

## مؤلفة الكتاب ...

**/ حنان حسن**

**مدربة فى برامج الجرافيك**

**مدربة فى برامج الويب**

**مصممة مواقع**

**مصممة جرافيك**

**مدربة كمبيوتر عن بعد**

**مخرج ومصمم فنى للمجلات والكتب**

### للتواصل

**E-mail: [hanan215\\_hasan@hotmail.com](mailto:hanan215_hasan@hotmail.com)**

**Mobile : +20 - 01006268150**

**Whatsapp : +20 - 01006268150**

**Egypt- cairo**

## بسم الله الرحمن الرحيم

اللهم صلي على محمد وعلى آل محمد  
مجيد كما صليت على إبراهيم وعلى آل إبراهيم إنك حميد  
وبارك على محمد وعلى آل محمد كما  
إبراهيم إنك حميد مجيد باركت على إبراهيم وعلى آل

عن ابي هريرة عن النبي -صلي الله عليه وسلم قال :

من كتم علما يعلمه جاء يوم القيامة ملجما بلجام من نار

تم الاستعانة ببعض المواقع التعليمية والمراجع العلمية لاعداد هذه المادة العملية

الحمد لله الذى تتم بنعمته الصالحات . هذا المؤلف صدقة جارية  
على روح ابي - الله يرحمة - وصدقة جارية لى . سائلة المولى ان

يخدم كل طالب وباحث علم

ارجو ممن يستفيد منه ان ينشرة على اوسع نطاق .

جزاكم الله خيرا

اسالكم الدعاء

شرح مبسط

لغة الـ CSS 3



يعد الأصدار الجديد من لغة الأنماط الأنسايبية CSS وهو الاصدار 3.0 Css نموذج رائع لتحسين وتطوير تنسيقات الويب ، حيث بدء الاصدار 3.0 Css يظهر علي الأفق ويستعرض قوته الجمالية في دعم العديد من الخصائص والمزايا التي ستنجح لمجتمع مصممي الويب العديد من المزايا والخصائص الجمالية لأعمالهم وتوفير الوقت والجهد بعمل مثل هذه المؤثرات يدوياً .

لعل الآن المعيق الأساسي لهذه الأصدارة هو عدم توفر الدعم الكامل لها من قبل كافة المتصفحات فهناك من يدعمها وهناك لا وكالعادة المتصفح إنترنت أكسلور لا يدعم وهناك العديد من الخصائص ايضا التي لايدعمها فيرفوكس ويدعمها سفاري باختصار شديد أنها لم تصل لمستوي الدعم الكامل من قبل المتصفحات ولكنها المستقبل ويوماً ما سنجد أن كافة المستعرضات القياسية تدعمها وسيلجأ المصممون لعدم النظر إلي باقي المستعرض التي لا تدعمه لأنها تحتوي علي الكثير من المزايا الرائعة التي ستضف جو كبير من الرفاهية علي تصاميم الويب.

تتضيف 3.0 Css مجموعة من المحددات والخصائص الجديدة الي CSS التقليدية واصطلاح على تسمية كل مجموعة من هذه الخصائص او المحددات باسم الوحدة "modules" ، ولعل اهم الوحدات المضافة او الاضافات الجديدة و هي :

- وحدة المحددات الجديدة Selectors
- وحدة الحدود والخلفيات الجديدة Backgrounds and Borders
- وحدة تحسين نموذج الصندوق Box Model

- وحدة مؤثرات النصوص الجديدة Text Effects
- وحدة التحويلات الهندسية ذات البعدين 2D Transformations
- وحدة التحويلات الهندسية ذات الثلاثة ابعاد 3D Transformations
- وحدة الحركات Animations
- وحدة الصفحات متعددة الاعمدة Multiple Column Layout
- وحدة واجهة المستخدم User Interface

**وسوف يتم شرح هذه الوحدات المضافة او الاضافات الجديدة فى لغة الـ CSS3**

الاضافات الجديدة فى لغة الـ CSS3

## اولا : الحواف المستديرة Roudned Corners :

الحواف المستديرة من الاضافات الجديدة التى تمت اضافتها للغة الـ CSS وهى خاصية رائعة ومن مزايا تصاميم الويب الحديثة وأغلب المواقع من فئة الويب 2.0 تستخدم الحواف المستديرة لأنها تضيف جمالية علي التصميم وتشعرك بالأناقة .

وهذه الخاصية المميزة أصبحت مدعومه وبكل سهولة مع CSS 3.0 ومدعومه بهذه المتصفحات ( فيرفوكس و سفاري ) ولكن إنترنت أكسلور لا يدعمها حتي بإصدارته الحديثة. IE 8.0

ويتم ذلك باستخدام خاصية الـ border-radius . وسوف يتم عرض امثلة لاستخدام هذه الخاصية كما يلى :

اولا عمل مستطيل بخلفية خضراء وحواف مستديرة ... لعمل ذلك يتم كتابة الكود التالى :

```
<!DOCTYPE html>

<html>

<head>

<style>

#rcorners1 {

    border-radius: 25px;

    background: #73AD21;

    padding: 20px;

    width: 200px;

    height: 150px;

}

</style>

</head>

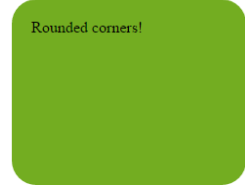
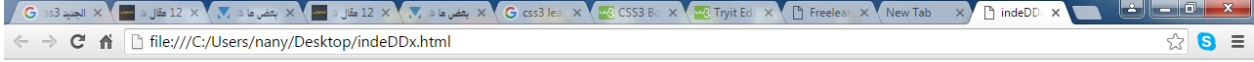
<body>

<p id="rcorners1">Rounded corners!</p>

</body>

</html>
```

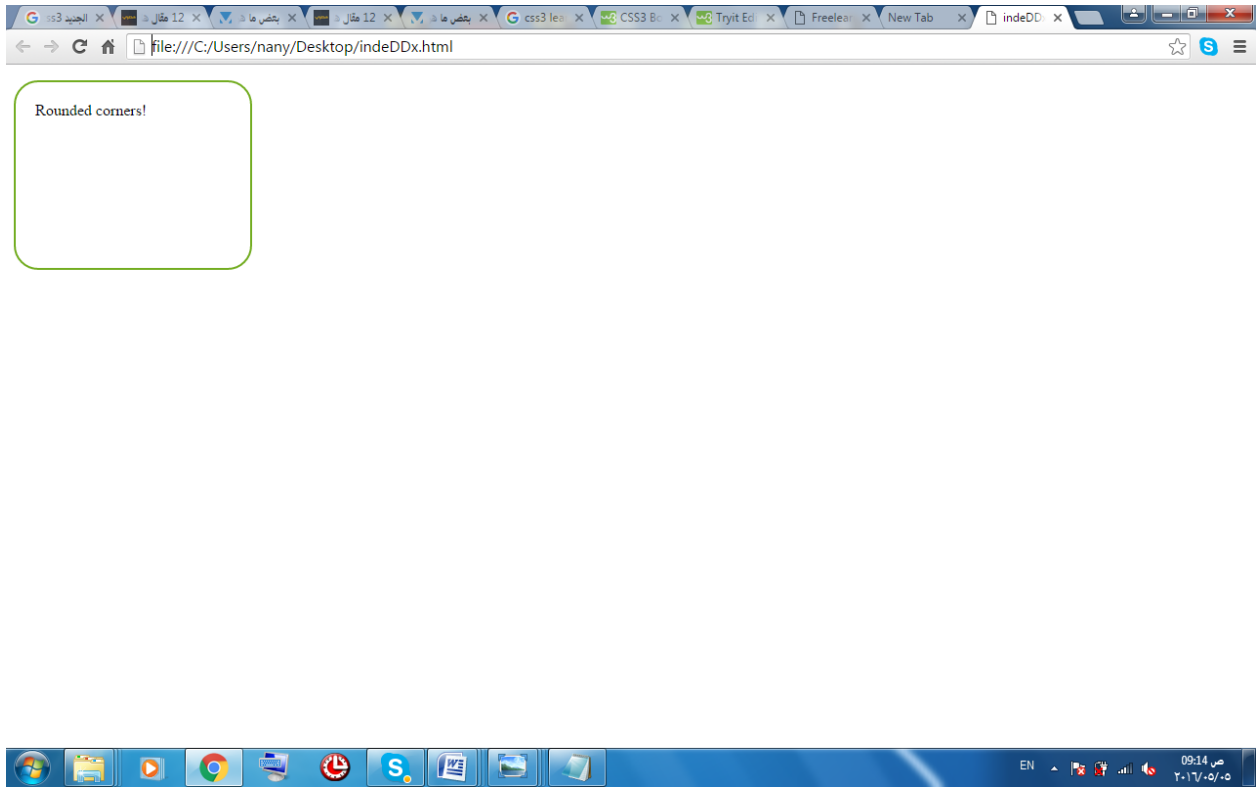
ويظهر في المتصفح كما يلي :



ثانيا : عمل مستطيل باطار ذات حواف مستديرة ... لعمل ذلك يتم كتابة الكود التالي :

```
<!DOCTYPE html>
<html>
<head>
<style>
#rcorners2 {
    border-radius: 25px;
    border: 2px solid #73AD21;
    padding: 20px;
    width: 200px;
    height: 150px;
}
</style>
</head>
<body>
<p id="rcorners2">Rounded corners!</p></body>
</html>
```

ويظهر فى المتصفح كما يلى :



ثانيا : عمل مستطيل بخلفية صورة وذات حواف مستديرة ... لعمل ذلك يتم كتابة الكود التالى :

```
<!DOCTYPE html>
<html>
<head>
<style>
#rcorners3 {
    border-radius: 25px;
    background: url(paper.gif);
    background-position: left top;
    background-repeat: repeat;
    padding: 20px;
    width: 200px;
    height: 150px;
}
</style>
</head>
<body>
<p id="rcorners3">Rounded corners! </p>
```



رابعاً : عمل مستطيل ذات حواف مستديرة ولكن الحواف مستديرة بقيم مختلفة...  
في هذه الحالة يتم كتابة قيم لكل حافة من حواف المستطيل الأربعة ( إذا كنت  
ترغب في تغيير الأربعة جوانب ) . و لعمل ذلك يتم كتابة الكود التالي :

```
<!DOCTYPE html>  
  
<html>  
  
<head>  
  
<style>  
  
#rcorners4 {  
  
    border-radius: 20px 50px 30px 5px;  
  
    background: #73AD21;  
  
    padding: 20px;  
  
    width: 200px;  
  
    height: 150px;  
  
}
```

```
#rcorners5 {  
  
    border-radius: 15px 50px 30px;  
  
    background: #73AD21;  
  
    padding: 20px;  
  
    width: 200px;
```

```
#rcorners6 {  
  border-radius: 15px 50px;  
  background: #73AD21;  
  padding: 20px;  
  width: 200px;  
  height: 150px;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>Four values - border-radius: 15px 50px 30px 5px:</p>
```

```
<p id="rcorners4"></p>
```

```
<p>Three values - border-radius: 15px 50px 30px:</p>
```

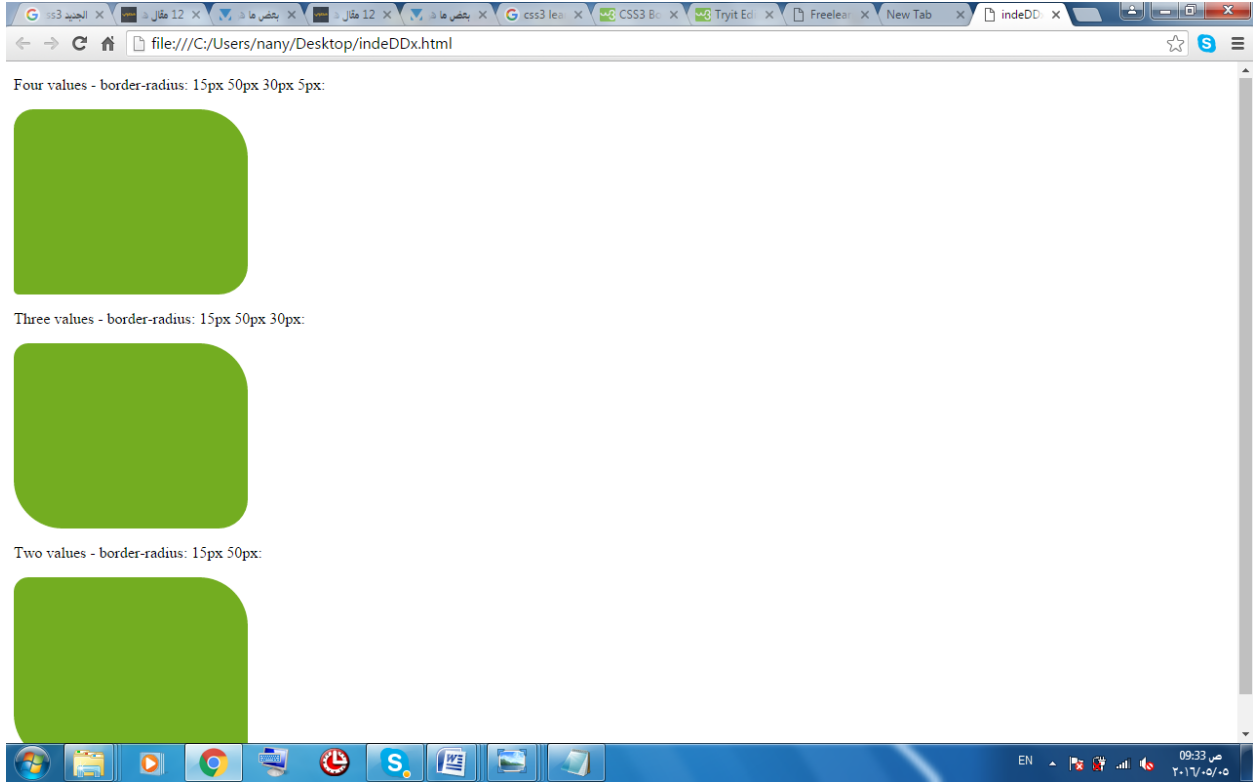
```
<p id="rcorners5"></p>
```

```
<p>Two values - border-radius: 15px 50px:</p>
```

```
<p id="rcorners6"></p>
```

```
</body>
```

ويظهر في المستعرض كما يلي :



يمكن ايضا عمل حواف بشكل اخر مستخدما نسبة او بعمليات حسابية .. ويتم ذلك  
بكتابة الكود كما يلي :

```
<!DOCTYPE html>
<html>
<head>
<style>
#rcorners7 {
    border-radius: 50px/15px;
    background: #73AD21;
    padding: 20px;
    width: 200px;
    height: 150px;
}

#rcorners8 {
    border-radius: 15px/50px;
    background: #73AD21;
    padding: 20px;
    width: 200px;
    height: 150px;
}
```

```
#rcorners9 {  
    border-radius: 50%;  
    background: #73AD21;  
    padding: 20px;  
    width: 200px;  
    height: 150px;  
}  
</style>  
</head>  
<body>  
  
<p>Elliptical border - border-radius: 50px/15px:</p>  
<p id="rcorners7"></p>  
  
<p>Elliptical border - border-radius: 15px/50px:</p>  
<p id="rcorners8"></p>  
  
<p>Ellipse border - border-radius: 50%:</p>  
<p id="rcorners9"></p>  
  
</body>  
</html>
```

ويظهر في المستعرض كما يلي :



وبهذا نكون قد انتهينا من شرح الخاصية الاولى الى تمت اضافتها في الاصدار 3  
وهي الحواف المستديرة Roudned Corners :

## ثانيا : اضافة صورة الأطارات **Border Image** :

باستخدامك للغة الـ CSS التقليدية يمكنك أن تضيف مجموعة من الأطارات لأي عنصر وليكن مثلا نص او رابط او صورة وهذا الأطار يعاني من التقليدية حيث يكون براوز سواء لون معين او محدد النقاط او ثلاثي الشكل ولكن بالنهاية أشكال محددة ومع الجيل الجديد من CSS يمكنك أن تضيف لهذه الأطارات صورة معينة تختارها بنفسك ..ويمكنك انشاء اطارات حاوية لعناصر الصفحة تحتوي علي صور

ويتم ذلك باستخدام خاصية الـ **border-image**

والمثال التالي يوضح كيفية استخدام خذة الخاصية ..

بفرض اني اريد اضافة براوز او اطار لنص ما .. هذا الاطار يكون على شكل



صورة وليكم الصورة التالية

لعمل ذلك يتم كتابة الكود التالي :

```
<!DOCTYPE html>

<html>

<head>

<style>

#borderimg {

    border: 10px solid transparent;

    padding: 15px;

    -webkit-border-image: url(border.png) 30 round; /* Safari 3.1-5 */

    -o-border-image: url(border.png) 30 round; /* Opera 11-12.1 */

    border-image: url(border.png) 30 round;

}

</style>

</head>

<body>

<p id="borderimg">Here, the middle sections of the image are
repeated to create the border.</p>

<p>Here is the original image:</p>

<p><strong>Note:</strong> Internet Explorer 10, and earlier versions,
do not support the border-image property.</p>
```



**ملحوظة مهمة جدا** ... لابد من اضافة الصورة السابقة فى نفس الفولدر الموجود  
بـ ملف الـ border.png ( الصورة هي border.png ) .

ويظهر المستعرض كما يلي



**ملحوظة مهمة جدا** .. يلاحظ ان الصورة قد تم تكررها حول الكلام لتكوين  
البرواز او الاطار . لعمل ذلك التكرار سيتم باستخدام قيمة ( round ) ام اذا اردنا  
ان لا يتم تكرار الصورة ومدھا ای عمل استریتش لها يستخدم قيمة  
( stretched )

للإطار

و يتم إنشاء الإطار بطريقتين:

• **repeated**: بتكرار جزء من الصورة.

• **stretch**: بتمديد جزء من الصورة.

و الشكل الجانبي يوضح الفرق بين الطريقتين

المذكورتين.



ولعمل

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
#borderimg {
```

```
border: 10px solid transparent;
```

```
padding: 15px;
```

```
-webkit-border-image: url(border.png) 30 stretch; /* Safari 3.1-5 */
```

```
-o-border-image: url(border.png) 30 stretch; /* Opera 11-12.1 */
```

```
border-image: url(border.png) 30 stretch;
```

```
}
```

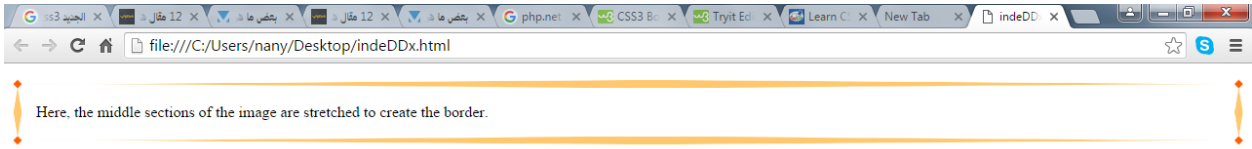
```
</style>
```

```
</head>
```

```
<body>
```

```
<p id="borderimg">Here, the middle sections of the image are stretched to create the border.</p>
```

```
<p>Here is the original image:</p>
```



Here is the original image:



**Note:** Internet Explorer 10, and earlier versions, do not support the border-image property.



يمكن عمل اطارات او برواز بقيم مختلفة كما يلي :

```
<!DOCTYPE html>

<html>

<head>

<style>

#borderimg {

    border: 10px solid transparent;

    padding: 15px;

    -webkit-border-image: url(border.png) 20% round; /* Safari 3.1-5 */

    -o-border-image: url(border.png) 20% round; /* Opera 11-12.1 */

    border-image: url(border.png) 20% round;

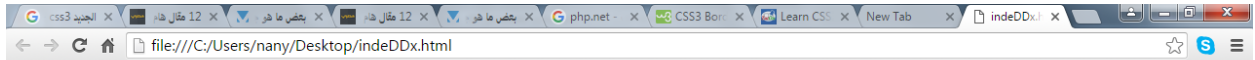
}

</style>

</head>

<body>
```

## ويظهر كما يلي :



Here, the middle sections of the image are stretched to create the border.

Here is the original image:



**Note:** Internet Explorer 10, and earlier versions, do not support the border-image property.



او کما یلی

```
<!DOCTYPE html>
<html>
<head>
<style>
#borderimg1 {
    border: 10px solid transparent;
    padding: 15px;
    -webkit-border-image: url(border.png) 50 round; /* Safari 3.1-5 */
    -o-border-image: url(border.png) 50 round; /* Opera 11-12.1 */
    border-image: url(border.png) 50 round;
}
#borderimg2 {
    border: 10px solid transparent;
    padding: 15px;
    -webkit-border-image: url(border.png) 20% round; /* Safari 3.1-5 */
    -o-border-image: url(border.png) 20% round; /* Opera 11-12.1 */
    border-image: url(border.png) 20% round;
}
#borderimg3 {
    border: 10px solid transparent;
    padding: 15px;
    -webkit-border-image: url(border.png) 30% round; /* Safari 3.1-5 */
    -o-border-image: url(border.png) 30% round; /* Opera 11-12.1 */
```

### ثالثا : الخلفيات الجديدة CSS3 Backgrounds

فى CSS3 تم اضافة خاصيتين جديدتين لمساعدة مطور الويب على التحكم بالخلفيات بشكل افضل ، الاول هى خاصية background-size والثانية هى background-origin .

#### 1- خاصية background-size

هى خاصية تسمح بتحديد ابعاد الصورة الخلفية بالبيكسل مما يسمح باعطاء الصورة كخلفية ابعادا تختلف عن الابعاد الاصلية للصورة وهو ما لم يكون ممكنا قبل CSS3 ويتم استخدام هذه الخاصية كما يلى ..

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
#example1 {
```

```
    border: 1px solid black;
```

```
    background:url(img_flwr.gif);
```

```
    background-repeat: no-repeat;
```

```
    padding:15px;
```

```
}
```

```
#example2 {
```

```
    border: 1px solid black;
```

```
    background:url(img_flwr.gif);
```

```
    background-size: 100px 80px;
```

```
    background-repeat: no-repeat;
```

```
    padding:15px;
```



```
<p>Original background-image:</p>
```

```
<div id="example1">
```

```
<h2>Lorem Ipsum Dolor</h2>
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam  
nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam  
erat volutpat.</p>
```

```
<p>Ut wisi enim ad minim veniam, quis nostrud exerci tation  
ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo  
consequat.</p>
```

```
</div>
```

```
<p>Resized background-image:</p>
```

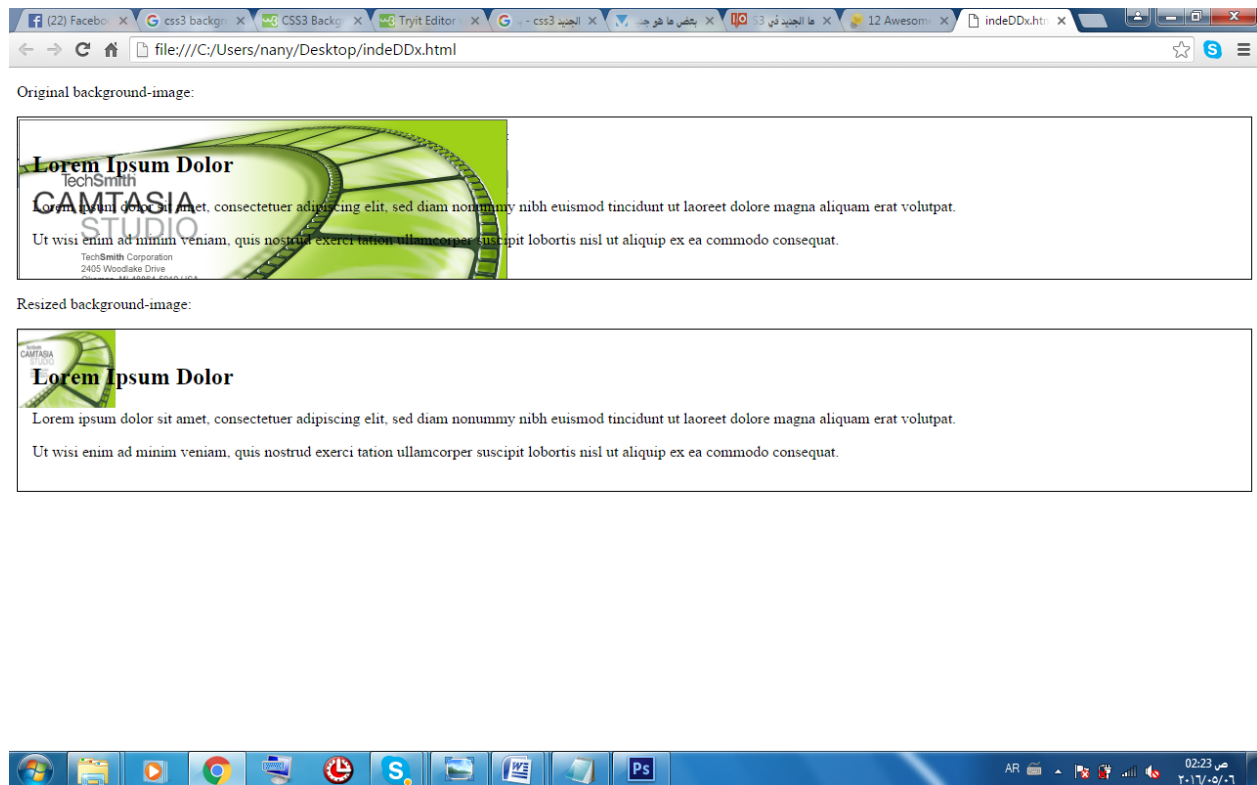
```
<div id="example2">
```

```
<h2>Lorem Ipsum Dolor</h2>
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam  
nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam  
erat volutpat.</p>
```

```
<p>Ut wisi enim ad minim veniam, quis nostrud exerci tation  
ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo  
consequat </p>
```

ويظهر في المستعرض كما يلي :



ويمكن ايضا استخدام قيمتان اخرى للتحكم فى مقاس الخلفية ، هذان القيمتان هما

**contain** and **cover**

بالنسبة لـ `contain` يتم زيادة مقاس الصورة المستخدمة كخلفية مع مراعاة المحافظة على النسبة بين الطول والعرض

بالنسبة لـ `cover` يتم زيادة مقاس الصورة المستخدمة كخلفية بحيث تغطي كل المساحة الموجودة فيها كخلفية

مثال على `contain`

```
<!DOCTYPE html>

<html>

<head>

<style>

#example1 {

    border: 1px solid black;

    background:url(gggttt.png);

    background-size:contain;

    background-repeat: no-repeat;

    padding:15px;

}

</style>

</head>

<body>

<p>Resized background-image:</p>

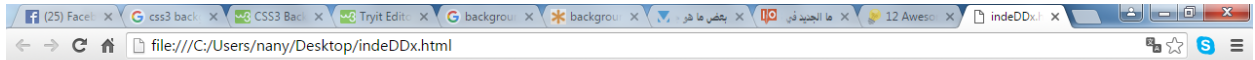
<div id="example1">

<h2>Lorem Ipsum Dolor</h2>

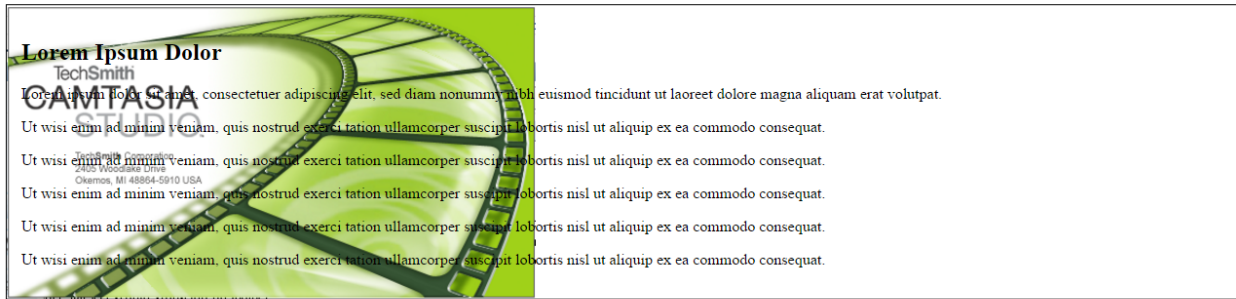
<p>Lorem ipsum dolor sit amet, consetetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</p>

<p>Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.</p>
```

ويظهر بالمستعرض كما يلي :



Resized background-image:



cover مثال على

```
<!DOCTYPE html>

<html>

<head>

<style>

#example1 {

    border: 1px solid black;

    background:url(gggttt.png);

    background-size: cover;

    background-repeat: no-repeat;

    padding:15px;

}

</style>

</head>

<body>

<p>Resized background-image:</p>

<div id="example1">

<h2>Lorem Ipsum Dolor</h2>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</p>

<p>Ut wisi enim ad minim veniam, quis nostrud exercitation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.</p>
```

ويظهر في المستعرض كما يلي ..



2- الخاصية الثانية background-origin

اما الخاصية الثانية فهي background-origin و التي تحدد نموذج الصندوق ليتم تطبيق الخلفية على اساسه بمعنى اخر من خلاله سيتم تحديد وضع الصورة المستخدمة كخلفية بالنسبة للصندوق ، وللتوضيح اكثر شاهد وضع هذه الصور

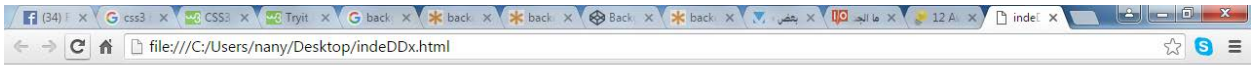


وخاصية background-origin يمكن ان تاخذ احد القيم التالية :

border-box, padding-box, content-box

**مثال على border-box**

```
<!DOCTYPE html>
<html>
<head>
<style>
#example1 {
border: 16px solid rgba(23,23,23,0.5);
height: 300px;
width: 300px;
background: url(cat.jpg) no-repeat;
background-origin: border-box;
background-size: cover;
```



background-origin: border-box;





## مثال علی padding-box

```
<!DOCTYPE html>

<html>

<head>

<style>

#example1 {

border: 16px solid rgba(23,23,23,0.5);

height: 300px;

width: 300px;

background: url(cat.jpg) no-repeat;

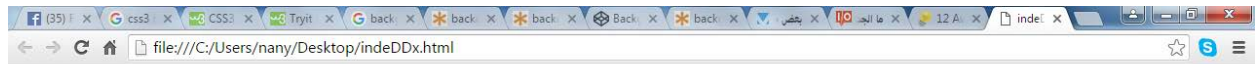
background-origin: padding-box;

background-size: cover;

padding: 16px;

margin: 0 auto;
```

ويظهر كما يلي



background-origin: border-box;

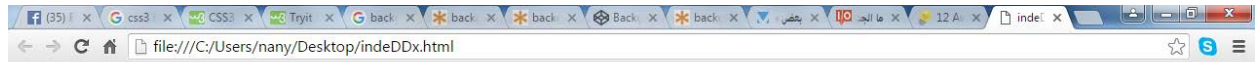


مثال على content-box

```
<!DOCTYPE html>
```

```
<html>
```





background-origin: border-box;



ملاحظة هامة ::

يجب مراعاة ان هناك بعض الخصائص التي يجب اضافة بادئة حتى يتمكن المستعرض من فهمها كما يلي

```
/* Safari 3-4, iOS 1-3.2, Android 1.6- */  
-webkit-border-radius: 12px;
```

```
/* Firefox 1-3.6 */  
-moz-border-radius: 12px;
```

```
/* Opera 10.5, IE 9, Safari 5, Chrome, Firefox 4, iOS 4,  
Android 2.1+ */  
border-radius: 12px;
```

### 3 - الخلفيات المتعددة Multiple Backgrounds :

وهي تسمح لك بإضافة أكثر من خلفية لنفس العنصر ، لتحكم أفضل وأسهل ،  
وهذه الخلفيات قد تكون بمقاسات مختلفة ، مع إمكانية تحديد مكان ومقاس كل  
صورة من صور الخلفيات

مثال على اضافة 3 صور كخلفية بقاسات وامكان مختلفة

```
<!DOCTYPE html>

<html>

<head>

<style>

#example1 {

    background: url(border.png) left top no-repeat, url(border.png) right bottom
no-repeat, url(dd.jpg) left top repeat;

    padding: 15px;

    background-size: 50px, 130px, auto;

}

</style>

</head>

<body>

<div id="example1">

<h1>Lorem Ipsum Dolor</h1>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</p>

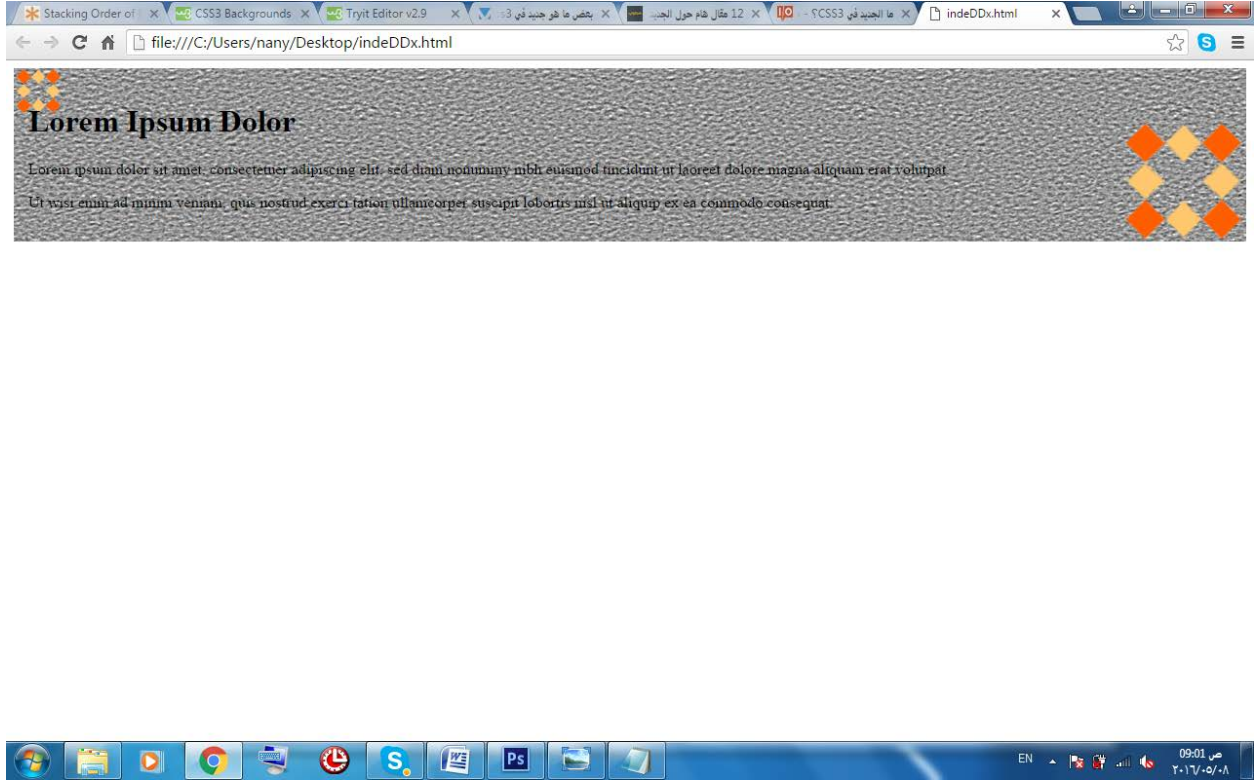
<p>Ut wisi enim ad minim veniam, quis nostrud exercitation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.</p>

</div>

</body>

</html>
```

## ويظهر في المستعرض كما يلي





## رابعاً : الالوان CSS3 Colors

لعل الألوان متاحة من قبل في الإجيلال السابقة من CSS ولكن الجديد أنك احياناً تريد ملء مساحة لونية محددة بلون معين ولكن تريد أن يكون هذا اللون بيه درجة شفافية تسمح للكائن خلفه او الشكل الذي بخلفيته أن يظهر وليس يظهر بشكل لون كامل ولا تري أي شئ خلفه .. مجرد مساحة لونية كاملة ، أصبح هذا متاح في الألوان في CSS الأصدار الجديد حيث يمكنك اختيار أي درجة لونية وكذلك اختيار درجة الشفافية لهذا اللون.

و وكما هو معروف انة الإجيلال السابقة من CSS كانت اللغة تمكن من تحديد الالوان اما باستخدام النظام العشري hexadecimal ( "#ff0000" ) او باسم اللون ( yellow ) او باستخدام نظام الـ rgb ("rgb(255, 0, 0)")

Colors in CSS are most often specified by:

- a valid color name - like "red"
- an RGB value - like "rgb(255, 0, 0)"
- a HEX value - like "#ff0000"

اما في الاصدار الجديد من اللغة فقد تم تحديد اللون مع اضافة بعض القيم كما يلي

:

## اولا تحديد اللون بالتحكم فى قيم RGBA : rgba(red, green, blue, alpha)

فى هذا النظام بيتم تحديد اللون كما كان سابقا بالاضافة الى alpha ، وتتراوح alpha ما بين ادنى قيمة وهى 0.0 الى اعلى قيمة وهى 1.0 كما هو موضح بالشكل :

```
rgba(255, 0, 0, 0.2);  
rgba(255, 0, 0, 0.4);  
rgba(255, 0, 0, 0.6);  
rgba(255, 0, 0, 0.8);
```

ولتطبيق ذلك يتم كتابة الكود كما يلى :

```
<!DOCTYPE html>

<html>

<head>

<style>

#p1 {background-color:rgba(255,0,0,0.3);}

#p2 {background-color:rgba(0,255,0,0.3);}

#p3 {background-color:rgba(0,0,255,0.3);}

#p4 {background-color:rgba(192,192,192,0.3);}

#p5 {background-color:rgba(255,255,0,0.3);}

#p6 {background-color:rgba(255,0,255,0.3);}

</style>

</head>

<body>

<p>RGBA colors:</p>

<p id="p1">Red</p>

<p id="p2">Green</p>

<p id="p3">Blue</p>

<p id="p4">Grey</p>

<p id="p5">Yellow</p>

<p id="p6">Cerise</p>

</body>

</html>
```

او بكتابة الكود بشكل اخر كما يلي وتظهر نفس النتيجة

```
<!DOCTYPE html>

<html>

<head>

<style>

#p1 {background-color:rgba(255,0,0,0.3);}

#p2 {background-color:rgba(0,255,0,0.3);}

#p3 {background-color:rgba(0,0,255,0.3);}

#p4 {background-color:rgba(192,192,192,0.3);}

#p5 {background-color:rgba(255,255,0,0.3);}

#p6 {background-color:rgba(255,0,255,0.3);}

</style>

</head>

<body>

<p>RGBA colors:</p>

<p div id="p1">Red</p>

<p div id="p2">Green</p>

<p div id="p3">Blue</p>

<p div id="p4">Grey</p>

<p div id="p5">Yellow</p>

<p div id="p6">Cerise</p>

</body>

</html>
```

## ثانيا : تحديد اللون بالتحكم فى قيم **hsl(hue, saturation, lightness)**.

فى هذا النظام يتم تحديد قيمة اللون بالتحكم فى قيمة اللون فى ثلاثة اشياء :

- **Hue** يعنى قيمة اللون فى عجلة الالوان هل اللون احمر ولا اصفر ولا

بنفسجى ، وهو يتراوح ما بين (0 to 360) ،

- 0 (or 360) is red
- 120 is green
- 240 is blue

- ايضا التحكم فى درجة الوضوح او درجة التشبع يعنى احمر قوى جدا ولا احمر ضعيف جدا **Saturation** ( سواء بالزيادة او التقليل ) وهى تكون نسبة مئوية ، نسبة 100% هى اعلى درجة وضوح او تشبع للون

- 100% is the full color

- ايضا التحكم فى الاضاءة او تفتيح اللون او تغميق اللون **lightness** وهى تكون نسبة مئوية ، 0% تعنى اسود او غامق و 100% تعنى ابيض او فاتح .

- 0% is dark (black) and 100% is white.

كما بالشكل التالى :

```
hsl(0, 100%, 30%);  
hsl(0, 100%, 50%);  
hsl(0, 100%, 70%);  
hsl(0, 100%, 90%);
```

## ويكتب الكود كما يلي :

```
<!DOCTYPE html>

<html>

<head>

<style>

#p1 {background-color:hsl(120,100%,50%);}

#p2 {background-color:hsl(120,100%,75%);}

#p3 {background-color:hsl(120,100%,25%);}

#p4 {background-color:hsl(120,60%,70%);}

#p5 {background-color:hsl(290,100%,50%);}

#p6 {background-color:hsl(290,60%,70%);}

</style>

</head>

<body>

<p>HSL colors:</p>

<p id="p1">Green</p>

<p id="p2">Light green</p>

<p id="p3">Dark green</p>

<p id="p4">Pastel green</p>

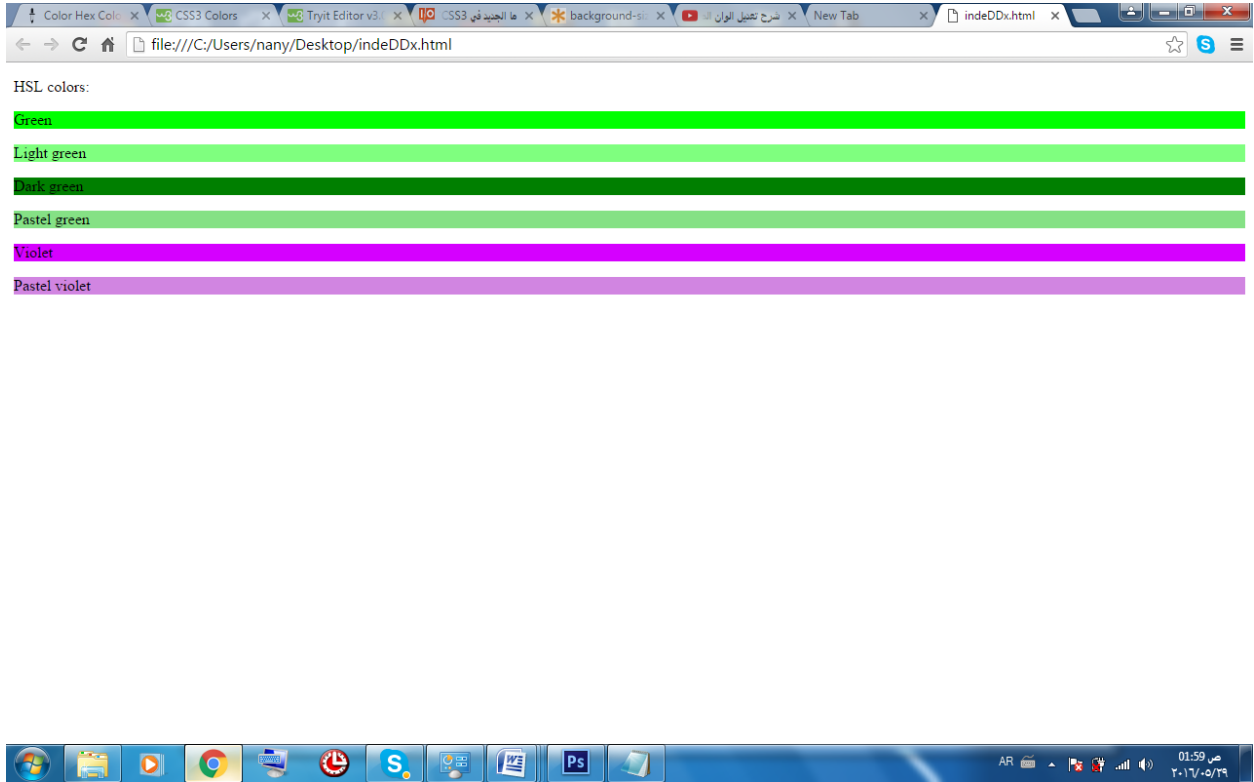
<p id="p5">Violet</p>

<p id="p6">Pastel violet</p>

</body>

</html>
```

و يظهر فى المستعرض كما يلى :



ثالثا : تحديد اللون بالتحكم فى قيم (hue, saturation, lightness, alpha)

كما هو موضح بالشكل :

```
hsla(0, 100%, 30%, 0.3);  
hsla(0, 100%, 50%, 0.3);  
hsla(0, 100%, 70%, 0.3);  
hsla(0, 100%, 90%, 0.3);
```

ويكتب الكود كما يلي :

```
<!DOCTYPE html>

<html>

<head>

<style>

#p1 {background-color:hsla(120,100%,50%,0.3);}

#p2 {background-color:hsla(120,100%,75%,0.3);}

#p3 {background-color:hsla(120,100%,25%,0.3);}

#p4 {background-color:hsla(120,60%,70%,0.3);}

#p5 {background-color:hsla(290,100%,50%,0.3);}

#p6 {background-color:hsla(290,60%,70%,0.3);}

</style>

</head>

<body>

<p>HSLA colors:</p>

<p id="p1">Green</p>

<p id="p2">Light green</p>

<p id="p3">Dark green</p>

<p id="p4">Pastel green</p>

<p id="p5">Violet</p>

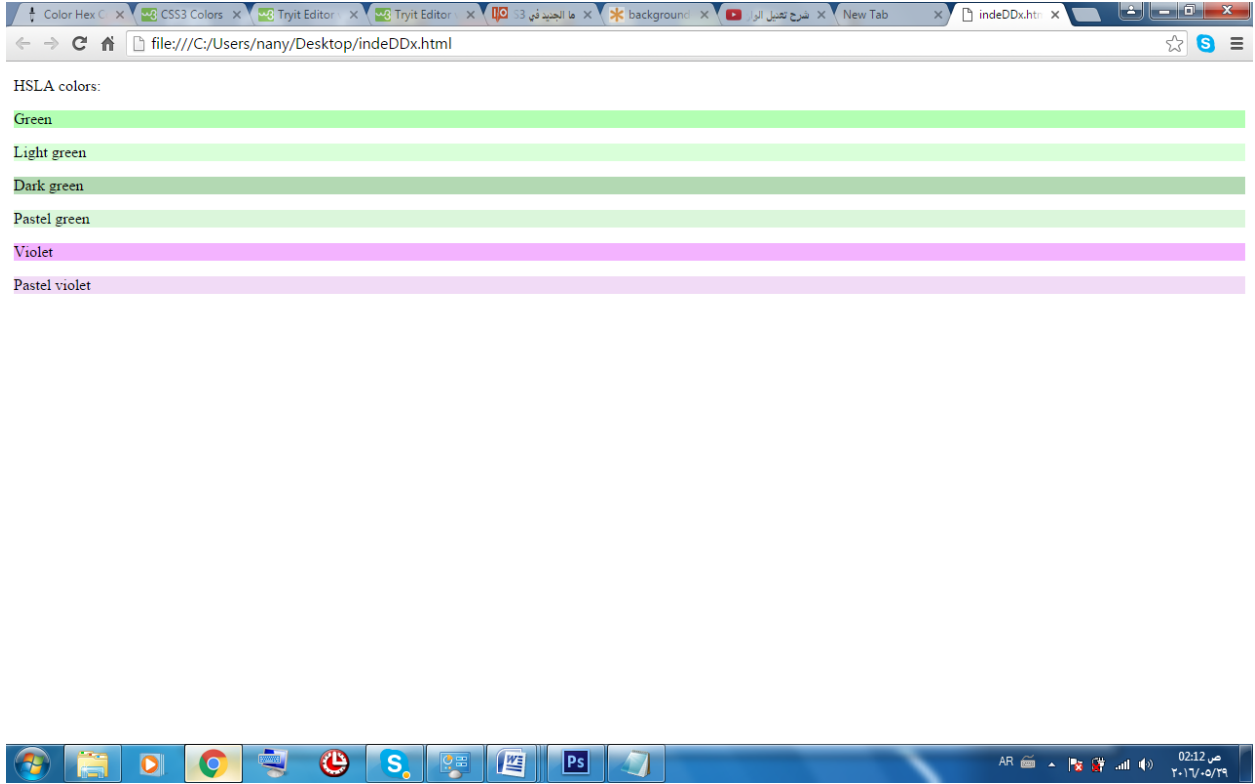
<p id="p6">Pastel violet</p>

</body>

</html>
```



## ويظهر فى المستعرض كما يلى



## رابعاً : تحديد اللون بالتحكم فى قيم شفافية اللون Opacity

تحديد اللون باستخدام النظام اللونى ( rgb ) ودرجة شفافية اللون وهى تتراوح ما بين ادنى قيمة وهى 0.0 الى اعلى قيمة وهى 1.0 كما هو موضح بالشكل :

```
rgb(255, 0, 0);opacity:0.2;  
rgb(255, 0, 0);opacity:0.4;  
rgb(255, 0, 0);opacity:0.6;  
rgb(255, 0, 0);opacity:0.8;
```

ويكتب الكود كما يلى :

```
<!DOCTYPE html>

<html>

<head>

<style>

#p1 {background-color:rgb(255,0,0);opacity:0.6;}

#p2 {background-color:rgb(0,255,0);opacity:0.6;}

#p3 {background-color:rgb(0,0,255);opacity:0.6;}

#p4 {background-color:rgb(192,192,192);opacity:0.6;}

#p5 {background-color:rgb(255,255,0);opacity:0.6;}

#p6 {background-color:rgb(255,0,255);opacity:0.6;}

</style>

</head>

<body>

<p>RGB colors with opacity:</p>

<p id="p1">Red</p>

<p id="p2">Green</p>

<p id="p3">Blue</p>

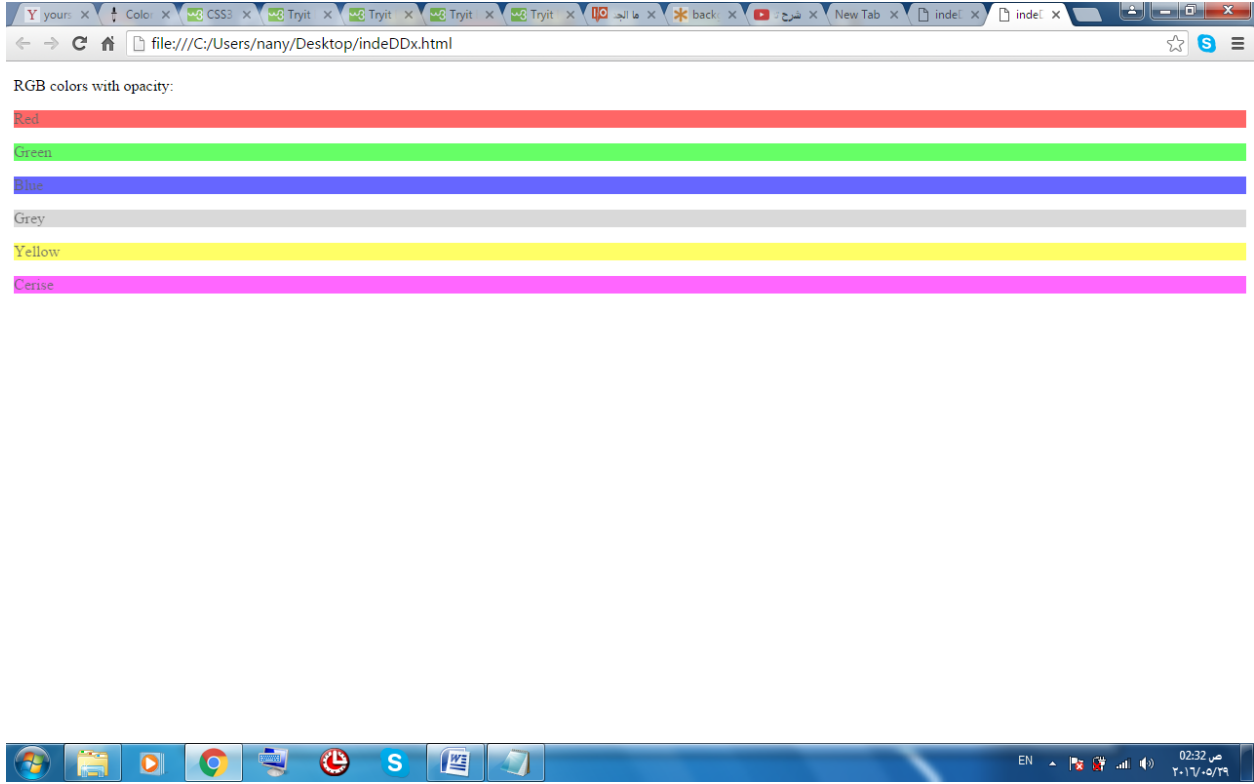
<p id="p4">Grey</p>

<p id="p5">Yellow</p>

<p id="p6">Cerise</p>

</body>

</html>
```



وبهذا نكون قد انتهينا من رابع مكون او اضافة جديدة من الوحدات المضافة او  
الاضافات الجديدة فى لغة الـ CSS3

## خامسا : التدرجات اللونية CSS3 Gradients

خاصية جديدة أخرى تم اضافتها للغة وهي إضافة التدرجات اللونية ، والتي يمكن إضافتها بكود بسيط بدل صنع صورة خاصة ووضعها ، وهناك اكثر من شكل للتدرج اللوني كما يلي :

- التدرج اللوني بشكل طولي من اعلى الى اسفل وهو التقليدي

### Linear Gradient - Top to Bottom (this is default)

ويكون بالشكل الموضح بالصورة :



ويكتب الكود كما يلي :

```
<!DOCTYPE html>

<html>

<head>

<style>

#grad1 {

    height: 200px;

    background: red; /* For browsers that do not support gradients */

    background: -webkit-linear-gradient(red, yellow); /* For Safari 5.1 to 6.0 */

    background: -o-linear-gradient(red, yellow); /* For Opera 11.1 to 12.0 */

    background: -moz-linear-gradient(red, yellow); /* For Firefox 3.6 to 15 */

    background: linear-gradient(red, yellow); /* Standard syntax (must be last) */

}

</style>

</head>

<body>

<h3>Linear Gradient - Top to Bottom</h3>

<p>This linear gradient starts at the top. It starts red, transitioning to yellow:</p>

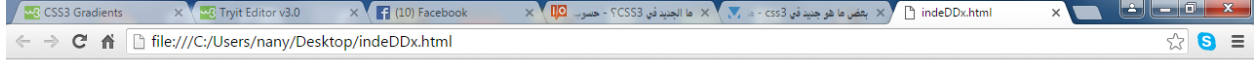
<div id="grad1"></div>

<p><strong>Note:</strong> Internet Explorer 9 and earlier versions do not support gradients.</p>

</body>

</html>
```

و يظهر فى المستعرض كما يلى يلى



### Linear Gradient - Top to Bottom

This linear gradient starts at the top. It starts red, transitioning to yellow:



Note: Internet Explorer 9 and earlier versions do not support gradients.



• التدرىح اللونى بشكل طولى من الشمال الى اليمين

### Linear Gradient - Left to Right

كما موضح بالصورة



```
<!DOCTYPE html>

<html>

<head>

<style>

#grad1 {

    height: 200px;

    background: red; /* For browsers that do not support gradients */

    background: -webkit-linear-gradient(left, red , yellow); /* For Safari 5.1 to 6.0 */

    background: -o-linear-gradient(right, red, yellow); /* For Opera 11.1 to 12.0 */

    background: -moz-linear-gradient(right, red, yellow); /* For Firefox 3.6 to 15 */

    background: linear-gradient(to right, red , yellow); /* Standard syntax (must be

last) */

}

</style>

</head>

<body>

<h3>Linear Gradient - Left to Right</h3>

<p>This linear gradient starts at the left. It starts red, transitioning to yellow:</p>

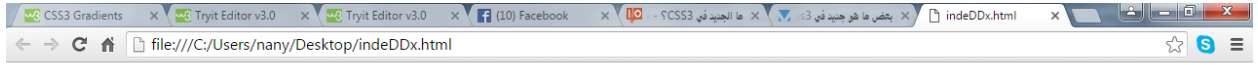
<div id="grad1"></div>

<p><strong>Note:</strong> Internet Explorer 9 and earlier versions do not support

gradients.</p>

</body>

</html>
```



### Linear Gradient - Left to Right

This linear gradient starts at the left. It starts red, transitioning to yellow:



**Note:** Internet Explorer 9 and earlier versions do not support gradients.



- التدرج اللوني بشكل طولي مع تحديد اتجاهة الافقى والراسى

### Linear Gradient – Diagonal

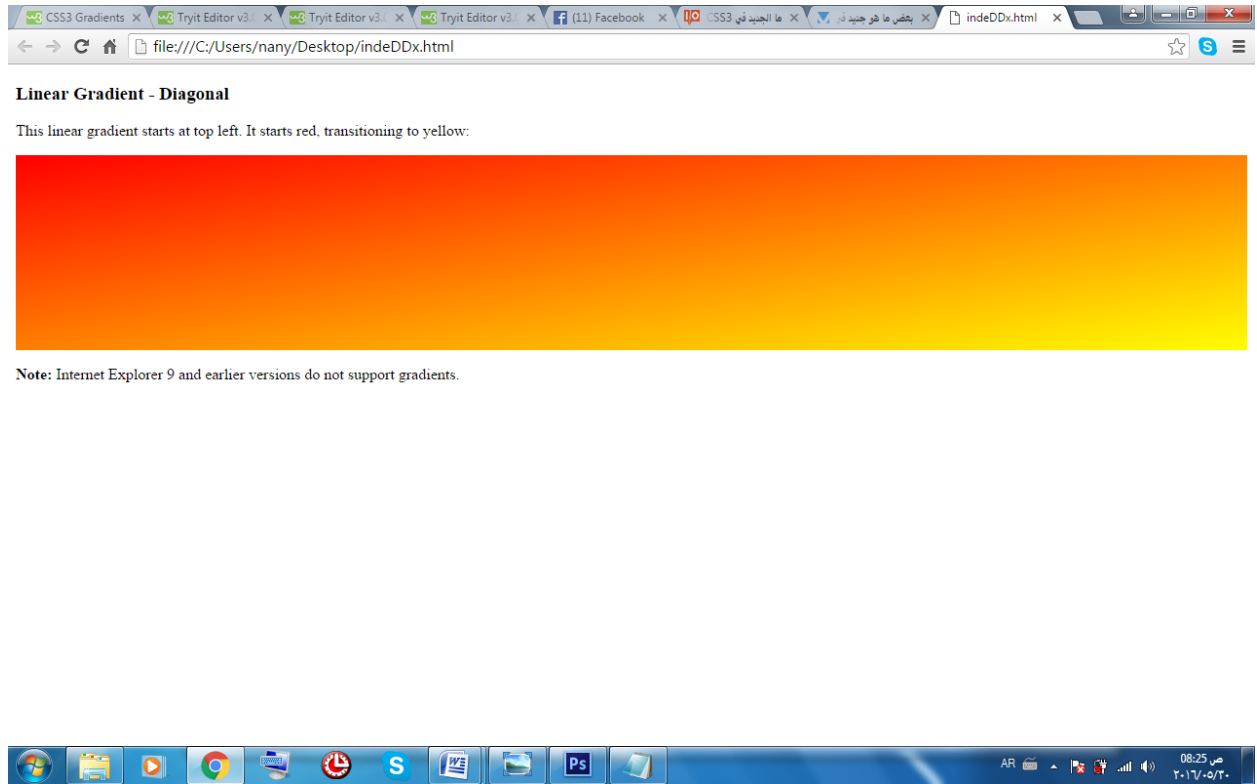
ويكتب الكود كما يلى





```
<!DOCTYPE html>
<html>
<head>
<style>
#grad1 {
  height: 200px;
  background: red; /* For browsers that do not support gradients */
  background: -webkit-linear-gradient(left top, red, yellow); /* For Safari 5.1 to 6.0 */
  background: -o-linear-gradient(bottom right, red, yellow); /* For Opera 11.1 to 12.0 */
  background: -moz-linear-gradient(bottom right, red, yellow); /* For Firefox 3.6 to 15 */
  background: linear-gradient(to bottom right, red, yellow); /* Standard syntax (must be
last) */
}
</style>
</head>
<body>

<h3>Linear Gradient - Diagonal</h3>
<p>This linear gradient starts at top left. It starts red, transitioning to yellow:</p>
```



- عمل تدرّيج لوني مع تحديد اتجاهة التدرّيج اللوني بتحديد زاوية التدرّيج اللوني

- Using Angles

```
<!DOCTYPE html>
<html>
<head>
<style>
#grad1 {
    height: 100px;
    background: red; /* For browsers that do not support gradients */
    background: -webkit-linear-gradient(0deg, red, yellow); /* For Safari 5.1 to 6.0 */
    background: -o-linear-gradient(0deg, red, yellow); /* For Opera 11.1 to 12.0 */
    background: -moz-linear-gradient(0deg, red, yellow); /* For Firefox 3.6 to 15 */
    background: linear-gradient(0deg, red, yellow); /* Standard syntax (must be last) */
}
```

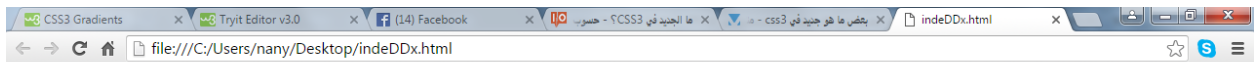
```
#grad2 {
    height: 100px;
    background: red; /* For browsers that do not support gradients */
    background: -webkit-linear-gradient(90deg, red, yellow); /* For Safari 5.1 to 6.0 */
    background: -o-linear-gradient(90deg, red, yellow); /* For Opera 11.1 to 12.0 */
    background: -moz-linear-gradient(90deg, red, yellow); /* For Firefox 3.6 to 15 */
```

```
#grad3 {  
    height: 100px;  
    background: red; /* For browsers that do not support gradients */  
    background: -webkit-linear-gradient(180deg, red, yellow); /* For Safari 5.1 to 6.0 */  
    background: -o-linear-gradient(180deg, red, yellow); /* For Opera 11.1 to 12.0 */  
    background: -moz-linear-gradient(180deg, red, yellow); /* For Firefox 3.6 to 15 */  
    background: linear-gradient(180deg, red, yellow); /* Standard syntax (must be last) */  
}  
  
#grad4 {  
    height: 100px;  
    background: red; /* For browsers that do not support gradients */  
    background: -webkit-linear-gradient(-90deg, red, yellow); /* For Safari 5.1 to 6.0 */  
    background: -o-linear-gradient(-90deg, red, yellow); /* For Opera 11.1 to 12.0 */  
    background: -moz-linear-gradient(-90deg, red, yellow); /* For Firefox 3.6 to 15 */  
    background: linear-gradient(-90deg, red, yellow); /* Standard syntax (must be last) */  
}  
  
</style>
```

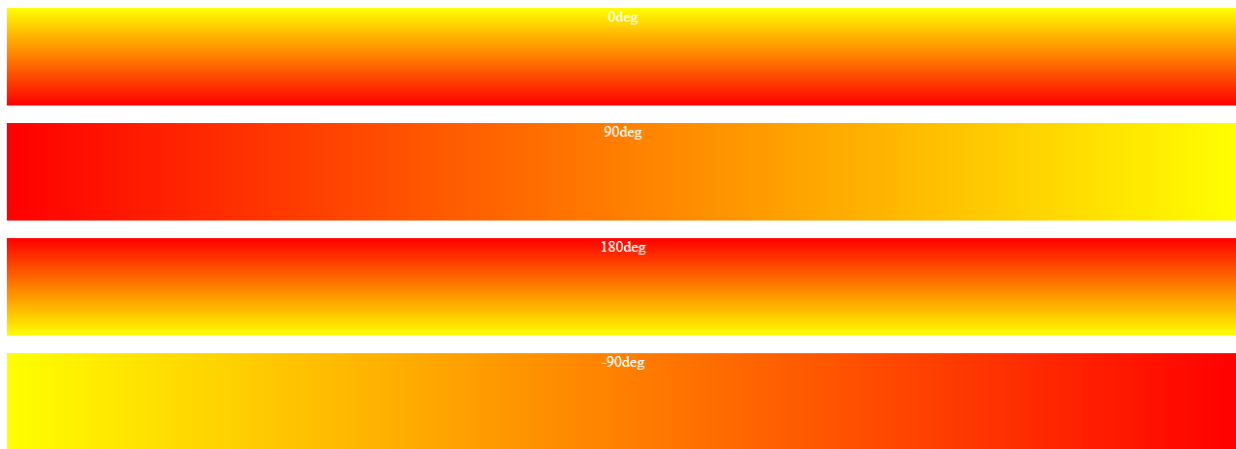
```
</head>
```

```
<body>
```

```
<h3>Linear Gradients - Using Different Angles</h3>
```



### Linear Gradients - Using Different Angles



**Note:** Internet Explorer 9 and earlier versions do not support gradients.



- عمل تدریج متعدد الالوان

## *Using Multiple Color Stops*

```
<!DOCTYPE html>

<html>

<head>

<style>

#grad1 {

    height: 200px;

    background: -webkit-linear-gradient(red, yellow, green); /* For Safari 5.1 to 6.0 */

    background: -o-linear-gradient(red, yellow, green); /* For Opera 11.1 to 12.0 */

    background: -moz-linear-gradient(red, yellow, green); /* For Firefox 3.6 to 15 */

    background: linear-gradient(red, yellow, green); /* Standard syntax (must be last) */

}

#grad2 {

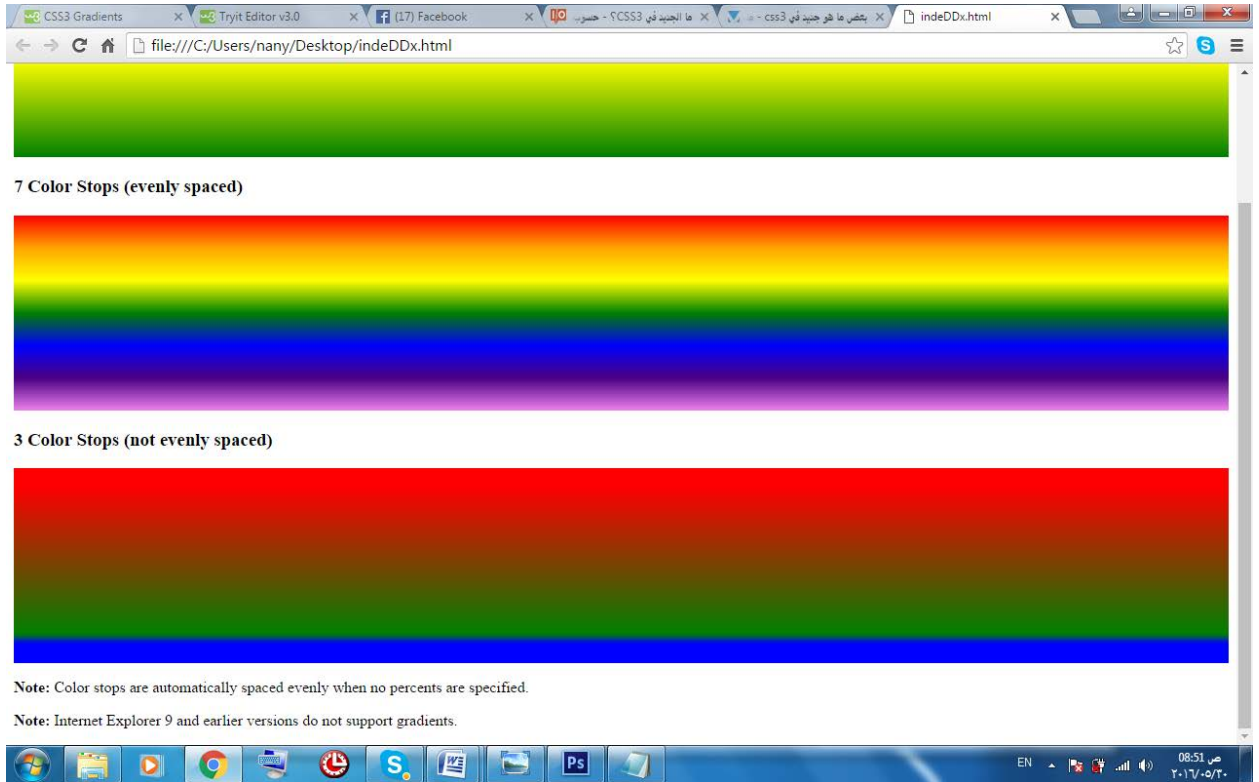
    height: 200px;

    background: -webkit-linear-gradient(red, orange, yellow, green, blue, indigo, violet); /* For Safari 5.1 to 6.0 */

    background: -o-linear-gradient(red, orange, yellow, green, blue, indigo, violet); /* For Opera 11.1 to 12.0 */

    background: -moz-linear-gradient(red, orange, yellow, green, blue, indigo, violet); /* For Firefox 3.6 to 15 */
```

```
#grad3 {  
    height: 200px;  
  
    background: -webkit-linear-gradient(red 10%, green 85%, blue 90%); /* For Safari 5.1 to  
6.0 */  
  
    background: -o-linear-gradient(red 10%, green 85%, blue 90%); /* For Opera 11.1 to 12.0  
*/  
  
    background: -moz-linear-gradient(red 10%, green 85%, blue 90%); /* For Firefox 3.6 to 15  
*/  
  
    background: linear-gradient(red 10%, green 85%, blue 90%); /* Standard syntax (must be  
last) */  
}  
  
</style>  
  
</head>  
  
<body>  
  
<h3>3 Color Stops (evenly spaced)</h3>  
  
<div id="grad1"></div>  
  
<h3>7 Color Stops (evenly spaced)</h3>  
  
<div id="grad2"></div>
```



- مثال على عمل تدرّيج متعدد الالوان من الشمال الى اليمين



```
<!DOCTYPE html>

<html>

<head>

<style>

#grad1 {

    height: 55px;

    background: -webkit-linear-gradient(left, red, orange, yellow, green, blue, indigo, violet); /* For Safari 5.1 to 6.0 */

    background: -o-linear-gradient(left, red, orange, yellow, green, blue, indigo, violet); /* For Opera 11.1 to 12.0 */

    background: -moz-linear-gradient(left, red, orange, yellow, green, blue, indigo, violet); /* For Fx 3.6 to 15 */

    background: linear-gradient(to right, red, orange, yellow, green, blue, indigo, violet); /* Standard syntax (must be last) */

}

</style>

</head>

<body>

<div id="grad1" style="text-align:center;margin:auto;color:#888888;font-size:40px;font-weight:bold">
```

- عمل تدرّيج لوني مستخدماً الشفافية

*Using Transparency*



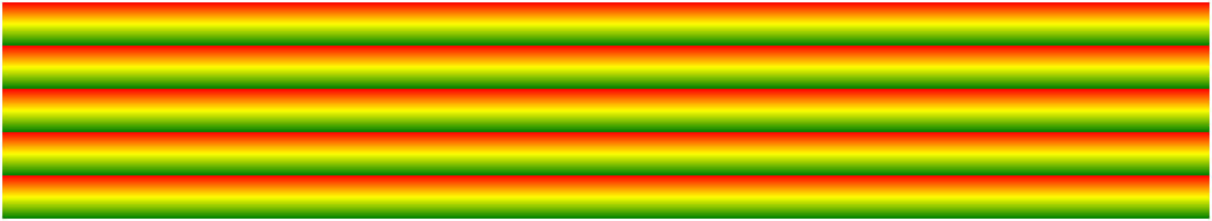
```
<!DOCTYPE html>
<html>
<head>
<style>
#grad1 {
    height: 200px;
    background: -webkit-linear-gradient(left, rgba(255,0,0,0), rgba(255,0,0,1)); /* For Safari 5.1 to 6.0
*/
    background: -o-linear-gradient(right, rgba(255,0,0,0), rgba(255,0,0,1)); /* For Opera 11.1 to 12.0
*/
    background: -moz-linear-gradient(right, rgba(255,0,0,0), rgba(255,0,0,1)); /* For Firefox 3.6 to 15
*/
    background: linear-gradient(to right, rgba(255,0,0,0), rgba(255,0,0,1)); /* Standard syntax (must
be last) */
}
</style>
</head>
<body>
```

```
<h3>Linear Gradient - Transparency</h3>
```

<p>To add transparency, we use the rgba() function to define the color stops. The last parameter in

- عمل تكرار للتدرج اللوني

*Repeating a linear-gradient*



```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
#grad1 {
```

```
    height: 200px;
```

```
    background: -webkit-repeating-linear-gradient(red, yellow 10%, green 20%); /* For Safari 5.1 to 6.0 */
```

```
    background: -o-repeating-linear-gradient(red, yellow 10%, green 20%); /* For Opera 11.1 to 12.0 */
```

```
    background: -moz-repeating-linear-gradient(red, yellow 10%, green 20%); /* For Firefox 3.6 to 15 */
```

```
    background: repeating-linear-gradient(red, yellow 10%, green 20%); /* Standard syntax (must be last) */
```

```
}
```

```
#grad2 {
```

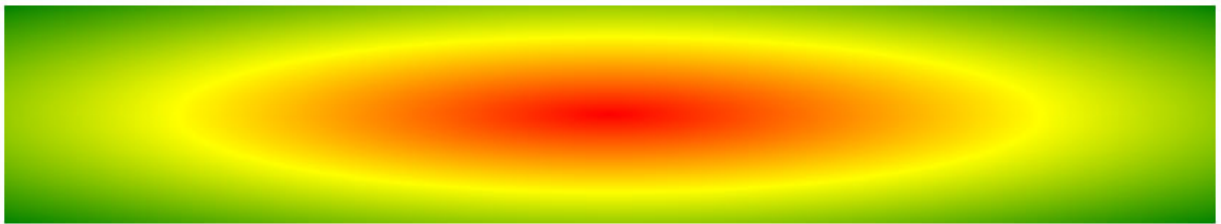
```
    height: 200px;
```

```
    background: -webkit-repeating-linear-gradient(45deg,red,yellow 7%,green 10%); /* For Safari 5.1 to 6.0 */
```

```
    background: -o-repeating-linear-gradient(45deg,red,yellow 7%,green 10%); /* For Opera 11.1 to
```

- عمل تدریج لونی بشکل دائری

*CSS3 Radial Gradients*



```
<!DOCTYPE html>

<html>

<head>

<style>

#grad1 {

    height: 150px;

    width: 200px;

    background: red; /* For browsers that do not support gradients */

    background: -webkit-radial-gradient(red, yellow, green); /* Safari 5.1 to 6.0 */

    background: -o-radial-gradient(red, yellow, green); /* For Opera 11.6 to 12.0 */

    background: -moz-radial-gradient(red, yellow, green); /* For Firefox 3.6 to 15 */

    background: radial-gradient(red, yellow, green); /* Standard syntax (must be last) */

}

</style>

</head>

<body>

<h3>Radial Gradient - Evenly Spaced Color Stops</h3>

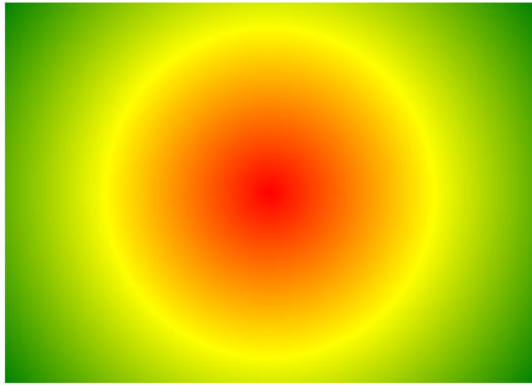
<div id="grad1"></div>
```

- عمل تدریج لونی بشكل دائری مع تحديد مساحة كل لون من ألوان التدریج اللونی

## Radial Gradient - Differently Spaced Color Stops

```
<!DOCTYPE html>
<html>
<head>
<style>
#grad1 {
  height: 150px;
  width: 200px;
  background: red; /* For browsers that do not support gradients */
  background: -webkit-radial-gradient(red 5%, yellow 15%, green 60%); /* Safari 5.1-6.0 */
  background: -o-radial-gradient(red 5%, yellow 15%, green 60%); /* For Opera 11.6 to 12.0 */
  background: -moz-radial-gradient(red 5%, yellow 15%, green 60%); /* For Firefox 3.6 to 15 */
  background: radial-gradient(red 5%, yellow 15%, green 60%); /* Standard syntax (must be last) */
}
</style>
</head>
<body>
<h3>Radial Gradient - Differently Spaced Color Stops</h3>
<div id="grad1"></div>
<p><strong>Note:</strong> Internet Explorer 9 and earlier versions do not support gradients.</p>
</body>
</html>
```





- عمل تدریج لونی بشکل دائری محدد

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
#grad1 {
```

```
    height: 150px;
```

```
    width: 200px;
```

```
    background: -webkit-radial-gradient(red, yellow, green); /* For Safari 5.1 to 6.0 */
```

```
    background: -o-radial-gradient(red, yellow, green); /* For Opera 11.6 to 12.0 */
```

```
    background: -moz-radial-gradient(red, yellow, green); /* For Fx 3.6 to 15 */
```

```
    background: radial-gradient(red, yellow, green); /* Standard syntax (must be last) */
```

```
}
```

```
#grad2 {
```

```
    height: 150px;
```

```
    width: 200px;
```

```
    background: -webkit-radial-gradient(circle, red, yellow, green); /* For Safari 5.1 to 6.0 */
```

```
    background: -o-radial-gradient(circle, red, yellow, green); /* For Opera 11.6 to 12.0 */
```

```
    background: -moz-radial-gradient(circle, red, yellow, green); /* For Fx 3.6 to 15 */
```

```
    background: radial-gradient(circle, red, yellow, green); /* Standard syntax (must be last) */
```

```
}
```

```
</style>
</head>
<body>

<h3>Radial Gradient - Shapes</h3>

<p><strong>Ellipse (this is default):</strong></p>
<div id="grad1"></div>

<p><strong>Circle:</strong></p>
<div id="grad2"></div>

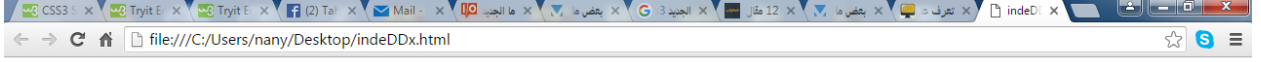
<p><strong>Note:</strong> Internet Explorer 9 and earlier versions do not support gradients.</p>

</body>
</html>
```

وبهذا نكون قد انتهينا من خامس مكون او اضافة جديدة من الوحدات المضافة  
او الاضافات الجديدة فى لغة الـ CSS3

## سادسا : اضافة الظل CSS3 Shadow Effects

### 1. اضافة الظل للنصوص وتحديد حجمة CSS3 Text Shadow



#### **Text-shadow effect!**

Note: Internet Explorer 9 and earlier versions, do not support the text-shadow property.



```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
h1 {
```

```
    text-shadow: 2px 2px;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

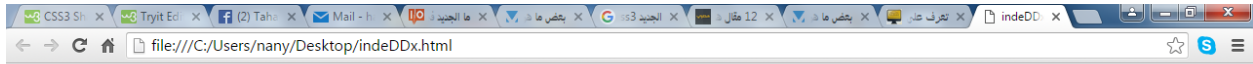
```
<h1>Text-shadow effect!</h1>
```

```
<p><b>Note:</b> Internet Explorer 9 and earlier versions, do not support the text-shadow property.</p>
```

```
</body>
```

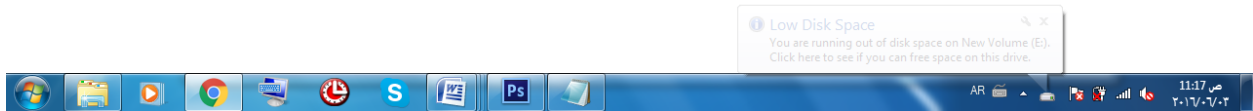
```
</html>
```

## 2. اضافة الظل للنصوص وتحديد الحجم و اللون CSS3 Text Shadow



### **Text-shadow effect!**

Note: Internet Explorer 9 and earlier versions, do not support the text-shadow property.



```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
h1 {
```

```
    text-shadow: 2px 2px red;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>Text-shadow effect!</h1>
```

```
<p><b>Note:</b> Internet Explorer 9 and earlier versions, do not support the text-shadow property.</p>
```

```
</body>
```

```
</html>
```

### 3. اضافة الظل للنصوص وتحديد الحجم و اللون ودرجة الدقة CSS3 Text Shadow blur

```
<!DOCTYPE html>

<html>

<head>

<style>

h1 {

    text-shadow: 2px 2px 5px red;

}

</style>

</head>

<body>

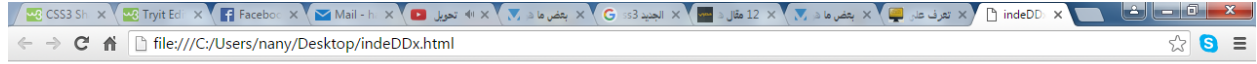
<h1>Text-shadow effect!</h1>

<p><b>Note:</b> Internet Explorer 9 and earlier versions, do not support the text-shadow property.</p>

</body>

</html>
```

## 4. اضافة الظل للنصوص وتحديد الحجم و اللون مستخدما الابيض والاسود



### *Text-shadow effect!*

**Note:** Internet Explorer 9 and earlier versions, do not support the text-shadow property.





```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
h1 {
```

```
    color: white;
```

```
    text-shadow: 2px 2px 4px #000000;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

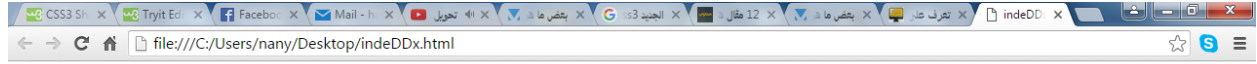
```
<h1>Text-shadow effect!</h1>
```

```
<p><b>Note:</b> Internet Explorer 9 and earlier versions, do not support the text-shadow property.</p>
```

```
</body>
```

```
</html>
```

## 5. مثال اضافة الظل للنصوص



### **Text-shadow effect!**

**Note:** Internet Explorer 9 and earlier versions, do not support the text-shadow property.



```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
h1 {
```

```
    text-shadow: 0 0 3px #FF0000;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>Text-shadow effect!</h1>
```

```
<p><b>Note:</b> Internet Explorer 9 and earlier versions, do not support the text-shadow property.</p>
```

```
</body>
```

```
</html>
```

## 6. اضافة الظل للنصوص مستخدما اكثر من ظل

# Multiple Shadows

```
<!DOCTYPE html>

<html>

<head>

<style>

h1 {

    text-shadow: 0 0 3px #FF0000, 0 0 5px #0000FF;

}

</style>

</head>

<body>

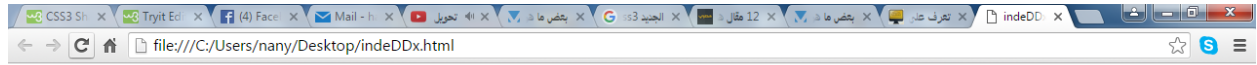
<h1>Text-shadow effect!</h1>

<p><b>Note:</b> Internet Explorer 9 and earlier versions, do not support the text-shadow property.</p>

</body>

</html>
```

## 7. مثال على اضافة الظل للنصوص مستخدما الابيض و الازرق الفاتح والغامق



### Text-shadow effect!

**Note:** Internet Explorer 9 and earlier versions, do not support the text-shadow property.



```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
h1 {
```

```
    color: white;
```

```
    text-shadow: 1px 1px 2px black, 0 0 25px blue, 0 0 5px darkblue;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>Text-shadow effect!</h1>
```

```
<p><b>Note:</b> Internet Explorer 9 and earlier versions, do not support the text-shadow property.</p>
```

```
</body>
```

```
</html>
```

يمكن الاستعانة بأشكال أخرى للـ **CSS Box Shadow** على الرابط التالي

[/https://css-tricks.com/snippets/css/css-box-shadow](https://css-tricks.com/snippets/css/css-box-shadow)

### سابعا : النصوص CSS3 Text

سوف نتطرق في الجزء الخاص بالنصوص الى اربعة من الخصائص الجديدة التي اضيفت في الاصدار الجديد من اللغة :

8.اولا : اضافة الظل للنصوص وتحديد الحجم و اللون ودرجة الدقة والكثافة والشفافية – وقد سبق وان تناولنا بالشرح هذا الجزء – راجع اضافة الظل في لغة الـ CSS

9.ثانيا :تقطيع النص في آخر الكلمة أو بعد رموز التنقيط ، إلا في حالة الكلمات الطويلة (كعناوين الويب و الروابط مثلا ) فيتم التقطيع داخلها حتى يتلاءم النص بالضبط مع حدود المساحة او الطبقة CSS3 Word Wrap

والمثال التالي يوضح الخاصية السابقة

```
<!DOCTYPE html>
<html lang="ar" dir="rtl">
<head>
  <meta charset="utf-8">
  <style>
    div {width: 160px;border :1px solid black;}
    .break {
      word-break: break-all;
    }
    .wrap {
      word-wrap: break-word;
    }
  </style>
</head>
<body>
  <h2></h2> </h2> التصرف التلقائي
  <div> هذا النص يحتوي على كلمة طويلة ستجاوز حدود الطبقة .
  </div>
  <h2>break-all</h2>
  <div class="break"> تقوم بتقطيع النص في أي مكان من الكلمة لتتلاءم بالضبط مع حدود الطبقة word-break الخاصة
  </div> و هذه كلمة طويلة سيتم تقطيعها أيضا
  <h2>break-word</h2>
  <div class="wrap"> تقوم بتقطيع النص في آخر الكلمة ، إلا في حالة الكلمات word-wrap الخاصة
  </div> الطويلة فسيتم التقطيع داخلها
</body>
</html>
```



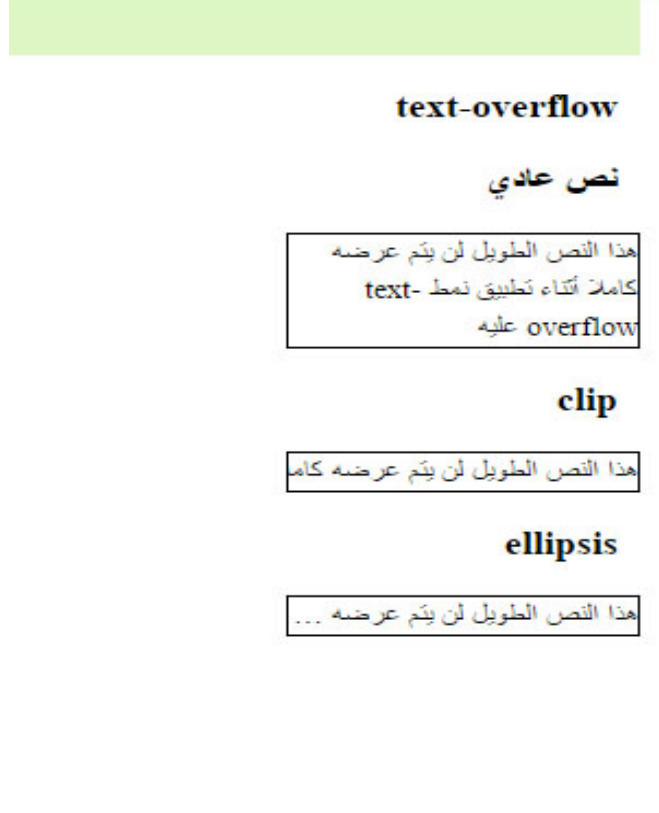


10. ثالثا : التحكم فى النصوص التي تخرج من المحتوى CSS3 Text Overflow . هذه الخاصية text-overflow تقوم بتحديد وضعية النص عندما يتجاوز هذا النص حدود الطبقة او المساحة . ويمكننا إعطاءها القيم التالية : clip|ellipsis|string|initial|inherit

ويوضح المثال التالى هذه الخاصية مستخدما القيم المختلفة لها

```
<!DOCTYPE html>
<html lang="ar" dir="rtl">
<head>
<meta charset="utf-8">
<style>
div {width : 160px; border :1px solid black;}
.overflows {
white-space: nowrap; /* تحذف الفراغات البيضاء و تجبر استكمال النص في سطر واحد */
overflow : hidden; /* إخفاء النص الذي يتجاوز حدود المساحة او الطبقة */
}
.clip {
text-overflow: clip;
}
.ellipsis{
text-overflow: ellipsis;
}
</style>
</head>
<body>
<h1>text-overflow</h1>
<h2>نص عادي</h2>
<div>عليه text-overflow هذا النص الطويل لن يتم عرضه كاملا أثناء تطبيق نمط
```

ويظهر في المستعرض كما يلي



11. رابعا : تقطيع النص وسط الكلمة أو خارجها حتى يتلاءم النص بالضبط مع

حدود المساحة او الطبقة CSS3 Word Break

مثال توضحي على ذلك

```
<!DOCTYPE html>

<html>

<head>

<style>

p.test1 {

    width: 140px;

    border: 1px solid #000000;

    word-break: keep-all;

}

p.test2 {

    width: 140px;

    border: 1px solid #000000;

    word-break: break-all;

}

</style>

</head>

<body>

<p class="test1">This paragraph contains some text. This line will-break-at-hyphens.</p>

<p class="test2">This paragraph contains some text. The lines will break at any character.</p>

<p><b>Note:</b> The word-break property is not supported in Opera 12 and earlier versions.</p>

</body>

</html>
```

## ويظهر في المستعرض كما يلي

Result Size: 705 x 596

This paragraph  
contains some text.  
This line will-break-  
at-hyphens.

This paragraph contain  
s some text. The lin  
es will break at any c  
haracter.

**Note:** The word-break property is not supported in Opera 12 and earlier versions.

## شكل يوضح جميع القيم للخاصيتين السابقتين

قيم word-wrap	
الشرح	القيمة
يتم تقطيع الكلمات فقط في النقاط المسموح بها مبدئياً	normal
تسمح بالتقطيع داخل بعض الكلمات	break-word
إرجاع الخاصية لقيمتها المبدئية : قراءة المزيد حول <a href="#">INITIAL</a>	initial
الخاصية تورت قيمة نمط الوسم الأب : قراءة المزيد حول <a href="#">INHERIT</a>	inherit
قيم word-break	
الشرح	القيمة
يتم تقطيع الكلمات فقط في النقاط المسموح بها مبدئياً	normal
تسمح بالتقطيع داخل و خارج أي كلمة	break-all
عدم السماح بالتقطيع داخل الكلمة	keep-all
إرجاع الخاصية لقيمتها المبدئية : قراءة المزيد حول <a href="#">INITIAL</a>	initial
الخاصية تورت قيمة نمط الوسم الأب : قراءة المزيد حول <a href="#">INHERIT</a>	inherit

وبهذا نكون قد انتهينا من سابع مكون او اضافة جديدة من الاضافات الجديدة في

لغة الـ CSS3

## ثامناً : الخطوط CSS3 Web Fonts

هذه الخاصية ليست جديدة رغم أنها ظهرت مع CSS3 فقد ظهرت الخاصية مسبقاً في نسخ ie القديمة، ثم تم التخلي عنها للمشاكل التي كانت فيها، وظهرت بحلة جديدة مجدداً في آخر نسخة.

مفهوم هذه الخاصية ببساطة لمن لا يعرفها هو أنك تستطيع استعمال أي خطّ تريده في الموقع بدلاً من الخطوط الافتراضية للمتصفحات أو [الخطوط الويب](#) [الأمنة](#) (والتي أصبحت مملة جداً)، وهي بسيطة جداً ظاهرياً، فكل ما عليك فعله هو إدراج مسار الخط، وإعطائه اسماً، ثم استخدامه لاحقاً في ملف ال CSS الخاص بك.

ويتم كتابة الخاصية كما يلي :

مثال على استخدام الخاصية

```
<!DOCTYPE html>

<html>

<head>

<style>

@font-face {

    font-family: 'Droid Arabic Naskh';

    src: url('droidnaskh-regular.ttf');}

font-weight: 400;
font-style: normal;
}

@font-face {

    font-family: myFirstFont;

    src: url(sansation_bold.woff);

    font-weight: bold;

}

div {

    font-family: myFirstFont;

}

</style>

</head>

<body>

<div>

With CSS3, websites can finally use fonts other than the pre-selected "web-safe" fonts.

</div>

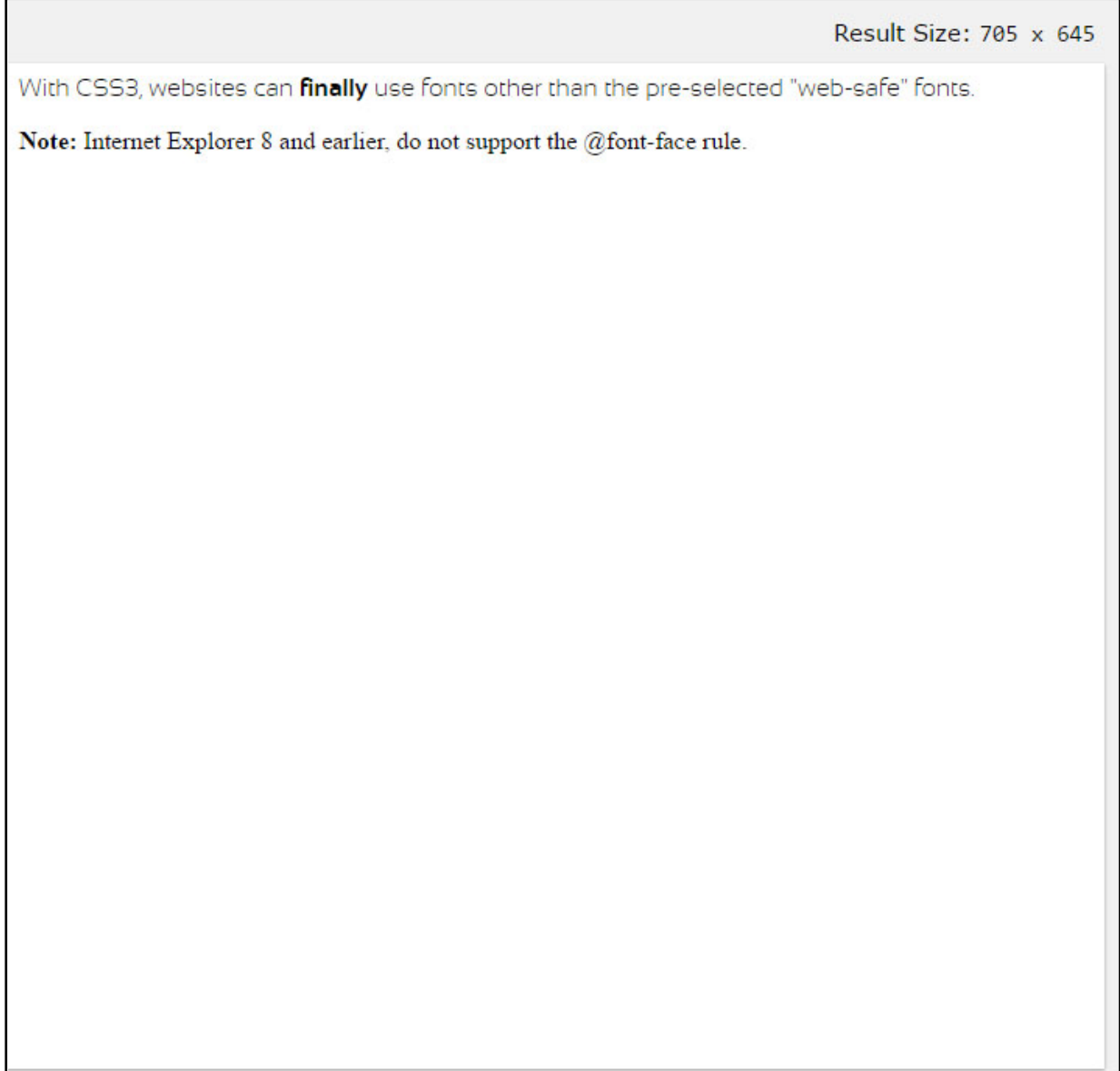
<p><b>Note:</b> Internet Explorer 8 and earlier, do not support the @font-face rule.</p>

</body>

</html>
```



ويظهر فى المستعرض كما يلى



شرح الكود:-

1. نبدأ الخاصية بتصريح @font-face

2- ثم أسفله نكتب اسم الخط الذي سنستعمل به الخط لاحقاً: font-family: (Droid Arabic Naskh); نستطيع استعمال أيّ اسم نريده، شرط أن يكون بأحرف لاتينية ولا يبدأ برقم أو بعلامة ترقيم ما.

3- في السطر الثاني نحدد مسار الخط-droidnaskh- src: url(‘regular.ttf’);

4- السطران الثالث والرابع غير أساسيان، ولكن سنحتاجهما إذا أردنا أن ندرج أكثر من وزن وتنسيق للخط الواحد، فالسطر الأول هو وزن الخط، و 400 تعني عادي (وكذلك 700 تعني سميك و 300 تعني خفيف)

أخيراً سطر تنسيق الخط: font-style: normal; تعني إدراج الخط بدون تنسيقات (نضع italic بدلاً من normal إذا كان الخط المدرج مائل)

**وبهذا نكون قد انتهينا من ثامن مكون او اضافة جديدة فى لغة الـ CSS3**

## تاسعا : التحويلات ثنائية الأبعاد CSS 2D Transforms

من بين الوحدات التي أضافتها css3 هي التحويلات ثنائية الأبعاد . بواسطة خاصيتين transform و transform-origin يمكننا التلاعب بإحداثيات عنصر من عناصر html على محوري x و y أفقيا و عموديا . تستعمل الخاصية transform مجموعة من الدوال ، تتيح لنا التأثير على العناصر ، إذ يمكننا تحريكها ، تغيير مقاييس أبعادها ، تغيير درجة الإنحراف ، الدوران و تغيير حجمها .

سنتعرف في هذا الموضوع على الدوال ثنائية الأبعاد . قبل ذلك لنلقي نظرة على المتصفحات التي تدعم الخاصيتين والموضحة بالشكل التالي :

رقم أول إصدار للمتصفحات التي بدأت تدعم الخاصيتين					
					الخاصية
10.0 9.0 -ms-	23.0 15.0 -webkit- 12.1 10.5 -o-	6.1 3.2 -webkit-	16.0 3.5 -moz-	36.0 4.0 -webkit-	transform
10.0 9.0 -ms-	23.0 15.0 -webkit- 12.1 10.5 -o-	6.1 3.2 -webkit-	16.0 3.5 -moz-	36.0 4.0 -webkit-	transform-origin (يحدد = بقيمتين)

الآن سوف نقوم بالبداية في شرح الدوال التي تتماشى مع الخاصية: transform والدوال هي :

- `translate()`

- rotate()
- scale()
- skewX()
- skewY()
- matrix()

### اولا : الدالة ( translate(x,y) ) :

يتم تحريك العنصر من وضعيته الأصلية حسب محوري (X من اليسار) و (Y من أعلى)  
في المثال أسفله سنقوم بتحريك العنصر ب (30 بيكسل) من اليسار و (50 بيكسل) من الأعلى

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <style>
    body{width:50%;}
    .mydiv{width:150px;height:100px;background:#8ac805;border:1px solid
gray;color:#fff;float:left;}p{border-bottom:2px solid black;padding-bottom:5px;}
    .translation{
      -ms-transform: translate(30px,50px);
      -moz-transform: translate(30px,50px);
      -webkit-transform: translate(30px,50px);
      transform: translate(30px,50px);
    }
  </style>
</head>
<body>
  <h1>CSS3 2D translate</h1>
  <div class="mydiv">المكان الأصلي</div><p>flammekueche kuglopf Salu bissame Hans turpis météor
Heineken turpis, so quam. amet, gewurztraminer ch'ai ac sit Salut isamme suspendisse adipiscing
porta schneck hopla Huguette schpeck et Racing. blottkopf, Gal ! libero, merci vielmols salu hopla
Oberschaeffolsheim quam, placerat ornare geht's ac wie amet tristique mamsell wurscht</p><div
style="clear:both;"></div>
  <div class="translation mydiv">translate(30px,50px)</div><p>flammekueche kuglopf Salu
bissame Hans turpis météor Heineken turpis, so quam. amet, gewurztraminer ch'ai ac sit Salut
bisamme suspendisse adipiscing porta schneck hopla Huguette schpeck et Racing. blottkopf, Gal !
libero, merci vielmols salu hopla Oberschaeffolsheim quam, placerat ornare geht's ac wie amet
tristique mamsell wurscht</p>
</body>
</html>

```

## CSS3 2D translate

المكان الأصلي

flammekueche kuglopf Salu bissame Hans turpis météor Heineken turpis, so  
quam. amet, gewurztraminer ch'ai ac sit Salut bisamme suspendisse adipiscing  
porta schneck hopla Huguette schpeck et Racing. blottkopf, Gal ! libero, merci  
vielmols salu hopla Oberschaeffolsheim quam, placerat ornare geht's ac wie  
amet tristique mamsell wurscht

flammekueche kuglopf Salu bissame Hans turpis météor Heineken turpis, so  
quam. amet, gewurztraminer ch'ai ac sit Salut bisamme suspendisse adipiscing  
a schneck hopla Huguette schpeck et Racing. blottkopf, Gal ! libero, merci  
nols salu hopla Oberschaeffolsheim quam, placerat ornare geht's ac wie  
tristique mamsell wurscht

translate(30px,50px)

الدالتان translateX و translateY

دالتان مشتقتان من . translate

translateX : تُحرّك العنصر أفقيا على محور X

مثال : transform: translateX(30px);

translateY : تُحرّك العنصر عموديا على محور Y

مثال : transform: translateY(50px);

لاستعمال الدالتين معا :

: transform: translateX(30px) translateY(50px);

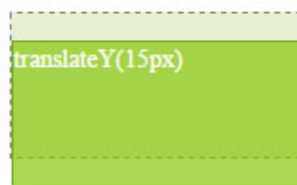
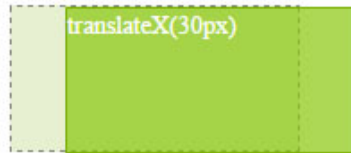
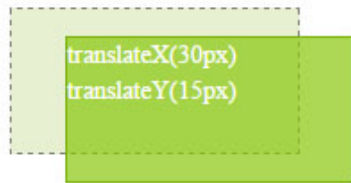
هذه الطريقة تعطينا نفس النتيجة السابقة

transform: translate(30px,50px)

أعطيكُم بعض الأمثلة:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <style>
    .box,.moved{width:160px;height:80px;background:rgba(138,200,15,0.7);border:1px solid
#78af03;color:#fff}
    .box{border:1px dashed gray;background:#e7f1d2;margin-bottom: 30px;}
    .translationXY {
      -ms-transform: translateX(30px) translateY(15px);
      -moz-transform: translateX(30px) translateY(15px);
      -webkit-transform:translateX(30px) translateY(15px);
      -o-transform: translateX(30px) translateY(15px);
      transform: translateX(30px) translateY(15px);
    }
    .translationX {
      -ms-transform: translateX(30px);
      -moz-transform: translateX(30px);
      -webkit-transform:translateX(30px);
      -o-transform: translateX(30px);
      transform: translateX(30px);
    }
    .translationY {
      -ms-transform: translateY(15px);
      -moz-transform: translateY(15px);
      -webkit-transform:translateY(15px);
      -o-transform: translateY(15px);
      transform: translateY(15px);
    }
  </style>
</head>
<body>
  <h1>CSS3 translateX AND translateY</h1>
  <div class="box">
    <div class="translationXY moved">translateX(30px) translateY(15px)</div>
  </div>
  <div class="box">
    <div class="translationX moved">translateX(30px)</div>
  </div>
  <div class="box">
    <div class="translationY moved">translateY(15px)</div>
  </div>
</body>
</html>
```

## CSS3 translateX AND translateY





## ثانيا : الدالة (rotate(deg)

تتيح هذه الدالة دوران العنصر ، نعطيها القيمة بعدد الدرجات . (deg) يمكن للقيمة أن تكون سلبية أو إيجابية ، كما يمكننا إعطاء قيمة أكبر من  $360^\circ$  درجة . مثلا  $400^\circ$  درجة ستعطينا  $40^\circ$  درجة فقط . حاليا لا يهم إن كتبتم  $40^\circ$  أو  $400^\circ$  درجة فالأمر لا يختلف . لكن أثناء تنشيط العناصر معنى  $400^\circ$  درجة سيختلف تماما و سيعني أن العنصر سيقوم بدورة كاملة حول النقطة الأصلية ( $360^\circ$ ) ثم يضيف  $40^\circ$  درجة، سوف نتناول مثال على هذه الداله :

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <style>
    body{width:50%;}
    .mydiv{width:100px;height:100px;background:#8ac805;border:1px solid
gray;color:#fff;float:left;}p{border-bottom:2px solid black;padding-bottom:25px;}
    .rotation {
      -ms-transform: rotate(20deg);
      -webkit-transform: rotate(20deg);
      -moz-transform: rotate(20deg);
      -o-transform: rotate(20deg);
      transform: rotate(20deg);
    }
  </style>
</head>
<body>
  <h1>CSS3 2D Rotate</h1>
  <div class="mydiv">المكان الأصلي</div><p>flammekueche kuglopf Salu bissame Hans turpis météor
Heineken turpis, so quam. amet, gewurztraminer ch'ai ac sit Salut isamme suspendisse adipiscing
porta schneck hopla Huguette schpeck et Racing. blottkopf, Gal ! libero, merci vielmols salu hopla
Oberschaeffolsheim quam, placerat ornare geht's ac wie amet tristique mamsell wurscht</p><div
style="clear:both;"></div>
  <div class="rotation mydiv">rotate(20deg)</div><p>flammekueche kuglopf Salu bissame Hans
turpis météor Heineken turpis, so quam. amet, gewurztraminer ch'ai ac sit Salut bisamme
suspendisse adipiscing porta schneck hopla Huguette schpeck et Racing. blottkopf, Gal ! libero,
merci vielmols salu hopla Oberschaeffolsheim quam, placerat ornare geht's ac wie amet tristique
mamsell wurscht</p>
</body>
</html>
```

ويظهر بالمستعرض بهذه الطريقة

### CSS3 2D Rotate

المكان الأصلي

flammekueche kuglopf Salu bissame Hans turpis météor Heineken turpis, so quam.  
amet, gewurztraminer ch'ai ac sit Salut bisamme suspendisse adipiscing porta schneck  
hopla Huguette schpeck et Racing. blottkopf, Gal ! libero, merci vielmols salu hopla  
Oberschaeffolsheim quam, placerat ornare geht's ac wie amet tristique mamsell wurscht

rotate(20deg)

flammekueche kuglopf Salu bissame Hans turpis météor Heineken turpis, so quam.  
amet, gewurztraminer ch'ai ac sit Salut bisamme suspendisse adipiscing porta schneck  
hopla Huguette schpeck et Racing. blottkopf, Gal ! libero, merci vielmols salu hopla  
Oberschaeffolsheim quam, placerat ornare geht's ac wie amet tristique mamsell wurscht

### ثالثا : الدالة scale(x,y)

تتيح لنا الدالة scale تغيير حجم العنصر ، تكبيره أو تصغيره حسب محوري X و

Y المحور X يلعب على حجم عرض العنصر و Y على طوله

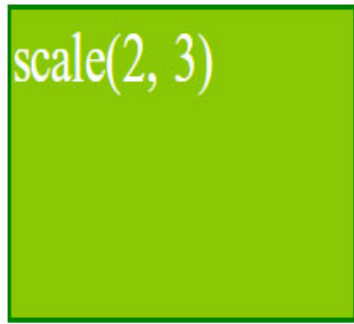
سوف اطرح مثال فية سنزيد عرض العنصر بضعفي حجمه الأصلي و طوله

بثلاثة أضعاف

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<style>
.mydiv{width:100px;height:60px;background:#8ac805;border:1px solid green;color:#fff;margin:0
0 80px 100px;}
.scales{
-ms-transform: scale(2 ,3);
-webkit-transform: scale(2 ,3);
-moz-transform: scale(2 ,3);
-o-transform: scale(2 ,3);
transform: scale(2 ,3);
}
</style>
</head>
<body>
<h1>CSS3 2D Scale</h1>
<div class="mydiv">الحجم الأصلي</div>
<div class="mydiv scales"> scale(2, 3)</div>
</body>
</html>
```

### CSS3 2D scale

الحجم الأصلي



الدالتان scaleY و scaleX

مشتقتان من الدالة scale

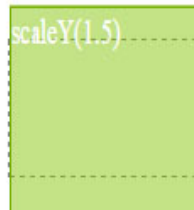
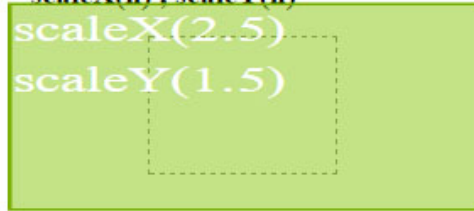
scaleX : تكبير عرض العنصر أفقيا وفق محور X

scaleY : تكبير طول العنصر عموديا وفق محور Y

مثال توضحي :

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<style>
.box,.moved{width:110px;height:80px;background:rgba(138,200,15,0.5);border:1px solid #79af03;color:#fff;}
.box{border:1px dashed gray;background:#fff;margin:0 0 40px 70px;}
.scale-xy {
  -webkit-transform:scaleX(2.5) scaleY(1.5);
  -moz-transform: scaleX(2.5) scaleY(1.5);
  -o-transform:  scaleX(2.5) scaleY(1.5);
  -ms-transform:  scaleX(2.5) scaleY(1.5);
  transform:  scaleX(2.5) scaleY(1.5);
}
.scale-x {
  -webkit-transform:scaleX(2.5);
  -moz-transform:  scaleX(2.5);
  -o-transform:  scaleX(2.5);
  -ms-transform:  scaleX(2.5);
  transform:  scaleX(2.5);
}
.scale-y {
  -webkit-transform:scaleY(1.5);
  -moz-transform:  scaleY(1.5);
  -o-transform:  scaleY(1.5);
  -ms-transform:  scaleY(1.5);
  transform:  scaleY(1.5);
}
</style>
</head>
<body>
<h1>scaleX(n) , scaleY(n)</h1>
<div class="box">
  <div class="moved scale-xy">scaleX(2.5) scaleY(1.5)</div>
</div>
<div class="box">
  <div class="moved scale-x">scaleX(2.5)</div>
</div>
<div class="box">
  <div class="moved scale-y">scaleY(1.5)</div>
</div>
</body>
</html>
```

**scaleX(n) , scaleY(n)**





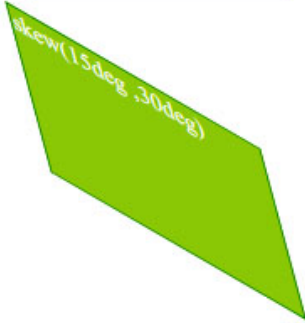
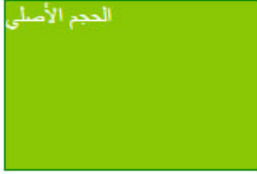
## رابعاً : الدالة skew(x,y)

الدالة skew تلعب على درجة ميلان زوايا العنصر وفق محوري X و Y .  
في المثال أسفله ، يميل العنصر 15° درجة حول محور X و 30° درجة حول محور Y

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <style>
    .mydiv{width:150px;height:100px;background:#8ac805;border:1px solid green;color:#fff;margin:0
0 50px 20px;}
    .oblique {
      -ms-transform: skew(15deg ,30deg);
      -webkit-transform: skew(15deg ,30deg);
      -moz-transform: skew(15deg ,30deg);
      -o-transform: skew(15deg ,30deg);
      transform: skew(15deg ,30deg);
    }
  </style>
</head>
<body>
  <h1>CSS3 2D Skew</h1>
  <div class="mydiv">الحجم الأصلي</div>
  <div class="mydiv oblique"> skew(15deg ,30deg)</div>
</body>
</html>
```

## CSS3 2D skew

الحجم الأصلي



## الدالتان skewY و skewX

الدالتان مشتقتان من الدالة . skew() بطبيعة الحال ، كل دالة ستأخذ قيمة واحدة

فقط ، مثلا إذا أردنا أن يميل العنصر  $15^\circ$  درجة حول محور X

```
transform: skewX(15deg)
```

أما إذا أردنا استعمال الدالتين معا:

```
transform: skewX(15deg) skewY(15deg);
```

## خامسا : الدالة () matrix

هذه الدالة عبارة عن مصفوفة تجمع بين وظائف كل الدوال الأخرى ، من تحريك و ميلان و دوران و تغيير الحجم و نعطيهها 6 قيم . لسوء الحظ أن الدالة تستعمل حسابات رياضية معقدة و ليست في مستوى الجميع . بالنسبة للمعلمين يمكنكم الإطلاع على هذا الرابط لفهم طريقة توظيف المصفوفات في لغة "css33" المقال باللغة الإنجليزية"

سنقتصر حاليا على المثال التالي :

لنفترض أننا نريد إجراء مجموعة من التحويلات التي رأيناها ، دفعة واحدة:

```
transform: rotate(15deg) translateX(230px) scale(1.5, 2.6) skew(220deg, -150deg) translateX(230px);
```

الكود السابق يعادل المصفوفة أسفله

```
transform: matrix(1.06, 1.84, 0.54, 2.8, 466px, 482px);
```

## سادسا : transform-origin

تستعمل هذه الخاصية موازاة مع الخاصية transform  
تتيح لنا تغيير إحداثيات العنصر الذي أجرينا عليه التحويلات . أي تغيير نقطة  
الأصل للمحورين X و Y . في التحويلات الثلاثية الأبعاد يمكننا أيضا اللعب  
على عمق المحور Z .  
القيمة المبدئية للخاصية ، تُمرکز النقطة الأصلية "0" وسط العناصر .  
و هي تعادل :

```
transform-origin: 50% 50%;
```

الخاصية سهلة الإستعمال ، أعطيكم بعض الأمثلة لتفهموا دورها جيدا:

```

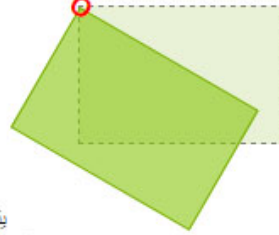
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<style>
p{float:right;width:100px;text-align:right;}
.cadre,.rotation{width:110px;height:80px;background:rgba(138,200,15,0.6);border:1px solid #79af03;color:#fff;}
.cadre{border:1px dashed gray;background:#e9f1d7;margin:0 0 15px 30px;}
.axis1{margin-top:-12px;margin-left:-7px;}
.axis3{margin-bottom:-11px;margin-right:-6px;}
h4 {border-top:1px solid silver;}
.rotation {
-webkit-transform:rotate(30deg);
-moz-transform: rotate(30deg);
-o-transform: rotate(30deg);
-ms-transform: rotate(30deg);
transform: rotate(30deg);
font-weight:bold;font-family:Arial;color:red;
}
.origine0 {
-webkit-transform-origin:0 0;
-moz-transform-origin: 0 0;
-o-transform-origin: 0 0;
-ms-transform-origin: 0 0;
transform-origin: 0 0;
}
.origine50 {
-webkit-transform-origin: 50% 50%;
-moz-transform-origin: 50% 50%;
-o-transform-origin: 50% 50%;
-ms-transform-origin: 50% 50%;
transform-origin: 50% 50%;
display: table-cell;
vertical-align: middle;
text-align:center;
}
.origine100 {
-webkit-transform-origin:100% 100%;
-moz-transform-origin: 100% 100%;
-o-transform-origin: 100% 100%;
-ms-transform-origin: 100% 100%;
transform-origin: 100% 100%;
display: table-cell;
vertical-align: bottom;
text-align:right;
}
</style>
</head>
<body>
<h1>rotate(30deg)</h1>
<h4>origine: 0 0</h4>
<p> <math>\hat{y}</math> يتيم الدوران على نقطة الأصل أعلى يسار العلية </p>
<div class="cadre">
<div class="rotation origine0"><div class="axis1">O</div></div>
</div>
<h4>origine: 50% 50%</h4>
<p> <math>\hat{y}</math> يتيم الدوران على نقطة الأصل وسط العلية </p>
<div class="cadre">
<div class="rotation origine50">O</div>
</div>
<h4>origine: 100% 100%</h4>
<p> <math>\hat{y}</math> يتيم الدوران على نقطة الأصل أسفل يمين العلية </p>
<div class="cadre">
<div class="rotation origine100"><div class="axis3">O</div></div>
</div>

```

**rotate(30deg)**

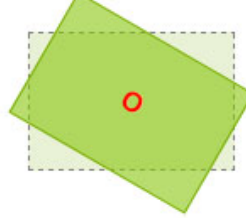
يتم الدوران على نقطة الأصل أعلى يسار العلية

**origine: 0 0**



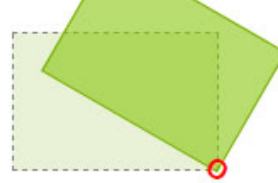
يتم الدوران على نقطة الأصل وسط العلية

**origine: 50% 50%**



يتم الدوران على نقطة الأصل أسفل يمين العلية

**origine: 100% 100%**



بالإضافة للنسبة المئوية ، يمكننا استعمال مجموعة من القيم الأخرى مع

الخاصية `transform-origin` و هي :

`top|bottom|right|left|center|length|inherit|initial`

مثلا ; `transform-origin: 50% 50%` يمكننا أيضا كتابتها كالتالي:

```
transform-origin: center center;
```

مجموع هذه القيم تتعلق بخاصية transform-origin في فضاء ثنائي الأبعاد . أما في فضاء ثلاثي الأبعاد ، يمكننا أيضا تحديد النقطة الأصلية لمحور . "Z" لكن القيمة الوحيدة المسموح بها لهذا المحور هي قيمة length : ، مثال:

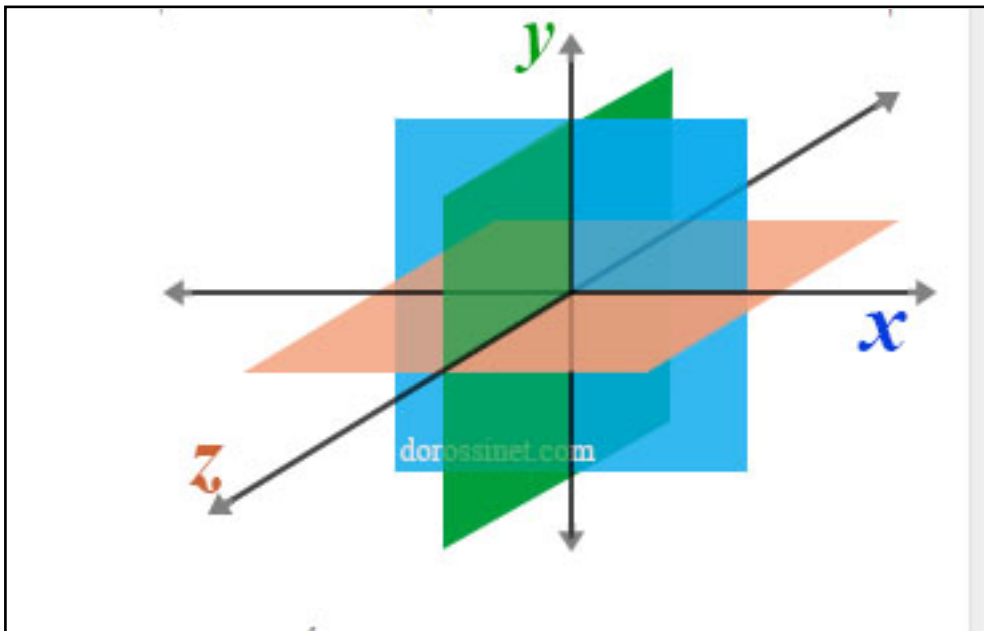
```
transform-origin: right bottom 2cm;
```

**وبهذا نكون قد انتهينا من تاسع مكون او اضافة جديدة فى لغة الـ CSS3**

## عاشرا : التحويلات ثلاثية الأبعاد CSS 3D Transforms

تعاملنا في الدرس السابق مع التحويلات ثنائية الأبعاد ، و التي تركز على محورين ، محور العرض "X" و محور الطول . "Y" لتحويل العناصر في فضاء ثلاثي الأبعاد ، سنضيف محورا ثالثا ، و هو محور "Z" الذي يمنحنا التحكم في عمق العناصر كما في طولها وعرضها.

في مجال الويب ، ألفنا الإشتغال في فضاء ثنائي الأبعاد . حيث أي صفحة ممثلة بمحورين ، محور X الأفقي و محور Y العمودي . النقطة الأصلية التي تمثل تقاطع المحورين مبدئيا توجد أعلى يسار الصفحة . إذا حركنا العناصر إلى اليسار فهي تأخذ قيمة سلبية و قيمة إيجابية على يمين محور . X نفس الشيء على محور Y ، التحرك إلى أعلى يأخذ قيمة سلبية و إيجابية إلى أسفل الصفحة . بالنسبة للفضاء الثلاثي الأبعاد ، و المُمثّل بالمحور . Z هذا المحور يمثل مسطحا يتجه نحو المشاهد ، و يمثل العمق ، القيم الإيجابية تُحرك العنصر في اتجاهكم و تجعله أقرب إليكم و السلبية تحركه بعيدا عنكم.





سنتعرف في هذا الموضوع على مجموعة من الخصائص الجديدة . قبل ذلك لنلقي نظرة على المتصفحات التي تدعم ذلك الاضافة الجديدة والموضحة بالشكل التالي :

رقم أول إصدار للمتصفحات التي بدأت تدعم الخاصيات					
					الخاصية
10.0	23.0 15.0 -webkit-	4.0 -webkit-	16.0 10.0 -moz-	36.0 12.0 -webkit-	transform
10.0	23.0 15.0 -webkit-	4.0 -webkit-	16.0 10.0 -moz-	36.0 12.0 -webkit-	transform-origin (بتلات قيم)
11.0	23.0 15.0 -webkit-	4.0 -webkit-	16.0 10.0 -moz-	36.0 12.0 -webkit-	transform-style
10.0	23.0 15.0 -webkit-	4.0 -webkit-	16.0 10.0 -moz-	36.0 12.0 -webkit-	perspective
10.0	23.0 15.0 -webkit-	4.0 -webkit-	16.0 10.0 -moz-	36.0 12.0 -webkit-	perspective-origin
10.0	23.0 15.0 -webkit-	4.0 -webkit-	16.0 10.0 -moz-	36.0 12.0 -webkit-	backface-visibility

الدوال التي سنستعملها مع الخاصية transform ، هي تقريبا نفسها التي تعاملنا معها سابقا ، مع إضافة البعد الثالث:

- rotateX() .
- rotateY() .
- rotateZ() .
- scaleZ() .
- translateZ() .

## اولا : الدالة **rotateX(a)**

تتيح الدالة `rotateX()` دوران العنصر حول المحور السيني "X" بدون التأثير على حجمه . تحدد القيمة بدرجة زاوية الدوران . إذا كانت القيمة إيجابية يتم الدوران في اتجاه عقارب الساعة ، و العكس إذا كانت سلبية.  
مكان أو نقطة الدوران يتم تحديدها باستعمال الخاصية `transform-origin`

مثال على ذلك :

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <style>
    h3 {direction:rtl;}

    .box,.rotation{width:200px;height:200px;background:rgba(138,200,15,0.8)
;border:1px solid #79af03;}
    .box{border:1px dashed gray;background:#e9f1d7;}
    .rotation-x:hover {
      -webkit-transform:rotateX(140deg);
      -moz-transform: rotateX(140deg);
      transform: rotateX(140deg);
    }
  </style>
</head>
<body>
  <h1>CSS3 3D RotateX</h1>
  <div class="box">
    <div class="rotation rotation-x">rotateX(140deg) <h3> ضع مؤشر الفأرة
    X درجة حول محور 140 على العنصر و سيدور </h3> </div>
  </div>
</body>
</html>

```

داخل المستعرض ضع مؤشر الفأرة على العنصر و سيدور 140 درجة حول محور X

### CSS3 3D RotateX

```
rotateX(140deg)
```

ضع مؤشر الفأرة على العنصر و  
سيدور 140 درجة حول محور X

### ثانيا : الدالة rotateY(a)

تتيح الدالة rotateY() دوران العنصر حول المحور الصادي "Y" بدون التأثير على حجمه . تحدد القيمة بدرجة زاوية الدوران . إذا كانت القيمة ايجابية يتم الدوران في اتجاه عقارب الساعة ، و العكس إذا كانت سلبية.  
مكان أو نقطة الدوران يتم تحديدها باستعمال الخاصية transform-origin

مثال على ذلك

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <style>
    h3 {direction:rtl;}

    .box,.rotation{width:200px;height:200px;background:rgba(138,200,15,0.8)
;border:1px solid #79af03;}
    .box{border:1px dashed gray;background:#e9f1d7;}
    .rotation-y:hover {
      -webkit-transform:rotateY(140deg);
      -moz-transform: rotateY(140deg);
      transform: rotateY(140deg);
    }
  </style>
</head>
<body>
  <h1>CSS3 3D RotateY</h1>
  <div class="box">
    <div class="rotation rotation-y">rotateY(140deg) <h3> ضع مؤشر الفأرة
    </h3> </div>
    </div>
  </body>
</html>

```

### CSS3 3D RotateY

```
rotateY(140deg)
```

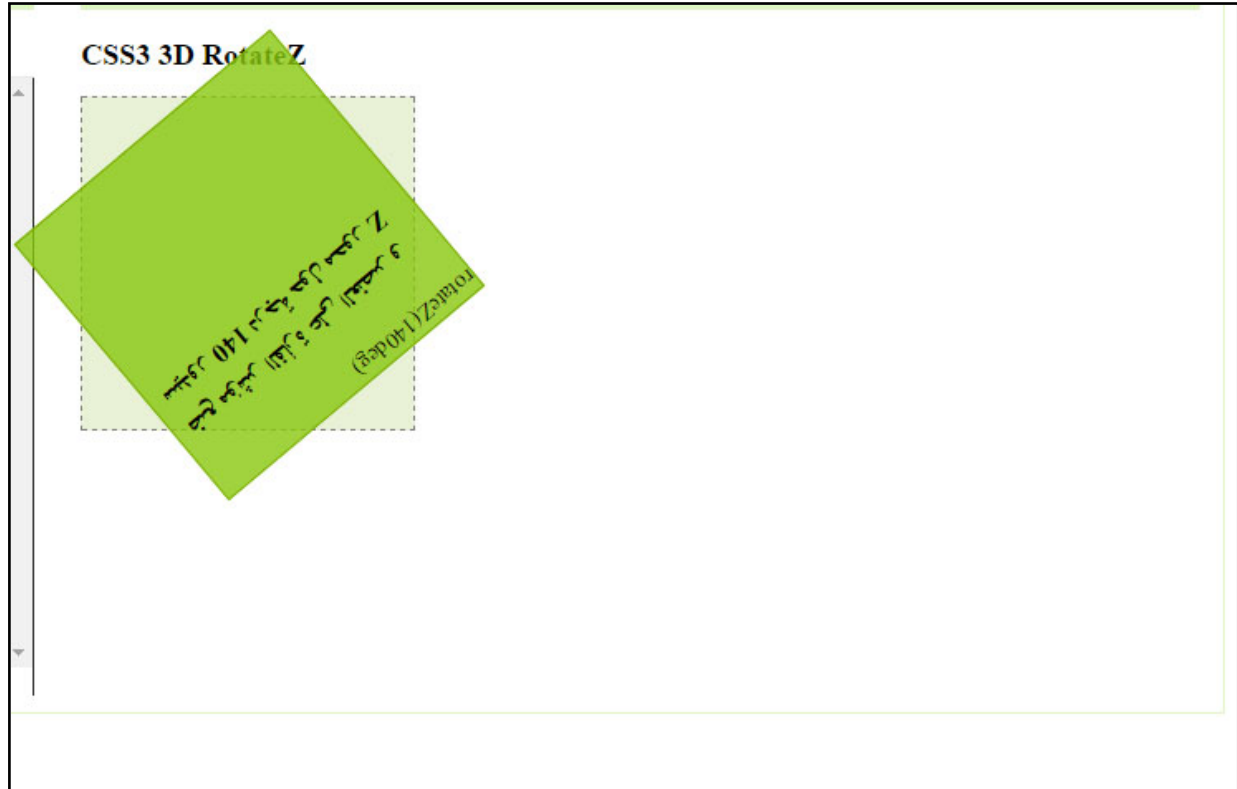
ضع مؤشر الفأرة على العنصر و  
سيدور 140 درجة حول محور Y

### ثالثا : الدالة rotateZ(a)

تتيح الدالة rotateZ() دوران العنصر حول محور "Z" بدون التأثير على حجمه .  
تحدد القيمة بدرجة زاوية الدوران . إذا كانت القيمة إيجابية يتم الدوران في اتجاه عقارب الساعة ، و العكس إذا كانت سلبية.  
مكان أو نقطة الدوران يتم تحديدها باستعمال الخاصية transform-origin

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <style>
    h3 {direction:rtl;}

    .box,.rotation{width:200px;height:200px;background:rgba(138,200,15,0.8)
;border:1px solid #79af03;}
    .box{border:1px dashed gray;background:#e9f1d7;}
    .rotation-z:hover {
      -webkit-transform:rotateZ(140deg);
      -moz-transform: rotateZ(140deg);
      transform: rotateZ(140deg);
    }
  </style>
</head>
<body>
  <h1>CSS3 3D RotateZ</h1>
  <div class="box">
    <div class="rotation rotation-z">rotateZ(140deg) <h3> ضع مؤشر الفأرة
    Z درجة حول محور 140 على العنصر و سيدور </h3> </div>
  </div>
</body>
</html>
```



### رابعاً : $rotate3d(x,y,z,a)$

تتيح الدالة  $rotate3d()$  دوران العنصر حول محور مُحدّد بدون التأثير على حجمه . تحدد القيمة بدرجة زاوية الدوران . إذا كانت القيمة إيجابية يتم الدوران في اتجاه عقارب الساعة ، و العكس إذا كانت سلبية.

في فضاء ثلاثي الأبعاد ، يمكن تحديد محور الدوران بالإعتماد على قيم المتجهات  $a[x,y,z]$  ، نقطة الدوران يتم تحديدها باستعمال الخاصية  $transform-origin$

x : تُحدّد إحداثيات x للمتجه الذي يمثل محور الدوران

y : تُحدّد إحداثيات y للمتجه الذي يمثل محور الدوران



z : تُحدّد إحداثيات z للمتجه الذي يمثل محور الدوران  
a : تمثل درجة زاوية الدوران.

مثلا :

rotate3d(1,0,0,80deg) اختزال rotateX(80deg)

rotate3d(0,1,0,80deg) اختزال rotateY(80deg)

rotate3d(0,0,1,80deg) اختزال rotateZ(80deg)

للجمع بين الدوال الثلاث في دالة rotate3d() يجب استعمال المصفوفات ، للذين  
لم يدرسوا المصفوفات في مادة الرياضيات ، يمكنكم نسيان الأمر و ترك هذه الدالة  
جانبا ،

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <style>

.box,.rotation{width:200px;height:200px;background:rgba(138,200,15,0.8)
;border:1px solid #79af03;}
  .box{border:1px dashed gray;background:#e9f1d7;}
  .rotation-3d {
    -webkit-transform:rotate3d(0.7, 0.5, 0.7, 80deg);
    -moz-transform: rotate3d(0.7, 0.5, 0.7, 80deg);
    transform: rotate3d(0.7, 0.5, 0.7, 80deg);
  }
</style>
</head>
<body>
  <h1>CSS3 Rotate3d</h1>
  <div class="box">
    <div class="rotation rotation-3d"><h3>rotate3d(0.7, 0.5, 0.7,
80deg)</h3></div>
  </div>
</body>
</html>
```

## CSS3 Rotate3d



## خامسا : الدالة translate3d()

translateX , translateY , translateZ تقوم كل دالة بتحريك العنصر وفق محورها الخاص . بالنسبة للدالة translateZ لن يظهر تأثيرها على متصفحكم لأننا نشتغل في فضاء ثنائي البعد . يمكننا جمع بين هذه الدوال في دالة واحدة ، هي translate3d(x,y,z)

```
.translate-3d {  
-webkit-transform: translate3d(20px, 30px, 10px);  
-moz-transform: translate3d(20px, 30px, 10px);  
transform: translate3d(20px, 30px, 10px);  
}
```

## الدالة scale3d()

رأينا في الدرس السابق أن الدالتين scaleX و scaleY تؤثران على حجم العنصر ، بتكبيره أو تصغيره ، مثلا إذا كان لدينا عنصر بعرض 100px و طبقنا عليه scaleX(2) سيصبح بعرض 200px. بالنسبة للدالة scaleZ ، تعطينا تصرفا مختلفا نوعا ما ، ما دمنا نشتغل في فضاء ثنائي الأبعاد ليس له أي عمق ، رغم أنها دالة ثلاثية الأبعاد إلا أنها تتصرف تقريبا مثل translateZ . إذ تظهر لنا كأنها تقوم بتحريك العنصر على محور Z . يمكننا جمع قيم scale في دالة مختصرة و هي scale3d . المثال أسفله نفس التأثير سيطبق على العنصرين div1 و div2

```
.div1 {  
transform: scaleX(2) scaleY(1.5) scaleZ(0.75);  
}  
  
.div2 {  
transform: scale3d(2,1.5,0.75);  
}
```

## CSS3 Perspective

لتفعيل الفضاء الثلاثي الأبعاد ، يحتاج العنصر إلى منظور perspective في فضاء ثلاثي الأبعاد ، تقوم perspective بالتأثير على المسافة التي تفصل المشاهد بالعنصر الذي يراه . أي بين مُسطح المستوى  $z=00$  و المشاهد . كلما كانت المسافة أقرب إلا و ازدادت التأثيرات البصرية . و كلما ابتعد المشاهد عن مسطح  $z$  إلا وكانت النتائج خادعة . مثلا أي قيمة أدنى من 2000 بيكسل ستعطي المشاهد تأثيرات جد بارزة ، أما إذا بالغنا في رفع القيمة ، مثلا 900 بيكسل ستبدو النتائج أقل تأثيرا .

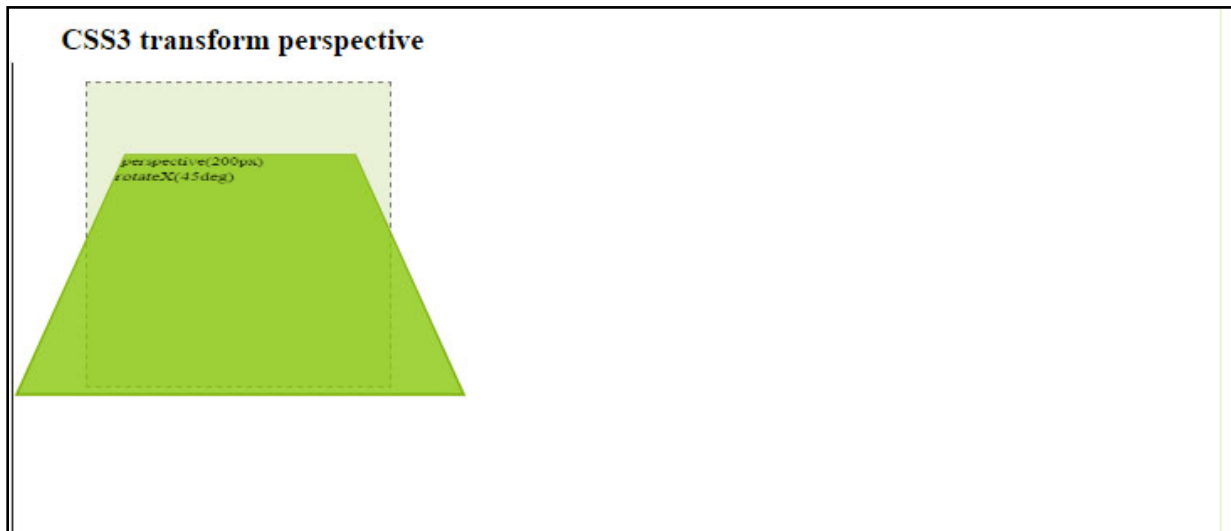
يمكننا إضافة perspective بطريقتين : إما استعمالها كدالة مع

الخاصية: transform

مثال ...

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <style>

.box,.rotation{ width:200px;height:200px;background:rgba(138,200
,15,0.8);border:1px solid #79af03;}
  .box{border:1px dashed gray;background:#e9f1d7;margin-
left:15px;}
  .rotation-x {
    -webkit-transform:perspective(200px) rotateX(45deg);
    -moz-transform:  perspective(200px) rotateX(45deg);
    transform:      perspective(200px) rotateX(45deg);
  }
</style>
</head>
<body>
  <h1>CSS3 transform perspective</h1>
  <div class="box">
    <div class="rotation rotation-x">perspective(200px)
rotateX(45deg)</div>
  </div>
</body>
</html>
```



أو استعمالها كخاصية منفردة

```
.parent {  
  perspective: 200px;  
}  
.child {  
  transform: rotateX(45deg);  
}
```

متى يجب أن نستعمل الدالة `perspective()` و متى يجب استعمال

الخاصية `perspective` ؟

يوجد فرق بين الحالتين ، عندما نريد التأثير على عنصر واحد فقط ، نستعمل

الدالة `perspective()` مع الخاصية `transform`

أما إذا أردنا تطبيق نفس التأثير على مجموعة من العناصر نستعمل

الخاصية perspective. لأن هذه الأخيرة ، أصلا لا تؤثر على الوسم الذي نطبقها عليه ، بل تؤثر على الوسوم الأبناء المتواجدين داخل هذا الوسم الأب  
لنأخذ المثال التالي:

```
<div class="parent">  
  <div class="child"> 1 العنصر الابن 1</div>  
  <div class="child"> 2 العنصر الابن 2</div>  
  <div class="child"> 3 العنصر الابن 3</div>  
  <div class="child"> 4 العنصر الابن 4</div>  
</div>
```

```
.parent {  
  -webkit-perspective:500px;  
  -moz-perspective: 500px;  
  perspective:      500px;  
}  
  
.child {  
  width:100px;height:100px;  
  background:rgba(138,200,15,0.8);  
  border:1px solid #79af03;  
  display:inline-block;  
  -webkit-transform:rotateX(30deg);  
  -moz-transform: rotateX(30deg);  
  transform:      rotateX(30deg);  
}
```



جميع الوسوم الأبناء "child" ستأخذ نفس التأثير ، و سيتم رؤيتها من نقطة موحدة  
كما تبين الصورة أسفله :



أما إذا طبقنا على نفس الوسوم الدالة perspective() ، فكل وسم سيتم التأثير عليه  
بشكل منفرد ، و ستتم رؤيته من المسافة المتعلقة به ، كما تبين الصورة أسفله:

```
.child {  
  /* ... */  
  -webkit-transform:perspective(500px) rotateX(30deg);  
  -moz-transform: perspective(500px) rotateX(30deg);  
  transform: perspective(500px) rotateX(30deg);  
}
```



## CSS3 perspective-origin

بالنسبة للخاصية perspective يمكننا تغيير نقطة المنظور (نقطة التلاشي) .  
مبدئياً توجد هذه النقطة وسط العنصر الأب ، وتأخذ قيمة center: أو 50% كما  
رأينا سابقاً في الخاصية transform-origin.

```
.parent {  
  -webkit-perspective:500px;  
  -moz-perspective: 500px;  
  perspective:      500px;  
  
  -moz-perspective-origin: 25% 75%;  
  -webkit-perspective-origin: 25% 75%;  
  perspective-origin:      25% 75%;  
}
```

## CSS3 transform-style

الخاصية transform-style ، تحدد ما إذا كان الوسم الإبن سيتموضع في فضاء  
ثلاثي الأبعاد ، أم سيتم تسطيحه في فضاء الوسم الأب .

تأخذ الخاصية ثلاث قيم: flat|preserve-3d|inherit

flat : سيتم تموضع الوسم الإبن في مسطح الوسم الأب

preserve-3d : سيتم تموضع الوسم الإبن في فضاء ثلاثي الأبعاد

```
div {
  -webkit-transform: rotateY(45deg);
  -moz-transform: rotateY(45deg);
  transform: rotateY(45deg);

  -webkit-transform-style: preserve-3d;
  -moz-transform-style: preserve-3d;
  transform-style: preserve-3d;
}
```

### CSS3 backface-visibility

تُحدد هذه الخاصية ما إذا كان يجب إخفاء أو إظهار الوجه الخلفي للعنصر أثناء دورانه.

تأخذ الخاصية القيم التالية : visible|hidden|initial|inherit

```
.box1 {
  -moz-transform: rotateY(180deg);
  -webkit-transform: rotateY(180deg);
  transform: rotateY(180deg);

  -moz-backface-visibility: hidden;
  -webkit-backface-visibility: hidden;
  backface-visibility: hidden;
}
```

وبهذا نكون قد انتهينا من عاشر مكون او اضافة جديدة فى لغة الـ CSS3

## 11 : التاثيرات الانتقالية CSS3 transitions

transition هي إحدى الإضافات الجديدة في اللغة CSS3 ، تتيح لنا التحكم في الفترة الإنتقالية أثناء تغيير أنماط عنصر ما دون اللجوء إلى استعمال الفلاش أو جافاسكريبت . عادة عندما نُغير مثلا لون عنصر أثناء مرور مؤشر الفأرة فوقه "hover" ، فإننا ننتقل بصفة حادة و فورية من النمط الأصلي إلى النمط النهائي . باستعمال transition يمكننا التحكم في وقت بداية الفترة الإنتقالية و سرعتها و مدتها ، عبر مجموعة من الخاصيات ، أبدأ بسردها . ثم سنتعرف عليها عبر أمثلة .  
اولا المتصفحات التي تدعم هذه الخاصية الجديدة هي موضحة في الشكل التالي:

رقم أول إصدار للمتصفحات التي بدأت تدعم الخاصيات					
					الخاصية
10.0	12.1 10.5 -o-	6.1 3.1 -webkit-	16.0 4.0 -moz-	26.0 4.0 -webkit-	transition
10.0	12.1 10.5 -o-	6.1 3.1 -webkit-	16.0 4.0 -moz-	26.0 4.0 -webkit-	transition-delay
10.0	12.1 10.5 -o-	6.1 3.1 -webkit-	16.0 4.0 -moz-	26.0 4.0 -webkit-	transition-duration
10.0	12.1 10.5 -o-	6.1 3.1 -webkit-	16.0 4.0 -moz-	26.0 4.0 -webkit-	transition-property
10.0	12.1 10.5 -o-	6.1 3.1 -webkit-	16.0 4.0 -moz-	26.0 4.0 -webkit-	transition-timing-function

مثال توضيحي للتعرف على الخاصية

```
<!DOCTYPE html>

<html>

<head>

<style>

div {

    width: 100px;

    height: 100px;

    background: red;

    -webkit-transition: width 3s; /* safari(3.1 => 6.1) , chrome(4.0 => < 26.0)*/

    -moz-transition: width 3s; /* firefox(4.0 => < 16.0) */

    -o-transition: width 3s; /* opera(10.5 => 12.0) */

    transition: width 3s; /* الخاصية الأصلية */

}

div:hover {

    width: 300px;

}

</style>

</head>

<body>

<p><b>Note:</b> This example does not work in Internet Explorer 9 and earlier

versions.</p>

<div></div>

<p>Hover over the div element above, to see the transition effect.</p>

</body>
```

**Note:** This example does not work in Internet Explorer 9 and earlier versions.



Hover over the div element above, to see the transition effect.

خاصية Transitions تعنى التأثيرات التي تحدث أثناء الفترة الإنتقالية من نمط إلى آخر لنفس العنصر . لتفعيل هذه التأثيرات ، يجب على الأقل تحديد شيئين أساسيين:

خاصية CSS التي سنطبق عليها التأثير مثل (width, height, color, transform ...)

ثم المدة الزمنية التي سيستغرقها التأثير (مدة الفترة الإنتقالية)

لنأخذ المثال أسفله . سنطبق التأثير على الخاصية "width" و ستستغرق المرحلة الأنتقالية 3 ثواني.

لنقوم بتطبيق التأثير على أكثر من خاصية

لتطبيق التأثير على مجموعة من الخاصيات ، نُفرِّق بينها بعلامة " , "

في المثال أسفله سنطبق التأثير على العناصر التالية :

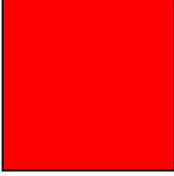
width,height,background-color,transform

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <style>
    .box {
      border: 1px solid black;
      width: 100px;
      height: 100px;
      background-color: red;
      -webkit-transition:width 2s, height 2s, background-color 2s, -
webkit-transform 2s;
      transition:width 2s, height 2s, background-color 2s, transform
2s;
    }
    .box:hover {
      width:200px;
      height:200px;
      background-color: yellow;
      -webkit-transform:rotate(180deg);
      transform:rotate(180deg);
    }
  </style>
</head>
<body>
  <h1>CSS3 transition</h1>
  <h2>ضع مؤشر الفأرة على العنصر ليبدأ التأثير</h2>
  <div class="box"></div>
  <p style="color:green;"><b>عندما نُبعد مؤشر الفأرة عن العنصر ، يعود  
لنمطه الأصلي</b></p>
</body>
</html>
```



## CSS3 transition

ضع مؤشر الفأرة على العنصر ليبدأ التأثير



عندما تُبعد مؤشر الفأرة عن العنصر ، يعود لنمطه الأصلي

## خاصيات transition المختصرة

من أجل التحكم السليم في التأثيرات الإنتقالية ، يمكننا استعمال مجموعة من الخاصيات المختصرة:

transition-property : تحديد إسم أو أسماء الخاصيات التي سيشملها التأثير

transition-duration : تحديد الوقت الذي سيستغرقه التأثير

transition-timing-function : تحديد منحنى السرعة للتأثير الإنتقالي

transition-delay : تحدد وقت بداية التأثير

## transition-property

بدل كتابة الخاصيات التي سيشملها التأثير كقيم للخاصية . transition و لأسباب تنظيمية و تسهيل الإستعمال ، من الأفضل القيام بسرد هذه الخاصيات كقيم للخاصية . transition-property مثلا إذا أردنا تطبيق التأثير على الطول و العرض و اللون ، نستعملها كالتالي:

```
.box{ transition-property : width, height, color; }
```

## transition-duration

الخاصية transition-duration ، تهتم بتحديد الوقت الذي سيستغرقه التأثير . يمكننا تحديد مدة زمنية واحدة لكل الخاصيات التي سيشملها التأثير ، في المثال أسفله ، سيستغرق التأثير ثانيتين لكل من الطول و العرض .

```
.box {  
  transition-property: width, height;  
  transition-duration: 2s;  
}
```

أو تحديد مدة زمنية مختلفة لكل خاصية على حدة ، في المثال أسفله سيستغرق التأثير ثانيتين للعرض و أربع ثواني للطول

```
.box {  
  transition-property: width, height;  
  transition-duration: 2s, 4s;  
}
```

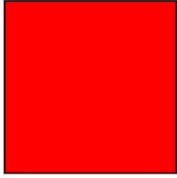
## مثالا للخاصيتين:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<style>
.box {
width: 100px;
height: 100px;
background-color: red;
-webkit-transition-property: width, height, background-color;
transition-property: width, height, background-color;
-webkit-transition-duration: 2s, 3s, 4s;
transition-duration: 2s, 3s, 4s;
}

.box:hover {
width: 250px;
height: 250px;
background-color: green;
}
</style>
</head>
<body>
<h1>CSS3 transition duration</h1>
<h2>ضع مؤشر الفأرة على العنصر ليبدأ التأثير</h2>
<div class="box"></div>
<p style="color:green;"><b>عندما نُبعد مؤشر الفأرة عن العنصر ، يعود لنمطه </b></p>
</b></p>
</body>
</html>
```

## CSS3 transition

ضع مؤشر الفأرة على العنصر ليبدأ التأثير



عندما تُبعد مؤشر الفأرة عن العنصر ، يعود لنمطه الأصلي

## *transition-delay*

تُحدد الوقت الذي يجب انتظاره قبل بدأ تفعيل التأثير

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<style>
.box {
  width: 100px;
  height: 100px;
  background-color: red;
  -webkit-transition-property: width;
  -webkit-transition-duration: 3s;
  -webkit-transition-delay: 2s;
  transition-property: width;
  transition-duration: 3s;
  transition-delay: 2s;
}

.box:hover {
  width: 300px;
}
</style>
</head>
<body>
<h1>CSS3 transition delay</h1>
<h2>عند وضع مؤشر الفأرة على العنصر ستنتظر ثانيتين قبل أن يبدأ التأثير</h2>
<div class="box"></div>
</body>
</html>
```

عند وضع مؤشر الفأرة على العنصر ستنتظر ثانيتين قبل أن يبدأ التأثير

CSS3 transition delay

عند وضع مؤشر الفأرة على العنصر ستنتظر ثانيتين قبل أن يبدأ التأثير



## transition-timing-function

```
.box {  
  -webkit-transition-timing-function: linear;  
  transition-timing-function: linear;  
}
```

تتيح لنا الخاصية transition-timing-function اللعب على سرعة مراحل التأثير الإنتقالية ، مثلا يمكننا بدأ التأثير بسرعة بطيئة و إنهائه بطريقة أسرع ، أو الحفاظ على نفس السرعة خلال كل وقت التأثير ... إلخ .

تأخذ هذه الخاصية مجموعة من القيم :

cubic-bezier() |linear |ease |ease-in |ease-out |ease-in-out |initial  
|inherit

شرح هذه القيم موضح في الشكل التوضيحي التالي :

قيم الخاصية transition-timing-function	
الشرح	القيمة
مكعب بيزي ، يمكننا تحديد 4 قيم للسرعة . كل قيمة تتراوح بين 0 و 1	cubic-bezier(n,n,n,n)
يحافظ التأثير على نفس السرعة من البداية إلى نهاية المرحلة الإنتقالية هذه القيمة تعادل مكعب بيزي <code>cubic-bezier(0,0,1,1)</code>	linear
القيمة الإقتراضية : يبدأ التأثير ببطء ثم يزداد سرعة ثم ينتهي ببطء هذه القيمة تعادل مكعب بيزي <code>cubic-bezier(0.25,0.1,0.25,1)</code>	ease
يبدأ التأثير ببطء هذه القيمة تعادل مكعب بيزي <code>cubic-bezier(0.42,0,1,1)</code>	ease-in
ينتهي التأثير ببطء هذه القيمة تعادل مكعب بيزي <code>cubic-bezier(0,0,0.58,1)</code>	ease-out
يبدأ التأثير ببطء و ينتهي ببطء هذه القيمة تعادل مكعب بيزي <code>cubic-bezier(0.42,0,0.58,1)</code>	ease-in-out

```
.box1 {  
  -webkit-transition-timing-function: linear;  
  transition-timing-function: linear;  
}  
.box2 {  
  -webkit-transition-timing-function: ease;  
  transition-timing-function: ease;  
}  
.box3 {  
  -webkit-transition-timing-function: ease-in;  
  transition-timing-function: ease-in;  
}  
.box4 {  
  -webkit-transition-timing-function: ease-out;  
  transition-timing-function: ease-out;  
}  
.box5 {  
  -webkit-transition-timing-function: ease-in-out;  
  transition-timing-function: ease-in-out;  
}
```



```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<style>
h4 { text-align:center;color:#fff;}
div {
width: 100px;
height: 50px;
background-color: red;
-webkit-transition-property: width;
transition-property: width;
-webkit-transition-duration: 2s;
transition-duration: 2s;
}
div:hover { width: 300px;}
.box1 {
-webkit-transition-timing-function: linear;
transition-timing-function: linear;
}
.box2 {
-webkit-transition-timing-function: ease;
transition-timing-function: ease;
}
.box3 {
-webkit-transition-timing-function: ease-in;
transition-timing-function: ease-in;
}
.box4 {
-webkit-transition-timing-function: ease-out;
transition-timing-function: ease-out;
}
.box5 {
-webkit-transition-timing-function: ease-in-out;
transition-timing-function: ease-in-out;
}
</style>
</head>
<body>
```

يظهر في المستعرض كما يلي مع ملاحظة ضع مؤشر الفأرة على العنصر ليبدأ التأثير

### CSS3 transition-timing-function

ضع مؤشر الفأرة على العنصر ليبدأ التأثير

- linear
- ease
- ease-in
- ease-out
- ease-in-out



وبهذا نكون قد انتهينا من مكون او اضافة جديدة في لغة الـ CSS3

## 12 : الحركة - التأثيرات الحركية - CSS3 Animations

تم اضافة خاصية جديدة فى الاصدار الجديد وهى Animations او الحركة . قبل خاصية animation الجديدة كان يتم انشاء الحركات بتحريك عناصر HTML باستخدام JavaScript أو الصور المتحركة مثل GIF أو استخدام استخدام إضافات مثل Adobe Flash. مع إصدار CSS3 أصبح بإمكاننا الإستغناء عن سكريبتات الفلاش و جافاسكريبت لتنشيط حركة العناصر أثناء المرحلة الإنتقالية من نمط إلى آخر .

ولإنشاء سلسلة متحركة لعنصر ما ، نحتاج إلى خاصيتين أساسيتين و هما "@keyframes" التي تحدد شكل أنماط البداية و النهاية و كذلك الأنماط البينية للمرحلتين . ثم خاصية "animation" التي تصف الحركة في حد ذاتها و تدرج منها مجموعة من الخاصيات الداعمة ، سأعرضها عليكم أولا ثم سنتعرف عليها تدريجيا عبر أمثلة.

وكما تعودنا قبل شرح الاضافة الجديدة يتم توضيح المستعرضات التي تدعم هذه الخاصية الجديدة والموضح فى الشكل التالى :

Property					
@keyframes	43.0 4.0 -webkit-	10.0	16.0 5.0 -moz-	9.0 4.0 -webkit-	30.0 15.0 -webkit- 12.0 -o-
animation	43.0 4.0 -webkit-	10.0	16.0 5.0 -moz-	9.0 4.0 -webkit-	30.0 15.0 -webkit- 12.0 -o-

## الخاصية Animation

لإنشاء العنصر الذي سيتم تنشيطه ، نستعمل الخاصية animation و نحدد لها قيمتين على الأقل : إسم العنصر الذي سيتم تنشيطه ، و الوقت الذي سيستغرقه التنشيط كما يلي :

```
div {  
  animation: mytest 5s;  
}
```

القيمة الأولى "mytest" هي إسم العنصر الذي سيتم تنشيطه ، هي قيمة افتراضية يمكنكم تسميتها كما شئتم . أما القيمة الثانية " 5s" تُحدّد المدة الزمنية التي سيستغرقها التنشيط .

بهذا أصبح العنصر div جاهزا للتنشيط . لتفعيل العملية يجب تمرير إسم العنصر الذي أنشأناه "mytest" للخاصية الثانية. @keyframes

```
@keyframes mytest {  
  /* ... */  
}
```

## الخاصية : @keyframes

داخل الخاصية @keyframes نقوم بتحديد شكل النمط الذي سيكون عليه العنصر في زمن ما أثناء التنشيط ، يمكننا تحديد نمط البداية و النهاية و كذلك الأنماط البيئية .

لتحديد نمط البداية و النهاية يمكننا استعمال مفتاحي " from , to " : و هي تعني "من ، إلى : "

داخل "from" سنقوم بتحديد أنماط البداية.

داخل "to" نحدد أنماط النهاية.

```
@keyframes mytest {  
  from {  
    background-color: blue;  
  }  
  to {  
    background-color: red;  
  }  
}
```

في المثال أسفله ، سنقوم بتنشيط العنصر المسمى "mytest" لمدة 6 ثواني . و سنقوم بتغيير لونه من الأزرق "لون البداية" إلى اللون الأحمر "لون النهاية"

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <style>
div {
  width:150px;
  height:150px;
  background-color:blue;
  -webkit-animation: mytest 6s;
  animation: mytest 6s;
}
@-webkit-keyframes mytest {
  from { background-color: blue; }
  to { background-color: red; }
}

@keyframes mytest {
  from { background-color: blue; }
  to { background-color: red; }
}
  </style>
</head>
<body>
  <div></div>
</body>
</html>
```



يمكننا تطبيق العدد الذي نريده من الأنماط داخل . @keyframes لنرى مثالا آخر .

سنقوم بتحريك وسم العنوان h1 من اليمين إلى اليسار و نقوم بتغيير لونه في الوقت نفسه. كما يلي :

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<style>
h1 {
  -webkit-animation: mytest 5s;
  animation: mytest 5s;
}

@-webkit-keyframes mytest {
  from {
    margin-left: 70%;
    color: red;
  }

  to {
    margin-left: 0%;
    color: green;
  }
}

@keyframes mytest {
  from {
    margin-left: 70%;
    color: red;
  }

  to {
    margin-left: 0%;
    color: green;
  }
}
</style>
</head>
<body>
  <h1>عنوان متحرك</h1>
</body>
</html>
```



عنوان متحرك

يمكننا أيضا استعمال مفاتيح النسب المؤية بدل "from ... to" ، و هذا سيساعدنا على تحديد أنماط بينية أخرى للعنصر بالإضافة إلى نمط البداية و النهاية. كما فى المثال التالى ::

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <style>
    div {
      width:150px;
      height:150px;
      background-color:blue;
      -webkit-animation: mytest 6s;
      animation: mytest 6s;
    }
    @-webkit-keyframes mytest {
      0% { background-color: blue;}
      25% { background-color: green; }
      50% { background-color: red; }
      75% { background-color: yellow; }
      100% { background-color: orange; }
    }

    @keyframes mytest {
      0% { background-color: blue;}
      25% { background-color: green; }
      50% { background-color: red; }
      75% { background-color: yellow; }
      100% { background-color: orange; }
    }
  </style>
</head>
<body>
  <div></div>
</body>
</html>
```

## animation الخاصيات المختصرة

لدينا مجموعة من الخاصيات المختصرة الداعمة و التي تسهل علينا التحكم في تنشيط العناصر . لكل واحدة دورها المنوط بها . رأينا منها في الأمثلة السابقة خاصيتين كنا قد استعملناهما كقيم للخاصية animation هما:

• animation-name : إسم العنصر الذي سيتم تنشيطه ، أطلقنا عليه

اعتباطيا إسم. "mytest"

• animation-duration : الوقت الذي ستستغرقه عملية التنشيط

```
div {  
  animation-name: mytest;  
  animation-duration: 6s;  
}
```

باقي الخاصيات المختصرة أعرضها عليكم ، ثم سندرسها عبر أمثلة:

- animation-delay : تحدد الوقت الذي يجب انتظاره قبل بدأ التنشيط
- animation-direction : تحدد ما إذا كان التنشيط سيلعب في الإتجاه المعاكس أو يعود إلى نقطة البداية أو يتم تكراره
- animation-iteration-count : تحدد عدد المرات التي سيتم فيها تكرار التنشيط
- animation-play-state : تمكنا من توقيف التنشيط أو تفعيله
- animation-timing-function : تحدد منحنى السرعة للتنشيط مثلا يبدأ بسرعة و يختتم ببطء
- animation-fill-mode : تحدد الأنماط التي سيأخذها العنصر عندما لا يكون نشيطا : " عند انتهاء التنشيط أو قبل بدئه"

### **الخاصية : animation-iteration-count**

الخاصية animation-iteration-count تحدد عدد المرات التي سيتم فيها إعادة لعب التنشيط . و نعطيها القيم التالية | initial | infinite | number : inherit

## القيمة **number** : تحديد عدد المرات التي سيتم فيها إعادة لعب التنشيط

مثال

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<style>
div {
width: 50px;
height: 50px;
position: relative;
background: red;
-webkit-animation: moving 4s;
-moz-animation: moving 4s;
animation: moving 4s;
-webkit-animation-iteration-count: 3;
-moz-animation-iteration-count: 3;
animation-iteration-count: 3;
}

@-webkit-keyframes moving {
from {margin-left: 0%;}
to {margin-left: 90%;}
}

@keyframes moving {
from {margin-left: 0%;}
to {margin-left: 90%;}
}
</style>
</head>
<body>
<h1>animation-iteration-count : number</h1>
<div></div>
</body>
</html>
```

**animation-iteration-count : number**



## القيمة infinite : يتحرك بواسطتها العنصر بصفة مستمرة ، دون توقف

مثال ...

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<style>
div {
width: 50px;
height: 50px;
position: relative;
background: red;
-webkit-animation: moving 4s;
-moz-animation: moving 4s;
animation: moving 4s;
-webkit-animation-iteration-count: infinite;
-moz-animation-iteration-count: infinite;
animation-iteration-count: infinite;
}

@-webkit-keyframes moving {
from {margin-left: 0%;}
to {margin-left: 90%;}
}

@keyframes moving {
from {margin-left: 0%;}
to {margin-left: 90%;}
}
</style>
</head>
<body>
<h1>animation-iteration-count : infinite</h1>
<div></div>
</body>
</html>
```

## الخاصية animation-direction

الخاصية animation-direction تحدد ما إذا كان التنشيط سيلعب في الإتجاه العكسي أو وفق دورات متكررة . و تأخذ القيم التالية:

normal| reverse| alternate| alternate-reverse| initial| inherit

وسنقوم بشرح هذه القيم ..

**القيمة normal : القيمة الافتراضية ، يتحرك التنشيط بشكل طبيعي**

مثال ...

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<style>
div {
width: 50px;
height: 50px;
background-color:red;
-webkit-animation: moving 5s;
animation: moving 5s;
-webkit-animation-iteration-count: infinite;
animation-iteration-count: infinite;
-webkit-animation-direction: normal;
animation-direction: normal;
}
@-webkit-keyframes moving {
from { margin-left: 90%;}
to { margin-left: 0%;}
}
@keyframes moving {
from { margin-left: 90%;}
to { margin-left:0%;}
}
</style>
</head>
<body>
<h1>animation-direction : normal</h1>
<div></div>
</body>
```



## القيمة reverse : يلعب التنشيط بشكل عكسي

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<style>
div {
  width: 50px;
  height: 50px;
  background-color:red;
  -webkit-animation: moving 5s;
  animation: moving 5s;
  -webkit-animation-iteration-count: infinite;
  animation-iteration-count: infinite;
  -webkit-animation-direction: reverse;
  animation-direction: reverse;
}
@-webkit-keyframes moving {
  from { margin-left: 90%;}
  to { margin-left: 0%;}
}
@keyframes moving {
  from { margin-left: 90%;}
  to { margin-left: 0%;}
}
</style>
</head>
<body>
  <h1>animation-direction : reverse</h1>
  <div></div>
</body>
</html>
```

## القيمة alternate : يتم تغيير اتجاه التنشيط بعد انتهاء كل دورة

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<style>
div {
width: 50px;
height: 50px;
background-color:red;
-webkit-animation: moving 5s;
animation: moving 5s;
-webkit-animation-iteration-count: infinite;
animation-iteration-count: infinite;
-webkit-animation-direction: alternate;
animation-direction: alternate;
}
@-webkit-keyframes moving {
from { margin-left:90%;}
to { margin-left:0%;}
}
@keyframes moving {
from { margin-left:90%;}
to { margin-left:0%;}
}
</style>
</head>
<body>
<h1>animation-direction : alternate</h1>
<div></div>
</body>
</html>
```

## : القيمة alternate-reverse

يتم تغيير اتجاه التنشيط بعد انتهاء كل دورة عكس اتجاهه alternate

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<style>
div {
width:50px;
height:50px;
background-color:red;
-webkit-animation: moving 5s;
animation: moving 5s;
-webkit-animation-iteration-count: infinite;
animation-iteration-count: infinite;
-webkit-animation-direction: alternate-reverse;
animation-direction: alternate-reverse;
}
@-webkit-keyframes moving {
from { margin-left:90%;}
to { margin-left:0%;}
}
@keyframes moving {
from { margin-left:90%;}
to { margin-left:0%;}
}
</style>
</head>
<body>
<h1>animation-direction : alternate-reverse</h1>
<div></div>
</body>
</html>
```

## الخاصية : animation-timing-function

تحدد الخاصية animation-timing-function منحنى السرعة لتنشيط العنصر ، أي سرعة الانتقال من نمط لآخر بطريقة سلسلة. يمكننا استعمال منحنى "cubic bezier مكعب بييزي" ، أو استعمال القيم المعرفة مسبقا :

linear|ease|ease-in|ease-out|cubic-bezier(n,n,n,n)|initial|inherit

وسوف نقوم بشرح بعض القيم ...

**القيمة cubic-bezier(n,n,n,n)** : مكعب بييزي ، يمكننا تحديد 4 قيم للسرعة . كل قيمة تتراوح بين 0 و 1

**القيمة linear** : يحافظ التنشيط على نفس السرعة من البداية إلى نهاية الدورة  
هذه القيمة تعادل مكعب بييزي(cubic-bezier(0,0,1,1))

**القيمة ease** : القيمة الافتراضية : يبدأ التنشيط ببطء ثم يزداد سرعة ثم ينتهي ببطء  
هذه القيمة تعادل مكعب بييزي(cubic-bezier(0.25,0.1,0.25,1))

**القيمة ease-in** : يبدأ التنشيط ببطء . هذه القيمة تعادل مكعب بييزي-cubic-bezier(0.42,0,1,1)

**القيمة ease-out** : ينتهي التنشيط ببطء هذه القيمة تعادل مكعب بييزي-cubic-bezier(0,0,0.58,1)

المثالين أسفله سيعطيان نفس النتيجة

## باستعمال القيم المُعرّفة

```
.box1 { animation-timing-function: linear; }  
.box2 { animation-timing-function: ease; }  
.box3 { animation-timing-function: ease-in; }  
.box4 { animation-timing-function: ease-out; }  
.box5 { animation-timing-function: ease-in-out; }
```

## إستعمال مكعب بيّزي

```
.box1 { animation-timing-function: cubic-bezier(0,0,0.25,1); }  
.box2 { animation-timing-function: cubic-bezier(0.25,0.1,0.25,1); }  
.box3 { animation-timing-function: cubic-bezier(0.42,0,1,1); }  
.box4 { animation-timing-function: cubic-bezier(0,0,0.58,1); }  
.box5 { animation-timing-function: cubic-bezier(0.42,0,0.58,1); }
```

مشاهدة النتيجة بكتابة المثال بالكامل

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<style>
h4 { text-align:center;color:#fff;}
div {
width: 100px;
height: 50px;
position: relative;
background-color: red;
-webkit-animation: moving 4s;
animation: moving 4s;
-webkit-animation-iteration-count: infinite;
animation-iteration-count: infinite;
}

.box1 {
-webkit-animation-timing-function: cubic-bezier(0,0,0.25,1);
animation-timing-function: cubic-bezier(0,0,0.25,1);
}
.box2 {
-webkit-animation-timing-function: cubic-bezier(0.25,0.1,0.25,1);
animation-timing-function: cubic-bezier(0.25,0.1,0.25,1);
}
.box3 {
-webkit-animation-timing-function: cubic-bezier(0.42,0,1,1);
animation-timing-function: cubic-bezier(0.42,0,1,1);
}
.box4 {
-webkit-animation-timing-function: cubic-bezier(0,0,0.58,1);
animation-timing-function: cubic-bezier(0,0,0.58,1);
}
.box5 {
-webkit-animation-timing-function: cubic-bezier(0.42,0,0.58,1);
animation-timing-function: cubic-bezier(0.42,0,0.58,1);
}

@-webkit-keyframes moving {
from {margin-left: 0%;}
to {margin-left: 80%;}
}
@keyframes moving {
from {margin-left: 0%;}
to {margin-left: 80%;}
}
</style>
</head>
<body>
<h1>animation-timing-function </h1>
<div class="box1"><h4>linear</h4></div>
<div class="box2"><h4>ease</h4></div>
<div class="box3"><h4>ease-in</h4></div>
<div class="box4"><h4>ease-out</h4></div>
<div class="box5"><h4>ease-in-out</h4></div>
</body>
</html>
```

## الخاصية animation-play-state

مُكّننا الخاصية animation-play-state من توقيف التنشيط أو تفعيله . يمكننا إعطاءها قيمتين :

running : القيمة الافتراضية ، تقوم بتشغيل التنشيط

paused : تقوم بتوقيف التنشيط

```
.box {  
  -webkit-animation-play-state:paused;  
  animation-play-state:paused;  
}
```

## الخاصية : animation-delay

تُحدّد الخاصية animation-delay المدة الزمنية التي يجب انتظارها قبل بدء التنشيط

```
.box {  
  -webkit-animation-delay: 3s;  
  animation-delay: 3s;  
}
```

مثال يجمع أغلب الخاصيات التي رأيناها ...

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<style>
div {
width: 70px;
height: 70px;
background: black;
position: relative;
-webkit-animation: moving 5s ease-in 2s infinite alternate;
animation: moving 5s ease-in 2s infinite alternate;
}

@-webkit-keyframes moving {
0% {background:black; left:0; top:0;}
25% {background:red; left:80%; top:0;}
50% {background:green; left:80%; top:250px;}
75% {background:orange; left:0; top:250px;}
100% {background:blue; left:0; top:0;}
}

@keyframes moving {
0% {background:black; left:0; top:0;}
25% {background:red; left:80%; top:0;}
50% {background:green; left:80%; :top:250px;}
75% {background:orange; left:0; top:250px;}
100% {background:blue; left:0; top:0;}
}
</style>
</head>
<body>
<div></div>
</body>
</html>
```



مثال اخر ...

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<style>
.circle {
width: 100px;
height: 100px;
position:relative;
margin-right: 20px;
border-radius: 50%;
background: -webkit-linear-gradient(orange, yellow);
background: -moz-linear-gradient(orange, yellow);
background: -o-linear-gradient(orange, yellow);
background: linear-gradient(orange, yellow);

-webkit-animation: ball 2s ease-in-out 1s infinite alternate;
animation: ball 2s ease-in-out 1s infinite alternate;
}

@-webkit-keyframes ball {
from { transform: translateX(0) rotate(0);}
to { transform: translateX(450px) rotate(3turn);}
}
@keyframes ball {
from { transform: translateX(0) rotate(0);}
to { transform: translateX(450px) rotate(3turn);}
}
</style>
</head>
<body>
<div class="circle"></div>
</body>
</html>
```

**وبهذا نكون قد انتهينا من مكون او اضافة جديدة فى لغة الـ CSS3**

## 13 : تنسيق الصور CSS3 Images

### اولا : Rounded Images

لعمل حواف دائرية للصور يتم ذلك من خلال استخدام `border-radius` والذي سبق وتم شرحها

مثال ...

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
  border-radius: 8px;
}
</style>
</head>
<body>
<h2>Rounded Images</h2>
<p>Use the border-radius property to create rounded images:</p>
<br>

<br>
</body>
</html>
```

## Rounded Images

Use the border-radius property to create rounded images:



مثال اخر ...

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
img {
```

```
    border-radius: 50%;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h2>Rounded Images</h2>
```

```
<p>Use the border-radius property to create circled images:</p>
```

```

```

```
</body>
```

```
</html>
```

Result Size: 705 x 645

## Rounded Images

Use the border-radius property to create circled images:



## ثانيا : Thumbnail Images

لجعل الصور تاخذ شكل Thumbnail يتم ذلك باستخدام خصائص الاطار - `border property` - الذى سبق وتم شرحها

مثال ...

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
  border: 1px solid #ddd;
  border-radius: 4px;
  padding: 5px;
  width: 150px;
}
</style>
</head>
<body>

<h2>Thumbnail Images</h2>
<p>Use the border property to create thumbnail images:</p>



</body>
</html>
```

## Thumbnail Images

Use the border property to create thumbnail images:



مثال اخر لعمل Thumbnail للصور بالاضافة الى روابط

```
<!DOCTYPE html>

<html>

<head>

<style>

img {

    border: 1px solid #ddd;

    border-radius: 4px;

    padding: 5px;

    width: 150px;

}

img:hover {

    box-shadow: 0 0 2px 1px rgba(0, 140, 186, 0.5);

}

</style>

</head>

<body>

<h2>Thumbnail Image as Link</h2>

<p>Use the border property to create thumbnail images. Wrap an anchor around the image to use it as a link.</p>

<p>Hover over the image and click on it to see the effect.</p>

<a target="_blank" href="paris.jpg">

</a>

</body>






</html>
```

## 14: تنسيق العنصر باستخدام خاصية CSS The object-fit Property

خاصية object-fit Property تعنى ملء العنصر لمساحة محددة . اى ان العنصر سوف يشغل مساحة محددة .  
وهذه الخاصية يدعمها المتصفحات التالية :

### Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
object-fit	31.0	16.0	36.0	7.1	19.0

فمثلا لو لدى عنصر الصورة ابعادها 200 px 400 px هذه الصورة ممكن ان تحتل نفس الابعاد او اقوم بتحديد مساحة جديدة تشغلها وبالتالي يتغير مساحة الصورة الظاهرة

المثال التالى يوضح هذه الخاصية :

سوف اقوم بادراج صورة بدون استخدام خاصية object-fit كما يلى :



```
<!DOCTYPE html>
<html>
<head>
<style>
img {
  width:200px;
  height:400px;
}
</style>
</head>
<body>

<h2>Image</h2>


</body>
</html>
```

فتظهر الصورة في المستعرض كما يلي :

## Image



تظهر الصورة بابعادها ومقاستها الطبيعية

بفرض انى سوف استخدم خاصية object-fit وتحديد القيمة cover كما فى

المثال التالى :

```
<!DOCTYPE html>

<html>

<head>

<style>

img {

  width:200px;

  height:400px;

  object-fit:cover;

}

</style>

</head>

<body>

<h2>The object-fit Property</h2>



</body>

</html>
```

فتظهر الصمورة بملء هذة المساحة وسوف يتم قطع جزء من الصورة فتظهر فى  
المستعرض كما يلى :

**Image**



```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>Not Using object-fit</h2>
```

```
<p>Here we use do not use "object-fit", so when we resize the browser window, the aspect ratio of the images will be destroyed:</p>
```

```
<div style="width:100%;height:400px;">
```

```

```

```

```

```
</div>
```

```
</body>
```

```
</html>
```

فتظهر الصور في المستعرض كما يلي :

## Not Using object-fit

Here we use do not use "object-fit", so when we resize the browser window, the aspect ratio of the images will be destroyed:



سوف نعيد نفس المثال ولكن باستخدام خاصية object-fit وتحديد القيمة cover  
كما يلي :

```
<!DOCTYPE html>

<html>

<body>

<h2>Using object-fit</h2>

<p>Here we use "object-fit: cover;", so when we resize the browser window, the aspect ratio of the images is preserved:</p>

<div style="width:100%;height:400px;">

</div>

</body>

</html>
```

فتظهر الصور كما يلي في المستعرض :

## Using object-fit

Here we use "object-fit: cover;", so when we resize the browser window, the aspect ratio of the images is preserved:





الخاصية object-fit لها بعض القيم منها كما يلي :

```
.fill {object-fit: fill;}  
.contain {object-fit: contain;}  
.cover {object-fit: cover;}  
.scale-down {object-fit: scale-down;}  
.none {object-fit: none;}
```

والمثال التالي يوضح الفرق بين كل قيمة :

```
<!DOCTYPE html>

<html>

<head>

<style>

.fill {object-fit: fill;}

.contain {object-fit: contain;}

.cover {object-fit: cover;}

.scale-down {object-fit: scale-down;}

.none {object-fit: none;}

</style>

</head>

<body>

<h1>The object-fit Property</h1>

<h2>No object-fit:</h2>



<h2>object-fit: fill (this is default):</h2>



<h2>object-fit: contain:</h2>



<h2>object-fit: cover:</h2>



<h2>object-fit: scale-down:</h2>



<h2>object-fit: none:</h2>



</body>

</html>
```

## The object-fit Property

No object-fit:



object-fit: fill (this is default):



**object-fit: contain:**



**object-fit: cover:**

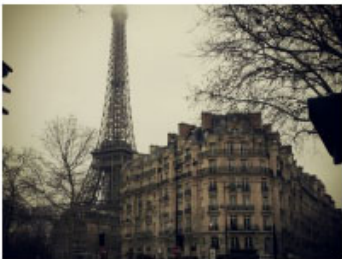


**object-fit: scale-down:**

**object-fit: cover:**



**object-fit: scale-down:**



**object-fit: none:**



## 14: تنسيق الازرار CSS Buttons

لتنسيق الازرار باستخدام اللغة يكتب الكود التالي الموضح فى المثال التالى :

```
<!DOCTYPE html>
<html>
<head>
<style>
.button {
    background-color: #4CAF50;
    border: none;
    color: white;
    padding: 15px 32px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 16px;
    margin: 4px 2px;
    cursor: pointer;
}
</style>
</head>
<body>
<h2>CSS Buttons</h2>
<button>Default Button</button>
<a href="#" class="button">Link Button</a>
<button class="button">Button</button>
<input type="button" class="button" value="Input Button">
```

وتظهر الازرار فى المستعرض كما يلى

## CSS Buttons

Default Button

Link Button

Button

Input Button



● للتعديل في الازار يستخدم الخاصية التالية background-color

مثال على تعديل الالوان :

```
<!DOCTYPE html>

<html>

<head>

<style>

.button {

    background-color: #4CAF50; /* Green */

    border: none;

    color: white;

    padding: 15px 32px;

    text-align: center;

    text-decoration: none;

    display: inline-block;

    font-size: 16px;

    margin: 4px 2px;

    cursor: pointer;

}

.button2 {background-color: #008CBA;} /* Blue */

.button3 {background-color: #f44336;} /* Red */

.button4 {background-color: #e7e7e7; color: black;} /* Gray */

.button5 {background-color: #555555;} /* Black */

</style>

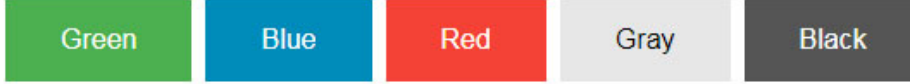
</head>

<body>
```

ويظهر في المستعرض كما يلي

## Button Colors

Change the background color of a button with the background-color property:



● للتعديل وتغيير حجم الازرار يستخدم الخاصية Button Sizes كما يلي :

```
<!DOCTYPE html>

<html>

<head>

<style>

.button {

    background-color: #4CAF50; /* Green */

    border: none;

    color: white;

    padding: 15px 32px;

    text-align: center;

    text-decoration: none;

    display: inline-block;

    margin: 4px 2px;

    cursor: pointer;

}

.button1 {font-size: 10px;}

.button2 {font-size: 12px;}

.button3 {font-size: 16px;}

.button4 {font-size: 20px;}

.button5 {font-size: 24px;}

</style>

</head>

<body>

<h2>Button Sizes</h2>

<p>Change the font size of a button with the font-size property:</p>
```

ويظهر في السمعرض كما يلي

## Button Sizes

Change the font size of a button with the font-size property:



● للتحكم فى المسافة بين كل زرار واخر يتم باستخدام خاصية **padding** كما يلى



```
<!DOCTYPE html>
<html>
<head>
<style>
.button {
    background-color: #4CAF50; /* Green */
    border: none;
    color: white;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 16px;
    margin: 4px 2px;
    cursor: pointer;
}

.button1 {padding: 10px 24px;}
.button2 {padding: 12px 28px;}
.button3 {padding: 14px 40px;}
.button4 {padding: 32px 16px;}
.button5 {padding: 16px;}
</style>
</head>
<body>
```

## Button Sizes

Change the padding of a button with the padding property:

10px 24px

12px 28px

14px 40px

32px 16px

16px

- لعمل ازرار ذات حواف دائرية Rounded Buttons يتم ذلك باستخدام خاصية **border-radius** كما يلي

```
<!DOCTYPE html>

<html>

<head>

<style>

.button {

    background-color: #4CAF50; /* Green */

    border: none;

    color: white;

    padding: 20px;

    text-align: center;

    text-decoration: none;

    display: inline-block;

    font-size: 16px;

    margin: 4px 2px;

    cursor: pointer;

}

.button1 {border-radius: 2px;}

.button2 {border-radius: 4px;}

.button3 {border-radius: 8px;}

.button4 {border-radius: 12px;}

.button5 {border-radius: 50%;}

</style>

</head>

<body>
```

## Rounded Buttons

Add rounded corners to a button with the border-radius property:





● لعمل اطار للازرار Colored Button Borders يتم ذلك باستخدام

خاصية **border** كما يلي

```
<!DOCTYPE html>

<html>

<head>

<style>

.button {

    background-color: #4CAF50; /* Green */

    border: none;

    color: white;

    padding: 15px 32px;

    text-align: center;

    text-decoration: none;

    display: inline-block;

    font-size: 16px;

    margin: 4px 2px;

    cursor: pointer;

}

.button1 {

    background-color: white;

    color: black;

    border: 2px solid #4CAF50;

}

.button2 {

    background-color: white;

    color: black;

    border: 2px solid #008CBA;
```

## Colored Button Borders

Use the border property to add a border to the button:

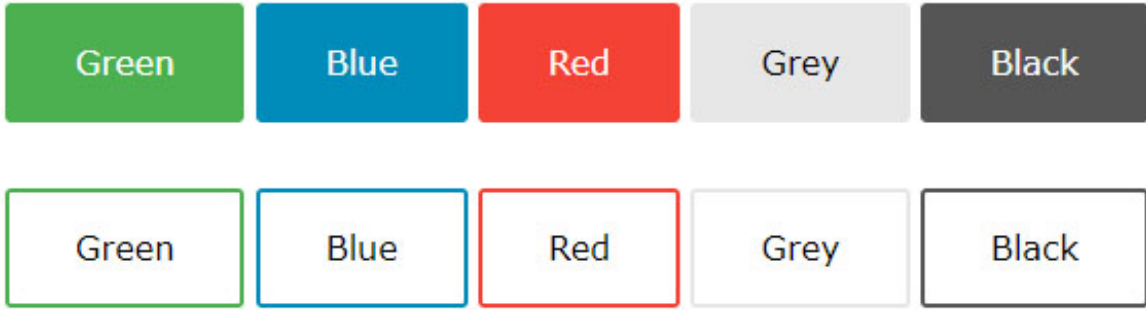


## لعمل تأثير على الازرار عند مرور الماوس عليها Hoverable Buttons

يتم ذلك باستخدام خاصية ال `hover`: كما يلي:

```
<!DOCTYPE html>
<html>
<head>
<style>
.button {
  background-color: #4CAF50; /* Green */
  border: none;
  color: white;
  padding: 16px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  margin: 4px 2px;
  -webkit-transition-duration: 0.4s; /* Safari */
  transition-duration: 0.4s;
  cursor: pointer;
}
.button1 {
  background-color: white;
  color: black;
  border: 2px solid #4CAF50;
}
.button1:hover {
  background-color: #4CAF50;
```

## Hoverable Buttons



- لعمل ظل على الازرار Shadow Buttons يتم ذلك باستخدام خاصية `box-shadow` كما يلي

```
<!DOCTYPE html>

<html>

<head>

<style>

.button {

    background-color: #4CAF50; /* Green */

    border: none;

    color: white;

    padding: 15px 32px;

    text-align: center;

    text-decoration: none;

    display: inline-block;

    font-size: 16px;

    margin: 4px 2px;

    cursor: pointer;

    -webkit-transition-duration: 0.4s; /* Safari */

    transition-duration: 0.4s;

}

.button1 {

    box-shadow: 0 8px 16px 0 rgba(0,0,0,0.2), 0 6px 20px 0 rgba(0,0,0,0.19);

}

.button2:hover {

    box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24),0 17px 50px 0 rgba(0,0,0,0.19);

}

</style>

</head>

<body>

<h2>Shadow Buttons</h2>

<p>Use the box-shadow property to add shadows to the button:</p>
```

## Shadow Buttons

Use the box-shadow property to add shadows to the button:



للتحكم في شفافية الازرار Disabled Buttons يتم ذلك من خلال خاصية *opacity* كما يلي

```
<!DOCTYPE html>
<html>
<head>
<style>
.button {
```



## Disabled Buttons

Use the opacity property to add some transparency to the button (make it look disabled):

Normal Button

Disabled Button



للتحكم في حجم الازرار Button Width يتم ذلك باستخدام خاصية **width** كما يلي :

```
<!DOCTYPE html>

<html>

<head>

<style>

.button {

    background-color: #4CAF50; /* Green */

    border: none;

    color: white;

    padding: 15px 32px;

    text-align: center;

    text-decoration: none;

    display: inline-block;

    font-size: 16px;

    margin: 4px 2px;

    cursor: pointer;

}

.button1 {width: 250px;}

.button2 {width: 50%;}

.button3 {width: 100%;}

</style>

</head>

<body>

<h2>Button Width</h2>

<p>Use the width property to change the width of the button:</p>

<p><strong>Tip:</strong> Use pixels if you want to set a fixed width and use percent for responsive buttons (e.g. 50% of its parent element). Resize the browser window to see the effect.</p>
```

## Button Width

Use the width property to change the width of the button:

**Tip:** Use pixels if you want to set a fixed width and use percent for responsive buttons (e.g. 50% of its parent element). Resize the browser window to see the effect.



250px



50%



100%

- لتحكم فى اتجاة الاورار وتعويمها يتم ذلك باستخدام خاصية float: كما يلى :

```
<!DOCTYPE html>

<html>

<head>

<style>

.btn-group .button {

    background-color: #4CAF50; /* Green */

    border: none;

    color: white;

    padding: 15px 32px;

    text-align: center;

    text-decoration: none;

    display: inline-block;

    font-size: 16px;

    cursor: pointer;

    float: left;

}

.btn-group .button:hover {

    background-color: #3e8e41;

}

</style>

</head>

<body>

<h2>Button Groups</h2>

<p>Remove margins and float the buttons to create a button group:</p>
```

## Button Groups

Remove margins and float the buttons to create a button group:



Remember to clear floats after, or else will this p element also float next to the buttons.

- لعمل اطار للازرار Bordered Button يتم ذلك باستخدام خاصية ال **border** كما يلي :

```
<!DOCTYPE html>
<html>
<head>
<style>
.btn-group .button {
  background-color: #4CAF50; /* Green */
  border: 1px solid green;
  color: white;
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  cursor: pointer;
  float: left;
}
.btn-group .button:not(:last-child) {
  border-right: none; /* Prevent double borders */
}
.btn-group .button:hover {
  background-color: #3e8e41;
}
</style>
</head>
<body>
```

## • لعمل زرار على الصورة Button on Image يتم ذلك باستخدام خاصية

```
<!DOCTYPE html>

<html>

<head>

<style>

.container {

    position: relative;

    width: 100%;

    max-width: 400px;

}

.container img {

    width: 100%;

    height: auto;

}

.container .btn {

    position: absolute;

    top: 50%;

    left: 50%;

    transform: translate(-50%, -50%);

    -ms-transform: translate(-50%, -50%);

    background-color: #f1f1f1;

    color: black;

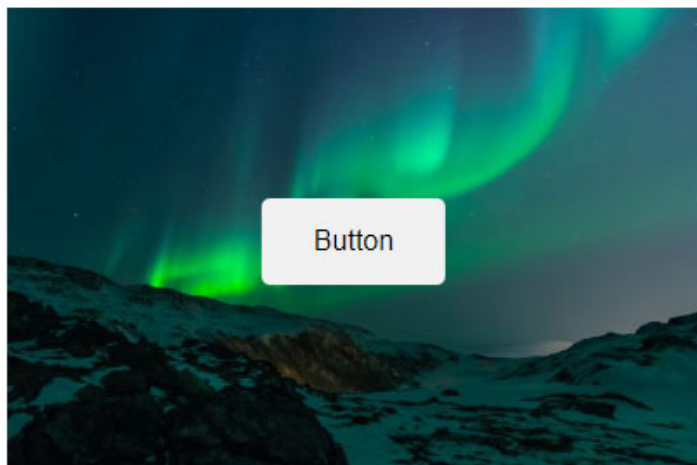
    font-size: 16px;

    padding: 16px 30px;

    border: none;
```

## Button on Image

Add a button to an image:



## Animated Buttons • لعمل زرار متحرك

```
<!DOCTYPE html>

<html>

<head>

<style>

.button {

  display: inline-block;

  border-radius: 4px;

  background-color: #f4511e;

  border: none;

  color: #FFFFFF;

  text-align: center;

  font-size: 28px;

  padding: 20px;

  width: 200px;

  transition: all 0.5s;

  cursor: pointer;

  margin: 5px;

}

.button span {

  cursor: pointer;

  display: inline-block;

  position: relative;

  transition: 0.5s;

}

.button span:after {

  content: '\00bb';
```



## لعمل زرار بتاثير ال click يتم ذلك كما يلي :

```
<!DOCTYPE html>

<html>

<head>

<style>

.button {

  display: inline-block;

  padding: 15px 25px;

  font-size: 24px;

  cursor: pointer;

  text-align: center;

  text-decoration: none;

  outline: none;

  color: #fff;

  background-color: #4CAF50;

  border: none;

  border-radius: 15px;

  box-shadow: 0 9px #999;

}

.button:hover {background-color: #3e8e41}

.button:active {

  background-color: #3e8e41;

  box-shadow: 0 5px #666;

  transform: translateY(4px);

}
```

## Animated Buttons - "Pressed Effect"



Click Me

## لعمل زرار ذات تاثير ال Fade in كما يلي :

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
.button {
background-color: #f4511e;
border: none;
color: white;
padding: 16px 32px;
text-align: center;
font-size: 16px;
margin: 4px 2px;
opacity: 0.6;
transition: 0.3s;
display: inline-block;
text-decoration: none;
cursor: pointer;
}
.button:hover {opacity: 1}
</style>
</head>
<body>
<h2>Fading Buttons - "Fade in Effect"</h2>
<button class="button">Hover Over Me</button>
</body>
```

## Fading Buttons - "Fade in Effect"

Hover Over Me

**object-fit: none:**



## 15 : عمل ترقيم للصفحات Pagination

### ● عمل ترقيم بسيط Simple Pagination

اذا كان لديك موقع الكترونى يحتوى على عدد كبير من الصفحات يمكنك عمل ترقيم له وذلك من خلال الكود التالى

```
<!DOCTYPE html>

<html>

<head>

<style>

.pagination {

    display: inline-block;

}

.pagination a {

    color: black;

    float: left;

    padding: 8px 16px;

    text-decoration: none;

}

</style>

</head>

<body>

<h2>Simple Pagination</h2>

<div class="pagination">

    <a href="#">&laquo;</a>

    <a href="#">1</a>

    <a href="#">2</a>

    <a href="#">3</a>

    <a href="#">4</a>

    <a href="#">5</a>

    <a href="#">6</a>

    <a href="#">&raquo;</a>

</div>

</body>

</html>
```

## Simple Pagination

« 1 2 3 4 5 6 »

● لعمل ترقيم ذات تاثير وذلك بالمرور بالماوس عليـة Active and

Hoverable Pagination



```
<!DOCTYPE html>

<html>

<head>

<style>

.pagination {

    display: inline-block;

}

.pagination a {

    color: black;

    float: left;

    padding: 8px 16px;

    text-decoration: none;

}

.pagination a.active {

    background-color: #4CAF50;

    color: white;

}

.pagination a:hover:not(.active) {background-color: #ddd;}

</style>

</head>

<body>

<h2>Active and Hoverable Pagination</h2>

<p>Move the mouse over the numbers.</p>

<div class="pagination">

    <a href="#">&laquo;</a>

    <a href="#">1</a>

    <a class="active" href="#">2</a>

    <a href="#">3</a>

    <a href="#">4</a>

    <a href="#">5</a>

    <a href="#">6</a>

    <a href="#">&raquo;</a>

</div>
```

## Active and Hoverable Pagination

Move the mouse over the numbers.

« 1 2 3 4 5 6 »

- لعمل ترقيم ذات تاثير وبحواف دائرية وذلك بالمرور بالماوس عليـة  
Rounded Active and Hoverable Buttons

```
<!DOCTYPE html>

<html>

<head>

<style>

.pagination {

    display: inline-block;

}

.pagination a {

    color: black;

    float: left;

    padding: 8px 16px;

    text-decoration: none;

}

.pagination a.active {

    background-color: #4CAF50;

    color: white;

    border-radius: 5px;

}

.pagination a:hover:not(.active) {

    background-color: #ddd;

    border-radius: 5px;

}

</style>

</head>

<body>

<h2>Rounded Active and Hover Buttons</h2>

<div class="pagination">

<a href="#">&laquo;</a>

<a href="#">1</a>
```

# Rounded Active and Hover Buttons

« 1 2 3 4 5 6 »

● لعمل ترقيم ذات اطار Bordered Pagination

```
<!DOCTYPE html>

<html>

<head>

<style>

.pagination {

    display: inline-block;

}

.pagination a {

    color: black;

    float: left;

    padding: 8px 16px;

    text-decoration: none;

    transition: background-color .3s;

    border: 1px solid #ddd;

}

.pagination a.active {

    background-color: #4CAF50;

    color: white;

    border: 1px solid #4CAF50;

}

.pagination a:hover:not(.active) {background-color: #ddd;}

</style>

</head>

<body>

<h2>Pagination with Borders</h2>

<div class="pagination">

<a href="#">&laquo;</a>

<a href="#">1</a>

<a href="#" class="active">2</a>

<a href="#">3</a>

<a href="#">4</a>

<a href="#">5</a>

<a href="#">6</a>

<a href="#">&raquo;</a>
```

## Pagination with Borders

«	1	2	3	4	5	6	»
---	---	---	---	---	---	---	---



## ● لعمل مسافة بين الترقيم Space Between Links

```
<!DOCTYPE html>

<html>

<head>

<style>

.pagination {

  display: inline-block;

}

.pagination a {

  color: black;

  float: left;

  padding: 8px 16px;

  text-decoration: none;

  transition: background-color .3s;

  border: 1px solid #ddd;

  margin: 0 4px;

}

.pagination a.active {

  background-color: #4CAF50;

  color: white;

  border: 1px solid #4CAF50;

}

.pagination a:hover:not(.active) {background-color: #ddd;}

</style>

</head>

<body>

<h2>Pagination with Margins</h2>

<div class="pagination">

  <a href="#">&laquo;</a>

  <a href="#">1</a>

  <a href="#" class="active">2</a>

  <a href="#">3</a>
```

## Pagination with Margins

« 1 2 3 4 5 6 »

## ● للتعديل فى حجم الترقيم Pagination Size

```
<!DOCTYPE html>

<html>

<head>

<style>

.pagination {

    display: inline-block;

}

.pagination a {

    color: black;

    float: left;

    padding: 8px 16px;

    text-decoration: none;

    transition: background-color .3s;

    border: 1px solid #ddd;

    font-size: 22px;

}

.pagination a.active {

    background-color: #4CAF50;

    color: white;

    border: 1px solid #4CAF50;

}

.pagination a:hover:not(.active) {background-color: #ddd;}

</style>

</head>

<body>

<h2>Pagination Size</h2>

<p>Change the font-size property to make the pagination smaller or bigger.</p>

<div class="pagination">

<a href="#">&laquo;</a>

<a href="#">1</a>

<a href="#" class="active">2</a>
```

## Pagination Size

Change the font-size property to make the pagination smaller or bigger.

«	1	2	3	4	5	6	»
---	---	---	---	---	---	---	---




## 16- لتقسم وتخطيط الصفحة الى عدد من الاعمدة CSS Multi-column Layout

يتم ذلك باستخدام الخاصية الجديدة `column` وهذا الخاصية يدعمها كل من المستعرضات التالي:

### Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit- or -moz- specify the first version that worked with a prefix.

Property					
column-count	50.0 4.0 -webkit-	10.0	52.0 2.0 -moz-	9.0 3.1 -webkit-	37.0 15.0 -webkit- 11.1
column-gap	50.0 4.0 -webkit-	10.0	52.0 2.0 -moz-	9.0 3.1 -webkit-	37.0 15.0 -webkit- 11.1
column-rule	50.0 4.0 -webkit-	10.0	52.0 2.0 -moz-	9.0 3.1 -webkit-	37.0 15.0 -webkit- 11.1
column-rule-color	50.0 4.0 -webkit-	10.0	52.0 2.0 -moz-	9.0 3.1 -webkit-	37.0 15.0 -webkit- 11.1
column-rule-style	50.0 4.0 -webkit-	10.0	52.0 2.0 -moz-	9.0 3.1 -webkit-	37.0 15.0 -webkit- 11.1

● انشاء عدد من الاعمدة CSS Create Multiple Columns

يتم ذلك باستخدام خاصية ال `column-count` كما هو موضح بالمثل التالي

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
.newspaper {
```

```
  -webkit-column-count: 3; /* Chrome, Safari, Opera */
```

```
  -moz-column-count: 3; /* Firefox */
```

```
  column-count: 3;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p><b>Note:</b> Internet Explorer 9, and earlier versions, does not support the column-count property.</p>
```

```
<div class="newspaper">
```

```
  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exercitation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duiis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum.
```

```
</div>
```

```
</body>
```

```
</html>
```

**Note:** Internet Explorer 9, and earlier versions, does not support the column-count property.

Lorem ipsum dolor sit amet,  
consectetur adipiscing elit, sed  
diam nonummy nibh euismod  
tincidunt ut laoreet dolore magna  
aliquam erat volutpat. Ut wisi  
enim ad minim veniam, quis  
nostrud exerci tation ullamcorper  
suscepit lobortis nisl ut aliquip ex

ea commodo consequat. Duis  
autem vel eum iriure dolor in  
hendrerit in vulputate velit esse  
molestie consequat, vel illum  
dolore eu feugiat nulla facilisis at  
vero eros et accumsan et iusto  
odio dignissim qui blandit  
praesent luptatum zzril delenit

augue duis dolore te feugait nulla  
facilisi. Nam liber tempor cum  
soluta nobis eleifend option  
congue nihil imperdiet doming id  
quod mazim placerat facer  
possim assum.

## • لعمل مسافة بين الاعمدة وتحديد قيمتها CSS Specify the Gap Between Columns

يتم ذلك باستخدام خاصية **column-gap** كما في المثال التالي :



```
<!DOCTYPE html>

<html>

<head>

<style>

.newspaper {

  -webkit-column-count: 3; /* Chrome, Safari, Opera */

  -moz-column-count: 3; /* Firefox */

  column-count: 3;

  -webkit-column-gap: 40px; /* Chrome, Safari, Opera */

  -moz-column-gap: 40px; /* Firefox */

  column-gap: 40px;

}

</style>

</head>

<body>

<p><b>Note:</b> Internet Explorer 9, and earlier versions, does not support the column-gap property.</p>

<div class="newspaper">

  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum.

</div>

</body>
```

**Note:** Internet Explorer 9, and earlier versions, does not support the column-gap property.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper

suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio

dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum.

## ● لعمل مسطرة او فواصل بين الاعمدة CSS Column Rules

يتم ذلك باستخدام خاصية ال `column-rule-style` كما يلي :

```
<!DOCTYPE html>

<html>

<head>

<style>

.newspaper {

  -webkit-column-count: 3; /* Chrome, Safari, Opera */

  -moz-column-count: 3; /* Firefox */

  column-count: 3;

  -webkit-column-gap: 40px; /* Chrome, Safari, Opera */

  -moz-column-gap: 40px; /* Firefox */

  column-gap: 40px;

  -webkit-column-rule-style: solid; /* Chrome, Safari, Opera */

  -moz-column-rule-style: solid; /* Firefox */

  column-rule-style: solid;

}

</style>

</head>

<body>

<p><b>Note:</b> Internet Explorer 9, and earlier versions, does not support the column-rule-style property.</p>

<div class="newspaper">

  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum.
```

**Note:** Internet Explorer 9, and earlier versions, does not support the column-rule-style property.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper

suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio

dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum.

• لتحديد سمك المسطرة او الفاصل `column-rule-width`

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
.newspaper {
```

```
  -webkit-column-count: 3; /* Chrome, Safari, Opera */
```

```
  -moz-column-count: 3; /* Firefox */
```

```
  column-count: 3;
```

```
  -webkit-column-gap: 40px; /* Chrome, Safari, Opera */
```

```
  -moz-column-gap: 40px; /* Firefox */
```

```
  column-gap: 40px;
```

```
  -webkit-column-rule-style: solid; /* Chrome, Safari, Opera */
```

```
  -moz-column-rule-style: solid; /* Firefox */
```

```
  column-rule-style: solid;
```

```
  -webkit-column-rule-width: 1px; /* Chrome, Safari, Opera */
```

```
  -moz-column-rule-width: 1px; /* Firefox */
```

```
  column-rule-width: 1px;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p><b>Note:</b> Internet Explorer 9, and earlier versions, does not support the column-rule-width property.</p>
```

```
<div class="newspaper">
```

**Note:** Internet Explorer 9, and earlier versions, does not support the column-rule-width property.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper

suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio

dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum.

• لتحديد لون المسطرة او الفاصل `column-rule-color`

```
<!DOCTYPE html>

<html>

<head>

<style>

.newspaper {

  -webkit-column-count: 3; /* Chrome, Safari, Opera */

  -moz-column-count: 3; /* Firefox */

  column-count: 3;

  -webkit-column-gap: 40px; /* Chrome, Safari, Opera */

  -moz-column-gap: 40px; /* Firefox */

  column-gap: 40px;

  -webkit-column-rule-style: solid; /* Chrome, Safari, Opera */

  -moz-column-rule-style: solid; /* Firefox */

  column-rule-style: solid;

  -webkit-column-rule-width: 1px; /* Chrome, Safari, Opera */

  -moz-column-rule-width: 1px; /* Firefox */

  column-rule-width: 1px;

  -webkit-column-rule-color: lightblue; /* Chrome, Safari, Opera */

  -moz-column-rule-color: lightblue; /* Firefox */

  column-rule-color: lightblue;

}

</style>

</head>

<body>
```

**Note:** Internet Explorer 9, and earlier versions, does not support the column-rule-color property.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper

suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio

dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum.

● لتحديد سمك العمود Column Width



```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
.newspaper {
```

```
  -webkit-column-count: 3; /* Chrome, Safari, Opera */
```

```
  -moz-column-count: 3; /* Firefox */
```

```
  column-count: 3;
```

```
  -webkit-column-width: 100px; /* Chrome, Safari, Opera */
```

```
  -moz-column-width: 100px; /* Firefox */
```

```
  column-width: 100px;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p><b>Note:</b> Internet Explorer 9, and earlier versions, does not support the column-width property.</p>
```

```
<div class="newspaper">
```

```
  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duiis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum.
```

```
</div>
```

```
</body>
```

**Note:** Internet Explorer 9, and earlier versions, does not support the column-width property.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex

ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit

augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum.






## التحكم في واجهة المستخدم CSS User Interface

هذه الخاصية يدعمها المستعرضات التالية :

### Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit- or -moz- specify the first version that worked with a prefix.

Property					
resize	4.0	Not supported	5.0 4.0 -moz-	4.0	15.0
outline-offset	4.0	Not supported	5.0 4.0 -moz-	4.0	9.5

### ● تغيير حجم العنصر CSS Resizing

يتم ذلك باستخدام خاصية ال `resize` كما يلي :

```
<!DOCTYPE html>

<html>

<head>

<style>

div {

    border: 2px solid;

    padding: 20px;

    width: 300px;

    resize: horizontal;

    overflow: auto;

}

</style>

</head>

<body>

<h1>The resize Property</h1>

<div>

<p>Let the user resize only the width of this div element.</p>

<p>To resize: Click and drag the bottom right corner of this div element.</p>

</div>

<p><b>Note:</b> Internet Explorer does not support the resize property.</p>

</body>

</html>
```

## The resize Property

Let the user resize only the width of this div element.

To resize: Click and drag the bottom right corner of this div element.

**Note:** Internet Explorer does not support the resize property.

مثال اخر

```
<!DOCTYPE html>

<html>

<head>

<style>

div {

    border: 2px solid;

    padding: 20px;

    width: 300px;

    resize: vertical;

    overflow: auto;

}

</style>

</head>

<body>

<h1>The resize Property</h1>

<div>

<p>Let the user resize only the height of this div element.</p>

<p>To resize: Click and drag the bottom right corner of this div element.</p>

</div>

<p><b>Note:</b> Internet Explorer does not support the resize property.</p>

</body>

</html>
```

## The resize Property

Let the user resize only the height of this div element.

To resize: Click and drag the bottom right corner of this div element.

**Note:** Internet Explorer does not support the resize property.

مثال اخر

```
<!DOCTYPE html>

<html>

<head>

<style>

div {

    border: 2px solid;

    padding: 20px;

    width: 300px;

    resize: both;

    overflow: auto;

}

</style>

</head>

<body>

<h1>The resize Property</h1>

<div>

<p>Let the user resize both the height and the width of this div element.</p>

<p>To resize: Click and drag the bottom right corner of this div element.</p>

</div>

<p><b>Note:</b> Internet Explorer does not support the resize property.</p>

</body>

</html>
```



## The resize Property

Let the user resize both the height and the width of this div element.

To resize: Click and drag the bottom right corner of this div element.

**Note:** Internet Explorer does not support the resize property.

```
<!DOCTYPE html>

<html>

<head>

<style>

div.ex1 {

    margin: 20px;

    border: 1px solid black;

    outline: 4px solid red;

    outline-offset: 15px;

}

div.ex2 {

    margin: 10px;

    border: 1px solid black;

    outline: 5px dashed blue;

    outline-offset: 5px;

}

</style>

</head>

<body>

<h1>The outline-offset Property</h1>

<div class="ex1">This div has a 4 pixels solid red outline 15 pixels outside the border edge.</div>

<br>
```

## The outline-offset Property

This div has a 4 pixels solid red outline 15 pixels outside the border edge.

This div has a 5 pixels dashed blue outline 5 pixels outside the border edge.

## ● لعمل مربع او صندوق CSS Box Sizing

```
<!DOCTYPE html>

<html>

<head>

<style>

.div1 {

    width: 300px;

    height: 100px;

    border: 1px solid blue;

}

.div2 {

    width: 300px;

    height: 100px;

    padding: 50px;

    border: 1px solid red;

}

</style>

</head>

<body>

<div class="div1">This div is smaller (width is 300px and height is 100px).</div>

<br>

<div class="div2">This div is bigger (width is also 300px and height is 100px).</div>

</body>
```

This div is smaller (width is 300px and height is 100px).

This div is bigger (width is also 300px and height is 100px).

مثال آخر

```
<!DOCTYPE html>

<html>

<head>

<style>

.div1 {

  width: 300px;

  height: 100px;

  border: 1px solid blue;

  box-sizing: border-box;

}

.div2 {

  width: 300px;

  height: 100px;

  padding: 50px;

  border: 1px solid red;

  box-sizing: border-box;

}

</style>

</head>

<body>

<div class="div1">Both divs are the same size now!</div>

<br>

<div class="div2">Hooray!</div>
```

Both divs are the same size now!

Hooray!

```
<!DOCTYPE html>

<html>

<head>

<style>

body {

    margin: 0;

}

* {

    box-sizing: border-box;

}

input, textarea {

    width: 100%;

}

</style>

</head>

<body>

<form action="/action_page.php">

First name:<br>

<input type="text" name="firstname" value="Mickey"><br>

Last name:<br>

<input type="text" name="lastname" value="Mouse"><br>

Comments:<br>
```



First name:

Mickey

Last name:

Mouse

Comments:

Submit

**Tip:** Try to remove the box-sizing property from the style element and look what happens. Notice that the width of input, textarea, and submit button will go outside of the screen.

وبهذا نكون قد انتهينا من شرح هذه الوحدات المضافة او الاضافات الجديدة فى

لغة الـ CSS3