

CSS3

Elzero Web School ملخص كورس ال

“CSS & CSS3”



A Lot of Thanks To Osama Mohamed

Created By: Ahmed Adel

Contents

4	المقدمة
5	طريقة الكتابة Syntax
6	Name Conventions Rules, Comment
7	الخلفية Background
9	Padding, Margin, Border, Outline
11	Display "block, inline-block, inline"
12	Element Visibility
13	Grouping Multiple Selectors, Nesting
14	Dimensions 'Width, Height'
15	Overflow
16	النص Text
20	التوريث Inheritance
21	الخط Font
23	وحدات القياس CSS Units
24	مؤشر الماوس Mouse Cursor
25	Float and Clear
26	CSS Calculation
27	الشفافية Opacity
28	Position, Z-index
30	"Lists, Table" Styling
31	Pseudo "Classes, Element"
32	Vendor Prefixes
33	Border Radius

34	Box Shadow
35	The Box Model, Box Sizing
36	Transition
37	Important Declaration
38	Vertical Margin Collapse
39	CSS Variables
40	Flex Box 'Parent'
43	Flex Box 'Child'
46	Filters
47	Gradients تدرجات الالوان
48	Pointer Events & Caret Color
49	Grid "Parent"
52	Grid 'Child'
54	Transform "2D"
56	Transform Origin 2D
57	Transform 3D
60	Animation
63	CSS Selectors
66	Media Queries & Responsive Designs
67	CSS Global Values
68	Important links

المقدمة

معلومات عن لغة CSS:-

CSS هي اختصار لـ Cascading Style Sheet, وهي اللغة المسؤولة عن تنسيقات صفحة الويب وعن الطريقة التي سيعرض بها كود الـ HTML في الصفحة.

نصائح أثناء تعلم CSS:-

- 1- أنشاء مجلد خاص لكل عناصر الموقع مثل الملفات أو الصور.
- 2- Google it ابحث قبل ان تسأل.
- 3- مصادر مهمة للمعلومات [W3Schools](#) و [MDN](#).

كيف تعمل CSS مع HTML:-

يتم ربط صفحة HTML بملف CSS بوضع واختيار مسار ملف CSS في href.

مثال:

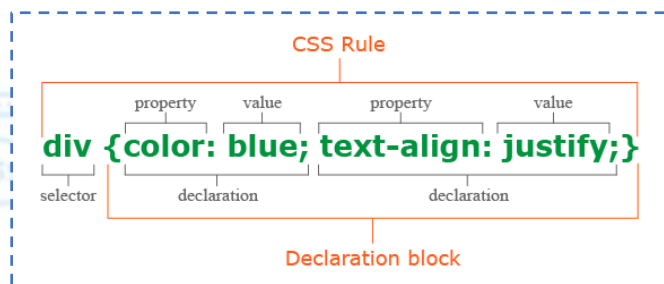
```
<link rel="stylesheet" href="CSS/main.css"><!-- External Style -->  
<style>p {color:blue;}</style><!-- Internal Style -->
```

إضافة التنسيق داخل عنصر HTML.

```
<p style="color: red;">Lorem, ipsum.</p><!-- Inline Style -->
```

طريقة الكتابة Syntax

Selector: هو الجزء الخاص باستدعاء عنصر HTML.



Property: خاصية يتم اضافتها للعنصر،

مثل "اللون, الحجم".

Value: القيمة التي يتم تحديدها

للخاصية مثال "اللون: أزرق".

{ } : هي الاقواس التي يكتب بداخلها الخصائص والقيم, وتسمى **Curly Braces**.

```
<P>Lorem ipsum dolor sit.</P>
```

Lorem ipsum dolor sit.

استدعاء باسم العنصر.

```
p {color: red;font-size: 14px}/* Select by element name */
```

Lorem, ipsum dolor.

استدعاء باستخدام Class,

```
<p class="test">Lorem, ipsum dolor.</p>
```

ملحوظة! لا يجب البدء بالأرقام عند تسمية الClass.

```
.test {color: gold} /* Select by Class */
```

```
<p id="test">Lorem, ipsum.</p>
```

Lorem, ipsum.

استدعاء باستخدام ID.

```
#test {color: blue} /*Select by ID */
```

Name Conventions Rules, Comment

شروط تسمية الـ **Class** او أي **identifier** اخر كالتالي:

1- تجنب بدأ التسمية بأرقام او رموز محجوزة مثال "class: 1test" ، "class: \$test".

2- يمكن البدء ببعض الرموز مثل "class: -test" ، "class: _test".

التعليق: تختلف طريقة كتابة التعليق عن الـ HTML لكن لا يختلف في الوظيفة.

```
div {  
  background-color: red; /* This is comment */  
}
```

يمكن استخدام التعليق في تهميش الكود:

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

مثال اخر:

```
div {  
  /* background-color: red;*/  
  background-color: blue;  
}
```

الخلفية Background

لون الخلفية Background Color :-

تحديد لون خلفية العنصر بطرق مختلفة، ويمكن تحديد مدى شفافية اللون، W3School.

```
div {
  background-color: red; /* Color Name */
  background-color: rgba(18, 50, 192, 0.5); /* Red Green Blue Alpha Channel*/
  background-color: #1231bb; /* Hex Code Color */
}
```

صورة الخلفية Background image :-

يمكن اختيار صورته وجعلها خلفية للعنصر، وتكون بنفس حجم العنصر، وإذا كان العنصر أكبر من اللازم تتكرر الصورة أفقياً ورأسياً.

```
div {
  background-image: url(/img/233231.jpg);
  background: no-repeat; /* Repeat is default value, repeat-x, repeat-y */
}
```

- :Background Attachment

تساعد علي التحكم في الخلفية وجعلها ثابتة او تتحرك مع المحتوى عند التحرك في الموقع.

```
div {
  background-attachment: fixed; /* default value: scroll */
}
```

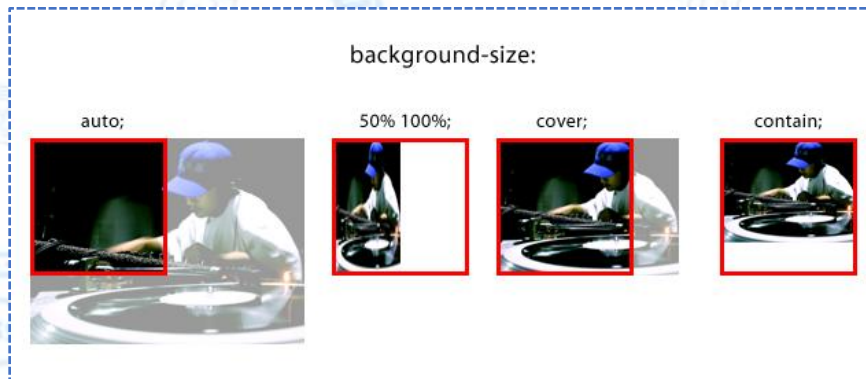
:Background Position التحكم في مكان عرض الصورة.

```
div {
  background-position: left center;
  background-position: 10px 30px;
  background-position: 10% 40%;
}
```

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

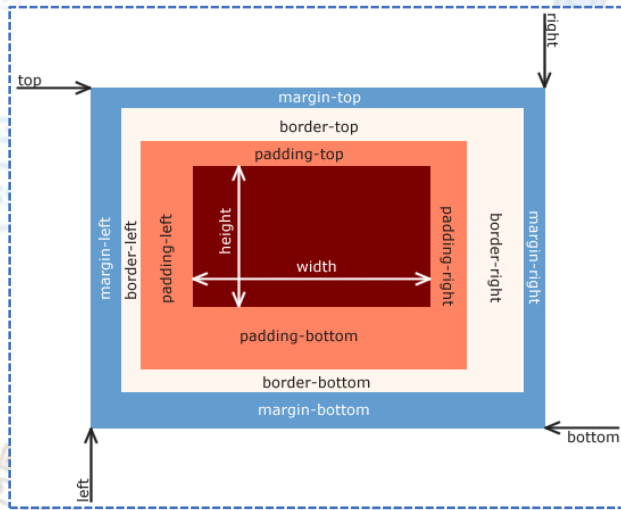
- :Background Size

يسمح بالتحكم في حجم الصورة وطريقة عرضها.



```
div {
  background-size: cover; /* default value: auto */
  background-size: contain;
}
```


Padding, Margin, Border, Outline



Padding: الهوامش الداخلية تعتبر جزء من عرض وطول العنصر، تحديد لون خلفية للعنصر سيتم تحديده للهوامش أيضا.

Margin: التحكم في الهوامش الخارجية للعنصر، يمكن استخدامه في توسيط العنصر في مكان ما والتحكم في المسافة بينه وبين العناصر الأخرى في الصفحة.

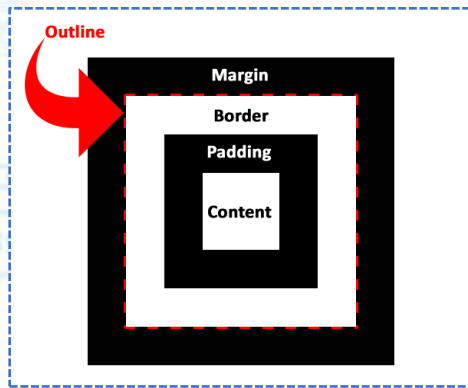
```
div {
  padding: 10px; /* top, right, bottom, left 'px, %'*/
  margin: auto; /* top, right, bottom, left 'px, %' can use negative num*/
}
```

Border: يقوم بالتحكم في شكل وحجم ولون الحواف الخارجية للعنصر.

```
div {
  border-width: 2px; /* top, right, bottom, left */
  border-color: red green; /* top, right, bottom, left */
  border-style: solid dashed; /* top, right, bottom, left */
  border: 2px solid red; /* Shorthand code */
}
```

ملحوظة! الـ "padding, border" مساحتهم تضاف لمساحة العنصر، ويمكن تجنب ذلك باستخدام

```
*{box-sizing: border-box}
```

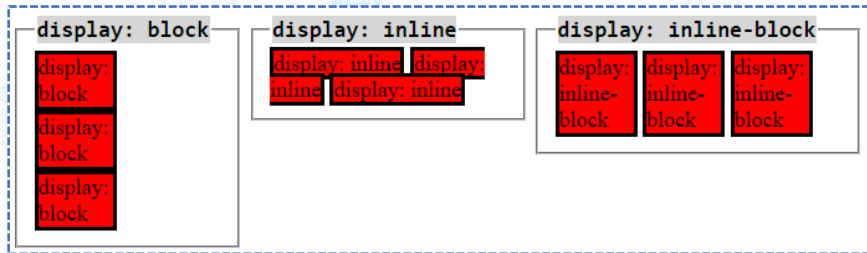


Outline: يقوم بإضافة خط خارجي للعنصر لكن لا يحسب من ضمن مساحة العنصر.

```
div {
  outline-width: 10px; /* one value for all declaration */
  outline-color: blue; /* one value for all declaration */
  outline-style: dashed; /* one value for all declaration */

  outline: 10px dashed blue; /* Shorthand code*/
}
```

Display “block, inline-block, inline”



Display: التحكم في

طريقة عرض العنصر.

```

/*

Block
- Take Full width if there no width set
- Add Line Break
- Respect Padding, Margin, Width, Height

Inline
- Dose not Respect Width, Height
- Respect Padding and Margin [Just left + right]
- Dose not Add Line Break
- Allow Element Before and After it

Inline-Block
- Allow Element Before and After it
- Respect Padding, Margin, Width, Height

*/

```

مثال:

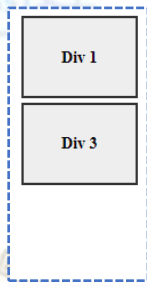
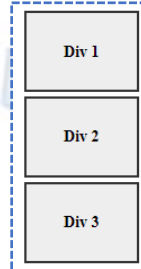
```

div {
background-color: #eee;
border: 3px solid #333;
width: 200px;
padding: 15px;
margin: 5px;
display: block
}

```

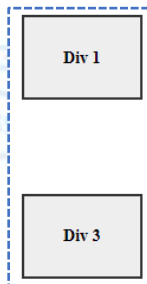
Element Visibility

طرق إخفاء العنصر: -



باستخدام **'Display: none'**: سيتم إخفاء العنصر نهائيا ولن يكون له مكان محجوز داخل الصفحة.

```
.div2 {
  display: none;
}
```



باستخدام **'visibility: hidden'**: سيتم إخفاء العنصر لكن سيكون له مكان محجوز داخل الصفحة.

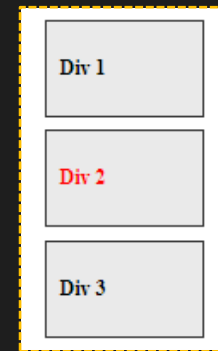
```
.div2 {
  visibility: hidden;
}
```

Grouping Multiple Selectors, Nesting

Selector Grouping: طريقة تستخدم لربط العناصر المتشابهة في الخواص ببعضها وكتابة الخواص مره واحده فقط بدلا من كتابة الخواص لكل عنصر علي حدة.

```
.div1, .div2, .div3 {
  width: 100px;
  background-color: #eaeaea;
  border: 2px solid #333;
  padding: 10px;
  margin: 10px;
}

.div2 {
  color: red;
}
```



Paragraph Inside Div
 Paragraph Inside Div with Class
 Paragraph Outside Div

Nesting: استدعاء عنصر محدد من داخل عنصر اخر او من بين مجموعة عناصر متشابهة.

```
<div>
  <p>Paragraph Inside Div</p>
  <p class="test">Paragraph Inside Div with Class</p>
</div>
<p class="test">Paragraph Outside Div</p>

div .test {
  color: red;
}
```

Dimensions 'Width, Height'

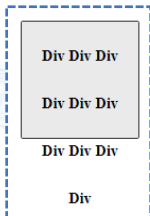
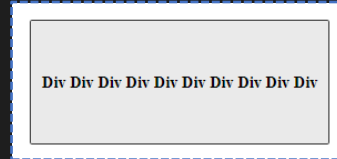
Width - Height: تحديد الطول والعرض للعنصر، [تفاصيل اكثر](#).

```
div {
  width: 70px;
  height: 70px;
}
```



Min 'Width Height': تحديد الحد الأدنى لمساحة العنصر، وتتمدد مساحة العنصر مع مساحة المحتوى.

```
div {
  min-width: 150px;
  min-height: 100px;
}
```



Max 'Width Height': تحديد الحد الأقصى لمساحة العنصر، لا تتمدد مساحة العنصر بازدياد مساحة المحتوى.

```
div {
  max-width: 100px;
  max-height: 100px;
}
```

ملاحظة! اذا كان العنصر block ستتغير مساحة العنصر وتتمدد مع مساحة المحتوى.

```
div {width: fit-content}
```

Overflow

>LoremLoremLoremLoremLorem
 ipsum dolor sit amet
 consectetur
 adipiscing elit.
 Mollitia fuga nesciunt
 quis quod tenetur
 debitis voluptate
 incidunt natus quae
 quaerat, soluta
 ratione earum
 explicabo corrupti
 cupiditate ipsa illum
 asperiores ea!

Overflow: يتحكم في الجزء الذي تجاوز العنصر،
 يمكن إخفاء هذا الجزء او إظهاره بطرق مختلفة.

>LoremLoremLoremLo
 ipsum dolor sit amet
 consectetur
 adipiscing elit.
 Mollitia fuga nesciunt
 quis quod tenetur
 debitis voluptate
 incidunt natus quae
 quaerat, soluta

Hidden: يقوم ب إخفاء المحتوى الذي تجاوز العنصر.

```
div { overflow: hidden; /* Default: visible */}
```

>LoremLoremLorem
 ipsum dolor sit
 amet consectetur
 adipiscing elit.
 Mollitia fuga
 nesciunt quis quod
 tenetur debitis
 voluptate incidunt

Scroll: يقوم بإضافة Scroll-bar حتي اذا كان المحتوى لم يتجاوز العنصر.

```
div { overflow: scroll }
```

ipsum dolor sit
 amet consectetur
 adipiscing elit.
 Mollitia fuga
 nesciunt quis quod
 tenetur debitis
 voluptate incidunt
 natus quae
 quaerat, soluta

Auto: هو الاختيار الأفضل لأنه يضيف Scroll-bar فقط في حالة اذا تجاوز
 المحتوى العنصر.

```
div { overflow: auto}
```

ملحوظة! يمكن اختيار اتجاه محدد والتحكم به.

```
div { overflow-x: hidden; overflow-y: auto}
```

النص Text

Lorem ipsum dolor sit.

Color: يقوم بتحديد لون النصوص فقط.

```
div { color: #e03e3e; /* Color: 'Color name' or 'Hex color' or 'RGBA' */ }
```

Lorem ipsum dolor sit.

Shadow: إضافة ظل للنص والتحكم فيه.

```
/* Text-Shadow: H-shadow V-shadow Blur Color */
div { text-shadow: -6px 4px 6px rgba(62, 73, 224, 0.8) }
```

Lorem ipsum dolor sit.

Text Align: تحديد اتجاه محاذاة النص أفقياً.

```
div { text-align: center } /* left Center Right and more */
```

Lorem ipsum  dolor sit.

Vertical align: التحكم في المحاذاة العمودية. [المزيد](#).

```
div img { vertical-align: middle }
```


Direction: تحديد اتجاه النص.

.Lorem ipsum dolor sit
السلام عليكم.

```
div { direction: rtl } /* rtl => right to left or ltr => left to right */
```

Text Decoration: التحكم في شكل وزخرفة النص.

none underline overline ~~line-through~~

```
.one { text-decoration: none }           .two { text-decoration: underline }
.three { text-decoration: overline }    .four { text-decoration: line-through }
```

Text Transform: تحويل حجم حروف النص وطريقة

عرضها.

UPPERCASE lowercase Capitalize

```
.one { text-transform: uppercase }      .two { text-transform: lowercase }
.three { text-transform: capitalize }
```

Letter Spacing: المسافة بين الحروف, يمكن تزويدها

او أنقصها.

Lorem ipsum dolor sit.

```
div { letter-spacing: 5px } /* You can use negative value EX: '-2px' */
```

Text Indent: إضافة مسافة قبل النص.

Lorem ipsum dolor sit.

```
div { text-indent: 130px } /* You can use negative value */
```

Line Height: يستخدم لتحديد المسافة بين السطور.

Lorem ipsum dolor sit amet consectetur
adipiscing elit. Saepe nihil in tempore
quaerat eum, facilis similique provident
cupiditate architecto voluptate?

```
div { line-height: 1.6 } /* Default: normal , px , em , % , num EX: 1.5 */
```

Word Spacing: يستخدم للتحكم في المسافة بين الكلمات.

Lorem ipsum dolor sit.

```
div { word-spacing: 10px } /* Default: normal , px , rem , % , ch */
```

White Space: التحكم في المسافات بين

Lorem ipsum dolor sit amet consectetur adipiscing elit. Ipsum, voluptate.

الكلمات وطريقة الفصل بينهم. [المزيد](#)

```
div { white-space: nowrap }
```

Word Break: التحكم في طريقة فصل الكلمة في حالة تخطيها العنصر.

LoremLoremLoremLoremLoremLoremLoremLoremLorem
TestTestTest

LoremLoremLoremLoremLoremLoremLo
remLoremLoremLorem TestTestTest

```
div { word-break: break-all }
```

Lorem ipsum dolor sit am...

>Lorem ipsum dolor sit amet consectetur.

Text Overflow: يتحكم في النص الذي

تجاوز المحتوي وطريقة عرضه.

```
div {
  overflow: hidden;
  text-overflow: ellipsis;
}
```



التوريث Inheritance

التوريث **Inheritance**: تجعل العنصر يأخذ

خواص العنصر الاب الذي بداخلة.

[المزيد](#)

Heading

Lorem ipsum dolor sit amet.

```
<div>
  <h1>Heading</h1>
  <p>Lorem ipsum dolor sit amet.</p>
</div>

div { border: solid 3px rgb(202, 52, 52) }

h1 { border: inherit }
```

الخط Font

Lorem ipsum dolor sit amet. 'Oswald'

Lorem ipsum dolor sit amet. 'Arial'

Lorem ipsum dolor sit amet. 'Serif'

Font Family: يمكنك من اختيار نوع الخط, وكل نوع له شكل مختلف.

```
div.one { font-family: 'Oswald', sans-serif } /* Oswald Google font */
div.two { font-family: arial }
div.three { font-family: serif }
```

Google Font: يمكن استخدام أنواع خطوط كثيرة من هذا الموقع عن طريق اضافتها وربطها بملف الHTML داخل وسم <head>, او يمكن تنزيل الخطوط و اضافتها في ملف الموقع.

[Google Font](#)

```
<link rel="preconnect" href="https://fonts.googleapis.com" >
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link rel="stylesheet" href="https://fonts.googleapis.com/css2?family=
Oswald:wght@300;400;700&display=swap" >
```

Font Size: يتم تحديد حجم الخط بالوحدات مثل ال "px", القيمة الافتراضية لحجم الخط العادي هي 16px.

```
div { font-size: 20px }
```

Lorem ipsum dolor sit.

Font Style: تحديد شكل للخط مثل النص المائل.

```
div { font-style: italic } /* default: normal */
```

LOREM IPSUM DOLOR SIT.

Font Variant: تحويل شكل الخط.

```
div.two { font-variant: small-caps } /* default: normal */
```

Lorem ipsum dolor sit.

Font Weight: تحديد وزن الخط.

```
div.three { font-weight: bold } /* default: normal */
```

وحدات القياس CSS Units

تقسم الوحدات الي نوع ثابت مثل "Absolute length units" وهذا النوع يضم الpx , اما النوع الثاني هو "Relative length units" هي وحدات طولية نسبية يحدد طولها بالنسبة لطول عنصر اخر ويتغير طولها مع تغير طول العنصر الاخر.


Unit	الوصف
Px	Pixels: قياس ثابت لا يتغير.
%	نسبة الي العنصر الاب.
em	2 = 2em ضعف الحجم الحالي .
rem	2 = 2rem ضعف حجم خط بالنسبه للحجم في الRoot.
vw	"Viewport Width" 1vw = 1% من المساحة الظاهره من عرض الشاشة.
vh	"Viewport Hight" 1vh = 1% من المساحة الظاهره من طول الشاشة.

[المزيد.](#)



مؤشر الماوس Mouse Cursor

Cursor: تغيير شكل المؤشر.

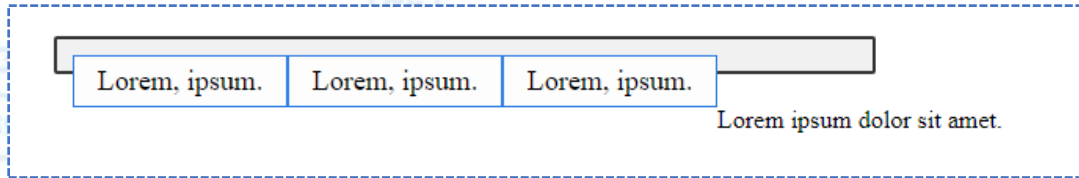
Lorem ipsum dolor sit 

```
div { cursor: not-allowed }
```

I auto	↕ move	👉🚫 no-drop	↔ col-resize
👁 all-scroll	👉 pointer	🚫 not-allowed	↕ row-resize
+ crosshair	🕒 progress	↔ e-resize	↗ ne-resize
👉 default	I text	↓ n-resize	↖ nw-resize
👉? help	↔ vertical-text	↑ s-resize	↗ se-resize
I inherit	🕒 wait	↔ w-resize	↖ sw-resize

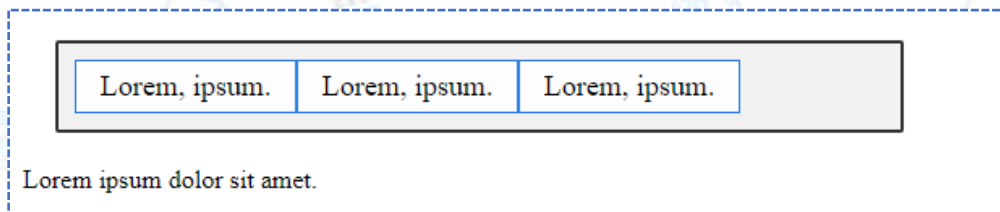
Float and Clear

Float: التحكم في مكان العناصر وتوزيعها علي اليسار او اليمين, لكن تحدث مشاكل واخطاء عند استخدام هذه الخاصية لكن يمكن حلها.



```
.box { float: left }
```

Clear: يستخدم لتصحيح أخطاء الFloat, يجب وضعة بعد نهاية عناصر الFloat.



```
.clearfix { clear: both }
```

```
<div class="parent">
  <div class="box">Lorem, ipsum.</div>
  <div class="box">Lorem, ipsum.</div>
  <div class="box">Lorem, ipsum.</div>
  <div class="clearfix"></div> <!-- Fixing float problems -->
</div>
<p>Lorem ipsum dolor sit amet.</p>
```



CSS Calculation

Calc: تحديد حجم العناصر بطريقة حسابية سهلة.

Lorem, ipsum.

Lorem, ipsum.

Lorem, ipsum.

Lorem, ipsum.

```
.box {  
  width: calc( (100% - 50px) / 4 );  
  margin-left: 10px;  
}
```



الشفافية Opacity

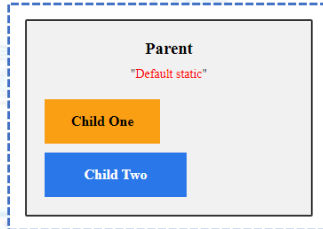
Opacity: تستخدم لتحديد شفافية العنصر.

Lorem, ipsum.

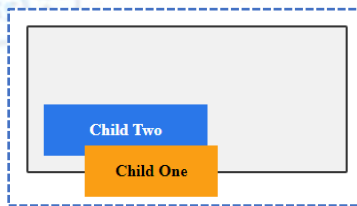
Lorem, ipsum.opacity 0.5

```
div.two { opacity: 0.5 } /* Default => 1 */
```

Position, Z-index

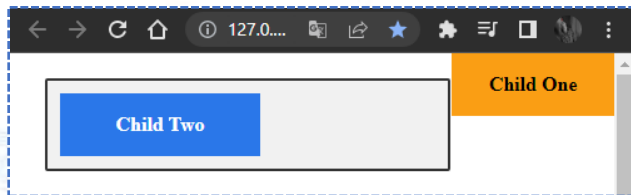


Position: التحكم في تحديد مكان ووضع العنصر من جميع الجهات "left, top, right, bottom" ويمكن التحديد باستخدام Px او %.



Relative: يتم تحديد مكان العنصر بالنسبة لمكانة الحالي وحتى اذا تحرك العنصر يكون حاجز المكان الذي تم إنشاؤه فيه.

```
div.one { position: relative; left: 40px; top: 100px }
```



Absolute: العنصر يكون مطلق ويتحرك بالنسبة لصفحة الويب بالكامل وليس بالنسبة للمكان الذي به, لا يعتبر مكانة محجوزا.

```
div.one { position: absolute; top: 0; right: 0 }
```



ملحوظة! إضافة Relative الي العنصر الاب سيمنع خروج العناصر التي بداخله عند استخدام absolute.

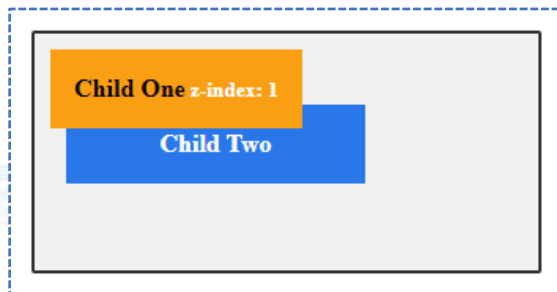
```
.parent {position: relative}
```

Fixed: يكون العنصر ثابت في المكان الذي سيتم تحديده له, يستخدم في انشاء زر Go To Top.

```
div.one { position: fixed; top: 0; right: 0 }
```

Sticky: يكون العنصر ثابت بمكانة لكن عند الوصل للعنصر بمؤشر الشاشة يبدأ بالتفاعل ويتغير مكانة علي حسب المكان الذي قمت بتحديدده يستخدم كثيرا في الNavbar.

```
div.one { position: sticky; top: 0 }
```

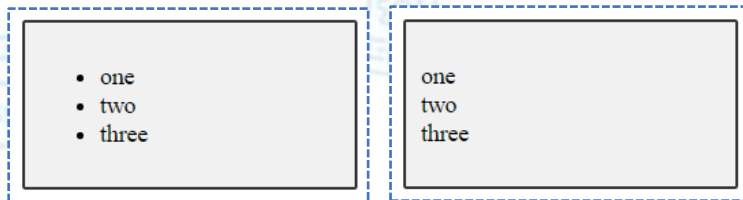


Z Index: تحديد ترتيب الطبقات, عناصر HTML دائما يكون اخر عنصر مكتوب هو ذا الأعلى طبقة, لا يعمل z-index الا مع استخدام الposition.

```
div.one { position: relative; z-index: 1}
```

“Lists, Table” Styling

List Style: تغيير شكل الرموز الخاصة بالقائمة او ازلتها.



```
ul { list-style: none; padding-left: 0 }
```

Name	Score	Stage
Ahmed	9/10	1
Mohamed	7.5/10	2
Sara	8/10	3
Total Score:	24.5	

Table: من الاسهل إضافة تنسيقات الجدول باستخدام CSS.

ملحوظة! border-spacing: تستخدم لتحديد المسافة بين خلايا الجدول.

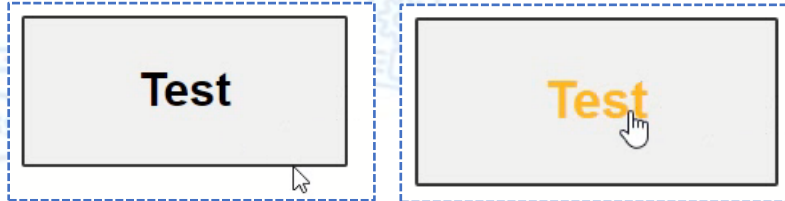
```
table { border-spacing: 0; border: 1px solid #ccc }
th, td { padding: 10px 25px; text-align: left}
td { border-top: 1px solid #ccc }
tbody tr:nth-child(odd) { background-color: #f2f2f2 }
tfoot { font-weight: 700 }
```



Pseudo “Classes, Element”

Pseudo Classes: أدوات تستخدم لإضافة والتحكم في وظائف العناصر.

مثل “:hover, :focus, :active, :disabled”



[.All Pseudo Classes](#)






```
h1:hover { color: #fdb925; cursor: pointer }
```

Pseudo Element: عناصر وهمية لكل منها وظيفة محددة ومختلفة.

مثال “::after, ::before, ::selection”

[.All Pseudo Element](#)

Vendor Prefixes

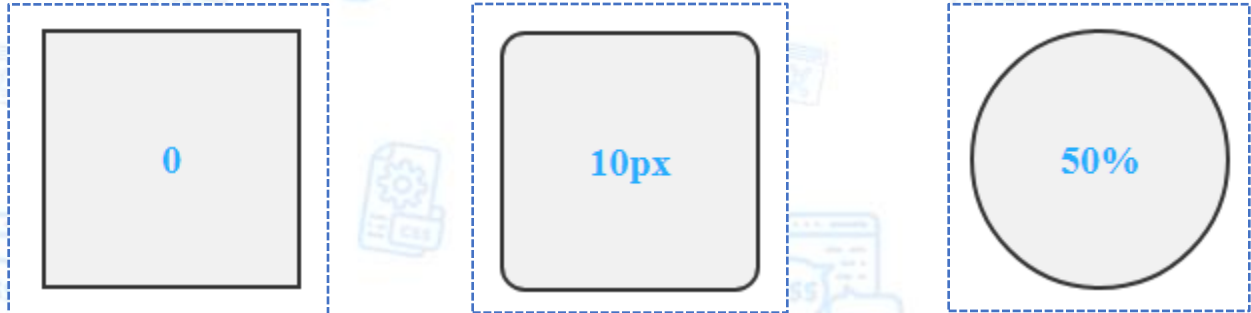
Browser	Prefix
 Firefox	-moz-
  Chrome, Safari	-webkit-
 Internet Explorer	-ms- , mso-
 Opera	-o-

Prefixes: أدوات تستخدم لضمان عمل الكود في متصفحات القديمة.

```
.box {
  transition: all .8s;
  -webkit-transition: all .8s;
  -moz-transition: all .8s;
  -ms-transition: all .8s;
  -o-transition: all .8s;
}
```

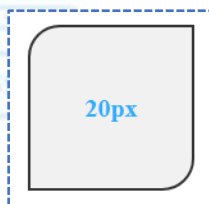
موقع "[Can I Use](#)" للتأكد من دعم الخصائص في إصدارات المتصفحات.

Border Radius



Border Radius: يستخدم لتحديد قطر الحواف الخارجية.

```
.box {
  width: 100px;
  height: 100px;
  background-color: #f1f1f1;
  border: 2px solid #333;
  border-radius: 50%; /* Default => 0 */
  -webkit-border-radius: 50%;
  -moz-border-radius: 50%;
  -ms-border-radius: 50%;
  -o-border-radius: 50%;
}
```

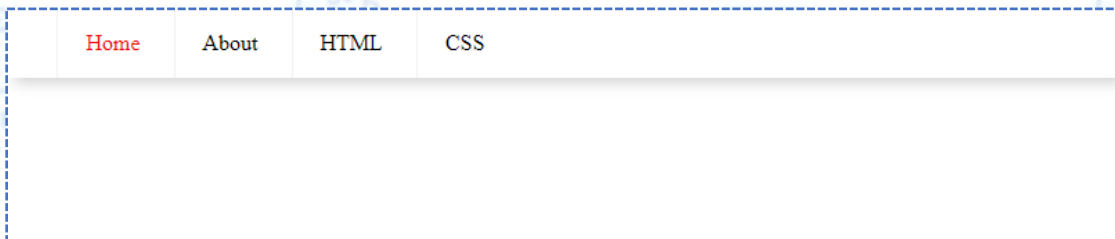


ويمكن تحديد القطر من جوانب معينة.

```
.box {
  border-top-left-radius: 20px;
  border-bottom-right-radius: 20px;
}
```

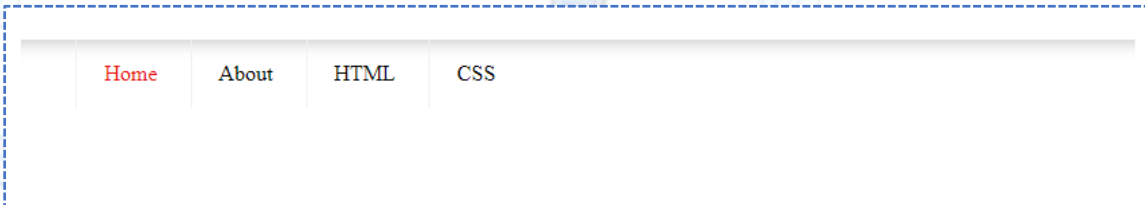
Box Shadow

Box Shadow: يقوم بأضافة ظل خارجي للعنصر ويمكن التحكم في اتجاه الظل ومدى وضوحه.



```
/* H-Shadow, V-Shadow, Blur, Color, inset 'if you want an inner shadow' */
.box { box-shadow: 0px 13px 11px -10px #00000031 }
```

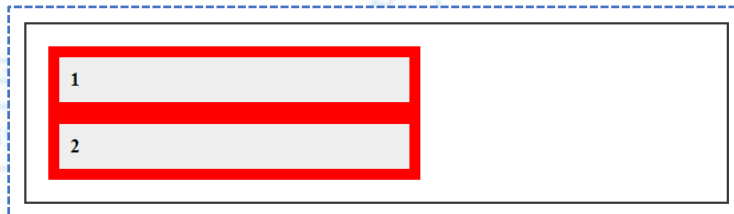
لأضافة ظل داخلي للعنصر يجب استخدام `inset`.



```
.box { box-shadow: 0px 13px 11px -10px inset #00000031 }
```

The Box Model, Box Sizing

Box Model: هو الطريقة التي يتعامل بها المتصفح مع طول وعرض العنصر والهوامش والحواف الخاصة به.



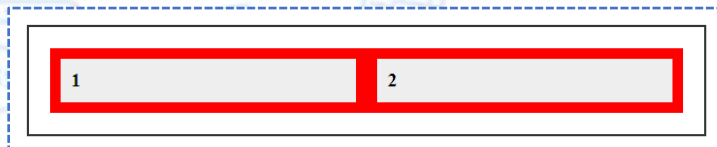
Content Box: يقوم المتصفح بي إضافة مساحة الهوامش والحواف الي مساحة العنصر الأساسية.

```
.parent { width: 600px; padding: 20px }

.one {
  width: 300px;
  padding: 10px; /* padding left + Right = 20px */
  border: solid 10px red; /* border left + Right = 20px */
  box-sizing: content-box; /* Default */
}
```

Border Box: لا تحسب مساحة الهوامش والحواف من مساحة العنصر, والمساحة النهائية

للعنصر تكون مساوية لنفس المساحة التي تم تحديدها له.



```
.one {
  width: 300px;
  padding: 10px; /* padding left + Right = 20px */
  border: solid 10px red; /* border left + Right = 20px */
  box-sizing: border-box;
}
```

Transition

Transition: التحكم في مدة انتقال العنصر من حالة الى حالة.

```
.box {  
  width: 300px;  
  height: 200px;  
  background-color: #eee;  
  transition-duration: .5s ; /* or by ms = '500ms' */  
  transition-delay: 1s;  
  transition-property: width, background-color; /* or all */  
  transition-timing-function: linear; /* Default ease */  
}
```

الاختصار

```
.box {  
  /* transition: Property, Duration, Delay, Timing Function */  
  transition: width .5s 1s linear;  
}
```

مثال اخر

```
.box { transition: all .5s 1s linear; }
```

Important Declaration

Important: تصريح يتم اعطائه لخاصية معينة, ويعبر عن أهميتها وسيقوم المتصفح بتطبيقها.

مثال: تم تحديد اللون من داخل HTML.

```
<div class="box" style="background-color: red;">  
  <h2>Test</h2>  
</div>
```

Test

بدون استخدام **!important**.

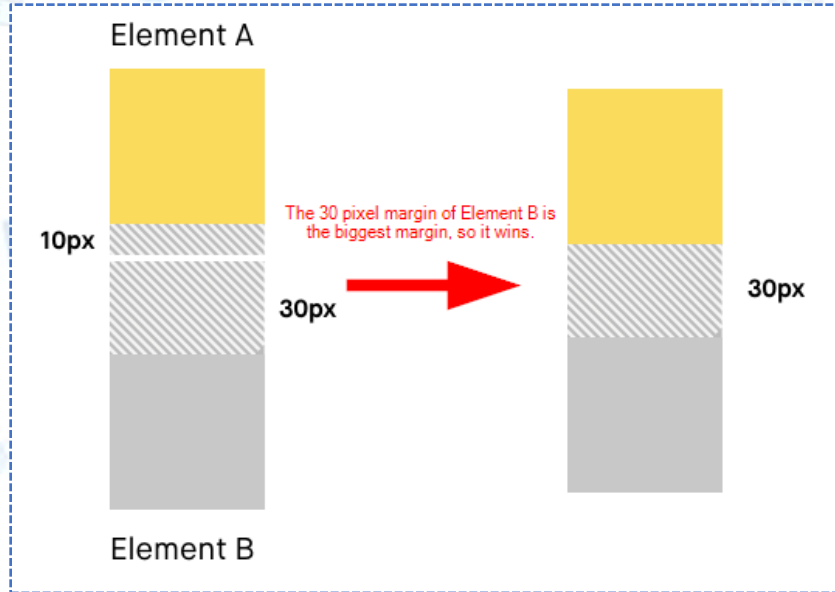
```
.box { background-color: green }
```

Test

بعد استخدام **!important**.

```
.box { background-color: green !important }
```

Vertical Margin Collapse



```

/*
  Margin Collapse

  [1] Only Vertical Margin Collapse
  [2] Bigger Margin Wins
  [3] Margin Collapsing With Elements Without Anything Between
  [4] Nesting Does Not Prevent Collapse
*/

```

CSS Variables

Variable: يتم تحديد قيمة لمتغير واستخدام هذا المتغير في الكثير من العناصر, بتغير قيمة المتغير ستتغير كل التنسيقات التي تم استخدام فيها المتغير.

[تفاصيل اكثر عن المتغيرات.](#)

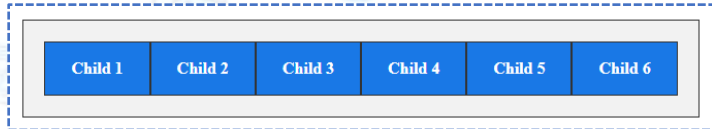
Variable

```
:root { --mainColor: red; } /* Global Variable */  
.box { background-color: var(--mainColor, #333); }
```

Variable

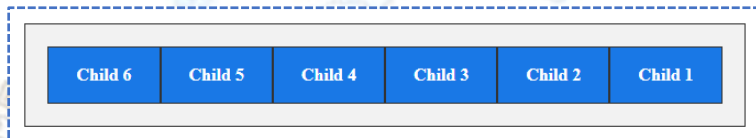
```
.box { --mainColor: blue; } /* Local Variable */
```

Flex Box 'Parent'



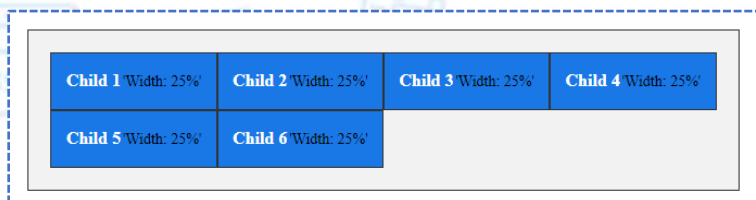
Display Flex: يتم توزيع مساحة عرض العنصر الأب بالتساوي علي عدد العناصر التي بداخله.

```
.parent { display: flex; /* display: flex => To Start Flexible Box */ }
```



Flex Direction: تحديد اتجاه العناصر ظهور العناصر, أعمدة او صفوف, او صفوف وأعمدة معكوسة.

```
.parent {
  display: flex;
  flex-direction: row-reverse; /*row, row-reverse, column, column-revers*/
}
```

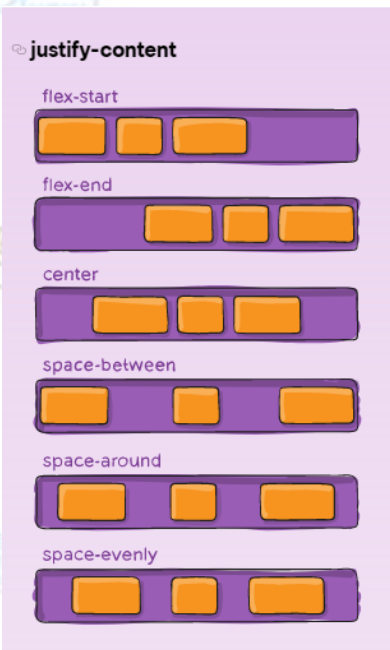


Flex Wrap: يتعامل مع العنصر الزائد اما انزلة في سطر جديد او لا.

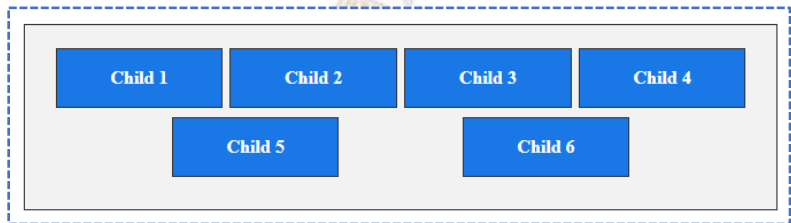
```
.parent {
  display: flex;
  flex-wrap: wrap; /* nowrap, wrap, wrap-reverse */
}
```


Flex Flow: هو اختصار لـ 'flex-direction, flex-wrap'

```
.parent {
  display: flex;
  flex-flow: row wrap; /* flex-flow: [Flex-Direction] + [Flex-Wrap] */
}
```

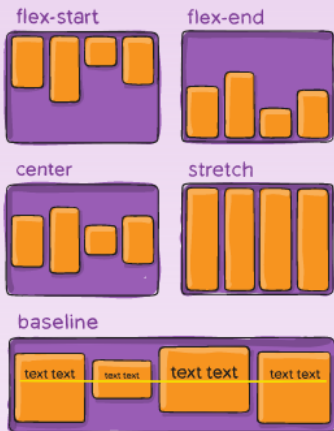


Justify Content: توزيع العناصر أفقياً داخل الـ flex box.

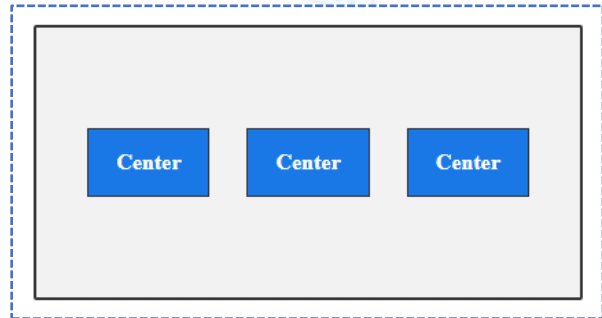


```
.parent {
  display: flex;
  flex-flow: row wrap;
  justify-content: space-evenly;
}
```

align-items

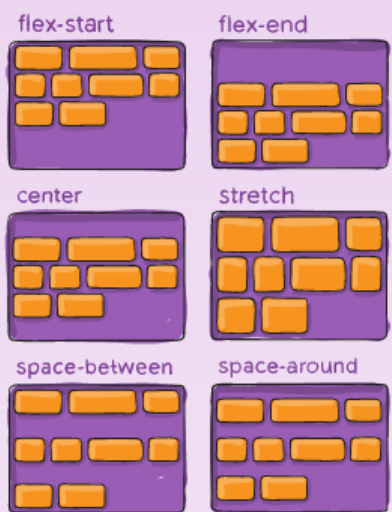


Align Items: توزيع العناصر عمودياً داخل ال flex box.

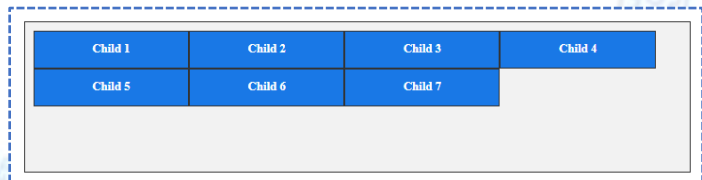


```
.parent {
  display: flex;
  flex-flow: row wrap;
  justify-content: space-evenly;
  align-items: center; /* Default stretch */
}
```

align-content

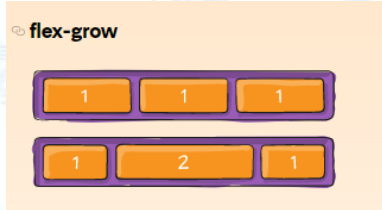


Align Content: يقوم بمحاذاة المحتوى ككل في الاتجاه العمودي.

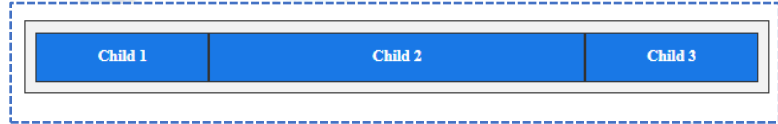


```
.parent {
  display: flex;
  flex-flow: row wrap;
  align-content: flex-start;
}
```

Flex Box 'Child'



Flex Grow: تمدد عنصر محدد عن باقي العناصر التي بجانبه.

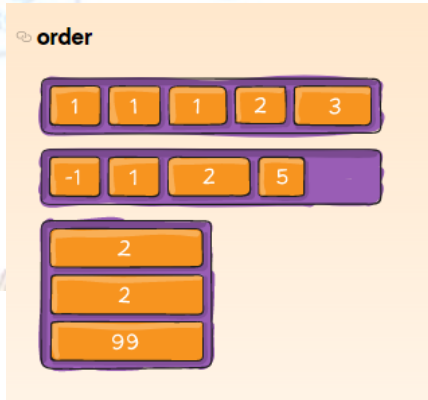


```
.child-2 { flex-grow: 1 } /* Default => 0 */
```

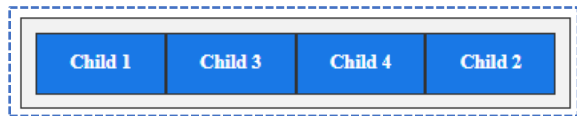


Flex Shrink: انكماش العنصر المحدد عن باقي العناصر التي بجانبه ولكن يحدث هذا الانكماش فقط في حالة مساحه العنصر الابر اقل من مساحة العناصر التي بداخله.

```
.child-2 { flex-shrink: 2 } /* Default => 1 */
```

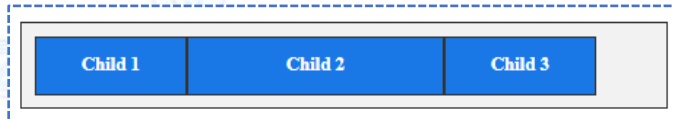


Order: تحديد ترتيب العناصر.



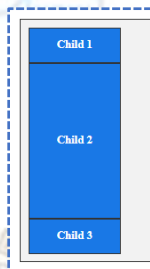
```
.child-2 { order: 1; } /* Default => 0 */
```

Flex Basis: يقوم بتحديد عرض او طول للعنصر, اذا كانت قيمة الـ 'row' Flex Direction سيتعامل مع عرض العنصر ما اذا كانت 'column' سيتعامل مع الطول.



في حالة `flex direction: row`.

```
.parent { flex-direction: row }
.child-2 { flex-basis: 220px;} /* Default => auto */
```

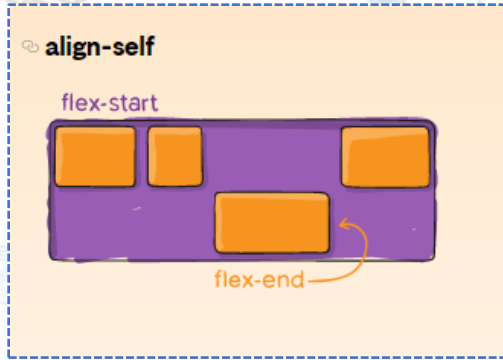


في حالة `flex direction: column`.

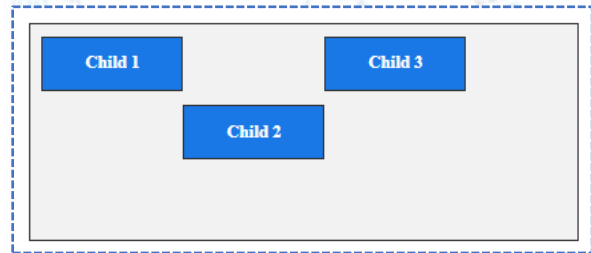
```
.parent { flex-direction: column }
.child-2 { flex-basis: 220px;} /* Default => auto */
```

Flex Shorthand: الكود المختصر.

```
/* flex: [Flex Grow] [Flex Shrink] [Flex Basis] */
.child-2 { flex: 0 1 auto ; } /* Default => 0 1 auto */
```



Align Self: محاذاة العنصر بنفسه عمودياً.



```
/* flex: flex-start | flex-end | center | baseline | stretch */
.child-2 { align-self: center } /* Default auto */
```

العاب ممتعة لتعلم الـ Flex Box:-

- [.FLEXBOX FROGGY](#)
- [.Flexbox Zombies](#)

Filters

Filters: لأضافة تأثيرات علي الوان الصور.

[المزيد عن الـFilters.](#)



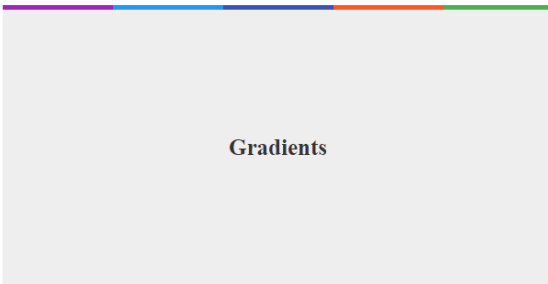
```
img {  
  filter: grayscale(100%); /* Default => none */  
  -webkit-filter: grayscale(100%);  
}  
  
img:hover {  
  filter: grayscale(0);  
  -webkit-filter: grayscale(0);  
}
```

تدرجات الالوان Gradients



Gradients Color: يقوم بأضافة الألوان وعمل تدرج بينهم.

```
div {
  background-image: linear-gradient(90deg,
    #d16ba5, #c777b9, #ba83ca, #aa8fd8,
    #9a9ae1, #8aa7ec, #79b3f4, #69bff8,
    #52cffe, #41dfff, #46eefa, #5ffbf1);
}
```



مثال اخر لأستخدام التدرج في انشاء حواف من جانب واحد.

```
div::before {
  content: '';
  position: absolute;
  top: -5px;
  height: 5px;
  width: 100%;
  background-image: linear-gradient(to right, #9c27b0 20%,
    #2196f3 20%, #2196f3 40%, #3f51b5 40%, #3f51b5 60%,
    #ff5722 60%, #ff5722 80%, #4caf50 80% );
}
```

Pointer Events & Caret Color



Caret Color: التحكم في لون مؤشر الكتابة.

```
p {
  /* Default => Auto */
  caret-color: red;
}
```



Pointer Events: التحكم في الاحداث التي يقوم بها المؤشر تغييرها

او منعها.



```
a { pointer-events: none; /* Default => Auto */ }
```


Grid “Parent”

Display Grid: يجب استخدام لتفعيل استخدام خواص الـ .grid.

Grid Template Columns: يستخدم لتحديد عدد الاعمدة.

Grid Template Rows: تحديد عدد الصفوف.

Gap: تحديد المسافات بين الصفوف والاعمدة.



```
.grid-parent {
  /* grid | inline-grid */
  display: grid;

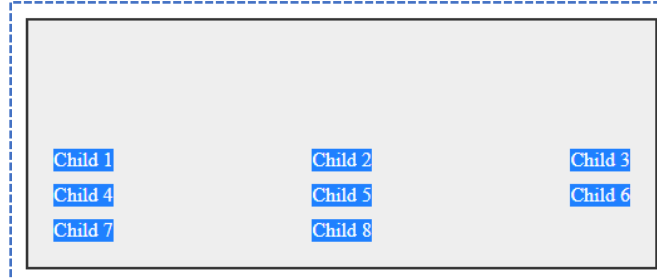
  /* Num Of Columns 'Px, %, Auto, fr, repeat()' */
  grid-template-columns: repeat(3, 1fr);

  /* Num Of Rows 'Px, %, Auto, fr, repeat()' */
  grid-template-rows: repeat(3, 1fr);

  /* [Row Gap] [Column Gap] */
  gap: 10px 20px;
}
```

Justify Content: التحكم في لمحتوي بشكل افقي.

Align Content: التحكم في المحتوى بشكل عمودي.



```
.grid-parent {  
  justify-content: space-between;  
  align-content: flex-end;  
}
```



Template Areas: يستخدم لتقسيم محتوى ال grid, يتم ربط القيمة بالعناصر باستخدام grid area واعطائها نفس الاسم الذي تم مسبقا.

مثال:-

```
.page {
  display: grid;
  grid-template-columns: repeat(10, 1fr);
  grid-template-rows: 60px auto 50px;
  height: 100vh;
  grid-template-areas: /* Layout */
  "logo logo . nav nav nav nav nav nav nav"
  "cont cont cont cont cont cont . aside aside aside"
  "foot foot foot foot foot foot foot foot foot";
}

h1 { grid-area: logo; background: #92A9BD }

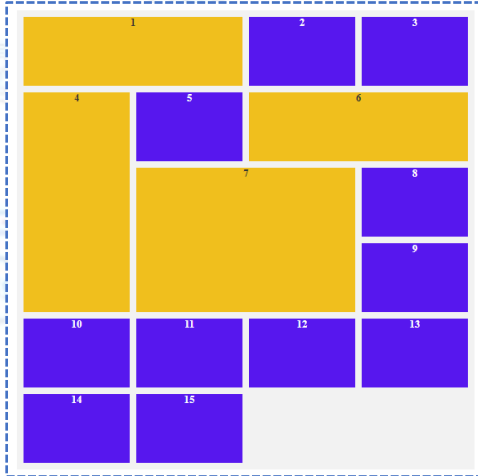
nav { grid-area: nav; background-color: #D3DEDC }

section { grid-area: cont; background-color: #c0c0c0 }

aside { grid-area: aside; background-color: #ffd700 }

footer { grid-area: foot; background-color: #17223B;color: #fff }
```

Grid 'Child'



Grid Column: تحديد بداية ونهاية العنصر من الاعمدة.

Grid Row: تحديد بداية ونهاية العنصر من الصفوف.

```

/*
- grid-column: 'start / end' | 'span [Num Of Column]'
- grid-row: 'start / end' | 'span [Num Of Row]'
*/

.child-1 { grid-column: 1 / 3 }

.child-6 { grid-column: span 2 }

.child-4 { grid-row: span 3 }

.child-7 { grid-column: span 2; grid-row: span 2 }

```

Grid Area: اختصار لـ grid row و grid column, ويمكن استخدام Grid Template Areas للعنصر الاب واستخدام الأسماء بدلا من الأرقام.

```

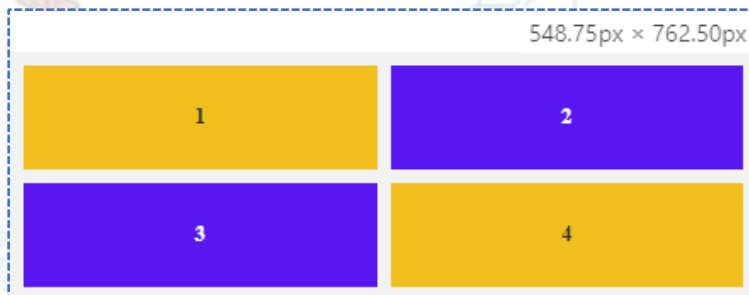
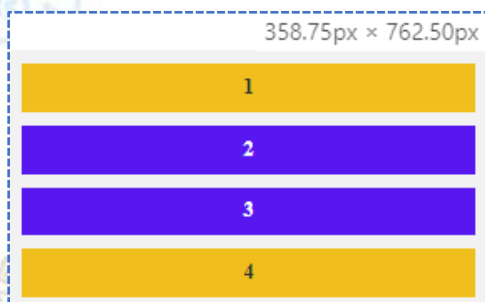
/* grid-area: [Row Start] [Column Start] [Row end] [Column end] */
.child-7 { grid-area: 3 / 2 / 5 / 4 }

```

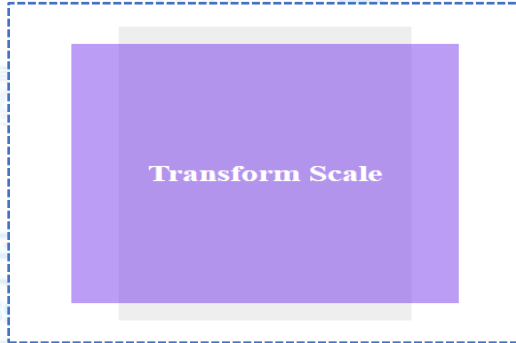
Min, Max And Auto Fill: يقوم بتحديد عدد الاعمدة المتاحة بشكل تلقائي, minmax تحديد الحد الادني والاقصي لعرض العمود.

```
.parent {
  display: grid;
  grid-template-columns: repeat(auto-fill, minmax(200px, 1fr)) /* Magic */}
```

لعبة تساعد في تعلم الgrid "[GRID GARDEN](#)"



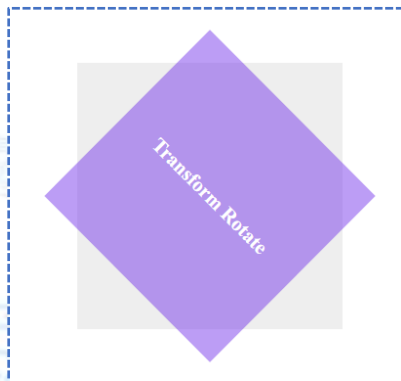
Transform “2D”



Scale: إمكانية تمدد العنصر, يمكن تحديد نسبه التمدد من الاتجاه الرأسي أو الافقي.

ملحوظه يتمدد العنصر ويتمدد معه العناصر التي بداخله, ويتمدد ال Margin أيضا مع العنصر.

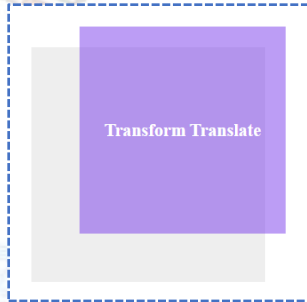
```
/* scale(X,Y) | scaleX() | scaleY() */
.child {
  transform: scale(1.5,-1); /* scale( [+Num or -Num] ) */
}
```



Rotate: امكانية تدوير العنصر بقيمة محددة.

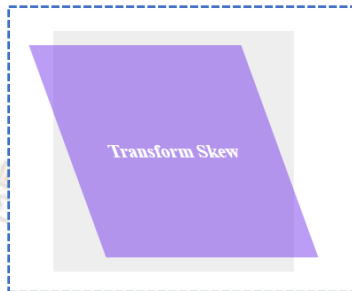
ملحوظة! يمكنك استخدام خاصية scale و rotate مع بعض لكن في سطر واحد, ويمكن استخدام خاصية transition لعمل تأثير أجمل.

```
/* Rotate( deg=> Degrees | rad=> Radians | grad=> Gradians | turn=> Turn) */
.child {transform: rotate(45deg);}
```



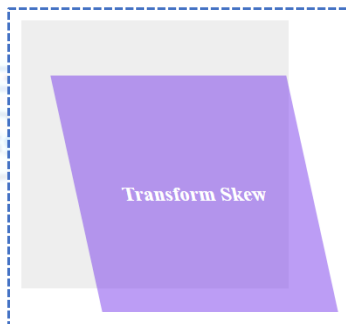
Translate: تحريك العنصر في الاتجاه الافقي والرأسي.

```
/* Translate(X,Y) | TranslateX() | TranslateY() */
.child { transform: translate(50px, -50px); }
```



Skew: امكانية انحرف العنصر من الاتجاه الافقي او الرأسي.

```
/* Skew(X,Y) | SkewX() | SkewY() */
.child { transform: skew(20deg); }
```



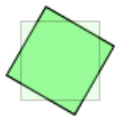
Matrix: اختصار لأضافة قيم مختلفة معا (Scale, Skew,) (translate

```
/* matrix( scaleX, skewY, skewX, scaleY, translateX, translateY ) */
.child { transform: matrix(1, 0, 0.22, 1, 50, 50) }
```

Transform Origin 2D

Transform Origin: هي النقطة التي يبدأ منها تأثير الـ `transform`, وبستخدم هذه الخاصية يمكنك اختيار المكان الذي تريد ان يبدا منه الـ `transform`.

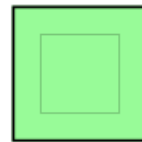
[معلومات أكثر](#)



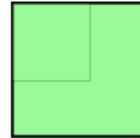
```
transform: rotate(30deg);
```



```
transform: rotate(30deg);  
transform-origin: 0 0;
```



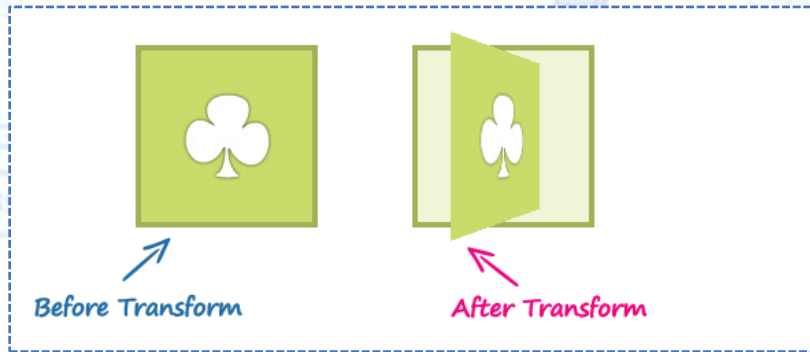
```
transform: scale(1.7);
```



```
transform: scale(1.7);  
transform-origin: 0 0;
```


Transform 3D

Rotate 3D: تدوير العنصر بشكل ثلاثي الأبعاد.



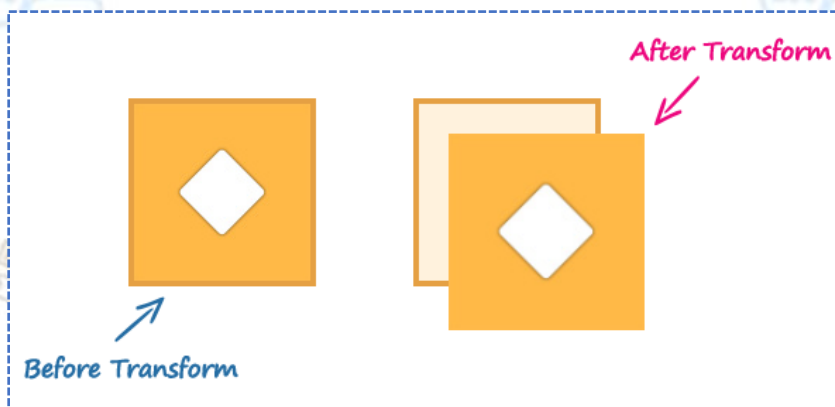
```
/* rotate3d(x, y, z, deg) */  
.child { transform: rotate3d(0, 1, 0, 60deg)}
```

Translate 3D: تحريك العنصر في الاتجاه الأفقي والرأسي والأمامي.

ملحوظة! لا يعمل الـ Translate 3D في الاتجاه Z, يجب اضافة خاصية Perspective للأب أولاً.

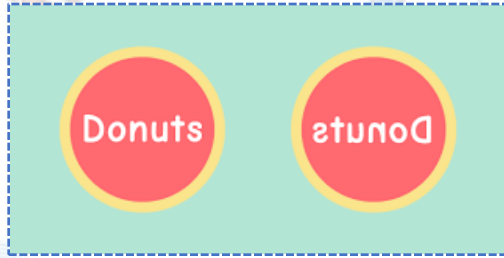
Perspective: يقوم بتحديد قيمة المسافة بين العنصر وعين المستخدم او منطقة الرؤية.

Perspective Origin: تحديد النقطة التي ننظر منها الي العنصر.



```
.parent { perspective: 500px }
```

```
.child { transform: translate3d(25px, 25px, 50px)} /* X Y Z */
```



Backface Visibility: يستخدم لأخفاء او اظهار الوجه الخلفي للعنصر, وهذا في حالة كنت تستخدم `Transform Rotate`.

```
/* Default Visible */
.face { backface-visibility: hidden }
```

Transform Style: يقوم بتفعيل تأثيرات الـ 3D علي العناصر , `Backface-visibility` لا يعمل بدون تفعيل هذا التأثير.

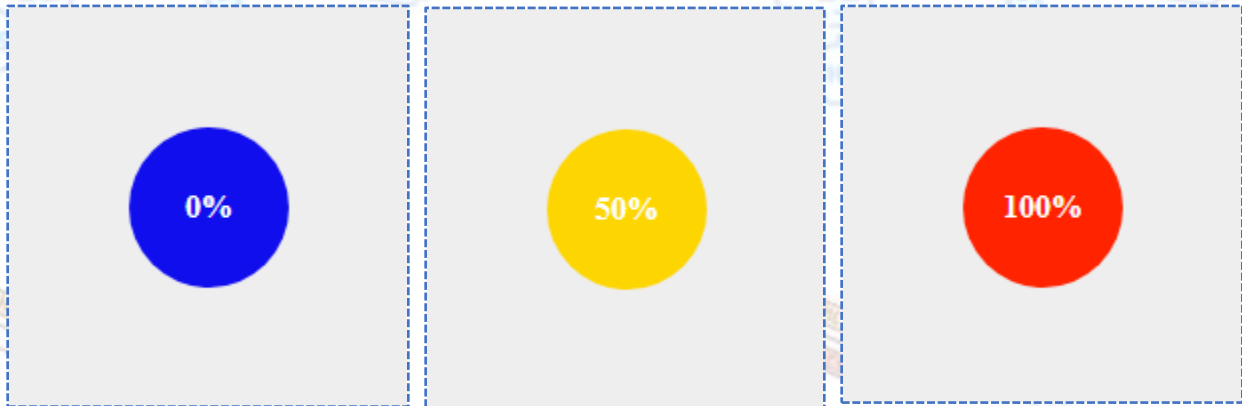
```
/* Default: flat */
.box { transform-style: preserve-3d }
```

Animation

Keyframes: يستخدم لإنشاء عدد معين من الاطارات المكون من تأثيرات Animation تحدث بالترتيب.

Animation Name: يوضع به اسم الKeyframes التي تم انشئها لتعمل علي العنصر.

Animation Duration: مدة تنفيذ الAnimation ولا يعمل بدونها.



```
.box {
  animation-name: change-color;
  animation-duration: 6s; /* Default: 0 */
}

/* @keyframes Name */
@keyframes change-color {

  0% { background-color: blue } /* from {} */

  50% { background-color: gold }

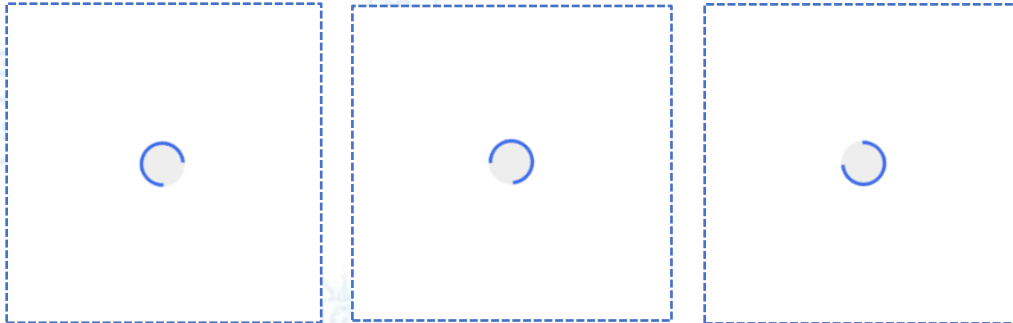
  100% { background-color: red } /* to {} */

}
```

Animation Iteration Count: تستخدم لتحديد عدد مرات تنفيذ تأثير الـ Animation.

Animation Timing Function: التحكم في كيفية عرض تأثير الـ Animation.

مثال عن تنفيذ Loading Spinner:-



```
.spinner {
  background-color: #eee;
  width: 50px;
  height: 50px;
  margin: 300px auto;
  border: solid 4px rgb(61, 107, 233);
  border-left: solid 4px #eee ;
  border-radius: 50%;

  animation-name: spinner;
  animation-duration: 1s;
  animation-iteration-count: infinite;
  animation-timing-function: linear;
}

@keyframes spinner {
  0% { transform: rotate(0deg); }
  100% { transform: rotate(360deg); }
}
```

Animation Direction: أتجاه تنفيذ الAnimation, يمكن التنفيذ بشكل عكسي او بشكل ترددي.

Animation Delay: تأخير بدأ الAnimation لوقت محدد.

Fill Mode: تجعل العنصر يحتفظ بأخر تأثير من الKeyFrames.

```
.box {  
  
  /* Default => normal | reverse | alternate | alternate-reverse */  
  animation-direction: reverse;  
  
  /* Default => 0 | second | milliseconds */  
  animation-delay: 2s;  
  
  /* Default => none | forwards | backwards | both */  
  animation-fill-mode: both ;  
  
  /* Default => running | paused*/  
  animation-play-state: paused;  
  
}
```

Animation: الطريقة المختصرة.

```
.box {  
  
  /* name duration timing-function delay iteration-count direction fill-mode */  
  animation: spinner 5s linear 0 infinite reverse both;  
  
}
```

CSS Selectors

description	Example	Selector
تحديد كل العناصر التي تحتوي علي class="box"	<code>.box { }</code>	<code>.class</code>
تحديد كل العناصر التي تحتوي علي class="box red" معاً.	<code>.box.red { }</code>	<code>.class1.class2</code>
تحديد العناصر التي تحتوي علي child التي تكون داخل parent	<code>.parent .child { }</code>	<code>.class1 .class2</code>
		,
تحديد العنصر الذي يحتوي علي Id="box"	<code>#box { }</code>	<code>#id</code>
تحديد كل العناصر.	<code>* { }</code>	<code>*</code>
تحديد كل عناصر ال<p>	<code>P { }</code>	<code>Element</code>
تحديد كل عناصر ال<P class="red">	<code>p.red { }</code>	<code>Element.class</code>
تحديد كل عناصر الp علي ان تكون من الابناء المباشرين داخل الDiv	<code>Div > p { }</code>	<code>></code>
تحديد اول عنصر بعد الDiv علي نفس المستوي.	<code>Div + p { }</code>	<code>+</code>
تحديد كل عناصر الDiv التي تسبق عنصر P علي ان تكون شقيقة لة علي نفس المستوي.	<code>P ~ Div { }</code>	<code>~</code>
كل العناصر التي تحتوي علي Attribute = title	<code>[title] { }</code>	<code>[Attribute]</code>
تحديد كل عناصر الdiv التي تحتوي علي title <div title=" ">	<code>Div[title] { }</code>	<code>Element[Attribute]</code>

تحديد كل العناصر التي تحتوي علي title="test"	[title="test"] { }	[Attribute=value]
<input type="text"> تحديد كل عناصر	Input[type="text"]{ }	Element[Attribute=value]
تحديد كل عناصر الـ title التي تحتوي علي كلمة box.	[title~=box] { }	[Attribute~=value]
تحديد كل عناصر الـ title التي تحتوي علي الحروف المحددة مثل "bo"	[title*=bo] { }	[Attribute*=value]
تحديد كل عناصر الـ title التي تبدأ بـ "b"	[title^=b] { }	[Attribute^=value]
تحديد كل عناصر الـ <p> بشرط ان تكون الابن الاول داخل الاب الخاص بها.	p:first-child { }	:frist-child
تحديد كل عناصر الـ <p> بشرط ان تكون الابن الاخير داخل الاب الخاص بها.	p:last-child { }	:last-child
تحديد عنصر الـ <p> الاول من نوعه داخل الاب الخاص به.	p:first-of-type { }	:first-fo-type
تحديد عنصر الـ <div> الاخير من نوعه داخل الاب الخاص به.	div:last-of-type { }	:last-fo-type
تحديد عنصر الـ <p> بشرط ان يكون الطفل الوحيد داخل الاب الخاص به.	p:only-child { }	:only-child
تحديد كل عناصر الـ <p> معدا التي تحتوي علي class. ملحوظه! يمكن استخدام كل انواع الـ Selector, المزيد .	p:not(.class) { }	:not(Selector)
تحديد كل عناصر الـ <p> بشرط ان يكون الابن الثاني داخل الاب الخاص به. ملحوظه! هناك طريق لتحديد العناصر مثل odd و المزيد even.	p:nth-child(2) { }	:nth-child(n)
تحديد عنصر الـ <p> بشرط ان يكون الابن الثاني من الاسفل.	p:nth-last-child(2) { }	:nth-last-child(n)
تحديد عنصر الـ <p> الرابع من نوعه داخل الاب الخاص به.	p:nth-of-type(4) { }	:nth-fo-type(n)

امثلة عن استخدامات بعض الSelectors :-

- **Root**: يمكن استخدامة بشكل مفيش في اضافت تعليمات عامة مثال:
تم انشاء متغيرات تحتوي علي الالوان الاساسية للموقع ويمكن استخدامها لحقا.

```
:root {
  --mainColor: #673ab7;
  --mainColorLight: #9c56ff
}

div { color: var(--mainColor) }
```

Lorem ipsum dolor, sit
amet consectetur
adipiscing elit. Tempore,
suscipit!

• .Selection

```
::selection {
  background-color: var(--mainColorLight);
  color: #fff
}
```

Name

• .Placeholder

يمكن استخدامها لحديد وأضافة
التنسيقات علي خاصية placeholder.

```
::placeholder {color: #818181;}
input:focus::placeholder { color: transparent; }
```

Media Queries & Responsive Designs




Media Queries: قواعد يتم استخدامها لتنفيذ تطبيق متجاوب مع وسائط وأبعاد الشاشات المختلفة.

وهناك انواع من ال Media Queries مثل ال Print وهي مسؤولة عن وضع الطباعة والتنسيقات التي ستوضع بها ستكون خاصه بوضع الطباعة فقط.

ملحوظة! يجب استخدام meta viewport وهي تعبر عن المساحة المرئية من الموقع لدي المستخدم وهي تساعد علي انشاء موقع متجاوب, [المزيد](#).

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

مثال عن طريقة الاستخدام:

Desktop	Tablet	Smartphone
		
@media screen and (min-width: 1024px) {...}	@media screen and (min-width: 768px) and (max-width: 1023px) {...}	@media screen and (max-width: 767px) {...}

[المزيد](#).

CSS Global Values

Inherit: يرث العنصر قيمة الخاصية من قيمة خصية العنصر الاب الذي بداخلة.

```
.child { color: inherit }
```

Initial: يقوم بأرجاع قيمة الخاصية الي القيمة الافتراضية التي تم انشائة عليها.

```
.child { color: initial }
```

Unset: سيعمل عمل inherit اذا كانت الخاصية قابلة للتوريث, وسيعمل عمل initial اذا كانت الخاصية غير قابلة للتوريث.

```
.child { color: unset }
```

Revert: يقوم بإعادة قيمة الخواص الي الوضع الافتراضي الخاص بالمتصفح "User Agent" وتختلف القيمة الافتراضية باختلاف المتصفحات احياناً.

```
.child { color: revert }
```

All: يرمز الي كل خواص CSS وتحديد قيمة هذه الخاصية يؤثر علي كل خواص ال CSS.

```
.child { all: initial }
```



Important links

[Color hunt](#) موقع خاص لاختيار الألوان وبه تشكيلات تتناسب مع بعضها.

[Eggradients](#) خاص بتدرجات الالوان.

[Pexels](#) موقع مجاني للحصول علي صور عالية الجودة.

[Flaticon](#) موقع جيد للحصول علي icons.

[Google Font](#) موقع به الكثير من أنواع الخطوط.

[CSS Battle](#) و [Frontend Mentor](#) مواقع ممتازة جدا لتطوير مهاراتك في CSS.