

بسمه تعالی

با سلام خدمت شما خواننده محترم

در ابتدا به پاس احترام به منبع اصلی این مقاله بایستی خدمت شما عرض کنم که این آموزش ترجمه (به همراه توضیحاتی مختصر در مباحثی که احساس کردم نیاز به تشریح بیشتر دارد) از سایت آموزشی بسیار عالی [www.w3schools.com](http://www.w3schools.com) می باشد.

همچنین این آموزش به صورت رایگان برای عموم علاقمندان در سایت کمیته امداد امام خمینی به آدرس زیر ارائه میگردد: [www.emdad.ir/central/amoozesh/css.asp](http://www.emdad.ir/central/amoozesh/css.asp)

در انتها خاطر نشان می کنم که تا قبل از تکمیل نهایی فایل دنالودی به همراه تاریخ آخرین آپدیت قرار داده شده و امیدوارم که مورد استفاده قرار گیرد.

علیرضا ترکمانی

[Arty125@emdad.ir](mailto:Arty125@emdad.ir)

## آموزش CSS

چه چیزی قبل از شروع باید بدانید؟

HTML / XHTML

CSS چیست؟

Css مخفف Cascading Style Sheets می باشد.

کلمه **Styles** به معنی مشخص کردن چگونگی نمایش عناصر HTML در صفحه وب می باشد. **Style** ها برای رفع برخی مشکلات به HTML 4,0 اضافه شدند.

**External Style Sheets** امکان کم کردن بار اضافی بسیاری را فراهم می کند. آنها در یک فایل **CSS** ذخیره شده و با تغییر یک عنصر اثر آن را در تمامی صفحات وب خود می توانید ببینید بدون اینکه نیاز داشته باشید در تک تک صفحات این تغییر را اعمال کنید.

**Style** ها چه مشکلاتی را رفع نمودند؟

HTML هیچگاه **tag** ی برای فرمت بندی یک **document** نداشت. HTML به منظور تعیین محتوای یک **document** تولید گردید. مثل `<h1> this is a heading </h1>`

زمانی که **tag** ها مثل `<font>` و خواص (**attributes**) مثل رنگ، به HTML 3,2 اضافه شد، شروع یک کابوس برای طراحان وب بود. طراحی یک سایت بسیار بزرگ با صفحات زیاد که در هر صفحه فونت ها و رنگها بایستی اضافه می شد، کاری بسیار سخت و طاقت فرسا بود. مخصوصا زمانی که مجبور بودید رنگ تمامی صفحات را تغییر دهید بایستی این تغییر را برای تک تک صفحات اعمال می کردید.

برای رفع این مشکل **the World Wide Web Consortium(W3C)** به ساخت **CSS** روی آورد.

در HTML 4,0 تمامی فرمت بندی ها بایستی از **HTML document** خارج شده و در یک فایل جداگانه با پسوند **CSS** ذخیره گردد.

تمامی مرورگرها امروز از **CSS** پشتیبانی می کنند.

## دستورات CSS

دستور CSS از دو جزء تشکیل شده است 1- یک انتخابگر selector 2- یک یا چند declaration همانند تعیین رنگ یا فونت و ..

Selector به طور معمول همان HTML element ی هست که می خواهید ظاهر آن را تغییر دهید.

هر declaration از دو جزء property و value تشکیل شده است. که property همان خاصیتی است که می خواهید ظاهر آن را تغییر دهید مثل font. هر property یک value دارد مثل blue برای property color به نام

در انتهای تمامی کدهای CSS بایستی از ; استفاده نمود. همچنین در صورتی که برای یک element بخواهیم تعدادی خاصیت (property) تعریف کنیم تمامی آنها را داخل {} قرار می دهیم. برای روشن تر شدن موضوع به مثال زیر توجه فرمایید:

```
p {color:red;text-align:center;}
```

در صورتیکه بخواهیم کد ما خوانا تر گردد می توانیم هر کد را در یک خط قرار دهیم به صورت زیر

```
p  
{  
color:red;  
text-align:center;  
}
```

### نوشتن توضیح (comment) برای کدهای css

Comment ها نوشته هایی هستند که ما برای توضیح عملکرد کدمان قبل از آن کد می نویسیم تا در صورتیکه که بعدهای نیاز به ویرایش پیدا کرد ، بتوانیم راحت تر موضوع کد را متوجه شویم. Comment ها توسط مرورگر تفسیر نمی شوند و نادیده گرفته می شوند.

Comment در CSS با `/*` شروع شده و با `*/` به پایان می‌رسد. شبیه مثال زیر :

```
/*This is a comment*/  
p  
{  
text-align:center;  
/*This is another comment*/  
color:black;  
font-family:arial;  
}
```

## Class و ID در CSS

از طریق selector هایی به نام ID و Class می‌توانیم به عناصر HTML دسترسی پیدا کرده تا بتوانیم استایل آنها را تغییر داده یا تعیین کنیم.

### ID selector

از این selector برای تعیین استایل یک عنصر (element) خاص استفاده می‌شود. این selector با یک `#` قبل از آن تعریف می‌شود. به مثال زیر توجه فرمایید :

```
#para1  
{  
text-align:center;  
color:red;  
}
```

`#para1` به این معنا است که `html element` ی مثل `<p>` داریم که `id` آن به نام `para1` می‌باشد. یعنی کد `html` ما به اینصورت تعریف شده است : `<p id=para1>`

در اینصورت کد `CSS` ی که ما نوشتیم به این معنی است که تمامی `text` ها در `<p id=para1>` وسطچین بوده و رنگ آن قرمز باشد.

**نکته :** برای تعریف نام یک `ID` هیچگاه از عدد در شروع نام آن استفاده نکنید.

### The class selector

از انتخابگر `class (selector)` برای تعیین استایل گروهی از عناصر `html` استفاده می‌شود. برخلاف `id` که فقط برای یک عنصر خاص استفاده می‌شود و نمی‌توان از آن `id` برای عناصر دیگر استفاده نمود.

با این شرایط می توانید از نام کلاس برای چندین عنصری که می خواهید تمامی خواص آن کلاس را داشته باشند استفاده کنید.

برای تعریف کلاس در CSS بایستی از یک . به همراه نام آن استفاده نمود.

در مثال زیر تمامی عناصری که با کلاس **center** تعریف شده اند، وسط چین می شوند.

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

همچنین می توانید کلاس را به عنصر خاصی از **html** نسبت دهید که فقط برای آن عنصر عمل نماید.

```
p.center {text-align:center;}
```

در مثال بالا تمامی عناصر **<P>** که با کلاس **center** تعریف شده اند (**<p class="center">**) وسط چین میگردند.

توجه : هیچگاه نام یک کلاس را با عدد شروع نکنید.

برای درک بهتر مباحث دو نمونه کد کامل ارائه می گردد :

نمونه 1 :

```
<html>
<head>
<style>
body {background-color:yellow;}
h1 {font-size:36pt;}
h2 {color:blue;}
p {margin-left:50px;}
</style>
</head>

<body>
```

```
<h1>This header is 36 pt</h1>
```

```
<h2>This header is blue</h2>
```

```
<p>This paragraph has a left margin of 50 pixels</p>
```

```
</body>
```

```
</html>
```

نمونه 2 :

```
<html>
```

```
<head>
```

```
<style>
```

```
body {background-color:tan;}
```

```
h1 {color:maroon;font-size:20pt;}
```

```
hr {color:navy;}
```

```
p {font-size:11pt;margin-left:15px;}
```

```
a:link {color:green;}
```

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

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

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

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>This is a header 1</h1>
```

```
<hr>
```

```
<p>You can see that the style  
sheet formats the text</p>
```

```
<p><a href="http://www.w3schools.com"  
target="_blank">This is a link</a></p>
```

```
</body>
```

```
</html>
```

چگونه کدهای CSS را در یک صفحه HTML مورد استفاده قرار دهیم؟

سه راه برای اضافه کردن کدها وجود دارد:

External style sheet -1

Internal style sheet -2

Inline style -3

**External Style Sheet**

وقتی که می خواهید استایل ها را برای چندین صفحه استفاده کنید و یا فکر می کنید که در آینده ممکن است که از آنها در چندین صفحه متفاوت استفاده شود از **External Style Sheet** استفاده کنید. از این طریق با تغییر فقط یک فایل می توانید کل صفحات وب خود را تغییر دهید.

برای اینکار هر صفحه بایستی از تگ `<link>` که در داخل `<head>` قرار گرفته است استفاده کرده باشد. به مثال زیر توجه نمایید:

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

کدهای یک **external style sheet** را می توان به وسیله یک **text editor** نوشت. این فایل نبایستی هیچ تگ **html** داشته باشد و این فایل با پسوند **CSS** ذخیره می گردد. یک مثال از یک فایل **CSS** را در زیر می بینید :

```
hr {color:sienna;}
p {margin-left:۲۰px;}
body {background-image:url("images/back۴۰.gif");}
```

نکته : بین مقدار خواص و واحد آنها فاصله قرار ندهید. مثلا نباید بین 20 و px که واحد آن است فاصله بگذارید.

## Internal Style Sheet

زمانیکه یک صفحه دارای استایل خاصی متفاوت از تمامی صفحات دیگر است از **internal style sheet** استفاده می کنیم. آن را در **head** به صورت زیر تعریف می کنیم:

```
<head>
<style>
hr {color:sienna;}
p {margin-left:۲۰px;}
body {background-image:url("images/back۴۰.gif");}
</style>
</head>
```

## Inline Style Sheet

با **inline style** بسیاری از مزیت های **CSS** را به دلیل ترکیب کردن محتوا و استایل از دست می دهیم. تا آنجا که امکان دارد از این روش استفاده نکنید.

برای استفاده از این روش، **style attribute** ها را در تک مرتبط با آن طوری تعریف می کنیم که می تواند هر کدام از **property** های **CSS** را شامل شود. به مثال زیر توجه نمایید :

```
<p style="color:sienna;margin-left:۲۰px;">This is a paragraph.</p>
```



## Multiple Style Sheet

اگر بعضی از **property** ها از طریق چند نوع **style sheet** تعیین شده باشند. یعنی مثلا برای تگ **h3** هم **external style** تعریف شده باشد و هم **internal style**. مقدار نهایی که برای آن **property** تعیین می گردد مربوط به **style sheet** ی است که درونی تر و نزدیک تر به **tag** است. در مثال ما یعنی مقادیر **internal style**

برای روشن تر شدن موضوع به مثال زیر توجه نمایید :

فرض کنید برای تگ **h3** یک **external style sheet**

```
h3
{
color:red;
text-align:left;
font-size:۸pt;
}
```

همچنین برای همان تگ در داخل صفحه وب یک **internal style sheet** به صورت زیر تعریف کرده ایم:

```
h3
{
text-align:right;
font-size:۲۰pt;
}
```

با این شرایط نتیجه به صورت زیر خواهد شد :

```
color:red;
text-align:right;
font-size:۲۰pt;
```

به این دلیل که چون در **internal** رنگ تعریف نشده بود پس مقدار رنگ **external** یعنی قرمز ست می شود. ولی چون دو خاصیت دیگر در **internal** نیز تعریف شده است، مقدار نهایی مرتبط با مقادیر **internal** خواهند بود.

نکته 1: با توجه به قانونی که در بالا گفته شد در صورتیکه این خواص به صورت **inline** نیز تعریف شده بود نتیجه نهایی برای خواص مشترک ، مقداری خواهد بود که در **inline** تعریف شده است. در نتیجه ترتیب اثر گذاری به صورت زیر است :

- Inline style sheet -1
- Internal style sheet -2
- External style sheet -3
- Browser default -4

نکته 2: در صورتیکه در تگ **<head>** لینکی که به **external style sheet** ارجاع می دهد بعد از لینکی که به **internal style sheet** ارجاع می دهد، قرار گیرد، اثر گذاری **external style sheet** بالاتر از **internal style sheet** خواهد بود. یعنی ترتیب اثر گذاری به ترتیب زیر تغییر می کند :

- Inline style sheet -1
- External style sheet -2
- Internal style sheet -3
- Browser default -4

## CSS Background

برای تنظیم خواص پس زمینه یک عنصر در CSS از **property** های زیر استفاده می کنیم :

- background-color
- background-image
- background-repeat
- background-attachment
- background-position

حال به ترتیب به تشریح آنها به می پردازیم :

## Background Color

این خاصیت رنگ پس زمینه یک عنصر (**element**) را مشخص می کند.

رنگ پس زمینه صفحه در **body selector** به صورت زیر تعیین می شود :

```
body {background-color:#b۰c۴de;}
```

در CSS تعیین یک رنگ اغلب به صورتهای زیر تعریف می شود

1 - a HEX value مثل "#ff0000"

2 - an RGB value مثل (۲۵۵,۰,۰) rgb

3 - a color name مثل "red"

در مثال زیر، برای عناصر h1, p و div رنگهای پس زمینه متفاوت تعریف شده است.

```
h1 {background-color:#۶۴۹۵ed;}
```

```
p {background-color:#e۰ffff;}
```

```
div {background-color:#b۰c۴de;}
```

## Background Image

با این خاصیت (property) می توانید تصویر پس زمینه یک عنصر را تغییر دهید. به صورت پیش فرض در صورتیکه تصویر کوچکتر از اندازه تعیین شده برای عنصر باشد، آنقدر تکرار می شود تا کل آن را بپوشاند.

تصویر پس زمینه برای یک صفحه به صورت زیر تعریف می شود :

```
body {background-image:url('paper.gif');}
```

## تکرار افقی یا عمودی تصویر در Background Image

به صورت پیش فرض خاصیت **background image** تصویر را هم به صورت افقی و هم عمودی تکرار می کند. در صورتیکه بخواهید تصویری که به عنوان پس زمینه عنصر خود انتخاب می کنید فقط به صورت افقی تکرار شود بایستی به صورت زیر عمل کنید :

```
body
```

```
{
```

```
background-image:url('gradient۲.png');
```

```
background-repeat:repeat-x;
```

```
}
```

تعیین جای دقیق **Background Image** به همراه عدم تکرار آن

نکته : زمانی که متنی به همراه عنصر خود دارید و می خواهید برای آن عنصر تصویر پس زمینه نیز انتخاب کنید، این تصویر باید جایی قرار گیرد که متن نیز خوانا باشد و تاثیری در خوانایی متن ایجاد نکند.

برای اینکه تصویر پس زمینه تکرار نشود به صورت زیر بایستی عمل کنیم:

```
body
{
background-image:url('img_tree.png');
background-repeat:no-repeat;
}
```

حال اگر این تصویر مانعی برای دیدن متون موجود در صفحه باشد و بخواهیم مکان قرار گرفتن تصویر را تغییر دهیم از خاصیت **background-position** استفاده می کنیم. مثالی در این زمینه که تصویر را گوشه بالا و راست صفحه قرار می دهد :

```
body
{
background-image:url('img_tree.png');
background-repeat:no-repeat;
background-position:right top;
}
```

نحوه نوشتن خاصیت **background** به صورت خلاصه و کوتاه

همانطور که تا الان دیدید، پس زمینه خواص متفاوتی دارد که در صورتیکه بخواهید از تمامی آنها یا برخی از آنها استفاده کنید، می توانید آنها را در یک خط به صورت خلاصه بنویسید. برای نوشتن تمامی خواص پس زمینه در یک خط از خاصیت **background** به صورت زیر استفاده می کنیم :

```
body {background:#ffffff url("img_tree.png") no-repeat right top;}
```

زمانیکه از این روش برای نوشتن دستورات استفاده می کنید ترتیب قرار گرفتن **value** ها به صورت زیر خواهد بود :

```
background-color -1
background-image -2
background-repeat -3
background-attachment -4
background-position -5
```

مهم نیست که اگر نخواهید برخی از این خواص را استفاده کنید، مهم این است که ترتیب صحیح باشد. یعنی اگر خاصیت شماره 4 را نمی خواهید استفاده کنید. ترتیب نوشتن **value** ها بایستی به ترتیبی که گفته شد بدون نوشتن **value** مرتبط با خاصیت شماره 4 باشد.

مثال زیر یک مثال کامل در این زمینه می باشد.

```
!>DOCTYPE html<
```

```
>html<
```

```
>head<
```

```
>style<
```

```
body
```

```
}
```

```
margin-left:200px;
```

```
background:5#d9ab 2url('img_tree.png') no-repeat top left;
```

```
{
```

```
.container
```

```
}
```

```
text-align:center;
```

```
{
```

```
.center_div
```

```
}
```

```
border:1px solid gray;
```

```
margin-left:auto;
```

```
margin-right:auto;
```

```
width:;%90
```

```
background-color:#d0f0f;6
```

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

```
padding:8px;
```

```
{
```

```
>body<
```

```
>div class="container"<
```

```
>div class="center_div"<
```

```
>h1>Hello World!</h1<
```

```
>p>This example contains some advanced CSS methods you may not have  
learned yet. But, we will explain these methods in a later chapter in the  
tutorial.</p<
```

نکته : چطور می توان پس زمینه را ثابت کرد طوری که بعد از **scroll** کردن صفحه، پس زمینه به همراه آن حرکت نکند؟

برای این منظور از خاصیت **background-attachment** استفاده کرده و **value** آن را **fixed** تعیین می کنیم. برای حس کردن موضوع کد زیر را اجرا کرده و نتیجه را ببینید :

```
!>DOCTYPE html<
```

```
>html<
```

```
>head<
```

```
>style<
```

```
body
```

```
}
```

```
background-image:url('w3css.gif;('
```

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

```
background-attachment:fixed;
```

```
{
```

```
/>style<
```

```
/>head<
```

```
>body<
```

```
>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>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>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>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>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/>
```

## CSS Text

### رنگ متون Text Color

خاصیت **color** برای تعیین رنگ متن استفاده می شود.

در CSS رنگها به 3 روش زیر تعیین می شوند :

- a Hex Value مثل "#ff0000"

- an RGB value مثل "rgb(255,0,0)"

- نام رنگ مثل "red"

رنگ پیش فرض متون یک صفحه در قسمت **body** تعریف می گردد :

```
body {color:blue;}
```

```
h1 {color:#00ff00;}
```

```
h2 {color:rgb(255,0,0);}
```

**نکته :** بر اساس توافقنامه W3C در صورتیکه خاصیت **color** را تعیین کنید بایستی خاصیت

**background-color** را نیز تعیین نمایید.

### Text Alignment

خاصیت **Text-align** برای تنظیم ترازبندی افقی یک متن تعیین می گردد. متن می تواند وسط چین، راست

چین یا چپ چین باشد.



وقتی خاصیت **text-align** به حالت **justify** تعیین می گردد، تمامی خطوط طوری کشیده می شوند که عرض تمامی آنها به همراه فاصله چپ و راست آنها از لبه صفحه یکسان گردد.(شبيه مجله و روزنامه)

مثال :

```
h1 {text-align:center;}
p.date {text-align:right;}
p.main {text-align:justify;}
```

## Text Decoration

خاصیت **text-decoration** برای حذف یا اضافه کردن خط (زیرخط، خط وسط یا خط بالای متن) به متن استفاده می شود. معمولاً از این خاصیت برای برداشتن زیرخط از لینکها به منظور نمایش بهتر در صفحه استفاده می شود.

مثال :

```
a {text-decoration:none;}
h1 {text-decoration:overline;}
h2 {text-decoration:line-through;}
h3 {text-decoration:underline;}
```

نکته : بهتر است از زیرخط برای متونی که لینک نیستند استفاده نکنید تا باعث سردرگمی کاربران نشود.

## Text-Transformation

خاصیت **text-transformation** برای تعیین حروف بزرگ یا کوچک متون استفاده می شود. از آن برای تغییر تمامی متن به حروف بزرگ یا حروف کوچک و یا تغییر حرف اول کلمه به بزرگ استفاده می شود.

مثال :

```
p.uppercase {text-transform:uppercase;}
p.lowercase {text-transform:lowercase;}
p.capitalize {text-transform:capitalize;}
```

## Text Indentation

خاصیت **text-indent** برای شروع یک پاراگراف به کار می رود به طوریکه باعث می شود اولین خط مقداری جلوتر از بقیه خطها شروع شود. مقدار جلوتر بودن بر اساس عددی است که ما مشخص می کنیم.

```
p {text-indent:۵۰px;}
```

ایجاد فاصله بین کاراکترها و خطوط

از خاصیت **letter-spacing** برای ایجاد فاصله بین کاراکترهای کلمات استفاده می شود.

```
h1 {letter-spacing:5px;}
```

از خاصیت **line-height** برای ایجاد فاصله بین خطوط استفاده می شود.

```
p.small {line-height:%170;}
```

## Direction Text

از خاصیت **direction** برای تعیین جهت شروع نوشتن متن استفاده می شود. مثلا **rtl** به معنی این است که متون از سمت راست نوشته و نمایش داده شوند.

```
direction:rtl
```

```
direction:ltr
```

## تغییر فضای خالی بین کلمات

از خاصیت **word-spacing** برای افزایش یا کاهش فضای خالی (**space**) بین کلمات استفاده می شود

```
word-spacing:10px;
```

از کار انداختن قابلیت **wrap** شدن متن (به طوریکه متن طولانی فقط در یک خط نوشته شود)

از خاصیت **white-space** برای از کار انداختن **Wrap** شدن متون استفاده می شود :

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

## اضافه کردن سایه به متن

از خاصیت **text-shadow** برای ایجاد سایه به متن می توان استفاده کرد.

```
h1 {text-shadow:2px 2px #ff0000}
```

عدد اول فاصلی افقی سایه با متن اصلی، عدد دوم فاصله عمودی سایه با متن اصلی و مقدار آخر رنگ سایه می باشد.

## تمامی خواص مرتبط با متن در یک نگاه

Description	Property
Sets the color of text	color
Specifies the text direction/writing direction	direction
Increases or decreases the space between characters in a text	letter-spacing
Sets the line height	line-height
Specifies the horizontal alignment of text	text-align
Specifies the decoration added to text	text-decoration
Specifies the indentation of the first line in a text-block	text-indent
Specifies the shadow effect added to text	text-shadow
Controls the capitalization of text	text-transform
Used together with the direction property to set or return whether the text should be overridden to support multiple languages in the same document	unicode-bidi
Sets the vertical alignment of an element	vertical-align
Specifies how white-space inside an element is handled	white-space
Increases or decreases the space between words in a text	word-spacing

## CSS Font

### CSS Font Families

دو خانواده از فونت ها در CSS وجود دارد:

1- Generic family : گروهی از فونت ها با ظواهر یکسان مثل serif و monospace

2- Font Family : گروه خاصی از فونتها مثل Times new roman و Arial

### Font Family

برای تنظیم فونت یک متن از خاصیت **font-family** استفاده می کنیم. برای این خاصیت بایستی چندین فونت را تعریف کرد که در صورتیکه مرورگر نتوانست یکی را تشخیص دهد یا آن را پشتیبانی نمی کرد، امکان استفاده از فونت دیگری را داشته باشد.

برای این منظور، با فونتی که دوست دارید شروع کنید و در ابتدا آن را تعریف کنید و سپس در انتها از یک فونت متعلق به **generic family** استفاده کنید تا مطمئن شوید در هر شرایطی متون شما درست نمایش داده می شود.

نکته : اگر نام یک فونت از یک کلمه بیشتر بود، بایستی آن را داخل " " قرار دهید. برای تعیین بیش از یک فونت در خاصیت **font-family** بایستی بین فونتها از کاما (,) استفاده کنید. مثال :

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

## Font Style

خاصیت **font-style** معمولا برای اینکه متن ها را به حالت **italic** تغییر دهید استفاده می شود. این خاصیت 3 مقدار می گیرد :

**normal** : متن به صورت معمول دیده می شود

**Italic** : متن به صورت حروف کج **italic** دیده می شود.

**Oblique** : خیلی شبیه به **italic** می باشد و متن حالت خمیده دارد. (کمتر پشتیبانی می شود در مرورگرها)

مثال :

```
p.normal {font-style:normal;}  
p.italic {font-style:italic;}  
p.oblique {font-style:oblique;}
```

## Font Size

خاصیت **font-size** اندازه متن را مشخص می کند. تعیین اندازه فونت در وب بسیار مهم است. شما بایستی اندازه ها را طوری تعیین کنید که پاراگراف ها شبیه هدرها دیده شوند و یا بالعکس.

همیشه سعی کنید از تگ **html** مناسب مثل **<h1>** تا **<h6>** برای هدرها و **<p>** برای پاراگراف ها استفاده کنید.

اندازه فونت می تواند عددی ثابت (**absolute size**) و یا نسبی (**Relative size**) باشد.

## Absolute size:

- اندازه فونت ها در تمامی مرورگرها یکسان است و تغییر نمی کند. (برای کسانی که مشکل بینایی دارند خوب نیست)

## Relative Size:

- اندازه بر اساس عناصر اطراف آن تعیین می گردد.
- به کاربر امکان تغییر اندازه فونت را در مرورگر می دهد.

نکته : اگر اندازه فونت را مشخص نکنید ، اندازه پیش فرض برای متن در حالت نرمال مثل پاراگراف، 16px می باشد.

## تعیین اندازه فونت بر اساس pixel

در صورتیکه اندازه فونت را بر اساس pixel تنظیم کنید، امکان کنترل کامل بر روی اندازه فونت را خواهید داشت.

مثال :

```
h1 {font-size: 40px;}  
h2 {font-size: 30px;}  
p {font-size: 14px;}
```

در مثال بالا با استفاده از IE 9 به بالا ، safari, Opera, Firefox و می توانید اندازه متن را تغییر دهید.

نکته : در صورتیکه از IE 9 به پایین استفاده می کنید، نمی توانید اندازه فونت را در صورتیکه کد شما همانند مثال بالا باشد تغییر دهید.

متون در تمامی مرورگر ها به وسیله ابزار zoom امکان تغییر اندازه دارند. (هر چند که این امکان اندازه کل صفحه را تغییر می دهد نه فقط متون را)

## تعیین اندازه فونت با استفاده از Em

برای اجتناب از مشکل تغییر اندازه با نسخه های قدیمی اینترنت اکسپلورر، برخی از developer ها ، از واحد em به جای pixel استفاده می کنند. استفاده از واحد اندازه em به وسیله W3C پیشنهاد شده است.

1em برابر است با اندازه پیش فرض فونت یعنی 16px. بنابراین از طریق فرمول زیر می توان این دو واحد را به هم تبدیل کرد :  $pixel/16 = 1em$

مثال :

```
h1 {font-size: 2.5em;} /* 40px/16=2.5em */  
h2 {font-size: 1.875em;} /* 30px/16=1.875em */  
p {font-size: 0.875em;} /* 14px/16=0.875em */
```

## استفاده از ترکیب Percent و Em :

با توجه به اینکه متاسفانه هنوز برخی از مرورگرها قدیمی با اندازه فونت ها مشکل دارند و متن را بزرگتر یا کوچکتر از آن چیزی که هست نمایش می دهند، روشی که در تمامی مرورگرها کار می کند و مشکل را برطرف می کند این است که در قسمت `<body>` اندازه فونت را به صورت پیش فرض به درصد همانند مثال زیر تعیین کنید.

```
body {font-size:۱۰۰%;}
h۱ {font-size:۲,۰em;}
h۲ {font-size:۱,۸۷۰em;}
p {font-size:۰,۸۷۰em;}
```

حال کد ما به طور کامل و درست کار می کند و در تمامی مرورگرها اندازه متن درست نمایش داده می شود. همچنین در تمامی مرورگرها امکان zoom و تغییر اندازه متن وجود دارد.

تمامی خواص فونتها در CSS به طور خلاصه

Description	Property
Sets all the font properties in one declaration	font
Specifies the font family for text	font-family
Specifies the font size of text	font-size
Specifies the font style for text	font-style
Specifies whether or not a text should be displayed in a small-caps font	font-variant
Specifies the weight of a font	font-weight

## CSS Links

### Styling Links

استایل لینکها را می توان با تمامی خواص موجود در CSS تغییر داد. بسته به اینکه state لینک چیست، میتوان استایل آن را تغییر داد.

چهار state برای لینک ها وجود دارد :

- 1- `a:link` – لینکی که هنوز روی آن کلیک نشده و دیده نشده است.
- 2- `a:visited` – لینکی که کاربر بر روی آن کلیک کرده و آن را دیده است.
- 3- `a:hover` – لینکی که کاربر اشاره گر موس را بر روی آن قرار داده ولی کلیک نکرده است.
- 4- `a:active` – لینکی که در همان لحظه کلیک شده است.

مثال :

```
a:link {color:#FF۰۰۰;} /* unvisited link */
a:visited {color:#۰۰FF۰۰;} /* visited link */
a:hover {color:#FF۰۰FF;} /* mouse over link */
a:active {color:#۰۰۰۰FF;} /* selected link */
```

هنگامیکه برای **state** چندین لینک، استایل تعیین می کنید، قوانین زیر را در ترتیب نوشتن استایل ها بایستی رعایت نمایید :

دستور **a:hover** بایستی بعد از **a:link** و **a:visited** نوشته شود.

دستور **a:action** بایستی بعد از **a:hover** نوشته شود.

### انواع معمول استایل like ها

در مثال بالا دیدید که رنگ لینک ها بر اساس **state** ی که پیدا می کنند، تغییر می کند.

حالا برخی دیگر از روشهایی که می توانیم استایل لینکها را تغییر دهیم را با هم مرور می کنیم :

### Text Decoration

خاصیت **text-decoration** اکثرا برای برداشتن زیرخط موجود در لینکها استفاده می شود. مثال :

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

در این دو زیرخط لینک برداشته می شود.

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

در این دو حالت زیر خط لینک نمایش داده می شود.

### Background Color

خاصیت **background-color** رنگ پس زمینه لینک را تعیین می کند.

```
a:link {background-color:#B۲FF۹۹;}
a:visited {background-color:#FFFF۸۵;}
a:hover {background-color:#FF۷۰۴D;}
a:active {background-color:#FF۷۰۴D;}
```

در مثال بالا نشان دادیم که بسته به **state** لینک، رنگ پس زمینه آن چه رنگی باشد.

در زیر دو مثال کامل در خصوص استایل دادن به لینکها را به همراه انواع حالات مختلف آورده شده است.

مثال 1 :

```
!DOCTYPE html<
```

```
>html<
```

```
>head<
>style<
a.one:link {color:#ff{;0000
a.one:visited {color:0000#ff{;
a.one:hover {color:#ffcc{;00

a.two:link {color:#ff{;0000
a.two:visited {color:0000#ff{;
a.two:hover {font-size:{;%150

a.three:link {color:#ff{;0000
a.three:visited {color:0000#ff{;
a.three:hover {background:66#ff{;66

a.four:link {color:#ff{;0000
a.four:visited {color:0000#ff{;
a.four:hover {font-family:monospace{;

a.five:link {color:#ff0000;text-decoration:none{;
a.five:visited {color:0000#ff;text-decoration:none{;
a.five:hover {text-decoration:underline{;
/>style<
/>head<

>body<
>p>Mouse over the links to see them change layout.</p<

>p><b><a class="one" href="default.asp" target="_blank">This link
changes color</a></b></p<
>p><b><a class="two" href="default.asp" target="_blank">This link
changes font-size</a></b></p<
```



```
>p><b><a class="three" href="default.asp" target="_blank">This link
changes background-color</a></b></p>
>p><b><a class="four" href="default.asp" target="_blank">This link
changes font-family</a></b></p>
>p><b><a class="five" href="default.asp" target="_blank">This link changes
text-decoration</a></b></p>
/>body<
```

```
/>html<
```

: مثال 2

```
!>DOCTYPE html<
>html<
>head<
>style<
a:link,a:visited
}
display:block;
font-weight:bold;
color:#FFFFFF;
background-color:98#bf;21
width:120px;
text-align:center;
padding:4px;
text-decoration:none;
{
a:hover,a:active
}
background-color:7#A991A;
{
/>style<
/>head<
```

```
>body<
>a href="default.asp" target="_blank">This is a link</a<
/>body<
/>html<
```

## CSS Lists

خاصیت **list** به شما اجازه انجام موارد ذیل را می دهد.

- تعیین نشانگرهای متفاوت برای مواردی که ترتیبی نیستند و می خواهید آنها را برای نمایش لیست کنید. مثل نشانگر \* )
- تعیین نشانگرهای متفاوت برای مواردی که ترتیبی هستند و می خواهید آنها را برای نمایش لیست کنید. مثل نشانگر a یا II )
- می توانید از یک تصویر به عنوان یک نشانگر برای لیست کردن موضوعات خود استفاده کنید.

در ادامه به شرح تمامی حالات می پردازیم.

### انواع نشانگرهای مورد استفاده برای موضوعات مختلف

نوع نشانگر (list item marker) به وسیله خاصیت **list-style-type** تعیین می گردد. مثال :

```
ul.a {list-style-type: circle;}
ul.b {list-style-type: square;}
```

```
ol.c {list-style-type: upper-roman;}
ol.d {list-style-type: lower-alpha;}
```

برخی از **value** هایی که برای این خاصیت در بالا ذکر گردیده برای لیست ها ترتیبی و برخی دیگر برای لیست هایی هستند که ترتیب در آنها لحاظ نشده است. مثال کامل ذیل موضوع را براتون روشن می کند :

```
!>DOCTYPE html<

>html<

>head<

>style<

ul.a {list-style-type:circle{;

ul.b {list-style-type:square{;
```

```
ol.c {list-style-type:upper-roman{;
```

```
ol.d {list-style-type:lower-alpha{;
```

```
>body<
```

```
>p>Example of unordered lists:</p<
```

```
>ul class="a"<
```

```
> li>Coffee</li<
```

```
> li>Tea</li<
```

```
> li>Coca Cola</li<
```

```
/>ul<
```

```
>ul class="b"<
```

```
> li>Coffee</li<
```

```
> li>Tea</li<
```

```
> li>Coca Cola</li<
```

```
/>ul<
```

```
>p>Example of ordered lists:</p<
```

```
>ol class="c"<
```

```
> li>Coffee</li<
> li>Tea</li<
> li>Coca Cola</li<
/>ol<

>ol class="d"<
> li>Coffee</li<
> li>Tea</li<
> li>Coca Cola</li<
/>ol<

/>body<

/>html<
```

استفاده از یک تصویر به عنوان نشانگر موضوعی :

برای استفاده از یک تصویر به عنوان نشانگر خاصیت `list-style-image` را به صورت مثال ذیل تعریف می کنیم :

```
!>DOCTYPE html<

>html<

>head<

>style<

ul
}

list-style-image:url('sqpurple.gif;('

{

/>style<
```

```
</>head<
```

```
>body<
```

```
>ul<
```

```
>li>Coffee</li<
```

```
>li>Tea</li<
```

```
>li>Coca Cola</li<
```

```
</ul<
```

```
</body<
```

```
</html<
```

تصاویر مثال بالا در تمامی مرورگر های به یک صورت نمایش داده نمی شود، IE و Opera نشانگر تصویری را یک مقدار نسبت به **firefox, chrome** و **safari** بالاتر نمایش می دهد.

اگر می خواهید **image-marker** (نشانگر تصویری) در تمامی مرورگرها به صورت یکسان نمایش داده شود بایستی از روش **crossbrowser** که در زیر به شرح آن می پردازیم استفاده نمایید:

## Crossbrowser Solution

در مثال زیر تمامی نشانگرهای تصویری به صورت یکسان در تمامی مرورگرها نمایش داده می شوند :

```
<!DOCTYPE html<
```

```
<html<
```

```
<head<
```

```
<style<
```

```
ul
```

```
}
```

```
list-style-type:none;
```

```
padding:0px;
margin:0px;
{
ul li
}
background-image:url(sqpurple.gif);
background-repeat:no-repeat;
background-position:0px 5px ;
padding-left:14px;
{
/>style<
/>head<
>body<
>ul<
>li>Coffee</li<
>li>Tea</li<
>li>Coca Cola</li<
/>ul<
/>body<
/>html<
```

توضیح مثال بالا :

برای تگ ul :

- مقدار **List-item-type** را به منظور حذف **list-item-marker** برابر **none** قرار دادیم.
- برای سازگاری با **cross-browser** مقدار هر دو خاصیت **padding** و **margin** را صفر قرار دادیم.

برای تمامی تگ های **li** موجود در **ul** :

- آدرس لینک تصویری که بایستی نمایش داده شود را تعیین کرده و مقدار **no-repeat** را برای جلوگیری از تکرار آن تعیین می کنیم.
- محل قرار گرفتن تصویر را از طریق خاصیت **background-position** تعیین می کنیم.
- محل قرار گرفتن متن در لیست ها را با خاصیت **padding-left** تعیین می کنیم.

استفاده از خاصیت **list** در یک خط و به صورت خلاصه :

این امکان وجود دارد که تمامی خواص مرتبط با لیست را در یک خط تعریف نماییم که به این حالت تعریف **shorthand property** گفته می شود.

خاصیت **shorthand** ی که برای لیست استفاده می شود **list-style** می باشد. مثال :

```
ul
}
list-style:square url("sqpurple.gif");
{
```

وقتی از خاصیت **list** به صورت خلاصه نویسی (**shorthand**) استفاده می کنید بایستی ترتیب زیر را برای مقدار دهی رعایت نمایید : (بدیهی است در صورتیکه برخی از مقادیر را تعریف نکنید، در صورتیکه ترتیب را رعایت کرده باشید، CSS آن را تشخیص داده و اعمال می کند).

List-style-type -1

List-style-position -2

List-style-image -3

تمامی خواص list در یک نگاه :

Description	Property
Sets all the properties for a list in one declaration	list-style
Specifies an image as the list-item marker	list-style-image
Specifies if the list-item markers should appear inside or outside the content flow	list-style-position
Specifies the type of list-item marker	list-style-type

## ساخت جدول در CSS

### Table Borders

برای تعیین border جداول در CSS از خاصیت border استفاده می کنیم. مثال زیر برای عناصر th و td جدول یک border سیاه درست می کند.

```
<style>
```

```
table,th,td
```

```
{
```

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

```
}
```

```
</style>
```

توجه دارید که در جدول مثال بالا border به صورت دو خطی می باشد. این به این دلیل است که هر دوی th و td دارای border های جداگانه می باشند. برای نمایش border به صورت تک خط، از خاصیت border-collapse استفاده نمایید.

### Collapse Borders

خاصیت border-collapse همانطور که در بالا گفته شد، برای اینکه border به صورت تک خط نمایش داده شود، استفاده می گردد. به مثال زیر توجه نمایید :

```
<style>
```

```
table
```



```
{
border-collapse:collapse;
}
table, td, th
{
border:1px solid black;
}
</style>
```

### خاصیت width و Height جدول

برای تعیین عرض و ارتفاع یک جدول از خاصیت **width** و **height** استفاده می گردد.

مثال زیر عرض جدول را 100% (یعنی اندازه عرض صفحه) ارتفاع عنصر **th** را 50 تعیین می کند.

```
<style>
```

```
table,td,th
```

```
{
```

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

```
}
```

```
table
```

```
{
```

```
width:100%;
```

```
}
```

```
th
```

```
{
```

```
height:100px;
```

```
}  
</style>
```

## Table Text Alignment

تنظیم محل قرار گرفتن یک متن در یک جدول از طریق خواص `text-align` و `vertical-align` تعیین می گردد.

خاصیت `text-align` تنظیمات مرتبط با قرار گرفتن افقی متن اعم از راست چین، چپ چین یا وسط را تعیین می کند (`left, right, center`). مثال :

```
<style>  
table,td,th  
{  
border:1px solid black;  
}  
td  
{  
text-align:right;  
}  
</style>
```

خاصیت `vertical-align` تنظیمان قرار گرفتن عمودی متن مثل بالا، پایین یا وسط را تعیین می کند. (`top, bottom, middle`). مثال :

```
<style>  
table, td, th  
{  
border:1px solid black;  
}
```

```
td
{
height:50px;
vertical-align:bottom;
}
</style>
```

## Table Padding

برای کنترل فاصله ی بین محتوا و border یک جدول ، از خاصیت padding برروی عناصر th و td استفاده می کنیم. این خاصیت باعث می شود محتوای موجود در یک td از سمت چپ به مقداری که تعیین کردیم ایم با لبه کناری جدول فاصله بگیرد. هر چه این اندازه بزرگتر گردد اندازه کل td نیز بزرگتر می گردد. مثال :

```
<style>
table, td, th
{
border:1px solid black;
}
td
{
padding:15px;
}
</style>
```

## Table Color

برای تعیین رنگ **border** ها و یا متون و یا رنگ پس زمینه، از خاصیت **color** می توان استفاده کرد. در مثال زیر رنگ **border** جدول به همراه متن داخل آن و رنگ پس زمینه آن را تعیین کرده ایم.

```
<!DOCTYPE html>
<html>
<head>
<style>
table, td, th
{
border:1px solid green;
}
th
{
background-color:green;
color:white;
}
</style>
</head>

<body>
<table>
<tr>
<th>Firstname</th>
<th>Lastname</th>
<th>Savings</th>
</tr>
<tr>
```

```
<td>Peter</td>
<td>Griffin</td>
<td> 100</td>
</tr>
<tr>
<td>Lois</td>
<td>Griffin</td>
<td> 150</td>
</tr>
<tr>
<td>Joe</td>
<td>Swanson</td>
<td> 300</td>
</tr>
<tr>
<td>Cleveland</td>
<td>Brown</td>
<td> 250</td>
</tr>
</table>
</body>
</html>
```

مثالی در ارتباط با ایجاد یک جدول زیبا و تجملی :

```
<!DOCTYPE html>
<html>
<head>
```

```
<style>
#customers
{
font-family:"Trebuchet MS", Arial, Helvetica, sans-serif;
width:100%;
border-collapse:collapse;
}
#customers td, #customers th
{
font-size:1em;
border:1px solid #98bf21;
padding:3px 7px 2px 7px;
}
#customers th
{
font-size:1.1em;
text-align:left;
padding-top:5px;
padding-bottom:4px;
background-color:#A7C942;
color:#ffffff;
}
#customers tr.alt td
{
color: 000000;
background-color:#EAF2D3;
```

```
}
</style>
</head>

<body>
<table id="customers">
<tr>
<th>Company</th>
<th>Contact</th>
<th>Country</th>
</tr>
<tr>
<td>Alfreds Futterkiste</td>
<td>Maria Anders</td>
<td>Germany</td>
</tr>
<tr class="alt">
<td>Berglunds snabbköp</td>
<td>Christina Berglund</td>
<td>Sweden</td>
</tr>
<tr>
<td>Centro comercial Moctezuma</td>
<td>Francisco Chang</td>
<td>Mexico</td>
</tr>
<tr class="alt">
```

<td>Ernst Handel</td>

<td>Roland Mendel</td>

<td>Austria</td>

</tr>

<tr>

<td>Island Trading</td>

<td>Helen Bennett</td>

<td>UK</td>

</tr>

<tr class="alt">

<td>Königlich Essen</td>

<td>Philip Cramer</td>

<td>Germany</td>

</tr>

<tr>

<td>Laughing Bacchus Winecellars</td>

<td>Yoshi Tannamuri</td>

<td>Canada</td>

</tr>

<tr class="alt">

<td>Magazzini Alimentari Riuniti</td>

<td>Giovanni Rovelli</td>

<td>Italy</td>

</tr>

<tr>

<td>North/South</td>

<td>Simon Crowther</td>



```
<td>UK</td>
</tr>
<tr class="alt">
<td>Paris spécialités</td>
<td>Marie Bertrand</td>
<td>France</td>
</tr>
</table>
</body>
</html>
```

مثالی دیگر در ارتباط با نحوه تعیین مکان caption یک جدول:

```
<!DOCTYPE html>
<html>
<head>
<style>
caption {caption-side:bottom;}
</style>
</head>

<body>

<table border="1">
<caption>Table 1,1 Customers</caption>
<tr>
<th>Company</th>
<th>Contact</th>
```

```
<th>Country</th>
</tr>
<tr>
<td>Alfreds Futterkiste</td>
<td>Maria Anders</td>
<td>Germany</td>
</tr>
<tr>
<td>Berglunds snabbköp</td>
<td>Christina Berglund</td>
<td>Sweden</td>
</tr>
<tr>
<td>Centro comercial Moctezuma</td>
<td>Francisco Chang</td>
<td>Mexico</td>
</tr>
<tr>
<td>Ernst Handel</td>
<td>Roland Mendel</td>
<td>Austria</td>
</tr>
<tr>
<td>Island Trading</td>
<td>Helen Bennett</td>
<td>UK</td>
</tr>
```

```
<tr>
<td>Magazzini Alimentari Riuniti</td>
<td>Giovanni Rovelli</td>
<td>Italy</td>
</tr>
<tr>
<td>North/South</td>
<td>Simon Crowther</td>
<td>UK</td>
</tr>
</table>
```

```
<p><b>Note:</b> IE8 supports the caption-side
property if a !DOCTYPE is specified.</p>
</body>
</html>
```

## CSS Box Model

تمامی عناصر **html** به عنوان یک جعبه (**box**) به صورت شکلی که در زیر آمده است در نظر گرفته می شوند. عبارت **box model** زمانیکه درباره طراحی و طرح بندی صحبت می کنیم، مطرح می شود.

**Box model** در **css** به این معنی است که عنصر شما را در داخل یک چارچوب (همانند جعبه) در نظر می گیرد که این چارچوب دارای خواص زیر است : **margins,borders,padding** به همراه محتوای اصلی مرتبط با عنصر.

**Box model** به ما این امکان را می دهد که اطراف عنصر خود **border** قرار داده و فاصله عناصر را با بقیه عناصر تعیین نماییم. تصویر زیر مفهوم **box model** را نشان می دهد.



حال به تشریح قسمت‌های مختلف آن می پردازیم :

**Margin** : فضای خالی اطراف **border** می باشد که هیچ رنگ پس زمینه ای نمی گیرد و کاملاً **transparent** می باشد.

**Border** : حاشیه (**border**) دور تا دور فضای **padding** و محتوا را می تواند بپوشاند. با تعیین رنگ **box** رنگ حاشیه نیز به همان رنگ تغییر می کند.

**Padding** : فضای خالی اطراف محتوای اصلی می باشد. **Padding** تحت تاثیر رنگ پس زمینه **box** قرار می گیرد.

**Content** : محتوای اصلی **box** شامل متون و تصاویر می باشد.

برای اینکه بتوانید عرض و طول یک عنصر را طوری تعیین کنید که در تمامی مرورگرها به درستی دیده شوند، شما بایستی نحوه عملکرد **box model** را کاملاً بدانید.

### Width and Height of an Element

**نکته مهم** : زمانیکه خاصیت عرض و طول یک عنصر را توسط **CSS** تعیین می کنید، به این معنی است که طول و عرض **content area** (به شکل دقت نمایید) را تعیین می نمایید. برای محاسبه اندازه کامل یک

عنصر، بایستی اندازه ی `padding, border` و `margin` را نیز به مقدار طول و عرض اضافه کنید. در این صورت اندازه اصلی و واقعی عنصر را بدست می آورید. در مثال زیر مجموع عرض عنصر `300px` می باشد.

```
width:250px;
```

```
padding:10px;
```

```
border:5px solid gray;
```

```
margin:10px;
```

بیاید با هم محاسبه کنیم تا بفهمیم چطور اندازه کامل عنصر `300` می شود :

250 به عنوان عرض

20 به عنوان `padding` (10 سمت راست + 10 سمت چپ)

10 به عنوان `border` ( 5 سمت راست + 5 سمت چپ)

20 به عنوان `margin` (10 سمت راست + 10 سمت چپ)

که در مجموع می شود : 300

حال فرض کنید که فضایی که شما دارید `250px` می باشد. در نتیجه اندازه های شما بایستی به صورت ذیل باشد :

```
width:220px;
```

```
padding:10px;
```

```
border:5px solid gray;
```

```
margin:0px;
```

به صورت کلی محاسبه اندازه نهایی عرض عنصر بایستی با فرمول زیر محاسبه شود :

`Width + left padding + right padding + left border + right border + left margin + right margin`

و اندازه نهایی ارتفاع از فرمول ذیل محاسبه می شود :

`Height + top padding + bottom padding + top border + bottom border + top margin + bottom margin`

مواردی که بایستی برای سازگاری مرورگرها به آن واقف بود :

نسخه های IE8 و قبل از آن ، padding و border را در داخل width قرار می دهند. به این معنی که وقتی اندازه عرض را 200 تعیین می کنید. این مرورگرها فرض را بر این می گیرند که مجموع border و padding و width برابر 200 است. برای رفع این ناسازگاری در ابتدای صفحات html خود از تگ ذیل استفاده نمایید :

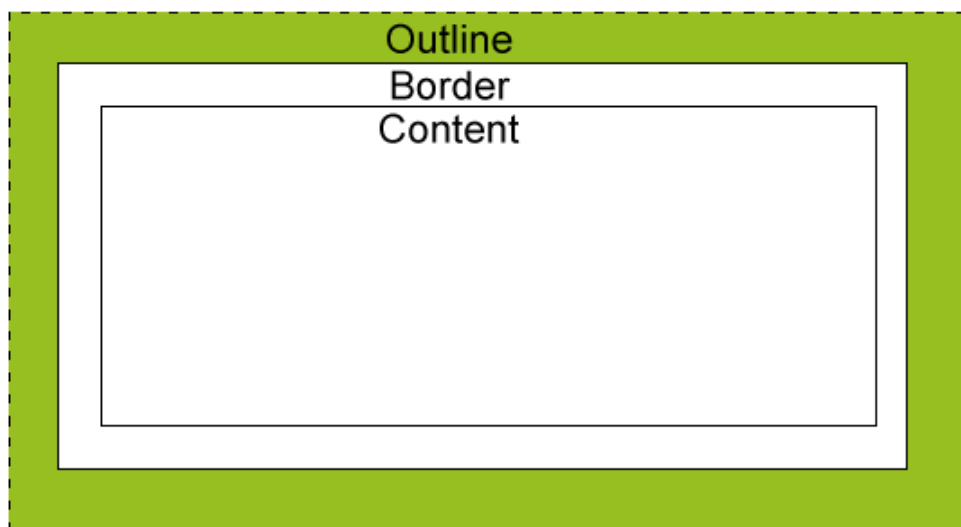
```
<!DOCTYPE html>
```

## CSS Outlines

Outline خطی است که در اطراف عنصر مورد نظر کشیده می شود. این خط در محیط بیرونی border کشیده می شود(چسبیده به border). از این خاصیت برای برجسته کردن بیشتر آن عنصر یا محتوای مورد نظر استفاده می شود.

می توانید مقادیر color style, width و برای یک خاصیت outline تعیین نمایید.

توجه داشته باشید که مختصات و اندازه های outline بر روی اندازه های مرتبط با خود عنصر اثری ندارد.



تمامی خواص مرتبط با **outline** در یک نگاه :

Values	Description	Property
<i>outline-color</i> <i>outline-style</i> <i>outline-width</i> inherit	Sets all the outline properties in one declaration	<a href="#">outline</a>
<i>color_name</i> <i>hex_number</i> <i>rgb_number</i> invert inherit	Sets the color of an outline	<a href="#">outline-color</a>
none dotted dashed solid double groove ridge inset outset inherit	Sets the style of an outline	<a href="#">outline-style</a>
thin medium thick <i>length</i> inherit	Sets the width of an outline	<a href="#">outline-width</a>

مثالهایی در این زمینه :

1- کشیدن یک خط (outline) اطراف یک عنصر

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p  
{  
border:1px solid red;
```

```
outline:green dotted thick;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p><b>Note:</b> IE8 supports the outline properties only if a !DOCTYPE is
```

```
specified.</p>
```

```
</body>
```

```
</html>
```

-2 تعیین Style یک outline

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p {border:1px solid red;}
```

```
p.dotted {outline-style:dotted;}
```

```
p.dashed {outline-style:dashed;}
```

```
p.solid {outline-style:solid;}
```

```
p.double {outline-style:double;}
```

```
p.groove {outline-style:groove;}
```

```
p.ridge {outline-style:ridge;}
```

```
p.inset {outline-style:inset;}
```

```
p.outset {outline-style:outset;}
```

```
</style>
```

```
</head>
```

```
<body>
```



```
<p class="dotted">A dotted outline</p>
```

```
<p class="dashed">A dashed outline</p>
```

```
<p class="solid">A solid outline</p>
```

```
<p class="double">A double outline</p>
```

```
<p class="groove">A groove outline</p>
```

```
<p class="ridge">A ridge outline</p>
```

```
<p class="inset">An inset outline</p>
```

```
<p class="outset">An outset outline</p>
```

**Note:** IE8 supports the outline properties only if a !DOCTYPE is specified.

```
</body>
```

```
</html>
```

3- تعیین رنگ یک outline

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p
```

```
{
```

```
border:1px solid red;
```

```
outline-style:dotted;
```

```
outline-color: 00ff00;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
<p><b>Note:</b> IE8 supports the outline properties only if a !DOCTYPE is specified.</p>
</body>
</html>
```

تعیین عرض یک outline

```
<!DOCTYPE html>
<html>
<head>
<style>
p.one
{
border:1px solid red;
outline-style:solid;
outline-width:thin;
}
p.two
{
border:1px solid red;
outline-style:dotted;
outline-width:3px;
}
</style>
</head>
<body>
```

```
<p class="one">This is some text in a paragraph.</p>
```

```
<p class="two">This is some text in a paragraph.</p>
```

```
<p><b>Note:</b> IE8 supports the outline properties only if a !DOCTYPE is  
specified.</p>
```

```
</body>
```

```
</html>
```

## CSS Margin

از طریق خاصیت **margin** فضای خالی اطراف عنصر خود را تعیین می کنید. از طریق **margin** می توان فضای اصراف یک عنصر را خالی کرد(در صورتیکه از **border** استفاده کرده باشیم، منظور فضای خارج از **border** می باشد)

**Margin** خاصیت **background-color** ندارد و کاملاً **transparent** می باشد.

می توان به طور مستقل مقادیر **top, right, bottom** و **left** را در **margin** تعیین نمود. می توان از دستور خلاصه **margin** برای تعیین تمامی مقادیر در یک دستور استفاده نمود.

مقادیری که می توان به **margin** داد عبارتند از :

**Auto** : خود مرورگر **margin** را تعیین می کند.

**Length** : می توان با مقیاس های **px, pt, cm and etc** اندازه **margin** را تعیین نمود. مقدار پیش فرض **0px** می باشد.

**%** : **margin** را بر اساس درصدی از اندازه عرض عنصر مربوطه تعیین می کند.

**Inherit** : مشخص می کند که **margin** بایستی از عنصر پدر ارث بری کند.

نکته : میتوان از مقادیر منفی به منظور **overlap** کردن محتوا استفاده کرد.

## تعیین جهات مختلف margin

مثال زیر نحوه تعیین اندازه جهات مختلف margin را به صورت جداگانه نشان می دهد:

```
<!DOCTYPE html>
<html>
<head>
<style>
p
{
background-color:yellow;
}
p.margin
{
margin-top:100px;
margin-bottom:100px;
margin-right:50px;
margin-left:50px;
}
</style>
</head>

<body>
<p>This is a paragraph with no specified margins.</p>
<p class="margin">This is a paragraph with specified margins.</p>
</body>

</html>
```

## استفاده از حالت خلاصه خاصیت margin

می توان برای خلاصه کردن کد، تمامی مقادیر را در یک خط و با استفاده از خاصیت margin تعیین نمود. به مثال زیر توجه نمایید :

```
margin:100px 50px;
```

به ترتیب مقادیری که در این حالت تعیین می کنید به این صورت دیده می شوند:

عدد اول به منزله **top margin** ، عدد دوم **right** عدد سوم : **bottom** و عدد چهارم **left** حال اگر فقط 3 عدد تعریف نمایید، عدد وسط به منزله **right** و **left** در نظر گرفته می شود.

```
margin:۲۰px ۵۰px ۷۰px;
```

اگر فقط 2 عدد تعریف نمایید، عدد اول به منزله **top** و **bottom** و عدد دوم به منزله **left** و **right** در نظر گرفته می شود.

```
margin:۲۰px ۵۰px;
```

و در انتها اگر فقط 1 عدد تعریف نمایید، آن عدد به منزله تمامی چهار جهت در نظر گرفته می شود.

```
margin:۲۰px;
```

## CSS Padding

با استفاده از خاصیت **padding** می توان، اندازه فضای بین **border** و محتوای یک عنصر را تعیین نمود. در اصل فضای اطراف محتوای یک عنصر را (در داخل **border**) خالی می کند. با تعریف رنگ پس زمینه عنصر فضای **padding** نیز از همان رنگ تاثیر می پذیرد.

تعیین چهار جهت بالا،پایین، راست و چپ **padding** به صورت مستقل قابل تغییر و تعیین است. و تمامی توضیحاتی که در خصوص تعیین مقادیر **margin** ارائه شد، برای **padding** نیز به همان صورت قابل اعمال است.

نحوه تعریف مقادیر **padding** به صورت جداگانه و یا به صورت خلاصه در یک خط در مثال زیر آورده شده است :

```
padding-top:25px;
```

```
padding-bottom:25px;
```

padding-right:50px;

padding-left:50px;

به صورت خلاصه در یک خط :

padding:25px 50px;

مثالی در این زمینه :

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p.ex1 {padding:2cm;}
```

```
p.ex2 {padding:0,5cm 3cm;}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p class="ex1">This text has equal padding on each side. The padding on each side is 2cm.</p>
```

```
<p class="ex2">This text has a top and bottom padding of 0,5cm and a left and right padding of 3cm.</p>
```

```
</body>
```

```
</html>
```

## Grouping Selectors

خیلی وقتها عناصری داریم که استایل های یکسانی دارند همانند مثال زیر :

```
h1  
{  
color:green;  
}  
h2  
{  
color:green;  
}  
p  
{  
color:green;  
}
```

برای اینکه نیازی به نوشتن این همه کد نباشد ، می توان انتخابگرها را (selectors) دسته بندی کرد. به طوریکه تمامی انتخابگرهایی که مقدار یکسان دارند را در یک خط نوشته و با کاما(,) آنها را از هم جدا کرد. مثال زیر همان کد بالاست ولی به صورت گروه بندی شده و خلاصه :

```
h1,h2,p  
{  
color:green;  
}
```

کد کامل آن به شرح ذیل است :

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
h1,h2,p  
{  
color:green;  
}  
</style>
```

```
</head>
```

```
<body>
```

```
<h1>Hello World!</h1>
```

```
<h2>Smaller heading!</h2>
```

```
<p>This is a paragraph.</p>
```

```
</body>
```

```
</html>
```

## Nesting Selectors

می‌توانید برای انتخابگری (selector) که در درون انتخابگر دیگری است، استایل تعیین کرد.

در مثال زیر یک استایل برای تمامی عناصر درون انتخابگر `<p>` تعیین شده است، همچنین یک استایل برای تمامی عناصری که به وسیله کلاسی با نام `marked` تعریف شده است، مشخص کرده ایم و در آخر، برای عناصری که در درون `<p>` تعریف شده اند و کلاس آنها `Marked` می‌باشد، استایل دیگری تعیین کرده ایم.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p
```

```
{
```

```
color:blue;
```

```
text-align:center;
```

```
}
```

```
.marked
```

```
{
```

```
background-color:red;
```

```
}
```

```
.marked p
```



```
{  
color:white;  
}  
</style>  
</head>  
<body>
```

```
<p>This paragraph has blue text, and is center aligned.</p>
```

```
<div class="marked">
```

```
This isn't a paragraph, therefore has no blue text nor alignment.
```

```
</div>
```

```
<div class="marked">
```

```
<p>p elements inside a "marked" classed element keeps the alignment style, but has a different text color.</p>
```

```
</div>
```

```
</body>
```

```
</html>
```

## CSS Dimensions

خواص مرتبط به ابعاد و اندازه در CSS

Values	Description	Property
auto <i>length</i> % inherit	Sets the height of an element	<a href="#">height</a>
none <i>length</i> % inherit	Sets the maximum height of an element	<a href="#">max-height</a>
none <i>length</i> % inherit	Sets the maximum width of an element	<a href="#">max-width</a>
<i>length</i> % inherit	Sets the minimum height of an element	<a href="#">min-height</a>
<i>length</i> % inherit	Sets the minimum width of an element	<a href="#">min-width</a>
auto <i>length</i> % inherit	Sets the width of an element	<a href="#">width</a>

یک مثال :

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
img.normal
```

```
{
```

```
height:auto;
```

```
}
```

```
img.big
```

```
{
```

```

height:120px;
}
p.ex
{
height:100px;
width:100px;
}
</style>
</head>

<body>
<br>

<p class="ex">The height and width of this paragraph is 100px.</p>
<p>This is a paragraph with no height and width specified.
This is a paragraph with no height and width specified.
This is a paragraph with no height and width specified.</p>
</body>
</html>

```

## CSS display and Visibility

با استفاده خاصیت **display** می توانیم تعیین کنیم که آیا یک عنصر نمایش داده شود یا خیر و یا اینکه چطور نمایش داده شود. همچنین با خاصیت **visibility** می توان تعیین کرد که یک عنصر قابل دیدن باشد یا خیر.

### Hiding an Element

یک عنصر را می توانیم با مقدار دادن **"none"** خاصیت **display** و یا مقدار دادن **"hidden"** به خاصیت **visibility** پنهان نماییم. هرچند بایستی توجه داشته باشید که این دو خاصیت نتایج متفاوتی خواهند داشت.

با پنهان کردن یک عنصر به وسیله خاصیت **visibility** آن عنصر مخفی می شود ولی فضایی که اشغال کرده هست را نگه می دارد یعنی به اندازه عنصر آن قسمت خالی می ماند در حالیکه خاصیت **display** علاوه بر

مخفی کردن عنصر فضای اشغال شده توسط عنصر را خالی می کند انگار که اصلا آن عنصر وجود خارجی نداشته است. برای روشن شدن موضوع به دو مثال زیر توجه نمایید :

مثال 1 : پنهان شدن از طریق خاصیت **visibility**

```
<!DOCTYPE html>
<html>
<head>
<style>
h1.hidden {visibility:hidden;}
</style>
</head>

<body>
<h1>This is a visible heading</h1>
<h1 class="hidden">This is a hidden heading</h1>
<p>Notice that the hidden heading still takes up space.</p>
</body>
</html>
```

مثال 2 : پنهان شدن از طریق خاصیت **display**

```
<!DOCTYPE html>
<html>
<head>
<style>
h1.hidden {visibility:hidden;}
</style>
</head>

<body>
```

```
<h1>This is a visible heading</h1>
<h1 class="hidden">This is a hidden heading</h1>
<p>Notice that the hidden heading still takes up space.</p>
</body>
</html>
```

همانطور که ملاحظه می کنید در مثال دوم فاصله ای بین خطوط وجود ندارد و عنصر علاوه بر مخفی شدن، فضایی را نیز اشغال نمی کند.

## CSS Display – Block and Inline Elements

یک **block element**، عنصری است که قبل و بعد از آن یک خط فاصله وجود دارد و تمامی عرض موجود صفحه را استفاده می کند همانند `<div>`، `<p>`، `<h1>`

یک **inline element** فقط مقدار عرضی که مورد نیاز دارد استفاده می کند و قبل و بعد از آن فاصله ای وجود ندارد. مثل: `<a>`، `<span>`

## Changing how an Element is Displayed

تغییر نحوه نمایش یک عنصر از حالت **inline** به **block** و بالعکس به منظور اینکه صفحه شما همانطور که می خواهید دیده شود یا از استانداردهای وب پیروی کنید، مورد استفاده قرار می گیرد.

در مثال زیر خاصیت **display** آیتم ها مورد نظر را به صورت **inline element** لیست می کند.

```
<!DOCTYPE html>
<html>
<head>
<style>
li{display:inline;}
</style>
</head>
<body>
<p>Display this link list as a horizontal menu:</p>
<ul>
```

```
<li><a href="/html/default.asp" target="_blank">HTML</a></li>
<li><a href="/css/default.asp" target="_blank">CSS</a></li>
<li><a href="/js/default.asp" target="_blank">JavaScript</a></li>
<li><a href="/xml/default.asp" target="_blank">XML</a></li>
</ul>
</body>
</html>
```

در حالیکه مثال زیر آیتم های مورد نظر را به صورت **block** نمایش می دهد

```
<!DOCTYPE html>
<html>
<head>
<style>
span
{
display:block;
}
</style>
</head>
<body>

<h2>Nirvana</h2>
<span>Record: MTV Unplugged in New York</span>
<span>Year: 1993</span>
<h2>Radiohead</h2>
<span>Record: OK Computer</span>
<span>Year: 1997</span>

</body>
</html>
```

نکته: تعیین خاصیت **display** یک عنصر فقط نحوه نمایش آن عنصر را تغییر می دهد و نمی تواند نوع عنصر را تغییر دهد. به این معنی که نمی توانید از عنصر با ماهیت **inline** مثل `<a>` در داخل عنصر دیگری با ماهیت **block** مثل `<span>` استفاده کنید و توقع داشته باشید چون خاصیت **display** عنصر **span** به **block** تغییر پیدا کرده است، پس `<a>` به صورت **block** نمایش داده شود. در این شرایط باز هم عنصر `<a>` به صورت **inline** و در ادامه خط قبلی خود دیده می شود.

## CSS Positioning

خاصیت **positioning** امکان تعیین مکان قرار گرفتن عنصر را به شما می دهد. همچنین می توانید عنصر را در پشت عنصر دیگری قرار داد و تعیین کرد در صورتیکه محتوای یک عنصر بسیار طولانی و بزرگ است چه باید کرد.

عناصر به وسیله خواص **top, bottom, left, right** تعیین مکان می شوند. با اینحال این خواص در صورتیکه خاصیت **position** تعیین مقدار نشده باشد، عمل نخواهند کرد. همچنین بسته به مقداری که به خاصیت **position** می دهیم، عملکرد آنها متفاوت خواهد بود.

چهار مقدار برای **position** وجود دارد که به شرح آنها می پردازیم:

## Static Positioning

به صورت پیش فرض تمامی عناصر **html** مقدار **static** دارند. عناصری که مقدار **static** دارند تحت تاثیر مقادیر **top, bottom, left, right** قرار نمی گیرند.

## Fixed Positioning

عنصری که مقدار **position** آن **fixed** می باشد، موقعیت مکانی آن مرتبط با پنجره مرورگر می باشد. به این معنی که در صورتیکه صفحه یا پنجره **scroll** بخورند، آن عنصر حرکت نمی کند. به مثال زیر توجه نمایید:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p.pos_fixed
```

```
{
```

```
position:fixed;
```

```
top:30px;
```

```
right:5px;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p class="pos_fixed">Some more text</p>
```

```
<p><b>Note:</b> IE7 and IE8 supports the fixed value only if a
```

```
!DOCTYPE is specified.</p>
```

```
<p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some  
text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some  
text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p>
```

```
</body>
```

```
</html>
```

نکته : IE7 و IE8 فقط در صورتیکه در ابتدای صفحه !DOCTYPE آورده شده باشد، از مقدار **fixed** پیشتیبانی می کنند.

توجه داشته باشید که در صورتیکه یک عنصر **fixed** را جابجا کنید، فضای قبلی که در اختیار آن بوده خالی می شود و می تواند در اختیار عنصر دیگری قرار گیرد.

## Relative Positioning

عنصری با مقدار **relative** مکانش به نسبت مکان معمول خودش تعیین می شود. به مثال زیر توجه نمایید :

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
h2.pos_left
```



```
{  
position:relative;  
left:-20px;  
}
```

```
h2.pos_right
```

```
{  
position:relative;  
left:20px;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h2>This is a heading with no position</h2>
```

```
<h2 class="pos_left">This heading is moved left according to its normal position</h2>
```

```
<h2 class="pos_right">This heading is moved right according to its normal position</h2>
```

```
<p>Relative positioning moves an element RELATIVE to its original position.</p>
```

```
<p>The style "left:-20px" subtracts 20 pixels from the element's original left position.</p>
```

```
<p>The style "left:20px" adds 20 pixels to the element's original left position.</p>
```

```
</body>
```

```
</html>
```

محتوای عنصری که **relative** هست می تونه بر روی عناصر دیگه حرکت کنه و یا **overlap** بشه. اما فضای رزرو شده عنصر همچنان در حالت نرمال رزرو می ماند. در صورتیکه که متنی را جابجا کنید و متن دیگری بعد از آن بنویسید، متن دوم نمی تواند دقیقا جای متن جابجا شده را بگیرد و آن فضا هنوز در اختیار متن اول باقی خواهد ماند.

مثال :

```
<!DOCTYPE html>
<html>
<head>
<style>
h2.pos_top
{
position:relative;
top:-50px;
}
</style>
</head>

<body>
<h2>This is a heading with no position</h2>

<h2 class="pos_top">This heading is moved upwards according to its normal position</h2>

<p><b>Note:</b> Even if the content of the relatively positioned element is moved, the reserved space
for the element is still preserved in the normal flow.</p>

</body>

</html>
```

## Absolute Positioning

مکان عنصری با مقدار **absolute** متناسب با اولین عنصر پدری است که مقدار **position** آن متفاوت از **static** تعیین می گردد. اگر هیچ عنصری با این مشخصات پیدا نشد، **block** ی که خود را با آن تطبیق می دهد **<html>** می باشد.

عناصری با مقدار **absolute** پس از تغییر جان، فضای قبلی را خالی می کنند و می توانند با دیگر عناصر **overlap** شوند. به مثال زیر توجه فرمایید :

```
<!DOCTYPE html>
<html>
<head>
<style>
h2
{
position:absolute;
left:100px;
top:150px;
}
</style>
</head>

<body>
<h2>This is a heading with an absolute position</h2>

<p>With absolute positioning, an element can be placed anywhere on a page. The heading below is placed 100px from the left of the page and 150px from the top of the page.</p>

</body>

</html>
```

## Overlapping Elements

هنگامی که عناصر خارج از حالت عادی قرار گرفتن در صفحه، مکان دهی شوند می توانند با سایر عناصر **overlap** شوند. خاصیت **z-index** ترتیب قرار گرفتن یک عنصر را مشخص می کند. به این معنی که کدام عنصر بایستی در جلو یا پشت عناصر دیگر قرار گیرد. که این خاصیت می تواند مقدار مثبت یا منفی داشته باشد.

به مثال زیر توجه فرمایید :

```
<!DOCTYPE html>
<html>
<head>
<style>
img
{
position:absolute;
left:0px;
top:0px;
z-index:-1;
}
</style>
</head>

<body>
<h1>This is a heading</h1>

<p>Because the image has a z-index of -1, it will be placed behind the text.</p>
</body>
</html>
```

نکته : در صورتیکه دو عنصر بدون مشخص کردن خاصیت **z-index** بر روی هم قرار گیرند (**overlap**)، عنصری که آخر معرفی می شود، در جلوی بقیه قرار می گیرد.

تمامی خواص مکان دهی در یک نگاه :

Values	Description	Property
auto <i>length</i> % inherit	Sets the bottom margin edge for a positioned box	<a href="#">bottom</a>
<i>shape</i> auto inherit	Clips an absolutely positioned element	<a href="#">clip</a>
<i>url</i> auto crosshair default pointer move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize text wait help	Specifies the type of cursor to be displayed	<a href="#">cursor</a>
auto <i>length</i> % inherit	Sets the left margin edge for a positioned box	<a href="#">left</a>
auto hidden scroll visible inherit	Specifies what happens if content overflows an element's box	<a href="#">overflow</a>
absolute fixed relative static inherit	Specifies the type of positioning for an element	<a href="#">position</a>
auto <i>length</i> % inherit	Sets the right margin edge for a positioned box	<a href="#">right</a>
auto <i>length</i> % inherit	Sets the top margin edge for a positioned box	<a href="#">top</a>
<i>number</i> auto inherit	Sets the stack order of an element	<a href="#">z-index</a>

