كل عنصر من عناصر القائمة .

قيم القائمة الغير مرتبة :-

1. circle لتكون بشكل دائري .
2. square لتكون بشكل مربع .
3. disc لتكون بشكل قرص فارغ .
4. none لعدم وضع أي شيء .

قيم القائمة المرتبة :-

1. upper-roman
2. upper-alpha
3. upper-latin
4. lower-roman
5. lower-alpha
6. lower-latin
7. decimal

### • list-style-image:url( );

وهي تستخدم لوضع صورة خاصة أمام كل عنصر من عناصر القائمة كرمز له , حيث نضع بين قوسيه اسم وامتداد الصورة واسم المجلد إذا كانت في مجلد آخر .

مثال/

```
ul { list-style-image:url("الامتداد.اسم الصورة"); }
```

**list-style-position:** •

هذه الخاصية تستخدم لتحريك القائمة إلى الداخل ( ترك مسافة عن البداية ) وهي تأخذ قيمتين هما :-

1. **inside** لتحريك القائمة إلى داخل الصفحة .
2. **outside** وهي تمثل القيمة الافتراضية .

مثال/

**list-style-position:inside;**

**list-style:** •

تستخدم هذه الخاصية كاختصار لبعض خصائص القائمة حيث نسند لها عدة قيم , القيمة الأولى تمثل قيمة الخاصية **list-style-position** والقيمة الثانية تمثل قيمة الخاصية **list-style-image** , مع ملاحظة الترتيب في القيم .

مثال/

**list-style:inside url('image.gif');**

## الفصل الرابع ( خصائص متنوعة )

### • width

وهي تستخدم لتحديد عرض فقرة او اطار معين وهي تأخذ قيمة نسبة مئوية او رقم بالبيكسل .

مثال/

```
div { width:400px; }
```

❖ يمكن أن نعطيها القيمة ( auto ) وفي هذه الحالة سيكون العرض بنفس عرض الشاشة مهما كبرت او صغرت ولا يظهر شريط التمرير في حالة تصغير الشاشة .

### • min-width

وهي تستخدم لتحديد أقل عرض ممكن للإطار وهي تأخذ رقم بالبيكسل او نسبة مئوية .

### • max-width

وهي تستخدم لتحديد أكبر عرض ممكن للإطار وهي تأخذ رقم بالبيكسل او نسبة مئوية .

### • height

وهي تستخدم لتحديد طول فقرة او اطار معين وهي تأخذ قيمة نسبة مئوية او رقم بالبيكسل .

مثال/

```
div { height:500px; }
```

- **min-height**

وهي تستخدم لتحديد أقل ارتفاع ممكن للإطار وهي تأخذ رقم بالبيكسل او نسبة مئوية .

- **max-height**

وهي تستخدم لتحديد أكبر ارتفاع ممكن للإطار وهي تأخذ رقم بالبيكسل او نسبة مئوية .

- **opacity**

من خلال هذه الخاصية يمكننا التحكم بدرجة الشفافية عبر اسناد قيمة من 0.0 الى 1.0 حيث ان القيمة 0.0 تعني عدم الظهور والقيمة 1.0 تعني الظهور الكامل والقيم الاخرى تكون ما بين هذين القيمتين .

- **resize**

من خلال هذه الخاصية يمكن التحكم بالحجم من حيث التكبير والتصغير ويمكن أن نعطيها القيمة none لعدم السماح للمستخدم بتغيير الحجم

مثال/

```
textarea { resize:none; }
```

- **float**

تسمى خاصية التعويم وهي تستخدم للحفاظ على تموضع متناسق للعناصر حيث يتم تعويم جميع العناصر إلى جهة واحدة بالنسبة لعنصر محدد والتعويم يكون إما إلى اليسار **left** او إلى اليمين **right** بمعنى انه من خلال هذه الخاصية يمكن تحديد اتجاه ( مكان ) عنصر معين في الصفحة يمين او يسار الصفحة , ويمكن أن تأخذ القيمة ( none ) لإلغاء التعويم .

مثال/

```
ul { float:right; }
```

❖ عند تعويم عنصر ما فإن العناصر التي تليه سوف تلتف حوله ( تعوم ) بما يتلاءم مع تعويمه أما العناصر التي قبله لا تتأثر بالتعويم . أي مثلاً يتم تعويم صورة ويوضع بجوارها نص وذلك حتى يكون النص مرتب بجوارها .

#### • clear

تستخدم هذه الخاصية لمنع تعويم عناصر بجوار عنصر معين , وهي تأخذ القيمة **left** لمنع التعويم من اليسار والقيمة **right** لمنع التعويم من اليمين والقيمة **both** لمنع التعويم من الطرفين

#### • position:relative;

تستخدم هذه الخاصية على إزاحة عنصر معين إلى جهة معينة بمقدار معين بالنسبة للعنصر الذي فوقه أو الذي بجواره وليس بالنسبة إلى الصفحة , مع ملاحظة إنه عندما يتم إزاحة العنصر إلى مكان جديد فإنه يبقى شاغر حيز مكانه القديم وكأنه لا يزال موجود فيه وذلك بخلاف الخاصية ( **position-absolute** ) , وهي تأخذ احد هذه القيم :-

1. **top**: رقم px

2. **bottom**: رقم px

3. **left**: رقم px

4. **right**: رقم px

❖ ويمكن أن تأخذ رقم مسبوق بإشارة سالب ليعمل بشكل عكسي .

مثال/ يمكن أن يكتب هذا المثال بطريقتين :- الطريقة الأولى :-

**position:relative; bottom:-100px; right:70px;**

الطريقة الثانية :-

**position:relative;**

**bottom:-100px;**

**right:70px;**

### • position:absolute;

تستخدم هذه الخاصية لإزاحة عنصر معين إلى جهة معينة بمقدار معين بالنسبة إلى الصفحة وتستخدم في الغالب مع الصور ويمكن استخدامها مع غير الصور , مع ملاحظة إنه عندما يتم إزاحة عنصر معين إلى مكان جديد فإنه لا يبقى يشغل حيز مكانه القديم بل يشغل مكانه الجديد فقط بخلاف الخاصية ( **position-relative** ) , وهي تأخذ احد القيم التالية :-

1. top: رقم px

2. bottom: رقم px

3. left: رقم px

4. right: رقم px

❖ ويمكن أن تأخذ رقم مسبق بإشارة سالب ليعمل بشكل عكسي .

مثال/ يمكن أن يكتب المثال بطريقتين :- الطريقة الأولى :-

**position:absolute; top:150px; left:100px;**

الطريقة الثانية :-

**position:absolute;**

**top:150px;**

**left:100px;**

### • position:

تستخدم هذه الخاصية لجعل عنصر معين ثابت في الصفحة ولا يتغير مكانه عند النزول أو الصعود في داخل الصفحة إذا كانت الصفحة كبيرة , وهي تأخذ القيمة **fixed** .

مثال/

**div { position:fixed; }**

## • z-index:

تعمل هذه الخاصية فقط مع الخاصية ( **position:** ) حيث إنها سوف تحدد أي من العناصر سيظهر فوق الآخر ( إذا كانت العناصر متداخلة ) حيث سوف نعطي للعنصر الذي يظهر في الأسفل الخاصية **z-index:** تحمل القيمة ( 1 ) وللعنصر الذي سيكون فوقه القيمة ( 2 ) والذي فوقه القيمة ( 3 ) وهكذا , مع ملاحظة أن الكلام السابق ينطبق إذا كان العنصران يحملان الخاصية ( **position:** ) أما إذا كان عنصر واحد فقط هو الذي يحمل الخاصية ( **position:** ) وأردناه أن يظهر في الأعلى فوق بقية العناصر نعطيه الخاصية **z-index:** حاملة القيمة ( 1 ) ( مع ملاحظة أنه يظهر في الأعلى فوق بقية العناصر بشكل افتراضي ) أما إذا أردناه أن يظهر في الأسفل تحت بقية العناصر نعطيه الخاصية **z-index:** حاملة القيمة ( -1 ) .

❖ عندما أتحدث هنا عن الخاصية ( **position** ) أقصد بها كل خصائص ( **position** )  
مثلا ( **position-relative:** ) او ( **position-absolute:** ) أو ( **position:** ) .

مثال/

```
p { position:absolute;
top:20px;
z-index:-1; }
```

## • overflow:

وهي تستخدم لإظهار مقبض الحركة وتأخذ الخاصية **scroll** , وتأخذ الخاصية **hidden** لأخفاء مقبض الحركة .

## • cursor

وهي تستخدم للتحكم بشكل سهم الفأرة وهي تأخذ عدة قيم منها .

- 1- pointer لتظهر شكل اليد للاختيار .
- 2- none ليختفي السهم .
- 3- auto ليظهر السهم بالشكل الطبيعي له ( أي بحسب المكان الذي سيقف فيه ) .
- 4- default ليبقى بشكل سهم دائماً .
- 5- progress ليظهر وكأنه يحمل شيء ما .
- 6- url وهي تستخدم لوضع صورة لتظهر مكان السهم .

مثال/

**Cursor:pointer;**

## ● إخفاء العناصر

من الممكن إخفاء عناصر HTML باستخدام CSS بطريقتين :-

1. إسناد القيمة **none** إلى الخاصية **display** .
2. إسناد القيمة **hidden** إلى الخاصية **visibility** . ( ملاحظة// اذا اسندنا القيمة **visible** الى هذه الخاصية فإن العنصر المحدد سيظهر )

والاختلاف بينهما هو أن الطريقة الثانية تخفي العنصر مع الحفاظ على الحيز الذي يشغله فارغاً أما الطريقة الأولى تخفي العنصر مع عدم الحفاظ على الحيز الذي يشغله مما يؤثر على تموضع العناصر .

## ● أنواع عرض العناصر

يتم عرض العناصر في HTML بطريقتين :-

1. بشكل كتلة **block** :- حيث يتم النزول سطرًا قبل العنصر و سطر بعد العنصر هذه الحالة نراها مع عناصر مثل **<h1>** و **<p>** و **<div>** وفي هذه الحالة ستكون كلها كجزء واحد مترابطة فيما بينها .
2. بشكل سطري **inline** :- حيث يتم وضع العنصر ضمن نفس السطر دون الحاجة للنزول سطرًا وهذه الحالة نراها مع عناصر مثل **<a>** و **<span>** , حيث أن في هذه الحالة تكون مقسمة إلى أجزاء .

تسمح CSS بتغيير نوع العرض للعناصر من **block** إلى **inline** وبالعكس عن طريق إسناد إحدى القيمتين **block** و **inline** إلى الخاصية **display** ويمكن أن نعرض العناصر بخط واحد ( سطر أفقي ) وفي نفس الوقت تكون جزء واحد , هنا سنستخدم هذه الخاصية **inline-block** .



### ● الأعمدة المتعددة

هذه الخاصية جديدة في CSS3 ويجب أن نسبق الخاصية بالبادئة الخاصة بكل مستعرض ليتم عرضها على المستعرضات المختلفة , ومن خلال هذه الخاصية يمكن عرض محتوى العناصر على شكل أعمدة متعددة ( بشكل الجرائد ) وذلك بطريقتين :-

1. عدد أعمدة ثابت وعرض عمود متغير :- ويتم ذلك من خلال إسناد عدد الأعمدة إلى الخاصية ( **column-count** ) التي تقسم المحتوى إلى العدد المذكور من الأعمدة ويتغير عرض الأعمدة مع تغير حجم نافذة المستعرض , والخاصية ( **column-gap** ) تستخدم لتحديد المسافة بين العمودين , والخاصية ( **column-rule** ) تمكننا من وضع فواصل ذات مظهر ولون محدد بين الأعمدة .

مثال/

```
-webkit-column-count:2;
-webkit-column-gap:20px;
-webkit-column-rule:4px dotted black;
```

2. عرض عمود ثابت وعدد أعمدة متغير :- ويتم ذلك من خلال إضافة عرض العمود الواحد إلى الخاصية ( **column-width** ) وفي هذه الطريقة تحافظ الأعمدة على عرض ثابت لكل منها ويتغير عددها مع تغير حجم نافذة المستعرض , والخاصية ( **column-gap** ) تستخدم لتحديد المسافة بين العمودين , والخاصية ( **column-rule** ) تمكننا من وضع فواصل ذات مظهر ولون محدد بين الأعمدة .

مثال/

```
-ms-column-width:200px;
-ms-column-gap:30px;
-ms-column-rule:4px dotted black;
```

## الفصل الخامس

### ( التحويلات الهندسية وخصائص الحركة )

❖ هذا الموضوع ( الفصل الخامس ) جديد في CSS3 ويجب أن نسبق خواصه بالبادئة الخاصة بكل متصفح لتعرض في المتصفحات المختلفة .

أولاً :- التحويلات الهندسية ذات البعدين 2D

#### • الخاصية transform

وهي تأخذ احد هذه التوابع الموضحة في الجدول

التابع	الشرح
translate(x,y)	ينقل العنصر إلى الإحداثيات الممررة إليه
translateX(n)	ينقل العنصر على المحور X مسافة n
translateY(n)	ينقل العنصر على المحور Y مسافة n
scale(x,y)	يغير بعدي العنصر إلى البعدين الممررين إليه
scaleX(n)	يغير بعد العنصر X إلى القيمة n
scaleY(n)	يغير بعد العنصر Y إلى القيمة n
rotate(α)	يقوم بتدوير العنصر بزاوية α
skew(Xα , Yα)	يقوم بإجراء تحويل الانحراف وفق الزاويتين Xα و Yα
skewX(α)	يقم بإجراء تحويل الانحراف على المحور X وفق الزاوية α
skewY(α)	يقم بإجراء تحويل الانحراف على المحور Y وفق الزاوية α

مثال/

```
div { -webkit-transform:skew(30deg , 20deg);
width:100px; height:75px;
background-color:red; }
```

**ملاحظة/** الحرف ( deg ) في المثال السابق التي تم تمريرها بعد الارقام إلى تابع التحويل تعني درجة ( degree ) وذلك تمييزاً عن الراديان والذي يرمز له بالأحرف ( rad ) .

### ثانياً :- التحويلات الهندسية ثلاثية الأبعاد 3D

#### ● الخاصية transform

تتم التحويلات ثلاثية الأبعاد بنفس الخاصية السابقة ( transform ) لكن تأخذ هذه التوابع المختلفة قليلاً

التابع	الشرح
translate3d(x,y,z)	ينقل العنصر إلى الإحداثيات الممررة إليه
translateX(n)	ينقل العنصر على المحور X مسافة n
translateY(n)	ينقل العنصر على المحور Y مسافة n
translateZ(n)	ينقل العنصر على المحور Z مسافة n
scale3d(x,y,z)	يغير أبعاد العنصر إلى الأبعاد الممررة إليه
scaleX(n)	يغير بعد العنصر X إلى القيمة n
scaleY(n)	يغير بعد العنصر Y إلى القيمة n
scaleZ(n)	يغير بعد العنصر Z إلى القيمة n
rotate3d(x,y,z,α)	يقوم بتدوير العنصر على المحاور الثلاثة بزاوية α
rotateX(α)	يقوم بتدوير العنصر على المحور X بزاوية α
rotateY(α)	يقوم بتدوير العنصر على المحور Y بزاوية α
rotateZ(α)	يقوم بتدوير العنصر على المحور Z بزاوية α
perspective	يحدد منظور الرؤية

## ثالثاً :- الانتقال المتحرك

### • transition

يمكن إنشاء تأثير حركة على عناصر المستند تحت تأثير زمني معين والذي يتم تحقيقه عبر هذه الخاصية ( **transition** ) والصيغة العامة لها هي :-

**transition:property1 duration1 , property2 duration2 , ..... etc ;**

حيث أن **property** تمثل اسم الخاصية التي ستتغير و **duration** تعبر عن الزمن , ويمكن تغيير خاصية واحدة فقط أو عدة خواص معاً أو كل الخواص وسنكتب **all** وبعدها الزمن .

مثال/

```
<html> <head>
<style type="text/css" >
div { width:100px;
      height:100px;
      background:red;
      -webkit-transition:width 2s , height 2s; }
div:hover { width:200px;
            height:200px;
            -webkit-transform:rotate(180deg); }
</style> </head> <body>
<div> Hello world </div>
</body> </html>
```

حيث إن **s** تشير إلى الثانية **second**

## الفصل السادس ( الوسوم الزائفة )

### • a:hover

من خلال هذه الخاصية ( الوسوم الزائفة ) يمكن أن نضيف تأثير على العنصر عند ملامسة مؤشر الفأرة ذلك العنصر ويمكن أن نكتب اسم أي وسم وليس فقط الوسوم `<a>` ( كتبناه كمثال ويمكن استخدام غيره من الأوسمة ) .

مثال/

```
div { background-color:red; }
```

```
div:hover { background-color:#333;
```

```
font-size:20px; }
```

❖ يمكن أن نكتب اسم ( id ) أو ( class ) الخاص بالوسم هكذا :-

مثلاً ( #name:hover { } )

### • a:active

هذه الخاصية ( الوسوم الزائفة ) يستخدم مع الوسوم `<a>` ( الروابط ) فقط , ومن خلاله نستطيع أن نضيف بعض التأثيرات على الرابط الذي يتم النقر عليه في هذه اللحظة .

مثال/

```
a:active { color:red; }
```

- **a:visited**

هذه الخاصية (الوسم الزائفة) يستخدم مع الوسم **<a>** (الروابط) فقط, ومن خلاله نستطيع أن نضيف بعض التأثيرات على الرابط الذي تم زيارته.

مثال/

```
a:visited { color:yellow; }
```

- **a:link**

هذه الخاصية (الوسم الزائفة) يستخدم مع الوسم **<a>** (الروابط) فقط, ومن خلاله نستطيع أن نتحكم بالروابط أو الوصلة التشعبية.

مثال/

```
a:link { color:#f03; }
```

- **input:focus**

من خلال هذه الخاصية (الوسم الزائفة) يمكن تطبيق التأثيرات على حقول الإدخال **input**.

مثال/

```
input:focus { border:3px solid #333;
background-color:#ff0; }
```

- **first-letter**

من خلال هذه الخاصية (الوسم الزائفة) نستطيع أن نطبق تأثير معين على الحرف الأول من كل فقرة (النص), حيث يجب أن نسبق هذه الخاصية باسم الوسم ونضع التأثير بين القوسين.

مثال/

```
p:first-letter { color:red;
font-size:10pt; }
```

### • first-line

من خلال هذه الخاصية (الوسم الزائف) نستطيع التحكم وتطبيق تأثير معين على أول سطر من الفقرة (النص)، حيث يجب أن نسبق هذه الخاصية باسم الوسم ونضع التأثير بين القوسين  
مثال/

```
p:first-line { color:green;
              font-size:20px; }
```

### • before

من خلال هذه الخاصية (الوسم الزائف) يمكننا إدراج محتوى قبل العنصر، حيث يجب أن نسبق هذه الخاصية باسم الوسم ونضع التأثير بين القوسين.  
مثال/ لإضافة صورة قبل كل فقرة

```
p:before { content:url('image.gif'); }
```

❖ ويمكن أن نضع قبلها اسم class أو id الوسم بدلاً من اسم الوسم، مثال /

```
.tt:before { content:url('image.gif'); }
```

### • after

من خلال هذه الخاصية (الوسم الزائف) يمكننا إدراج محتوى بعد العنصر، حيث يجب أن نسبق هذه الخاصية باسم الوسم ونضع التأثير بين القوسين.  
مثال/ إضافة علامة تعجب بعد كل فقرة

```
p:after { content:" !"; }
```

❖ ويمكن أن نضع قبلها اسم class أو id الوسم بدلاً من اسم الوسم، مثال /

```
.tt:after { content:" !"; }
```

## • @media

من خلال هذه الخاصية ( الوسوم الزائفة ) يمكن تخصيص شكل ومظهر الصفحة بناءً على ما يقوم بعرضها من شاشة حاسوب او آلة طباعة او غيرها .

مثال/

```
<html>
<head><style type="text/css" >
@media screen {
p { text-align:justify;
direction:rtl; }
body { background-color:green; }
}
@media print {
p { text-align:justify; } }
</style></head><body>
<p> My Text </p>
</body>
</html>
```

يمكن ان نضع بعد @media احد القيم التالية :-

الشرح	القيمة
جميع اجهزة العرض	all
للاجهزة الناطقة	aural
الاجهزة المعتمدة على اللمس	braille
طابعات الاجهزة المعتمدة على اللمس	embossed
للاجهزة الكفية	handheld
كل الطابعات	print
لاجهزة الإسقاط ( لراشق مثلاً )	projection
لشاشات الحاسوب	screen
لكل الاجهزة التي تستخدم شاشات ذات عرض محرف ثابت ( كالمبرقة الكاتبة )	ttv
لكل الاجهزة الشبيهه بالتلفاز	tv



## • ( ) @media screen and

من خلال هذه الخاصية يمكننا التحكم بالموقع بحسب حجم الشاشة المفتوحة حيث سنضع بداخل هذين القوسين أكبر أو أصغر مقدار لحجم الشاشة قبل تنفيذ هذه الخاصية مثلاً :-

( max-width:400px ) او ( min-width:400px ) وبعدها نضع قوسين معقوفين ونكتب بداخلهم الخواص وما سيجري عليها من تغييرات في حال تغير حجم الشاشة إلى أصغر أو أكبر من هذا المقدار بحسب القيمة المعطاة .

مثال/

```
p { font-size:40pt; }
```

```
@media screen and ( max-width:300px ) {
```

```
  p { font-size:10pt;
```

```
    color:red; } }
```

❖ يمكن أن نكتب هذه الخاصية بكل ما تحتويه من خصائص وتغييرات داخلها في صفحة CSS جديدة ( ثانية ) مع ملاحظة عدم كتابة اسم هذه الخاصية في البداية بل نكتب الخصائص التي في داخلها مباشرة ( كأى صفحة CSS عادية ) لأننا سنكتب اسم هذه الخاصية عند استدعائها من داخل صفحة html , وسوف نستدعيها في صفحة html من خلال وسم خاص يكتب داخل الوسوم <head> وهذا الوسوم هو

```
<link rel="stylesheet" type="text/css" media="screen and ( max-width:300px)"
```

```
href="اسم الصفحة.css" />
```

هذه القيم تتغير بحسب القيمة التي

نرغب في إعطائها للخاصية

وفائدة هذه الخاصية في السيطرة على شكل الموقع وعرضه بشكل مناسب ليتوافق مع حجم الشاشة في الجهاز العارض كأن يكون هاتف جوال او جهاز لوحي او شاشة حاسوب .

❖ إذا أردنا أن يطبق التأثير فقط إذا كانت الشاشة تحمل هذه الدقة وليس تصغير أو تكبير الشاشة , مثلا نعطي دقة قليلة بحيث لا يمكن أن يحمل هذه الدقة جهاز حاسوب بل تتوفر فقط على الهاتف الجوال في هذه الحالة يجب أن نضيف ( **device** ) إلى الخاصية لتكون بهذا الشكل :-

مثال/ يطبق التأثير فقط على الجوال

**@media screen and ( max-device-width:400px ) { ..... }**

وكذلك إذا وضعناها في ملف CSS مستقل واستدعيها من داخل ملف html سنضيف ( **device** ) إلى وسم الاستدعاء .

مثال/

**<link rel="stylesheet" type="text/css" href="اسم الملف.css" media="screen and (max-device-width:400px)" />**

### • @font-face

هذه الخاصية جديدة في **CSS3** , من خلال هذه الخاصية ( الوسوم الزائفة ) يمكننا إضافة خطوط جديدة حتى وإن لم تكن موجودة في متصفح زائر الموقع حيث في السابق كان يجب الالتزام بالخطوط الشائعة لكن هذه الخاصية حلت المشكلة حيث يتم وضع ملف الخط ضمن ملفات الموقع ويتم إسناد مساره إلى الخاصية **src** لقاعدة **@font-face** وتستخدم كما يلي :-

أولاً :- يتم تعريف اسم عائلة جديدة باستخدام القاعدة **@font-face** .

ثانياً :- يسند اسم العائلة الذي انشأ سابقاً إلى الخاصية **font-family** بشكل تقليدي .

ملاحظة/ المستعرض Explorer يدعم الخطوط ذات النوع **.eot** فقط بينما تدعم باقي المستعرضات النوعين **.ttf** و **.otf**.

مثال/

```

<style type="text/css"> يمكن أن نضع أي اسم
@font-face { font-family:myfont;
src:url("name.ttf") , url("name.eot") format("opentype"); /* IE */
} نفس الاسم السابق
div { font-family:myfont; }
</style>

```

**ملاحظة/** يمكن تغيير صيغ الخط ( نوع الخط ) من خلال الذهاب إلى الموقع التالي :-  
[www.freefontconverter.com](http://www.freefontconverter.com) الذي يتيح لنا إمكانية التحويل بين صيغ الخطوط المختلفة

### • @keyframes

هذه الخاصية جديدة في **CSS3** ويجب أن نكتب البادئات الخاصة بكل متصفح لكي تعرض في المتصفحات المختلفة , وهذه الخاصية ( الوسوم الزائفة ) مسئول بالتعاون مع الخاصية (**animation**) لإنشاء حركات , ويتم إنشاء الحركة بخطوتين :-

1- يتم تعريف اسم خاص للحركة في القاعدة **@keyframes** وتحديد ماهيتها عبر تحديد الوضع الابتدائي **from** والوضع النهائي **to** وعند تطبيق الحركة يتم الانتقال من الوضع الابتدائي إلى الوضع النهائي للعنصر خلال فترة زمنية مما يجعلنا نشعر بالحركة.

2- يتم إسناد الحركة وزمن تنفيذها إلى الخاصية **animation** الخاصة بالعنصر .

مثال/

```

<html> <head>
<style type="text/css" >
div { width:100px;
      height:100px;
      background:red;
      -webkit-animation:myAnimation 5s; }
@-webkit-keyframes myAnimation {
  from { background:red; }
  to { background:yello; }
</style> </head> <body>
<div> </div>
</body> </html>

```

حيث إن **s** تشير إلى الثانية **second**

❖ يمكن تخصيص شكل العنصر في كل لحظة من لحظات الحركة عبر إنشاء قاعدة CSS خاصة بهذه اللحظة ضمن قاعدة **@keyframes** .

مثال/

```

@-webkit-keyframes myAnimation {
  0% { background:red; }
  15% { background:green; width:150px; color:white; }
  50% { height:350px; }
  100% { background:yello; } }

```

حيث إن ( **0%** ) تعني لحظة بداية الحركة و ( **100%** ) تعني لحظة نهاية الحركة والقيم الأخرى تعبر عن فترات أثناء حدوث الحركة .

❖ تحدث هذه التغيرات أثناء الحركة ويعود العنصر إلى شكله الطبيعي عند انتهاء الحركة .

## ( ملاحظات عامة )

**أولاً :-** في بعض الخصائص الجديدة في CSS3 يجب أن نستخدم البادئة الخاصة بكل متصفح لكي يتم عرض هذه الخاصية بشكل صحيح في المتصفحات المختلفة ولكي تظهر الخاصية في كل المتصفحات يجب أن نكتبها عدة مرات مع تغيير بادئة المتصفح , ويجب أن نكتبها مرة بدون أي بادئة لتعرض في بعض المتصفحات الأخرى , المتصفح فايرفوكس يستخدم البادئة (-moz-) والمتصفح أوبرا يستخدم البادئة (-o-) والمتصفح جوجل كروم وسفاري يستخدمان البادئة (-webkit-) والمتصفح اكسلورر يستخدم البادئة (-ms-).

**ثانياً :-** يتم كتابة التعليق في CSS بواسطة النجمة والشطرة المائلة .

مثال/

*/\* التعليق \*/*

**ثالثاً :-** يمكن أن نستخدم في CSS مجموعة من وحدات القياس وهي موضحة في الجدول بالأسفل :-

الوحدة	الشرح
px	بيكسل
%	بالنسبة المئوية
em	كل ( 1 em ) يساوي ( 16 px )
in	إنش
cm	سنتي متر
mm	ملي متر
pt	نقطة ( النقطة = 1/72 إنش )
pc	بيكا ( البيكا = 12 نقطة = 1/6 إنش )

### رابعاً :- id و class

إذا أردنا أن نصمم تأثير خاص في CSS ونضيف هذا التأثير على بعض الوسوم فيجب أن نعطي هذا الوسم الذي نريد تطبيق التأثير عليه اسم معين في صفحة html باستخدام ( id ) او ( class ) ومن ثم نستدعيه في صفحة CSS بذكر اسمه لكن نسبق اسمه بنقطة ( . ) إذا كنا نستخدم ( class ) او نسبقه بعلامة شبك (#) إذا كنا نستخدم ( id ) .

مثال/

```
<html> <head>
<style type="text/css" >
.myclass { color:red; }
#myid { color:yello; }
</style> </head> <body >
<p id="myid"> the text </p>
<p class="myclass"> the text </p>
</body> </html>
```

❖ الفرق بين ( id ) و ( class ) هو أننا إذا استخدمنا طريقة ( id ) (( أي وضعنا (#) في صفحة CSS وأعطينا اسم معين واستدعيناه باستخدام ( id ) بذكر اسمه )) في هذه الطريقة لا يمكن أن نذكر هذا الاسم باستخدام ( id ) في وسم آخر لتطبيق نفس الخواص عليه , أما في طريقة ( class ) (( أي وضعنا ( . ) في صفحة CSS وأعطينا اسم معين واستدعيناه باستخدام ( class ) بذكر اسمه )) في هذه الطريقة نستطيع أن نذكر هذا الاسم باستخدام ( class ) في وسم آخر لتطبيق نفس التأثيرات ( الخواص ) عليه .

**خامساً :- التداخل**

إذا أردنا تطبيق تنسيق معين على عدة أوسمة يمكننا أن نذكر أسماء هذه الأوسمة قبل التنسيق ونفصل بينها بفارزة وبعدها فاصلة واحدة ثم اسم الوسم التالي وهكذا , وهذا الكلام ينطبق حتى على أسماء ( class ) و ( id ) .

مثال/

```
h1, p, .MyClass { color:#333; }
```

❖ إذا كتبنا وسمين وفصلنا بينهم بفاصلة فقط فهذا التأثير يطبق إذا كان الوسم الثاني بداخل الوسم الأول وسيطبق التأثير على الوسم الثاني فقط .

مثال/

```
ul ul { color:red; }
```

وليس بالضرورة أن يكون الوسمين من نفس النوع , يمكن أن يكونا مختلفين .

❖ إذا طبقنا تأثير معين على الوسم <div> فإن هذا التأثير سيشمل كل الوسوم الأخرى الموجودة داخله وإذا أردنا تطبيق التأثير على وسم معين في داخل <div> يمكن ذكر الوسم **div** ثم نذكر اسم الوسم .

مثال/

```
div p {color:white; }
```

وإذا كنا نستخدم طريقة **class** او **id** لتطبيق التأثير فأننا سوف نذكر الاسم ثم فاصلة ثم اسم الوسم الذي نريد تطبيق التأثير عليه .

مثال/

```
.My-class p {color:red; }
```

**سادساً :-** ملاحظات حول القوائم

1. إذا كانت لدينا قائمة (`<ul>`) مكونه من عدة نقاط (`<li>`) وأردنا أن نعطي لكل هذه النقط في القائمة عرض أو ارتفاع معين ( نجعل كل نقطة في إطار ونحدد عرضها أو ارتفاعها كلها لتكون بشكل متساوي ) فيجب أن نضيف الخاصية (`display`) حاملة القيمة (`block`) وذلك لكي يتم اعتبار كل نقطة من هذه النقاط هي كتلة لوحدها وفي هذه الحالة سيعطي كل النقاط عرض واحد او ارتفاع واحد , ( نستخدم هذا الكلام إذا كنا نريد أن نكون قائمة هي عبارة عن أزرار للانتقال عبر صفحات الموقع من خلال وسم القائمة حيث سنجعل كل نقطة من نقاط القائمة هي عبارة عن وصلة تشعبية ( رابط ) ونحيطها بإطار لتكون مثل الزر )

مثال/

```
ul li a { display:block;
width:100px;
background-color:red; }
```

2. إذا كانت لدينا قائمة وأردنا أن نعرض النقاط الموجودة فيها بشكل أفقي ( الواحدة بجانب الأخرى ) وليس بشكل عمودي فيجب أن نضيف الخاصية (`float`) في وسم تعداد النقاط `<li>` ونسند لها القيمة (`left`) إذا كان اتجاه القائمة من اليسار إلى اليمين, ونسند لها القيمة (`right`) إذا كان اتجاه القائمة من اليمين إلى اليسار , مع ملاحظة أنه من الضروري أن نكون وسم `<div>` داخل وسم الـ `<div>` الرئيسي ( الوسم الذي يحتوي كامل القائمة ) ومن ثم نعطيه الخاصية `clear:both;` وذلك لكي نمنع التعويم من التأثير على باقي الوسوم .

أو يمكننا أن نجعل ترتيب النقاط في القائمة بشكل أفقي بطريقة أخرى وذلك بإضافة الخاصية (`direction`) ونسند لها القيمة (`block-inline`) , لكن يفضل استخدام الطريقة الأولى .

3. لإنشاء قائمة منبثقة عند الوقوف على زر في القائمة الرئيسية , أولاً ننشأ أزرار القائمة الرئيسية من خلال وسم القائمة والنقاط تكون عبارة عن روابط تشعبية والقائمة المنبثقة تكون عبارة عن قائمة أخرى ننشئها داخل النقطة التي نريد أن تظهر القائمة الفرعية عند الوقوف على هذه النقطة من القائمة الرئيسية حيث إننا سنخفي القائمة الفرعية من



خلال الخاصية ( **display** ) حاملة القيمة ( **none** ) , ثم بعد ذلك نستخدم الخاصية ( **display** ) حاملة القيمة ( **block** ) لإظهارها وذلك داخل الوسم الزائف ( **li:hover** ) لكي تظهر عند وقوف مؤشر الفأرة عليها .

```
#li_menu:hover #ul2 { display:block; }
```

حيث أن ( **#li\_menu** ) هو اسم الوسم <li> الموجود في القائمة الرئيسية الذي يحمل القائمة الثانية ( المنبثقة ) المسماة ( **#ul2** ) , حيث أن معنى الكود السابق هو عند وقوف مؤشر الفأرة على النقطة المسماة ( **#li\_menu** ) أجزى التأثير على القائمة المسماة ( **#ul2** ) .

**ملاحظة/** لكي نتمكن من التحكم بعرض وارتفاع القائمة الثانية ( المنبثقة ) يجب أن نضيف الخاصية ( **position:** ) تحمل القيمة ( **absolute;** ) وبدون أي قيم أخرى إلى وسم القائمة الثانية ( المنبثقة ) <ul> في صفحة CSS

```
ul { position:absolute; }
```

**سابعاً :-** الوسم الزائف ( \* )

للتحكم بالصفحة ككل نستخدم الخاصية ( \* ) وهي علامة النجمة حيث من خلال هذه الخاصية ( الوسم الزائف ) يمكننا التحكم بكامل الصفحة , مثلاً لو أردنا أن نتخلص من الحاشية العلوية التي تظهر بين أعلى الصفحة وأول عنصر موجود في الأعلى سنضيف لها الخاصية ( **margin:** ) ونسند لها القيمة ( **0** ) , كما في هذا المثال/

```
* { margin:0; }
```