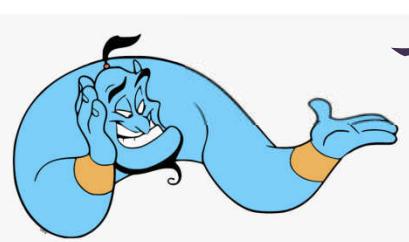


# CSS 3



2022

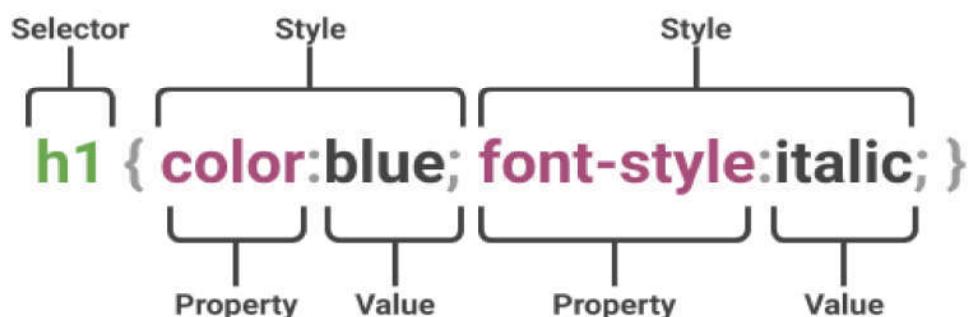


محتوى عربي بسيط

## وظيفة لغة الـ css

لغة ال CSS هي المسئولة عن عمل **style** (تنسيق) لعناصر ال HTML مثل اللون ونوع الخط ، الحجم ،....

## قواعد الكتابة syntax



## مكان كتابة الاكواد

### الطريقة الاولى (Inline)

```
<h1 style="color:red;">Content</h1>
```

### طريقة (٢) Internal

```
<head>
  <style>
    h1{
      color: red;
    }
  </style>
</head>
```

### طريقة (٣) External

↳ فى حالة وجود ملف خارجى موجود فى نفس المسار

```
<head>
  <link rel="stylesheet" href="style.css">
  <style>
  </style>
</head>
```

# دليل المبرمج CSS3-part.1

للحصول على حالة وجود ملف css داخل مجلد موجود في نفس مسار ملف الـ HTml

```
<link rel="stylesheet" href="css/style.css">
```

ملاحظة

اسم الملف الخارجي : style.css

## اضافة تعليق

طريقة اضافة تعليق وصفى للكود: لا يتم تنفيذه ولكنه بمثابة سطر يذكينا او يخبرنا بمحظى الكود يمكن استعمال الاختصار /ctrl+ / لتحويل نص الى تعليق

## الرموز البرمجية

الرمز البرمجي	الاسم	الاختصار
:	colon	ك + shift
;	semicolon	ك
" "	quotation mark	quotation mark
,	comma	و
{ }	curly brackets	د+shift ج+shift
/*write Comments*/	Comments	Ctrl+/

الأولوية التنفيذ في لغة css تكون من فوق تحت

## مثال تطبيقي

```
<head>
<!--External--&gt;
3--) &lt;link rel="stylesheet" href="css/style.css"&gt; &lt!--red--&gt;
&lt;!--Internal--&gt;
2--) &lt;style&gt;&lt;!--blue--&gt;
    h1{color: blue;}
&lt;/style&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;!--Inline--&gt;
1--)   &lt;h1 style="color:green ;"&gt;Content&lt;/h1&gt;
&lt;/body&gt;</pre>
```

الأولوية هنا لللون الأخضر ثم اللون الأزرق ثم اللون الأحمر

## important

تغيير ترتيب الاولوية ووضع اولوية اجباري

```
<div style="color: red ; font-size: 40px;" class="one">One</div>
```

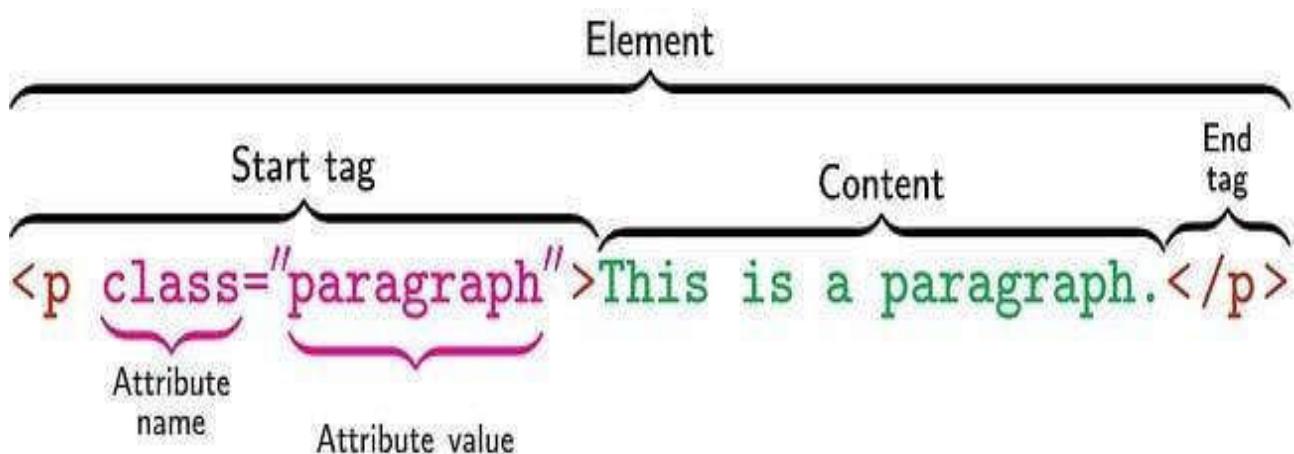
```
.one{  
    color: green !important;  
}
```

هذا الاولوية للون الاخضر لأن الخاصية important



## تميز العناصر (قص العناصر)

اذا كان لدينا عنصرين بنفس الاسم كيف نميز بينهم لكي نعطى  
لكل منهم تنسيق مختلف  
↳ عن طريق اعطاء العنصر class



# دليل المبرمج CSS3-part.1

```
<style>
p{color: red;}
.p{
  color: blue;
}
</style>
</head>
<body>
  <p>p1</p>
  <p class="p">p2</p>
</body>
```

اقتناص العنصر( selector ) المميز بـ كلاس عن طريق وضع نقطة قبل اسم العنصر

```
.p{
  color: blue;
}
```

ملاحظة : يمكن ان نميز اكثـر من عـنصر بـواسـطة كـلاـس

```
<body>
  <p class="p">p1</p>
  <p class="p">p2</p>
</body>
```

## التميـز بـواسـطة ID

**ID** يـشـبه رقم البطـاقـة ، فـريـد لا يـتـكرـر (يمـيز عـنصر وـاحـد فـقـط)

```
<p id="pop">Css</p>
```

لـعـمل selector نـصـبـ شـبـاك قـبـلـ العـنصر

```
#pop{
  color: rgb(57, 232, 13);
  background-color: rgb(31, 124, 206);
}
```

## طريقة الكتابة

Case Type	Example
Original Variable as String	some awesome var
Camel Case	someAwesomeVar
Snake Case	some_awesome_var
Kebab Case	some-awesome-var
Pascal Case	SomeAwesomeVar
Upper Case Snake Case	SOME_AWESOME_VAR

## تنسيق النصوص

### تغيير الالوان

p{color: red;}

الألوان السداسية hexadecimal

p{color: #FC0000}

الألوان (rgb) يوجد بها خاصية

p{color: rgb(255, 0, 0); }

تضليل خلف الفقرة او النص

{background-color: red; }

**bold**

جعل النص ثقيل bold

font-weight: bold;

اعادة الضبط للوضع الافتراضي

font-weight: normal;

# دليل المبرمج CSS3-part.1

الشفافية

```
{opacity:0.5 ; }
```

```
<div class="container">
  <div class="div_One">Product One</div>
  <div class="div_Two">Product Two</div>
</div>
```

```
.continer{
background-color: #ff0057;
opacity: 0.5;
}
```

تدرج اللون من اتجاه معين

```
p{
  background-image: linear-gradient(to right, white,red);
}
```

الاتجاه

```
div { direction: rtl; }
```

```
<div>
  . شكر لكم .
</div>
```

```
div {
direction: rtl;// Arabic
direction: ltr; english
}
```

المحاذة

محاذة النص اتجاه اليمين او الشمال او الوسط center

```
text-align: left;
```

التباعد والمسافات

التباعد بين حروف الكلمات

```
letter-spacing: 7px;
```

التباعد بين الكلمات

```
word-spacing: 20px;
```

التباعد بين الاسطر

```
line-height: 40px;
```

لعمل مسافة في اول الكلام للسطر الاول فقط

```
text-indent: 60px;
```

## Using text-indent

In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since. 'Whenever you feel like criticizing anyone,' he told me, 'just remember that all the people in this world haven't had the advantages that you've had.'

### underline

لوضع خط تحت النص

```
text-decoration: underline;
```

لوضع خط فوق النص

```
text-decoration: overline;
```

لإزالة الخط الموجود تحت اللينك

```
a{  
    text-decoration: none;  
}
```

### للشطب على كلمة

```
text-decoration: line-through;
```

### حجم الحروف

الحجم الافتراضي للخطوط 16px

لتحويل اول حرف من كل كلمة (حروف كبيرة)

```
text-transform: capitalize;
```

## لجعل الحروف كلها كبيرة

```
text-transform: uppercase;
```

تحويل الحروف الكبيرة الى حروف صغيرة

```
text-transform: lowercase;
```

لجعل النص غير مرتبط بحجم ال div

```
white-space: nowrap;
```

## طريقة استخدام نوع الخط

```
font-family: "Comic Sans MS", cursive, sans-serif ;
```

## أفضل الخطوط الآمنة للوب

- Arial (sans-serif)
- Verdana (sans-serif)
- Tahoma (sans-serif)
- Trebuchet MS (sans-serif)
- Times New Roman (serif)
- Georgia (serif)
- Garamond (serif)
- Courier New (monospace)
- Brush Script MT (cursive)

[https://www.w3schools.com/cssref/css\\_websafe\\_fonts.php](https://www.w3schools.com/cssref/css_websafe_fonts.php)

```
<title>CSS</title>
    <link rel="preconnect" href="https://fonts.googleapis.com"><link
rel="preconnect" href="https://fonts.gstatic.com" crossorigin><link
href="https://fonts.googleapis.com/css2?family=Cairo:wght@900&display=s
wap" rel="stylesheet">
```

```
body{
    font-family: 'Cairo', sans-serif;
}
```

## وحدة القياس em

وحدة القياس **em** تساوى نفس حجم الخط من عنصر الاب ، وفي حالة لم يكن هناك عنصر اب يحتوى على خاصية **font-size** فى هذه الحالة ستكون **1em** بحجم **16px**

```
<div>
    welcom this my wep
    <span>Test em</span>
</div>
```

```
div {
    font-size: 16px;
}
div span {
    font-size: 2em;
}
```

فى هذا المثال **2em** تساوى  $16 * 2 = 32$   
فى حالة اذا كان عنصر الاب لا يحتوى على حجم للخط ، سيأخذ المساحة من  
الجذر **body**

```
<div>
    welcom this my wep
    <span>Test em</span>
</div>
```

```
body {
    font-size: 40px;
}
div {
}
div span {
    font-size: 2em;
}
```

فى هذه الحالة ستكون المساحة  $40 * 2 = 80$

## وحدة القياس rem

تأخذ معيار المساحة من الجذر Html

```
<div>
    welcom this my wep
        <span>Test em</span>
</div>
```

```
body{
    font-size: 90px;
}

div span {
    font-size: 2rem;
}
```

في المثال السابق سيأخذ حجم الخط من الحجم الافتراضي من الجذر Html وليس  
 $(2*16=32) -- 16 \text{px}$  وهو body

الجذر Html مساحة الخط الافتراضي له 16px

```
html{
    font-size: 16px;
}
```

النسبة المئوية :

```
<div>
    welcom this my wep
        <span>Test em</span>
</div>
```

```
div {
    font-size: 200px;
}
div span {
    font-size: 50%; /*50% from 200px = 100px*/
}
```

تعنى ان حجم الابن سيأخذ 50% من حجم عنصر الاب

## وحدة القياس 1vw



تعنى منطقة العرض الظاهرة  
1vw تعنى 1% من مساحة العرض الكلية للصفحة  
فى هذا المثال منطقة العرض px 1000  
مساحة العنصر div تساوى 10vw اي 10% من المساحة الكلية للصفحة

```
div {  
    width: 10vw;  
background-color: rgb(118, 111, 22);  
}  
  
<div>  
    welcom this my wep Test vw  
</div>
```

## عمل شادو للنص

```
h1{  
color: rgb(0, 0, 0);  
text-shadow: 1px 1px 10px #ff0000;  
}
```

القيمة الاولى فوق وتحت – والثانية لمين وشمال والثالثة للنعومة

# Ahmed sabry

لعمل اكثـر من شادو

```
h1{  
color: rgb(0, 0, 0);  
text-shadow: 1px 2px 4px #ff0000,  
2px 2px 1px #d1ce08;  
}
```

# Ahmed sabry

## قائمة العناصر المرتبة "ul"

- وضع نقاط قبل كل عنصر في القائمة

```
ul{ list-style-type: disc; }
```

لإزالة الترقيم " تستخدم في ال nav

```
ul{list-style-type: none;}
```

لجل الترقيم داخل العناصر

```
ul{ list-style-position: inside; }
```

لجل الترقيم خارج ال عناصر

```
ul{ list-style-position: outside; }
```

لوضع صورة ( ادراج رمز ) مكان الترقيم

```
ul{ list-style-image: url(img.jpg); }
```

(( الحدود ))

## لعمل اطار للعناصر

```
ul li{ border: 1px solid #dddddd; }
```

لعمل اطار غير كامل

```
border-left: 2px outset ;  
border-left: 2px inset;
```

لعمل اطار ناقص ضلع

```
border-width: 2px;  
border-style: solid solid solid none;  
border-color: yellow;
```

## انواع اشكال الاطار

Solid - or - dotted -double - groove -ridge

this my frist test wep

## لعمل اطار لمين وشمال فقط

```
border-right: 2px inset;  
border-left: 2px outset;  
border-color: yellow;
```

this my frist test wep

## لعمل دائرة border-radius

### الوضع الافتراضي لهذه الخاصية 0px ويظهر كمربع

```
<div></div>
```

```
div {  
    width: 100px;  
    height: 100px;  
    background-color: red;  
    margin: auto;  
    border-radius: 50px;  
}
```

لعمل دائرة يجب ان يكون  
مساحة الطول والعرض متساويان  
يجب ان تكون الـ **border-radius** تساوى نصف مساحة الطول والعرض

يمكن التحكم في اطراف المربع من اتجاه محدد

```
div {  
    width: 100px;  
    height: 100px;  
    background-color: red;  
    margin: auto;  
    border-radius: 0px;  
    border-top-right-radius: 20px;  
    border-bottom-left-radius: 20px;  
}
```

## الفرق بين ال outline وال border

```
<div class="div1">  
Div one  
</div>  
  
<div class="div2">  
    DIV tow  
</div>
```

```
.div1{  
    border: 8px solid;  
color: blue;  
width: 100px;  
background-color: brown;  
margin-bottom: 10px; }  
  
.div2{  
    outline: 8px solid green;  
  
width: 100px;  
color: blue;  
background-color: crimson; }
```

Div one

DIV tow



العنصر يفضل مكانة **Outline** والحدود بتترسم حوله  
يعمل ازاحة للعنصر بمقدار حجم الحدود **border**

## CSS Selector

### :Before, :After و استخدامها مع "Content"

```
<div> open your photo </div>
<div> open your video </div>
```

```
div::before
{
color: lightcoral;
content: "click here to";
}
```

النتيجة ظهور نص ال **content** قبل جميع النصوص الموجودة في **div**

[click here to open your photo](#)  
[click here to open your video](#)

مثال متقدم لتوضيح الفكرة :

```
<div class="one">This is some text in a div element.</div>
<div class="one">This is some text in a div element.</div>
```

```
.one{
background-color: cornflowerblue;
width: 60px;
position: relative;
border: 1px solid black;
}
.one::after{
content: "";
position: absolute;
width: 3px;
height: 50px;
right: -8px;
top: 0;
background-color: blue;
border-radius: 30px;}
.one::before
{content: "";
width: 20px;
height: 20px;
background: red;
position: absolute;
left: -20px;
top: 0px;
}
```

## التفاعل مع الاحداث

### عند تمرير الماوس على العنصر hover

تغير لون اللينك الحالى

#### Hover: لاضافة الخط عند الوقوف على اللينك بالماوس:

```
a:hover{  
    text-decoration:underline ;  
}
```

#### Hover: تغير لون اللينك الحالى عند الوقف عليه:

```
a:hover{  
    color: teal;  
}
```

### تطبيق على الحدود التمسات الجمالية للصور

```
img{  
    margin: 50px auto;  
    border-radius: 40%;  
    padding: 5px;  
    background-color: rgb(93, 18, 18);  
    border: 1px solid #463838;  
    box-shadow: -1px 0px 10px 10px #daa909;  
    transition:all.5s ease-in-out  
}
```



لو حابين نخلی الشادو ينور عند الوقف بالماوس على الصورة

```
img:hover{  
    box-shadow: -1px 0px 10px 10px #daa909;
```

## عند النقر على العنصر active

تتغير لون اللينك الحالى عند الضغط عليه

```
a:active{  
    color: rgb(116, 3, 72);  
}
```

## اللينكات التي تم زيارتها مسبقا visited

```
<a href="https://www.twitter.com/">link_1</a>  
<a href="https://alibakr.000webhostapp.com/">link_2</a>
```

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

## focus

عند النقر داخل عنصر ادخال

```
<input class="class_input" type="text">
```

```
.class_input:focus{  
background-color: yellow;  
border-color: rgb(177, 12, 158);  
outline: none;  
}
```

## Cursor

تغيير شكل الماوس عند الوقف على العنصر Cursor:

```
cursor: help; /* pointer */
```

ex <https://codepen.io/chriscoyier/pen/uCwfB>

checked عند التحديد

```
<input class="class_checkbox" type="checkbox">
```

```
.class_checkbox:checked {display: none;}
```

الغاء الانتقال ( الرابط التشعبي )

```
<a href="https://www.google.com/">google</a>
```

```
a {  
    pointer-events: none;  
}
```

## CSS Pseudo-classes

التحكم في خواص عنصر من خلال عنصر اخر عند حدث معين

```
<input class="class_checkbox" type="checkbox">  
<label for="">label_checkbox</label>  
  
.class_checkbox:checked +label{  
color: red;  
}
```

في المثال السابق عند الحدث **checkbox** سنتحكم في خواص العنصر **label**

التحكم في خواص عنصر من خلال عنصر اخر عند حدث معين

```
input[type="checkbox"]:checked +label{  
color: red;  
}
```

```
<input class="class_checkbox" type="checkbox">  
<label for="">label_checkbox</label>
```

للتحكم في عنصر من خلال نوعه وخصائصه

```
input[type="text"]:disabled {  
background-color: yellow;  
}
```

```
<input class="class_checkbox" type="text"  
disabled value="text_name">  
<label for="">label_checkbox</label>
```

مثال اخر

```
input:required +label{color: red;}
```

```
<input type="text" required value="text_name">  
<label for="">label_checkbox</label>  
<input type="text">
```

مثال اخر

```
input:required + span:before{  
content:'****';  
color: red;  
}
```

```
<input type="text" required value="text_name">  
<span></span>
```

## Pseudo Elements

التحكم في الحرف الاول من النص

```
<div class="one">welcom</div>
```

```
.one::first-letter{  
background-color: red;  
color: wheat;  
font-size: 60px;  
font-weight: bold;  
}
```

التحكم في السطر الاول

```
.one::first-line{  
color: blue;}
```

```
<div class="one">  
Lorem ipsum dolor sit amet consectetur  
adipisicing elit. Beatae delectus maiores, sint similique omnis  
pariatur at sit sed error velit, doloremque explicabo facilis!  
Laboriosam quos non excepturi at quas exercitationem?  
Lorem ipsum, dolor sit amet consectetur adipisicing elit. Earum  
laboriosam aperiam non tenetur eveniet? Ut dolor quod ab animi soluta,  
similique pariatur totam vel repudiandae, nobis odio, facere delectus  
quasi?  
</div>
```

التحكم في لون التحديد

## selection

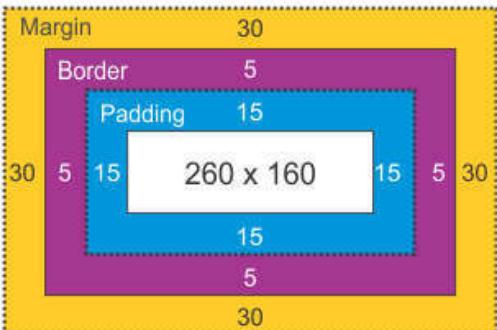
```
::selection {  
color: red;  
background: yellow; }  
.on::selection{ background: blue; }
```

```
<h1>Select some text on this page:</h1>  
<p class="on">This is a paragraph.</p>  
<div>This is some text in a div element.</div>
```

:: تعنی اي عنصر يحتوى على الخاصية

```
::placeholder  
{ color: red; }  
<input type="text" placeholder="write_name">
```

## المسافات حول المحتوى



Padding  
border  
margin

## padding

المسافة الداخلية **padding** هي المسافة بين المحتوى المكتوب وبين الحدود من (اعلى - اسفل - لمين - شمال )

```
h1{ padding-top: 20px;  
    padding-bottom: 20px;  
    padding-right: 20px;  
    padding-left: 20px;  
    border: solid 2px;  
}
```

(قيمة واحدة) مسافة موحدة من جميع الاتجاهات

```
h1{padding: 10px;}
```

(قيمتين) مسافة موحدة (اعلى واسفل) - (لمين وشمال )

توضيح وليس كود

```
padding: top-bottom right-left;
```

```
h1{ padding: 30px 20px ; }
```

ثلاث قيم : (من فوق)- (لمين+شمال) - (من تحت)

```
h1{  
    padding: 30px 20px 10px;  
}
```

اربع قيم مثل عقارب الساعة : (فوق)- (لمين) (تحت) (شمال)

```
h1{ padding: 30px 20px 10px 5px; }
```

## margin

هى الهوامش التى تحيط بالـ (border)  
وظيفته : عمل هوامش "مسافات " حول العنصر حسب الاتجاه المحدد  
**التعامل مع ال margin**

ترتيب القيم مثل عقارب الساعة (فوق - لمين - تحت - شمال )

```
margin: 10px 20px 10px 40px;
```

عن طريق تحديد اسم الاتجاه (top - right- bottom- left)

```
h1{  
    border: solid 2px;  
    margin-top: 10px;  
    margin-right: 10px;  
    margin-bottom: 10px;  
    margin-left: 10px;  
}
```

قيمة واحدة (مسافة موحدة من جميع الاتجاهات)

```
h1{margin: 5px ;  
}
```

كودين بنفس الوظيفة **margin-left**

```
margin-left: 50px;  
margin: 0 0 0 50px;
```

**auto**

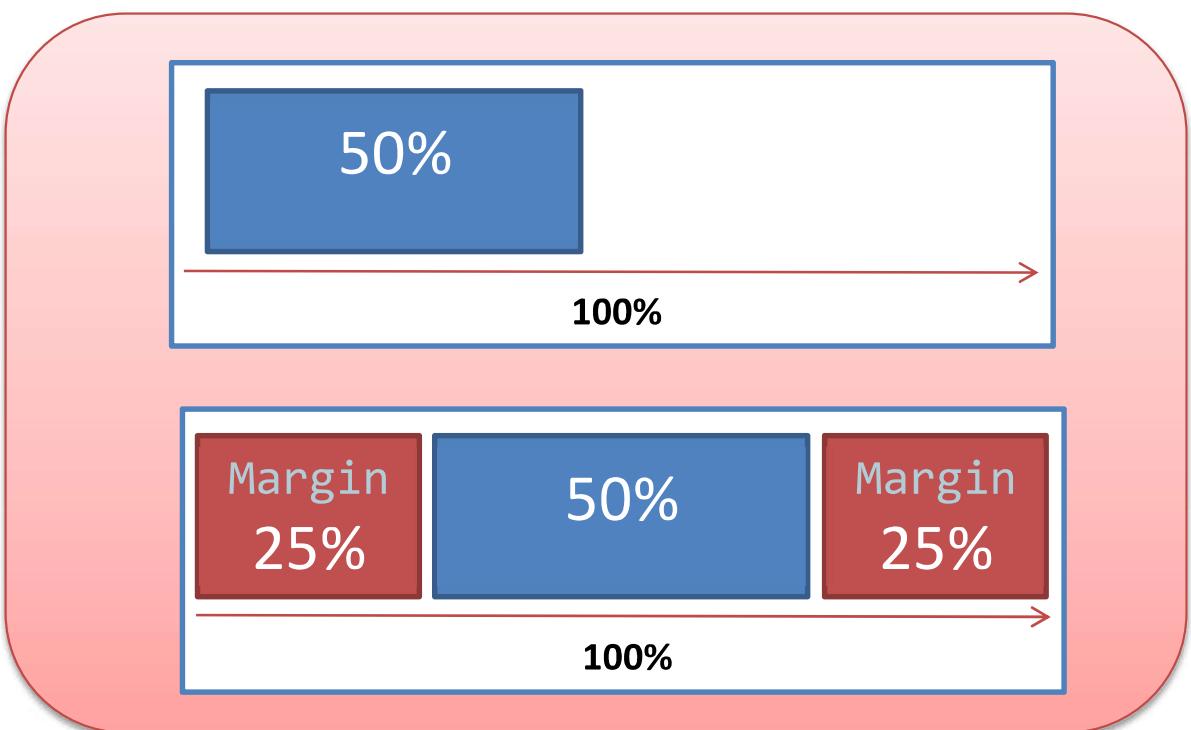
لجعل المسافة الفارغة اللي على اليمين تساوى المساحة الفارغة للشمال

```
margin: auto;
```

: مثال

```
<style>  
    div {background-color: #DDD;  
        width: 50%;  
        margin-left: 25%;  
        margin-right: 25%;  
    }  
</style>
```

تحليل : المساحة الكلية لل div 50% يبقى لنا مساحة 50% يمكن بالطريقة التقليدية ان نحدد مقدار الازاحة من الشمال بمقدار 25 ومن اليمين 25 لكي يتم توسيط العنصر



## تحديد أقل وأقصى مساحة للعنصر

لتحديد أقل قيمة ممكنة للعرض

```
min-width: 50px;
```

لتحديد أكبر قيمة ممكنة للعرض

```
max-width: 100;
```

```
max-height: 400px;  
min-height: 200px;
```

## Margin Collapse

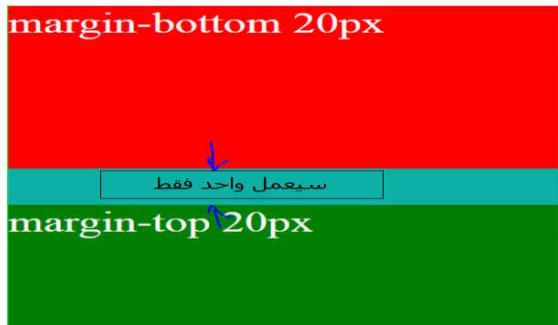
الهوامش المقابلة بالطول اذا تساوت المساحة سيعمل واحد منهم فقط

# CSS3-part.1 دليل المولى

```
<div class="parent">
  <div class="one"></div>
  <div class="two">
    <div class="up"></div>
    <div class="down"></div>
  </div>
</div>
```

```
.parent {
  overflow: hidden;
  margin: auto;
  width: 400px;
  height: 200px;
  padding: 10px;
  background-color: #eee;
}
.parent .one,
.parent .two {
  float: left;
  width: 50%;
  height: 100%;
}
.one {
  background-color: rgb(44, 233, 63);
}
.two {
  background-color: rgb(14, 175, 164);
}
.up {
  background-color: red;
  color: white;
  height: 90px;
  margin-bottom: 20px;
}
.down {
  background-color: green;
  color: white;
  height: 90px;
  margin-top: 20px;
}
```

الحاوية tow ارتفاعها 200px تحتوى على Div\_up ارتفاعه 90px وDiv\_Down ارتفاعه 90px ، مجموع ارتفاع العنصرين 180px ويوجد مساحة فارغة بمقدار 20px من العنصر الاول ، ثم قمنا بعطاء العنصر الداخلى الاول مساحة فارغة بمقدار 20px فاصبح مجموع المساحة 200px الان لو قمنا بعطاء العنصر الثانى 20px فلن يعمل لأن المساحة قد اكتملت



لله الهاشم الاكبر بياخذ الاولوية في التنفيذ والهاشم

```
.up {  
    background-color: red;  
    color: white;  
    height: 90px;  
    margin-bottom: 20px; // will run  
}  
.down {  
    background-color: green;  
    color: white;  
    height: 90px;  
    margin-top: 5px; not work  
}
```

في حالة وجود فاصل بين الهاشم : سوف يعمل الهاشين معاً

```
<div class="parent">  
    <div class="one"></div>  
    <div class="two">  
        <div class="up">margin-bottom 20px </div>  
        <hr>  
        <div class="down">margin-top 5px </div>  
    </div>
```

## Display - Block, Inline Block, Inline

```
/* display: block; */
```

- ١- يأخذ العرض بالكامل بشكل افتراضي - لو لم يتم تعيين قيمة
- ٢- يقوم بإضافة سطر اسفل العنصر
- ٣- يحترم الهوامش الداخلية والخارجية + الطول والعرض

```
/* display: inline; */
```

- ١- يأخذ عرض النص فقط
- ٢- لا يضيف سطر اسفل العنصر
- ٣- لا يحترم الطول والعرض
- ٤- يحترم padding وال margin من الليمين والشمال فقط
- ٥- العنصر الذى سياتى بعده سيكون فى نفس السطر

```
/* display: inline-block; */
```

- ١- لا يضيف سطر اسفل العنصر
- ٢- يحترم الهوامش الداخلية والخارجية + الطول والعرض

لعمل ٢ div جنب بعض فى سطر واحد

```
margin-right: 20px;  
display: inline-block;
```

this is div 1

this is div 2

الخاصية width: fit-content;

المعتاد ان العنصر block يأخذ عرض بالكامل ولكن اذا اضفنا الخاصية fit-content، سيكون عرض div بحجم المحتوى ولن يأخذ العرض بالكامل

## ملحوظة : fit-content افضل من استعمال inline-block

```
<div>
  Div1Div1Div1Div1Div1Div1Div1Div1Div1Div1Div1Div1
</div>
```

```
div {
  background-color: red;
  padding: 10px;
  display: block;
  width: fit-content;
}
```

مثال قبل وبعد

Div1Div1Div1Div1Div1Div1Div1Div1Div1Div1Div1Div1

Div1Div1Div1Div1Div1Div1Div1Div1Div1Div1Div1

## display: none

لو عندك ٢ div وعملت للأول display: none هيختفي وهيحل مكانة (في نفس الموضع) ال div الثاني

```
<div class="one">one</div>
<div class="tow">tow</div>
```

```
div {
  background-color: red;
  margin: 10px 0;
}
.one {
  display: none;
}
```

## Visible

لو عندك ٢ div وعملت للاول **hidden** والثاني **visible** فسيكون الثاني يظهر

```
<div class="one">one</div>
<div class="tow">tow</div>
```

```
div {
    background-color: red;
    margin: 10px 0;
}
.one {
    visibility: hidden;
}
```

## تجميع العناصر المشتركة في نفس خصائص

يستخدم الـ **Selectors** لتجميع العناصر المشتركة

```
.one,.tow,.three{
    color: mintcream;
    padding: 10px;
    margin: 10px;
    width: 500px; }

.one{ background-color: mediumblue; }
.tow{ background-color: rgb(48, 245, 13); }
.three{ background-color: rgb(114, 12, 12); }
```

## (التعشيش) Nesting

وظيفته : اقتناص عنصر داخل عنصر اب

```
<div>
    <h1>Developer</h1>
    <p>Ahmed in Div</p>
</div>
```

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

مثال اخر

```
<div>
    <p class="pop">Ahmed in Div</p>
</div>
    <p class="pop">Ahmed out Div</p>
```

```
div .pop{
    color: green;
}
```

↳ .. سيتم التطبيق على العنصر p الموجود داخل تاج div

## الselector

```
<div class="main"> this is the main div

    <aside> this is the aside
        <div> this is the div from aside </div>
    </aside>

        <p> this is child prgraf </p>
        <p>this is child prgraf</p>

        <div>this is child Div</div>
        <div>this is child Div</div>
</div>
```

لاحظ / الكلاس main حاوية لم لكل المحتوى الذى بداخله  
لاحظ ان هناك div ضمن وسم aside  
← وضع خصائص لاي div موجود فى الكلاس main

# دليل المبتدئين CSS3-part.1

```
.main{  
    background-color: midnightblue;  
    color: mintcream;  
    padding: 10px;  
}  
.main div{  
    background-color: orangered;  
    color: papayawhip;  
    padding: 10px;  
    margin: 10px;  
}  
.main aside{  
    background-color: rgb(14, 221, 204);  
    color: rgb(0, 0, 0);  
    padding: 10px;  
    margin: 10px;  
}  
.main p{  
    background-color: rgb(128, 231, 9);  
    color: rgb(34, 22, 2);  
    padding: 10px;  
    margin: 10px;  
}
```

النتيجة : ولكن ال **div** الموجود داخل ال **aside** لم يأخذ لون وخصائص ال **aside** الموجود بداخلها عشان اطبق عليه خصائص الكلاس **main**



## تطبيق خصائص على عنصر (Div) مباشر داخل كلاس main

```
.main > div{  
    background-color: orangered;  
    color: papayawhip;  
    padding: 10px;  
    margin: 10px;  
}
```

بمعنى طبق الخصائص على جميع الـ div المباشر فقط للكلاس main

```
<div class="main"> this is the main div  
  
    <aside> this is the aside  
        <div> this is the div from aside </div>  
/* لا يعنى عنصر مباشر للكلاس لأنه وضع فى وسم آخر */  
    </aside>  
        <p> this is child prgraf </p>  
        <p>this is child prgraf</p>  
        <div>this is child Div</div>  
        <div>this is child Div</div>  
</div>
```

## اقتناص عنصر له اكثر من كلاس

```
<div class="class_one class_tow"> tist </div>
```

```
.class_one.class_tow{  
    background-color: red;  
}
```

## اقتناص عنصر داخل عنصر / و اقتناص عنصر داخل عنصر

```
.class_one div , .class_tow p{  
    background-color: red;  
}
```

```
<div class="class_one">  
    <div>Div_testing</div>  
</div>  
<div class="class_tow">  
    <p>P_testing</p>  
</div>
```

## اقتناص اى برجراف يحتوى على هذا الكلاس

```
p.class_name{  
background-color: red;  
}  
</style>  
</head>  
<body>  
<div class="class_name">  
Div_testing  
</div>  
<p class="class_name">  
P_testing  
</p>
```

## اقتناص - عنصر محدد لا يحتوى على محتوى

```
div:empty {  
width: 100px;  
height: 20px;  
background: #ff0000;  
}
```

```
<div></div>  
<div>Div_not_empty</div>
```

## اقتناص اى عنصر لا يحتوى على محتوى

```
:empty {  
width: 100px;  
height: 20px;  
background: #ff0000;  
}
```

## تحديد العناصر بدون عمل class

للتحكم في تنسيق أول سطر ولو تم ترحيل الكلام الى السطر الى بعده اثناء تكبير الصفحة لن يتم تلوين الا السطر الاول فقط

```
p::first-line{  
    color: rgba(182, 12, 83, 0.671);  
}
```

لتحديد اي برجراف (بعد منه مباشر div)

```
p+div  
{  
background-color: lightcoral;  
}
```

```
<p>this prgraf one </p>  
<p>this prgraf tow </p>  
<div> this Div next prgraf tow </div>  
<p>this other </p> //next direct div  
<div>this Div next prgraf other</div>
```

# دليل المبرمج CSS3-part.1

لتحديد اي div ياتى بعد برجراف ((شرط ان لا يكون داخل وسم )) اخر  
بمعنى اخر : اي div موجود فوق منه برجراف " مش شرط فوق منه مباشر"

```
P ~ div
{
    background-color: lightcoral;
}
```

```
<div>firest_Div</div> /* لن يتم التنفيذ /
<p>p</p>
<div>Div after p</div>
<h1>h1</h1>
<div>Div after h1</div>
<p>p</p>
/* section على الحاوية الموجودة داخل ال
<section>
    <div>Div_in_section</div>
</section>
<div>Div after p</div>
```

## للتحكم فى اول برجراف ياتى بعد ال parent div

```
div p:first-child{
    background-color: maroon;
}

<div>div1
    <p>firest p after parent div </p> /* اول ابن برجراف داخل حاوية
    <p>p2</p>
    <p>p3</p>
</div>
```

## مثال اخر للتوضيح

```
p:first-child{color: red; }

<p>p</p> <!-- الابن الاول للعنصر
<span>Div</span>
<div>
    <p>p1</p> <!-- Div
    <p>p2</p> <!-- Div
</div>
```

# دليل المبرمج CSS3-part.1

## اختيار اخر ابن داخل اب

```
div p:last-child{color: red;}  
  
<div>div1  
  <p>p1 <span>span_1</span></p>  
  <p>p2 <span>span_2</span></p>  
  <p>p3</p>  
</div>
```

```
div p:last-child{color: red;}
```

## اختيار اخر ابن داخل اب

```
<div>div1  
  <p>p1  
    <span>firest_p1</span>  
    <span>last_p_1</span>  
  </p>  
  <p>p2  
    <span>firest_p2</span>  
    <span>last_p_2</span>  
  </p>  
  
  <p>p3  
    <span>firest_p3</span>  
    <span>last_p_3</span>  
  </p>  
  
</div>
```

## اختيار اول ابن نوعه برجراف

```
p:first-of-type{color: red;}
```

```
<div class="container">  
  <div>Div_1</div>  
  <section>section_1</section>  
  <p>p_1</p>  
  <p>p_2</p>  
</div>
```

## اختيار اخر ابن نوعه برجراف

```
p:last-of-type{color: red;}
```

```
<div class="container">
  <div>Div_1</div>
  <section>section_1</section>
  <p>p_1</p>
  <p>p_2</p>
</div>
```

اقتناص برجراف اذا كان لا يوجد ابناء في الاب غيره

```
p:only-child{color: red;}
```

```
<div class="container">
  <section>section_1</section>
  <p>p_1</p>
</div>
```

لن يتم الاقتناص الا بعد حذف الـ section

اقتناص (ابن وحيد) من نوع برجراف داخل اب .  
في حالة اذا كان الاب يوحتوى على اكثر من برجراف لن يعمل الكود

```
p:only-of-type{color: red;}
```

```
<div class="container">
  <section>section_1</section>
  <p>only_p</p>
</div>
```

## للتحكم فى اول حرف فقط فى السطر

```
p::first-letter{
```

التحكم فى خصائص الـ `li` من خلال وظيفته واسمها

```
<a href="#" title="test">Home</a>
```

```
a[title=test]
{
    color: lightcoral;
}
```

التحكم فى لون النص داخل التكს من خلال نوعه او اسمه

```
<input type="search">
```

```
input[type="search"]
{
    color: lightseagreen;
}
```

للتحكم فى اي عنصر يحتوى على خاصية معينة

```
[type="password"]
{
background-color: yellow;
color: red;
}
```

للتحكم فى سمة تحتوى على الكلمة

```
[title~=superman] {
    border: 5px solid yellow;
}
```

```

```

للتحكم فى سمة تحتوى على الكلمة

(حتى لو الكلمة مربوطة بكلمة اخرى ) سيمكن من قراءة الكلمة

```
[title*=superman] {
    border: 5px solid yellow;
}
```

```

```

# دليل المبرمج CSS3-part.1

للتحكم في اي خاصية تبدا بكلمة معينة

```
[title^=yes] { border: 5px solid yellow; }
```

```
  
  

```

اقتناص جميع الفقرات ماعدا الابن الاول

```
p:not(:first-child){color: red;}
```

```
<div class="container">  
  <p>first-child</p>  
  <p>2-child</p>  
  <p>3-child</p>  
</div>
```

اقتناص جميع الفقرات ماعدا عنصر لديه كلاس معين

```
p:not(.class_name){color: red;}
```

```
<div class="container">  
  <p>1-child</p>  
  <p>2-child</p>  
  <p class="class_name">3-child</p>  
</div>
```

اختيار ابن حسب ترتيبه

```
p:nth-child(3){color: red;}
```

```
<div class="container">  
  <p>1-child</p>  
  <div>Div_1</div>  
  <p>2-child</p> <!--index3-->  
  <p>3-child</p>  
</div>
```

اقتناص الارقام الزوجية

```
p:nth-child(even){color: red;}
```

```
<div class="container">  
  <p>1-child</p>  
  <p>2-child</p>  
  <p>3-child</p>  
  <p>4-child</p>  
</div>
```

# دليل المبرمج CSS3-part.1

## اقتناص الارقام الفردية

```
p:nth-child(odd){color: red;}
```

## اقتناص العنصر الاخير

```
p:nth-last-child(1){color: red;}
```

## اقتناص العنصر قبل الاخير

```
p:nth-last-child(2){color: red;}
```

## اقتناص ثالث عنصر ابن من نوع فقرة

```
p:nth-of-type(3){color: red;}
```

```
<div class="container">
  <p>1-child</p> <!--1-->
  <div>Div_1</div>
  <p>2-child</p> <!--2-->
  <p>3-child</p> <!--3-->
  <p>4-child</p>
</div>
```

## اقتناص اخر عنصر ابن من نوع فقرة

```
p:nth-last-of-type(1){color: red;}
```

```
<div class="container">
  <p>1-child</p>
  <div>Div_1</div>
  <p>2-child</p>
  <p>3-child</p>
  <p>4-child</p>
  <section>section_1</section>
</div>
```

## Overflow

وظيفته : اذا كان هناك Div له ارتفاع محدد وبداخله "نص" اكبر من الارتفاع المطلوب يمكن اخفاء محتوى النص الذى يزيد طوله عن ارتفاع Div او عمل scroll

```
<div>
    Lorem ipsum dolor sit amet consectetur adipisicing elit. Veritatis
    quidem quia tenetur. Iusto neque corrupti eum quaerat et quam quisquam,
    odio debitis necessitatibus fugit? Odio maxime quia debitis veritatis
    repellat!
</div>
```

```
    <style>
        div {
            background-color: coral;
            height: 150px;
            /* hidden or scroll */
            overflow: hidden;
            overflow: scroll;
        }
    </style>
```

ملحوظة: فى حالة كان النص اصغر من ارتفاع ال Div واردنا استعمال الخاصية scroll سيظهر scroll ، ونحن لسنا فى حاجة الى اظهاره

**الحل البديل استعمال الخاصية overflow: auto;**

```
div {
    height: 150px;
    overflow: auto;
}
```

## حذف النص الذى يزيد عن حجم Div من ناحية (x) العرض والطول (y) (overflow-x overflow-y )

```
<div>
    https://www.youtube.com/watch?v=FQLRSbVdk28&list=PLDoPjvoNmBAzjsz06
    gkzlSrlev53MGIKe&index=16
    Lorem ipsum dolor sit, amet consectetur adipisicing elit.
    Perferendis inventore officia accusamus laboriosam assumenda id
    molestiae consequatur iste deserunt? Maiores, consectetur voluptatem?
    Expedita iusto unde eligendi, odit voluptate veniam recusandae.
</div>
```

```
div {
height: 150px;
width: 150px;
background-color: coral;
overflow-y: hidden;
overflow-x: hidden;
}
```

## Text – Overflow

```
<div title="Welcome To Elzero Web School, Hello">
    Welcome To Elzero Web School, Hello
</div>
```

```
div {
background-color: #ddd;
padding: 10px;
width: 200px;
white-space: nowrap;
overflow: hidden;
text-overflow: ellipsis;
}
div:hover {
    overflow: visible;
}
```

## Text – Alignment

```
<div>
    Ahmed sabry  css
</div>
```

```
div {
background-color: coral;
text-align: center;
}
img {
    vertical-align: middle;//
}
```

## The vertical-align Property

### **vertical-align: baseline (default):**

An  image with a default alignment.

### **vertical-align: text-top:**

An  image with a text-top alignment.

### **vertical-align: text-bottom:**

An  image with a text-bottom alignment.

### **vertical-align: sub:**

An  image with a sub alignment.

### **vertical-align: sup:**

An  image with a super alignment.

## احترام النص لمساحة الحاوية

### white-space

#### white-space: nowrap;

الغرض منه جعل الكلام غير مرتبط بحاوية ال div بمعنى لو النص طويلاً هيطبع  
بها ال div

this is div in my web sit welecom in your seit-- this is div in my web sit welecom in your seit

this is div in my web sit welecom in your seit-- this is div in my web sit welecom in your seit

```
div{  
    background-color: burlywood;  
    padding: 10px;  
    border: 1px solid #140101;  
    width: 300px;  
    white-space: nowrap;  
    overflow: hidden;  
}
```

لو عملنا overflow: hidden يخفى الكلام الزائد عن حجم ال div

this is div in my web sit welecom in your seit-- t

لو عملنا text-overflow: clip مش هيحصل حاجة لأنها الوضع الافتراضي  
"reset"

text-overflow: clip;

overflow : هيحط نقط مكان الكلام المخفى بال ellipsis

text-overflow: ellipsis;

this is div in my web sit welecom in your se...

## text-overflow: ellipsis

هتنفع في حالة ان النص معمول في وسم ال div نفسه ولكن لو محطوط في برجراف داخل div مش هيطبق عليه ومش هيظهر النقط

## overflow: visible

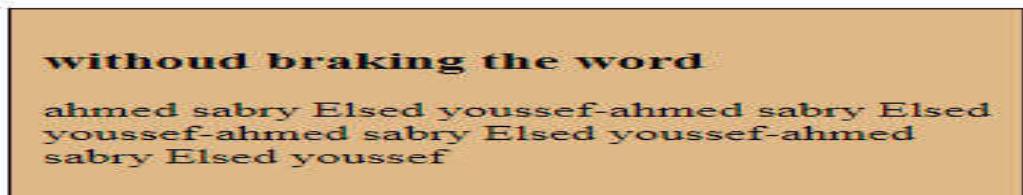
```
div:hover{ overflow: visible; }
```

وجيـت وقـفت بالـماوس عـلـى الـDiv هـتلـغـي خـاصـيـة overflow: hidden والـكلـام هـيـاـخـد رـاحـتـه و هـيـطـلـع خـارـج الـDiv  
خدـ بالـك

```
<div>
    <h3> without breaking the word </h3>
    <p> نـص مـكتـوب </p>
</div>
```

```
div{
    background-color: burlywood;
    padding: 10px;
    border: 1px solid #140101;
    width: 300px;
}
```

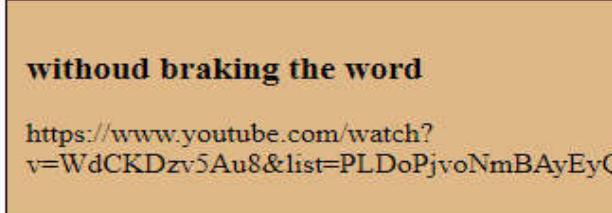
الـنتـيـجة : فـي حـالـة كـتـبـنا فـي الـبـرـجـارـف نـص هـيـفـضـل دـاخـل الـDiv وـمـش هـيـطـلـع خـارـج الـDiv



**without breaking the word**

ahmed sabry Elsed youssef-ahmed sabry Elsed youssef-ahmed  
youssef-ahmed sabry Elsed youssef-ahmed  
sabry Elsed youssef

فـي حـالـة لو وـضـعـنـا لـيـنـك او كـلـمـة طـوـيـلة هـتـطـلـع خـارـج الـDiv



**without breaking the word**

<https://www.youtube.com/watch?v=WdCKDzv5Au8&list=PLDoPjvoNmBAyEyQaHOHO1HJtmSgGt07VC&index=9>

# دليل المبرمج CSS3-part.1

في المثال التالي عندنا div وداخله لينك

```
<div>
    <h3> without breaking the word </h3>
    <p> لينك او نص طويل بدون مسافات بين الحروف </p>
</div>
```

المطلوب : كسر اللينك الموجود في عنصر آخر داخل ال div وهو البرجرا ف -  
عشان ميطلعش خارج ال div .

نستعمل div الذي يحتوى على البرجرا ف : word-wrap: break-word;

```
div{
    background-color: burlywood;
    padding: 10px;
    border: 1px solid #140101;
    width: 300px;
    word-wrap: break-word;
}
```

this div

<https://www.youtube.com/watch?v=WdCKDzv5Au8&list=PLDoPjvoNmBAyEyQaHOHO1HJtmSgGt07VC&index=9>

اجبار النص على عدم الخروج من الحاوية (منع الامتداء)

word-break: break-all;

```
div {
width: 100px;
word-break: break-all;
}
```

<https://www.youtube.com/watch?v=Hphs2vBjmQs&list=PLDoF>

<https://www.youtube.com/watch?v=Hphs2vBjmQs&list=PLDoPjvoNmBA>

## white-space: normal

تستخدم هذه الخاصية لعمل بريك "قطع النص" بعد وضع مسافة

```
<div>
  <h1>this div </h1>
  <p>https://www.youtube.com/watch?v=WdCKDzv5Au8&list=PLDoPjvoNmBAyEyQaHOHO1HJtmSgGt07VC&index=9</p>
</div>
```

this div

<https://www.youtube.com/watch?v=WdCKDzv5Au8&list=PLDoPjvoNmBAyEyQaHOHO1HJtmSgGt07VC&index=9>

white-space: normal;

## Inheritance التوريث

```
<div>
  <h3>Learn Css</h3>
  <p>Welcome To The Website</p>
</div>
```

```
div {
  text-align: center;
  padding: 20px;
  background-color: #EEE;
  font-size: 20px;
  border: 2px solid rgb(193, 7, 106);
}
div p {
  border: 2px solid;
  border-color: inherit;
  padding: inherit;
}
```

```
مكتسب من div
div { index2.html:8
   text-align: center;
   padding: ▶ 20px;
   background-color: □ #EEE;
   font-size: 20px;
   border: ▶ 2px solid
    ■rgb(193, 7, 106);
}
```

+

العناصر الموجودة بداخل الـ Div سوف ترث الخصائص الخاصة بالحاوية ولكن لا ترث كل العناصر

العناصر المخفية في الصورة هي عناصر لا تورث والعناصر الظاهرة مثل- text-align سوف يتم توريثها ، مثال العنصر h3 لها حجم خط افتراضي لذلك لا يمكن للعنصر h3 ان يرث الخصيـة font size ، بينما يمكن لعنصر p ان يرث من الحاوية الخاصـية font size

**ملاحظة** ....

الخاصـية border فى الوضع الطبيعي لا تورث للابناء ، والهـوامش لا تورث ولكن يمكن ان نرث اللون والهـوامش من عنصر الاب عن طريق اضافة السمة inherit

```
div p {
  border: 2px solid;
  border-color: inherit;
  padding: inherit;
}
```

والخصـائص المشطوبة مثل font size لا تورث فى حالة كان العنصر User agent اي له خـصـائص افتراضـية الخاصـص المشطوبة يمكن ان تكون عـناصر كانت من المفترض ان تورث من الـ اب الى الـ ابناء ولكن تم التعـديل على الخـاصـية على الـ ابن فـلم تـورث

```
div {text-align: center;}
```

```
div p {text-align: left;}
```

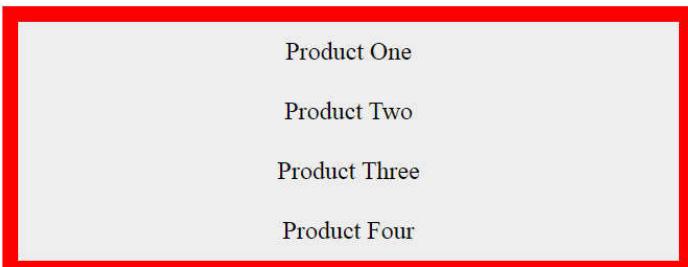
فى المـثال السـابق سيـتم تـوريـث الخـاصـية text-align: center للـعنـصر p ولكن تم التعـديل على الخـاصـية على الـ ابن "left" لذلك لن تـورـث

## المستوى الثاني

### Float

```
<div class="father">
  <div>Product One</div>
  <div>Product Two</div>
  <div>Product Three</div>
  <div>Product Four</div>
</div>
<div class="clear"></div>
<p>This Is P For Testing Float</p>
```

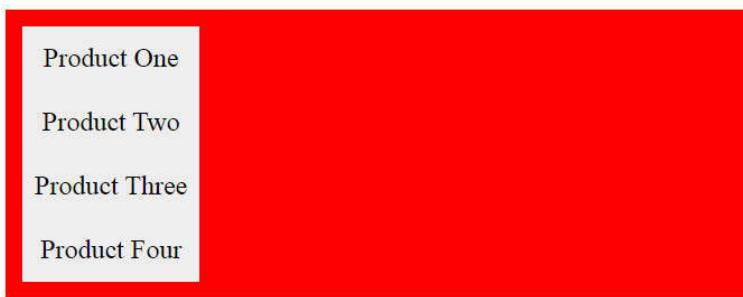
```
.father {
  background-color: red;
  padding: 10px;
}
.father div {
  padding-top: 10px;
  padding-bottom: 10px;
  text-align: center;
  background-color: #eee;
  /* width: 25%; float: left; */
}
```



This Is P For Testing Float

جعل عرض الابناء ٢٥%

```
.father div {
  /* width: 25%; float:; */ }
```



This Is P For Testing Float

لاحظ عنصر الاب اتسع في الطول لكي يحمل عناصر الابناء  
اعطاء الخاصية `float left` س يجعل الابناء يتربون في نفس الصف وليس  
العمود

Product One      Product Two      Product Three      Product Four

This Is P For Testing Float

العناصر أصبحت تطفو فوق عنصر الاب وترتب في صف واحد ولكن عنصر  
الاب ليس له ارتفاع لذلك لم يغطي الاطار الاحمر الابناء

يمكن حل هذه المشكلة عن طريق استخدام الخاصية `overflow: hidden` لجعل  
عنصر الاب يشمل الابناء بداخله ويغطيهم في الارتفاع

```
.father {  
background-color: red;  
padding: 10px;  
overflow: hidden;  
}
```

Product   Product   Product   Product  
One      Two      Three      Four

## مثال آخر لتجربة التعويم

العناصر تطفو بجانب بعضها في نفس السطر

```
<div class="father">
  <div>Product One</div>
  <div>Product Two</div>
  <div>Product Three</div>
  <div>Product Four</div>
</div>
<p>This Is P For Testing Float</p>
```

```
.father {
  background-color: red;
  padding: 10px;
}
.father div {
  padding-top: 10px;
  padding-bottom: 10px;
  text-align: center;
  background-color: #eee;
  width: 12%;
  float: left;
}
```

Product One   Product Two   Product Three   Product Four   This Is P For Testing Float

## لو قمنا باعطاء عنصر الاب الخاصية overflow: hidden

سيقوم بطرد العناصر التي ليست بداخل عنصر الاب الى الاسفل بعد الغصر ، بمعنى سيحتفظ الاب بالابناء فقط

```
.father {
  background-color: red;
  padding: 10px;
  overflow: hidden;
}
```

```
<div class="father">
  <div>Product One</div>
  <div>Product Two</div>
  <div>Product Three</div>
  <div>Product Four</div>
</div>
<p> يتم طرد هذا العنصر من الحاوية لانه ليس ابن للاب </p>
```

```
Product Product Product Product
One     Two    Three   Four
```

This Is P For Testing Float

الحل العملى والافضل : لعدم طفو عنصر البرجraf بجانب الابناء  
هو انشاء عنصر "فاصل" clear p class clear بعد اخر عنصر فى الابناء وقبل  
العنصر الذى لا نريده ان يطفو (p) نقوم باعطاءه خاصية clear: both

```
<div class="father">
  <div>Product One</div>
  <div>Product Two</div>
  <div>Product Three</div>
  <div>Product Four</div>
  <p class="clear"></p>
</div>
<p>This Is P For Testing Float</p>
```

```
.father {
  background-color: red;
  padding: 10px;
}
.father div {
  padding-top: 10px;
  padding-bottom: 10px;
  text-align: center;
  background-color: #eee;
  width: 12%;
  float: left;
}
.clear {
  clear: both;
}
```

## شرح خاصية float بطريقة عملية

العنصر الذى يأخذ القيمة float يجعل العنصر الى ياتى بعده يظهر خلفه ويمتد بجانبه فى العرض

```
<header>head</header>
<aside>aside</aside>
<main>main</main>
<footer>footer</footer>
```

```
aside{
    float: left;
}
aside{
    background-color: burlywood;
    width: 30%;
    height: 600px;
}
main{
    background-color: rgb(12, 119, 39);
    width: 31%;
    height: 600px;
}
footer, header{
    background-color: dimgray;
    padding: 30px 0;
}
```

العنصر aside اخذ اخاصة الطفو لذلك العنصر الذى ياتى بعده main (باللون الاخضر) ياتى خلفه ويمتد بجانبه اذا كان العرض اكبر من العنصر الذى يأخذ float



## مثال اخر

```
<header>head</header>
<aside>aside</aside>
<main>main</main>
<footer>footer</footer>
```

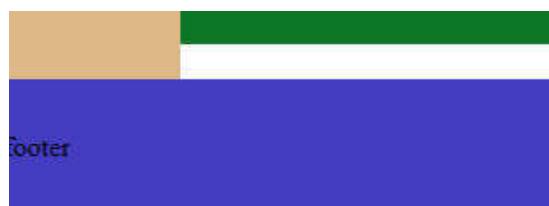
```
aside{/*1*/
    background-color: burlywood;
    width: 10%;
    height: 520px;
    float: left;}
main{/*2*/
    background-color: rgb(12, 119, 39);
    width: 31%;
    height: 500px;}
footer, header{
    background-color: dimgray;
    padding: 30px 0;}
footer{background-color: rgb(68, 61, 192);}
```

العناصر تترتيب ترتيب منطقى لذلك ذيل الصفحة footer ياتى بعد الـ main (اللون الاخضر) ولكن العنصر aside ارتفاعه اعلى من عنصر الـ main لذلك يظهر فوق عنصر الـ footer



يمكن ان نجعل الـ footer ياتى بعد العنصر الذى يأخذ الخاصية float عن طريق اعطاء الـ footer الخاصية clear: both لكي لا يجعل العنصر يطفو فوقه

```
footer{
    background-color: rgb(68, 61, 192);
    clear: both;}
```



## تجربة على الـ float

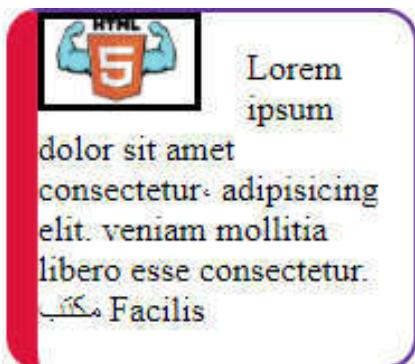
```
<section>
  
<p>Lorem ipsum dolor sit amet consectetur, adipisicing elit.
  veniam mollitia libero esse consectetur. Facilis provident
</p>
</section>
```

```
section{
  border: 2px solid rebeccapurple;
  border-left: 14px solid crimson;
  border-radius: 15px; /* overflow: hidden; */
}
img{
  margin-right: 20px;
  border: solid 3px;
  width: 40%;
  float: left;}
```

الصورة تأخذ حجم .٤٠ % من حجم الحاوية : نلاحظ ان ارتفاع  
الحاوية تحترم النص ولا تحترم الصورة .



عند تصغير مساحة العرض ، نجد ان النص يلتف حول الصورة من  
جهة الشمال لأن الصورة تأخذ الخاصية **float:left**



عند تكبير حجم الصورة نجد ان الحاوية لا تحترم ارتفاع الصورة لأن الصورة أصبحت فوق الحاوية لأنها تأخذ خاصية الطفو



*Lorem ipsum dolor sit amet consectetur adipisicing elit. veniam mollitia libero esse consectetur. مكتب Facilis*

يمكن ان نجعل الحاوية تحترم ارتفاع الصورة (التي تطفو ) عن طريق استخدام الخاصية **overflow: hidden** للحاوية

```
section{  
    overflow: hidden;  
}
```

نجد ان حجم الحاوية أصبح مرتبط بحجم الصورة

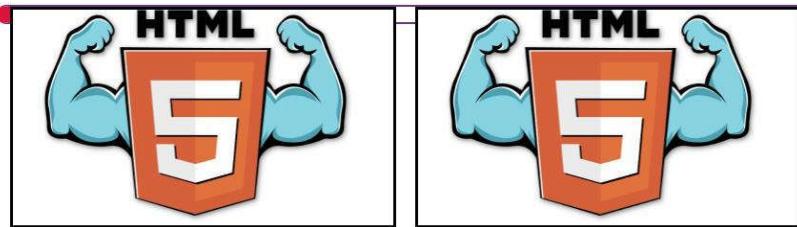


*Lorem ipsum dolor sit amet consectetur adipisicing elit. veniam mollitia libero esse consectetur. مكتب Facilis*

## استعمال clear: both

```
  
  
  
<p> </p>  
</section>
```

```
section{  
    border: 2px solid rebeccapurple;  
    border-left: 14px solid crimson;  
    border-radius: 15px;}  
  
img{  
margin-right: 20px;  
border: solid 3px;  
width: 40%;  
float: left;  
}
```

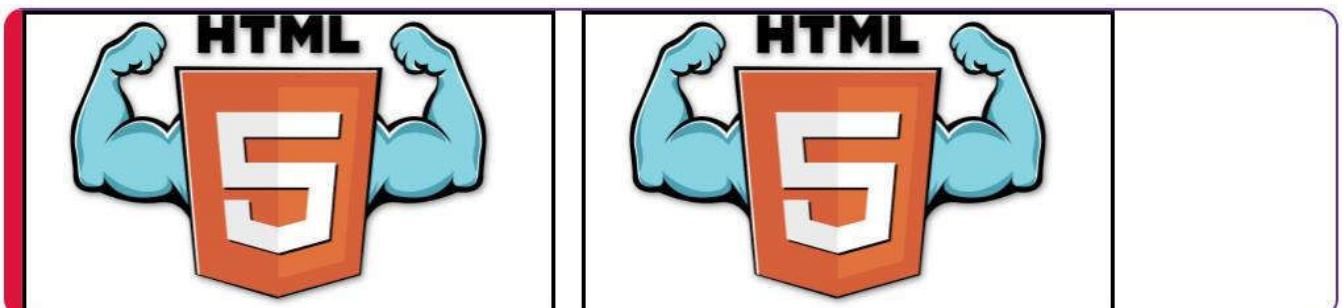


الحاوية لا تحترم تأثير الـ **Float**

لحل هذه المشكلة يمكن ان نضع عنصر بعد الصور ونعطيه خاصية  
**clear: both**

```
p{  
    clear: both;  
}
```

النتيجة



## حل اخر لـ اعطاء الحاوية الخاصية overflow: hidden

```
section{  
    overflow: hidden;  
}
```

## مثال اخر : بديل overflow: hidden

```
<section>  
    <h1>heding_1</h1>  
    <h1>heding_2</h1>  
</section>
```

```
section{  
    border: 2px solid rebeccapurple;  
    border-left: 14px solid crimson;  
    border-radius: 15px;  
}  
h1{  
    float: left;  
    background-color: burlywood;  
    padding: 10px;  
    border: 1px solid;  
    margin: 2px;  
}  
section::after{  
    content: "";  
    clear: both;  
    display: table;  
}
```

heding\_1 heding\_2

## شرح زائد

### استخدام ال float لـ div وداخل ال Span

this is test div float left

[view 100](#)

```
<div class="left">
    this is test div float left <span>view 100</span>
    </div>
```

```
.left{
    width: 500px;
    background-color: mediumblue;
    color: mintcream;
    padding: 10px;
    margin: 10px;
    float: left; }

span{
    color: rgb(238, 238, 48);
    float: right;
}
```

## ملحوظة

ال text-align يغير اتجاه النص كله ولا يغير جزء من النص الموجود مثلاً داخل " view 100" span

## float

### الاستخدام الفعلى للfloat

#### Simple float example

##### Float

Float ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam dolor, eu lacinia lorem placerat vulputate. Duis felis orci, pulvinar id metus ut, rutrum luctus orci. Cras porttitor imperdierit nunc, at ultricies tellus laoreet sit amet.

Sed auctor cursus massa at porta. Integer ligula ipsum, tristique sit amet orci vel, viverra egestas ligula. Curabitur vehicula tellus neque, ac ornare ex malesuada et. In vitae convallis lacus. Aliquam erat volutpat. Suspendisse ac imperdierit turpis. Aenean finibus sollicitudin eros pharetra congue. Duis ornare egestas augue ut luctus. Proin blandit quam nec lacus varius commodo et a urna. Ut id ornare felis, eget fermentum sapien.

Nam vulputate diam nec tempor bibendum. Donec luctus augue eget malesuada ultrices. Phasellus turpis est, posuere sit amet dapibus ut, facilisis sed est. Nam id risus quis ante semper consectetur eget aliquam lorem. Vivamus tristique elit dolor, sed pretium metus suscipit vel. Mauris ultricies lectus sed lobortis finibus. Vivamus eu urna eget velit cursus viverra quis vestibulum sem. Aliquam tincidunt eget purus in interdum. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

#### Simple float example

##### Float

Float ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam dolor, eu lacinia lorem placerat vulputate. Duis felis orci, pulvinar id metus ut, rutrum luctus orci. Cras porttitor imperdierit nunc, at ultricies tellus laoreet sit amet.

Sed auctor cursus massa at porta. Integer ligula ipsum, tristique sit amet orci vel, viverra egestas ligula. Curabitur vehicula tellus neque, ac ornare ex malesuada et. In vitae convallis lacus. Aliquam erat volutpat. Suspendisse ac imperdierit turpis. Aenean finibus sollicitudin eros pharetra congue. Duis ornare egestas augue ut luctus. Proin blandit quam nec lacus varius commodo et a urna. Ut id ornare felis, eget fermentum sapien.

Nam vulputate diam nec tempor bibendum. Donec luctus augue eget malesuada ultrices. Phasellus turpis est, posuere sit amet dapibus ut, facilisis sed est. Nam id risus quis ante semper consectetur eget aliquam lorem. Vivamus tristique elit dolor, sed pretium metus suscipit vel. Mauris ultricies lectus sed lobortis finibus. Vivamus eu urna eget velit cursus viverra quis vestibulum sem. Aliquam tincidunt eget purus in interdum. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

يخلى العنصر الذى معمول ليه "Div" float يأخذ مكانه الاول ومن ثم يلتئف النص حول ال div

```
<body>
    <h1>Simple float example</h1>
    <div class="box">Float</div>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam dolor, e. </p>
    <p>Sed auctor , eget fermentum sapien.</p>
    <p>Nam vulputate diam nec tempor bibendum. Donec luctus augue eget malesuada ults.</p>
</body>
```

```
.box {
    background-color: rgb(207,232,220);
    width: 150px;
    height: 100px;
    border-radius: 5px;
    float: left;
    padding: 1em;
```

```
margin-right: 15px;  
}
```

## Simple float example

Float left

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam dolor, eu lacinia lorem placerat vulputate.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam dolor, eu lacinia lorem placerat vulputate. Duis felis orci, pulvinar id metus ut, rutrum luctus orci. Cras porttitor imperdiet nunc, at ultricies tellus laoreet sit amet.

Sed auctor cursus massa at porta. Integer ligula ipsum, tristique sit amet orci vel, viverra egestas ligula. Curabitur vehicula tellus neque, ac ornare ex malesuada et. In vitae convallis lacus. Aliquam erat volutpat. Suspendisse ac imperdiet turpis. Aenean finibus sollicitudin eros pharetra congue. Duis ornare egestas augue ut luctus. Proin blandit quam nec lacinia varius commodo et a urna. Ut id ornare felis, eget fermentum sapien.

Nam vulputate diam nec tempor bibendum. Donec luctus augue eget malesuada ultrices. Phasellus turpis est, posuere sit amet dapibus ut, facilisis sed est. Nam id risus quis ante semper consectetur eget aliquam lorem. Vivamus tristique elit dolor, sed pretium metus suscipit vel. Mauris ultricies lectus sed lobortis finibus. Vivamus eu urna eget velit cursus viverra quis vestibulum sem. Aliquam tincidunt eget purus in interdum. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

## في هذا المثال : عملنا للبرجraf الاول كلاس

```
.special {  
background-color: rgb(79,185,227);  
padding: 10px;  
color: #fff;  
}
```

و عملنا للبوكس margin من جميع الاتجاهات عشان نعمل ازاحة ١٥ بيكسل من الشمال و يظهر البوكس فوق الخلفية الزرقاء للبرجاف

```
.box {  
background-color: rgb(207,232,220);  
width: 150px;  
height: 150px;  
border-radius: 5px;  
padding: 1em;  
float: left;  
margin: 15px;  
}
```

يمكن عمل نفس النتيجة السابقة بدون اعطاء البرجراF كلاس مثال:

```
<body>
  <h1>Simple float example</h1>

  <div class="wrapper">
    <div class="box">Float</div>

    <p>
      Lorem ipsum dolor sit amet,
      consectetur adipiscing elit.
      Nulla luctus aliquam dolor,
      eu lacinia lorem placerat vulputate. </p>
    </div>

    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam dolor, eu lacinia lorem placerat vulputate. </p>
    <p>Sed auctor cursus massa at porta. Integer ligula ipsum, tristique sit amet orci vel, viverra egestas ligula. Curabitur vehicula tellus neque, ac ornare ex malesuada et.</p>
    <p>Nam vulputate diam nec tempor bibendum. Donec luctus augue eget malesuada ultrices. Phasellus turpis est, posuere si</p>
  </body>
```

```
body {
  width: 90%;
  max-width: 900px;
  margin: 0 auto;
  font: .9em/1.2 Arial, Helvetica, sans-serif
}

.wrapper {
  background-color: rgb(79,185,227);
  padding: 10px;
  color: #fff;
}

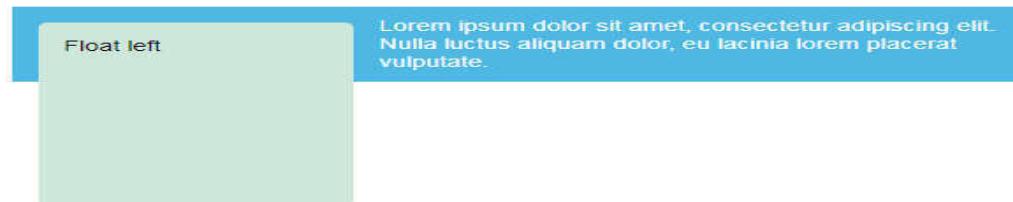
.box {
  float: left;
  margin: 15px;

  width: 150px;
  height: 150px;

  border-radius: 5px;
  background-color: rgb(207,232,220);
  padding: 1em;
}
```

عملنا div شامل وعملنا له كلاس احتوى بداخله على البوكس والبرجراف واعطينا خواص هذه الخواص

## Simple float example



Float left

**Placeholder text:** Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam dolor, eu lacinia lorem placerat vulputate.

Nam ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam dolor, eu lacinia lorem placerat vulputate. Duis felis orci, pulvinar id metus ut, rutrum luctus orci. Cras porttitor imperdiet nunc, at ultricies tellus laoreet sit amet.

Sed auctor cursus massa at porta. Integer ligula ipsum, tristique sit amet orci vel, viverra egestas ligula. Curabitur vehicula tellus neque, ac ornare ex malesuada et. In vitae convallis lacus. Aliquam erat volutpat. Suspendisse ac imperdiet turpis. Aenean finibus sollicitudin eros pharetra congue. Duis ornare egestas augue ut luctus. Proin blandit quam nec lacus varius commodo et a urna. Ut id ornare felis, eget fermentum sapien.

Nam vulputate diam nec tempor bibendum. Donec luctus augue eget malesuada ultrices. Phasellus turpis est, posuere sit amet dapibus ut, facilisis sed est. Nam id risus quis ante semper consectetur eget aliquam lorem. Vivamus tristique elit dolor, sed pretium metus suscipit vel. Mauris ultricies lectus sed lobortis finibus. Vivamus eu urna eget velit cursus viverra quis vestibulum sem. Aliquam tincidunt eget purus in interdum. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

```
<body>
    <h1>Simple float example</h1>
    <div class="box">Float left</div>
    <p class="special">
        Lorem ipsum dolor sit amet,
        consectetur adipiscing elit.
        Nulla luctus aliquam dolor,
    </p>
    <p class="cleared">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nula luctus aliquam dolor, eu lacinia lorem placerat vulputate. Duis felis orci, pulvinar id metus ut, rutrum luctus orci. Cras porttito</p>
    <p>Sed auctor cursus massa at porta. Integer ligula ipsum, tristique sit amet orci vel, viverra egestas ligula. Curabitur vehicula tellus neque, ac ornare ex malesuada et. In vitae convallis lacus. Aliquam erat
    </p>
    <p>Nam vulputate diam nec tempor bibendum. Donec luctus augue eget malesuada ultrices. Phasellus turpis est, posuere sit amet dapibus ut, facilisis sed est. Nam id risus quis ante semper consectetur eget aliquam lorem. Vivamus tristique elit dolor, sed pretium metus suscipit vel. Mauris ultricies lectus sed lobortis finibus. Vivamus eu urna eget velit cursus viverra quis vestibulum sem. Aliquam tincidunt eget purus in interdum. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.
    </p>
</body>
```

.cleared { clear: left; }

عملنا كلاس للبرجراف وعملنا تنظيف لاي فلوت ناحية الشمال وبالتالي الكلام نزل من اول السطر وساب مكان الالتفاف فاضي

## Overflow

```
<div>
    <p>ths is pragraf </p>
    <p>ths is pragraf </p>
</div>
```

```
div{
    background: burlywood;
}
p{
    float: left;
}
```

فى المثال السابق اللى هيحصل ان البرجراف هيطبع من ال **div** والاثنين هيطلعوا جنب بعض فى سطر واحد بدون خلفية ال **div**

بمعنى ان البرجراف لم يصبح يتاثر بال **div**

```
*{ margin: 0px;
padding: 0px;}
div{ height: 5px;
margin: 0px;
padding: 0px;
background: burlywood;}
div p{ float: left;}
```

ths is pragrafths is pragraf

## overflow: hidden

هيطلع الاثنين فى سطر واحد ولكن هيخدوا خلفية ال **div** بدون مشاكل

Mastering

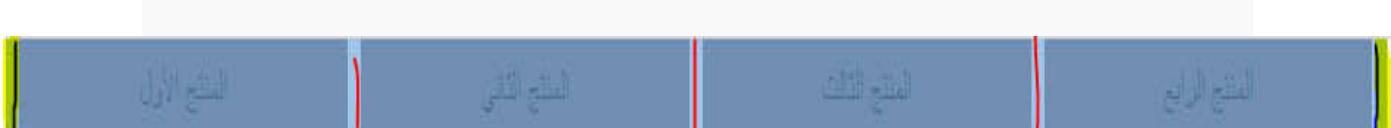
## تقسيم المساحة على العناصر

```
<div>Product One</div>
<div>Product Two</div>
<div>Product Three</div>
<div>Product Four</div>
```

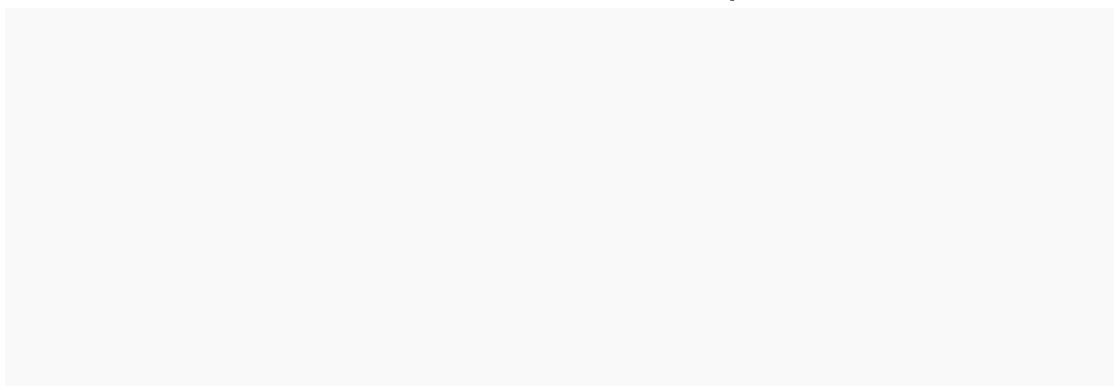
```
body {
    margin: 0;
}
div {
    padding-top: 10px;
    padding-bottom: 10px;
    text-align: center;
    background-color: rgb(113, 98, 98);
    width: 23.75%; /* 100/5=23.75      4*23.75=    (95%) */
    float: left;
    margin-left: 1%; /* 5*1=   (5%) */
}
```

لدينا مساحة الصفحة ١٠٠ % ولدينا ٤ حاويات ونريد اعطاء هامش من الشمال قبل كل حاوية بمقدار ١% اي ٤% بالإضافة الى هامش ١% بعد اخر حاوية اي اجمالي ٥% فتصبح المساحة المتاحة المتبقية ٩٥% الان نريد تقسيم مساحات الحاويات بالتساوي لذلك نقسم ٩٥% على عدد الحاويات (٤) ستصبح النتيجة

٢٣.٧٥%



الهامش الداخلية ٣ اما العدد الكلى للهامش ٥ ( يوجد هامش قبل العنصر الاول وهامش بعد العنصر الاخير )



## الخاصية calc

يمكن تقسيم المساحة بشكل الى عن طريق استعمال هذه الدالة

```
div {  
    padding-top: 10px;  
    padding-bottom: 10px;  
    text-align: center;  
    background-color: rgb(113, 98, 98);  
    width: calc(95%/4) ;  
    float: left;  
    margin-left: 1%;  
}
```

استخدام هامش بوحدة قياس البكسل

```
<div>Product One</div>  
<div>Product Two</div>  
<div>Product Three</div>  
<div>Product Four</div>
```

```
body {  
    margin: 0;  
}  
div {  
    padding-top: 10px;  
    padding-bottom: 10px;  
    text-align: center;  
    background-color: #eee;  
    float: left;  
    width: calc((100% - 50px) / 4);  
    margin-left: 10px;  
}  
/*  
Content Width Is: 100%  
1Spaces =10px  
5 Spaces 10px * 5 => 50px From Content Width  
100% - 50px Available Space  
4 Elements The Width Is: (100% - 50px) / 4  
*/
```

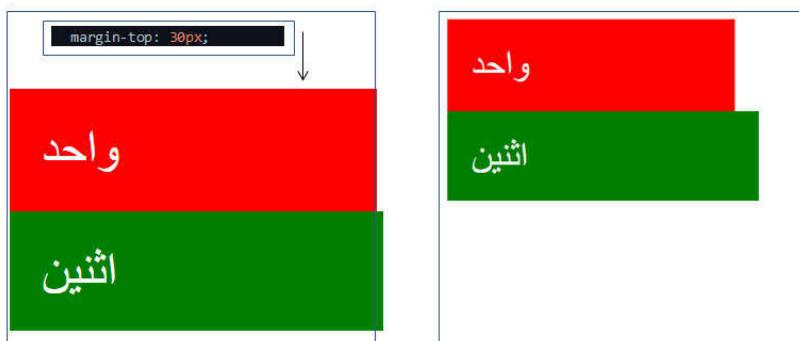
## مواضع العناصر

### position: static;

الوضع الطبيعي ان العنصر اذا تم ازاحته من اعلى سيأخذ معه العنصر التالي الى اسفل

```
<div class="one">One</div>
<div class="two">Two</div>
```

```
div {
    color: white;
    width: 100px;
    padding: 10px;
}
.one {
    margin-top: 30px;
    background-color: red;
}
.two {
    background-color: green;
    width: 110px;
}
```



## position relative

نقدر من خلاله نحرك العنصر بس المكان اللي هيترزح منه هيسيب مكانه مساحة فاضية زى ال margin

```
position: relative;  
top: 9px;
```



## مثال عملى

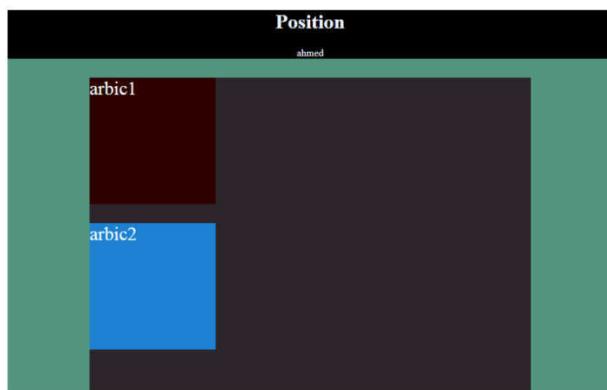
```
<div class="one">One</div>  
<div class="two">Two</div>  
  
div {  
    color: white;  
    width: 100px;  
    padding: 10px;  
}  
.one {  
    top: 30px;  
    background-color: red;  
    position: relative;  
}  
.two {  
    background-color: green;  
    width: 110px;  
}
```

فى هذا المثال الحاوية الثانية ستظل مكانها اما الحاوية الاولى div class 1 يأخذ الموضع relative بالإضافة الى top 30px فانه سيتحرك بالنسبة لموضعه من اعلى بمقدار 30px ويترك مكانه فارغا – وادا تحرك الى ان يصل لموضع Div2 فإنه سيظهر فوقه .



## مثال تطبيقي ..

لدينا مربع رمادي " wrapper " بداخلة ٢ div



coopy;vopyriht by Eidewiny

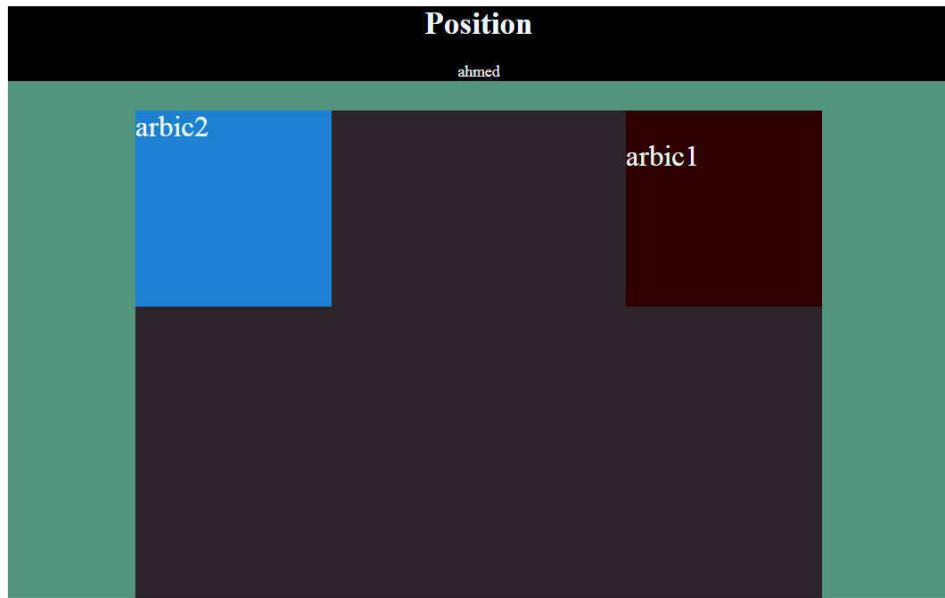
```
<body>
<div class="contanier">
    <header>
        <h1>Position  </h1>
        <p>ahmed</p>
    </header>

    <div class="wrapper">
        <div class="arbic1">
            <p>arbic1</p>
        </div>

        <div class="arbic2">
            <p>arbic2</p>
        </div>
    </div>
</div>
<footer> <p>coopy;vopyriht by Eidewiny</p></footer>
</body>
```

```
.contanier{  
    width: 960px;  
    margin: 0 auto;  
    background-color: rgba(12, 105, 74, 0.705);}  
header {  
    background-color: black;  
    color: aliceblue;  
    text-align: center;  
    margin-bottom: 0px;}  
.wrapper{  
width: 700px;  
height: 500px;  
margin: 10px auto;  
background: #2c252a;  
}  
.arbic1{  
position: relative;  
top: 20px;  
width: 200px;  
height: 200px;  
  
font-size: 30px;  
color: azure;  
background: #2e0101;  
  
margin-bottom: 10px;}  
.arbic2{  
  
width: 200px;  
height: 200px;  
  
font-size: 30px;  
color: azure;  
background: #1e80d1;}  
  
footer{  
background: #ac6a6a;  
height: 50px;  
text-align: center;  
font-size: 30px;}
```

لو عملنا لل div1 float: right - div2 هتطلع الى المكان الاصلى لل div1

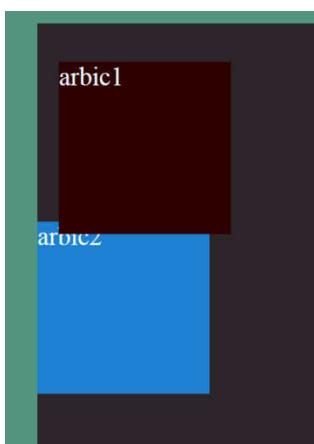


coopy;vopyriht by Eidewiny

دلوقتى هنشيل ال float ونحط لل div1 الاتى

```
position: relative;  
top: 0px;  
left: 499px;
```

النتيجة العنصر الاول هيت حريكه ولكن  
هيفضل حاجز مساحته والعنصر الثانى لن يتغير مكانه



## position: absolute

وظيفة الخاصية : تجعل العناصر التي تاتى بعده لا تحترم مساحتها وتترتب خلفه (تاخذ مكانه) فى نفس موضوعه ولاكن خلفه

```
<div class="one">One</div>
<div class="two">Two</div>
```

```
div {
color: white;
width: 100px;
padding: 10px;
}
.one {
background-color: red;
position: absolute;
}
.two {
background-color: green;
width: 110px;
}
```

النتيجة :



بالنسبة لل定位 و التمركز (الاتجاهات) يتحرك بالنسبة للعنصر الاب في حالة عدم وجود حاوية تغطيه سيأخذ عنصر `body` اب له

```
<div class="one">One</div>
<p>text</p>
<div class="two">Two</div>
```

```
div {
color: white;
width: 100px;
padding: 10px; }
.one {
background-color: red;
position: absolute;
bottom: 0px;
right: 0px; }
.two {
background-color: green;
width: 110px; }
```

العنصر ترك مكانه فاصل والعنصر الذى ياتى بعده(p) اخذ مكانه

ملحوظة هامة لما نعمل لعنصر position: absolute يجب ان نضع لعنصر الاب الذى يحتويه position: relative

السبب عشان العنصر يتحرك top او left بمقدار القيمة التى تضعها بالنسبة للاب "div" الذى يشمله " wrapper" وليس بالنسبة لل body

```
<div class="two">Two</div>
<div class="wrapper">
    <div class="one">One</div>
</div>
```

```
div {
color: white;
width: 100px;
padding: 10px;
}
.one {
background-color: red;
position: absolute;
right: 0px;
top: 0px;
}
.two {
background-color: green;
width: 110px;
}
.wrapper{
background-color: rgb(44, 35, 35);
width: 200px;
height: 100px;
position: relative;
}
```

## ملحوظة

الاب " wrapper" الذى يغطى العنصر اذا لم يأخذ القيمة relative  
العنصر الذى بداخله صاحب القيمة position: absolute; هيتمركز موضوع  
بالنسبة لل body وبناءا عليه لما نعمل سكرول للصفحة مكانة هيتغير خالص فى  
كل مرة هنعمل فيها زوم للصفحة

## position: fixed

### لعمل عنصر ثابت في الصفحة

١- بتثبت العنصر في مكان ثابت داخل الصفحة بمعنى (اي كلام داخل الصفحة اذا تعارض موقعه في نفس موضع ذلك العنصر ) هيظهر تحت منه و هيتحرك تحت منه ومهما طلعت scroll او نزلت هيظهر معاك في نفس مكانه

٢- مكانة غير مربوط بال div الى موجود بداخله يعني ممكن يطلع برا ال div لانه غير مندرج في الحاوية

```
<p>Lorem ipsum dolor sit, amet consectetur adipisicing elit. Eligendi  
veniam, inventore officia, accusantium soluta dolores debitis ducimus  
necessitatibus dolor, at est fuga et ex. Totam soluta sit quia corporis  
</p>  
<div class="one">One</div>
```

```
body{  
    height: 9000px;  
}  
div {  
color: black;  
width: 100px;  
padding: 10px;  
}  
.one{  
background-color: red;  
position: fixed;  
top: 20px;  
right: 20px;  
}
```

## position: sticky

وظيفته عند الوصول الى العنصر بواسطة scroll : يتم تثبيت العنصر كأنه Fixed

```
<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Hic sequi,  
at accusantium vitae officia, nobis quam omnis blanditiis est, eaque  
quod ratione voluptatibus. Ab architecto voluptates aperiam et quis  
amet!</p>  
<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Hic sequi,  
at accusantium vitae officia, nobis quam omnis blanditiis est, eaque  
quod ratione voluptatibus. Ab architecto voluptates aperiam et quis  
amet!</p>  
<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Hic sequi,  
at accusantium vitae officia, nobis quam omnis blanditiis est, eaque  
quod ratione voluptatibus. Ab architecto voluptates aperiam et quis  
amet!</p>  
<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Hic sequi,  
at accusantium vitae officia, nobis quam omnis blanditiis est, eaque  
quod ratione voluptatibus. Ab architecto voluptates aperiam et quis  
amet!</p>  
<div class="sticky">Add_Cart</div>
```

```
body{  
    height: 9000px;  
}  
div{  
background-color: brown;  
color: black;  
width: 100px;  
padding: 10px;  
}  
.sticky {  
position: sticky;  
top: 0;  
}
```

## Z-Index

خاصية تهتم بترتيب الابعاد (البعد الاول – الثاني – الثالث ) نستطيع من خلال هذه الخاصية نستطيع ترتيب اولوية الطبقات فوق بعض

```
<div class="one">One</div>
<div class="two">Two</div>
<div class="three">Three</div>
```

```
div {
    color: white;
    width: 100px;
    padding: 10px;
    text-align: center;
    position: absolute;
}
.one {
    background-color: red;
    left: 20px;
    top: 20px;
    z-index: 3;
}
.two {
    background-color: green;
    left: 15px;
    top: 15px;
    z-index: 2;
}
.three {
    background-color: blue;
    z-index: 1;
}
```

## عداد الفقرات counter-increment

```
counter-increment: test_counter;
```

```
<div class="parent">
    <div class="one" data-text="Something">
        Lorem ipsum dolor sit amet consectetur adipisicing elit. Laborum
        labore culpa dolores, autem voluptatum atque,
        in dolore asperiores facilis libero dolorum magni iure doloribus
        quae pariatur! Similique ipsum nobis quos.
    </div>
</div>
<div class="parent">
    <div class="one">
        Lorem ipsum dolor sit amet consectetur adipisicing elit. Laborum
        labore culpa dolores, autem voluptatum atque,
        in dolore asperiores facilis libero dolorum magni iure doloribus
        quae pariatur! Similique ipsum nobis quos.
    </div>
</div>
```

```
.parent {
    margin-bottom: 20px;
    counter-increment: test_counter;
}
.one {
    background-color: #eee;
    padding: 10px;
    width: 600px;
    float: left;
    position: relative;
}
.one::after {
    content: counter(test_counter);
    position: absolute;
    background-color: red;
    color: white;
    width: 20px;
    height: 20px;
    text-align: center;
    font-weight: bold;
    right: -10px;
    top: 50%;
    margin-top: -10px;
}
```

يمكن التحكم بالـ **hover** عن طريق الـ **before** والـ **after**

```
.one::after {  
    content: counter(test_counter);  
    position: absolute;  
    background-color: red;  
    color: white;  
    width: 20px;  
    height: 20px;  
    text-align: center;  
    font-weight: bold;  
    right: -10px;  
    top: 50%;  
    margin-top: -10px;  
    display: none; /*hidden*/  
}  
  
.one:hover::after{  
    display: block; /*show*/  
}
```

## الـ box

### box-shadow

```
div {  
border: 10px solid #f50606;  
width: 200px;  
height:100px ;  
margin:20px auto  
    box-shadow: 10px 10px 8px #888888;  
    box-shadow: 5px 5px 5px 9px #-----;  
يمكن عمل اكثرب من شادوو } ;
```

القيمة الاولى للتحريك شمال ولمين و القيمة الثانية فوق وتحت  
الثالثة لتحديد حدة او نعومة ال blur و الرابعة لحجم انتشار ال

### Box Sizing

```
<section>  
    <div class="left">left</div>  
    <div class="right">right</div>  
</section>
```

هذا هو الوضع الافتراضي : box-sizing: content-box;

```
*{  
    box-sizing: content-box;  }  
section{  
width: 940px;  
margin: 20px auto;  
background: #eeeeee;  
overflow: hidden; }  
.left{  
    float: left;  
    width: 340px;  
    background-color: lightcoral;  
color: lightcyan;  
padding: 1px 1px 1px 1px;  
border: 1px solid #27010157; }  
.right{  
    float: right;  
    width: 600px;  
    background-color: rgb(12, 4, 56);  
color: lightcyan; }
```

ال border و ال padding هيازروا على عرض ال div لأن مساحتهم يتزيد  
على العرض الاصلي

فى حالة ان قيمة **box-sizing: content-box;** = **box-sizing**



الحل : مبيحسبش ال **padding** و **border**: **border-box**

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

هيسمح بوضع حدود او عمل **padding** من غير ما يزود على المساحة  
الأصلية وبالتالي الشكل مش هيلعب معايا

```
*{  
    box-sizing: border-box;  
}  
section{  
    width: 940px;  
    margin: 20px auto;  
    background: #eeeeee;  
    overflow: hidden;  
}  
.left{  
    float: left;  
    width: 20%;  
    background-color: lightcoral;  
color: lightcyan;  
border: 4px solid #27010157;  
}  
.right{  
    float: right;  
    width: 80%;  
    background-color: rgb(12, 4, 56);  
color: lightcyan;  
}
```



## Transition

### transition-duration

التحكم في وقت حركة الانتقال  
بمعنى (الانتقال من حالة الى حالة في مدة زمنية)

```
<div class="one">One</div>

.one {
    background-color: rgb(167, 7, 122);
    width: 200px;
    height: 100px;
    transition-duration: 0.5s;
}
.one:hover {width: 280px;}
/*
duration مدة الانتقال
1- 1s = ثانية
2-0.5 = نصف ثانية
2- 1000ms = ثانية
*/
```

### transition-delay

تأخير تشغيل حركة الانتقال (تشغيل الانتقال بعد كام ثانية)  
الوضع الافتراضي بعد صفر ثانية

```
.one {
    background-color: rgb(167, 7, 122);
    width: 200px;
    height: 100px;
    transition-duration: 0.5s;
    transition-delay: 1s;
}
```

يمكن تطبيق حركة الانتقال على خاصية محددة

```
.one {
    background-color: rgb(167, 7, 122);
    width: 200px;
    height: 100px;
    transition-duration: 0.5s;
    transition-delay: 0.3s;
    transition-property: width; }
```

لتطبيق على كل الخصائص

```
transition-property: all;
```

## transition-timing

للتحكم فى كريف السرعة : الوضع الافتراضى **ease**

```
transition-timing-function: ease;
```

```
linear= سرعة منتظم  
ease= slow fast slow  
ease-in= بدء بطيئ ثم سرعة منتظم  
ease-out= سرعة منتظم ونهاية بطيئة  
ease-in-out= بدء بطيئ ونهاية بطيئ
```

طريقة اختزال مجموعة اسطر في سطر واحد

يستقبل الخاصية التى نريد التطبيق عليها  
: يستقبل مدة التأخير **transition-duration**

```
transition: width 1s ;
```

باقي الخصائص تأخذ الوضع الافتراضى

**function: ease**

**transition-delay:0s**

التحكم فى تأخير كل خاصية على حدى

**transition-duration**

```
.one {  
background-color: rgb(167, 7, 122);  
width: 200px;  
height: 100px;  
transition: width 1s , height 9s;  
}  
.one:hover {  
width: 280px;  
height: 400px;  
}
```

## طريقة أخرى

```
.one {  
background-color: rgb(167, 7, 122);  
width: 200px;  
height: 100px;  
transition: width , height ;  
transition-duration: 1s , 6s;  
/* first property after 1s; */  
/* second property after 6s; */  
}  
.one:hover {  
/*first property*/ width: 280px;  
/*second property*/height: 300px;  
}
```

## تطبيق وقت محدد بالتساوي لخصائصتين

```
.one {  
background-color: rgb(167, 7, 122);  
width: 200px;  
height: 100px;  
transition: width , height ;  
transition-duration: 3s /* apply time with tow property*/  
}  
.one:hover {  
width: 280px;  
height: 300px;  
}
```

## shorthand

```
transition: all , 0.1s 3s ease ;  
/*1-property 2-duration 3-delay 4-timing-function */
```

## انمیشن البعد الثانی

### Transform 2D

```
img:hover{  
    transform: translate(x,y);  
    transform: translate(0px, -10px);
```

القيمة الاولى x: للتحريك اتجاه الليمين او الشمال (يقبل موجب وسالب)

القيمة الثانية y : للتحريك اعلى او اسفل (يقبل موجب وسالب)

للحكم في وقت الحركة

```
img{  
    transition: all .3s ease-in-out;  
}
```

## ال Transform Scale 2D

```
<div class="container">
    <div class="scale">
        this is 2D TRansform scale()
    </div>
</div>
```

```
.container{
    width: 100px;
    background-color: khaki;
    margin: 10px auto;
    padding: 10px;
    border: 1px solid #7c0662;
    height: 50px;
}
.scale{
    width: 100px;
    background-color: rgb(30, 56, 3);
    color: lemonchiffon;
    text-align: center;
    height: 50px; }
.scale:hover{
    -webkit-transform:scale(1.1,1.1);
    transform: scale(1.1,1.1);
}
```

**transform: scale(2,2);**

بمعنى ضعف مساحة العرض والقيمة الثانية تعنى ضعف مساحة الطول

**transform: scale(.5,.5);**

بمعنى نص الطول ونص العرض

**transform: none "reset"**

لعمل عرض فقط للصورة

لاحظ ان القيمة واحد : تعنى نفس الحجم والقيمة ٢ تعنى الضعف

```
img:hover{
    transform: scale(2,1) (الطول 1,العرض 2 ) {
```

## Transform Rotate

X محور طولى : من ٠ الى ٣٦٠ محور (دوران كامل اتجاه اليمين) ويمكن استعمال ارقام سالبة للدوران من الشمال  
٧ محور راسى : ٠ الى ٣٦٠ (دوران كامل اتجاه اليمين) ويمكن استعمال ارقام سالبة للدوران من الشمال

```
div:hover{  
    transform: rotatey(180deg);  
}
```



```

```

```
img{  
    transition: all.1s ease-in-out;  
}  
img:hover{  
    transform: rotatex(40deg);  
}
```



```
transform: rotatey(180deg);
```



# دليل المبرمج CSS3-part.1

```
transform: rotateZ(180deg);
```



```
ربع دوران ;  
نص دوران ;  
دوران الا ربع ;  
دوران كامل ;  
عدد ٢ دوران ;  
*/
```

## استعمال خاصية الدوران مع خاصية التكبير

```
transform: scale(1.3,1.3) rotate(5deg);
```

## مثال تطبيقي

```
<div class="div_imge">  
      
</div>
```

```
.div_imge{  
    border: 1px solid red;  
    overflow: hidden;  
    width: 200px;  
    height: 200px;  
}  
img{  
    transition: 0.4s ease-in-out;  
    opacity: 0.6;  
}  
.div_imge:hover img{  
    transform: scale(1.3,1.3) rotate(5deg);  
    opacity: 1;  
}
```

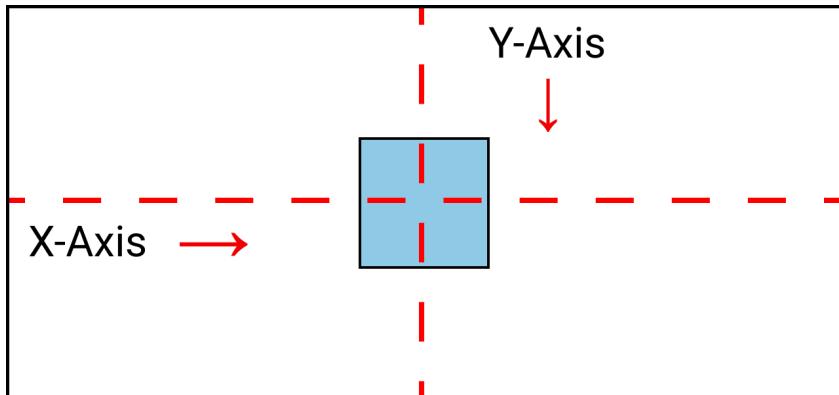
## Transform - Origin

<https://developer.mozilla.org/en-US/docs/Web/CSS/transform-origin>

### تحديد البؤرة "مركز" التحريك

```
<div> <span class="Origin"> </div>
```

```
/*
  transform-origin
  Syntax
  Default Values => 50% 50% 0
  2D Transform => transform-origin(X, Y)
  3D Transform => transform-origin(X, Y, Z)
  X-Axis
  - CSS Unit [px, em, rem]
  - %
  - Keyword
    -- Left = 0%   -- Center = 50%   -- Right = 100%   Y-Axis
  - CSS Unit [px, em, rem]
  - %
  - Keyword
    -- Top = 0%    -- Center = 50%      -- Bottom = 100%
*/
div {
  margin: 200px auto;
  width: 200px;
  height: 200px;
  background-color: rgb(255 0 0 / 39%);
  color: white;
  display: flex;
  justify-content: center;
  align-items: center;
  font-weight: bold;
  font-size: 30px;
  /* transform-origin: 50% 50%; */
  transform-origin: left bottom;
  transform: rotate(0deg);
}
.Origin {
  position: absolute;
  width: 20px;
  height: 20px;
  background-color: red;
  border-radius: 50%;
  left: 0%;
  top: 100%;
  transform: translate(-50%, -50%);
}
```



داخل الحاوية قمنا بعمل span على شكل دائرة موقعها في الجنوب اقصى الشمال لعنصر الاب (Div) ، لكن نرى المركز على شكل دائرة (للتجريب فقط)

وضعنا بورة لعنصر الاب مركز للتحريك

ففى الاسفل اقصى الشمال عن طريق النزول من اعلى بمقدار ١٠٠ ومن الشمال صفر.

لو قمنا بعمل فحص العنصر في المتصفح ، ثم قمنا بتغيير قيمة الاستدارة سنجد الدوران يتم من المركز الذى قمنا بتحديده

```
transform: rotate(0deg);
```



# دليل المبرمج CSS3-part.1

جعل المركز فى الاعلى اقصى الليمين

```
div {  
    transform-origin: right top;  
}
```

كلاس الـ Span للتوضيح فقط

```
.Origin {  
left: 100%;  
top: 0%;  
}
```



اتجاه فوق اقصى الشمال

```
div {  
    transform-origin: left top;  
}
```

```
.Origin {  
top: 0%;  
}
```



# دليل المبتدئ CSS3-part.1

```
div {  
    transform-origin: right Bottom;  
}  
.Origin {  
left: 100%;  
top: 100%;  
}
```



```
div {  
    transform-origin: left Bottom;  
}  
  
.Origin {  
left: 0%;  
top: 100%;  
}
```



## تجربة مفصلة لمين وشمال

```
div {  
    transform-origin: left top;  
    transform: rotatey(0deg);  
}
```

```
.Origin {  
left: 0%;  
top: 0%;  
}
```

## مفصلة من فوق لتحت

```
div {  
    transform-origin: left top;  
  
    transform: rotatey(0deg);  
}
```

```
.Origin {  
left: 0%;  
top: 0%;  
}
```

الدرجة 90deg تعنى درجة قائمة : عند الوصول الى هذه الدرجة يحتفى العنصر  
تجربة ظهور المربع من اسفل الى اعلى

```
div {  
margin: 200px auto;  
width: 200px;  
height: 200px;  
background-color: rgb(255 0 0 / 39%);  
color: white;  
display: flex;  
justify-content: center;  
align-items: center;  
font-weight: bold;  
font-size: 30px;  
/* transform-origin: 50% 50%; */  
transform-origin: left bottom;  
transform: rotatex(90deg);  
}  
.Origin {  
position: absolute;  
width: 20px;  
height: 20px;  
background-color: red;  
border-radius: 50%;  
left: 0%;  
top: 100%;  
transform: translate(-50% , -50%);  
}
```

## انمیشن 3D

### Transform 3D

عن طريق هذه الخاصية يمكن التحرير خلف وامام

لتفعيل خاصية بعد الثالث نكتب فى عنصر الاب Perspective

الخاصية Perspective تستقبل مقدار بعد الثالث Z

```
اب العنصر body {  
perspective: 600px;  
}
```

```
img:hover{  
    transform: translate3d(0px,0px,20px)  
}
```

القيمة الاولى (x) للاتجاه لمين وشمال و القيمة الثانية (y)  
لاتجاهات فوق وتحت والقيمة الثالثة (z) للبعد التقرير تباعد scale

يمكن التعويض عن الكود السابق بالكود التالي

```
img:hover{  
    transform: translatez(40px)  
}
```

transform: translatez(100px);

translatey و translatex و scale لعمل translatez

## مثال (١)

```
<body>
    
</body>
```

```
body{perspective: 800px;}
img{transition: all.1s ease-in-out;
margin: 400px 100px 900px 1400px ;
transition: all .5s ease-in-out;}
img:hover{transform: translate3d(0,0px, 100px);}
}
```

## مثال (٢)

```
<div>
    
</div>

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

```
div{
    margin-top: 50px;
    margin-bottom: 100px;
    display: inline-block;
}
div:first-of-type{ /* selector first div */
    perspective: 600px; /* المنظور - بعد الصورة عن عيني بمقدار */
}
div:nth-of-type(2){/* selector div num 2 */
    perspective: 400px;
}
div:last-of-type{/* selector last div */
    perspective: 300px;
}
img{width: 200px;
    transition: all .5s ease-in-out;
    transform: translateZ(100px); /* تأثير الصورة بمقدار */
}
/* img:hover{
    transform: translateZ(100px);
} */
```

# دليل المبرمج CSS3-part.1

في المثال التالي وضعنا البعد بين العنصر وبين العين ٦٠٠ بكسل للعنصر الاول ثم وضعنا قيمة رقمية لتقريب اى صورة بمقدار ١٠٠ بكسل من ال ٦٠٠ بكسل ملحوظة لو جعلنا التقريب ٦٠٠ بكسل الصورة هتختفى عشان هتبقى قدام عيني

## **perspective-origin**

تستخدم لوضع البؤرة على العنصر لتحديد النقطة التي سوف ترى من خلالها الصورة

### **الوضع الافتراضي**

```
perspective-origin center center; /*or 50% 50%*/
```

او

```
perspective-origin: 50% 50%;
```

يمكن التعديل على القيم عن طريق كتابة اسم الاتجاه

القيمة الاولى (شمال او لمين )

القيمة الثانية (فوق او تحت)

```
perspective-origin: 90% 20%;
```

## Transform scale3D rotate

لعمل زوم وتدوير

```
body {  
    perspective: 600px;  
}
```

```
img:hover{  
    transform: scale3d(3, 1, 1) rotatey(20deg);  
}
```



هيلف زى المراجيح من تحت ل فوق

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

```
body{  
    perspective: 500px;  
}  
div{  
  
    width: 300px;  
    height: 300px;  
    margin: 50px auto;  
    background-color: red;  
    transition: all .9s ease-in-out;  
}  
div:hover{  
    transform: rotatex(180deg);  
}
```

# دليل المبرمج CSS3-part.1

لعمل دوران للعنصر وتحديد " Anchor Point " اللي هيتحرك من خلالها العنصر او الصورة

```
<div>
  
</div>
```

```
body{
  perspective: 500px;
}
img{
  width: 100px;
}
div{
  width: 300px;
  height: 300px;
  margin:50px auto;

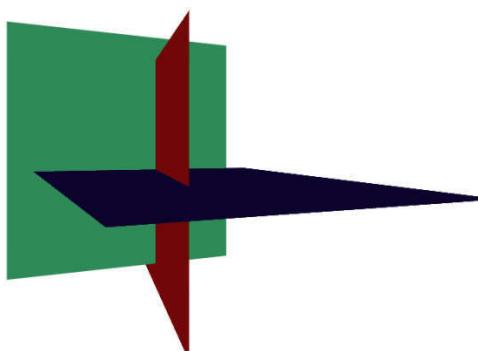
  transition: all .9s ease-in-out;
}
div:hover{
  transform: rotate(360deg);
  transform-origin: top left;
}
```



## لجعل العناصر المرتبطة في div واحد يتطبق عليها خاصية ال 3d

```
<div class="father">
    <div class="child1"></div>
    <div class="child2"></div>
</div>
```

```
body{
    perspective: 500px;
}
.father{
    width: 300px;
    height: 300px;
    margin: 50px auto;
    position: relative;
    background-color: seagreen;
    transition: all .5s ease-in-out;
    transform:rotatey(20deg);
    transform-style: preserve-3d; من بدل
}
.child1{
    width: 300px;
    height: 300px;
    position: absolute;
    left: 30px;
    top: 30px;
    background-color: rgb(112, 8, 8);
    transform: translateZ(50px) rotatey(45deg)
}
.child2{
    width: 300px;
    height: 300px;
    position: absolute;
    left: 30px;
    top: 30px;
    background-color: rgb(13, 4, 46);
    transform: translateZ(150px) rotatex(45deg)
}
```



# دليل المبرمج CSS3-part.1

## عمل كارت يلف الوجه يختفي ويظهر الظهر

```
<div class="father">
    <div class="front"> this is front </div>
    <div class="back"> this is back </div>
</div>
```

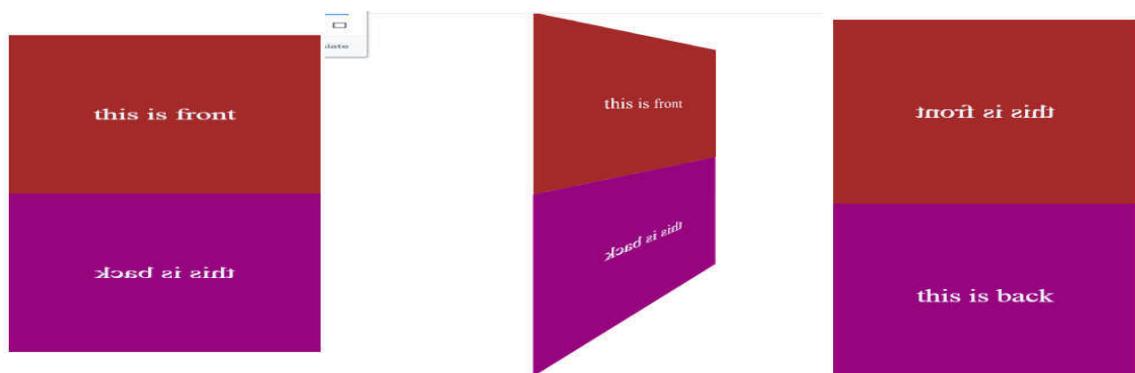
```
body{
    perspective: 500px; /* بعد عن العين*/
    .father{ /* خصائص الاب */
        width: 300px;
        height: 300px;
        margin: 50px auto;
        position: relative;
        transition: all .5s ease-in-out;
        transform-style: preserve-3d; }

    .father div/* لاي خصائص داخل div */ {
        position: absolute; /* نفهم ده السطر حقنا لو */
        left: 0;
        top: 0;
        width: 100%;
        height: 100%;
        line-height: 300px;
        color: aliceblue;
        font-size: 30px;
        text-align:center; }

    .front/* لواجهة خصائص */ {
        z-index:2;
        background-color: brown; }

    .back/* لظهر خصائص */ {
        z-index:1;
        background-color: rgb(151, 5, 127);
        transform: rotatey(180deg);
        /* display: none; */

    .father:hover{
        transform: rotatey(180deg)
    }
}
```



## Create Flip Card

```
<body>
<div class="card">
<!--front-->
    <div class="front">
        <header>
            
        </header>
        <h3>chat</h3>
        <p> this is face p</p>
        <span>hover to see more</span>
    </div>
<!--baack-->
    <div class="back">
        <ul>
            <li>facebook : ahmed sbry </li>
            <li>google : ahmed sbry </li>
            <li>youtup : ahmed sbry </li>
        </ul>
    </div>
</div>
```

```
body { perspective: 600px; }
.card { width: 300px;
height: 320px;
margin: 50px auto;
position: relative;
transform-style: preserve-3d;
transition: all 1s ease-in-out; }
.card > div {
position: absolute;
top: 0;
left: 0;
width: 100%;
height: 100%;
background-color: #eee;
border-radius: 10px;
border: 1px solid #08f;
text-align: center;
color: #da0d0d; }
.card .front header {background-color: #333;
width: 100%;
height: 80px;
border-radius: 10px 10px 0 0 ; }
```

```
.front img {  
    width: 100px;  
    height: 100px;  
    border-radius: 50%;  
    position: relative;  
    top: 30px;  
    border: 2px solid #ffffff;  
}  
.front {  
    text-align: center;  
    background-color: #ddd;  
    z-index: 4;  
    transform-origin: 50% 50%;  
    transform-style: preserve-3d;  
    backface-visibility: hidden  
}  
.card .front h3 {  
    margin-bottom: px;  
    line-height: 100px;  
    color: rgb(173, 31, 102);  
    font-size: 25px;  
    text-transform: uppercase;  
}  
.card .front p {  
    margin: 0 auto;  
    line-height:2;  
    width: 90%;  
}  
.card .front span {  
display: inline-block;  
margin-top: 5px;  
color: #08f;  
font-weight: bold;  
cursor: pointer;  
}  
.card .back {  
    z-index:3;  
    transform: rotateY(180deg)  
}  
.card:hover {  
    transform: rotateY(180deg)  
}
```

لو حذفنا السطر ده وعدلنا فى قيم الz هيبان

## Transform Translate

```
<div class="container">
    <div class="Translate">
        براحتك موقع اسألنى
    </div>
</div>
```

```
.Translate:hover{
    transform: translate(0px,10px);
}
```

القيمة الاولى راسى والقيمة الثانية عمودى



## لعمل ثلاث تأثيرات مختلفة

```
transform: translate(10px,10px) scale(.75) rotate(20deg);
```

لجعل الانتقال يأخذ ثانية

```
.Translate{
    width: 100px;
    background-color: rgb(30, 56, 3);
    color: lemonchiffon;
    text-align: center;
    height: 50px;
    transition: all 1s ease-in-out;
}
```

## Transform Skew

```
<div class="container">
    <div class="Skew">
        موقع اسألني براحتك
    </div>
</div>
```

```
.Skew{
    width: 100px;
    background-color: rgb(30, 56, 3);
    color: lemonchiffon;
    text-align: center;
    height: 50px;
    transition: all 1s ease-in-out;
}
.Skew:hover{
    transform: skew(10deg , 10deg);
}
```

- مثال آخر :

```
<h1>The transform Property</h1>

<h2>transform: rotate(20deg);</h2>
<div class="a">Hello World!</div>
<br>
<h2>transform: skewY(20deg);</h2>
<div class="b">Hello World!</div>
<br>
<h2>transform: scaleY(1.5);</h2>
<div class="c">Hello World!</div>
```

```
div.a { width: 150px;
    height: 80px;
    background-color: yellow;
    -ms-transform: rotate(20deg); /* IE 9 */
    transform: rotate(20deg); }
div.b { width: 150px;
    height: 80px;
    background-color: yellow;
    -ms-transform: skewY(20deg); /* IE 9 */
    transform: skewY(20deg); }
div.c { width: 150px;
    height: 80px;
    background-color: yellow;
    -ms-transform: scaleY(1.5); /* IE 9 */
    transform: scaleY(1.5); }
```

## Animation - KeyFrames

تغییر لون تدريجی من لون الى لون

```
<div>Animation</div>
```

```
div {
    width: 100px;
    height: 100px;
    background-color: red;
    position: absolute;
    left: 50%;
    top: 50%;
    display: flex;
    justify-content: center;
    align-items: center;
    color: white;
    transform: translate(-50%, -50%);
    animation-name: change-color; /* اسم للتأثير */
    animation-duration: 2s; /* مدة لتنفيذ التأثير */
}
/* Code One */
@keyframes change-color {
    from {background-color: red;}
    to {background-color: blue; }
}
/* Code Two */
@keyframes change-color {
    0% {background-color: red;}
    100% { background-color: blue; }
}
/* Code Three */
@keyframes change-color {
    0% {background-color: red;}
    5% { background-color: blue; }
    80% {background-color: black;}
    100% {background-color: red;}
}
```

## عمل دائرة دوران

```
<div>s</div>
```

```
div{  
    /* position: absolute; */  
    left: 50%;  
    top: 50%;  
    width: 100px;  
    height: 100px;  
    background-color: aliceblue;  
    border-radius: 50% ;  
    border: 4px solid black;  
    text-align: center;  
    border-left-color: brown;  
    transition: all 3s ease-in-out;  
}  
div:hover{  
    transform: rotate(1turn);  
}
```

## طريقة اخرى

```
<div>s</div>
```

```
div{  
    left: 50%;  
    top: 50%;  
    width: 100px;  
    height: 100px;  
    background-color: aliceblue;  
    border-radius: 50% ;  
    border: 4px solid black;  
    text-align: center;  
    border-left-color: brown;  
    /* transition: all 3s ease-in-out; */  
    animation-name: write_name_spin;  
    animation-duration: 1s; /* لاتغير */  
}  
@keyframes write_name_spin{  
    0%{transform: rotate(0turn);}  
    100%{transform: rotate(1turn);}  
}
```

لجعل الانمیشن يتكرر الى ما لا نهاية

```
div{  
    animation-iteration-count: infinite;  
}
```

لجعل الانمیشن يتكرر مرتين على سبيل المثال

```
div{  
    animation-iteration-count: 2;  
}
```

للتحكم في كيف السرعة من بداية الحركة الى النهاية

```
animation-timing-function: linear;
```

انعکاس الحركة دوران ( بالعكس )

```
    animation-direction: reverse;
```

لفة لمين و لفة شمال

```
    animation-direction: alternate;
```

لفة شمال و لفة لمين

```
    animation-direction: alternate-reverse;
```

الوضع الافتراضي

```
    animation-direction: normal;
```

عدم التشغيل الا بعد مرور ثانية (تأخير)

```
    animation-delay: 1s;
```

## القيمة السالبة بتسرق من الوقت (تقديم وليس تأخير)

```
div{  
    /* position: absolute; */  
    left: 50%;  
    top: 50%;  
    width: 100px;  
    height: 100px;  
    background-color: aliceblue;  
    border-radius: 50% ;  
    border: 4px solid black;  
    text-align: center;  
    border-left-color: brown;  
    /* transition: all 3s ease-in-out; */  
    animation-name: write_name_spin;  
    animation-duration: 5s; /* وقت الحركة */  
    animation-iteration-count: 2;  
    animation-timing-function: linear;  
    animation-direction: normal;  
    animation-delay: -2s;  
  
}  
@keyframes write_name_spin{  
    0%{background-color: red;}  
    50%{background-color: blue;}  
    100%{background-color: black;}  
}
```

## الاحتفاظ باخر قيمة

```
        animation-fill-mode: forwards; /* color_black */
```

## الاحتفاظ باول قيمة

```
        /* انعكاس الحركة دوران (بالعكس)
```

```
        animation-fill-mode: both; /* color_red */
```

## ايقاف الانميشن عند الـ hover

```
div{  
    animation-play-state: running;  
}
```

```
div:hover{  
    animation-play-state: paused;  
}
```

## خصائص الصور

[https://www.w3schools.com/cssref/pr\\_background-position.asp](https://www.w3schools.com/cssref/pr_background-position.asp)

لوضع صورة داخل ال div

```
div{  
    background-image: url(a.jpg);  
}
```

للدخول على ملف موجود في نفس مسار ملف ال Html

```
background-image: url(img/a.PNG);
```

للدخول على صورة خارج المسار

```
background-image: url(..../a.PNG);
```

لجعل الصورة في المنتصف

```
background-position: center;
```

لجعل الصورة لا تتكرر مرتين

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

## لجعل الصورة ثابتة لا تتحرك والمحتوى فقط هو الذى يتحرك (مثل العلامة المائية)

```
<html>
<head>
<style>
body {
    background-image: url("img_tree.gif");
    background-repeat: no-repeat;
    background-attachment: fixed;
}
</style>

</head>
<body>
<h1>The background-attachment Property</h1>
<p>The background-
image is fixed. Try to scroll down the page.</p>
<p>The background-
image is fixed. Try to scroll down the page.</p>
<p>The background-
image is fixed. Try to scroll down the page.</p>
<p>If you do not see any scrollbars, try to resize the browser window. </p>
</body>
</html>
```

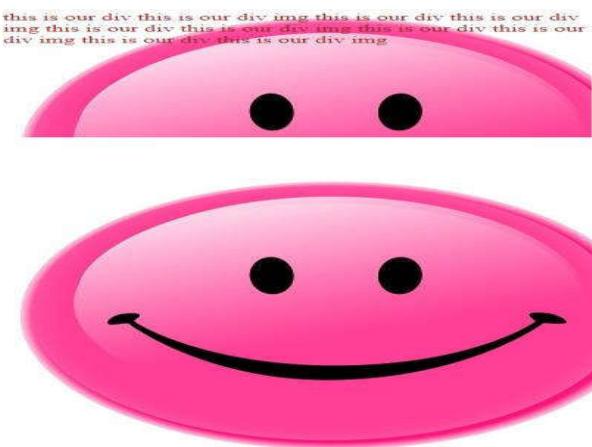
## مثال اخر

```
<div class="one"> </div>
<p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Velit
quam, asperiores libero iusto necessitatibus sequi, deleniti fugiat
magnam quo nam obcaecati assumenda impedit tenetur nesciunt mollitia id
laborum ullam enim!</p>
</body>
```

```
body{
    height: 1800px;
}
.one{
height: 300px;
/* background-image: url(http://wscreenwallpapers.com/wp-
content/uploads/2015/04/new-york-8-wallpaper-1080p-HD.jpg); */
background-image: url('a.PNG');
background-attachment: fixed;
background-repeat: no-repeat;
width:100%;
border: 2px solid black;
}
```

## حجم الصورة داخل ال div

ال الطبيعي ال div هيأخذ من الصورة على قد حجمه وباقى الصورة مش هيظهر  
فى حالة لو كبرنا ال div على قد حجم الصورة – الصورة تظهر كاملة



```
<body>
<div>
    this is our div    this is our div img
    this is our div    this is our div img
    this is our div    this is our div img
</div>

```

```
div{
    width:400px;
    height:200px;
    background: url(fa.jpg);
    margin-bottom: 60px;
}
```

فى حالة لو عايزين الصورة تظهر كاملة داخل ال div

هتلعب فى حجم الصورة ونخليها على قد حجم ال Div

```
div{ color: brown;
width:400px;
height:200px;
background: url(fa.jpg);
background-size: 400px 200px; الطول العرض 200px
margin-bottom: 60px;
}
```

ملحوظة : لو ال div كان اكبر من الصورة – الصورة هيتعلمنها تكرار

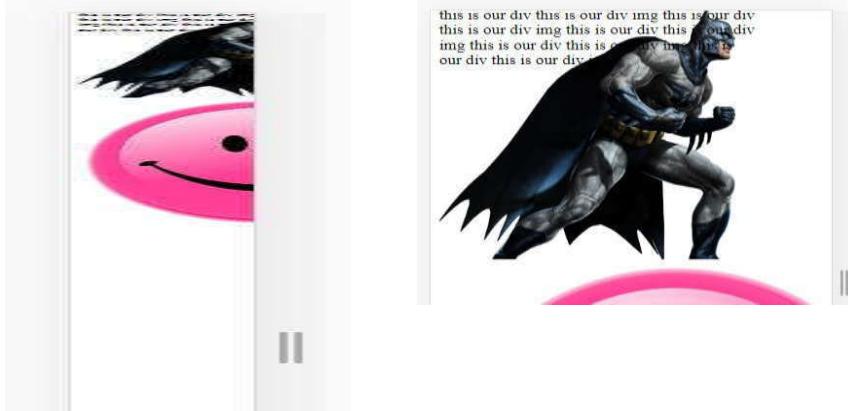
يمكن جعل الصورة تأخذ حجم ال Div تلقائي ((( افضل طريقة ))))

عن طريق background-size: 100% و نضع اي قيمة في الطول واي قيمة للعرض

```
div{  
width:600px;  
height:200px;  
background: url(fa.jpg);  
background-size: 100% 100%;  
}
```

خاصية ال background-size: cover;

لما بنصغر الشاشة بتحتفى جزء ولما بنكبر الشاشة بيبدا الجزء المختفى  
**بالظهور**



## css lazy load images

تحميل صور كسولة

```

```

```
<section>
    <h1>img1</h1>
    
</section>
<section>
    <h1>img2</h1>
    
</section>
<section>
    <h1>img3</h1>
    
</section>
<section>
    <h1>img4</h1>
    
</section>
<section>
    <h1>img5</h1>
    
</section>
```

```
img{
width: 90%;
height: 90%;
display: block;
margin-top: 200px auto;
border-radius: 15px;
}
section{
height: 900px;
margin-top: 1000px;
border: solid 4px;
width: 100%;
}
```

## Filters

- فلتر أبيض واسود

```

```

```
.class_img{  
    transition: 2s;  
    filter: grayscale(100%);  
}  
.class_img:hover{  
    filter: grayscale(0%);  
}
```

- فلتر الضباب

```
.class_img{  
    transition: 2s;  
    filter: blur(2px);  
}  
.class_img:hover{  
    filter: blur(0);  
}
```

- فلتر الشفافية

```
filter: opacity(0.20);  
filter: opacity(1);  
filter: opacity(80%);  
filter: opacity(50%);
```

```
.class_img{  
    transition: 2s;  
    filter: opacity(0.60);  
}  
.class_img:hover{  
    filter: opacity(1)
```

### فلتر الشادو

```
filter: drop-shadow(30px 10px 4px #4444dd);  
top-down  
right-left  
3- الانتشار  
3- color
```

## Global Variable

إنشاء متغير على مستوى جميع العناصر (root)

```
<div class="one">test</div>
```

```
/*  
:root{  
--write_name_Variable:blue;  
}  
    var(Variabe Name, قيمة بديلة فى حالة عدم وجود المتغير)  
/*/  
:root{  
--x:blue;  
}  
.one{  
background-color: var(--x);  
/* background-color: red; */  
}
```

مثال اخر

```
<div class="one">test</div>
```

```
:root{  
--x:10px;  
}  
.one{  
border: var(--x) solid red;  
padding: calc(20px + var(--x));  
}
```

## Local Variable

إنشاء متغير على مستوى كلاس

```
<div class="one">test</div>

.one{
--x:10px;
}
.one{
  border: var(--x) solid red;
  padding: calc(20px + var(--x));
}
```

درج الالوان

```
<div class="one"></div>
```

يمكن وضع اكثـر من لون

```
.one{
background-image: linear-gradient(to right , red ,blue);
width: 400px;
height: 600px;
}
/*
1- الاتجاه
2- اللون الاول
3- اللون الثاني
*/
```

يمكن تحديد النسبة المئوية لعرض كل لون

```
background-image: linear-gradient(to right , rgb(28, 185, 224) 20%
,rgb(183, 0, 255)80%);
```

## مثال اخر لتحديد النطاق

```
background-image: linear-gradient(to right , red 20%, blue 20%, blue  
40% , green 40%);
```

```
.one{  
background-image: linear-gradient(to right,  
red 20%,  
blue 20%,  
blue 40% ,  
green 40%  
);  
width: 400px;  
height: 10px;  
}
```

مساحة اللون الاحمر 20% ثم مساحة اللون الازرق من ٢٠ - الى ٤ ثم مساحة اللون الاخضر ابتداء من ٤ الى الآخر .

## Cursors text

### تغيير لون موشر الماوس

الخاصية **contenteditable** تسمح بالتعديل

```
<p contenteditable="true">  
Lorem ipsum dolor sit amet consectetur adipisicing elit.  
Repellendus est, inventore quam ducimus assumenda culpa  
veniam placeat voluptatibus reprehenderit nemo dignissimos,  
sapiente ea numquam in ab? Molestias tempora eveniet nisi?  
</p>
```

```
p {  
    caret-color: rgb(24, 4, 88);  
}
```

## عمل اسهم بالكود

```
<div class="one"></div>
```

```
.one
{
width: 20px;
height: 20px;
background: rgb(243, 222, 222);
border: solid 2px;
border-width: 8px;
border-color: red blue green black ;
}
```

## تكاملة

```
.one{
left: 20px;
width: 0px;
height: 0px;
background: rgb(243, 222, 222);
border: solid 2px;
border-width: 8px;
border-color: transparent black transparent transparent ;
}
```

## او بهذه الطريقة

```
.one{
    border: 1px solid red;
margin-left: 20px;
position: relative;
}
.one::before{
    position: absolute;
    content: "";
left: -20px;
width: 0px;
height: 0px;
background: rgb(243, 222, 222);
border: solid 2px;
border-width: 8px;
/* border-color: transparent #e91e63 transparent transparent ; */
border-color: transparent transparent transparent #e91e63 ;
}
```

الجزء الثالث

flex box & grid css

1- flex box



## الفرق بين استخدام الـ `float` والـ `flex`

```
<h3>Flex</h3>
<div class="flex">
  <div>One</div>
  <div>Two</div>
  <div>Three</div>
  <div>Four</div>
</div>
<hr />
<h3>Float</h3>
<div class="float">
  <div>One</div>
  <div>Two</div>
  <div>Three</div>
  <div>Four</div>
</div>
```

```
.flex { /*الاب*/
background-color: #f00;
background-color: rgb(207, 170, 170);
width: 600px;
padding: 20px;
margin: 20px auto;
display: flex; /*من اجل استخدام الخاصية*/
flex-direction: row; /* column */ /*row-reverse*/
}
.flex div {/*الابناء*/
background-color: #f00;
color: white;
width: 25%; /*عرض كل عنصر*/
text-align: center;
padding: 20px;
}
```

## flex-direction

لترتيب العناصر في صف او عمود

```
.flex { /*الاب*/
    display: flex; /*من اجل استخدام الخاصية*/
    flex-direction: row; /* عمود او صف - 1-column */
}
```

في حالة تغيير اتجاه الصفحة يأخذ الـ flex-direction نفس اتجاهه

```
<html lang="en">
```

جعل flex-direction يأخذ اتجاه مختلف لاتجاه العام للصفحة

بمعنى اخر الترتيب بالعكس من الاخر للاول – من الاول للآخر

```
flex-direction: row-reverse;
/*or*/
flex-direction: column-reverse;
```



ملحوظة : خاصية الـ `flex` تقوم بعمل `overwrite` للمساحات  
بمعنى لو وصعنا مساحة لعنصر الاب مساحة 100% ووضعنا  
اربع عناصر ابناء كل ابن 25% من مساحة الاب سيتم تقسيم  
المساحة بالتساوی ، ولاكن اذا اضفنا عنصر خامس سيتم تصغير  
مساحة الابناء الخامسة ليكون مجموع مساحتهم موزع بالتساوی

100/5

## الـ flex قلبه واسع يتسع كل الابناء

```
<h3>Flex</h3>

<div class="flex">
  <div>One</div>
  <div>Two</div>
  <div>Three</div>
  <div>four</div>
  <div>five</div>
</div>
```

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

.flex {
  border: 1px solid black;
  background-color: rgb(200, 158, 158);
  width: 100%;
  display: flex;
  flex-direction: row;
}

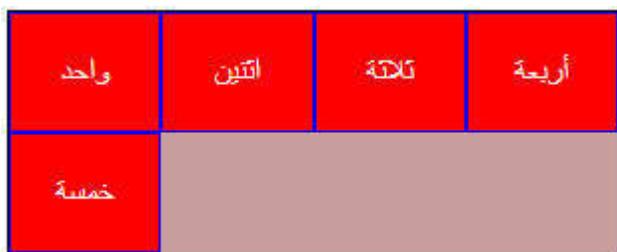
.flex div {
  border: 1px solid blue;
  background-color: #f00;
  color: white;
  width: 25%;
  text-align: center;
  padding: 20px;
}
```

## flex-wrap: wrap;

اذا كنا نرغب فى جعل الحاوية الاب لا تأخذ مساحة ابناء اكثرا من ١٠٠% بحيث نطرد العنصر الزائد

```
<h3>Flex</h3>
<div class="flex">
  <div>One</div>
  <div>Two</div>
  <div>Three</div>
  <div>four</div>
  <div>five</div>
</div>
```

```
.flex div {
  width: 25%; /*1 div =25%   total= (25*5= 125)   */
}
```



## justify-content

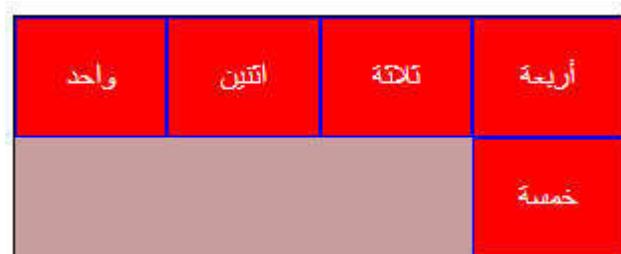
الوظيفة : التحكم فى اتجاه العنصر المطروح

الوضع الافتراضى flex-start

```
justify-content:flex-start ;
```

تغير اتجاه العنصر المطروح فقط

```
justify-content:flex-end ;
```



## توسيط العنصر المطرود

```
justify-content:center ;
```

### وضع مسافات داخلية بين الابناء

```
justify-content: space-between;
```

```
<h3>Flex</h3>
<div class="flex">
  <div>One</div>
  <div>Two</div>
  <div>Three</div>
  <div>four</div>
</div>
```

```
* {box-sizing: border-box;}
.flex {
  border: 1px solid black;
  background-color: rgb(200, 158, 158);
  display: flex;
  justify-content: space-between;
}
.flex div {
  border: 1px solid blue;
  background-color: #f00;
  color: white;
  width: calc(98%/4); /*=25*/
  text-align: center;
  padding: 20px;
}
```

### تحليل

فمنا بتقسيم مساحة ٩٨ على عدد العناصر في المساحة وتركتنا مساحة 2% من المساحة الكلية للهوا من المساحة الداخلية .

سيتم توزيع المساحة المتبقية 2% بالتساوي على العناصر

### وضع مساحة قبل وبعد جميع العناصر

```
justify-content: space-around;
```

## توسيط النص داخل العنصر (أفقيا - بشكل عرضي)

```
justify-content: center;
```

```
<div class="flex">Welcom</div>
```

```
.flex {border: 1px solid black;  
background-color: rgb(200, 158, 158);  
width: 300px;  
height: 300px;  
display: flex;  
justify-content: center;  
}
```

## Align Items

محاذة العناصر راسيا - الوضع الافتراضي **stretch**

```
<h1>Flex</h1>  
<div class="flex">  
  <div>One</div>  
  <div>Two</div>  
  <div>Three</div>  
</div>
```

```
* {box-sizing: border-box;}  
.flex {  
  border: 3px solid black;  
  background-color: rgb(43, 40, 40);  
  width: 600px;  
  height: 150px;  
  display: flex;  
  align-items: stretch;  
}  
.flex div {  
  border: 3px solid blue;  
  background-color: #f00;  
  color: white;  
  width: calc(100% / 3);  
  text-align: center;  
}
```



# دليل المبرمج CSS3-part.1

```
align-items: center;
```



```
align-items: flex-start;
```



جعل العناصر في نهاية عنصر الاب

```
align-items: flex-end;
```



## تجربة وضع عنصر غير متساوی في الطول

محاذاة العناصر راسيا - الوضع الافتراضي stretch

```
.flex {  
    align-items: stretch;  
}  
  
<div class="flex">  
    <div>One</div>  
    <div class="tow">Two</div>  
    <div>Three</div>  
    <div>four</div>  
</div>  
  
* {  
    box-sizing: border-box;  
}  
.flex {  
    border: 1px solid black;  
    background-color: rgb(200, 158, 158);  
    display: flex;  
    justify-content: space-between;  
}  
.flex div {  
    border: 1px solid blue;  
    background-color: #f00;  
    color: white;  
    width: calc(98%/4); /*=25*/  
    text-align: center;  
    padding: 20px;  
}  
.tow{  
    height: 200px;  
}
```

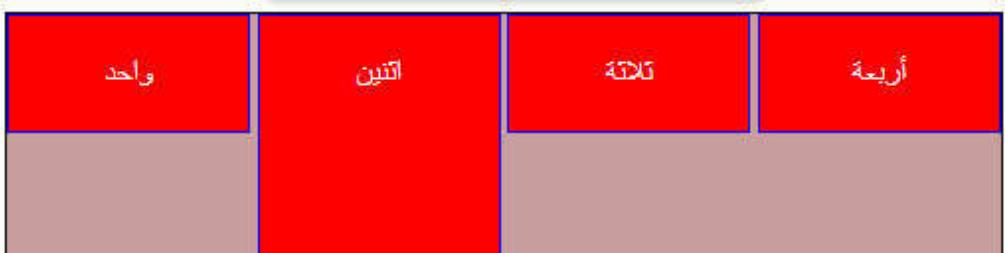
جميع العناصر تأخذ ارتفاع اكبر عنصر  
لو قنا باضافة السطر التالي لن يتم اي تعديل

```
.flex {  
    align-items: stretch;  
}
```

## الخاصية : flex-start

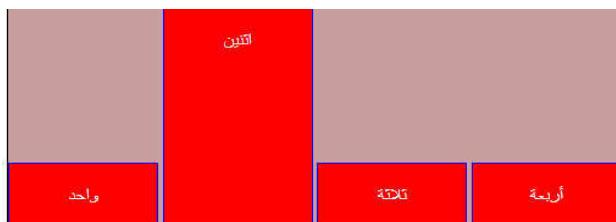
تجعل العنصر الذى يزيد ارتفاعه عن باقى العناصر يتمدد بمفرده الى الاسفل

```
align-items: flex-start;
```



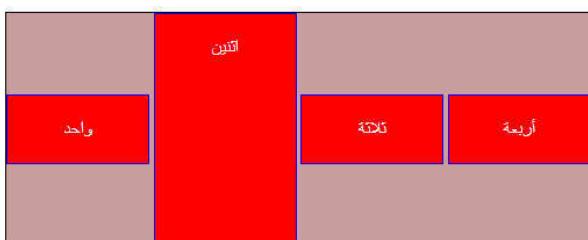
## الخاصية : flex-end

تجعل العنصر الذى يزيد ارتفاعه عن باقى العناصر يتمدد بمفرده الى الاعلى  
(باقى العناصر لن يتغير ارتفاعها )



## الخاصية : align-items: center

تجعل العنصر الذى يزيد ارتفاعه عن باقى العناصر يتمدد بمفرده الى الاعلى  
(باقى العناصر تتوسط فى منتصف اصول عنصر )



## توسيط النص داخل العنصر (راسيا - طولي)

```
<div class="flex">Welcom</div>
```

```
.flex {  
    border: 1px solid black;  
    background-color: rgb(200, 158, 158);  
    width: 300px;  
    height: 300px;  
    display: flex;  
    align-items: center;  
}
```

### لجعل النص في الاسفل (في الاتجاه الطولي)

```
align-items: flex-end;
```



### لجعل النص في الاعلى (في الاتجاه الطولي)

```
align-items: flex-start;
```

## توسيط النص بدون خاصية الـ Flex

```
<div class="no_flex">no_flex</div>
```

```
.no_flex{  
    border: 1px solid black;  
    background-color: rgb(200, 158, 158);  
    width: 300px;  
    height: 300px;  
    line-height: 300px; /* توسيط للنص طولي */  
    text-align: center; /* توسيط النص عرضي */  
}
```

## التحكم في عنصر واحد فقط

### Align Self

القيمة الافتراضية auto

```
<h1>Flex</h1>
<div class="flex">
  <div class="Div_one">One</div> /* تطبيق على عنصر واحد فقط */
  <div>Two</div>
  <div>Three</div>
</div>
```

```
* {box-sizing: border-box;}
.flex {
  border: 3px solid black;
  background-color: rgb(43, 40, 40);
  width: 600px;
  height: 150px;
  display: flex;
  align-items: stretch;
}
.flex div {
  border: 3px solid blue;
  background-color: #f00;
  color: white;
  width: calc(100% / 3);
  text-align: center;
}
.flex .Div_one{
background-color: aqua;
/* align-self: flex-end; */
align-self: center;
}
```

## Align Content

```
<h1>Flex</h1>
<div class="flex">
  <div>One</div>
  <div>Two</div>
  <div>Three</div>
  <div>Four</div>
  <div>Five</div>
</div>
```

```
* {
  box-sizing: border-box;
}
.flex {
  border: 1px solid blue;
  background-color: #eee;
  width: 600px;
  height: 400px;
  padding: 20px;
  margin: 20px auto;
  display: flex;
  flex-wrap: wrap;
  align-items: flex-end;
  /* align-content: center; */
}
.flex div {
  background-color: #f00;
  color: white;
  width: calc(100% / 3);
  text-align: center;
  padding: 20px;
}
```

بعد عمل `align-items: flex-end` ، وقمنا بعمل `flex-wrap` بعد طرد العناصر الزائدة سنجد مساحة فارغة ( هى عبارة عن طول الحاوية الاب – باقى مساحتها يتم توزيعه على شكل مساحات فارغة



حل مشكلة توزيع العناصر بالكامل في اتجاه محدد (في الاعلى – في الوسط – في الاسفل) بعد عمل (wrap) بدون ترك مساحات فارغة نقوم باستخدام align-items بدل من align-content

```
align-content:flex-end ;
```



يمكن استعمال الخاصية ( flex-start -flex-end - center )

"space-between"

```
align-content:space-between ;
```



## مساحة فاغة حول كل عنصر

التباعد بين كل زوج من العناصر المجاورة هو نفسه المساحة الفارغة قبل العنصر الأول وبعد العنصر الأخير تساوي نصف المسافة بين كل زوج من العناصر المجاورة

`justify-content: space-around;`

### توزيع المساحة الفارغة حول العناصر بالتساوي

التباعد بين كل زوج من العناصر المجاورة ، وحافة البداية - العنصر الأول ، وحافة النهاية والعنصر الأخير ، كلها متماثلة تماماً.

`justify-content: space-evenly;`

<https://developer.mozilla.org/en-US/docs/Web/CSS/justify-content>

## Flex Box Child

### Flex Box Grow

الوضع الافتراضي لهذه الخاصية (0)

```
<div class="flex">
    <div>1</div>
    <div>2</div>
    <div class="class_Div_3">3</div>
    <div>4</div>
</div>
```

```
* { box-sizing: border-box; }
.flex {
    background-color: #eee;
    width: 800px;
    padding: 20px;
    margin: 20px auto;
    display: inline-flex;
}
.flex div {
    background-color: #f00;
    color: white;
    width: 80px;
    text-align: center;
    padding: 20px;
```

```
margin-right: 5px;  
flex-grow: 1; /* جميع العناصر تتساوى في التمدد بنفس المقدار */  
.flex .class_Div_3{  
    flex-grow: 3;}
```

يمكن ان يتمدد العنصر المميز بكلس ، الذى يأخذ الخاصية " **flex-grow** " – ثلاثة اضعاف مساحة باقى الحاويات - فى حالة وجود مساحة فارغة



## Flex Box Shrink

الوضع الافتراضي لهذه الخاصية (1)

```
<div class="flex">  
    <div>1</div>  
    <div>2</div>  
    <div class="class_Div_3">3</div>  
    <div>4</div>  
</div>
```

```
* {box-sizing: border-box;}  
.flex {  
    background-color: #eee;  
    width: 800px;  
    padding: 20px;  
    margin: 20px auto;  
    display: inline-flex;  
}  
.flex div {  
    background-color: #f00;  
    color: white;  
    width: 100px;  
    text-align: center;  
    padding: 20px;  
    margin-right: 5px;  
}  
.flex .class_Div_3{  
    flex-shrink:3; /* اصغر ثلاثة اضعاف */  
}
```

عند تقليل مساحة الاب سيكون تحفظ جميع الابناء بمساحتها عدا العنصر الابن الذى يحتوى على الخاصية – **Shrink** يكون له الاولوية فى التصغير ، الى ان يصل الى الحجم المخصص بعد ذلك تبدا باقى العناصر فى التصغير .

## Flex Box order

تغيير ترتيب الحاويات

```
<div class="flex">
  <div class="class_Div_1" >1</div>
  <div class="class_Div_2" >2</div>
  <div class="class_Div_3" >3</div>
  <div class="class_Div_4" >4</div>
</div>
```

```
* {box-sizing: border-box;}
.flex {
  background-color: #eee;
  width: 800px;
  padding: 20px;
  margin: 20px auto;
  display: inline-flex;
}
.flex div {
  background-color: #f00;
  color: white;
  width: 100px;
  text-align: center;
  padding: 20px;
  margin-right: 5px;
}
.flex .class_Div_1{
  order: 4;
}
.flex .class_Div_2{
  order: 3;
}
.flex .class_Div_3{
  order: 2;
}
.flex .class_Div_4{
  order: 1;
}
```

## Flex Basis

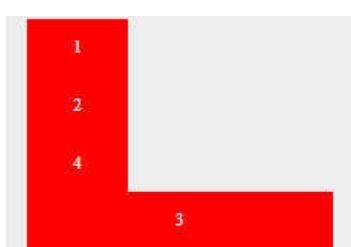
الوضع الافتراضي لهذه الخاصية (auto)

```
<div class="flex">
```

```
<div class="class_Div_1" >1</div>
<div class="class_Div_2" >2</div>
<div class="class_Div_3" >3</div>
<div class="class_Div_4" >4</div>
</div>
```

```
* {box-sizing: border-box;}
.flex {
    background-color: #eee;
    width: 800px;
    padding: 20px;
    margin: 20px auto;
    display: inline-flex;
    /* flex-direction: row; */
    flex-direction: column;
}
.flex div {
    background-color: #f00;
    color: white;
    width: 100px;
    text-align: center;
    padding: 20px;
    margin-right: 5px;
}
.flex .class_Div_3{
    width: 300px;
    /* flex-basis: 300px; */
}
```

عندما يكون الـ direction من صفات في الوضع الطبيعي يحسب المساحة (width) بالعرض ولكن عند تحويله إلى عمود يحسب المساحة بالعرض مما يؤدي إلى شكل خاطئ



ول لكن مع الخاصية direction يكون flex-basis عندما يكون الـ  
عمود يحسب المساحة بالطول وعندما يكون صف يحسب المساحة  
بالعرض

```
.flex {  
background-color: #eee;  
width: 800px;  
padding: 20px;  
margin: 20px auto;  
display: inline-flex;  
flex-direction: column; }
```

```
.flex .class_Div_3{  
/* width: 300px; */  
flex-basis: 300px;}
```

يمكن عمل max-width

```
* {box-sizing: border-box;}  
.flex {  
background-color: #eee;  
width: 800px;  
padding: 20px;  
margin: 20px auto;  
display: inline-flex;  
flex-direction: row;  
}  
.flex div {  
background-color: #f00;  
color: white;  
width: 100px;  
text-align: center;  
padding: 20px;  
margin-right: 5px;  
}  
.flex .class_Div_3{  
order: 2;  
flex-basis: 300px;  
max-width: 200px;}
```

shorthand

```
flex: 0 1 100px;  
/*      flex-grow: 0;  
        flex-shrink: 1;  
        flex-basis: 100px;  
*/
```

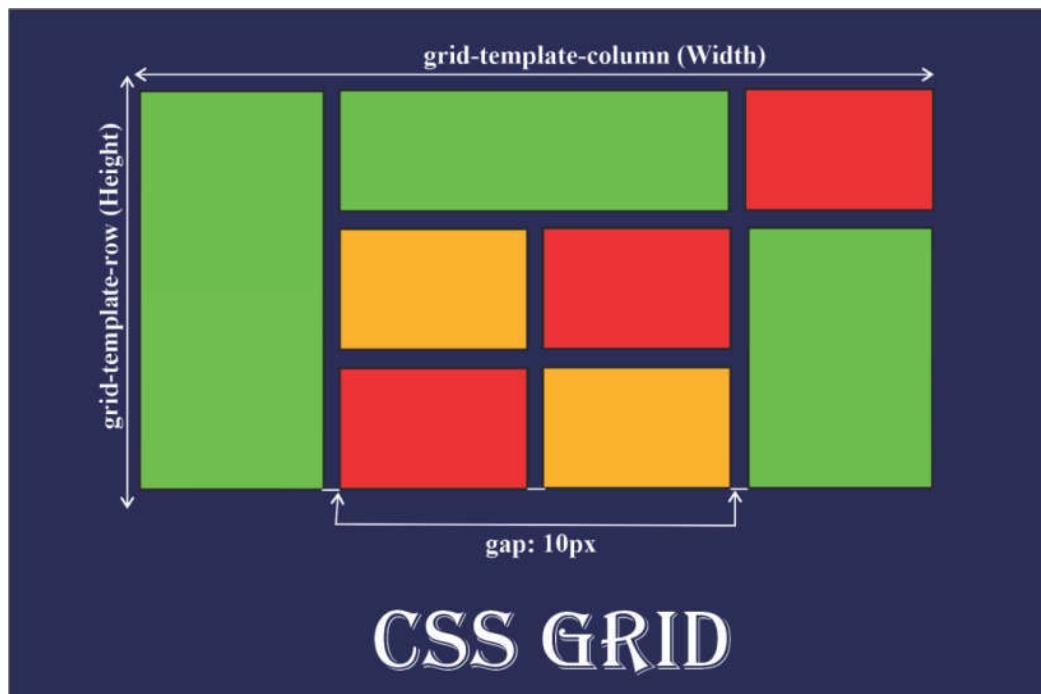
## لاحظ السطر التالي

لو وضعنا في القيمة الثانية على سبيل المثال `100px` سيتم وضعها بدلاً من `shrink` لأن هذه القيمة لا تصلح للـ `flex-basis` `shrink`

```
flex: 0 100px;
```

الخاصية `display: inline-flex` لعنصر `el` تمكنا من عرض عنصر `block` بجانب عنصر `block` في نفس السطر

## 2- grid system



## التعريف:

هو نظام شبكي وهمي لتنسيق الموقع

## display: grid

الوظيفة : تشغيل خاصية الـ grid

## grid-template-columns

لتقسيم الجدول الى عدد محدد من الاعمدة

```
grid-template-columns: ;
```

مثال

```
<div class="parent">
  <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>
```

```
* {box-sizing: border-box;}
.parent {
  margin: 20px auto;
  width: 800px;
  height: 500px;
  background-color: #ddd;
  display: grid;
  grid-template-columns: 100px 100px 100px ; /*3 value = 3 columns */
/*3columns =>one colum=3  */
}
.parent div {
  background-color: red;
  color: white;
  padding: 20px;
  font-size: 30px;
  font-weight: bold;
  text-align: center;
}
```

لاحظ هنا : لدينا عمودين فقط كل عمود تبلغ مساحته 50%

```
grid-template-columns: 50% 50%;
```

الخاصية Auto تعنى نسب متساوية : في هذا المثال لدينا قيم auto لثلاث اعمدة لو قمنا باضافة قيمة رابعة سيصبح لدينا اربع اعمدة

```
grid-template-columns: auto auto auto;
```

لاحظ هنا العمود الاول 50% والعمود الثاني auto اي سيأخذ باقى المساحة المتبقية

```
grid-template-columns: 50% auto;
```

الدالة repeat تستقبل عدد الاعمدة والمساحة

هنا في هذا المثال لدينا عمودين auto اي مساحة متساوية

```
grid-template-columns: repeat(2, auto); /*number columns , width */
```

الخاصية 1fr تعنى المساحة المتبقية ولكن وجوده مع العمود الـ Auto يجعل العمود Auto مساحته بحجم النص المكتوب فقط

```
.parent {  
    margin: 20px auto;  
    width: 800px;  
    height: 500px;  
    background-color: #ddd;  
    display: grid;  
    grid-template-columns: 50% auto;  
    grid-template-columns: auto auto 1fr 1fr;  
}
```

يمكن اختصاره

```
grid-template-columns: repeat(2, auto) repeat(2, 1fr);
```

## التحكم في ارتفاع الصفوف

```
.parent {  
    margin: 20px auto;  
    width: 800px;  
    height: 500px;  
    background-color: #ddd;  
    display: grid;  
    grid-template-columns: 50% auto;  
  
    grid-template-columns: 1fr 1fr 1fr 1fr;  
    grid-template-rows: 100px;  
}
```

الارتفاع الكلى ٥٠٠ ، والصف الاول ١٠٠ يتبقى ٤٠٠ سيتم توزيعهم  
تلقائى على باقى الصفوف  
يمكن تحديد ارتفاع كل سطر على حدى

```
grid-template-rows: 100px 300px 100px;  
row-gap & column-gap
```

المسافة بين الاعمدة والمسافة بين الصفوف

```
.parent {  
    margin: 20px auto;  
    width: 800px;  
    height: 500px;  
    background-color: #ddd;  
    display: grid;  
    grid-template-columns: 50% auto;  
  
    grid-template-columns: 1fr 1fr 1fr 1fr;  
    grid-template-rows: repeat(3,1fr);  
    row-gap: 3px ;  
    column-gap: 3px;  
}
```

القيمة الاولى للمسافة بين الصفوف والقيمة الثانية للمسافة بين الاعمدة

shorthand

```
gap: 3px 8px ;
```

## fr vs auto

يحتفظ بكل المساحة المتاحة : **fractional (fr)**

يحتفظ بمساحة بحجم المحتوى : **(auto)**

```
<div class="parent">
  <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>
```

```
* {box-sizing: border-box;}
.parent {
  margin: 20px auto;
  width: 800px;
  height: 500px;
  background-color: #ddd;
  display: grid;
  grid-template-columns:repeat(4,1fr);
  grid-template-rows:repeat(3,1fr);
  gap: 3px 3px ;
  justify-content: center;
}
.parent div {
  background-color: red;
  color: white;
  padding: 20px;
  font-size: 30px;
  font-weight: bold;
  text-align: center;
}
```

الخاصة `text-align` لن تقوم بوظيفتها في ظل وجود `grid-template-columns` بوحدة قياس "fr" لأنها تحتجز المساحة بالكامل لنفسها

```
grid-template-columns:repeat(4,1fr);
```

لأن لو قمنا بتغيير وحدة القياس من `fr` إلى `auto` سوف تعمل الخاصية `text-align` لأن الـ `auto` سيأخذ مساحة بحجم النص ويترك باقي المساحة.

```
grid-template-columns:repeat(4,auto);
```

"Html" مثال اخر "بنفس كود الـ

```
* {box-sizing: border-box;}  
.parent {  
    margin: 20px auto;  
    width: 800px;  
    height: 500px;  
    background-color: #ddd;  
    display: grid;  
    grid-template-columns:repeat(4,auto);  
    grid-template-rows:repeat(3,auto);  
    gap: 3px 3px ;  
    justify-content: space-between;  
}  
.parent div {  
    background-color: red;  
    color: white;  
    padding: 20px;  
    font-size: 30px;  
    font-weight: bold;  
    text-align: center;  
}
```

هنا أخذ مساحة بحجم النص فقط وترك باقي المساحة لأن الخاصية "align-content / justify-content" تتطلب مساحة من أجل التحرير أو ازاحة بين العناصر.

عند استخدام الـ `grid` لا نحتاج إلى عمل `flex-wrap` - لأن العناصر الزائدة يتم طردتها تلقائياً.

## grid-template-areas

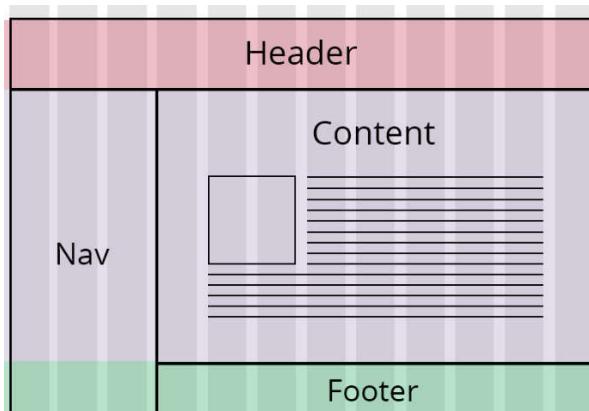
توزيع العناصر

```
<div class="page">
  <h1>Learn</h1>
<nav>
  <ul>
    <li>home</li>
    <li>about</li>
    <li>test</li>
    <li>context with me</li>
  </ul>
</nav>
<section>content</section>
<aside>aside</aside>
<footer>footer</footer>
```

```
* {box-sizing: border-box;
padding: 0;
margin: 0;
}
.page{
  display: grid;
  height: 100vh;
  grid-template-columns: repeat(10,1fr); /* عشر اعمدة */
  grid-template-rows:repeat(3,auto) ; /* ثلاثة صفوف بالتساوي */
}
```

الخاصية `display: grid` مع الارتفاع ستقوم بتقسيم العناصر ، ثم قمنا بتقسيم الاعمدة بالتساوي

#المطلوب توزيع العناصر بهذا الشكل



١- نقوم بتسمية منطقة "الاسم الذى تريده" للعنصر المراد توزيعه

```
h1{grid-area: logo; }  
nav{ grid-area: nav; }
```

تحليل

قمنا بجز مكان للعنصر h1 فى منطقة باسم **logo** وقمنا بجز  
مكان للعنصر nav فى منطقة باسم "**nav**"

٢- نقوم باضافة التوزيع الذى نرغبه في الاعمدة

```
.page{  
    display: grid;  
    height: 100vh;  
    grid-template-columns: repeat(10,1fr);  
    grid-template-rows:repeat(3,auto) ;  
    grid-template-areas:"logo logo nav nav nav nav nav nav nav";  
}
```

تحليل : اول عمودين للعنصر h1 وباقى الاعمدة للعنصر Nav

تكملاً للمثال :

```
.page{  
    display: grid;  
    height: 100vh;  
    grid-template-columns: repeat(10,1fr);  
    grid-template-rows:repeat(3,auto) ;  
    /*first row*/  
    grid-template-areas:"logo logo nav nav nav nav nav nav nav nav"  
    /*second row*/  
    "content content content content content content content content  
    aside aside"  
    /*third row*/  
    "footer footer footer footer footer footer footer footer  
    footer";  
}  
h1{grid-area: logo; }  
nav{grid-area: nav; }  
section{ grid-area: content; }  
aside{ grid-area: aside; }  
footer{ grid-area: footer; }
```

## تحليل :

الصف الاول عمودين لعنصر `h1` والباقي للمحتوى ، السطر الثاني ٨ اعمدة للمحتوى وعمودين للشريط الجانبي ، الصف الثالث لمنطقة ال `footer`

يمكن التحكم في ارتفاع الصفوف

```
grid-template-rows: 50px auto 50px;
```

## Grid Column And Grid Row

يمكن ان نقوم بدمج خلتين او اكثر في اتجاه **عرضى** عن طريق **grid-column** الخاصية

1								
2	3	4	5	6	7			
8	9	1	2	3	4			
5	6	7	8	9				

## طريقة تحديد نطاق الخلايا

نكتب رقم البداية ثم رقم النهاية + ١

```
<div style="grid-column-start: 1 ; grid-column-end: 7;">1</div>
```

## Shorthand

تحليل : بداية من الخلية رقم واحد الى الخلية رقم (٦)

```
grid-column: 1 / 7
```

# دليل المبرمج CSS3-part.1

```
<div class="parent">
/*دمج 6 خلايا على نفس الصف بداية من الخلية رقم واحد الى الخلية رقم 6*/
/*اذا اردنا كتابة عدد تزيد عليه رقم */
    <div style="grid-column: 1 / 7">1</div>
    <div>2</div>
    <div>3</div>
    <div>4</div>
    <div>5</div>
    <div>6</div>
    <div>7</div>
    <div style="grid-row: span 2">8</div>
    <div>9</div>
    <div>1</div>
    <div>2</div>
    <div>3</div>
    <div style="grid-column: span 3">4</div>
    <div>5</div>
    <div>6</div>
    <div>7</div>
    <div>8</div>
    <div>9</div>
</div>
```

```
* {box-sizing: border-box;}
.parent {
    margin: 20px auto;
    width: 800px;
    height: 500px;
    background-color: #ddd;
    display: grid;
    grid-template-columns: repeat(6, 1fr);
    grid-template-rows: repeat(3, auto);
    gap: 10px 10px;
}
.parent div {
    background-color: red;
    color: white;
    padding: 20px;
    font-size: 30px;
    font-weight: bold;
    text-align: center;
}
```

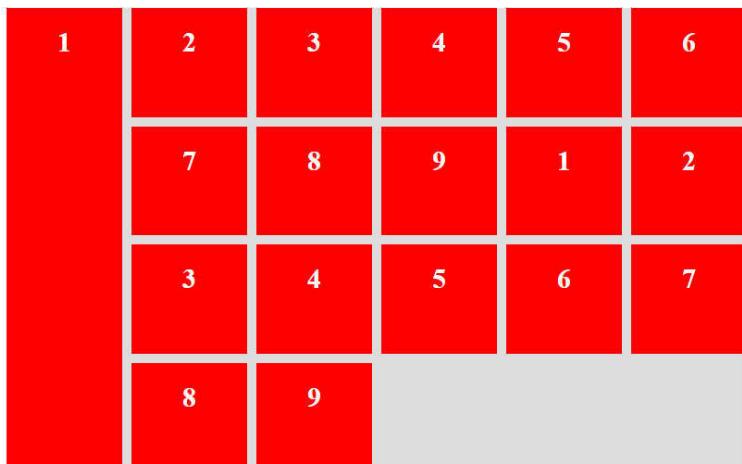
### grid-row

يمكن ان نقوم بدمج خلية او اكثرب في اتجاه طولى عن طريق  
الخاصية

### تحليل

"دمج اربع خلايا" في اتجاه طولى

```
<div style="grid-row: 1 / 5">1</div>
```



لتحديد عدد الخلايا المراد دمجها

```
<div style="grid-column: span 2">8</div>
```

## ٦ ميزة استخدام `:grid-column: span` ★○

عند دمج عدد خلايا اكبر من عدد الخلايا المتاحة في نفس الخط يتم عمل wrap

1	2	3	4	5	6
7	8	9	1	2	3
4	5	6	7	8	9

مثال

لو اردنا دمج من اول الخلية رقم خمسة في الصف الاول الى الخلية رقم ٧ ، سيتم عمل wrap للخلايا المدمجة

1	2	3	4			
5			6	7	8	
9	1	2	3	4	5	
6	7	8	9			

```
* {box-sizing: border-box;}  
.parent {  
    margin: 20px auto;  
    width: 800px;  
    height: 500px;  
    background-color: #ddd;  
    display: grid;  
    grid-template-columns: repeat(6, 1fr);  
    grid-template-rows: repeat(3, auto);  
    gap: 10px 10px;  
}  
.parent div {  
    background-color: red;  
    color: white;  
    padding: 20px;  
    font-size: 30px;  
    font-weight: bold;  
    text-align: center;  
}  
<div class="parent">  
    <div>1</div>  
    <div>2</div>  
    <div>3</div>  
    <div>4</div>  
    <div style="grid-column: span 3">5</div>  
    <div>6</div>  
    <div>7</div>  
    <div>8</div>  
    <div>9</div>  
    <div>1</div>  
    <div>2</div>  
    <div>3</div>  
    <div>4</div>  
    <div>5</div>  
    <div>6</div>  
    <div>7</div>  
    <div>8</div>  
    <div>9</div>  
</div>
```

## ابداً من رقم خلية معين

```
style="grid-column:2 ;
```

	1	2	3	4	5
6	7	8	9	1	
2	3	4	5	6	7
8	9				

## Grid Area

## تحريك عنصر من مكانه

```
<div class="parent">
  <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>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>
```

```
* {box-sizing: border-box;}  
.parent {  
    margin: 20px auto;  
    width: 800px;  
    height: 500px;  
    background-color: #ddd;  
    display: grid;  
    grid-template-columns: repeat(6, 1fr);  
    grid-template-rows: repeat(3, auto);  
    gap: 10px 10px;  
}  
.parent div {  
    background-color: red;  
    color: white;  
    padding: 20px;  
    font-size: 30px;  
    font-weight: bold;  
    text-align: center;  
}
```

1	2	3	4	5	6
7	8	9	1	2	3
4	5	6	7	8	9

## طريقة التحرير

```
<div class="parent">
  <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 style="grid-row-start:2 ; background-color: brown; ">9</div>
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
  <div>5</div>
  <div>6</div>
  <div>7</div>
  <div>8</div>
  <div>9</div>
</div>
```

: تحليل

٢ لنقل الخلية الى بداية الصف رقم grid-row-start:2

1	2	3	4	5	6
9	7	8	1	2	3
4	5	6	7	8	9

يمكن ان ننقل الخلية في العمود رقم ٢

```
grid-column-start: 2;
```

```
<div style="grid-row-start:2 ;grid-column-start: 2 ; background-color:  
blue; ">9</div>
```

1	2	3	4	5	6
7	9	8	1	2	3
4	5	6	7	8	9

يمكن ان نحدد نهاية رقم الصف نفس رقم الـ start مضاد اليه  
واحد لنقف فى نفس الخلية

```
grid-row-end: 3;
```

يمكن ان نحدد رقم العمود - عن طريق الخاصية column-start  
ويمكن ان نحدد (عدد الخلايا المدمجة فى اتجاه العرضي) grid  
عن طريق الخاصية column-end

```
<div style=" * تحديد السطر*/  
grid-row-start:2 ; /*اتجاه طولى*/  
grid-row-end: 4; /* تحديد بداية العمود*/  
grid-column-start: 2 ; /*اتجاه عرضى*/  
grid-column-end: 6 ;/*اتجاه عرضى*/  
background-color: blue;  
">9  
</div>
```

## Shorthand

```
<div style=" grid-area:2/2/4/6; background-color: black;">9
```

القيمة الاولى : row-start

القيمة الثانية : column -start

القيمة الثالثة : row- end

القيمة الرابعة : column - end

## Grid - Min, Max And Auto Fill

```
<div class="parent">
  <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>
```

```
* {box-sizing: border-box; margin: 0;}
.parent {
  margin: 20px auto;
  height: 500px;
  background-color: #ddd;
  display: grid;
  grid-template-columns:minmax(200px, 400px) repeat(2,1fr);
}
.parent div {
  background-color: red;
  color: white;
  padding: 20px;
  font-size: 30px;
  font-weight: bold;
  text-align: center;
}
```

## تحليل :

اقل مساحة للعمود الاول 200 واقصى مساحة 400 ، باقى مساحة العرض في العمودين سيتم توزيعها  
ملحوظة : اقصى مساحة لا يمكن ان تتعداها لان وحدة ال "fr" ستحجز باقى المساحة المتبقية .

## تجربة

```
grid-template-columns:minmax(200px,600px ) repeat(2,400px);
```

العمود الاول لن يصل الى اقصى قيمة لأن العمودين سيأخذ كل عمود منهم مساحة 400 ، اما العمود الاول فسيأخذ المساحة المتبقية من نطاق (اقل مساحة واقصى مساحة )

## وضع عدد معين من الاعمدة بمساحة محددة موزعة بالتساوي

```
grid-template-columns: repeat(9,100px);
```

## auto-fill

لانشاء عدد من الاعمدة تتناسب مع المساحة بمعنى تقسيم مساحة العرض المتاحة على مساحة عرض العمود الواحد لكي تشمل مساحة العرض على العدد الذي تستطيع عرضه

```
grid-template-columns: repeat(auto-fill,100px);
```

ملحوظة : لنفترض ان لدينا مساحة 1930 ولدينا خمس حاويات كل حاوية بمساحة 200px سيكون اجمالي المساحة لجميع الحاويات 1000px وسيبقى 30px

```
<div class="parent">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
  <div>5</div>
</div>
```

```
* {box-sizing: border-box; margin: 0;}  
.parent {  
margin: 20px auto;  
height: 500px;  
background-color: #ddd;  
display: grid;  
/* grid-template-columns: repeat(auto-fill, minmax(200px, 1fr)); */  
grid-template-columns: repeat(auto-fill, 200px);  
/* gap: 1px; */  
}  
.parent div {  
background-color: red;  
color: white;  
padding: 20px;  
font-size: 30px;  
font-weight: bold;  
text-align: center;  
}
```



## الحل في عدم ترك مساحة متباعدة فارغة ؟

```
grid-template-columns: repeat(auto-fill,  
minmax( 200px,1fr));
```

اذا كان هناك مساحة فارغة بعد توزيع الاعمدة بعرض ٢٠٠ للعنصر سيلجا تلقائى الى استخدام وحدة ال fr لتوزيع المساحة المتبقية على جميع العناصر