

تلخيص كورس CSS3

م/أسامة الزiero



كورس CSS3

بسم الله الرحمن الرحيم والصلوة والسلام على رسول الله

Vendor prefixes

البريفكس التي ينبغي استخدامها مع خصائص CSS3

```
-webkit-box-shadow: // Google Chrome, Safari, IOS, Android  
-moz-box-shadow: // Firefox  
-o-box-shadow: // Opera  
-ms-box-shadow: // Internet Explorer
```

موقع يعرفك المتصفح يدعم أكواد html5 و Css3

<https://caniuse.com/#home>

Icon Explanations



Supported by Edge/Internet Explorer



Supported by Firefox



Supported by Google Chrome



Supported by Safari



Supported by Opera



Supported by Edge/Internet Explorer with prefix -ms-



Supported by Firefox with prefix -moz-



Supported by Google Chrome with prefix -webkit-



Supported by Safari with prefix -webkit-



Supported by Opera with prefix -webkit-

الحوار المستدير : border-radius

تحدد الخاصية **border-radius** نصف قطر زوايا العنصر.

نصيحة: تتيح لك هذه الخاصية إضافة حدود مستديرة للعناصر! يمكن أن تحتوي هذه الخاصية من واحد إلى أربع قيم. فيما يلي القواعد:

Four values - border-radius: 15px 50px 30px 5px;

وتكون القيم هكذا :

first value top-left أعلى يسار

second value top-right أعلى يمين

third value bottom-right تحت يمين

fourth value bottom-left تحت يسار

تنطبق القيمة الأولى على الزاوية العلوية اليسرى ، والقيمة الثانية تنطبق على الزاوية العلوية اليمنى ، والقيمة الثالثة تنطبق على الركن السفلي الأيسر ، والقيمة الرابعة تنطبق على الركن السفلي الأيسر

ملاحظة: بستطيع التحكم بها بهذا طريقة سواء بالبيكسل أو بالنسبة المئوية

نصف قطر زوايا العنصر border-radius تحدد الخاصية

نصيحة: تتيح لك هذه الخاصية إضافة حدود مستديرة للعناصر

```
<style>
#rcorners {
    border-radius: 10px 90px 90px 10px;
    background: #73AD21;
    padding: 20px;
    width: 500px;
    height: 100px;
}
</style>
</head>
<body>
<p>.نصف قطر زوايا العنصر border-radius تحدد الخاصية</p>
<div id="rcorners"></div>
<!--نصيحة: تتيح لك هذه الخاصية إضافة حدود مستديرة للعناصر-->
```

: Box Shadow

تقوم خاصية **box-shadow** بإرفاق ظلال أو أكثر لعنصر.

Default value: none

Mلاحظة : هناك قيمتين مطلوبتين ولا بد من وضعهما حتى يعمل **shadow**

[vertical Shadow + Horizontal Shadow]

القيمة الأفقية والقيمة العمودية وهما قيمتان يقبلان القيمة السالبة

مثال على عمل **shadow** باستخدام القيمة الأفقية والقيمة العمودية

[vertical Shadow + Horizontal Shadow + color]

```
<style>
    *{direction: rtl}
#example1 {
    border: 1px solid;
    padding: 10px;
    box-shadow: 5px 10px #888888;}
</style>
</head>
<body>
<h2> بستخدام القيمة الأفقية والقيمة العمودية مثال على عمل shadow </h2>
<div id="example1">
<p>[ vertical Shadow + Horizontal Shadow + color ]</p>
</div>
```

Shadow Syntax

box-shadow: **none** | **h-offset v-offset blur spread color** | **inset** | **initial** | **inherit**;

و هذه هي قيم **shadow** المستخدمة بالترتيب

blur : ضباب الظل ولا يقبل قيم بالسالب

Spread : انتشار الظل

Inset : وهو انتشار الظل داخل box

: انتشار الظل خارج box وهو الطبيعي outset

h-offset v-offset blur spread color

مثال على استخدام 5 قيم لل shadow

مثال

box-shadow: 5px 10px 8px 10px #888888;



لعمل أكثر من shadow ملاحظة: لإرافق أكثر من ظل واحد بعنصر ، أضف قائمة ظلال مفصولة بفواصل (مثال "جرّبها بنفسك" أدناه).

**box-shadow: -5px -5px 8px #F00,
5px 5px 8px #00F,
10px 10px 8px #080**

: Box sizing

تسمح لنا خاصية **box-sizing** CSS بتضمين الحشو والحدود بإجمالي العرض والارتفاع للعنصر.
يعرف الخاصية **box-sizing** كيف يتم حساب عرض عنصر الارتفاع، ينبغي أن تشمل الحشو والحدود ، أو لا .

يعني : هل تأخذ العرض صافي لها حتى في حالة إضافة border

CSS Syntax

box-sizing: content-box|border-box|initial|inherit;

. تتضمن خصائص العرض والارتفاع default : **content-box**

(وخصائص min / max) المحتوى فقط. لا يتم تضمين الحدود والخشوة

: فهي تحكم العنصر ولا تسمح له بالخروج عن width الذي تم تحديده له

مثال تم استخدام **content-box** انظر لم يؤثر في المحتوى وخرج div عن المحتوى



```
<style>
div.container {width: 100%;border: 2px solid black;}
div.box0 {box-sizing: content-box;
    width: 50%;
    border: 5px solid red;
    float: left;}
div.box {box-sizing: content-box;
    width: 50%;
    border: 5px solid red;
    float: right;}
</style>
</head>
<body>
<div class="container">
    <div class="box0">This div occupies the left half</div>
    <div class="box">This div occupies the right half</div>
    <div style="clear:both;"></div>
</div>
```

مثال تم استخدام **border-box** انظر تم التأثير في ال div ولم يخرج عن المحتوى



```
<style>
div.container {width: 100%;border: 2px solid black;}
div.box0 {box-sizing: border-box;
    width: 50%;
    border: 5px solid red;
    float: left;}
div.box {box-sizing: border-box;
    width: 50%;
    border: 5px solid red;
    float: right;}
</style>
```



ال **box-sizing**: تعمل على عدم خروج العنصر من المساحة التي تم تحديها له سواء قمت بعمل padding & border له

ملاحظة : لا يعمل على متصفح الأوبرا

: Text shadow

يضيف **text-shadow** الظل ظلاً للنص.
تقبل هذه الخاصية قائمة بالظلال مفصولة بفواصل يتم تطبيقها على النص.

```
h1 {  
    -webkit-text-shadow: 10px 10px 10px #F00;  
    -moz-text-shadow: 10px 10px 10px #F00;  
    -o-text-shadow: 10px 10px 10px #F00;  
    text-shadow: 10px 10px 10px #F00  
}
```

وهو قريب من درس ال **box shadow**
البريفكس الخاصة ب **shadow** في **Css3**

CSS Syntax

```
text-shadow: h-shadow v-shadow blur-radius color|none|initial|inherit;
```

أيضاً لدينا قيمتين مهمتين هما

[vertical Shadow + Horizontal Shadow]

القيمة العمودية + القيمة الأفقية

Default value is 0 و **Blur-radius** هو نسبة انتشار ال **shadow**

Text-shadow with red neon glow

```
h1 {  
    text-shadow: 5px 5px 5px #FF0000;  
}
```

: القيمة الإقتراضية و معناها لا يوجد ظل **None**

: Text overflow

خاصية خاصة بالتحكم في النص الزائد عن المحتوى
تحدد الخاصية **text-overflow** للنص كيف يجب أن يتم توجيه المحتوى
الفائض الذي لا يتم عرضه إلى المستخدم. يمكن قصها أو عرض علامة القطع (...) أو
عرض سلسلة مخصصة.

: **overflow** كل من الخصائص التالية مطلوبة للنص

[white-space: nowrap; overflow: hidden;]

ملاحظة: يتم استخدام خاصية `text-overflow: hidden;` مع `overflow: visible;` وهذا لإخفاء النص الزائد الذي لا يتم عرضه للمستخدم

CSS Syntax

`text-overflow: clip|ellipsis|string|initial|inherit;`

يتم قص النص ولا يمكن الوصول إليه **Default value:** Clip

This is some lo...

تقديم علامة حذف ("...") لتمثيل النص المقطوع **: Ellipsis**

This is some...

تقديم السلسلة المعطاة لتمثيل النص المقطوع **: String**

This is some lo...

مثال / استخدام خاصية `text-overflow: hidden;` مع `overflow: visible;`

```
<style>
div.a {
    white-space: nowrap;
    width: 100px;
    overflow: hidden;
    text-overflow: ellipsis;
    border: 1px solid #000000;}
```

This is some...

```
</style>
</head>
<body>
<div class="a">This is some long text that will
not fit in the box.</div>
```

مراجعة:

: white-space

توجد خاصية `white-space` تجعل الكلام أما أن يستمر في سطر واحد مهما كان طويلاً أو ينزل بالكلمة للسطر الثاني وتم شرحها في كورس Css

```
<style>
div.a {
    width: 280px;
    white-space: normal;
    text-overflow: ellipsis;
    border: 1px solid #000000;}
```

This is some long text that will not fit in
the box.

```
<div class="a">This is some long text that will not fit in the box.</div>
```

:Word-warp

وهي خاصية تجعل الكلام الزائد ينزل إلى السطر الثاني مباشرة ولا يخرج خارج مساحة ال div

خاصية word-wrap تسمح للكلمات الطويلة لتكون قادرة على كسر و التفاف على السطر التالي.

CSS Syntax

```
word-wrap: normal|break-word|initial|inherit;
```

: Normal كسر الكلمات فقط في نقاط كسر المسموح بها

: Break-word يسمح لكسر الكلمات غير القابلة للكسر

word-wrap: normal (default):

```
This div contains a very long  
word:  
thisisaveryveryveryveryveryverylongword.  
The long word will break and  
wrap to the next line.
```

word-wrap: break-word:

```
This div contains a very long  
word:  
thisisaveryveryveryveryve  
rylongword. The long word  
will break and wrap to the next  
line.
```

```
<style>  
div {width: 200px; border: 1px solid #000000;}  
  
div.a {word-wrap: normal;}  
  
div.b {word-wrap: break-word;}  
</style>  
</head>  
<body>  
<h2>word-wrap: normal (default):</h2>  
<div class="a"> This div contains a very long word:  
thisisaveryveryveryveryverylongword. The long word will break and  
wrap to the next line.</div>  
<h2>word-wrap: break-word:</h2>  
<div class="b"> This div contains a very long word:  
thisisaveryveryveryveryverylongword. The long word will break and  
wrap to the next line.</div>  
</body>
```

الوضع الإفتراضي لها Normal وهي السماح للكلام بمعادرة مساحة ال div

Without Break The Word

https://www.google.com.sa/webhp_sourceid=chrome_instant&ion=1&espv=2&ie=UTF_8

Without Break The Word

https://www.google.com.sa/webhp_sourceid=chrome_instant&ion=1&espv=2&ie=UTF_8

أما **Break-word** فهو يكسر الكلمة
وينزل بها للسطر الثاني

الفرق بين white-space & word-wrap

لما **white-space**: لا يقوم بعمل التزام بمساحة ال div بل إنه ينزل بالكلمة للسطر الثاني مع أول مسافة بين الكلمة والكلمة الأخرى

لما **Word-wrap**: فلا يهتم بموضع المسافات بل يكسر الكلمة مباشرة وينزل بها للسطر الثاني التزاما بمساحة ال div

مثال يوضح الفرق بينهم

```
div.a {white-space: normal;}  
  
div.b {word-wrap: break-word;}
```

```
<style>  
div {width: 150px; border: 1px solid #000000;}  
div.a {white-space: normal;}  
div.b {word-wrap: break-word;}  
</style>  
</head>  
<body>  
<h2>white-space: normal :</h2>  
<div class="a"> This div contains a very long word:  
thisisaveryveryveryveryveryverylongword. The long word will  
break and wrap to the next line.</div>  
<h2>word-wrap: break-word:</h2>  
<div class="b"> This div contains a very long word:  
thisisaveryveryveryveryveryverylongword. The long word will  
break and wrap to the next line.</div>  
</body>  
</html>
```

white-space: normal :

This div contains a very long word:
thisisaveryveryveryveryveryveryveryverylongword.
The long word will break and wrap to the next line.

word-wrap: break-word:

This div contains a very long word:
thisisaveryveryveryververyverylongword.
The long word will break and wrap to the next line.

2D Transforms

تتيح لك تحويلات CSS ترجمة العناصر وتدويرها وتغيير حجمها وتشويشها.
التحويل هو تأثير يسمح لعنصر بتغيير الشكل والحجم والموضع
يدعم CSS التحويلات ثنائية وثلاثية الأبعاد.



Browser Support for 2D Transforms

Property	Chrome	Edge	Firefox	Safari	Opera
transform	36.0 4.0 -webkit-	10.0 9.0 -ms-	16.0 3.5 -moz-	9.0 3.2 -webkit-	23.0 15.0 -webkit- 12.1 10.5 -o-
transform-origin (two-value syntax)	36.0 4.0 -webkit-	10.0 9.0 -ms-	16.0 3.5 -moz-	9.0 3.2 -webkit-	23.0 15.0 -webkit- 12.1 10.5 -o-

CSS 2D Transforms

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

Rotate : يعني دوران العنصر

Rotate: دوران العنصر وهو يقبل قيم سلبية بالسالب

تقوم طريقة **rotate()** بتدوير عنصر في اتجاه عقارب الساعة أو عكس اتجاه عقارب الساعة وفقاً لدرجة معينة.

يؤدي استخدام القيم السالبة إلى تدوير العنصر عكس اتجاه عقارب الساعة

ال **rotate** يقبل زاوية معينة

deg = degrees يعني درجة

```
div {  
    -ms-transform: rotate(20deg); /* IE 9 */  
    -webkit-transform: rotate(20deg); /* Safari */  
    transform: rotate(20deg);  
}
```

: Scale

وهو يقوم بعمل تمدد لعنصر للطول والعرض أي [width + height] تعمل طريقة **scale()** على زيادة أو إنفاص حجم العنصر (وفقاً للمعايير المحددة للعرض والارتفاع).



-ms-transform: scale(width, height)
-webkit-transform: scale(width, height)

```
div {  
    -ms-transform: scale(2, 3); /* IE 9 */  
    -webkit-transform: scale(2, 3); /* Safari */  
    transform: scale(2, 3);  
}
```

وهذا يعني أنه يضرب ال **width*2 ,height*3** وهكذا فهذه القيم التي نضعها بين الأقواس يقوم بضربها المتصفح في العرض والأرتفاع.

ويمكن وضع قيمة واحدة واعتمادها لل **width + height** هكذا

```
-ms-transform: scale(2); /* IE 9 */  
-webkit-transform: scale(2); /* Safari */  
transform: scale(2); /* Standard syntax */}
```

ويمكن كتابتها بطريقة أخرى
transform: scale(1.5) أو transform: scaley(1.5)

```
-ms-transform: scale(2); /* IE 9 */  
-webkit-transform: scale(2); /* Safari */  
transform: scale(2); /* Standard syntax */
```

لتحديد الجهة المراد عمل scale لها وهي y فقط أو x فقط

والقيمة الإفتراضية لها / Transform: none,

: Translate

وهو يقوم بتحريك العنصر لمسافة معينة هذه المسافة تعتمد على (x + y) و y, x يقبل قيم سلبية وإيجابية
تحرك طريقة translate() عنصراً من موقعها الحالي وفقاً للمعايير المعطاة
للمحور X والمحور Y

```
div {  
-ms-transform: translate(50px, 100px); /* IE 9 */  
-webkit-transform: translate(50px, 100px); /* Safari */  
transform: translate(50px, 100px);  
}
```

وهنا لابد من كتابة قيمتين لاعتمادها لل x + y

ملاحظة : فإذا كتبت قيمة واحد فسوف يتم اعتمادها لل x فقط أما ال y فلن يتم اعتماد قيمة x لها أي أنهم لا يشتركون في القيمة مثل Scale
وذلك يمكن عمل translate & translateX

Transform: translateX(40px)

Transform: translateY(40px)

يمكن استخدام أكثر من خاصية transform في نفس الوقت

```
.translate:hover {  
-webkit-transform: translateX(-160px) scale(.75) rotate(20deg);  
transform: translateX(-160px) scale(.75) rotate(20deg)
```

فنلاحظ هنا أننا استخدمنا ال scale + translate + rotate في نفس الوقت

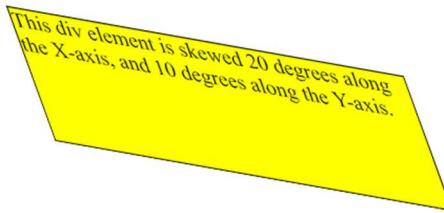
: skew

وهو يقوم بعمل ميلان للعنصر أو انحراف بزايا معينة هذه الزوايا أنت من تقوم بإعطائها له.

تقوم طريقة **skew()** بعمل ميلان لعنصرًا على المحور X و Y بواسطة الزوايا المحددة **skew(x,y)**.

```
-ms-transform: skew(20deg,10deg); /* IE 9 */  
-webkit-transform: skew(20deg,10deg); /* Safari */  
transform: skew(20deg,10deg); /* Standard syntax */
```

```
<style>  
div {margin:50px 50px;  
width: 300px;  
height: 100px;  
background-color: yellow;  
border: 1px solid black;}  
div#myDiv {  
-ms-transform: skew(20deg,10deg); /* IE 9 */  
-webkit-transform: skew(20deg,10deg); /* Safari */  
transform: skew(20deg,10deg); /* Standard syntax */  
}</style>  
</head>  
<body>  
<div id="myDiv">  
This div element is skewed 20 degrees along the X-axis, and  
10 degrees along the Y-axis.  
</div>
```



مثال /

وال **skew** يقبل قيم سلبية ولا يعتمد القيمة لـ **y** ، **x** بل لابد من كتابة كل قيمة

ويعتمد القيمة بـ **deg = degrees** بالدرجة للزوايا

إذا لم يتم تحديد المعلمة الثانية ، فإنها تحتوي على قيمة صفرية. لذا ، فإن المثال التالي ينطوي عنصر <div> ٢٠ درجة على طول المحور:

يعني : إذا تمت كتابة **skew(10deg)** فإنه يحسب هذه القيمة للعنصر **X** فقط
ويحسب للعنصر **Y** صفر تلقائياً

: طريقة يزيل عنصر على طول المحور Y من زاوية معينة **skewY()**

: طريقة ت先把 عنصرًا على طول المحور X بزاوية معينة **skewx()**

: matrix

وهي تجمع كل ما تعلمناه في ال transform فال matrix تقوم بعملها وحدها

طريقة يجمع بين جميع أساليب تحويل 2D في واحد **matrix()**

تأخذ طريقة **matrix()** ستة معلمات ، تحتوي على وظائف رياضية ، والتي تسمح لك rotate, scale, move (translate), and skew

elements



ال matrix تقبل 6 قيم

Transform : matrix(a,b,c,d,x,y)

وهذا المقصود بقيم a b c d x y

The parameters are as follow: matrix(scaleX(),skewY(),skewX(),scaleY(),translateX(),translateY())

```
transform: matrix( a, b, c, d, X, Y );
transform: matrix( ScaleX, SkewY, SkewX, ScaleY, TranslateX, TranslateY )  
I
```

ولكي ثبت هذا سوف نختبر ال translate

```
img:hover {
    transform: matrix( 1, 0, 0, 1, 300, 30 );
    /*transform: translate(300px, 30px)*/
}
```

في هذا المثال اعطينا لل matrix نفس القيم التي اعطيناها لل translate وقمنا بتجربتها

وهذا لما قمنا بتجربة ال scale

```
img:hover {
    transform: matrix( -1, 0, 0, 1, 0, 0 );
    /*transform: scale(-1, 1)*/
}
```

Transform 3D

تتيح لك CSS تتنسيق العناصر باستخدام التحويلات ثلاثية الأبعاد.

Browser Support for 3D Transforms

تحدد الأرقام في الجدول إصدار المتصفح الأول الذي يدعم الخاصية بشكل كامل.
الأرقام متبوعة بـ -webkit- أو -moz- أو -o- تحديد الإصدار الأول الذي يعمل مع بادئة prefix

Property	Chrome	Edge	Firefox	Safari	Opera
transform	36.0 12.0 -webkit-	10.0	16.0 10.0 -moz-	9.0 4.0 -webkit-	23.0 15.0 -webkit-
transform-origin (three-value syntax)	36.0 12.0 -webkit-	10.0	16.0 10.0 -moz-	9.0 4.0 -webkit-	23.0 15.0 -webkit-
transform-style	36.0 12.0 -webkit-	11.0	16.0 10.0 -moz-	9.0 4.0 -webkit-	23.0 15.0 -webkit-
perspective	36.0 12.0 -webkit-	10.0	16.0 10.0 -moz-	9.0 4.0 -webkit-	23.0 15.0 -webkit-
perspective-origin	36.0 12.0 -webkit-	10.0	16.0 10.0 -moz-	9.0 4.0 -webkit-	23.0 15.0 -webkit-
backface-visibility	36.0 12.0 -webkit-	10.0	16.0 10.0 -moz-	9.0 4.0 -webkit-	23.0 15.0 -webkit-

rotate 3d

[rotate3d() + rotateZ() + rotateY() + rotateX()]

: تقوم الطريقة بتدوير عنصر حول محوره X عند درجة معينة



: تقوم الطريقة بتدوير عنصر حول محوره y عند درجة معينة

عند درجة معينة

: تقوم الطريقة بتدوير عنصر حول محوره Z عند درجة معينة

عند درجة معينة

```

<style>
div {
    width: 150px;
    height: 100px;
    background-color: yellow;
    border: 1px solid black;}
#myDiv {
    -webkit-transform: rotateZ(90deg); /* Safari */
    transform: rotateZ(90deg); /* Standard syntax */}
</style>
</head>
<body>

<div>
This a normal div element.
</div>
<div id="myDiv">
This div element is rotated 90 degrees.
</div>

```

This a normal div element.

This div element is rotated 90 degrees.

This a normal div element.

مثال على استخدام `rotateZ()`

نفس المثال على `rotateX()`

This div element is rotated 90 degrees.

```

-webkit-transform: rotatX(90deg); /* Safari */
transform: rotateX(70deg); /* Standard syntax */

```

rotate3d()

تقوم دالة CSS `rotate3d()` بتعريف تحويل يقوم بتدوير عنصر حول محور ثابت في مساحة ثلاثة الأبعاد ، دون تشويهه.

syntax

يتم تحديد مقدار الدوران الذي تم إنشاؤه بواسطة () : `rotate3d()` بواسطة 3 أرقام وزاوية واحدة تمثل قيمة العدد s إحداثيات [Z + Y + X] للمتجه الذي يشير إلى محور الدوران

يمثل <angle> زاوية الدوران ؛ إذا كانت موجبة ، ستكون الحركة في اتجاه عقارب الساعة ؛ إذا كان سالبًا ، فسيكون عقارب الساعة..

```

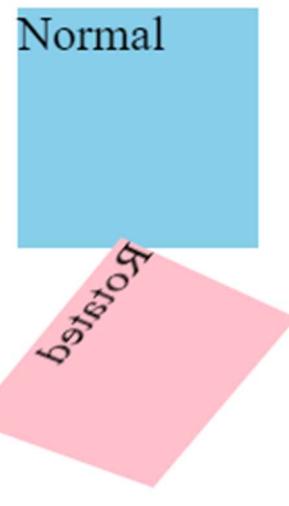
-webkit-transform: rotate3d(0, 1, 0, 60deg);
transform: rotate3d(0, 1, 0, 60deg);

```

ملاحظ : لكي يتم تشغيل خاصية 3d فالعنصر لابد أن يكون له منظور ثلاثي الأبعاد 3d أو علشان تعمل أكتفيت لخاصية 3dSpace عموماً فلابد أن يكون للعنصر منظور ثلاثي الأبعاد أو perspective

مثال /

```
<style>
body {perspective: 800px;}
div {
  width: 80px;
  height: 80px;
  background-color: skyblue;}
.rotated {
  transform: rotate3d(1, 2, -1, 192deg);
  background-color: pink;}
</style>
</head>
<body>
<div>Normal</div>
<div class="rotated">Rotated</div>
</body>
```



Translate 3d

وهو يقوم بتحريك العنصر لمسافة معينة هذه المسافة تعتمد على [Z+Y+X]

و x, y يقبل قيم سلبية وإيجابية

يقوم translate3d() بإعادة تشغيل وظيفة CSS عنصراً في مساحة ثلاثية الأبعاد. ويمكن كتابة القيم بالسالب

```
<head>
<style>
div {
  width: 60px;
  height: 60px;
  background-color: skyblue;
}
.moved {
  /* Equivalent to perspective(500px) translateX(10px) */
  transform: perspective(500px) translate3d(10px, 0, 0px);
  background-color: pink;
}
</style>
</head>
<body>
<div>Static</div>
<div class="moved">Moved</div>
<div>Static</div>
</body>
</html>
```



وفي حالة أردنا اختيار عنصر [Z + Y + X]

فلا حاجة لإضافة 3d
transform: translatez(10px);

ولكن لابد من إضافة المنظور ثلاثي الأبعاد كي يعمل

body{perspective: 500px }

Scale 3d

ولابد من دخول خاصية أخرى معه حتى يظهر الـ

Scale

transform: scale3d(2, 0.7, 0.2) translateZ(100px);

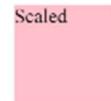
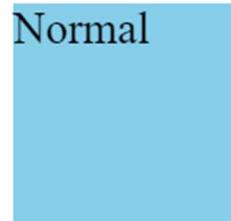
وفي حالة أردنا اختيار عنصر [Z + Y + X]

فلا حاجة لإضافة 3d ولكن لكي يعمل البعد الثالث Z فلا بد من دخول خاصية أخرى معه

transform: scalez(-1) translateZ(12px);

مثال /

```
<style>
    body{perspective: 500px }
div {
    margin: 50px auto;
    width: 80px;
    height: 80px;
    background-color: skyblue;}
.scaled {
    transform: scale3d(1, 1, -52) translateZ(12px);
    background-color: pink;}
</style>
</head>
<body>
<div>Normal</div>
<div class="scaled">Scaled</div>
</body>
```



Matrix 3d

ويقبل ١٦ قيمة في ال 3d

```
matrix3d(a1, b1, c1, d1, a2, b2, c2, d2, a3, b3, c3, d3, a4, b4, c4, d4)
```

للمزيد

<http://ds-overdesign.com/transform/matrix3d.html>

Perspective

ومعناه المنظور

يتم استخدام خاصية **perspective** لمنح عنصر ثلاثي الأبعاد موقعًا معيناً.

تحدد خاصية المنظور المسافة التي يكون الكائن بعيداً عن المستخدم. لذلك ، ستؤدي القيمة الأقل إلى تأثير ثلاثي الأبعاد أكثر كثافة من قيمة أعلى.

عند تعريف خاصية المنظور لعنصر ، تكون عناصر CHILD هي التي تحصل على مشاهدة المنظور ، وليس العنصر نفسه.

perspective: 300px; Default value : nane;

وهي تعني المسافة بين العنصر وبين البوزير المستخدم للشاشة

ويمكن تغييرها فهي ليست ثابتة

Perspective-origin

هو المكان الذي ينظر المشاهد منه للبوزير الخاص بالعنصر تحدد خاصية **perspective-origin** عند الموضع الذي يبحث فيه المستخدم عن العنصر ثلاثي الأبعاد.

عند تعريف خاصية **perspective-origin** لعنصر ، تكون عناصر CHILD هي التي ستحصل على التأثير ، وليس العنصر نفسه.

ملاحظة : يجب استخدام هذه الخاصية **perspective-origin** بالتزامن مع خاصية **perspective**!

Default value: 50% 50%

و هو عند كتابته فهو يأخذ قيمتين (x+y)

perspective-origin: x-axis y-axis|initial|inherit;

Syntax

X-axis : تحديد مكان العرض في المحور X

القيم الممكنة:

left : وهي القيمة يسار

Center : وهي قيمة في المنتصف

Right : وهي قيمة يمين

Length : وهي قيمة باليبيكسل

Default value: 50% %

Y-axis : تحديد مكان العرض في المحور y

القيم الممكنة:

top : وهي القيمة الأعلى

Center : وهي قيمة في المنتصف

bottom : وهي قيمة الأسفل

Length : وهي قيمة باليبيكسل

Default value: 50% %

```
-webkit-perspective: 100px; /* Safari 4-8 */  
-webkit-perspective-origin: left; /* Safari 4-8 */  
perspective: 100px;  
perspective-origin: left;
```

Transform-origin

و هذه الخاصية تسمح لك بتغيير البوزيشن للعنصر المعمول له transform تسمح لك خاصية **transform-origin** بتغيير موضع العناصر المتحولة. يمكن للتحولات ثنائية الأبعاد تغيير المحورين x و y لعنصر ما. يمكن للتحولات ثلاثية الأبعاد أيضًا تغيير المحور z لعنصر ما.

ملاحظة: يجب استخدام هذه الخاصية مع خاصية transform.

Default value: 50% 50%

و هو عند كتابته فهو يأخذ قيمتين (x+y)

Syntax

```
transform-origin: x-axis y-axis z-axis|initial|inherit;
```

تحديد مكان العرض في المحور X : **X-axis**

القيم الممكنة:
left : وهي القيمة يسار

Center : وهي قيمة في المنتصف

Right : وهي قيمة يمين

Length : وهي قيمة بالبيكسل

Default value: 50% % : وهي نسبة مئوية

تحديد مكان العرض في المحور y : **Y-axis**

القيم الممكنة:
top : وهي القيمة الأعلى

Center : وهي قيمة في المنتصف

: وهي قيمة الأسفل bottom

: وهي قيمة بالبيكسل Length

Default value: 50% % : وهي نسبة مئوية

لتحديد مكان العرض في المحور Z للتحولات ثلاثية الأبعاد : Z-axis

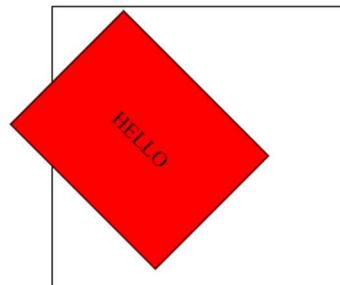
القيم الممكنة:

: وهي قيمة بالبيكسل Length

/ مثال /

```
<style>
#div1 {
    position: relative;
    height: 200px;
    width: 200px;
    margin: 100px;
    padding: 10px;
    border: 1px solid black;}
#div2 {
    padding: 50px;
    position: absolute;
    border: 1px solid black;
    background-color: red;
    transform: rotate(45deg);
    transform-origin: 20% 40%;}
</style>
</head>
<body>
<h1>The transform-origin Property</h1>
<div id="div1">
    <div id="div2">HELLO</div>
</div>
</body>
</html>
```

The transform-origin Property



نلاحظ في المثال /

استخدام خاصية transform مع خاصية transform-origin

Transform-style

تحدد خاصية **transform-style** كيفية عرض العناصر المتدخلة في مساحة ثلاثة الأبعاد.

ملاحظة: يجب استخدام هذه الخاصية مع خاصية **transform**.

transform-style: flat|preserve-3d|initial|inherit; **Syntax**

: يحدد أن العناصر الفرعية لن تحفظ بموقعها ثلاثي الأبعاد. هذا هو الافتراضي **default**

وهو لا يحافظ على 3d transform الذي قمت بعمله

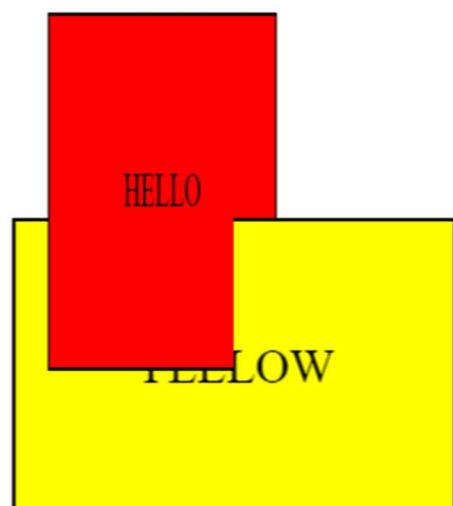
: يحدد أن العناصر الفرعية ستحافظ على موقعها ثلاثي الأبعاد **Preserve-3d**

يحافظ على 3d داخل العناصر الملحة داخل العنصر الأب

```
<style>
#div1 {
    position: relative;
    height: 200px;
    width: 200px;
    margin: 100px;
    padding: 10px;
    border: 1px solid black;}
#div2 {
    padding: 50px;
    position: absolute;
    border: 1px solid black;
    background-color: red;
    transform: rotateY(60deg);
    transform-style: preserve-3d;}
#div3 {
    padding: 40px;
    position: absolute;
    border: 1px solid black;
    background-color: yellow;
    transform: rotateY(-60deg);}
</style>
</head>
<body>
    <div id="div2">HELLO
        <div id="div3">YELLOW</div>
    </div>
</body>
```

يحافظ على ال 3d الخاص

بالعناصر عند استخدام



استخدام خاصية `transform-style` مع خاصية `transform`

Backface Visibility

تحدد خاصية `backface-visibility` ما إذا كان يجب أن يكون الوجه الخلفي للعنصر مرئياً عند مواجهة المستخدم.

الوجه الخلفي للعنصر هو صورة طبق الأصل للوجه الأمامي الذي يتم عرضه

هذه الخاصية مفيدة عندما يتم تدوير العنصر. فإنه يتيح لك اختيار ما إذا كان يجب على المستخدم رؤية الوجه الخلفي أم لا.

خاصية جميلة تفيد عند عمل المجسمات الثري دي،

CSS Syntax

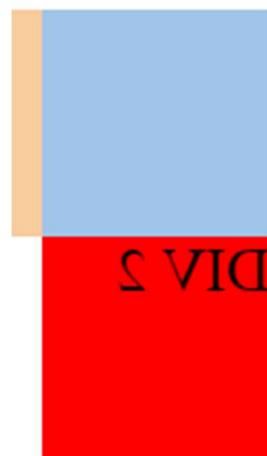
```
backface-visibility: visible|hidden|initial|inherit;
```

. الجانب الخلفي مرئي Default value : **Visible**

. الجانب الخلفي غير مرئي : **Hidden**

```
<style>
div {
  position: relative;
  height: 60px;
  width: 60px;
  background-color: red;
  -webkit-transform: rotateY(180deg); /* Chrome, Safari, Opera */
  transform: rotateY(180deg);}
#div1 {
  -webkit-backface-visibility: hidden; /* Chrome, Safari, Opera */
  backface-visibility: hidden;}
#div2 {
  -webkit-backface-visibility: visible; /* Chrome, Safari, Opera */
  backface-visibility: visible;}
</style>
</head>
<body>
<div id="div1">DIV 1</div>
<div id="div2">DIV 2</div>
</body>
```

مثال



Background New Features

تعيين الخاصية اختزال **background** كافة خصائص الخلفية في تعريف واحد.

خصائص جديدة في CSS3 :

background-image

background-size

background-position

background-color

background-origin

background-repeat

background-attachment

background-clip

Background-size

تحدد خاصية **background-size** حجم صور الخلفية.

هناك أربعة بنى جمل مختلفة يمكنك استخدامها مع هذه الخاصية:
بناء جملة الكلمة الرئيسية "auto" و "cover" و "contains" ، وبناء الجملة
نحو القيمة الواحدة يحدد عرض الصورة (الارتفاع يصبح "تلقائي") ، وبناء الجملة
الثانية القيمة (القيمة الأولى: عرض الصورة ، القيمة الثانية: الارتفاع) ، وبناء
الجملة الخفية (مفصولة بفواصل).

حيث أنها تتحكم في حجم العرض والإرتفاع

CSS Syntax

`background-size: auto|Length|cover|contain|initial|inherit;`

. يتم عرض صورة الخلفية في حجمها الأصلي **Default value : Auto**

: يضبط العرض والارتفاع لصورة الخلفية. تحدد القيمة الأولى العرض ، بينما تقوم القيمة الثانية بتعيين الارتفاع. إذا تم إعطاء قيمة واحدة فقط ، يتم تعيين القيمة الثانية على "auto"

: تغيير حجم صورة الخلفية لتغطية الحاوية بأكملها ، حتى إذا كان عليها أن تمدد الصورة أو تقطع قليلاً من إحدى الحواف

: قم بتغيير حجم صورة الخلفية للتأكد من أن الصورة مرئية بالكامل

```

<style>
  #img1{
    padding: 124px;
    border: 2px solid black;
    background-image: url(mountain.jpg);
    background-repeat: no-repeat;
    background-size: 700px 150px;}
</style>
</head>
<body>
  <div id="img1"></div>
</body>

```



`background-size: 46%;}`



`background-size:cover;}`



`background-size:contain;}`



cover يعني تأخذ مساحة ال div بالكامل يعني تعدل 100%

Contain

وهو يقوم بعمل scaling لـ background ليها ثم بعد ذلك يقوم بعمل تكرار للصورة في حالة أن الصورة حجمها أصغر من حجم ال div

ويقوم بعمل تكرار للصورة في حالة أن يكون

أكبر من ال `height` الخاص

بالصورة الخاصة بالخلفية أو ال `width`

حيث أنه يقوم بملئ الخلفية بالصورة دون تكرار لها ولكنها تتناسب مع حجم ال `div`

نلاحظ أن ال `cover`

لا يقوم بعمل تكرار للصورة في حالة أن ال `width` أصغر أو أكبر من حجم الصورة

background-clip

يحدد الخلفية من أين تبدأ
تحدد خاصية `background-clip` المسافة التي يجب أن تمتد بها الخلفية
(اللون أو الصورة) داخل عنصر ما

CSS Syntax

`background-clip: border-box|padding-box|content-box|initial|inherit;`

Default value : `border-box`

: تمتد الخلفية إلى الحافة الداخلية للحدود `padding-box`

: تمتد الخلفية إلى حافة مربع المحتوى `content-box`

border-box

```
<div id="example1">  
<p>.تمتد الخلفية خلف الحدود</p>  
</div>  
  
#example1 {  
    border: 10px dotted black;  
    padding: 15px;  
    background: #d3d3d3;  
    background-clip: border-box;}
```



تمتد الخلفية خلف الحدود.

padding-box

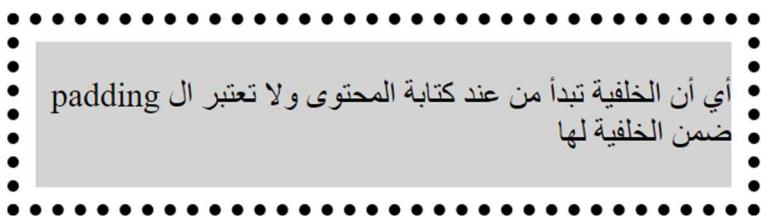
```
<div id="example2">  
<p>.تمتد الخلفية إلى الحافة الداخلية للحدود</p>  
</div>  
  
#example2 {  
    border: 10px dotted black;  
    padding: 15px;  
    background: #d3d3d3;  
    background-clip: padding-box;}
```



تمتد الخلفية إلى الحافة الداخلية للحدود.

content-box

```
<div id="example3">  
    أي أن الخلفية تبدأ من عند كتابة المحتوى ولا تعتبر ال  
    <p> ضمن الخلفية لها</p>  
</div>  
  
#example3 {  
    border: 10px dotted black;  
    padding: 15px;  
    background: #d3d3d3;  
    background-clip: content-box;}
```



أي أن الخلفية تبدأ من عند كتابة المحتوى ولا تعتبر ال padding ضمن الخلفية لها

Background-origin

تحدد الخاصية **background-origin** موضع الأصل (منطقة تحديد الخلفية) لصورة الخلفية.

ملاحظة : يحدد الخلفية من أين تبدا وهو نفس الخاصية السابقة **background-clip** ولكن هذه الخاصية للصور

ملاحظة : هذه الخاصية ليس لها أي تأثير إذا كان **background-clip** **fixed attachment**

CSS Syntax

```
background-origin: padding-box|border-box|content-box|initial|inherit;
```

Default value : **padding-box**
اليسرى لحافة الحشو

تبدأ صورة الخلفية من الزاوية العلوية اليسرى من الحد : **border-box**

تبدأ صورة الخلفية من الزاوية العلوية اليسرى للمحتوى

padding-box

```
<div id="example1">  
<h2>padding-box</h2>  
<p>تبدأ صورة الخلفية من الزاوية العلوية اليسرى لحافة الحشو</p>  
</div>
```

```
#example1 {  
    border: 10px double black;  
    padding: 25px;  
    background: url(paper.gif);  
    background-repeat: no-repeat;  
    background-origin: padding-box;  
}
```

padding-box

تبدأ صورة الخلفية من الزاوية العلوية اليسرى لحافة الحشو

border-box

```
<div id="example2">  
<h2>border-box</h2>  
<p>تبدأ صورة الخلفية من الزاوية العلوية اليسرى من الحد</p>  
</div>
```

```
#example2 {  
    border: 10px double black;  
    padding: 25px;  
    background: url(paper.gif);  
    background-repeat: no-repeat;  
    background-origin: border-box;}
```

border-box

تبدأ صورة الخلفية من الزاوية العلوية اليسرى من الحد

content-box

```
<div id="example3">  
<h2>content-box</h2>  
<p>تبدأ صورة الخلفية من الزاوية العلوية اليسرى للمحتوى</p>  
</div>
```

```
#example3 {  
    border: 10px double black;  
    padding: 25px;  
    background: url(paper.gif);  
    background-repeat: no-repeat;  
    background-origin: content-box;}
```

content-box

تبدأ صورة الخلفية من الزاوية العلوية اليسرى للمحتوى

Background-repeat

تحدد خاصية **background-repeat** ما إذا كانت صورة الخلفية ستتكرر أم لا.

بشكل افتراضي ، يتم تكرار صورة الخلفية عمودياً وأفقياً. **By default** **background-position**: يتم وضع صورة الخلفية وفقاً لخاصية **background-position**. إذا لم يتم تحديد موضع خلفية ، فسيتم وضع الصورة دائمًا في أعلى الزاوية اليسرى من العنصر.

CSS Syntax

```
background-repeat: repeat|repeat-x|repeat-y|no-repeat|initial|inherit;
```

repeat : تتكرر صورة الخلفية عمودياً وأفقياً. سيتم قص الصورة الأخيرة إذا لم تتناسب. هذا هو الافتراضي **default**

horizontally : يتم تكرار صورة الخلفية أفقياً فقط

vertically : يتم تكرار صورة الخلفية رأسياً فقط

no-repeat : لا يتم تكرار صورة الخلفية. سيتم عرض الصورة مرة واحدة فقط

space : يتم تكرار صورة الخلفية قدر الإمكان دون لقطة. يتم تثبيت الصور الأولى والأخيرة على جانبي العنصر ، ويتم توزيع المسافات البيضاء بالتساوي بين الصور

round: تتكرر صورة الخلفية ويتم سحقها أو تمددها لملء الفراغ (لا توجد فجوات)

The background-repeat Property

Here, the background image is repeated only vertically.

```
<style>
body {
    background-image: url("paper.gif");
    background-repeat: space;
}
</style>
</head>
<body>

<h1>The background-repeat Property</h1>
<p>Here, the background image is repeated only vertically.</p>
```

Background-position

تحدد الخاصية **background-position** موضع البداية لصورة الخلفية.
تمثيل: بشكل افتراضي ، يتم وضع **background-image** في الزاوية العلوية
اليسرى من عنصر ما ، وتكرارها عموديا وأفقيا.

CSS Syntax

left top
left center
left bottom
right top
right center
right bottom
center top
center center
center bottom

`background-position: value;`

إذا حددت كلمة رئيسية واحدة فقط ،

فستكون القيمة الأخرى هي "center"

: X% y%

القيمة الأولى هي الموضع الأفقي والقيمة الثانية هي الرأسية. أعلى الزاوية اليسرى
.٪. الزاوية اليمنى السفلى ٪.٪. إذا قمت بتحديد قيمة واحدة فقط ،
فستكون القيمة الأخرى ٪.٪. القيمة الافتراضية هي: ٪.٪.

: Xpos ypos

القيمة الأولى هي الموضع الأفقي والقيمة الثانية هي الرأسية. أعلى الزاوية اليسرى
هي ٪. يمكن أن تكون الوحدات بكسل (0px 0px) أو أي وحدات CSS
أخرى. إذا قمت بتحديد قيمة واحدة فقط ، فستكون القيمة الأخرى ٪.٪. يمكنك خلط
٪ والموقف.

```
<style>
body {
    background-image: url('w3css.gif');
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-position: center center;
}
</style>
</head>
<body>
```

```
<h1>The background-position Property</h1>
<p>Here, the background image will be positioned centered at top.</p>
```

The background-position Property

Here, the background image will be positioned centered at top.



background-position: 50%;

50% ستعطي نفس النتيجة

Background-attachment

تعين خاصية **background-attachment** ما إذا كانت صورة الخلفية تمرر scrolls مع باقي الصفحة أو ثابته fixed.

CSS Syntax

```
background-attachment: scroll|fixed|local|initial|inherit;
```

: scroll : سيتم تمرير صورة الخلفية مع الصفحة. هذا هو الافتراضي default

: fixed : لن يتم تمرير صورة الخلفية مع الصفحة ستظل ثابته

: local : سيتم تمرير صورة الخلفية بمحطويات العنصر

```
body{  
    background-image: url("img_tree.gif");  
    background-repeat: no-repeat;  
    background-attachment: scroll;  
}
```

[Try it Yourself »](#)

Overflow new features

:overflow-x

تحدد الخاصية **overflow-x** ما إذا كان سيتم قص المحتوى أو إضافة شريط تمرير أو عرض محتوى تجاوز السعة لعنصر مستوى الكتلة ، عند تجاوزه عند الحافتين اليسرى واليمنى **لتمييز**: استخدم الخاصية **overflow-y** لتحديد الاقطاع عند الحافتين العلوية والسفلى.

CSS Syntax

```
overflow-x: visible|hidden|scroll|auto|initial|inherit;
```

visible : لا يتم قص المحتوى ، وقد يتم عرضه خارج الحواف اليمنى واليسرى. هذا هو الافتراضي **default**

hidden : يتم قص المحتوى - ولا يتم توفير آية آلية للتمرير

scroll : يتم قص المحتوى ويتم توفير آية التمرير

auto : يجب أن تتسبب في توفير آية التمرير للصناديق الفائضة

```
<head>
<style>
div.ex1 {
    background-color: lightblue;
    width: 40px;
    overflow-x: scroll;}
</style>
</head>
<body>
<h2>overflow-x: scroll:</h2>
<div class="ex1">Lorem ipsum dolor sit amet,
consectetuer adipiscing elit...</div>
</body>
</html>
```

overflow-x: scroll:



: overflow-y

تحدد الخاصية **overflow-y** ما إذا كان سيتم قص المحتوى ، أو إضافة شريط تمرير ، أو عرض محتوى الفائض لعنصر مستوى الكتلة ، عند تجاوزه عند الحافتين العلوية والسفلية.

تلميح: استخدم الخاصية **overflow-x** لتحديد الاقطاع عند الحواف اليمنى واليسرى.

CSS Syntax

```
overflow-y: visible|hidden|scroll|auto|initial|inherit;
```

visible : لا يتم قص المحتوى ، وقد يتم عرضه خارج مربع المحتوى. هذا هو الافتراضي **default**

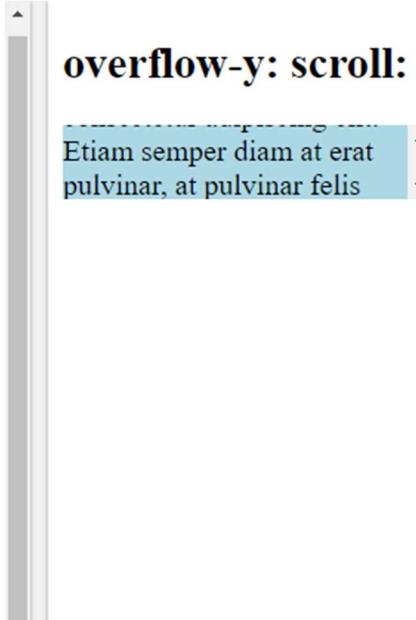
hidden : يتم قص المحتوى - ولا يتم توفير آلية للتمرير

scroll : يتم قص المحتوى ويتم توفير آلية التمرير

auto : يجب أن تنسحب في توفير آلية التمرير للصناديق الفائضة

```
<!DOCTYPE html>
<html>
<head>
<style>
div.ex1 {
    background-color: lightblue;
    height: 40px;
    width: 200px;
    overflow-y: scroll;}
</style>
</head>
<body>
<h2>overflow-y: scroll:</h2>
<div class="ex1">Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Etiam semper diam at erat pulvinar, at
pulvinar felis blandit. Vestibulum volutpat tellus diam,
consequat gravida libero rhoncus ut.</div>
</body>
</html>
```

overflow-y: scroll:



Etiam semper diam at erat pulvinar, at pulvinar felis

Resize

وهي يعني إعادة تحجيم وهي غير مدعومة في الأنترنت اكسيلولر تحدد خاصية **resize** ما إذا كان (وكيف) يمكن تغيير حجم عنصر من قبل المستخدم.

ملاحظة: لا تطبق خاصية **resize** على العناصر المضمنة أو لحظر العناصر التي يوجد بها **overflow = "visible"**. لذلك ، تأكد من تعين تجاوز السعة **"scroll", "auto", or "hidden"**.

CSS Syntax

```
resize: none|both|horizontal|vertical|initial|inherit;
```

: **none** القيمة الافتراضية. لا يمكن للمستخدم تغيير حجم العنصر

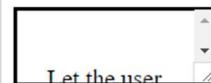
: **both** يمكن للمستخدم تغيير حجم ارتفاع العنصر وعرضه

: **horizontal** يمكن للمستخدم تغيير حجم عرض العنصر

: **vertical** يمكن للمستخدم تغيير حجم ارتفاع العنصر

```
<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><br/>وهي يعني إعادة تحجيم وهي غير مدعومة في الأنترنت اكسيلولر</p>
</body>
</html>
```

The resize Property



وهي يعني إعادة تحجيم وهي غير مدعومة في الأنترنت اكسيلولر:

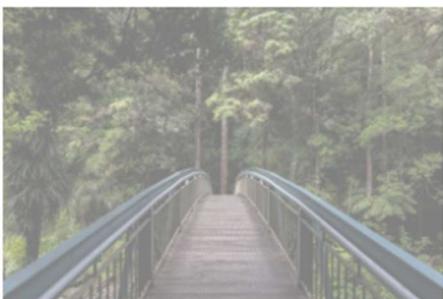
نلاحظ: أنه بإمكانك السحب بالماوس من جهة المنطقة اليمنى السفلى والسحب أفقى وراسى بكل اريحية

Opacity

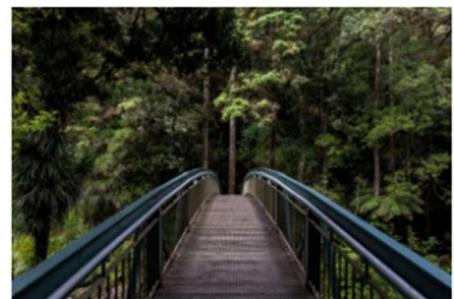
الشفافية وهي شفافية العنصر تقوم خاصية **opacity** بتعيين مستوى الشفافية لعنصر . يصف مستوى **opacity** مستوى الشفافية ، حيث ١ غير شفاف على الإطلاق ، ٠،٥ هو شفاف بنسبة ٥٠٪ ، و ٠ شفاف تماماً.



opacity 0.2



opacity 0.5



opacity 1
(default)

ملاحظة: عند استخدام خاصية **opacity** لإضافة شفافية إلى خلفية عنصر ما ، تصبح جميع عناصرها الفرعية شفافة أيضاً . هذا يمكن أن يجعل النص داخل عنصر شفاف بالكامل يصعب قراءته. إذا كنت لا تريد تطبيق **opacity** على العناصر الفرعية ، فاستخدم قيمة لون **RGBA** بدلاً من ذلك.

Syntax

opacity: number|initial|inherit;

يحدد الشفافية. من ٠،٠ (شفافة بالكامل) إلى ١،٠ (معتمة بالكامل) **number**

ملاحظة: تتم إضافة **filter** حتى تعمل على إنترنت أكسيلولر

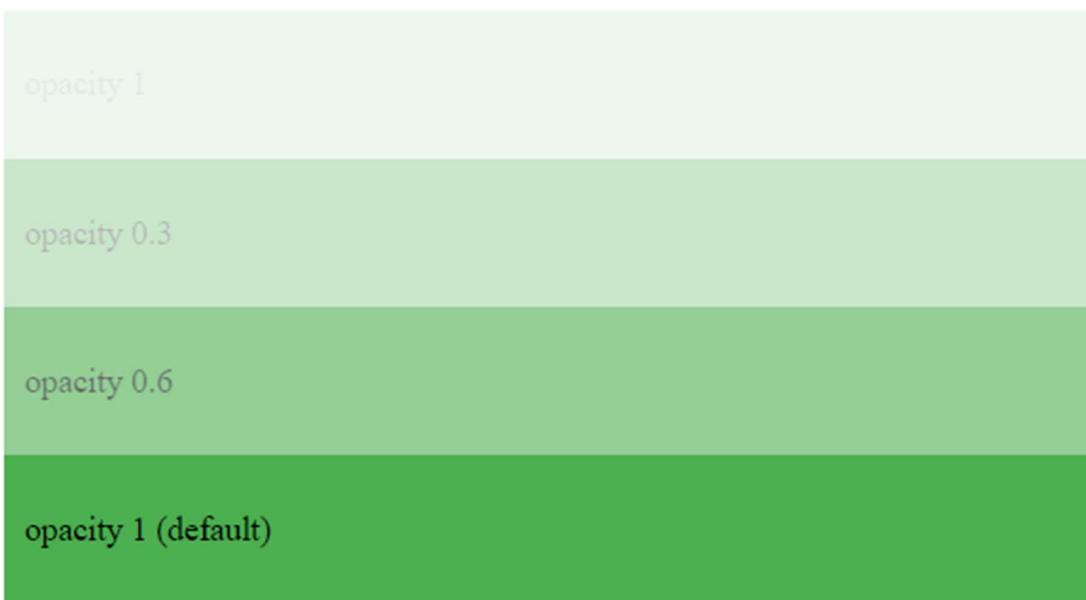
```
opacity: .9;  
filter: alpha(opcaity=90)
```

background-color: red

opacity: 0.5

/* filter: Alpha(opacity=50); /* IE8 and earlier

This element's opacity is 0.5! Note that both the text and the background-color are affected by the opacity level!



```
<style>
div {background-color: #4CAF50;padding: 10px;}
div.first {
    opacity: 0.1;filter: alpha(opacity=10); /* For IE8 and earlier */}
div.second {
    opacity: 0.3;
    filter: alpha(opacity=30); /* For IE8 and earlier */}
div.third {
    opacity: 0.6;
    filter: alpha(opacity=60); /* For IE8 and earlier */}
</style>
</head>
<body>
<div class="first"><p>opacity 1</p></div>
<div class="second"><p>opacity 0.3</p></div>
<div class="third"><p>opacity 0.6</p></div>
<div><p>opacity 1 (default)</p></div>
```

البوكس المرن : Flex box

ال flex box تقنية جديدة تعمل على المتصفحات الجديدة فقط ولا تعمل على المتصفحات القديمة

قبل وحدة Flexbox Layout ، كان هناك أربعة أوضاع للتخطيط :

لالأقسام في صفحة الويب : **Block**

for text: **Inline**

لبيانات الجدول ثنائي الأبعاد : **Table**

للموضع الصريح لعنصر : **Positioned**

تسهل الوحدة النمطية لمربع التصميم المرن تصميم بنية تخطيط مرنة دون استخدام تعوييم أو تحديد **structure without using float or positioning.** موضع.

Flexbox Elements

لبدء استخدام نموذج Flexbox ، يجب أولاً تعریف **flex container**.

1 2 3

يتمثل العنصر أعلى حاوية مرنة (المنطقة الزرقاء) مع ثلاثة عناصر مرنة.

```
<style>
.flex-container {
  display: flex; background-color: DodgerBlue;}
.flex-container > div {
  background-color: #f1f1f1;
  margin: 10px;
  padding: 20px;
  font-size: 30px;}
</style>
</head>
<body>
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>
```

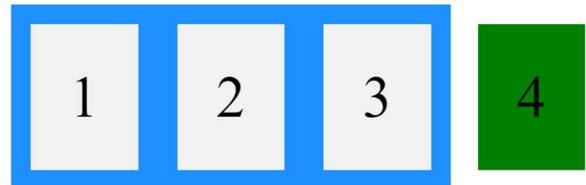
نلاحظ: اضافة `display:flex` فإن ال `div` تاتي بجوار بعضها

نستطيع وضع عنصر بجواره آخر في حاله استخدام `inline-flex`

```
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>
<div class="inline-container">4</div>
```

`display: inline-flex;`

```
<style>
.flex-container {
  display: inline-flex; background-color: DodgerBlue;}
.flex-container > div {
  background-color: #f1f1f1;
  margin: 10px;
  padding: 20px;
  font-size: 30px;}
.inline-container{
  display: inline-flex;
  margin: 10px;
  padding: 20px;
  font-size: 30px;
  background-color: green}
</style>
```



Parent

Flex-direction

Flex-wrap

Flex-flow

Justify-content

Align-items

Align-content

ملاحظة: مع جميع عناصر Flex box لابد من استخدام `display: flex`

```
display: flex;
justify-content: center;
```

Flex Direction

: Flex-direction

وهو اتجاه العنصر وليس اتجاه الكتابة
تحدد خاصية **flex-direction** الاتجاه الذي تريده الحاوية تتدفق العناصر
المرنة فيه.



الاتجاه المرن وهو يمشي معك حسب اتجاه الصفحة من اليمين للشمال أو العكس
direction : rtl & rtl

هناك 4 خصائص نستطيع التحكم فيها في طريقة عرض العنصر :

[Row + row-reverse + column + column-reverse]

Row: الاتجاه الأفقي

وهي تجعل العناصر مصفوفة في وضع من اليسار إلى اليمين في حالة الصفحة
اتجاهها rtl أما إن كان اتجاه الصفحة rtl فسيكون اتجاه العناصر من اليمين لليسار

```
<style>
.flex-container {
  display: flex;
  flex-direction: row;
  background-color: DodgerBlue;}
.flex-container > div {
  background-color: #f1f1f1;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;}
</style>
</head>
<body>
<div class="flex-container">
<div>1</div>
<div>2</div>
<div>3</div>
</div>
```



تم استخدام
display: flex;
flex-direction: row;

Row-reverse : الإتجاه العكسي للإتجاه الطبيعي

وهي تجعل العناصر مصفوفة في وضع من اليمين إلى الشمال في حالة الصفحة اتجاهها rtl أما إن كان اتجاه الصفحة ltr فسيكون اتجاه العناصر من الشمال إلى اليمين



`flex-direction: row-reverse;` نفس المثال السابق ولكن سنعدل فقط

Column : الإتجاه العمودي

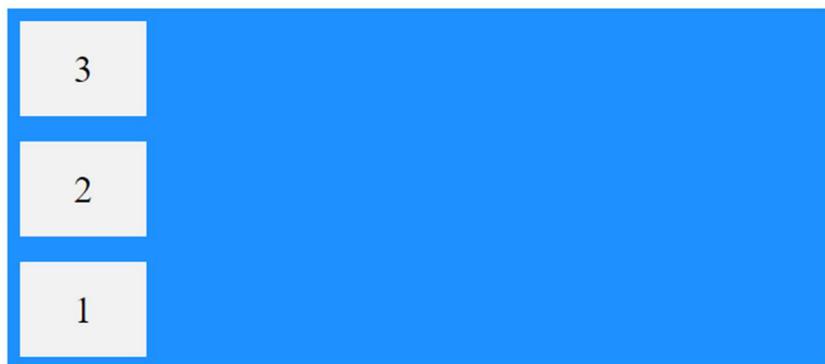
يضع العناصر بشكل عمودي من الأعلى للأسفل أيضا حسب اتجاه الصفحة كما تحدثنا



`flex-direction: column;` نفس المثال السابق ولكن سنعدل فقط

Column-reverse

يضع العناصر بشكل عمودي من الأسفل للأعلى أيضا حسب اتجاه الصفحة كما تحدثنا

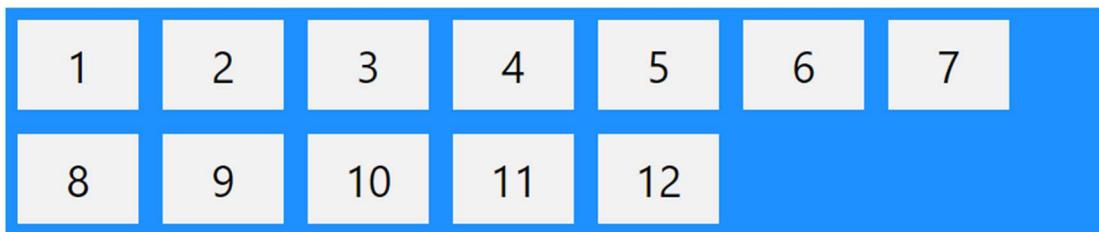


نفس المثال السابق ولكن سنعدل فقط

`flex-direction: column-reverse;`

Flex Wrap

Wrap يعني التفاف أو احتواء للعنصر
تحدد الخاصية **flex-wrap** ما إذا كان يجب التفاف العناصر المرنة أم لا
تتضمن الأمثلة أدناه 12 عنصراً مرتّباً ، لإثبات **flex-wrap** بشكل أفضل.



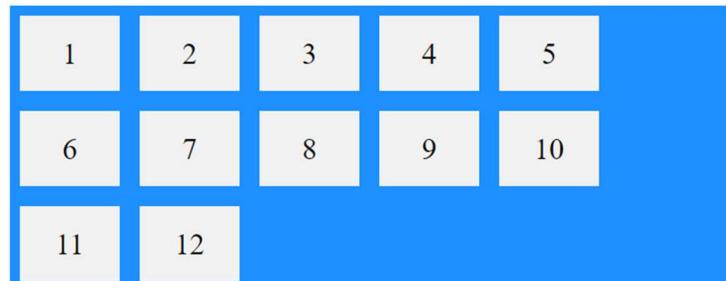
هناك ٣ خصائص نستطيع التحكم فيها في طريقة عرض العنصر :

[**warp + nowrap + wrap-reverse**]

```
<style>
.flex-container {
  display: flex;
  flex-wrap: wrap;
  background-color: DodgerBlue;
}
.flex-container > div {
  background-color: #f1f1f1;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>
<div class="flex-container">
  <div>1</div><div>2</div><div>3</div>
  <div>4</div><div>5</div><div>6</div>
  <div>7</div><div>8</div><div>9</div>
  <div>10</div><div>11</div><div>12</div>
</div>
```

Wrap

وهي عمل التفاف للعناصر في حالة الضرورة

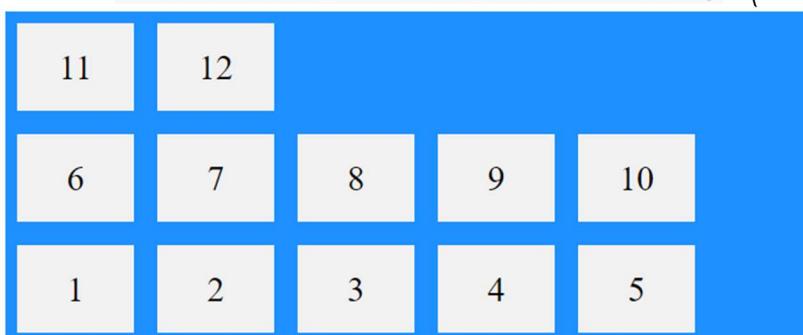


تم استخدام **flex-wrap: wrap;**

Wrap-reverse

وهي تعمل التفاف للعناصر ولكن التفاف معكوس فهي نفس wrap ولكنها معكوسه

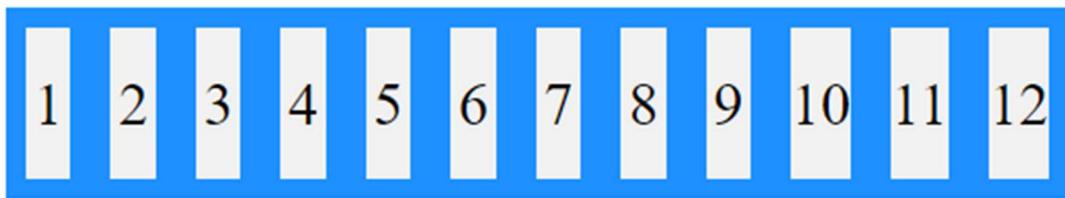
flex-wrap: wrap-reverse; نفس المثال السابق ولكن تم استخدام



nowrap

وهي Devaultvalue يعني لا تعمل التفاف للعناصر

`flex-wrap: nowrap;` نفس المثال السابق ولكن تم استخدام

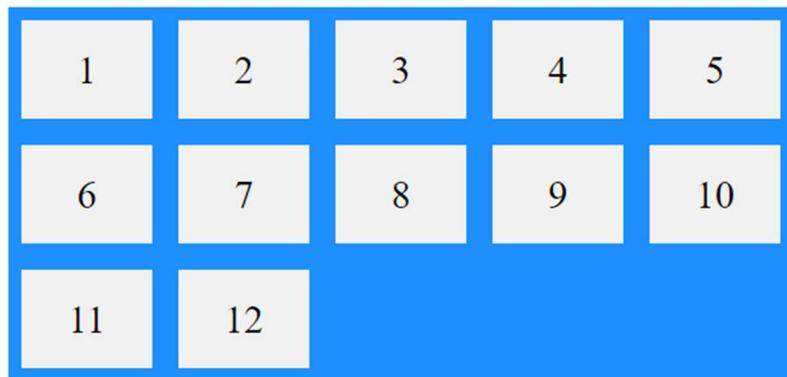


Flex-flow

`flex-direction + flex-wrap` هي الكود المختصر لـ

يعني تكتب كود القيمتين فيها مباشرة

`flex-flow: row wrap;` نفس المثال السابق ولكن تم استخدام



محاذاة المحتوى : justify-content

يتم استخدام الخاصية `justify-content` لمحاذاة العناصر المرنة



هناك 5 خصائص نستطيع التحكم فيها في طريقة عرض العنصر :

`[center + flex-start + flex-end + space-around
+ space-between]`

```

<style>
.flex-container {
  display: flex;
  justify-content: center;
  background-color: DodgerBlue;
}
.flex-container > div {
  background-color: #f1f1f1;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>

```

Center

وهو يقوم بتوسيط العناصر
تقوم قيمة المركز بمحاذاة العناصر المرنة

في منتصف container

تم استخدام:

justify-content: center;

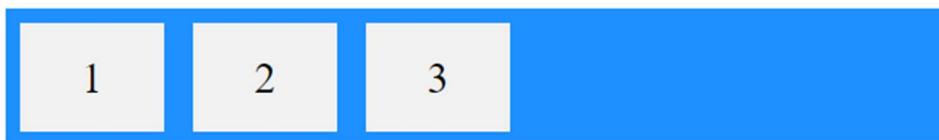


Flex-start

وهي تعني أن المحاذاة تبدأ من ال start

تقوم قيمة flex-start بمحاذاة العناصر المرنة في بداية container (وهذا هو (Default value)

نفس المثال السابق ولكن بإستخدام **justify-content: flex-start;**



Flex-end

ويقوم بعمل محاذاه من نهاية الصف

نفس المثال السابق ولكن بإستخدام **justify-content: flex-end;**



Space-between

يعلم مسافات متساوية بين العناصر

`justify-content: space-between;` نفس المثال السابق ولكن بإستخدام



Space-around

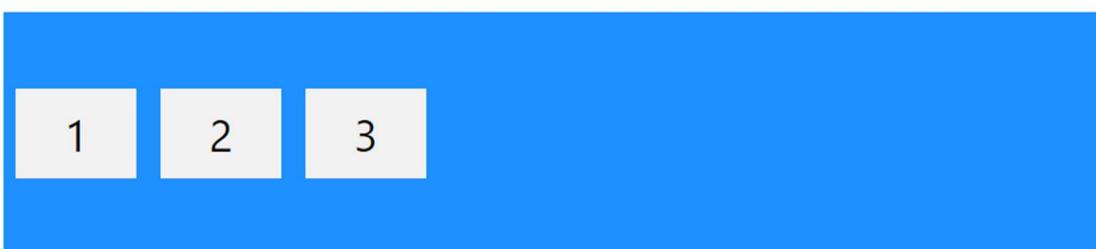
يعلم مسافات حول العنصر وبينه

`justify-content: space-around;` نفس المثال السابق ولكن بإستخدام



Align items

يتم استخدام خاصية `align-items` لمحاذاة العناصر المرنة عمودياً.



في هذه الأمثلة ، نستخدم حاوية عالية تبلغ ٢٠٠ بكسل ، لشرح الخاصية محاذاة بشكل أفضل

هناك 5 خصائص نستطيع التحكم فيها في طريقة عرض العنصر :

`[center + flex-start + flex-end + stretch + baseline]`

Center

تقوم قيمة المركز بمحاذاة العناصر المرنة في منتصف الحاوية

فإنه يأتي بالعنصر في المنتصف عمودياً

```

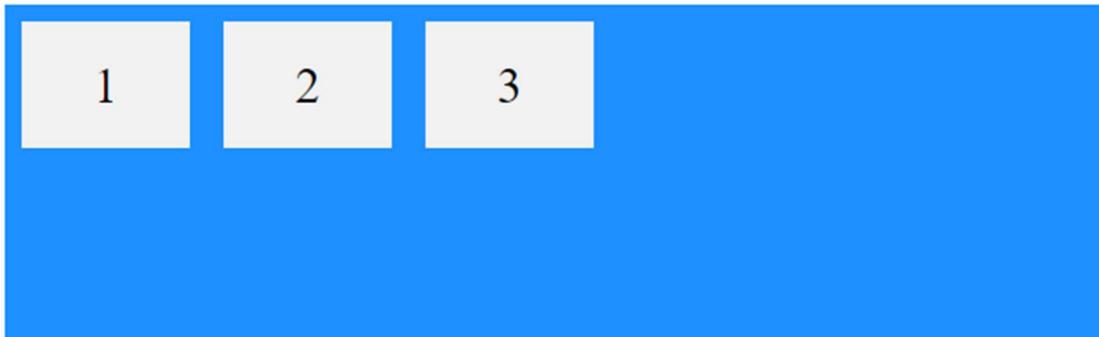
<style>
.flex-container {
  display: flex;
  height: 200px;
  align-items: center;
  background-color: DodgerBlue;
}
.flex-container > div {
  background-color: #f1f1f1;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>

```



flex-start

تقوم قيمة **flex-start** بمحاذاة العناصر المرنة في container أعلى



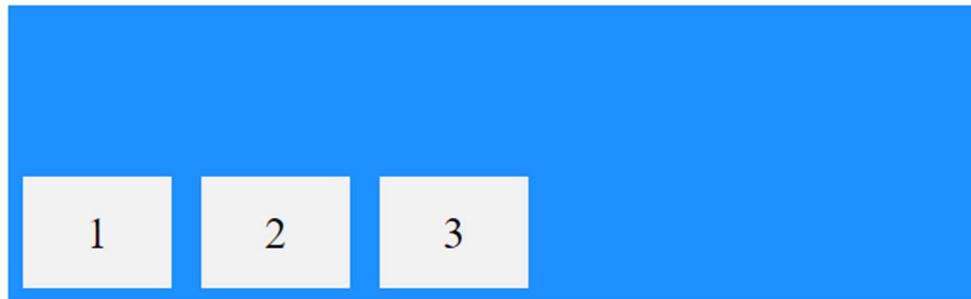
Align items & Justify-content الفرق بين

Justify-content : يتعامل مع ال div كعنصر
align : يتعامل مع Align items

flex-end

فإنه يأتي بالعنصر في الأسفل بشكل عمودي

تربط قيمة **flex-end** العناصر المرنة في أسفل container

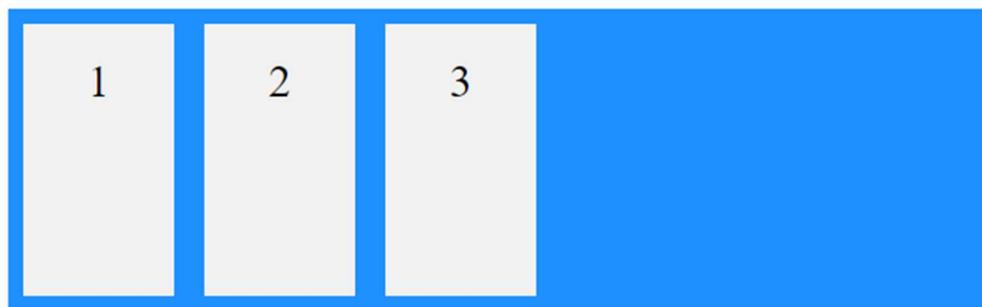


stretch

تمدد قيمة الامتداد العناصر المرنة لملء الحاوية (هذا هو الإعداد الافتراضي)

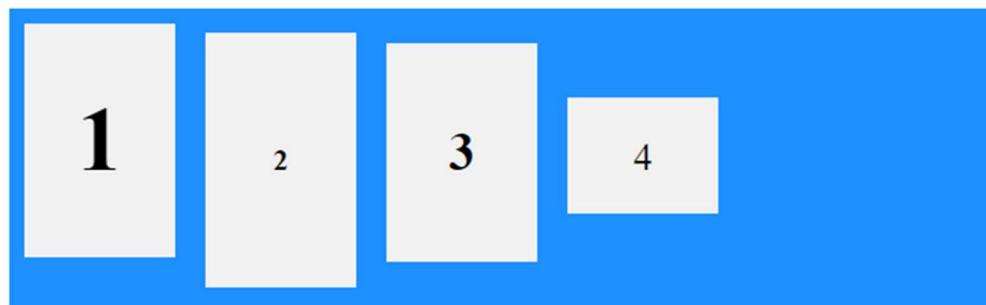
`align-items: stretch;`

نفس المثال السابق ولكن استخدمنا



baseline

وهي تقريباً ديفولت نورمال
تقوم قيمة **baseline** بمحاذاة العناصر المرنة مثل محاذاة خطوط الأساس الخاصة بها

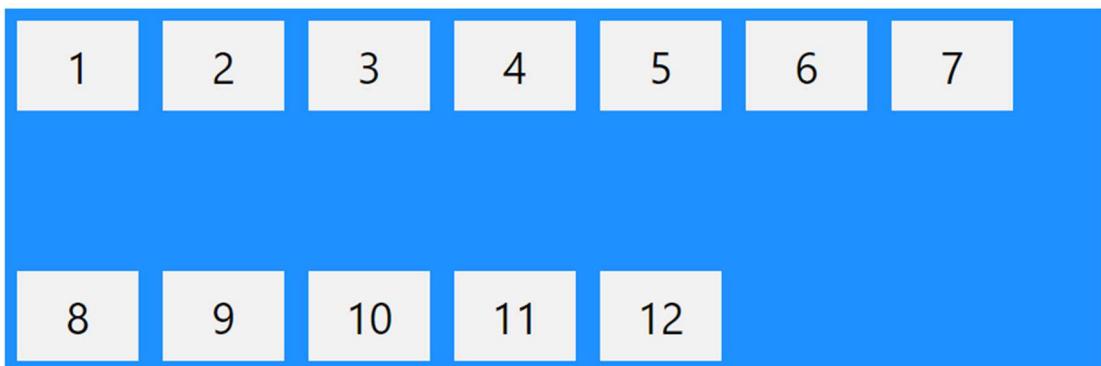


ملاحظة: يستخدم المثال حجمًا مختلفًا للخط لإظهار أن العناصر تم محاذاتها بواسطة خط الأساس النصي



Align-content

يتم استخدام الخاصية **align-content** لمحاذاة خطوط مرنة ال **div** يعمل عندما لا يكون هناك مساحة فارغة في ال **div**



في هذه الأمثلة ، نستخدم حاوية عالية بسعة ٦٠٠ بيكسل ، مع ضبط خاصية الغلاف المرن ، للتدليل على خاصية محاذاة المحتوى بشكل أفضل.

هناك ٦ خصائص نستطيع التحكم فيها في طريقة عرض العنصر :

[space-between + space-around + stretch + center + flex-start + flex-end]

Stretch

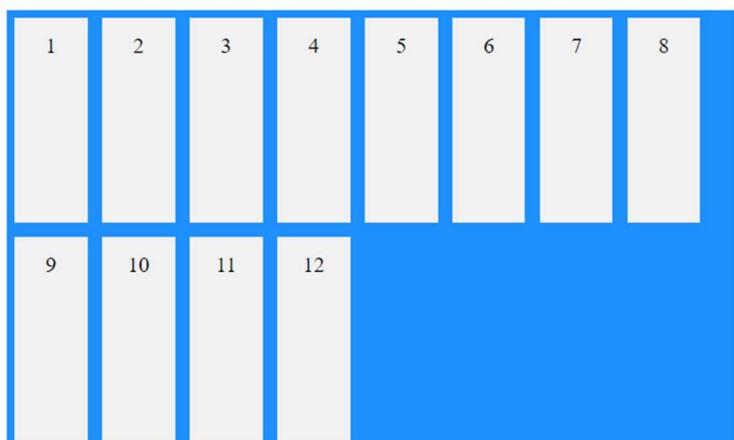
وهي الديفولت لها أيضا وسوف نستخدم **flex-wrap** حتى تظهر الخاصية تمتد القيمة **stretch** للخطوط المرنة لتسوّع المساحة المتبقية (this is)

(default)

تم استخدام

align-content: stretch;

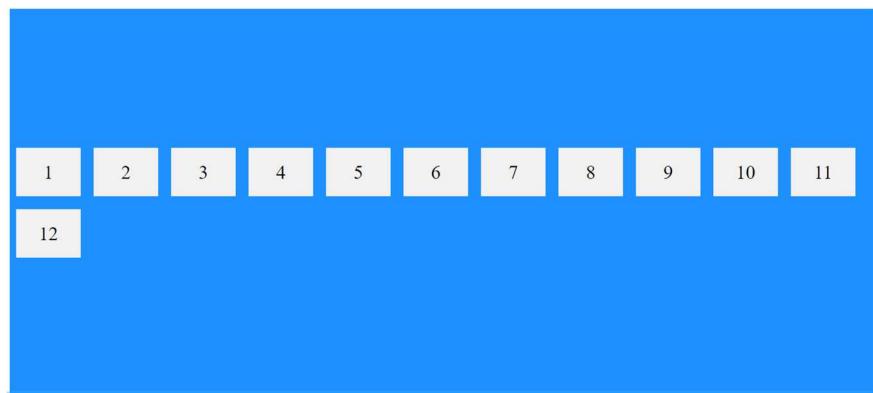
```
<style>
.flex-container {
  display: flex;
  height: 600px;
  flex-wrap: wrap;
  align-content: stretch;
  background-color: DodgerBlue;}
.flex-container > div {
  background-color: #f1f1f1;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;}
</style>
</head>
<body>
<div class="flex-container">
  <div>1</div><div>2</div><div>3</div>
  <div>4</div><div>5</div><div>6</div>
  <div>7</div><div>8</div><div>9</div>
  <div>10</div><div>11</div><div>12</div>
</div>
```



center

تعرض قيمة `center` خطوط فليكس في منتصف container

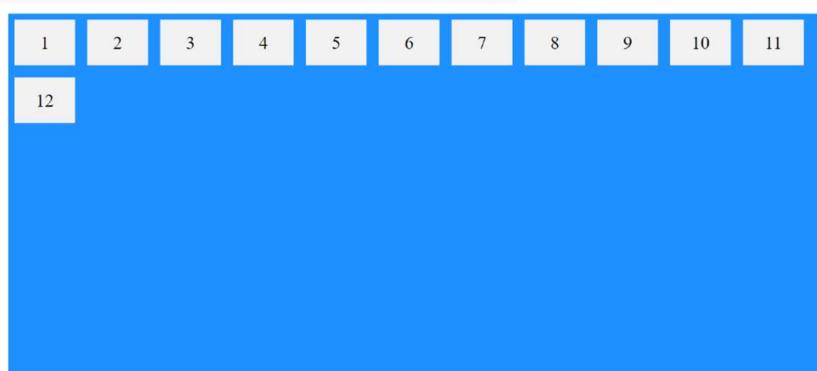
`align-content: center;` نفس المثال السابق وتم استخدام



Flex-start

تعرض قيمة `start-flex` الخطوط المرن في بداية container

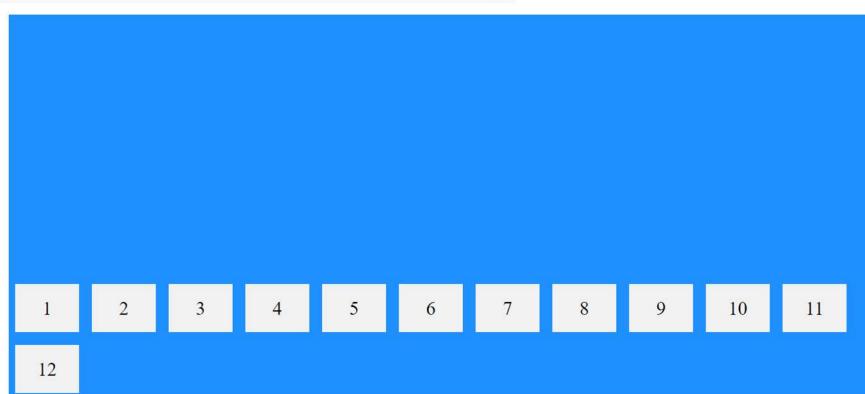
`align-content: flex-start;` نفس المثال السابق وتم استخدام



Flex-end

تعرض قيمة `flex-end` خطوط مرنة في نهاية container

`align-content: flex-end;` نفس المثال السابق وتم استخدام



الفرق بين Align-content & Justify-content

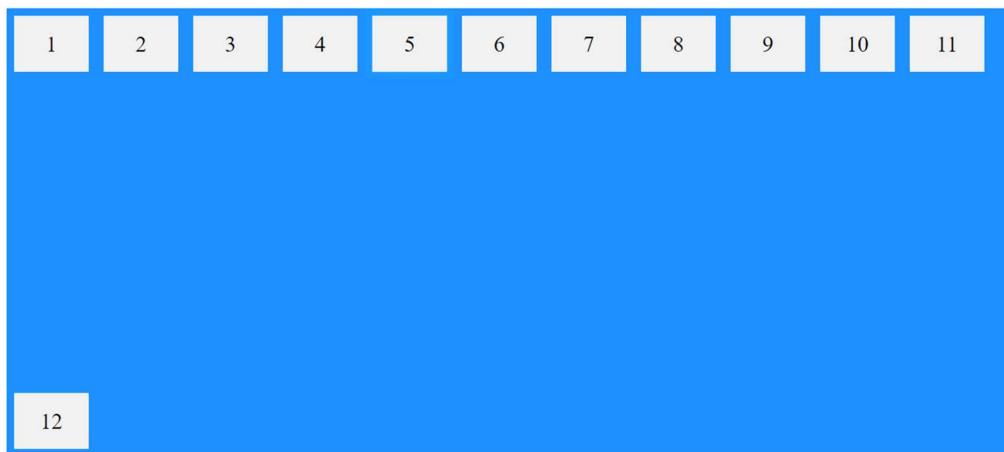
Justify-content : يتعامل بالعرض

Align-content : يتعامل بالطول

Space-between

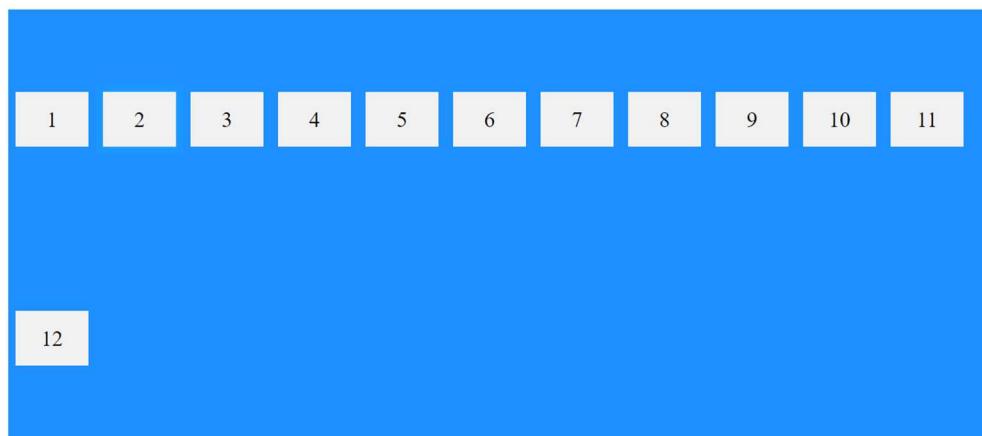
يقوم بعمل مسافات متساوية بين العناصر ولكن بالطول

تعرض الخطوط المرنة بمسافة متساوية بينهما



Space-around

يقوم مسافة قبل العنصر وبعده ولكن بالطول أيضاً
تعرض قيمة المسافة حول الخطوط المرنة بمسافة قبل ، وبين ، وبعدها



Perfect Centering التمركز المثالي

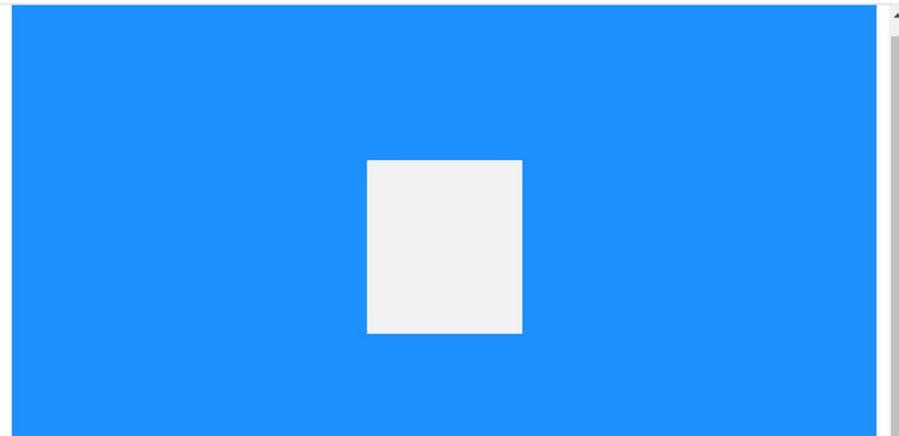
في المثال التالي ، سنحل مشكلة نمط شائع جدًا: التمركز المثالي.



الحل: قم بتعيين كل من خصائص `justify-content` و `align-items` إلى مركز ، وسيتم توسيط العنصر المرن بشكل مثالي:

```
.flex-container {  
    display: flex;  
    height: 300px;  
    justify-content: center;  
    align-items: center;  
}
```

```
<head>  
<style>  
.flex-container {  
    display: flex;  
    justify-content: center;  
    align-items: center;  
    height: 300px;  
    background-color: DodgerBlue;}  
.flex-container>div {  
    background-color: #f1f1f1;  
    color: white;  
    width: 100px;  
    height: 100px;}  
</style>
```



Child المرن Flex box:

Child Elements (Items)

تصبح direct child المباشرة لحاوية مرنّة عناصر (flex) تلقائياً عند وجودها داخل أب مرن



```
<style>
.flex-container {
  display: flex; background-color: #f1f1f1;
}
.flex-container > div {
  background-color: DodgerBlue;
  color: white;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
</div>
```



يمثل العنصر أعلاه أربعة عناصر مرنّة زرقاء داخل flex container. رمادية.

The flex item properties are

order

flex-grow

flex-shrink

flex-basis

flex

align-self

Order

وهي خاصية تسمح لك بترتيب العناصر بخلاف ترتيبها الأصلي
ليس من الضروري ظهور العنصر المرن الأول في التعليمية البرمجية كأول عنصر
في التخطيط
يجب أن تكون قيمة الطلب رقمًا ، القيمة الافتراضية هي صفر.

```
<style>
.flex-container {
  display: flex;
  align-items: stretch;
  background-color: #f1f1f1;
}
.flex-container>div {
  background-color: DodgerBlue;
  color: white;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>
<div class="flex-container">
  <div style="order: 3">1</div>
  <div style="order: 2">2</div>
  <div style="order: 4">3</div>
  <div style="order: 1">4</div>
</div>
```

تم استخدام

```
<div class="flex-container">
  <div style="order: 3">1</div>
  <div style="order: 2">2</div>
  <div style="order: 4">3</div>
  <div style="order: 1">4</div>
</div>
```

4

2

1

3

Flex-grow

وهي خاصية تسمح للعنصر بالتمدد وملئ الفراغ في الدف
تحدد الخاصية **flex-grow** مقدار عنصر Flex سوف ينمو نسبة إلى باقي
العناصر المرنة
يجب أن تكون القيمة رقمًا ، القيمة الافتراضية هي 0 .

1

2

3

الديفولت لها = (صفر)

نفس المثال السابق ولكن هذا 3 div فقط

```
<div class="flex-container">
  <div style="flex-grow: 1">1</div>
  <div style="flex-grow: 1">2</div>
  <div style="flex-grow: 5">3</div>
</div>
```

Flex-shrink

وهي خاصية تسمح للعنصر بالتلقلص
تحدد خاصية الانكماش المرن مقدار تقلص عنصر المرن بالنسبة لبقية العناصر
المرنة.
يجب أن تكون القيمة رقمًا ، القيمة الافتراضية هي 1 .

```
<div class="flex-container">  
  <div>1</div>  
  <div>2</div>  
  <div style="flex-shrink: 0">3</div>  
  <div>4</div>  
  <div>5</div>  
  <div>6</div>  
  <div>7</div>  
  <div>8</div>  
  <div>9</div>  
  <div>10</div>  
</div>
```

الديفولت لها = (1)

نفس المثال السابق ولكن تم تزويد
عدد الديفات واضافة الخاصية



ملاحظة : تم رفض تقليل عنصر 3 لأنه أخذ خاصية Flex-shrink:0 أما باقي العناصر فتم تقليلها

Flex-basis

يعني الأساس Basis

تستطيع بهذه الخاصية تحديد width لعنصر معين من عناصر ال flex-box أو
لأيتم معين جو ال flex-box

نفس المثال السابق ولكن مع 4 ديفات فقط

```
<div class="flex-container">  
  <div>1</div>  
  <div>2</div>  
  <div style="flex-basis: 200px">3</div>  
  <div>4</div>  
</div>
```



والقيمة الإفتراضية له هي : auto

وهي المساحة التي كان سيعطيها له ال flex في الأساس

The flex Property

الكود المختصر لل flex

الخاصية flex و flex-shrink و flex-grow نختصر بها في كود واحد basis

نستطيع كتابة كود مختصر لل flex يعبر عن الخصائص المتعددة له

```
flex: 0 1 auto;  
/* Flex: Flex-Grow | Flex-Shrink | Flex-Basis */
```

```
<div class="flex-container">  
  <div>1</div>  
  <div>2</div>  
  <div style="flex: 0 0 200px">3</div>  
  <div>4</div>  
</div>
```



align-self

تحدد الخاصية `align-self` المحاذة للعنصر المحدد داخل الحاوية المرنة.
تجاوز خاصية `align-self` المحاذة الافتراضية التي تم تعينها بواسطة
خاصية `align-items` الحاوية.

هناك ٦ خصائص نستطيع التحكم فيها في طريقة عرض العنصر :

[`auto + baseline + center + flex-start + flex-end + stretch`]



```
<style>
.flex-container {
  display: flex;
  height: 200px;
  background-color: #f1f1f1;}
.flex-container > div {
  background-color: DodgerBlue;
  color: white;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;}
</style>
</head>
<body>
<div class="flex-container">
<div style="align-self: auto">1</div>
<div style="align-self: flex-start">2</div>
<div style="align-self: flex-end">3</div>
<div style="align-self: baseline">4</div>
<div style="align-self: center">5</div>
<div style="align-self: stretch">6</div>
</div>
```

في هذا المثال تم تطبيق جميع الخصائص

Transition

وهو الإنقال

يعلم على التحكم في الإنقال من خاصية لخاصة أخرى أو بعبارة أخرى يعلم على التحكم في وقت الإنقال
تتيح لك CSS **transition** تغيير قيم الممتلكات بسلامة (من قيمة إلى أخرى) ، خلال مدة محددة.



كيفية استخدام **transition** ؟

لإنشاء تأثير إنقال ، يجب عليك تحديد شيئين:

١ - **css property** : خاصية css التي تريد إضافة تأثير لها

٢ - **duration** : مدة التأثير أو هو الوقت الذي يحتاجه العنصر للإنقال من حالة لحالة Default value: 0s
ملاحظة: إذا لم يتم تحديد جزء المدة ، فلن يكون لهذا الإنقال أي تأثير ، لأن القيمة الافتراضية هي 0.

يبداً تأثير التحويل عند تغيير قيمة خاصية (width) CSS المحددة

```
<head>
<style>
div {
    width: 100px;
    height: 100px;
    background: red;
    -webkit-transition: width 2s; /* For Safari 3.1 to 6.0 */
    transition: width 2s;
}
div:hover {width: 300px;}
</style>
</head>
<body>
<div></div>
</body>
</html>
```



لاحظ: عندما تضع الماوس فوق ال div سوف يزداد width بحركة انسيلابية جميلة ويمكن عمل التأثير لل height كذلك
transition: width 2s, height 4s;

٣ - Timing-function : منحى السرعة

هو الوقت الذي يحتاجه transition لكي يذهب من عنصر لعنصر

حدد منحى السرعة للانتقال :

تحدد خاصية timing-function منحى السرعة لتأثير الانتقال

ولها عدة قيم :

ease : يحدد تأثير النقل ببداية بطيئة ، ثم سريع ، ثم ينتهي ببطء (هذا هو الإعداد الافتراضي)

linear : يحدد تأثير انتقال بنفس السرعة من البداية إلى النهاية

ease-in : يحدد تأثير النقل ببداية بطيئة

ease-out : يحدد تأثير انتقال بنهاية بطيئة

ease-in-out : يحدد تأثيراً انتقالياً ببداية ونهاية بطبيتين

cubic-bezier(n,n,n,n) : يتيح لك تحديد القيم الخاصة بك في وظيفة

بزي مكعب

```
<style>
div {width: 100px;
  height: 50px;
  background: red;
  transition: width 2s;}
/* Standard syntax */
#div1 {transition-timing-function: linear;}
#div2 {transition-timing-function: ease;}
#div3 {transition-timing-function: ease-in;}
#div4 {transition-timing-function: ease-out;}
#div5 {transition-timing-function: ease-in-out;}
div:hover {width: 300px;}
</style>
</head>
<body>
<div id="div1">linear</div><br>
<div id="div2">ease</div><br>
<div id="div3">ease-in</div><br>
<div id="div4">ease-out</div><br>
<div id="div5">ease-in-out</div><br>
```



نلاحظ : أنشأنا استخدمنا [property css + duration] ثم استخدمنا timing-function في هذا المثال

٤ - Delay the Transition Effect : تأثير تأخير الإنقال

تحدد خاصية **delay** (بالثواني seconds) فترة تأخير Transition لتأثير الإنقال

وهو الوقت الذي يتأخره العنصر كي يبدأ في الإنقال

يعني وقت الإنتظار قبل حدوث إنقال الخاصية

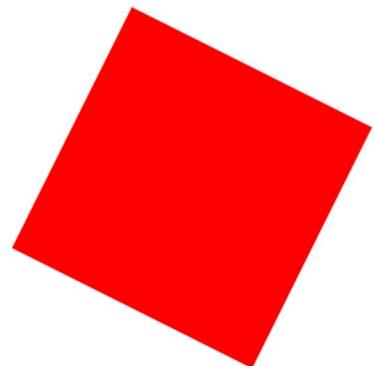
```
<style>
div {
    width: 100px;
    height: 100px;
    background: red;
    -webkit-transition: width 3s; /* Safari */
    -webkit-transition-delay: 1s; /* Safari */
    transition: width 3s;
    transition-delay: 1s;}
div:hover {width: 300px;}
</style>
</head>
<body>
<div></div>
</body>
```



نلاحظ: أنه عند تحديده بالمؤشر الماوس يتاخر بمقدار ثانية واحدة قبل أن يبدأ بحركة الإنقال ثم عندما يتم رفع الماوس يتاخر ايضاً بمقدار ثانية قبل أن يبدء في نهاية التأثير.

مثال // لل **transition** مع حركة دائرية لل

```
<style>
div {margin:50px auto;
    width: 100px;
    height: 100px;
    background: red;
    transition: width 2s, height 2s, transform 2s;}
div:hover {
    width: 200px;
    height: 200px;
    transform: rotate(180deg);}
</style>
</head>
<body>
<div></div>
</body>
```



ال kod المختصر لل **transition**

transition: Property Duration Timing-function Delay

بإمكاننا كتابة الأكواد أو الخصائص منفردة هكذا

```
<style>
div {
    width: 100px;
    height: 100px;
    background: red;
    /* Standard syntax */
    transition-property: width;
    transition-duration: 2s;
    transition-timing-function: linear;
    transition-delay: 1s;
}
div:hover {width: 300px;}
</style>
</head>
<body>
<div></div>
```



أو كتابتهم في كود واحد هكذا

transition :width 2s linear 1s;

معلومات إضافية:

أولاً : عند عمل Transition فلا بد من كتابة خاصيتيين أساسيتين حتى يعمل ال [Property + Duration] وهم **transition**

نستطيع إضافة أكثر من خاصية في نفس الوقت ولكن نفصل بينهم بـ (,)

transition: width 2s, height 4s;

بالنسبة **Property css**

transition : وهي تفعيل ال **Default value: all**

: وهي لإلغاء التفعيل **None**

```
transition-property: width, background-color; /* Default Value : all */
```

وينصح : بكتابة خاصية **Css** المراد حدوث التأثير لها منعاً لحدوث مشاكل

رابط لموقع يخبرك بالعناصر التي يمكن عمل transition لها

[CSS Transitions](https://developer.mozilla.org/ar/docs/Web/CSS/%D8%A7%D9%84%D8%B9%D9%86%D8%A7%D8%B5%D8%B1%D8%A7%D9%84%D8%AA%D9%8A%D9%85%D9%83%D9%86%D8%AA%D8%AD%D8%B1%D9%8A%D9%83%D9%87%D8%A7%D8%A8%D8%A7%D8%B3%D8%AA%D8%AE%D8%AF%D8%A7%D9%85)

Timing-function

Smooth : يبدأ على خطوات وينتهي على خطوات دون Steps

```
div p:first-of-type {  
    transition-timing-function: steps(2, start);  
}  
  
div p:last-of-type {  
    transition-timing-function: steps(2, end);  
}
```

Cubic-bezier

وهو لعمل معادلة للسرعة وهو يقبل 4 ارقام
رابط موقع للتدريب



Multi columns

متعدد الأعمدة

وهي خاصية تسمح لك بتقسيم div لأعمدة متعددة أو عمل أعمدة يتيح تخطيط CSS **multi-columns** تعریفًا سهلاً لأعمدة متعددة من النصوص - تماماً كما هو الحال في الصحف



: multi-columns

- **column-count**
- **column-gap**
- **column-rule-style**
- **column-rule-width**
- **column-rule-color**
- **column-rule**
- **column-span**
- **column-width**

: Column-count

وهي لتحديد عدد الأعمدة التي يجب تقسيم العنصر إليها تحدد خاصية **column-count** عدد الأعمدة التي ينبغي تقسيم العنصر إليها

Default value: auto

سينقسم المثال التالي النص في عنصر <div> إلى 3 أعمدة

ولابد من كتابة البريفكس الخاص بها

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

```

<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 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>

```

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

Lorem ipsum dolor sit amet, consectetur adipicing 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.

Column-gap

لتحديد الفجوة بين الأعمدة

تحدد خاصية **column-gap** الفجوة بين الأعمدة

Default value: normal

المثال التالي يحدد فجوة ٤٠ بكسل بين الأعمدة

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

Lorem ipsum dolor sit amet, consectetur adipicing 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.

البريفكس

```

-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;

```

column-rule

وهي ال border بين الأعمدة المسافات التي هي بين الأعمدة نستطيع وضع
فيها border
خاصية column-rule هي خاصية مختصرة لتعيين جميع خصائص عمود
القاعدة

Default value: medium none color

[width + style + color] :
column-rule: column-rule-width + column-rule-style
column-rule-color

```
div {  
    column-rule: 1px solid lightblue;  
}
```

ويمكن كتابته في كود مختصر هكذا
مع وضع البريفكس

```
<style>  
.newspaper {  
    column-count: 3;  
    column-gap: 40px;  
    column-rule: 1px solid lightblue;}  
</style>  
<p><b>Note:</b> Internet Explorer 9, and earlier versions, does not support the column-rule property.</p>  
<div class="newspaper">  
Lorem ipsum dolor sit amet, consectetuer 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>
```

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

lorem ipsum dolor sit amet, consectetuer 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

lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor 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.

: column-style

تحدد الخاصية column-rule-style نمط القاعدة بين الأعمدة

وهي مثل النمط الذي يأخذ border .. راجع درس border في css

border: 1px solid lightblue;

```
div {  
    column-rule-style: solid;  
}
```

:column-width

خاصية عرض العمود

تحدد خاصية **column-rule-width** عرض القاعدة بين الأعمدة

```
div {  
    column-rule-width: 1px;  
}
```

Default value: auto

: column-color

لون القاعدة بين الأعمدة **column-rule-color**

```
div {  
    column-rule-color: lightblue;  
}
```

column-span

تحدد خاصية **column-span** العرض المقترن للأعمدة

وهي إذا تم استخدام (h1-6) وتم عمل **column** سنجد أن ال **h2** أنت معهم في **column** في أول سطرين وهذا لا نريده فنقوم باستخدام ال **column-span** حتى تأتي ال **h2** وحدها في سطر وهذه الخاصية غير مدعومة في الفايرفوكس

Default value: none

```
<style>  
.newspaper {  
    column-count: 3;  
    column-gap: 40px;  
    column-rule: 1px solid lightblue;}  
h2 {column-span: all;}  
</style>  
</head>  
<body>  
<p><b>Note:</b> Firefox and Internet Explorer 9 (and earlier versions) do not support the column-span property.</p>  
<div class="newspaper">  
<h2>Lorem Ipsum Dolor Sit Amet</h2>  
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonumy 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>
```

Note: Firefox and Internet Explorer 9 (and earlier versions) do not support the column-span property.

Lorem Ipsum Dolor Sit Amet

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonumy 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.

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.

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.

```
h2 {  
    column-span: all;  
}
```

column-width

تحدد خاصية **column-width** العرض المقترن للأعمدة

يحدد المثال التالي أن العرض المقترن للأعمدة يجب أن يكون 100 بكسل

```
div {  
    column-width: 100px;  
}
```

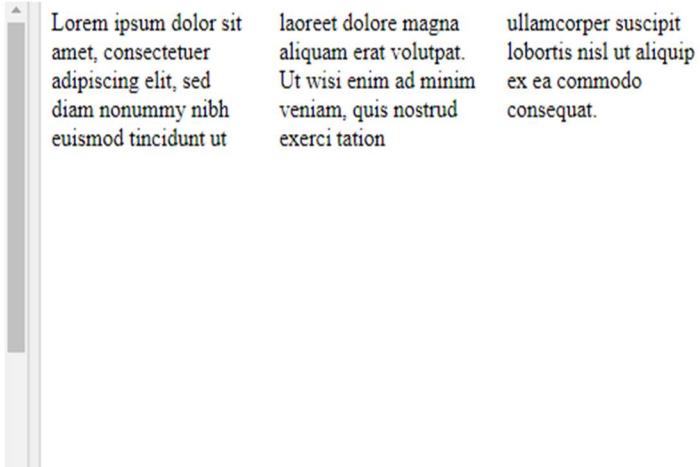
Columns

تعد خاصية الأعمدة خاصية **shorthand property** مختصرة لخصائص عرض العمود وعدد الأعمدة

[column-width + column-count]

Default value: auto auto

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
.newspaper1 {  
    -webkit-columns: 100px 3; /* Chrome, Safari, Opera */  
    -moz-columns: 100px 3; /* Firefox */  
    columns: 100px 3;  
}  
  
.newspaper2 {  
    -webkit-columns: 50px 4; /* Chrome, Safari, Opera */  
    -moz-columns: 50px 4; /* Firefox */  
    columns: 50px 4;  
}  
</style>
```



column-fill

تحدد خاصية **column-fill** كيفية تعبئة الأعمدة ، متوازنة أو لا
نصيحة: إذا أضفت ارتفاعاً إلى عنصر **multi-column**، يمكنك التحكم في
كيفية تعبئة المحتوى للأعمدة. يمكن أن يكون المحتوى متوازنًا أو ممتلئًا بالترتيب.

Property Values

: Balance

القيمة الافتراضية. يملا كل عمود بنفس المقدار من المحتوى ، ولكن لن يسمح للأعمدة أن تكون أطول من الارتفاع (لذلك ، قد تكون الأعمدة أقصر من الارتفاع حيث يقوم المتصفح بتوزيع المحتوى بشكل أفقي بالتساوي)

: Auto

يملا كل عمود حتى يصل إلى الارتفاع ، ويفعل ذلك إلى أن ينفد المحتوى (لذلك ، لن تملأ هذه القيمة بالضرورة جميع الأعمدة ولا تملأها بالتساوي)

```
<style>
div {
    -webkit-column-count: 3; /* Chrome, Safari, Opera */
    -moz-column-count: 3; /* Firefox */
    column-count: 3;
    height: 100px;
}
.newspaper1 {column-fill: auto;}
</style>
</head>
<body>
<h1>The column-fill Property</h1>
<h2>column-fill: auto:</h2>
<div class="newspaper1">
    Lorem ipsum dolor sit amet, consectetuer 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...
</div>
```

The column-fill Property

column-fill: auto:

Lorem ipsum dolor sit amet, consectetuer 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...	Lorem ipsum dolor sit amet, consectetuer 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...	Lorem ipsum dolor sit amet, consectetuer 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...
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Animation

وهو تحريك العناصر
تسمح CSS لـ **animation** بتصوير معظم عناصر HTML بدون استخدام Flash! أو JavaScript

من أهم وأقوى دروس ال CSS3 لأمكانياته الممتازة والحيوية في الموقع

What are CSS Animations?

تسمح الرسوم المتحركة بتغيير عنصر تدريجياً من نمط إلى آخر.

يمكنك تغيير العديد من خصائص CSS التي تريدها ، مثلما تريد.
لاستخدام الرسوم المتحركة لـ CSS ، يجب عليك أولاً تحديد بعض الإطارات المفتاحية للرسم المتحرك.

تحمل الإطارات الأساسية الأنماط التي سيحتويها العنصر في أوقات معينة.

نقاط

- ١ - الأفضل تسمية الأنميشن باسم الحركة نفسها التي يقوم بعملها وهذا لأنك ممكن تقوم بعمل أكثر من أنميشن فلا يحدث تشتبث وقتها
- ٢ - إذا كان كل مرحلة من مراحل الأنميشن لها نفس الخواص فنستطيع جمعها مع بعض وبإضافة كوما ،

```
0%, 100% {  
    top: 0;  
    left: 0  
}
```

The @keyframes Rule

عند تحديد أنماط CSS داخل القاعدة `@keyframes` ، سيتغير الرسم المتحرك تدريجياً من النمط الحالي إلى النمط الجديد في أوقات معينة للحصول على رسم متحرك ليعمل ، يجب عليك ربط الرسم المتحرك بعنصر..

لكي يعمل ال `@keyframes` فلا بد له من ثلاثة أشياء:

Required `animation-name` - ١

Required `animation-duration` - ٢

Required `css attribute` - ٣

: اسم الأنميشن **Animation-name**

تحدد خاصية `animation-name` المتحركة اسمًا لحركة `keyframes` المتحركة.

: Syntax

: يحدد اسم الإطار الرئيسي الذي تريد ربطه بالمتعدد **Keyframename**

: القيمة الافتراضية. يحدد أنه لن يكون هناك رسوم متحركة (يمكن استخدامه لإلغاء الرسوم المتحركة القادمة من السلسلة المتتالية) **none**

: وقت الأنميشن **Animation-duration**

تحدد مدة `animation-duration` المدة التي يجب أن تستغرقها الرسوم المتحركة لإكمال دورة واحدة.

: Syntax

: يحدد المدة الزمنية التي يجب أن تستغرقها الرسوم المتحركة لإكمال دورة واحدة. يمكن تحديد هذا بالثواني أو بالملي ثانية **seconds or milliseconds**. القيمة الافتراضية هي `0` ، مما يعني عدم حدوث أي رسوم متحركة **Time**

@keyframes في المثال أنشأ قمنا بإضافة `css attribute` داخل `@keyframes` نلاحظ

```

<style>
div {
    width: 100px;
    height: 100px;
    background-color: red;
    animation-name: example;
    animation-duration: 4s;}
/* Standard syntax */
@keyframes example {
    from {background-color: red;}
    to {background-color: yellow;}}
</style>
</head>
<body>
<div></div>

```

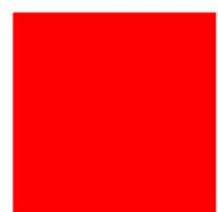


في المثال القادم ، حددنا متى سيتغير النمط باستخدام الكلمات الرئيسية "from" و "to" (التي تمثل 0% (start) و 100% (complete)). من الممكن أيضًا استخدام نسبة المئوية. باستخدام النسبة المئوية ، يمكنك إضافة العديد من تغييرات النمط كما تريده.

```

<head>
<style>
div {
    width: 100px;
    height: 100px;
    background-color: red;
    animation-name: example;
    animation-duration: 4s;}
/* Standard syntax */
@keyframes example {
    0% {background-color: red;}
    25% {background-color: yellow;}
    50% {background-color: blue;}
    100% {background-color: green;}}
</style>
</head>
<body>
<div></div>

```



اتجاه الأنيميشن : Animation-direction

تحدد خاصية **animation-direction** الرسوم المتحركة ما إذا كان يجب تشغيل الرسوم المتحركة إلى الأمام أو الخلف أو في دورات بديلة.

: Syntax

Normal : القيمة الافتراضية. يتم لعب الرسوم المتحركة كالمعتاد (لالأمام)

reverse : معكوس يمشي عكس القيم الطبيعية (الخلف)

alternate : يتم تشغيل الرسوم المتحركة للأمام أولاً ثم إلى الوراء

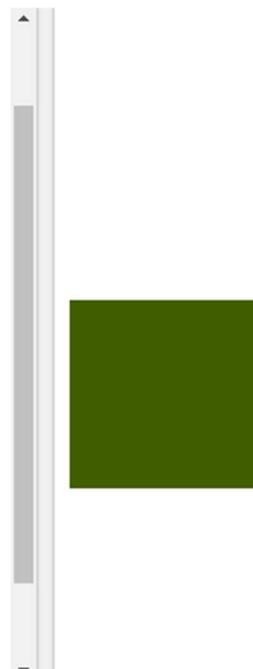
البديل وهو يمشي طبيعي وبعد أن ينتهي يمشي في عكس الإتجاه

Alternate-reverse : يتم تشغيل الرسوم المتحركة للوراء أولاً ، ثم إلى الأمام

وهو يمشي في عكس الإتجاه ثم يمشي طبيعي ثم يمشي في عكس الإتجاه وهكذا

ملاحظة: تحدد مدة الرسوم المتحركة المدة الزمنية التي يجب أن تستغرقها الرسوم المتحركة لإكمالها. إذا لم يتم تحديد خاصية مدة الرسوم المتحركة ، فلن تحدث أي رسوم متحركة ، لأن القيمة الافتراضية هي (0 seconds)..

```
<style>
div {
    width: 100px;
    height: 100px;
    background-color: red;
    position: relative;
    animation-name: example;
    animation-duration: 4s;
    animation-direction: reverse;}
@keyframes example {
    0% {background-color:red; left:0px; top:0px; }
    25% {background-color:yellow; left:200px; top:0px; }
    50% {background-color:blue; left:200px; top:200px; }
    75% {background-color:green; left:0px; top:200px; }
    100% {background-color:red; left:0px; top:0px; }
}
</style>
</head>
<body>
<div></div>
```



Animation-iteration-count : تكرار الأنميشن

تحدد خاصية **animation-iteration-count** التكرار عدد المرات التي يجب تشغيل الرسم المتحرك فيها.

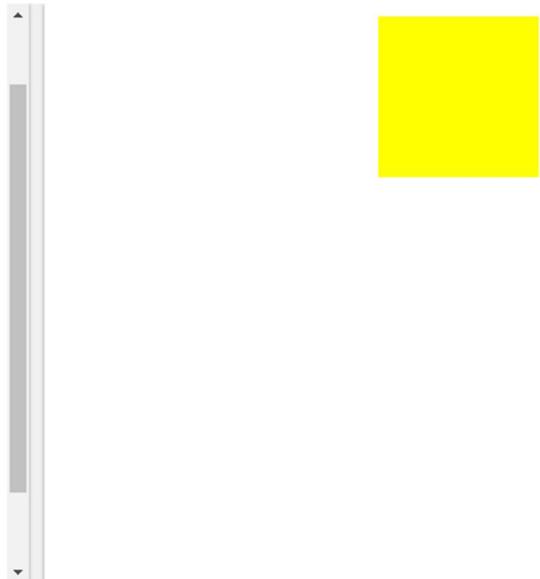
: Syntax

رقم يحدد عدد مرات تشغيل الرسم المتحرك. القيمة الافتراضية هي 1

يمكن استخدام الأرقام لاختيار مرات التكرار

يحدد أنه يجب تشغيل الرسوم المتحركة بلا حدود (لأبد) ومكان اختيار خاصية **infinite** لعمل تكرار لا نهائي أو مستمر وتستخدم في ال loading

```
<style>
div {
  width: 100px;
  height: 100px;
  background-color: red;
  position: relative;
  animation-name: example;
  animation-duration: 4s;
  animation-iteration-count: 3;
}
@keyframes example {
  0% {background-color:red; left:0px; top:0px;}
  25% {background-color:yellow; left:200px; top:0px;}
  50% {background-color:blue; left:200px; top:200px;}
  75% {background-color:green; left:0px; top:200px;}
  100% {background-color:red; left:0px; top:0px;}
}
</style>
</head>
<body>
<div></div>
```



Animation-delay : تأخير الأنميشن

تحدد خاصية **animation-delay** المتحركة تأخيرًا لبدء الرسوم المتحركة seconds (s) or milliseconds يتم تعريف قيمة تأخير الحركة في (ms).

Default value : 0

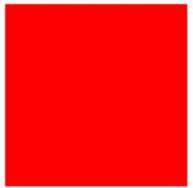
وهو يعني انتظار وقت معين ثم قم بعمل الأنميشن وهو يقبل الثانية والملي ثانية seconds (s) or milliseconds (ms)

: Syntax

: Time

اختياري. لتحديد عدد الثواني (ثوانٍ) أو مللي ثانية (ملاي ثانية) للانتظار قبل بدء الرسوم المتحركة.
القيمة الافتراضية هي `0`. يُسمح بالقيم السلبية. إذا كنت تستخدم القيم السالبة ، فسيبدأ الرسم المتحرك كما لو كان قد تم تشغيله للثواني `N` مللي ثانية /

```
<style>
div {
    width: 100px;
    height: 100px;
    background: red;
    position: relative;
    animation: mymove 5s 1;
    animation-delay: -2s;}
@keyframes mymove {
    from {left: 0px;}
    to {left: 200px;}}
</style>
</head>
<body>
<div></div>
```



: Animation-fill-mode

تحدد خاصية `animation-fill-mode` نمطًا للعنصر عند عدم تشغيل الرسم المتحرك (قبل أن يبدأ ، أو بعد أن ينتهي ، أو كلاهما) لا تؤثر الرسوم المتحركة لـ CSS على العنصر قبل تشغيل الإطار الرئيسي الأول أو بعد تشغيل الإطار الرئيسي الأخير. يمكن أن تتجاوز خاصية وضع ملء الصور المتحركة هذا السلوك.

: Syntax

: none : القيمة الافتراضية. لن تطبق الرسوم المتحركة أي أنماط على العنصر قبل أو بعد تنفيذه

: forwards : سيحتفظ العنصر بقيم النمط التي تم تعبيئها بواسطة الإطار الرئيسي الأخير (يعتمد على اتجاه الحركة والرسوم المتحركة - تكرار التكرار)

وهو يقف عند آخر keyframe له

: backwards : سيحصل العنصر على قيم النمط التي تم تعبيئها بواسطة الإطار المفتاحي الأول (يعتمد على اتجاه الحركة) ، ويحتفظ بذلك أثناء فترة تأخير الرسوم المتحركة

وهو يقوم بأخذ أول keyframe مهما كنت كاتب delay ويكمِل الأنيميشن ثم يقف في موضعه الأصلي

both: ستتبع الرسوم المتحركة القواعد لكل من الأمام والخلف ، مع توسيع خصائص الرسوم المتحركة في كلا الاتجاهين

يأخذ أول keyframe ثم يقف عند آخر keyframe

```
<style>
div {
    width: 100px;
    height: 100px;
    background: red;
    position: relative;
    animation: mymove 3s;
    animation-delay: 2s;
    animation-fill-mode: backwards; }

@keyframes mymove {
    from {top: 0px; background-color: yellow;}
    to {top: 200px;}}
</style>
</head>
<body>
<div></div>
</body>
```



: Animation-play-state

تحدد خاصية **animation-play-state** ما إذا كان الرسم المتحرك قيد التشغيل أو متوقفاً مؤقتاً.

ملاحظة: استخدم هذه الخاصية في جافا سكريبت لإيقاف رسم متحرك مؤقتاً في منتصف الدورة.

: Syntax

running : القيمة الافتراضية. يحدد أن الرسوم المتحركة قيد التشغيل

وهي تشغيل الأنيميشن

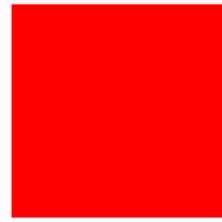
paused : يحدد أن الرسم المتحرك متوقف مؤقتاً

وهي إيقاف الأنيميشن Paused

```

<style>
div {
    width: 100px;
    height: 100px;
    background: red;
    position: relative;
    animation: mymove 5s infinite;
}
div:hover {animation-play-state: paused;}
@keyframes mymove {
    from {left: 0px;}
    to {left: 200px;}
}
</style>
</head>
<body>
<div></div>

```



Animation-timing-function

تحدد وظيفة **animation-timing-function** منحنى السرعة للصورة المتحركة.

يحدد منحنى السرعة الوقت الذي يستخدمه الرسم المتحرك للتغيير من مجموعة واحدة من أنماط CSS إلى أخرى.

يتم استخدام منحنى السرعة لإجراء التغييرات بسلامة.

: Syntax

linear : تتمتع الرسوم المتحركة بنفس السرعة من البداية إلى النهاية يبدأ وينتهي بنفس السرعة دون تغير

ease : القيمة الافتراضية. تتمتع الرسوم المتحركة ببداية بطيئة ، ثم سريعة ، قبل أن تنتهي ببطء

يبدأ ببطء ثم سريعا ثم ينتهي ببطء وهو **default**

ease-in : الرسوم المتحركة لها بداية بطيئة

يبدأ ببطء ثم تمشي طبيعي ثم تنتهي سريعا

ease-out : الرسوم المتحركة لها نهاية بطيئة

تبدأ سريعا ثم تنتهي ببطء

ease-in-out : تحتوي الرسوم المتحركة على بداية بطيئة ونهاية بطيئة
يبدأ وينتهي ببطء

:step-start مكافئ للخطوات (١ ، بداية)

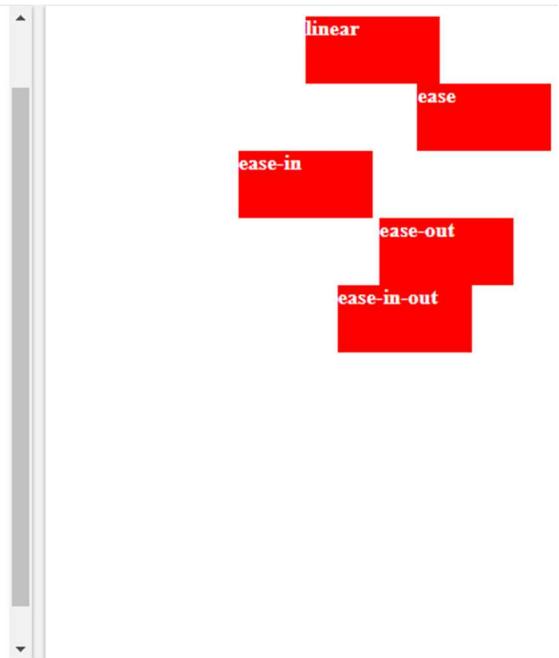
:step-end مكافئ للخطوات (١ ، النهاية)

:step يحدد وظيفة خطوة ، مع معلمتين . تحدد المعلمة الأولى عدد الفواصل الزمنية في الدالة . يجب أن يكون عدداً صحيحاً موجباً (أكبر من ٠). المعلمة الثانية ، وهي اختيارية ، هي إما قيمة "start" أو "end" ، وهي تحدد النقطة التي يحدث عنها تغيير القيم خلال الفترة الزمنية إذا تم حذف المعلمة الثانية ، يتم إعطاء القيمة "end"

يبدأ على خطوات وينتهي على خطوات دون Smooth

:Cubic-bezier تحديد القيم الخاصة بك في وظيفة بيري مكعب القيم المحتملة هي قيم رقمية من ٠ إلى ١

```
<style>
div {
  width: 100px;
  height: 50px;
  background: red;
  color: white;
  font-weight: bold;
  position: relative;
  animation: mymove 5s infinite;
}
#div1 {animation-timing-function: linear;}
#div2 {animation-timing-function: ease;}
#div3 {animation-timing-function: ease-in;}
#div4 {animation-timing-function: ease-out;}
#div5 {animation-timing-function: ease-in-out;}
@keyframes mymove {
  from {left: 0px;}
  to {left: 300px;}}
</style>
</head>
<body>
<div id="div1">linear</div>
<div id="div2">ease</div>
<div id="div3">ease-in</div>
<div id="div4">ease-out</div>
<div id="div5">ease-in-out</div>
</body>
</html>
```



Animation Shorthand Property

الكود المختصر Animation

Default value: none 0 ease 0 1 normal none running;

animation: name duration timing-function delay iteration-count direction fill-mode play-state;

يمكن كتابة كل كود منفرد

```
div {  
    animation-name: example;  
    animation-duration: 5s;  
    animation-timing-function: linear;  
    animation-delay: 2s;  
    animation-iteration-count: infinite;  
    animation-direction: alternate;  
}
```

ويمكن كتابته في كود واحد

```
div {  
    animation: example 5s linear 2s infinite alternate;  
}
```

ويمكنك تجرب هذا في مثال بنفسك.

موقع للتدريب على الأنيميشن

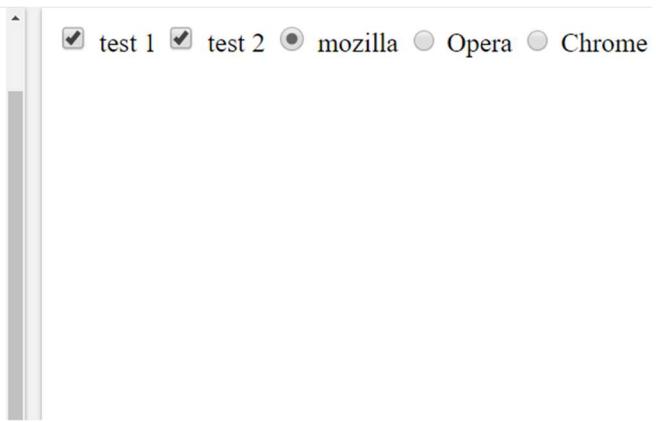
<https://coveloping.com/tools/css-animation-generator>

[/http://ianlunn.github.io/Hover](http://ianlunn.github.io/Hover)

Selectors

✓ وهي قم بإضافة علامة للمرربع **:Checked**

وهي خاصية تظهر في ال form وهي تطبق خواص ال css داخل ال form التي تم عمل لها checked



```
<html>
<style>
:checked {height: 15px; width: 15px;}
</style>
</head>
<body>
<form class="osama">
<input type="checkbox"> test 1
<input type="checkbox"> test 2
<input type="radio" name="browser"> mozilla
<input type="radio" name="browser"> Opera
<input type="radio" name="browser"> Chrome
</form>
</body>
```

وإذا تم تحديد ال input type=checkbox فلن يقوم بالتطبيق إلا عليه ولن يطبق

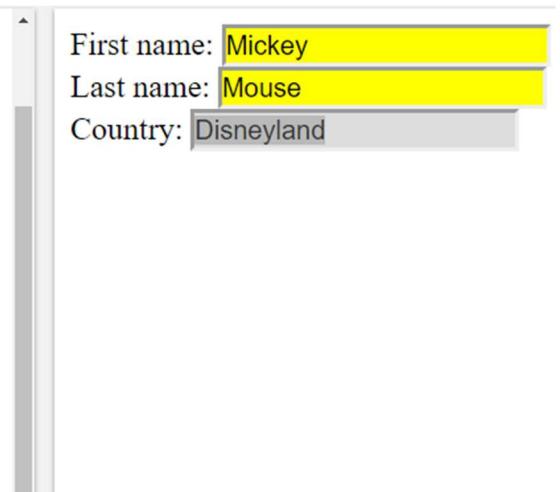
```
input[type="checkbox"]:checked {
    width: 100px
}
```

على ال radio

نلاحظ: نستطيع تحديده بال class .osama :checked { width: 100px }

text وهي الخاصية المعروفة غير مسموح بالكتابة بداخل **:disabled**
فأي شيء disabled اجعل خلفيته رمادي

وهي أي خاصية تمكين استخدام الحقل enabled اجعل لي خلفيتها **:enabled**
صفاء



```
<html>
<style>
input[type=text]:enabled {background: #ffff00;}
input[type=text]:disabled {background: #dddddd;}
</style>
</head>
<body>
<form action="">
First name: <input type="text" value="Mickey"><br>
Last name: <input type="text" value="Mouse"><br>
Country: <input type="text" disabled="disabled"
value="Disneyland">
</form>
</body>
```

First name: Mickey
Last name: Mouse
Country: Disneyland

ونستطيع كتابة disabled مباشرة هكذا

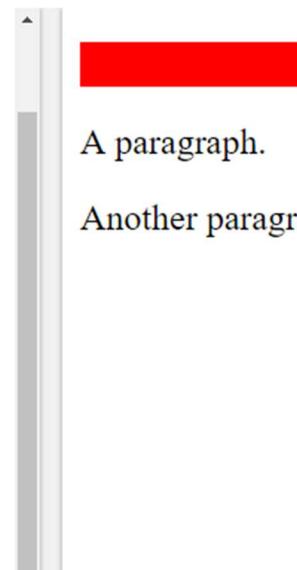
```
Country: <input type="text" disabled value="Disneyland">
```

ونضع أيضاً السيلكتور هذا

ووهذه معناها أي سيلكتور disabled اجعل له خلفية رمادي وهذا.

أي العناصر الفارعة طبق عليها :Empty

```
<head>
<style>
p:empty {
    width: 100px;
    height: 20px;
    background: #ff0000; }
</style>
</head>
<body>
<p></p>
<p>A paragraph.</p>
<p>Another paragraph.</p>
</body>
</html>
```



فإي عنصر فارغ طبق عليه هذه الخصائص attribute

ونستطيع تطبيقها والإستفادة منها في الجداول
وهكذا table :empty td

Not, Last-Child, First-Of-Type

:not(selector) وهي تعني لا تطبق هذه الخصائص على العنصر الذي قمت بتحديده

```
<head>
<style>
p {color: #000000;}
:not(p) {color: #ff0000; }
</style>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<div>This is some text in a div element.</div>
<a href="https://www.w3schools.com" target="_blank">Link to W3Schools!</a>
</body>
```

This is a heading

This is a paragraph.

This is another paragraph.

This is some text in a div element.
[Link to W3Schools!](https://www.w3schools.com)

نلاحظ: أنه لم يطبق هذه الخصائص على البرجراf (p) وطبقها على كل العناصر الأخرى

ملاحظة: وينفع استخدام class أيضاً بمعنى لا تطبق هذا على أي كلاس

```
section :not(.test) {  
    color: #F00  
}
```

أي طبق الخصائص على أول طفل من الأب فقط : **First-Child**

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p:first-child {  
    background-color: yellow;  
</style>  
</head>  
<body>  
<p>This paragraph is the first child of its parent (body).</p>  
<h1>Welcome to My Homepage</h1>  
<p>This paragraph is not the first child of its parent.</p>  
<div>  
    <p>This paragraph is the first child of its parent (div).</p>  
    <p>This paragraph is not the first child of its parent.</p>  
</div>  
</body>  
</html>
```

This paragraph is the first child of its parent (body).

Welcome to My Homepage

This paragraph is not the first child of its parent.

This paragraph is the first child of its parent (div).

This paragraph is not the first child of its parent.

Note: For :first-child to work in IE8 and earlier, a DOCTYPE must be declared.

طبق الخصائص على آخر طفل من الأب فقط : **Last-Child**

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p:last-child {background: #ff0000;}  
</style>  
</head>  
<body>  
<p>The first paragraph.</p>  
<p>The second paragraph.</p>  
<p>The third paragraph.</p>  
<p>The fourth paragraph.</p>  
</body>  
</html>
```

The first paragraph.

The second paragraph.

The third paragraph.

The fourth paragraph.

First-Of-Type يعني الأول من نوعه

```
<!DOCTYPE html>
<html>
<head>
<style>
p:first-of-type {background: red;}
</style>
</head>
<body>
<p>The first paragraph.</p>
<p>The second paragraph.</p>
<p>The third paragraph.</p>
<p>The fourth paragraph.</p>
</body>
</html>
```

The first paragraph.

The second paragraph.

The third paragraph.

The fourth paragraph.

نلاحظ : أنه طبق الخاصية على ال p الأول فقط من نوعه ولم يطبقها على باقي p

الفرق بين First-Of-Type & First-Child

First-Child : يطبق الخصائص على أول طفل في ترتيبه الأول داخل الأب

First-of-type : يطبق الخصائص على أول طفل من نوعه حتى لو كان يسبقه عناصر أخرى في الترتيب داخل الأب

Last-Of-Type

وهو الأخير من نوعه وهو عكس first-of-type تماماً

```
<!DOCTYPE html>
<html>
<head>
<style>
p:last-of-type {background: #ff0000;}
</style>
</head>
<body>
<h1>This is a heading</h1>
<p>The first paragraph.</p>
<p>The second paragraph.</p>
<p>The third paragraph.</p>
<p>The fourth paragraph.</p>
<...>
```

This is a heading

The first paragraph.

The second paragraph.

The third paragraph.

The fourth paragraph.

الطفل الوحد Only-Child

وهي خاصية تستخدم بمعنى طبق لـ `li` الخصائص على العنصر الوحد في الأب الذي لا يشترك فيه معه عناصر أخوات له

```
<!DOCTYPE html>
<html>
<head>
<style>
p:only-child {background: #adadad;}
</style>
</head>
<body>
<div><p>ابن وحيد</p></div>
<div><span>أخ</span><span>أخ</span></div>
</body>
</html>
```

ابن وحيد
أخ
أخ
أخ

نلاحظ: أنه طبق على `p` الوحيدة في `div` الأول خواص ال `css` ولم يطبق على `p` في `div` الثانية والتي يشترك معها عناصر أخرى `span+div`

الطفل الوحد من نوعة Only-Of-Type

ومعناها أنه يتم تطبيق خواص ال `CSS` على الطفل الوحد من نوعه وهو الذي لا يتكرر معه نفس الأخ من نفس نوعه مثل (`div + p + span`)
فكلهم وقتها وحيد من نوعه أما (`p + p`) فهذا أخ من نوعه فلا يكون وحيد من نوعه وقتها

```
<!DOCTYPE html>
<head>
<style>
p:only-of-type {background: #adadad;}
</style>
</head>
<body>
<div><p>وحيد من نوعة</p><span>أخ</span></div>
<div><p>أخ من نوعه</p><p>أخ من نوعه</p></div>
</body>
</html>
```

وحيد من نوعة
أخ
أخ من نوعه
أخ من نوعه

نلاحظ : أنه قام بتنسيق `p` الوحيد من نوعه في `div` الأول مع وجود `span` معه

أما ال p في ال div الثانية لم يقم بتنسيقها لوجود p أخرى من نوعه وبالتالي فلم تجعله وحيداً من نوعه.

التحديد : Selection

وهو التحديد في المتصفح عندما تقوم بعمل تحديد للنص في المتصفح فإن له لون ثابت تستطيع تغيير هذا اللون الثابت بهذه الخاصية ولكن نضيف لها أربع نقاط لا نقطتين كالعادة.

```
<head>
<style>
::-moz-selection { color: red; background: #a9a9a9; }
::selection {color: red;background: #a9a9a9; }
</style>
</head>
<body>
<h1>Select some text on this page:</h1>
<p>This is a paragraph.</p>
<div>This is some text in a div element.</div>
<p><strong>Note:</strong> ::selection is not supported in Internet Explorer 8 and earlier versions.</p>
<p><strong>Note:</strong> Firefox supports an alternative, the ::-moz-selection property.</p>
</body>
```

Select some text on this page:

This is a paragraph.

This is some text in a div element.

Note: ::selection is not supported in Internet Explorer 8 and earlier versions.

Note: Firefox supports an alternative, the ::-moz-selection property.

Nth Nth-Child

وهي خاصية **selectors** تستطيع بها تحديد العنصر الأبن الذي تريد تطبيق خصائص ال CSS عليه، وتستطيع هذا إما بتحديد رقم العنصر **number** في الترتيب في الأب أو بالكلمات المفتاحية **keyword** أو بالمعادلة **formula**

١ - رقم العنصر **number**

```
<html>
<head>
<style>
p:nth-child(3) {background: #ababab;}
</style>
</head>
<body>
<p>1</p>
<p>2</p>
<p>3</p>
<p>4</p>
<p>5</p>
<p>6</p>
</body>
</html>
```

1
2
3
4
5
6

ملاحظة: لا داعي لاستخدام **nth** لتحديد العنصر الأول أو الأخير لأنه يوجد خاصية **last-child** و**first-child** وهي مخصصة لهذا.

لا يشترط تحديد العنصر قبل **nth-child** بل يمكن كتابتها مباشرة وهي تقوم بتحديد العنصر الطفل داخل الأب المراد

[even + odd] ولها keyword – ٢

odd : وهو يقوم بتحديد الأولى ويترك الثانية ويحدد الثالثة ويترك الرابعة وهذا يقوم بالتحديد بنظام (١ و ٣ و ٥ و ٧ و ٩ و إلى آخره)

even : هو عكس **odd**
فهو يبدأ بالعنصر الثاني ويتجاهل الأول ثم يتتجاهل الثالث ويحدد الرابع ثم يتتجاهل الخامس ويحدد السادس

فهو يعمل بنظام (٢ و ٤ و ٦ و ٨ و ١٠ و وهذا)

/ مثال /

```

<head>
<style>
p:nth-child(odd) {background: #a5a5a5;}
p:nth-child(even) {background: #e6ffca;}
</style>
</head>
<body>
<p>1</p>
<p>2</p>
<p>3</p>
<p>4</p>
<p>5</p>
<p>6</p>
</body>

```



Nth-last-child : وهي للعد العكسي من الأسفل للأعلى

```

<!DOCTYPE html>
<html>
<head>
<style>
p:nth-last-child(2) {background: #b7b7b7;}
</style>
</head>
<body>
<p>1</p>
<p>2</p>
<p>3</p>
<p>4</p>
</body>
</html>

```

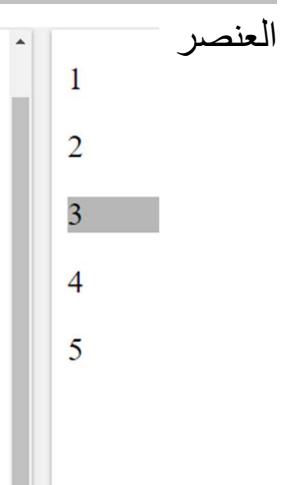


Nth-last-of-type : وهي للعد العكسي من الأسفل للأعلى ولكن من نفس نوع

```

<html>
<head>
<style>
p:nth-last-of-type(2) {background: #b7b7b7;}
</style>
</head>
<body>
<p>1</p>
<p>2</p>
<p>3</p>
<p>4</p>
<span>5</span>
</body>

```



نلاحظ : في هذا المثال نقول له انتقل إلى p رقم ٢ من نوعه من الأسفل وهو لا يقوم بعد span معه

Nth-of-type : وهي يعني انتقل إلى الطفل من نوعه

وفي هذا المثال نقول له انتقل إلى div رقم 10 من نوعه أي أنه لا يقوم بعد ال p

```
<!DOCTYPE html>
<html>
<head>
<style>
p:nth-of-type(2) {background: #b1b1b1;}
</style>
</head>
<body>
<p>1</p>
<p>2</p>
<p>3</p>
<p>4</p>
<p>5</p>
</body>
```

1
2
3
4
5

الفرق بين nth-child & nth-last-child & nth-last-of-type

nth-child : يقوم بالعد من الأعلى إلى الأسفل دون التقييد بنوع العنصر الأبن

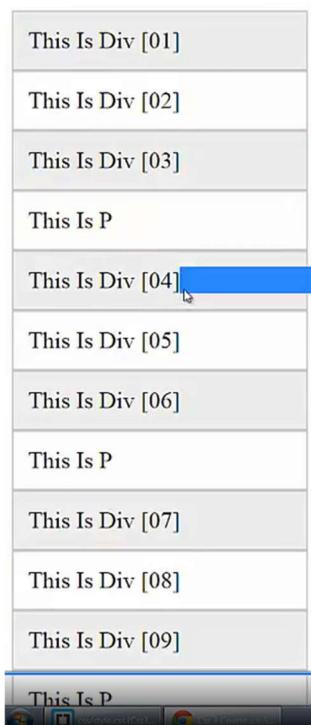
nth-last-child: يقوم بالعد من الأسفل إلى الأعلى دون التقييد بنوع العنصر الأبن ولكن لا يطبق خصائص CSS إلا على العنصر المستهدف

nth-last-of-type : يقوم بالعد من الأسفل إلى الأعلى ولكن من نفس نوع العنصر الأبن

الفرق بين nth-child & nth-of-type

nth-child : يقوم بعد العناصر الأبناء حتى لو لم تكن من نفس النوع

Nth-of-type : يقوم بعد العناصر الأبناء ولكن من نفس نوعها

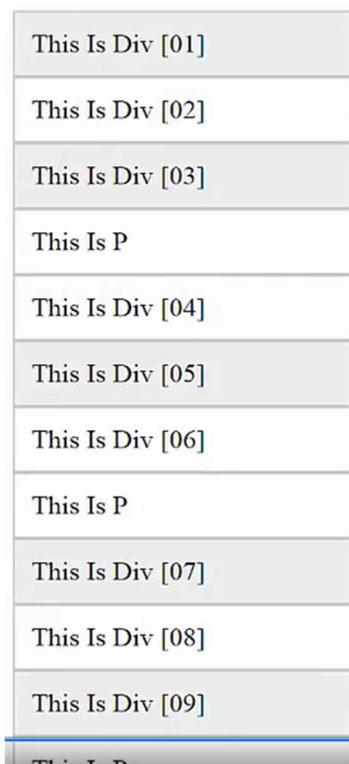


```
div, p {  
    padding: 10px;  
    width: 200px;  
    border: 1px solid #CCC;  
    margin: 0  
}  
  
div:nth-child(odd) {  
    background-color: #EEE  
}
```

```
<body>  
<div>This Is Div [01]</div>  
<div>This Is Div [02]</div>  
<div>This Is Div [03]</div>  
<p>This Is P</p>  
<div>This Is Div [04]</div>  
<div>This Is Div [05]</div>  
<div>This Is Div [06]</div>  
<p>This Is P</p>  
<div>This Is Div [07]</div>  
<div>This Is Div [08]</div>  
<div>This Is Div [09]</div>  
<p>This Is P</p>  
<div>This Is Div [10]</div>  
</body>
```

نلاحظ أن **nth-child** : يبدء في عد الأبناء من div+p أيضاً

رغم أنه تم تحديد div في كتابة الكود في CSS



```
div:nth-of-type(odd) {  
    background-color: #EEE  
}
```

نلاحظ هنا **nth-of-type** : يبدء في عد العناصر الأبناء

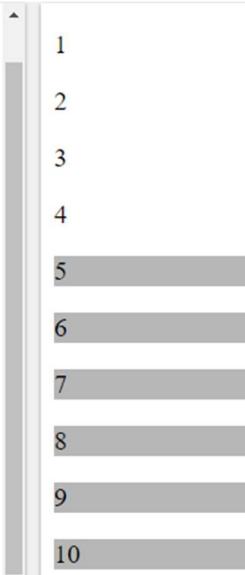
من نفس نوعها متجاهلاً ال p كأنه غير موجود تماماً

المعادلة الأولى formula- ٣

(n+number):

هذه المعادلة تقول فيها أبد لي من رقم العنصر الأبن كذا إلى النهاية

```
<html>
<head>
<style>
p:nth-child(n+5) {background: #b7b7b7;}
</style>
</head>
<body>
<p>1</p>
<p>2</p>
<p>3</p>
<p>4</p>
<p>5</p>
<p>6</p>
<p>7</p>
<p>8</p>
<p>9</p>
<p>10</p>
</body>
</html>
```

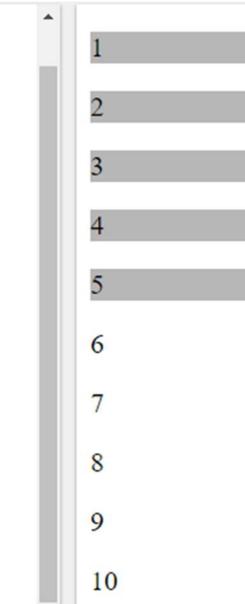


نلاحظ: أنه بدأ العد من العنصر الأبن الخامس حتى النهاية

سنلاحظ: عند استخدام nth-of-type نحصل على نفس النتيجة ولكن هذا لأن هذه الخاصية تقوم بعد العنصر الأبن من نوعة ولا تقوم بعد ال p

المعادلة الثانية : العد السلبي (-n+number) وهي العد العكسي

```
<html>
<head>
<style>
p:nth-child(-n+5) {background: #b7b7b7;}
</style>
</head>
<body>
<p>1</p>
<p>2</p>
<p>3</p>
<p>4</p>
<p>5</p>
<p>6</p>
<p>7</p>
<p>8</p>
<p>9</p>
<p>10</p>
</body>
</html>
```



نلاحظ: أنه قد قام بالعد من العنصر الأول رقم 5 ثم تراجع حتى رقم 1

المعادلة الثالثة: $(2n+1)$

شرح المعادلة /

$$\begin{array}{r|rrr} 2x_n + 1 & | & 2x_0 = 0 + 1 = [1] \\ 2x_n + 1 & | & 2x_1 = 2 + 1 = [3] \\ 2x_n + 1 & | & 2x_2 = 4 + 1 = [5] \\ 2x_n + 1 & | & 2x_3 = 6 + 1 = [7] \end{array}$$

حيث أن $(1, 3, 5, 7, \dots)$ هي العناصر الأبناء المراد تحديدهم وهم ناتج المعادلة التي قمنا بكتابتها وهي $(2n+1)$

```
<html>
<head>
<style>
p:nth-child(2n+5) {background: #b7b7b7;}
</style>
</head>
<body>


1



2



3



4



5



6



7



8



9



10


</body>
</html>
```



فمعنى $(2n+1)$ أنه يبدأ العد بطريقة فردية إلى مالانهاية 1,3,5,7,9,11,13 .. وهكذا ..

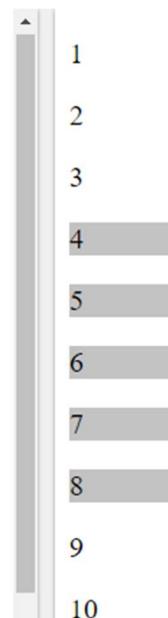
نستطيع تغيير المعادلة فهي ليست ثابتة ممكناً نكتب $(3n+1)$

$3xn+1$	$ $	$3 \times 0 = 0 + 1 = [1]$
$3xn+1$	$ $	$3 \times 1 = 3 + 1 = [4]$
$3xn+1$	$ $	$3 \times 2 = 6 + 1 = [7]$
$3xn+1$	$ $	$3 \times 3 = 9 + 1 = [10]$

وهكذا نستطيع تغيير المعادلة بأي رقم ونمشي على هذا الجدول وسوف نحصل على نتيجة التعديل

لإختيار مجموعة عناصر أطفال معينة : Range

```
<!DOCTYPE html>
<html>
<head>
<style>
p:nth-child(n+4):nth-child(-n+8) {background: #c3c3c3;}
</style>
</head>
<body>
<p>1</p>
<p>2</p>
<p>3</p>
<p>4</p>
<p>5</p>
<p>6</p>
<p>7</p>
<p>8</p>
<p>9</p>
<p>10</p>
</body>
```



ملاحظة: بهذه المعادلة نستطيع تحديد children

معينين نريد تحديدهم وأعطائهم خصائص معينة

`:nth-child(n+number):nth-child(-n+number)`

نلاحظ أننا نستخدم العد الإيجابي والعد العكسي معاً أو القيم السالبة والموجبة

Web Safe Font

يوجد أمنة متواجده font default في browsr ويوجد خطوط كثيره غير موجودة في browsr فما الحل لجعلها تظهر في الموقع بعد رفعه على النت لأنها قد تظهر على جهازك ولكنها غير موجودة في أجهزة الآخرين؟

" Web Safe Font

www.cssfontstack.com

وهي خطوط موجودة في كل نسخة ويندوز أو ماك أو أي نظام تشغيل بحيث تستطيع البحث ومعرفة هل الخط المستخدم في موقعك متواجد في نسخة الويندوز المختلفة أم لا.

ملاحظة: يجب أن تحتوي الخاصية font-family على العديد من أسماء الخطوط كنظام "احتياطي" ، لضمان أقصى توافق بين المستعرضات / أنظمة التشغيل. إذا كان المستعرض لا يدعم الخط الأول ، فإنه يحاول استخدام الخط التالي

ابدا بالخط الذي تريده ، وانتهى بعائلة عامة ، للسماح للمستعرض باختيار خط مشابه في العائلة العامة ، إذا لم تتوفر خطوط أخرى:

```
p {  
    font-family: "Times New Roman", Times, serif;  
}
```

font face

وهو يسمح لك بإستخدام font خارجي بدلا من web save font الموجودة في أنظمة التشغيل

```
@font-face {  
    font-family: OsamaElzero;  
    src: url('fonts/OsamaElzero.ttf')  
}
```

ولابد في ال @font face من وضع جميع أنواع الخط بكافة امتداده

[TTF, EOT, OTF, WOFF, WOFF2, SVG]: هناك أنواع للخطوط

```
TTF: True Type Font  
EOT: Embedded Type Font  
OTF: Open Type Font  
WOFF: Web Open Font  
WOFF2: Web Open Font 2  
SVG: Scalable Vector Graphic
```

```
/*  
 * Droid Arabic Kufi (Arabic) http://www.google.com/fonts/earlyaccess  
 */  
@font-face {  
    font-family: 'Droid Arabic Kufi';  
    font-style: normal;  
    font-weight: 400;  
    src: url("//fonts.gstatic.com/ea/droidarabickufi/v6/DroidKufi-Regular.eot");  
    src: url("//fonts.gstatic.com/ea/droidarabickufi/v6/DroidKufi-Regular.eot?#iefix") format('embedded-opentype'),  
        url("//fonts.gstatic.com/ea/droidarabickufi/v6/DroidKufi-Regular.woff2") format('woff2'),  
        url("//fonts.gstatic.com/ea/droidarabickufi/v6/DroidKufi-Regular.woff") format('woff'),  
        url("//fonts.gstatic.com/ea/droidarabickufi/v6/DroidKufi-Regular.ttf") format('truetype');  
}  
@font-face {  
    font-family: 'Droid Arabic Kufi';  
    font-style: normal;  
    font-weight: 700;  
    src: url("//fonts.gstatic.com/ea/droidarabickufi/v6/DroidKufi-Bold.eot");  
    src: url("//fonts.gstatic.com/ea/droidarabickufi/v6/DroidKufi-Bold.eot?#iefix") format('embedded-opentype'),  
        url("//fonts.gstatic.com/ea/droidarabickufi/v6/DroidKufi-Bold.woff2") format('woff2'),  
        url("//fonts.gstatic.com/ea/droidarabickufi/v6/DroidKufi-Bold.woff") format('woff'),  
        url("//fonts.gstatic.com/ea/droidarabickufi/v6/DroidKufi-Bold.ttf") format('truetype');  
}
```

نلاحظ هنا: في ملف CSS أنه `font-face`

استخدم اسم الخط

ثم وضع مسار الخط وهو في ()

وأنه استخدم أكثر من نوع لملف الخط بأكثر من امتداد

في ما يلي بعض مجموعات الخطوط الشائعة الاستخدام ، والتي تنظمها عائلة عامة.

Serif Fonts

font-family	Example text
Georgia, serif	This is a heading This is a paragraph
"Palatino Linotype", "Book Antiqua", Palatino, serif	This is a heading This is a paragraph
"Times New Roman", Times, serif	This is a heading This is a paragraph

Sans-Serif Fonts

font-family	Example text
Arial, Helvetica, sans-serif	This is a heading This is a paragraph
"Arial Black", Gadget, sans-serif	This is a heading This is a paragraph
"Comic Sans MS", cursive, sans-serif	This is a heading This is a paragraph
Impact, Charcoal, sans-serif	This is a heading This is a paragraph
"Lucida Sans Unicode", "Lucida Grande", sans-serif	This is a heading This is a paragraph
Tahoma, Geneva, sans-serif	This is a heading This is a paragraph
"Trebuchet MS", Helvetica, sans-serif	This is a heading This is a paragraph
Verdana, Geneva, sans-serif	This is a heading This is a paragraph

Monospace Fonts

font-family	Example text
"Courier New", Courier, monospace	This is a heading This is a paragraph
"Lucida Console", Monaco, monospace	This is a heading This is a paragraph

Media query

وهو مهم في عمل موقع Responsive

CSS3 هو تقنية CSS مقدمة في Media query

يستخدم قاعدة `@media` لتضمين كتلة من خصائص CSS فقط إذا كان الشرط معيناً صحيحاً

The screenshot shows a code editor interface with a toolbar at the top. The code in the editor is:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-color: lightgreen;
}

@media only screen and (max-width: 600px) {
    body {background-color: #cecece;}}
</style>
</head>
<body>


Resize the browser window. When the width of this document is 600 pixels or less, the background-color is "lightblue", otherwise it is "lightgreen".


</body>
</html>
```

The result preview on the right shows a green background. A note says: "Resize the browser window. When the width of this document is 600 pixels or less, the background-color is "lightblue", otherwise it is "lightgreen".". The result size is listed as 601 x 361.

نلاحظ في هذا المثال أن عرض الصفحة عندما يصل إلى 600px فسوف يتتحول للون الرمادي

```
@media only screen and (max-width: 600px) {
    body {
        background-color:#cecece;
    }
}
```

مراجعة

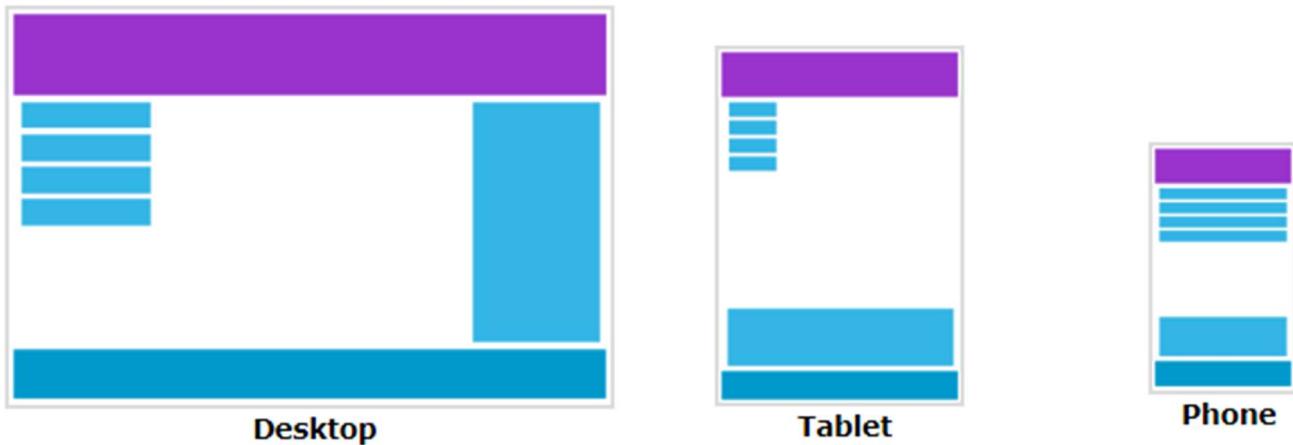
```
@media print {
    div {
        font-size: 100px;
        color: #F00
    }
}
```

عند استخدام media مع print

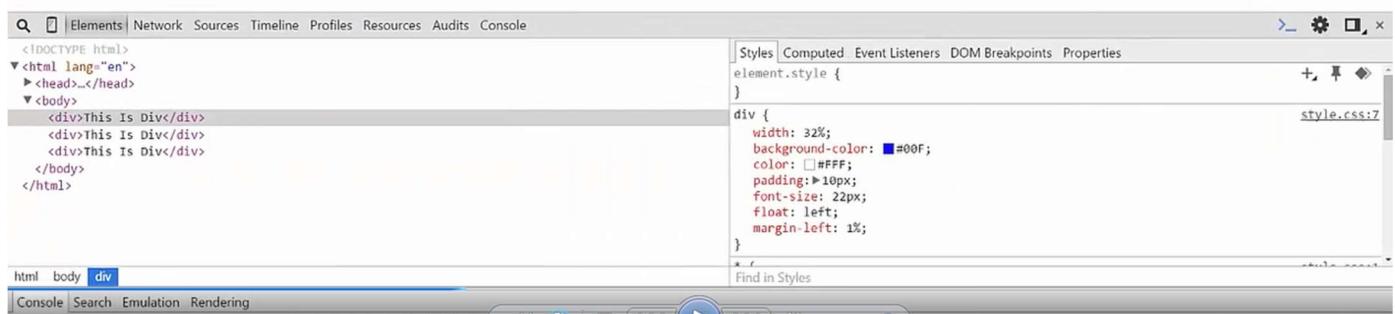
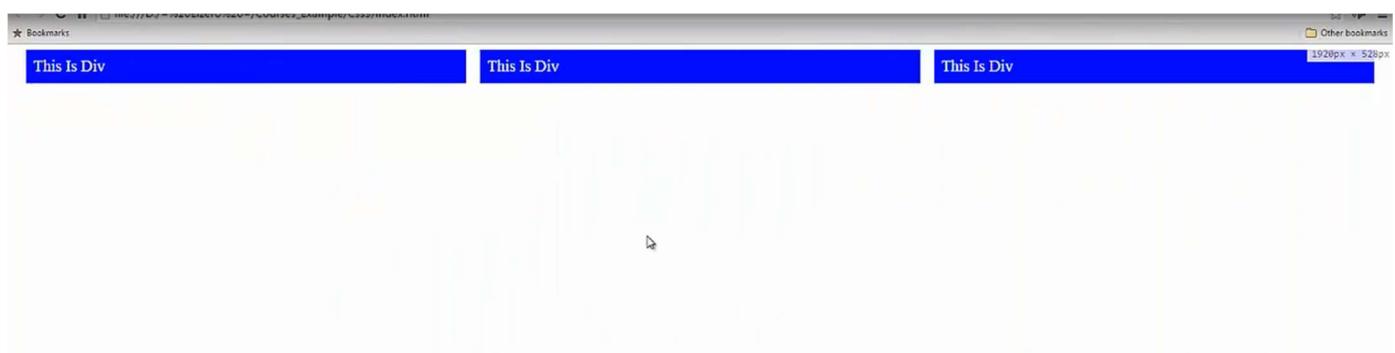
فإنه عند الطباعة سوف تكون بهذه الخصائص التي في الميديا

Add a Breakpoint

يمكن أن تساعد `@media` في إضافة نقطة توقف حيث تتصرف أجزاء معينة من التصميم بشكل مختلف على كل جانب من نقاط التوقف.



نلاحظ : أنه عندما وصل حجم الشاشة ل 500px تم تغيير خصائص ال CSS لها وعندما يزداد حجم الشاشة تعود لخصائصها الطبيعية في ال CSS التي تم وضعها لها



نلاحظ هنا التحكم في عرض الشاشة min-width & max-width:

التحكم في خصائص الشاشة عند الوصول لعرض معين

@media (min-width: 500px)

ومعناها أقل حجم للشاشة 500px فما أكثر فطبق عليه هذه الخصائص CSS

@media (max-width: 500px)

ومعناها أقصى حجم للشاشة هو 500px فما أقل فطبق عليه هذه الخصائص

مثال للتحكم في كبر وصغر حجم الشاشة وأخفاء وإظهار عناصر فيها

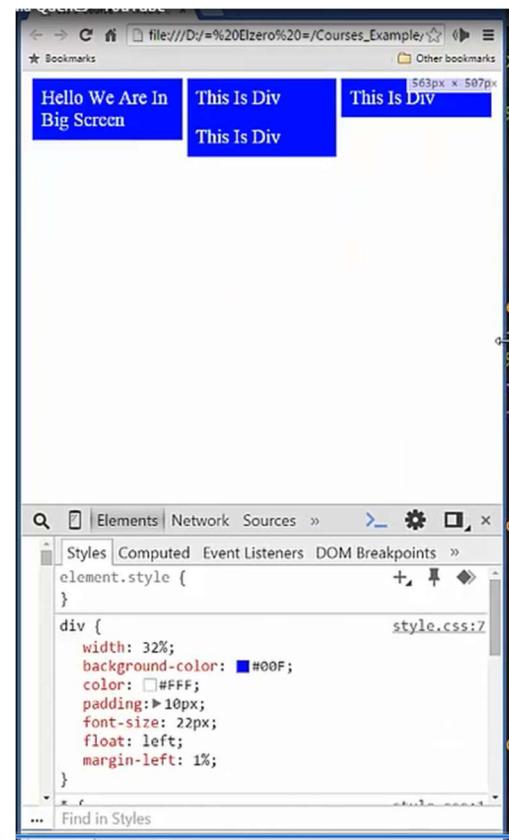
```
@media (max-width: 500px) {  
    div {  
        float: none;  
        background-color: #F00;  
        width: 100%;  
        margin-left: 0;  
        margin-bottom: 20px  
    }  
  
    .big {  
        display: none  
    }  
}  
  
@media (min-width: 501px) {  
  
    .small {  
        display: none  
    }  
}
```

```
<body>  
    <div class="big">Hello We Are In Big Screen</div>  
    <div>This Is Div</div>  
    <div>This Is Div</div>  
    <div>This Is Div</div>  
    <div class="small">Hello We Are In Small Screen</div>  
</body>
```

فلاجات :

أنه يتم التحكم في عناصر ال div بإخفاء

بعض العناصر وإظهار الأخرى



Elements Network

Styles Computed Event Listeners

```
element.style {  
}  
@media (max-width: 500px) style.css:18  
.big { style.css:28  
  display: none;  
}  
@media (max-width: 500px) style.css:18  
div { style.css:20  
  float: none;  
  background-color: #F00;  
  width: 100%;  
}  
...
```

Find in Styles

Elements Network Sources

Styles Computed Event Listeners DOM Breakpoints

```
element.style {  
}  
div { style.css:7  
  width: 32%;  
  background-color: #00F;  
  color: #FFF;  
  padding: 10px;  
  font-size: 22px;  
  float: left;  
  margin-left: 1%;  
}  
...
```

Find in Styles

ويمكن عمل استايل لكل حجم شاشة أو تحديد من حجم كذا لحجم كذا

```
@media (min-width: 1000px) and (max-width: 1600px) {  
  
  div {  
    float: none;  
    background-color: #F00;  
    width: 100%;  
    margin-left: 0;  
    margin-bottom: 20px  
  }  
}
```

لمعرفة مقاسات واجهات الشاشات للأجهزة على مستوى العالم

Screensiz.es/phone

PHONE	OPERATING SYSTEM	PHYSICAL SIZE "	PHYSICAL SIZE CM	WIDTH PX	HEIGHT PX	DEVICE WIDTH PX	PX PER INCH	PX DENSITY	ASPECT RATIO	POPULARITY %
iPhone 4 (4, 4S)	iOS	3.5	8.9	640	960	320	326	200% XHDPI	2 : 3	8.8%
Galaxy SIII	Android	4.8	12.0	720	1280	360	306	200% XHDPI	9 : 16	8.8%
iPhone 5 (5c, 5s)	iOS	4.0	10.0	640	1136	320	326	200% XHDPI	40 : 71	7.9%
Galaxy SII	Android	4.3	10.8	480	800	320	219	150% HDPI	3 : 5	6.4%
Google Nexus 4 by LG	Android	4.7	12.0	768	1280	384	320	200% XHDPI	3 : 5	6.1%
Galaxy Nexus	Android	4.6	11.8	720	1280	360	316	200% XHDPI	9 : 16	6.1%
Galaxy SIV	Android	5.0	12.7	1080	1920	360	441	300% XXHDPI	9 : 16	4.6%
Galaxy S Plus	Android	4.0	10.0	480	800	320	233	150% HDPI	3 : 5	3.7%
Galaxy Note II	Android	5.5	14.1	720	1280	360	267	200% XHDPI	9 : 16	3.7%
Galaxy S	Android	4.0	10.0	480	800	320	233	150% HDPI	3 : 5	3.6%
Nokia Lumia 520	Windows	4.0	10.0	480	800	320	233	150% HDPI	3 : 5	3.0%
Nokia Lumia 925	Windows	4.5	11.0	768	1280	384	332	200% XHDPI	3 : 5	2.9%
Nokia Lumia 1020	Windows	4.5	11.0	768	1280	384	332	200% XHDPI	3 : 5	2.9%

إذا كتبنا في جوجل media queries for standard devices

سوف تظهر لنا موقع توضح لنا المقاييس القياسية لكل الأجهزة

The screenshot shows a blog post titled "Media Queries for Standard Devices" from CSS-Tricks. The post discusses responsive design and creating experiences for different devices. It includes a quote from Josh Brewer and a sidebar for Treehouse advertising.

Last updated on: MARCH 24, 2015

If you think responsive's simple, I feel bad for you son. We got 99 viewports, but the iPhone's just one.
—Josh Brewer, March 10, 2010

A major component of responsive design is creating the right experience for the right device. With a gazillion different devices on the market, this can be a tall task. We've rounded up media queries that can be used to target designs for many standard and popular devices that is certainly worth a read.

If you're looking for a comprehensive list of media queries, this repository is a good resource.

If you're reaction to this is: *you should never base your breakpoints on devices!!* You have a good point. Justin Avery has a nice post on the possible pitfalls of using device-specific breakpoints. Choosing breakpoints based on your design and not specific devices is a smart way to go. But sometimes you just need a little help getting one particular situation under control.

► Phones and Handhelds

► Tablets

treehouse

Treehouse is where you go to learn HTML, CSS, and how to build iOS apps. It's a complete education in modern web and app technology, designed to get you ready for a hot new job or to kickstart your own business.

انتهى الكورس.