



آموزش

CSS

[www.tahlildadeh.com](http://www.tahlildadeh.com)

نویسنده: مهندس افشین رفوآ

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِیْمِ

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

کتاب آموزشی CSS

آموزشگاه تحلیل داده  
مؤلف مهندس اشین رفوآده



تقدیم به همه جویندگان علم که توان و امکان شرکت در کلاس های حضوری ما را ندارند.

## فهرست

9	مقدمه
9	توجه
10	چگونگی استفاده از CSS
10	CSS چیست؟
10	نمایش CSS
12	دستور نحوی CSS
13	توضیحات یا comment ها در CSS
14	انتخاب گرهای CSS
14	انتخاب گر شناسه (id selector)
15	انتخاب گر کلاس (class selector)
15	ادغام گروه ها / گروه بندی انتخاب گرها (grouping selectors)
16	Style Sheet ها
16	Style Sheet خارجی (external)
17	Style Sheet داخلی (internal)
17	Style های Inline
17	Style Sheet های چندگانه
19	پس زمینه
19	رنگ پس زمینه
20	تصویر پس زمینه
21	تصویر پس زمینه - تکرار تصویر پس زمینه با نمای افقی و عمودی
21	آموزش خاصیت های Image - Set position و no-repeat
22	آموزش خاصیت Background - Shorthand
24	متن در CSS
24	رنگ متن
24	تراز بندی متن (text alignment)
25	اعمال جلوه های بصری به متن (Text Decoration)
26	تنظیم تو رفتگی متن (TEXT INDENTATION)

28	..... فونت در CSS
28	..... خواص فونت در CSS
28	..... خانواده‌های فونت (font family) در CSS
29	..... خانواده ی فونت (font family)
30	..... سبک فونت ها (Font style)
30	..... خصوصیت نام برده سه مقدار متفاوت دارد
30	..... اندازه ی فونت (Font Size)
31	..... اندازه ی مطلق
31	..... تنظیم و انتخاب اندازه بر حسب پیکسل (با پیکسل)
31	..... تنظیم اندازه ی فونت با Em
32	..... بکاربردن ترکیبی از Percent و Em
33	..... All CSS Font Properties
33	..... آموزش Link
34	..... چهار حالت مختلف برای لینک در زیر فهرست شده
35	..... سبک های معمول لینک
35	..... رنگ پس زمینه (background color)
36	..... لیست در CSS
36	..... انواع لیست ها
36	..... نشانگرهای مختلف آیتم های لیست (list item marker)
37	..... انتخاب یک تصویر به عنوان نشانگر آیتم مورد نظر در لیست
38	..... آموزش خاصیت List - Shorthand
40	..... جدول در CSS
41	..... خاصیت border-collapse
41	..... طول و عرض جدول
42	..... تراز بندی متن به صورت افقی (Horizontal text alignment)
42	..... تراز بندی عمودی متن (vertical text alignment)
43	..... استفاده از خاصیت padding در جدول
43	..... تنظیم رنگ جدول

43	آموزش Box در Css
44	پهنا و ارتفاع عنصر
45	مبحث سازگاری مرورگرها
46	خاصیت Border در CSS
46	مقادیر Border-style
47	ضخامت حاشیه (Border width)
47	رنگ خط حاشیه (Border color)
48	تعریف حاشیه به صورت مجزا برای هر طرف در CSS
50	خاصیت Border-Shorthand
52	آموزش Outline
52	Outline / خط پیرامون در CSS
54	آموزش Margin
55	مقادیر ممکن (Possible Values)
56	مقدار دهی margin هر چهار جهت در یک تعریف
57	آموزش Padding
57	Padding (فاصله / حاشیه ی درونی عناصر)
58	مقادیر ممکن
58	تنظیم فاصله ی درونی عنصر به صورت تکی برای هر جهت
58	تعیین با استفاده از خاصیت چند مقداری padding
59	خاصیت padding می تواند از یک تا چهار مقدار داشته باشد
60	آموزش خاصیت display
61	خاصیت Display در CSS – عناصر Block و Inline
61	اصلاح نحوه ی نمایش یک عنصر
62	تعیین مختصات و محل قرار گیری عناصر در صفحه
63	تعیین موقعیت / موقعیت دهی به صورت ایستا (static positioning)
63	تعیین موقعیت قرار گیری عنصر به صورت دقیق و ثابت (Fixed positioning)
64	تعیین موقعیت یا موقعیت دهی به صورت نسبی (relative positioning)
64	موقعیت دهی به صورت مطلق (absolute positioning)

65	..... تمامی خاصیت های موقعیت دهی در CSS
67	..... آموزش Float
68	..... عناصر چگونه شناور می شوند ؟
69	..... لغو حالت شناور عناصر (خاصیت float) با استفاده از خاصیت Clear
69	..... تمامی خصوصیت های مربوط به float یا شناور سازی در CSS
70	..... شبه عناصر در CSS
70	..... شبه عناصر در CSS چه کاربردی دارند ؟
70	..... ترکیب نحوی (syntax)
71	..... شبه المان first-line
72	..... شبه المان first-letter
73	..... شبه کلاس ها و کلاس های CSS
73	..... ترکیب چندین شبه عنصر (Multiple pseudo-element)
74	..... CSS – شبه المان before
74	..... CSS – شبه عنصر after
74	..... شبه عنصر selection و کاربرد آن در CSS
75	..... سیتنکس pseudo-elements
76	..... first-line Pseudo-element
76	..... first-letter Pseudo-element
78	..... Pseudo-elements و کلاس های CSS
78	..... چندگانه Pseudo-elements
78	..... before Pseudo-element
78	..... Pseudo-elements
79	..... آموزش Navigatuion
79	..... نوار پیمایش – لیست لینک ها
80	..... نوار پیمایش عمودی
80	..... نوار پیمایش افقی
81	..... Inline های List Item
81	..... Floating های list item

82	آموزش گالری عکس
82	البوم تصویر
82	ایجاد تصاویر شفاف در CSS
83	– شفافیت تصویر – Hover Effect
85	Image Sprite چیست؟
85	Image Sprites – مثالی ساده
86	Image Sprites – ایجاد لیست پیمایشی (navigation list)
89	انواع رسانه ها در CSS
90	دستور @media
91	گزینه‌گر های صفت
91	گزینه‌گر [ صفت ] CSS
92	گزینه‌گر [ صفت = مقدار ] CSS
92	گزینه‌گر [ مقدار = ~ خصیصه ] CSS
92	گزینه‌گر [ خصیصه = مقدار ] در CSS
93	گزینه‌گر [ خصیصه ^ مقدار ] در CSS
93	گزینه‌گر [ attribute\$=value ] در CSS
94	گزینه‌گر [ attribute*=value ] و کاربرد آن در CSS
94	سبک دهی به فرم
94	سلکتورهای Grouping
95	تراز کردن عنصرهای block
95	نمونه هایی از عنصر block
96	تراز کردن در مرکز با استفاده از پراپرتی margin
96	تراز چپ و راست با استفاده از پراپرتی Position
97	مسئله سازگاری Crossbrowser
97	تراز چپ و راست با استفاده از پراپرتی float
98	مسئله سازگاری Crossbrowser



## زکات علم نشر آن است. حضرت علی

موسسه آموزشی تحلیل داده ، با حضور جمعی از متخصصین مجرب در زمینه برنامه نویسی در نظر دارد، مطالب آموزشی خود را در قالب کتاب های آموزشی و فیلم ، به صورت رایگان در دسترس عموم قرار دهد تا حتی آن دسته از عزیزانی که بنا به دلایل مالی، مسافت جغرافیایی و یا نداشتن وقت کافی ، امکان شرکت در دوره های حضوری برای آنها میسر نیست، از یادگیری بی بهره نمانند.

علاوه بر این علاقه مندان می توانند ، با ثبت نام در انجمن سایت تحلیل داده، سوالات خود را مطرح نموده و مدرسین آموزشگاه و اعضای انجمن در اسرع وقت، پاسخ های خود را، حتی الامکان به صورت فیلم، در دسترس عموم قرار دهند.

لذا از کلیه فعالان در این زمینه دعوت می شود، در این حرکت جمعی در کنار ما باشند و با حضور فعال خود در انجمن، گام موثری در بهبود سطح علمی جوانان کشور عزیزمان، ایران بردارند.

آموزشگاه تحلیکر داده

توجه

برای دانلود سورس کد مثال های کتاب ، به بخش مقالات Html در آدرس [www.tahlildadeh.com](http://www.tahlildadeh.com) مراجعه فرمایید.

## چگونگی استفاده از CSS

در این سلسله از مقالات، شما چگونگی استفاده از CSS جهت کنترل **style** و **layout** چندین صفحه وب بطور همزمان را یاد خواهید گرفت.

برای یادگیری **CSS**، شما باید درکی پایه ای از مفاهیم **HTML** و **XHTML** داشته باشید.

یک نمونه از CSS

```
body {  
  background-color :#d0e4fe;  
}  
h1 {  
  color :orange;  
  text-align :center;  
}  
p {  
  font-family: "Times New Roman";  
  font-size :20px;  
}
```

CSS چیست؟

CSS مخفف **Cascading Style Sheets** است.

**style** ها چگونگی نمایش عناصر **HTML** را تعریف می کنند.

**Style** ها برای حل یک مشکل به **HTML 4.0** اضافه شدند.

**External Style Sheets** می تواند در در انجام کار صرفه جویی کند.

**External Style Sheets** در فایل های **CSS** ذخیره می شوند.

نمایش CSS

سند **HTML** را می توان با **style** های مختلف نمایش داد..

**Style** ها مشکل بزرگی را حل کردند.

HTML هرگز به قصد در برگرفتن tag برای فرمت کردن یک سند ایجاد نشد.

HTML هرگز به قصد تعریف محتوای یک سند ایجاد نشد.

وقتی tagها و attribute های رنگ به HTML 3.2 اضافه شد، کابوسی برای برنامه نویسان بود. develop کردن وب سایت های بزرگ، که فونت ها و اطلاعات رنگ در هر صفحه مجزا بودند، فرآیندی طولانی و پرهزینه ای شد.

برای حل این مشکل، کنسرسیوم World Wide Web، تصمیم گرفت CSS را ایجاد کند.

در HTML 4.0، می توان کل قالب بندی را از سند HTML حذف کرد، و در فایل CSS جداگانه ای ذخیره کرد.

امروزه، همه مرورگرها CSS را ساپورت می کنند.

CSS، در انجام مقدار زیادی کار صرفه جویی می کند

CSS، تعریف می کند عناصر HTML چگونه نمایش داده می شوند.

معمولاً Style ها در فایل های css ذخیره می شوند. style sheet های خارجی شما راتنها با ویرایش یک فایل

قادر به تغییر ظاهر و layout همه صفحات موجود در یک وبسایت می کند.

مثال

فایل HTML زیر، لینکی به یک style sheet خارجی با تگ است

```
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="ex1.css" />
  </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>
```

این، فایل style sheet است

```
body {
  background-color: yellow;
```

```

}
h1 {
  font-size: 36pt;
}
h2 {
  color: blue;
}
p {
  margin-left: 50px;
}

```

## دستور نحوی CSS

یک **rule set** (یک حکم یا دستور است که نحوه ی پردازش المان ها را روی یک صفحه ی HTML برای مرورگر مورد نظر تعریف می کند) همان طور که در تصویر زیر مشاهده می کنید متشکل است از یک **selector** (انتخاب

گر) و **declaration block** (بلوک یا ساختمان اعلان) است .



**Selector** به آن عنصر **HTML** ی اشاره دارد که طراح وب مایل است سبک دهی کند .

یک **declaration block** دربردارنده ی یک یا چند تعریف (**declaration**) است که توسط نقطه ویرگول (**semicolon**) از هم جدا می شوند .

هر تعریف نیز خود شامل یک اسم خصوصیت (**property name**) و همچنین یک مقدار (**value**) است که توسط یک دو نقطه (**colon**) از هم جدا سازی می شوند .

## یک مثال CSS

همیشه یک تعریف CSS به نقطه ویرگول ختم می شود، گروه های تعریفی (declaration group) نیز داخل کاراکترهای {} محصور می شوند

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

به منظور افزایش قابلیت خوانایی CSS، شما می توانید هر بخش یک تعریف را روی خط مجزا قرار دهید . مثال

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

توضیحات یا comment ها در CSS

comment به منظور ارائه ی توضیح درباره ی کد مورد نظر استفاده می شود ، در صورت لزوم همچنین به شما کمک می کند متن برنامه (source code) را در آینده اصلاح کنید . توضیحات (comments) کاملاً توسط مرورگرها نادیده گرفته می شوند .

یک توضیح CSS با علامت /\* آغاز گردیده و با کاراکتر /\* پایان می یابد . توضیحات نیز ممکن است چندین خط را به خود تخصیص دهد.

مثال

```
p {  
  color: red;  
  /* This is a single-line comment */  
  text-align: center;  
}  
/* This is  
a multi-line  
comment */
```

## انتخاب گره‌های CSS

انتخاب گره‌های CSS به شما این امکان را می‌دهد که المان‌های HTML را انتخاب کرده و متناسب با نیاز خود آن‌ها را تنظیم و دستکاری (**manipulate**) کنید .

با استفاده از انتخاب گره‌های CSS می‌توان المان‌های دلخواه و مورد نظر HTML را بر اساس شناسه (ID)، کلاس، نوع (**type**)، خصیصه (**attribute**) پیدا (و یا انتخاب و گزینش) کرد .

### انتخاب گر المان (**the element selector**)

انتخاب گر عنصر در واقع المان‌ها را بر اساس اسم المان مربوطه انتخاب می‌کند . می‌توانید کلیه ی المان‌های **<p>** را روی صفحه درست بدین ترتیب انتخاب کنید (کلیه ی عناصر **<p>** وسط چین (**align-center**) شده و متن آن با رنگ قرمز نشان داده می‌شود) .

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

### انتخاب گر شناسه (**id selector**)

**id selector** از خصیصه ی شناسه (**id attribute**) HTML برای انتخاب یک المان معین بهره می‌گیرد .

یک شناسه داخل صفحه باید منحصر بفرد باشد، بنابراین انتخاب گر شناسه زمانی بکار گرفته می‌شود که بخواهید یک عنصر تک و منحصر بفرد را انتخاب کنید .

جهت انتخاب یک المان با شناسه ی ویژه (مشخص)، باید یک کاراکتر **hash (#)** را تایپ کرده و بدنبال این کاراکتر شناسه ی المان را قرار دهید .

قانون سبک دهی (**style rule**) زیر با شناسه ی "**para1**" به المان HTML اضافه می‌شوند

```
#para1 {  
  text-align: center;
```

```
color: red;
}
```

توجه هیچگاه اسم شناسه (ID name) را با یک عدد آغاز نکنید .

### انتخاب گر کلاس (class selector)

انتخاب گر کلاس المان ها را با یک خصیصه ی کلاس / **class attribute** ویژه آغاز می کند .

به منظور انتخاب المان ها با کلاسی معین، یک کاراکتر نقطه و به دنبال آن اسم کلاس را تایپ کنید

در مثال زیر، کلیه ی عناصر **HTML** با کلاس "center"، وسط چین (**align-center**) خواهند بود

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

همچنین می توان مشخص کرد که تنها عناصر خاصی از **HTML** دست خوش تغییر یک کلاس معین قرار گیرند (به عبارتی دیگر می توانید تعیین کنید که فقط المان های مدنظر شما توسط یک کلاس کنترل شده و تحت تاثیر آن قرار گیرند)

در مثالی که زیر ارائه شده، تمامی المان های **<p>** با کلاس "center" وسط چین (**align-center**) می شوند.

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

توجه از آغاز کردن اسم کلاس با عدد اکیداً خودداری کنید .

### ادغام گروه ها / گروه بندی انتخاب گرها (grouping selectors)

چنانچه المان هایی با تعاریف سبک (**style definition**) مشابه به مثال ذیل دارید، در آن صورت می توانید با

گروه بندی کدها آنها را سامان دهی کرده و در عین حال اندازه ی کدهای ذکر شده را به حداقل برسانند.

```

h1 {
  text-align: center;
  color: red;
}
h2 {
  text-align: center;
  color: red;
}
p {
  text-align: center;
  color: red;
}

```

برای این منظور کافی است انتخاب گرها را با استفاده از ویرگول از هم جدا ساخت .

نمونه ی زیر در حقیقت همان تکه کد فوق است که در آن سه گروه مزبور با هم در یک گروه ادغام شده اند و هر گروه با بهره گیری از ویرگول از گروه دیگر متمایز شده است

```

h1,h2,p{
  text-align: center;
  color: red;
}

```



Style Sheet ها

Style Sheet خارجی (external)

**Style Sheet** خارجی، هنگامی که **style** به صفحات زیادی اعمال می شود، مناسب است. با یک **Style Sheet**

خارجی، می توانید ظاهر کل یک سایت را با تغییر دادن یک فایل عوض کنید. هر صفحه باید با استفاده از تگ **<link>** به **style sheet** لینک شود. تگ **<link>**، درون بخش **head** قرار می گیرد.

```

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

```

**Style Sheet** خارجی، را می توان در یک **text editor** نوشت. این فایل نباید حاوی هیچ تگ **html** باشد.

**style sheet** شما باید با پسوند **.css** ذخیره شود. نمونه ای از یک **style sheet** در زیر نشان داده شده

است.

```

hr {color:sienna;}
p {margin-left:20px;}
body {background-image:url("images/back40.gif");}

```

فضاهای بین مقدار **صفت** و واحدها را خالی نگذارید!



## Style Sheet داخلی (internal)

**Style Sheet** داخلی باید هنگامی مورد استفاده قرار گیرد که یک سند واحد، **style** منحصر به فردی دارد. **Style** های داخلی با استفاده از تگ **<style>** در بخش **head** صفحه **HTML** تعریف می شود، مانند مثال زیر

```
<head>
<style type="text/css">
hr {
  color: sienna;
}
p {
  margin-left: 20px;
}
body {
  background-image: url("images/back40.gif");
}
</style>
</head>
```

### Inline های Style

استایل **inline**، با ترکیب کردن **content** با **presentation**، بسیاری از مزایای **style sheet** ها را از دست می دهد. از این متد باید به ندرت استفاده کرد.

برای استفاده از استایل های **inline**، باید از اتریبیوت استایل در تگ مربوط استفاده کرد. اتریبیوت استایل ممکن است هر نوع خصوصیت **CSS** را در بر گیرد. این مثال چگونگی تغییر دادن رنگ و حاشیه چپ یک پاراگراف را نشان می دهد

```
<p style="color:sienna;margin-left:20px">This is a paragraph</p>
```

### Style Sheet های چندگانه

اگر بعضی از صفت ها برای یک **selector** در **style sheet** های مختلف تعیین شده باشد، **value** ها از **style sheet** معینی به ارث برده خواهند شد.

مثلاً، یک **style sheet** خارجی دارای صفت های زیر برای سلکتور **h3** است

```
h3
{
Color:red;
```

```
text-align:left;
font-size:8pt;
}
```

و یک **style sheet** داخلی دارای صفت های زیر برای سلکتور **h3** است

```
h3
{
text-align:right;
font-size:20pt;
}
```

اگر صفحه ای با **style sheet** داخلی به یک **style sheet** خارجی لینک شود، صفت های **h3** اینگونه خواهد بود

```
Color:red;
text-align:right;
font-size:20pt;
```

رنگ، از **style sheet** خارجی ارث می برد و **text-alignment** و اندازه فونت با **style sheet** داخلی جایگزین می شود.

**Style** های چندگانه به یک استایل **cascade** می شوند .

**Style** ها را می توان به طریق زیر معین کرد

درون یک عنصر **HTML**

درون بخش **head** صفحه **HTML**

در یک فایل **CSS** خارجی

نکته حتی **style sheet** های خارجی چندگانه را نیز می توان درون یک سند **HTML** واحد **reference** کرد.

ترتیب **cascade** کردن

هنگامی که بیش از یک **style** که برای عنصر **HTML** تعیین شده، وجود داشته باشد، از کدام **style** استفاده می شود؟

معمولاً می توان گفت که همه **style** ها، با قوانین زیر به یک **style sheet** مجازی **cascade** می شود، و قانون چهارم بالاترین اولویت را دارد

پیش فرض مرورگر

**style sheet** خارجی

**style sheet** داخلی (در بخش **head**)

استایل **inline** (درون عنصر **HTML**)

بنابراین، یک استایل **inline**، بالاترین اولویت را دارد.

نکته اگر لینک به **style sheet** خارجی بعد از **style sheet** داخلی در **HTML <head>** داخلی قرار داده شود، **style sheet** خارجی، **style sheet** داخلی را **override** می کند.

پس زمینه

از خصوصیت های (**properties**) پس زمینه ی **CSS** به منظور تعریف جلوه های (**effect**) پس زمینه ی المان ها استفاده می شود.

خصوصیت های **CSS** که ویژه ی جلوه های پس زمینه بکار برده می شوند به ترتیب زیر می باشد

**background-color**

**background-image**

**background-repeat**

**background-attachment**

**background-position**

رنگ پس زمینه

خصوصیت (**property**) **background-color** در واقع رنگ پس زمینه ی یک المان را مشخص می کند .

رنگ پس زمینه ی یک صفحه ی متعارف به صورت زیر تعریف (set) می شود

مثال

```
body {  
  background-color: #b0c4de;  
}
```

در CSS، رنگ یک صفحه اغلب توسط مقادیر زیر تعیین می شود

یک مقدار HEX (شانزده شانزدهی) – مثل "#ff0000"

یک مقدار RGB (سرنام RED GREEN BLUE) – مانند rgb("255","0","0")

اسم رنگ دلخواه (color name) – مانند "red"

در مثال زیر، <h1>، <p>، و المان های <div> رنگ های پس زمینه ی متفاوتی دارند

مثال

```
h1 {  
  background-color: #6495ed;  
}  
p {  
  background-color: #e0ffff;  
}  
div {  
  background-color: #b0c4de;  
}
```

تصویر پس زمینه

خصوصیت background-image در حقیقت تصویری برای استفاده به عنوان پس زمینه ی یک المان

تعریف می کند .

به صورت پیش فرض، تصویر تکرار می شود تا بدین وسیله بتواند کل عنصر (element) مربوطه را تحت

پوشش قرار دهد .

تصویر پس زمینه به صورت زیر انتخاب می شود

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

در زیر مثالی از ترکیبی نامناسب از متن و تصویر پس زمینه ارائه شده است، به گونه ای که متن تقریباً ناخوانا است

```
body {  
  background-image: url("bgdesert.jpg");  
}
```

تصویر پس زمینه – تکرار تصویر پس زمینه با نمای افقی و عمودی

به صورت پیش فرض، خصوصیت **background-image** یک تصویر را با هر دو نما یا صورت افقی و عمودی تکرار می کند .

برخی از این تصاویر باید تنها به صورت افقی یا عمودی تکرار شوند، در غیر این صورت به گونه ای عجیب و نامتعارف به نظر می رسد، مشابه نمونه ی زیر

```
body {  
  background-image: url("gradient_bg.png");  
}
```

در صورتی که اگر تصویر مورد نظر تنها افقی تکرار شود (**repeat-x**)، پس زمینه ظاهر بهتری پیدا خواهد کرد

```
body {  
  background-image: url("gradient_bg.png");  
  background-repeat: repeat-x;  
}
```

**نکته** جهت تکرار تصویری به صورت عمودی از **background-repeat: repeat-y;** بهره بگیرید .

آموزش خاصیت های **Image - Set position** و **no-repeat**

**نکته** حین استفاده از یک تصویر پس زمینه، سعی کنید از آن تصویری استفاده کنید که قالب متن را برهم نمی زند .

به منظور نمایش تصویری برای تنها یکبار کافی است از خاصیت **background-repeat** بهره بگیرید (این خاصیت تعریف می کند که تصویر مربوطه تنها یک مرتبه تکرار شود)

مثال

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

در مثال فوق، تصویر پس زمینه درست در مکانی یکسان با متن نمایش داده شده است .

حال می خواهیم موقعیت (**position**) تصویر را تغییر دهیم تا از این طریق تصویر مذکور شکل متن مورد نظر را برهم نزند .

موقعیت تصویر توسط خاصیت **background-position** تعیین می گردد

مثال

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



آموزش خاصیت Shorthand - Background

همان طور که در مثال های فوق مشاهده می کنید، خاصیت های فراوانی وجود دارند که حین پرداختن به پس زمینه و مدیریت آن باید در نظر گرفت .

به منظور کاهش حجم کد، این امکان وجود دارد که تمامی خصوصیت ها (**properties**) را در تنها یک خصوصیت تعریف کرد . خاصیت مذکور همچنین **shorthand** (خصوصیت مختصر نویسی) اطلاق می گردد .

خاصیت مختصر نویسی (**shorthand property**) ویژه ی پس زمینه همان "**background**" می باشد

مثال

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

به خاطر داشته باشید که هنگام استفاده از خصوصیت مختصر نویسی، ترتیب مقادیر **property** ها به صورت زیر خواهد بود

**background-color**

background-image

background-repeat

background-attachment

background-position

هیچ اهمیتی ندارد اگر یکی از مقادیر **property** مفقود بوده یا در حال حاضر موجود نمی باشد، به خصوص تا زمانی که دیگر مقادیر موجود به ترتیبی که فوق ذکر شده سامان دهی شده باشند .

این مثال از نوع یا سطح پیشرفته ی **CSS** بهره می گیرد . تصویر همراه با بقیه ی صفحه، بالا یا پایین (**scroll**) نمی رود .

تمامی خصوصیت های پس زمینه ی **CSS** در جدول زیر تشریح شده

توصیف کاربرد	خاصیت (property)
کلیه ی خصوصیت های پس زمینه را طی (در) تنها یک تعریف (declaration) مشخص می کند	background
تعیین می کند آیا یک تصویر پس زمینه ثابت بوده و یا همراه با بقیه ی صفحه حرکت داشته و جا به جا شود (scroll)	background-attachment
رنگ پس زمینه ی یک عنصر را انتخاب می کند	background-color
تصویر پس زمینه ی یک المان را تعیین می کند	background-image
مکان شروع جای گیری یک تصویر پس زمینه را مشخص می کند	background-position
نحوه ی نمایش مجدد یا تکرار یک تصویر پس زمینه را تعیین می کند	background-repeat

## متن در CSS

### رنگ متن

خاصیت رنگ یا **color property** همان طور که از اسم آن پیدا است، ویژه ی انتخاب رنگ نوشته یا متن مورد نظر مورد استفاده قرار می گیرد .

در **CSS**، رنگ غالباً توسط مقادیر فهرست شده در زیر تعیین می شود

مقدار **HEX** (شانزده شانزدهی) – مثل "#ff0000"

مقدار **RGB** (RED BLUE GREEN) – مانند "rgb(255,0,0)"

یک اسم رنگ (color name) – مانند "red"

رنگ پیش فرض برای یک صفحه در بدنه ی انتخاب گر (**body selector**) تعریف می گردد .

مثال

```
body {  
  color: blue;  
}  
h1 {  
  color: #00ff00;  
}  
h2 {  
  color: rgb(255,0,0)  
}
```

**نکته** برای ایجاد **CSS** ی که با استانداردهای **W3C** مطابقت داشته باشد چنانچه خاصیت رنگ ( **color**

**property** ) را تعریف کردید، در آن صورت لازم است خاصیت **background-color** را نیز تعریف کنید .

تراز بندی متن (**text alignment**)

خصوصیت **text-align** در واقع یک متن را به صورت افقی تراز می کند (جهت چیدمان نوشته را در صفحه با

نمای افقی تنظیم می کند) .

نوشته می تواند در وسط صفحه قرار گیرد (وسط چین شود)، همچنین در سمت چپ یا راست قرار داده شود و یا

حتی از هر دو سمت راست و چپ تراز (**justify**) شود .



زمانی که خصوصیت **text-align** روی "justify" تنظیم می شود، هر خط تا آنجایی کشیده شده یا بسط داده می شود که از نظر پهنا (**width**) با هم برابر شوند، حاشیه های سمت چپ و راست نیز درست مشابه مجله یا روزنامه هر دو کاملاً باید راست و مستقیم باشند .

مثال

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

اعمال جلوه های بصری به متن (Text Decoration)

خاصیت **text-decoration** به منظور اعمال جلوه های تصویری به متن یا حذف آن ها از نوشته ی مورد نظر بکار گرفته می شوند .

خاصیت **text-decoration** عمدتاً جهت حذف زیرخط (**underline**) از لینک ها برای مقاصد طراحی مورد استفاده قرار می گیرد

مثال

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

از آن همچنین برای افزودن جلوه های بصری (**decoration**) به نوشته نیز استفاده می شود

مثال

```
h1 {
  text-decoration: overline;
}
h2 {
  text-decoration: line-through;
}
h3 {
  text-decoration: underline;
}
```

**توجه** توصیه می شود از کشیدن خط زیر متنی که لینک نیست خودداری کنید، زیرا با انجام این کار کاربر دچار سردرگمی می شود .

### تبدیل نوع متن از حروف کوچک به بزرگ و بالعکس (Text Transformation)

خاصیت **text-transform** به منظور تعیین کوچکی و بزرگی حروف (مدیریت اندازه ی حروف) در متن استفاده می شود .

از آن می توان برای تبدیل هر نوشته ی به حروف بزرگ یا کوچک استفاده کرد و یا حتی به کمک آن حروف اول هر کلمه را به حرف بزرگ تبدیل کرد

مثال

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



### تنظیم تو رفتگی متن (TEXT INDENTATION)

خاصیت **text-indent** را می توان به منظور تنظیم و مدیریت میزان تو رفتگی اولین خط متن مورد استفاده قرار داد (میزان تو رفتگی و نقطه شروع اولین خط از یک پاراگراف نوشته را اداره می کند).

مثال

```
p {
  text-indent: 50px;
}
```

کلیده ی خاصیت های مربوط به متن در **CSS** در جدول زیر با ذکر عملکرد آن ها فهرست شده

شرح عملکرد	خصوصیت (PROPERTY)
رنگ نوشته را تعیین می کند	color
جهت نوشتن / قرارگیری متن را تعریف می کند	direction
فاصله ی بین کاراکترهای یک متن را کاهش یا افزایش می دهد	Letter-spacing
فاصله ی بین خطوط یک پاراگراف توسط این این خاصیت مشخص می شود	Line-height
نحوه ی چیدمان (افقی) متن را مشخص می کند	Text-align
افزودن جلوه های تصویری به نوشته را امکان پذیر می سازد (به شما اجازه می دهد دکر یا تزیینات متن را کنترل کنید)	Text-decoration
نحوه ی چیدمان و تراز بندی یک متن را تعیین می کند	Text-align
جلوه یا ایفکت سایه که به نوشته اضافه می شود را تعیین می کند	Text-shadow
کوچک و بزرگی حروف را تعیین می کند	Text-transform
به همراه خاصیت direction به منظور تعیین این که آیا متن باید برای پشتیبانی از چندین زبان در یک سند واحد بازنویسی (override) شود یا خیر	Unicode-bidi
یک المان را به صورت یا با نمای عمودی تراز بندی می کند (می چیند)	Vertical-align
به واسطه ی این خاصیت می توان نحوه ی نمایش فاصله بین حروف را در متن تنظیم کرد	White-space
فاصله ی بین کلمات در یک نوشته توسط این خصوصیت تعیین می شود (فاصله ی بین کلمات در یک نوشته را کاهش یا افزایش می دهد) .	Word-spacing

## فونت در CSS

### خواص فونت در CSS

خانواده ی فونت ( **font family** ) (مجموعه فونت هایی که نمایانگر اشکال مختلف یک نوع **typeface** یا طرح حروف هستند)، پیرنگی (**boldness**)، اندازه و همچنین سبک یک نوشته یا متن را تعریف می کنند .



خانواده های فونت (font family) در CSS

در CSS، دو گروه اسم خانواده ی فونت وجود دارند

**generic family** (خانواده ی عمومی) - گروهی از فونت ها که ظاهری مشابه دارند (از میان آن ها می

توان به "Serif" و "Monospace" اشاره کرد).

**font family** (خانواده ی فونت) - یک نوع خانواده ی فونت ویژه (از جمله "Times New Roman" و

"Arial").

شرح	Font family	Generic family
فونت های Serif خط های کوچکی در انتهای برخی از	Times New Roman	Serif
کاراکترها دارند	Georgia	

پیشوند "Sans" یعنی بدون – این نوع فونت در انتهای کاراکترهای خود خط ندارند	Arial Verdana	Sans-serif
تمامی کاراکترهای monospace از یک میزان پهنا برخوردار هستند	Courier New Lucida Console	Monospace

**نکته** توجه داشته باشید که خواندن فونت های **sans-serif** در مقایسه با فونت های **serif** روی صفحه نمایش رایانه به مراتب آسان تر است .

### خانواده ی فونت (font family)

ویژه گی های خانواده ی فونت با خاصیت **font-family** تعیین می گردند .

خصوصیت **font-family** طبیعتاً باید دربردارنده ی اسم فونت های متعددی باشد که به عنوان یک نوع سیستم پشتیبانی یا یدکی عمل کند . در صورتی که مرورگر مورد نظر از نوع حروف (فونت) اولی پشتیبانی نکرد، در آن صورت فونت دوم را امتحان خواهد کرد .

کافی است با فونت دلخواه کار خود را آغاز کرده، سپس آن را با **generic family** به اتمام برسانید . با این کار شما به مرورگر اجازه می دهید در صورت عدم وجود فونت های دیگر، فونتی مشابه را از **generic family** انتخاب کند .

**نکته** چنانچه اسم یک خانواده ی فونت (**font family**) متشکل از چندین حروف (بیش از یک حرف) باشد، در آن صورت باید اسم فونت مزبور را داخل کاراکتر یا علامت نقل و قول دوتایی (**double quotation**) قرار دهید، مثال "**Times New Roman**".

در مثالی که در زیر ارائه شده، بیش از یک **font family** در لیست با استفاده از ویرگول از هم جدا شده اند

### مثال

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

## سبک فونت ها (Font style)

**Font style** یا همان سبک فونت عمدتاً به منظور مشخص کردن متن ها و نوشته های خوابیده و حروف کج بکار می روند .

خصوصیت نام برده سه مقدار متفاوت دارد

**normal** – که در آن متن به صورت عادی نمایش داده می شود.

**Italic** – که متن به صورت خوابیده نشان داده می شوند .

**Oblique** – که در آن نوشته یا حروف درست مشابه **italic** متمایل شده یا کج می شوند (بیشتر به منظور

شبیه سازی حالت **italic** یا حروف خوابیده بکار می رود) .

مثال

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



آموزشگاه تخنیکر داده ها

اندازه ی فونت (Font Size)

همان طور که از اسم آن پیدا است این خصیصه اندازه ی فونت را تعریف می کند .

قابلیت و توانایی مدیریت اندازه ی نوشته ها و متن ها در طراحی وب از اهمیت بسیار بالایی برخوردار است . با این وجود شما به هیچ وجه نباید با بهره گیری از قابلیت تنظیم اندازه ی فونت کاری کنید که پاراگراف ها ظاهری شبیه به تیترا (heading)، و یا تیترا ظاهری مشابه پاراگراف ها پیدا کنند .

همیشه سعی کنید از تگ های مناسب **HTML** استفاده کنید، به عنوان مثال از **<h1>** - **<h6>** و ویژه ی تیتراها و از **<p>** برای پاراگراف ها بهره بگیرید .

مقدار **font-size** ممکن است یک مقدار مطلق (**absolute**) و حتی نسبی (**relative**) باشد .

## اندازه ی مطلق

نوشته را به یک اندازه ی تعریف شده تنظیم می کند.

به کاربر اجازه ی تغییر اندازه ی متن را در همه ی مرورگرهای موجود نمی دهد (بنا به دلایل دسترسی نامناسب است).

اندازه ی مطلق زمانی کاملاً کارآمد تلقی می گردد که اندازه ی خروجی از پیش برای شما تعریف شده و مشخص باشد.

**توجه** در صورتی که اندازه ی فونت را مشخص نکنید، اندازه ی پیش فرض برای نوشته های معمولی مانند پاراگراف ها که  $16px = 1em$  است مورد استفاده قرار می گیرد .

تنظیم و انتخاب اندازه بر حسب پیکسل (با پیکسل)

تنظیم اندازه ی متن بر حسب پیکسل به شما امکان می دهد اندازه ی نوشته را با اختیار کامل مدیریت کنید

مثال

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

**نکته** در صورت استفاده از پیکسل، می توان با بهره گیری از ابزار بزرگنمایی (**zoom tools**) کل یک صفحه ی وب را مجدداً اندازه بندی کرد (به طور دلخواه اندازه ی آن تغییر داد) .

## تنظیم اندازه ی فونت با Em

برای اجازه دادن به کاربرها جهت اندازه بندی مجدد متن (قابل دسترسی از طریق فهرست گزینهش مرورگر مربوطه)، بسیاری از طراحان وب به جای استفاده از پیکسل، واحد **em** را بکار می برند .

**W3C** بهره گیری از واحد اندازه بندی فوق را به طراحان وب توصیه می کند .

**1em** در واقع معادل با اندازه ی فعلی فونت می باشد . اندازه ی پیش فرض نوشته در مرورگرهای کنونی برابر است با **16px** . بنابراین، اندازه ی پیش فرض **1em** به عبارتی همان **16px** است .

با بهره گیری از فرمول ذیل می توان واحد پیکسل را به **em** محاسبه و تبدیل کرد

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 */
}
```

در مثالی که در بالا مشاهده می کنید، اندازه ی نوشته بر حسب **em** در حقیقت برابر است با مثال پیشین که بر حسب پیکسل اندازه بندی شده بود . با این وجود، با استفاده از اندازه ی **em**، قابلیت تنظیم اندازه ی نوشته در تمامی مرورگرها وجود دارد .

متأسفانه، در ویرایش های قدیمی تر **IE** باز به مشکل برمی خوریم . هنگامی که سعی می کنیم اندازه ی متن را نسبت به قبل بزرگتر کنیم، متن بیش از حد دلخواه بزرگنمایی می شود و زمانی هم که سعی بر کوچک تر کردن نوشته می کنیم باز با همین چالش روبه رو می شویم .

بکاربردن ترکیبی از **Em** و **Percent**

راه حلی که در کلیه ی مرورگرها کارگر و موثر واقع می شود، این است که **font-size** پیش فرض را در قسمت یا عنصر بدنه **<body>** بر حسب درصد تنظیم کنیم

مثال

```
body {
  font-size: 100%;
}
h1 {
  font-size: 2.5em;
}
h2 {
  font-size: 1.875em;
}
```



```
p{
font-size: 0.875em;
}
```

اکنون کد ما به خوبی عمل کرده و نتیجه‌ی دلخواه را بدست می‌آید. به عبارتی دیگر همان اندازه‌ی متن را در تمامی مرورگرها پیاده کرده و بدین وسیله به همه‌ی مرورگرها اجازه می‌دهد نوشته را مطابق نیاز بزرگ یا کوچک نمایی کند.

تمامی خصوصیت‌های فونت CSS در جدول زیر با ذکر شرح کاربرد فهرست شده‌اند

### All CSS Font Properties

خاصیت	توضیح
font	کلیدهای خصوصیت‌های فونت را طی یک تعریف تنظیم و انتخاب می‌کند
font-family	Specifies the font family for text (خانواده‌ی فونتی که برای متن بکارگرفته می‌شود را تعیین می‌کند)
font-size	اندازه‌ی فونت (قلم نوشته) را برای نوشته‌ی مورد نظر تنظیم می‌کند
font-style	سبک فونت انتخابی را برای متن مشخص می‌کند
font-variant	مشخص می‌کند آیا یک متن با قلم نوشته (small-caps font) نمایش داده شود یا خیر
font-weight	این خصوصیت میزان توپر بودن (Weight) متن را تعریف می‌کند

### آموزش Link

می‌توان لینک‌ها را به شیوه‌های مختلف سبک دهی (بندی) کرد.

سبک دهی به لینک‌ها

لینک ها را می توان با هر خاصیت **CSS** ی سبک دهی کرد از جمله ی آن میتوان رنگ, نوع فونت و پس زمینه را نام برد .

### مثال

```
a {
  color: #FF0000;
}
```

به علاوه این امکان وجود دارد که لینک ها را بسته به **state** (حالتی) که در آن قرار دارند, به گونه ای متفاوت سبک دهی کرد .

چهار حالت مختلف برای لینک در زیر فهرست شده

- **alink** – لینکی معمولی که هنوز مورد بازدید قرار نگرفته است.
- **avisted** – لینکی که کاربر از آن بازدید کرده است.
- **ahover** – حالتی که در آن مکان نمای موس روی لینک قرار گرفته است.
- **aactive** – در این حالت کاربر روی لینک مورد نظر کلیک کرده.

### مثال

```
/* unvisited link */
alink {
  color: #FF0000;
}
/* visited link */
avisted {
  color: #00FF00;
}
/* mouse over link */
ahover {
  color: #FF00FF;
}
/* selected link */
aactive {
  color: #0000FF;
}
```

زمانی که سبک دلخواه را برای چندین حالت متعدد لینک انتخاب می کنید, قوانینی در رابطه با ترتیب وجود دارد که باید از آن ها پیروی کنید

- **ahover** باید حتماً پس از **alink** و **avisted** قرار گیرد.
- **aactive** باید پس از **ahover** بیاید.

### سبک های معمول لینک

در مثال فوق لینک مربوطه بسته به حالت (**state**) که در آن قرار دارد، رنگ عوض می کند .

در این بخش به دیگر شیوه های معمول در سبک دهی به لینک ها خواهیم پرداخت

افزودن جلوه های بصری به لینک ها (**Text decoration**)

خاصیت **text-decoration** غالباً به منظور حذف زیرخط ها (**underline**) از لینک ها مورد استفاده قرار

می گیرد

مثال

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

avisted {
  text-decoration: none;
}
ahover {
  text-decoration: underline;
}
aactive {
  text-decoration: underline;
}
```

رنگ پس زمینه (**background color**)

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

مثال

```
alink {
  background-color: #B2FF99;
}
avisted {
  background-color: #FFFF85;
}
```

```

ahover {
  background-color: #FF704D;
}
aactive {
  background-color: #FF704D;
}

```

## لیست در CSS

خاصیت های لیست در **CSS** به شما قابلیت های زیر را ارائه می دهد

به شما امکان می دهد نشانگرهای مختلف برای آیتم های فهرست ویژه ی لیست های رده بندی یا دارای ترتیب (**ordered**) انتخاب کنید.

به شما اجازه می دهد نشانگرهای مختلف برای آیتم های فهرست ویژه ی لیست های که با عدد یا حروف رده بندی یا نشانه گذاری نشده اند (**unordered**) انتخاب کنید.  
همچنین به شما این امکان را می دهد تصویر به عنوان نشانگر آیتم لیست انتخاب کنید .

### انواع لیست ها

در **HTML**، دو نوع متفاوت لیست وجود دارد

- **<ul> unordered lists** – تمامی اقلام یا آیتم های لیست با علامت **Bullets** نشانه گذاری شده اند .
- **<ol> ordered lists** – آیتم های لیست با عدد یا حروف علامت گذاری شده اند

با بهره گیری از امکاناتی که **CSS** در اختیار طراح وب قرار می دهد، می توان سبک دهی به فهرست ها و لیست ها را وارد مرحله ی تازه ای کرد، برای مثال از تصاویر به مثابه ی نشانگرهای آیتم های یک لیست استفاده کرد .

### نشانگرهای مختلف آیتم های لیست (list item marker)

نوع نشانگر آیتم های لیست به وسیله ی 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 مقادیر مثال بالا برای لیست های دارای ترتیب حروفی و عددی و برخی دیگر مختص فهرست های فاقد ترتیب عددی / حروفی است .

انتخاب یک تصویر به عنوان نشانگر آیتم مورد نظر در لیست

برای این منظور (انتخاب تصویر به عنوان نشانگر آیتم در لیست)، کافی است از خاصیت **list-style-image** بهره بگیرید

مثال

```

ul {
  list-style-image: url('sqpurple.gif');
}

```

مثالی که در بالا مشاهده می کنید در کلیه ی مرورگرها به طور مشابه نمایش داده نمی شوند . **IE** و

**Opera** هر دو **image-marker** (تصویر نشانگر) را کمی بالاتر نسبت به مرورگرهای **Chrome**، **Firefox** و **Safari** نمایش می دهد .

اگر می خواهید **image-marker** در تمامی مرورگرها در مکانی یکسان جاگذاری و نمایش داده شود، می توانید از مثال زیر کمک بگیرید که در آن **image-marker** در چندین مرورگر مختلف به طور مشابه و در موقعیتی یکسان قرار گرفته است .

مثال

```

ul {
  list-style-type: none;
  padding: 0px;
  margin: 0px;
}

```

```
ul li {
  background-image: url(sqpurple.gif);
  background-repeat: no-repeat;
  background-position: 0px center;
  padding-left: 15px;
}
```

## توضیح مثال

برای `<ul>`

به منظور حذف نشانگر آیتم لیست، `list-style-type` را روی `none` تنظیم کنید .  
 هر دو خصوصیت `padding` (ویژگی `padding` میزان فاصله و حاشیه ی محتوای درون یک بلاک را مشخص می کند) و `margin` (فاصله دو بلاک یا دو لایه را در صفحات وب تعیین می کند) را به `0px` تنظیم کنید  
 بنا به دلایل سازگاری یا `compatibility` جهت پشتیبانی از چندین مرورگر / سازگاری یک صفحه وب و کدهای آن با مرورگرهای گوناگون).

برای تمامی `<li>` در `<ul>`

`URL` تصویر مربوطه را انتخاب کرده و آن را تنها یکبار نمایش می دهد (بدون هیچ تکراری).

تصویر مورد نظر را هر جا که مایلید قرار دهید (`left 0px` و `vertical value center`).

نوشته را در لیست با `padding-left` جای گذاری کنید

## آموزش خاصیت List - Shorthand

`list-style property` در واقع یک خاصیت مختصر نویسی است . از خصوصیت مزبور جهت تعریف تمامی

خاصیت های لیست تنها در یک تعریف (`declaration`) استفاده می شود .

## مثال

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

در حین استفاده از خاصیت یا ویژه گی مختصر نویسی، توجه داشته باشید که ترتیب مقادیر `property` ها

به ترتیب زیر هستند

**list-style-type** (چنانچه **list-style-image** مشخص شده باشد، مقدار این خاصیت در صورت عدم امکان نمایش تصویر (به هر دلیلی)، نشان داده می شود).

**list-style-position** (مشخص می کند آیا نشانگرهای آیتم های لیست باید داخل یا خارج گردش محتوا (**content flow**) نمایش داده شوند).

**list-style-image** (یک تصویر را به عنوان نشانگر آیتم لیست انتخاب می کند).

در صورت عدم وجود یا در دسترس نبودن یکی از مقادیر **property** های فوق، مقدار پیش فرض برای خصوصیتی که در دسترس نیست درج می شود.

کلیه ی خاصیت های مربوط به لیست در جدول ذیل تشریح شده اند

خاصیت	شرح
list-style	این ویژه گی یک خصوصیت مختصر نویسی است که مقادیر سه خاصیت مختلف مربوط به لیست را در یک تعریف (declaration) قرار می دهد . در واقع تمامی خصوصیت های کی لیست را در تنها یک اعلان ست می کند .
list-style-image	یک تصویر را به عنوان نشانگر یا مارکر آیتم لیست انتخاب می کند . با استفاده از این ویژگی برای عکس نشانگر تعریف می کند و به صورت یک آیکن کوچک نمایش داده می شود و شامل دو حالت است که یکی none می باشد که در این حالت هیچ عکسی در لیست دیده نمی شود، و حالت url که با استفاده از این حالت آدرس تصویر مورد نظر را وارد می کنند .
list-style-position	در این ویژگی موقعیت قرارگیری نشانه و متن آیتم های لیست را بر حسب یکی از حالت های زیرتعیین می کند. که این ویژگی هم شامل دو حالت می باشد که یکی inside می باشد که در این حالت نشانه لیست و متن آیتم های آن کمی تو رفته تر نمایش داده می شوند و حالت outside نشانه لیست و متن آیتم های به صورت عادی نمایش داده می شوند

	(تعیین می کند که نشانگرهای آیتم های لیست داخل یا خارج content flow ظاهر شوند یا خیر) .
list-style-type	نوع نشانگر آیتم لیست را تعیین می کند (با استفاده از این ویژگی می توان marker لیست خود را به شکل دلخواه عوض کنید) .

### جدول در CSS

ظاهر یک جدول HTML را می توان به کمک CSS به شکل قابل توجهی بهبود بخشید . برای مثال جدول زیر را در نظر بگیرید

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Berglunds snabbköp	Christina Berglund	Sweden
Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK
Königlich Essen	Philip Cramer	Germany
Laughing Bacchus Winecellars	Yoshi Tannamuri	Canada
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy
North/South	Simon Crowther	UK



Paris spécialités	Marie Bertrand	France
The Big Cheese	Liz Nixon	USA
Vaffeljernet	Palle Ibsen	Denmark

جهت مشخص کردن حاشیه ی (border) جداول در CSS، کافی است از خاصیت border استفاده کنید .

مثال زیر یک حاشیه ی سیاه برای المان های <table>، <th> و <td> مشخص می کند

مثال

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

توجه داشته باشید که جدول مورد نظر در مثال فوق دارای دو حاشیه است . این امر به این خاطر است که جدول و المان های <th> و <td> حاشیه های مجزا دارند .

به منظور نمایش تنها یک حاشیه برای جدول مربوطه، از خاصیت border-collapse استفاده کنید .

خاصیت border-collapse

خصوصیت نام برده تعیین می کند آیا حاشیه های جدول در قالب تنها یک خط (border) جمع شوند یا به صورت مجزا و جداگانه تنظیم شوند

مثال

```
table {
border-collapse: collapse;
}
Table, th, td {
Border: 1px solid black;
}
```

طول و عرض جدول

طول و عرض یک جدول به ترتیب با خاصیت های height و width مشخص می شود .

مثال زیر پهنای جدول را به 100% و درازای المان های <th> را 50px قرار می دهد

مثال

```
table, td, th{
  border: 1px solid black;
}
table{
  width: 100%;
}
th{
  height: 50px;
}
```

تراز بندی متن به صورت افقی (Horizontal text alignment)

خاصیت **text-align** تراز بندی متن را تنظیم می کند (آن را راست چین، چپ چین، وسط چین) می کند .

به صورت پیش فرض، نوشته ی موجود در المان های <th> وسط چین هستند و متن موجود در المان های <td> چپ چین هستند .

مثال ذیل نوشته ی موجود در المان های <th> چپ چین می کند

مثال

```
th {
  text-align: left;
}
```

تراز بندی عمودی متن (vertical text alignment)

خاصیت **vertical-align**، تراز بندی را به صورت عمودی تنظیم می کند، به عبارتی دیگر جای دهی متن در بالا، پایین و وسط درون جدول .

به صورت پیش فرض، در تراز بندی عمودی، نوشته در وسط جدول قرار می گیرد (برای هر دو المان های <th> و <td> ) .

مثال ذیل نوشته را با خاصیت **vertical-align** برای المان های <td> در پایین جدول قرار می دهد

مثال

```
td {
  height: 50px;
  vertical-align: bottom;
}
```

## استفاده از خاصیت padding در جدول

جهت کنترل فاصله ی بین حاشیه و محتوای یک جدول، لازم است خاصیت **padding** را روی المان های **<td>** و **<th>** اعمال کنید

```
td {
  padding: 15px;
}
```

تنظیم رنگ جدول

مثال زیر رنگ حاشیه ها، متن و پس زمینه ی المان های **<th>** را تعیین می کند

مثال

```
table, td, th {
  border: 1px solid green;
}
th {
  background-color: green;
  color: white;
}
```

آموزش Box در CSS

تمامی عناصر **HTML** را می توان **box** در نظر گرفت. در **CSS**، عبارت "**Box Model**" هنگام صحبت در مورد **design** و **layout** استفاده می شود.

**CSS Box Model**، الزاماً یک **box** است که دور عناصر **HTML** را می گیرد، و تشکیل شده است از حاشیه ها، قاب ها (**border**)، **padding**، و محتوای واقعی.

**box model**، به ما اجازه می دهد قابی را اطراف عناصر و فضای عناصر در رابطه با عناصر دیگر قرار دهیم.

تصویر زیر، **box model** را تشریح می کند



#### توضیح بخش های مختلف

- **margin** نواحی اطراف **border** را پاک می کند. **margin** رنگ پیش زمینه ندارد، و کاملاً شفاف است
- **border** یک **border** اطراف **padding** و **content** را می پوشاند. **border**، از رنگ پیش زمینه **box** تاثیر می گیرد
- **padding** نواحی اطراف **content** را پاک می کند. **padding** از رنگ پیش زمینه **box** تاثیر می گیرد
- **content** محتوای **box**، یعنی جایی که متن و تصاویر ظاهر می شوند.

به منظور تنظیم صحیح پهنا و ارتفاع عنصر در همه مرورگرها، باید بدانید **box model** چگونه کار می کند.

#### پهنا و ارتفاع عنصر

نکته مهم هنگامی که خصیصه پهنا و ارتفاع عنصر را با **CSS** تعیین می کنید، فقط پهنا و ارتفاع ناحیه **content** را تعیین می کنید. به منظور درک اندازه عنصر، همچنین باید **padding**، **border**، و **margin** را نیز اضافه کنید.

کل پهنای عنصر در مثال زیر، **300 px** است

Width:250px;  
Padding: 10px;  
Border:5px solid gray;  
Margin: 10px;

بیا بید محاسبه ریاضی اش را هم انجام دهیم

250 px (پهنا)

+ 20 px (چپ و راست padding)

+ 10 px (چپ و راست border)

+ 20 px (چپ و راست margin)

= 300 px

تصور کنید فقط 250 px فضا دارید. بیا بید عنصری با پهنای کل بسازیم

Width:220px;  
Padding: 10px;  
Border:5px solid gray;  
Margin:0px;

کل پهنای عنصر همیشه باید بصورت زیر برگزار شود

کل پهنای عنصر = پهنا + padding چپ + padding راست + border چپ + border راست + margin چپ + margin راست

کل ارتفاع عنصر نیز باید بصورت زیر محاسبه شود

کل ارتفاع عنصر = ارتفاع + padding بالا + padding پایین + border بالا + border پایین + margin بالا + margin پایین

مبحث سازگاری مرورگرها

اگر مثال قبلی را در **Internet Explorer** امتحان کرده باشید، دیده اید که کل ارتفاع دقیقاً 250 px نبود.

**IE**، هنگام تعیین پراپرتی پهنا، **padding** و **border** را در پهنا محاسبه می کند، مگر اینکه یک **DOCTYPE** تعریف شود.

برای حل این مشکل، کفایست یک **DOCTYPE** به کد اضافه کنید

```
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<html>
<head>
  <style type="text/css">
    div.ex {
      width: 220px;
      padding: 10px;
      border :5px solid gray;
      margin: 0px;
    }
  </style>
</head>
```

## خاصیت Border در CSS

خاصیت های **Border** به شما این مکان را می دهند که **style**، اندازه و رنگ حاشیه ی یک المان را تغییر دهید.

نکته

توجه داشته باشید که هیچ یک از خاصیت های **Border** تا زمانی که خاصیت **Border-style** مقداردهی نشده باشند، تاثیری نخواهند داشت.

## مقادیر Border-style

- **None** هیچ حاشیه ای تعریف نمی شود.
- **Dotted** حاشیه ی نقطه چین تعریف می کند.
- **Dashed** حاشیه ی خط چین تعریف می کند.
- **Solid** یک حاشیه با خط ممتد رسم می کند.
- **Double** دو خط حاشیه تعریف می کند.
- **Groove** یک خط حاشیه ی سه بعدی نمایش می دهد که در آن خطوط بالایی و سمت چپ پررنگ تر از خطوط دیگر هستند.
- **Ridge** در این حالت خطوط حاشیه به صورت سه بعدی و با طیف رنگی نمایش داده می شوند.

- **Inset** این مقدار یک حاشیه ی داخلی سه بعدی تعریف می کند که در آن معمولا خطوط بالایی و سمت چپ دارای سایه و تیره تر از خطوط دیگر هستند.
- **Outset** حاشیه ی سه بعدی خارجی تعریف می کند که در آن خطوط پایینی و سمت راست دارای سایه و تیره تر از خطوط دیگر هستند.

### ضخامت حاشیه (Border width)

با استفاده از خاصیت **Border-width** می توان ضخامت خط حاشیه را تنظیم کرد.

عرض (**width**) یا بر حسب پیکسل تنظیم می شود و یا با بهره گیری از سه مقدار از پیش تعریف شده ی **thin**، **medium** و **thick**.

### نکته

خاصیت "**Border-width**" زمانی که به تنهایی بکار برده می شود، کار خاصی انجام نمی دهد. ابتدا باید به وسیله ی خاصیت "**Border-style**" خطوط حاشیه را مقدار دهی (تنظیم) کنید.

### مثال

```
p.one {
  border-style: solid;
  border-width: 5px;
}
p.two {
  border-style: solid;
  border-width: medium;
}
```

### رنگ خط حاشیه (Border color)

خاصیت **border-color** به منظور تنظیم و مقداردهی رنگ خط حاشیه مورد استفاده قرار می گیرد.

رنگ خط حاشیه را می توان با مقادیر زیر تنظیم کرد.

**Name** - اسم رنگ دلخواه را مشخص می کند، مانند "**red**".

**RGB** - یک مقدار **RGB** (**red blue green**) تعریف می کند.

**Hex** - یک مقدار شانزده شانزدهی (hex) مانند "#ff0000" مشخص می کند.

همچنین می توان رنگ پس زمینه را شفاف "transparent" انتخاب کرد.

چنانچه رنگ خط حاشیه مقداردهی نشده باشد، **border** رنگ خود را از خاصیت **color** مربوطه به ارث می برد.

#### توجه

خاصیت "**border-color**" اگر به تنهایی بکار گرفته شود، هیچ اثری نخواهد داشت. بدین معنا که شما باید در مرحله ی اول با استفاده از خاصیت "**border-style**"، خطوط حاشیه را مقداردهی کنید.

#### مثال

```
p.one {  
  border-style: solid;  
  border-color: red;  
}  
p.two {  
  border-style: solid;  
  border-color: #98bf21;  
}
```



تعریف حاشیه به صورت مجزا برای هر طرف در CSS

این امکان وجود دارد که برای حاشیه ی هر طرف ظاهری متفاوت تعریف کرد (برای هر طرف یک حاشیه ی متفاوت داشت).

#### مثال

```
p {  
  border-top-style: dotted;  
  border-right-style : solid;  
  border-bottom-style: dotted;  
  border-left-style: solid;  
}
```

مثال بالا را همچنین می توان با تنها یک خاصیت تنظیم یا مقدار دهی کرد.

#### مثال



```
p {
border-style: dotted solid;
}
```

خاصیت **border-style** می تواند از یک تا چهار مقدار داشته باشد.

**border-style dotted solid double dashed;**

حاشیه ی بالایی نقطه چین می شود.

حاشیه ی سمت راست عنصر با خطی معمولی کشیده می شود.

حاشیه ی پایینی با دو خط نمایش داده می شود.

حاشیه ی سمت چپ خط چین می شود.

**border-style dotted solid double;**

حاشیه سمت بالا نقطه چین می شود.

حاشیه های سمت راست و چپ با خط معمولی ترسیم می شوند.

حاشیه ی سمت پایینی دو خطه می شود.

**border-style dotted solid;**

حاشیه های سمت بالا و پایینی نقطه چین می گردند.

حاشیه های سمت راست و چپ با خطی معمولی کشیده می شوند.

**border-style dotted;**

هر چهار حاشیه با خطی معمولی تعریف می شوند.

خاصیت **border-style** در مثال بالا بکار گرفته شده، اگرچه قابلیت استفاده از آن با **border-width** و

**border-color** وجود نیز وجود دارد.

## خاصیت Border-Shorthand

همان طور که در مثال های فوق مشاهده می کنید، خاصیت های مختلفی وجود دارند که در برخورد و کار با خطوط حاشیه باید به آن ها توجه داشت.

به منظور کاهش اندازه ی کد (مختصر نویسی یا بهینه نوشتن کد)، همچنین می توان تمامی خاصیت های جداگانه را در یک خاصیت واحد تعریف کرد. این خاصیت را مختصر نویسی کد یا همان **short-hand property** می نامند.

خاصیت ذکر شده در واقع فشرده یا خلاصه ی خاصیت های مجزای زیر است.

**border-width**

**border-style** (الزامی)

**border-color**

```
p {  
  border: 5px solid red;  
}
```

مثال

تمامی خاصیت های مربوط به خطوط حاشیه در **CSS**

خاصیت	شرح
<u>border</u>	کلید ی خصوصیت های border را در تنها یک تعریف (declaration) ، تنظیم می کند
<u>border-bottom</u>	تمامی خصوصیت های خط حاشیه ی پایین (bottom border) را در یک تعریف تنظیم می کند؛ در برگیرنده خواص کلی خط حاشیه برای خط حاشیه پایین عنصر به صورت یک خاصیت چند مقداری است

<u>border-bottom-color</u>	رنگ حاشیه ی سمت پایین (پایین عنصر) را تنظیم می کند
<u>border-bottom-style</u>	style خط حاشیه ی پایین عنصر را تعریف می کند
<u>border-bottom-width</u>	ضخامت یا پهنای خط حاشیه ی عنصر سمت پایین را تنظیم می کند
<u>border-color</u>	رنگ خط حاشیه های هر چهار سمت را تعیین می کند
<u>border-left</u>	تمامی خصوصیت های خط حاشیه ی سمت چپ را در تعریفی واحد (تنها یک تعریف) تنظیم می کند
<u>border-left-color</u>	رنگ خط حاشیه ی سمت چپ (عنصر) را تعیین می کند
<u>border-left-style</u>	مقدار دهی به style خط حاشیه ی سمت چپ
<u>border-left-width</u>	ضخامت یا پهنای خط حاشیه ی سمت چپ را تنظیم م مقداردهی می کند
<u>border-right</u>	تمامی خصوصیت های خط حاشیه ی سمت راست را تعیین می کند
<u>border-right-color</u>	رنگ خط حاشیه ی سمت راست را تعیین می کند
<u>border-right-style</u>	سبک دهی (style) خط حاشیه ی سمت راست را مقداردهی می کند
<u>border-right-width</u>	پهنای خط حاشیه ی سمت راست را تنظیم می کند
<u>border-style</u>	سبک دهی (style) خط حاشیه ی هر چهار طرف را تنظیم می کند

<u>border-top</u>	کلیه ی خصوصیت های حاشیه ی سمت بالای عنصر را در یک تعریف مقداردهی می کند
<u>border-top-color</u>	رنگ خط حاشیه ی سمت بالا را تنظیم می کند
<u>border-top-style</u>	Style (سبک دهی) خط حاشیه ی سمت بالا را تنظیم می کند
<u>border-top-width</u>	ضخامت خط حاشیه ی سمت بالا را تنظیم می کند
<u>border-width</u>	پهنای خط حاشیه ی هر چهار طرف را تنظیم می کند

## آموزش Outline

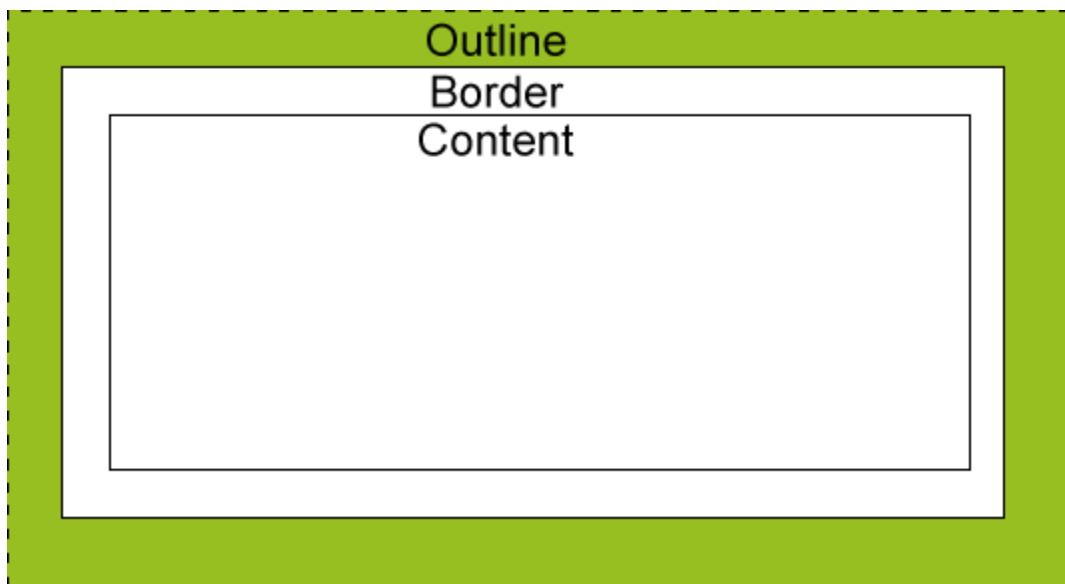
**Outline** (یا همان خط پیرامون) خطی است که پیرامون المان ها (بیرون از خط حاشیه) ترسیم می گردد تا بدین وسیله عنصر مورد نظر بیشتر به چشم آید یا برجسته تر دیده شود .

خصوصیت های **outline** نیز **style** یا سبک دهی، رنگ و پهنای خط حاشیه را تعیین می کنند .

## Outline / خط پیرامون در CSS

باید توجه داشته باشید که خصوصیت **outline** با خصوصیت **border** تفاوت دارد .

**Outline** جزئی از ابعاد یک المان محسوب نمی شود، به عبارتی دیگر عرض و طول کلی (**width** و **height**) المان مربوطه تحت تاثیر پهنای خط پیرامون قرار نمی گیرد .



همان طور که در جدول زیر مشاهده می کنید، تمامی خصوصیت های مربوط به **outline** همراه با شرح آن ها فهرست شده است

مقادیر	شرح	خصوصیت
<i>outline-color</i> <i>outline-style</i> <i>outline-width</i> inherit	تمامی خاصیت های خط پیرامون را در یک تعریف تنظیم می کند	<a href="#">outline</a>
<i>color_name</i> <i>hex_number</i> <i>rgb_number</i> invert inherit	رنگ خط پیرامون را تعیین می کند	<a href="#">outline-color</a>
none dotted	سبک دهی / style خط پیرامون را تنظیم می کند	<a href="#">outline-style</a>

		dashed solid double groove ridge inset outset inherit
<u>outline-width</u>	ضخامت خط پیرامون را تنظیم می کند	thin medium thick <i>length</i> inherit

## آموزش Margin

خاصیت های مربوط به **Margin** فضای اطراف المان ها را تعریف می کند .

**Margin** در حقیقت ناحیه ی اطراف یک عنصر را در بر می گیرد . حاشیه ی دور عنصر (**margin**) فاقد رنگ پس زمینه بوده و کاملاً شفاف (**transparent**) می باشد .

**Margin** جهات چهارگانه (حاشیه ی دور عناصر در چهار جهت راست، چپ، بالا و پایین) را می توان به صورت تکی و جداگانه اصلاح کرد (و یا هر چهار طرف را با استفاده از خاصیت مختصر نویسی **margin** ( **short-hand** ) **property**) یکجا و همزمان تغییر داد .

مقدار	شرح
auto	مرورگر حاشیه ی دور عناصر / margin را خودکار محاسبه و اندازه گیری می کند
length	حاشیه ی یک عنصر از عنصر / margin دیگر را بر حسب واحدهای (پیکسل) px, (نقطه) pt, cm, و غیره تعیین می کند . مقدار پیش فرض 0px می باشد
%	حاشیه ی دور عنصر را به میزان درصد پهنای المان دربرگیرنده مشخص می کند / فاصله ی دور عنصر را بر حسب درصد تعیین شده, برابر با کل عرض یا پهنای المان دربرگیرنده تعیین می کند
inherit	تعیین می کند که حاشیه ی دور عنصر باید از عنصر والد / parent element به ارث برده شود

**نکته** این امکان نیز وجود دارد با استفاده از مقادیر منفی کاری کرد که محتوا یا عناصر تا حدی روی هم قرار بگیرند (همپوشانی یا overlap رخ دهد) .

تعیین margin ی متفاوت برای هر چهار طرف (تعیین میزان حاشیه ی دور عناصر به صورت مجزا در هر طرف) در CSS این امکان وجود دارد که ویژه ی هر چهار جهت, حاشیه دور عنصر متفاوتی تنظیم کرد

مثال

```
p {  
  margin-top: 100px;  
  margin-bottom: 100px;  
  margin-right : 150px;  
  margin-left : 50px;  
}
```

## مقدار دهی margin هر چهار جهت در یک تعریف

به منظور بهینه نویسی کد در CSS و کاهش اندازه ی آن، می توان تمامی خصوصیت های **margin** (حاشیه ی دور عناصر در هر چهار طرف) را در تنها یک خاصیت یا یک **property** واحد، تعیین کرد . به این روش مقداردهی **shorthand property** می گویند .

خاصیت مختصر نویسی (**shorthand property**) برای مقداردهی هر چهار جهت، همان خاصیت "**margin**" می باشد .

### مثال

```
p {  
  margin: 100px 50px;  
}
```

خاصیت **margin** می تواند یک تا چهار مقدار داشته باشد

**Margin: 25px 50px 75px 100px;**

می باشد **25px** حاشیه ی دور عنصر سمت بالا

**50px** میزان حاشیه ی دور عنصر سمت راست

می باشد **75px** حاشیه ی دور عنصر سمت پایین

می باشد **100px** حاشیه ی دور عنصر سمت چپ نیز

**Margin: 25px 50px 75px;**

**25px** حاشیه ی دور المان سمت بالا

**50px** حاشیه ی دور عنصر سمت راست و چپ

**75px** حاشیه ی سمت پایین

**Margin: 25px 50px;**

**25px** حاشیه ی دور عنصر بالا و پایین



50px سمت راست و چپ نیز Margin

Margin: 25px;

25px حاشیه ی دور عنصر هر چهار طرف

خاصیت	شرح
margin	را در یک تعریف مقداردهی می کند)) حاشیه ی دور عنصر margin تمامی خصوصیت های
margin-bottom	میزان حاشیه ی دور عنصر سمت پایین را تنظیم می کند
margin-left	سمت چپ المان را تعیین می کند Margin
margin-right	میزان حاشیه عنصر سمت راست را تنظیم می کند
margin-top	سمت بالای عنصر را تنظیم می کند Margin

## آموزش Padding

خاصیت padding در CSS, فاصله ی بین خطوط حاشیه (border) ی عنصر و محتوای (content) عنصر را تعریف می کند (در واقع با استفاده از خاصیت ذکر شده می توان فاصله ی بین خط حاشیه / border و محتوای داخل عنصر مربوطه را تنظیم کرد) .

### Padding (فاصله / حاشیه ی درونی عناصر)

Padding ناحیه ی پیرامون محتوای یک المان را دربر می گیرد. توجه داشته باشید که حاشیه ی درونی عناصر (padding) تحت تاثیر رنگ پس زمینه قرار می گیرد .

**Padding** هر چهار جهت بالا، پایین، چپ و راست را می توان به صورت جداگانه به وسیله ی خواص (**property**) مجزا تنظیم کرد یا تغییر داد . همچنین می توان با بهره گیری از خاصیت مختصر نویسی (**shorthand property**), حاشیه ی درونی عناصر (**padding**) هر چهار جهت را یکجا تنظیم کرد .

مقادیر ممکن

مقدار	شرح
<i>Length</i>	فاصله ی درونی عنصر / padding را می توان بر حسب واحدی همچون px, pt, em و غیره تعیین کرد
%	به میزان درصد تعریف شده فاصله ی درونی عنصر را تعیین کرد

تنظیم فاصله ی درونی عنصر به صورت تکی برای هر جهت

در **CSS** این امکان وجود دارد که فاصله ی درونی چهار جهت را به صورت تکی یا مجزا مقدار دهی کرد (برای هر طرف **padding** متفاوتی تنظیم کرد)

مثال

```
p {
padding-top: 25px;
padding-right: 50px;
padding-bottom: 25px;
padding-left: 50px;
}
```

تعیین با استفاده از خاصیت چند مقداری padding

به منظور بهینه سازی کدنویسی و کاهش مقدار کدی که می نویسیم می توان خاصیت **padding** (میزان فاصله ی دور عنصر) هر چهار جهت را در یک تعریف مقدار دهی و تنظیم کرد . همان طور که پیش تر نیز ذکر شد به این روش مقدار دهی **shorthand property** یا خاصیت مختصر نویسی می گویند (که می توان آن را خاصیت چند مقداری نیز نامید) .

مثال

```
p {  
padding: 25px 50px;  
}
```

خاصیت padding می تواند از یک تا چهار مقدار داشته باشد

**Padding: 25px 50px 75px 100px;**

top padding = 25px

right padding = 50px

bottom padding = 75px

left padding = 100px

**padding: 25px 50px 75px;**

top padding = 25px

right and left paddings = 50px

bottom padding = 75px

**padding: 25px 50px;**

top and bottom paddings = 25px

right and left paddings = 50px

**padding: 25px;**

all four paddings = 25px

خاصیت	شرح
<u>padding</u>	کلیه ی خاصیت های padding در هر چهار جهت را با بهره گیری از خاصیت shorthand در تنها یک تعریف مقدار دهی می کند
<u>padding-bottom</u>	Padding سمت پایین را تنظیم می کند
<u>padding-left</u>	padding سمت چپ را مقداردهی می کند
<u>padding-right</u>	padding سمت راست یک عنصر را تنظیم و مقدار دهی می کند
<u>padding-top</u>	حاشیه درونی عنصر را در سمت بالا تنظیم می کند

## آموزش خاصیت display

خاصیت **display** تعیین می کند آیا المانی نمایش داده شود یا خیر و همچنین نحوه ی نمایش المان را مشخص می کند . خاصیت دیگر نیز **visibility** است که تعیین می کند عنصری قابل رویت باشد یا خیر (پنهان باشد) .

پنهان سازی المان – **display none** یا **visibility hidden**

پنهان سازی یک عنصر را می توان با تخصیص مقدار "**none**" به خاصیت **display** و یا تخصیص "**hidden**" به خاصیت **visibility** انجام داد .

با این وجود لازم است توجه داشته باشید که دو روش مذکور نتایج متفاوتی را به دست می دهند

**visibilityhidden** عنصر مربوطه را پنهان می کند ولی دقیقاً همان فضایی را که از قبل به خود اختصاص داده بود را اشغال می کند . به عبارتی دیگر المان مخفی خواهد شد ولی در عین حال **layout** (شمای کلی) را تحت تاثیر قرار می دهد

مثال

```
h1.hidden {  
  visibility: hidden;  
}
```

**display:none** عنصر مورد نظر را مخفی می سازد, ولی درست برخلاف **visibility:hidden** هیچ فضایی را اشغال نمی کند . المان پنهان گشته و صفحه ی مورد نظر به گونه ای نمایش داده می شود که گویی آن المان اصلاً حضور ندارد

مثال

```
h1.hidden {  
  display: none;  
}
```

خاصیت Display در CSS – عناصر Block و Inline

المان **block**, عنصری است کل پهنا یا عرض موجود را به خود تخصیص داده و اشغال می کند, یک خط فاصله قبل و بعد آن قرار می گیرد (در واقع به صورت عنصری مجزا و مستقل نمایش داده می شود)

مثال هایی از المان های **block** زیر ارائه شده

```
<h1>  
<p>  
<i>  
<div>
```

یک عنصر **inline** (درون خطی) تنها به اندازه ی لازم پهنا را می گیرد و هیچ خط فاصله ی ایجاد نمی کند, که حالت پیش فرض می باشد .

نمونه هایی از آن را زیر مشاهده می کنید

```
<span>  
<a>
```

اصلاح نحوه ی نمایش یک عنصر

تغییر یک المان درون خطی (**inline element**) به یک **block element**, و یا عکس آن, می تواند در تنظیم ظاهر وب به صورت دلخواه, ولی با رعایت استانداردهای فعلی وب برای طراح بسیار مفید واقع شود .

مثال ذیل المان های `<li>` را به صورت درون خطی (`inline`) نمایش می دهد

مثال

```
li {  
  display: inline;  
}
```

حال این مثال عناصر `<span>` را به صورت مجزا (`block`) نمایش می دهد

مثال

```
span {  
  display: block;  
}
```

**نکته** تنظیم خاصیت `display` تنها نحوه ی نمایش المان را تغییر می دهد و نه نوع المان مورد نظر را . از این رو یک عنصر درون خطی (`inline`) با `display: block` اجازه ی جای دادن دیگر المان های `block` در خود را ندارد .

### تعیین مختصات و محل قرار گیری عناصر در صفحه

خاصیت های تعیین موقعیت (`Positioning properties`) در `CSS` برای طراح وب این قابلیت را فراهم می کند که المان را در مکان مشخصی در صفحه جای گذاری کند . با استفاده از این المان همچنین می توان یک عنصر را پشت عنصر دیگری قرار داد، همچنین تعیین کرد در صورت بزرگ بودن محتوای یک المان چه باید رخ دهد . المان ها را می توان با بهره گیری از خاصیت های `left`، `bottom`، `top` و `right` تعیین موقعیت کرد . با این وجود باید توجه داشت که این خواص تنها در صورتی کار می کنند که خاصیت `position` از پیش تنظیم شده باشد . این را هم باید گفت که بسته به روش تعیین موقعیت (`positioning`) بکار برده شده، نحوه ی کارکرد آن ها تغییر می کند .

در کل چهار روش مختلف تعیین موقعیت وجود دارد که به ترتیب ذیل توضیح داده شده

## تعیین موقعیت / موقعیت دهی به صورت ایستا (static positioning)

المان های HTML به صورت پیش فرض با این حالت (به صورت ایستا) در صفحه جای گذاری (تعیین موقعیت) می شوند . در این حالت المان در موقعیتی که مرورگر حین تنظیم نمایش صفحه با در نظر گرفتن محل قرار گیری دیگر عناصر یا محتوا به آن تخصیص داده می شود، جای می گیرد .

المان هایی که بدین ترتیب در صفحه قرار داده شده باشند، تحت تاثیر خاصیت های **top**، **bottom**، **left** و **right** قرار نمی گیرد .

## تعیین موقعیت قرار گیری عنصر به صورت دقیق و ثابت (Fixed positioning)

المانی که مختصات **fixed** یا ثابت در صفحه داشته باشد، در واقع نسبت به موقعیت پنجره ی مرورگر تعیین مختصات می شود و در صورت بالا و پایین رفتن (**scroll**) صفحه نیز هیچ حرکتی نکرده و از جای خود تکان نمی

خورد

مثال

```
p.pos_fixed {  
  position: fixed;  
  top: 30px;  
  right: 5px;  
}
```

نکته IE7 و IE8 هر دو فقط زمانی از مقدار **fixed** پشتیبانی می کنند که یک **DOCTYPE**! (نوعی معرفی در ابتدای سند **sgml** یا **html** که یک شناسه ی عمومی یا سیستمی برای تعریف نوع سند / **DTD** اختصاص می دهد) تعریف شده باشد .

المان هایی که به صورت ثابت (**fixed**) در صفحه قرار داده شده باشند، در حقیقت از جریان (**flow**) عادی صفحه خارج می گردند .

المان هایی که موقعیت ثابت دارند می توانند روی دیگر عناصر قرار گیرند (همپوشانی یا **overlapping**) .

## تعیین موقعیت یا موقعیت دهی به صورت نسبی (relative positioning)

المانی که به صورت نسبی در صفحه ی HTML موقعیت دهی شده باشد می تواند نسبت به محل قرار گیری طبیعی خود تغییر مکان دهد و یا با دیگر عناصر همپوش شود (روی دیگر عناصر قرار گیرد)، با این وجود فضای رزرو شده در جریان عادی صفحه محفوظ می ماند .

### مثال

```
h2.pos_top {  
  position: relative;  
  top: 50px;  
}
```

المان هایی که بدین ترتیب موقعیت دهی شده باشند، اغلب به عنوان دربرگیرنده های **block** ویژه ی عناصری که به صورت مطلق (**absolute**) در صفحه قرار داده شده باشند بکار گرفته می شوند .

## موقعیت دهی به صورت مطلق (absolute positioning)

در این حالت عنصر نسبت به اولین المان بالای سر خود یا همان المان پدر (**parent element**) موقعیت دهی می گردد که نوع موقعیتی آن کاملاً متفاوت و غیر از حالت ایستا می باشد . در صورتی که چنین المانی یافت نشد، این عناصر نسبت به عنصر **html** تغییر موقعیت خواهد داد (**block** دربرگیرنده **<html>** خواهد بود)

### مثال

```
h2 {  
  position: absolute;  
  left: 100px;  
  top: 150px;  
}
```

در این حالت موقعیت دهی نیز المان ها از جریان عادی صفحه حذف می گردند .

عناصر با موقعیت مطلق می توانند بر روی دیگر عناصر قرار گیرند یا با آن ها هم پوش شوند (**overlap**) .

## عناصر همپوش (المان هایی که روی هم قرار می گیرند/ overlapping elements)

خاصیت **z-index** ترتیب و اولویت قرارگیری یک المان را تعیین می کند (اینکه کدام المان باید در جلو یا پشت دیگر عناصر قرار گیرد) . در واقع برای مدیریت اولویت قرار گیری عناصر بر روی یکدیگر، مقدار **z-index** بکار



می رود، بدین معنا که چند عنصر می توانند همزمان روی هم قرار گیرند . تعیین اولویت قرارگیری المان ها بر روی یکدیگر با تنظیم مقدار **z-index** تعیین می گردد .

یک المان می تواند **z-index** مثبت یا منفی داشته باشد

### مثال

```
img {
  position: absolute;
  left: 0px;
  top: 0px;
  z-index: -1;
}
```

عناصر با **z-index** بیشتر نسبت به المان هایی که **z-index** کمتری دارند، بالاتر قرار می گیرند .

**نکته** چنانچه دو المان بدون اینکه **z-index** ی تعیین شده باشد روی هم سوار شوند، آن المانی که در انتهای کد **HTML** قرار گرفته باشد، بالاتر و جلوی المان دیگری نمایش داده می شود .

تمامی خاصیت های موقعیت دهی در **CSS**

مقادیر	شرح	خاصیت
auto  <i>length</i>  %  inherit	لبه ی حاشیه ی دور عنصر / margin edge را برای کادر (BOX) موقعیت دهی شده تنظیم می کند	<u>bottom</u>
<i>shape</i>  auto  inherit	المانی که به صورت مطلق / absolute موقعیت دهی شده باشد را برش می دهد	<u>clip</u>

<u>cursor</u>	<p>نوع مکان نما یا اشاره گر که نمایش داده می شود را تعیین می کند</p>	<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
<u>left</u>	<p>لبه ی حاشیه ی دور عنصر سمت چپ را برای کادر / box موقعیت دهی شده تنظیم می کند</p>	auto <i>length</i> % inherit
<u>overflow</u>	<p>مشخص می کند در صورت بیرون زدن یا سر ریز شدن محتوا از کادر المان، چه اتفاقی بیافتد .</p>	auto hidden scroll

		visible inherit
<u>position</u>	نوع موقعیت المان را مشخص می کند	absolute fixed relative static inherit
<u>right</u>	لبه ی دور حاشیه سمت راست را برای کادر موقعیت دهی شده تعیین می کند	auto <i>length</i> % inherit
<u>top</u>	لبه ی حاشیه ی دور عنصر سمت بالا را برای کادر موقعیت دهی شده تنظیم می کند	auto <i>length</i> % inherit
<u>z-index</u>	اولویت و نحوه ی قرارگیری المان ها بر روی یکدیگر را تنظیم و مدیریت می کند	<i>number</i> auto inherit

## آموزش Float

خاصیت **Float** در **CSS** چه کاربردی دارد ؟

با بهره گیری از خاصیت **float** یا شناور سازی در **CSS** می توان المان مورد نظر را به سمت راست یا چپ شناور کرد یا موقعیت آن را در سمت چپ و راست تنظیم کرد . از این طریق المان های دیگر می توانند مجاور المان مورد نظر در یک خط قرار گیرند .

خاصیت مزبور بیشتر ویژه ی تصاویر مورد استفاده قرار می گیرد، اما حین کار با طرح کلی / **layout** نیز می توان از آن بهره گرفت .

### عناصر چگونه شناور می شوند ؟

المان ها را فقط می توان به صورت افقی شناور کرد، بدین معنا که تنها موقعیت عناصر را می توان در سمت چپ یا راست تنظیم کرد . امکان قرار دادن عناصر کنار هم در موقعیت های بالا و پایین وجود ندارد .

وقتی به یک عنصر ویژگی **float** به چپ یا راست تخصیص داده می شود ، عنصر مورد نظر به آن سمت به حرکت در آمده و بقیه عناصری که بعد از عنصر شناور آمده اند و شناور نیستند (خاصیت **float** را ندارند) به طرف مخالف رانده شده و پیرامون عنصر مورد نظر را می گیرند .

عنصر شناور می تواند در راست یا چپ ترین موقعیت ممکن قرار گیرد (در دورترین موقعیت سمت چپ یا راست المان دربرگیرنده و تا جایی که امکان و جا برای حرکت داشته باشند) .

عناصر بعد از یک عنصر با ویژگی **float** تغییر حالت خواهند داد.

عناصر قبل از یک عنصر با خصوصیت **float** تحت تاثیر قرار نمی گیرند.

یک عنصر با خاصیت **float** به اندازه محتوای داخل عنصر تغییر اندازه خواهد داد . این در حالی است که به صورت پیش فرض اندازه ای برای این المان در نظر نگرفته شده باشد.

مثال

```
img {  
  float: right;  
}
```

چنانچه چندین المان با خاصیت **float** را یکی پس از دیگری قرار دهید، در صورت وجود فضای لازم برای حرکت، عناصر در کنار هم به صورت شناور قرار می گیرند .

مثال

```
.thumbnail {  
  float: left;  
  width: 110px;  
  height: 90px;  
  margin: 5px;  
}
```

لغو حالت شناور عناصر (خاصیت **float**) با استفاده از خاصیت **Clear**

المان هایی که پس از عنصر با خاصیت **float** قرار دارند، همگی پیرامون المان نام برده (با خاصیت **float**) را احاطه می کنند . به منظور جلوگیری از وقوع این رخداد می توان خاصیت **clear** (خالی کردن یا پاک سازی فضای اطراف عناصر) را بکار برد . به وسیله ی خاصیت مذکور می توان تعیین کرد، عنصرهای دیگر اجازه ی دسترسی و قرار گیری در اطراف عنصر مربوطه (با خصوصیت **float**) را ندارند .

کافی است یک **text line** وارد گالری تصاویر کرده و خاصیت **clear** را بکار ببرید

```
.text_line {  
  clear: both;  
}
```

تمامی خصوصیت های مربوط به **float** یا شناور سازی در CSS

مقادیر	شرح	خاصیت
left right both none inherit	کناره هایی از المان شناور را که المان های دیگر اجازه ی قرار گرفتن در آن را ندارند تعیین می کند .	clear
left right none inherit	تعیین می کند آیا کادر (box) شناور شود یا خیر .	float

### شبه عناصر در CSS

#### شبه عناصر در CSS چه کاربردی دارند ؟

می توان با استفاده از شبه عناصر در CSS بخش های مختلف یک المان را سبک دهی کرد (style دلخواه را به آن اعمال کرد) .

از شبه عنصر های برای انجام کارهایی همچون

- سبک دهی یا اعمال سبک به اولین خط یا حرف یک المان
- محتوا مورد نظر را پیش یا پس از محتوای یک عنصر درج کرد

#### ترکیب نحوی (syntax)

ترکیب نحوی شبه عناصر به ترتیب زیر می باشد

```
selectorpseudo-element {
```

```
property:value;  
}
```

به دوتا دو نقطه که در مثال فوق بکار برده شده دقت کنید - **first-line** در مقابل **first-line**

سیستم حاشیه نویسی (**notation**) دو تا کاراکتر جایگزین یک دو نقطه در **CSS** گردید . این در واقع تلاشی بود از سوی کنسرسیوم وب جهان **W3C** که به منظور ایجاد وجه تمایز بین **pseudo-classes** (شبه کلاس ها) و **pseudo-elements** (شبه عناصر) صورت گرفت .

ترکیب نحوی تک دو نقطه ی در ویرایش های قدیمی تر **CSS2** و **CSS1** برای شبه کلاس ها و شبه عناصر بکار می رفت .

برای ایجاد سازگاری با ورژن های قبلی (سازگاری عقبرو **backward compatability**)، دستور نحوی تک دو نقطه ی را می توان برای **CSS2** و **CSS1** بکار برد .

شبه المان **first-line**

شبه المان **first-line** غالباً به منظور افزودن یا اعمال سبک (**Style**) خاص به اولین خط یک متن بکار می رود .

شبه المان **first-line** را تنها می توان به المان های **block** اعمال کرد .

مثال

خط اول نوشته یا متن را در کلیه ی المان های **<p>** قالب دهی (فرمت) کنید

```
pfirst-line {  
  color: #ff0000;  
  font-variant :small-caps;  
}
```

خصوصیت های زیر به شبه المان **first-line** اعمال می شوند

font properties (خصوصیت های فوت یا قلم نوشته)

color properties (خصوصیت های رنگ)

background properties (خصوصیت های مربوط به پس زمینه)

word-spacing (کنترل فاصله بین کلمات)

letter-spacing (کنترل فاصله ی بین حروف)

text-decoration (اعمال جلوه های بصری و تزئین نوشته)

vertical-align (ترازبندی در جهت عمودی)

text-transform (کنترل کوچک و بزرگی حروف)

line-height (تنظیم فاصله ی بین خطوط)

clear (آزاد کردن فضای پیرامون عنصر)

شبه المان first-letter

شبه المان first-letter به منظور اعمال سبک یا style خاصی به اولین خط یک نوشته بکار می رود .

شبه المان first-letter را تنها می توان به المان های block اعمال کرد .

مثال

تمامی حروف اول نوشته را در المان های <p> قالب دهی کنید

```
first-letter {  
  color: #ff0000;  
  font-size: xx-large;  
}
```

تمامی خصوصیت های ذیل به شبه المان first-letter اعمال می شوند

font properties

color properties

background properties

margin properties (میزان حاشیه ی دور عناصر را تنظیم می کند)



padding properties (خصوصیت هایی که حاشیه ی درونی عنصر را تنظیم می کند)

border properties (خصوصیت مربوط به تنظیم خطوط حاشیه)

text-decoration

vertical-align ( فقط در صورتی که "float" مقدار "none" داشته باشد )

text-transform

line-height

float (شناور سازی عنصر به چپ یا راست)

clear

```
p.introfirst-letter {  
  color: #ff0000;  
  font-size:200%;  
}
```

مثال فوق اولین حروف پاراگراف ها را با `class="intro"`، به رنگ قرمز و با اندازه ی بزرگتر نمایش می دهد .

ترکیب چندین شبه عنصر (Multiple pseudo-element)

این امکان وجود دارد که چندین شبه عنصر متعدد را با هم ترکیب کرد .

در مثال زیر همان طور که خود مشاهده می کنید اولین حرف یک پاراگراف قرمز و اندازه ی قلم نوشته یا فونت

`xx-large` خواهد بود . باقی حروف اولین خط آبی، و اندازه ی فونت آن `small-caps` می باشد . بقیه ی

پاراگراف نیز با رنگ و اندازه ی پیش فرض تنظیم می شود

مثال

```
pfirst-letter {  
  color: #ff0000;  
  font-size: xx-large;  
}  
pfirst-line {  
  color: #0000ff;  
  font-variant: small-caps;  
}
```

CSS – شبه‌المان before

با استفاده از این شبه‌عنصر می‌توان پیش از محتوای یک المان، کانتنت دیگری درج کرد و قرار داد .

مثال زیر پیش از هر المان **<h1>** یک تصویر را درج می‌کند

مثال

```
h1before {  
  content :url(smiley.gif);  
}
```

CSS – شبه‌عنصر after

با بهره‌گیری از این شبه‌عنصر می‌توان محتوای جدیدی پس از کانتنت یک المان ایجاد کرد .

آموزشگاه تخصصی داده

مثال زیر یک تصویر پس از هر المان **<h1>** درج می‌کند

مثال

```
h1after {  
  content :url(smiley.gif);  
}
```

شبه‌عنصر selection و کاربرد آن در CSS

شبه‌المان **selection** با آن بخشی از یک عنصر مطابقت دارد که توسط کاربر انتخاب شده باشد (این شبه‌عنصر در واقع به آن بخشی از سند که توسط کاربر با مکان‌نمای موس رنگی و انتخاب شده قوانین اعمال می‌کند) .

خصوصیت‌های **CSS** که در زیر آورده شده را می‌توان به **selection** اعمال کرد

## outline و cursor، background، color

به عنوان نمونه، مثال زیر متن انتخاب شده را در پس زمینه ی زرد، قرمز می کند

مثال

```
selection {  
  color: red;  
  background: yellow;  
}
```

کلیه ی شبه عناصر CSS در جدول زیر فهرست شده است

انتخاب گر	مثال	تشریح مثال
<a href="#">after</a>	<code>pafter</code>	پس از هر المان <p> محتوا درج می کند
<a href="#">before</a>	<code>pbefore</code>	پیش از تمامی المان های <p> محتوا ایجاد می کند
<a href="#">first-letter</a>	<code>pfirst-letter</code>	اولین حرف تمامی تگ های <p> را گزینش می کند
<a href="#">first-line</a>	<code>pfirst-line</code>	اولین خط کلیه ی عناصر <p> را انتخاب می نماید
<a href="#">selection</a>	<code>pselection</code>	آن بخشی از المان را انتخاب می کند که توسط کاربر گزینش شده باشد

## سیتنکس pseudo-elements

```
selectorpseudo-element {property:value;}
```

در کلاس های CSS را می توان از **pseudo-elements** نیز استفاده کرد

```
selector.classpseudo-element {property:value;}
```

## first-line Pseudo-element

از **first-line pseudo-element** برای اضافه کردن استایل ویژه به اولین خط متن استفاده می شود.

در مثال زیر، مرورگر خط اول متن را در عنصر **p**، مطابق با استایل "خط اول **pseudo-element**" آرایش می کند.

نکته مهم "خط اول **pseudo-element**" را فقط می توان با عنصرهای **block-level**

استفاده کرد.

نکته مهم پراپرتی های زیر به خط اول **pseudo-element** اعمال می شوند

font properties

color properties

background properties

word-spacing

letter-spacing

text-decoration

vertical-align

text-transform

line-height

clear

## first-letter Pseudo-element

از "حرف اول **pseudo-element**" برای اضافه کردن استایلی ویژه به اولین حرف متن استفاده می شود.

```
pfirst-letter
{
Color:#ff0000;
font-size:xx-large;
}
```

نکته مهم حرف اول **pseudo-element** را فقط می توان با عنصرهای **block-level**

استفاده کرد.

نکته مهم پراپرتی های زیر به حرف اول **pseudo-element** اعمال می شوند

font properties

color properties

background properties

margin properties

padding properties

border properties

text-decoration

("vertical-align (only if "float" is "none

text-transform

line-height

float

clear



## Pseudo-elements و کلاس های CSS

**Pseudo-elements** را می توان با کلاس های **CSS** ترکیب کرد

```
p.articlefirst-letter {color:#ff0000;}  
<p class="article">A paragraph in an article</p>
```

مثلا بالا حرف اول همه پاراگراف ها با **class=article** را قرمز نشان می دهد.

### Pseudo-elements چندگانه

می توان چندین **pseudo-elements** را نیز با هم ترکیب کرد.

در مثال زیر، حرف اول پاراگراف قرمز خواهد شد، با سایز فونت **xx-large**. بقیه خط اول، آبی و بصورت حروف کوچک خواهد بود. بقیه پاراگراف دارای اندازه فونت و رنگ پیش فرض خواهد بود.

```
pfirst-letter  
{  
Color:#ff0000;  
font-size:xx-large;  
}  
pfirst-line  
{  
Color:#0000ff;  
font-variant:small-caps;  
}
```



before Pseudo-element

**before pseudo-element** را می توان برای **insert** کردن مقدار یک **content** قبل از **content** یک عنصر

استفاده کرد.

مثلا زیر، تصویری را قبل از هر عنصر درج کرد.

```
h1before  
{  
Content:url(smiley.gif);  
}
```

### Pseudo-elements

ستون **CSS** نشان می دهد پراپرتی در کدام ورژن **CSS** تعریف می شود (**CSS1** یا **CSS2**).

CSS	شرح	نام Pseudo
2	بعد از عنصر ، Contrnt اضافه می کند.	After
2	قبل از عنصر ، Contrnt اضافه می کند.	Befor
1	یک استایل به اولین کاراکتر متن اضافه می کند.	First-Letter
1	یک استایل به اولین خط متن اضافه می کند.	First-Line

## آموزش Navigatuion

داشتن نوار پیمایش که استفاده از آن آسان باشد، برای هر وب سایتی مهم است.

با **CSS** می توان منوهای کسل کننده **HTML** را تبدیل به نوارهای پیمایش زیبا و جذاب کرد.

نوار پیمایش – لیست لینک ها

نوار پیمایش نیاز به **HTML** استاندارد بعنوان پایه اش دارد.

ما در مثال هایمان، نوار پیمایش را از روی لیست استاندارد **HTML** می سازیم.

اساساً نوار پیمایش لیستی از لینک هاست، پس استفاده از این عنصرها بسیار مناسب است

```
<ul>
```

```
<li><a href="default.asp">Home</a></li>
```

```
<li><a href="news.asp">News</a></li>
```

```
<li><a href="contact.asp">Contact</a></li>
```

```
<li><a href="about.asp">About</a></li>
```

```
</ul>
```

حالا بیایید **bullet**ها و **margin**ها و **padding** را از لیست حذف کنیم

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

مثال بالا توضیح می دهد

**list-style-type:none** — بولت ها را حذف می کند. نوار پیمایش نیاز به لیست **marker** ها ندارد.

تنظیم **margin**ها و **padding** روی **0**، تنظیمات پیش فرض مرورگر را حذف می کند.

کد موجود در مثال بالا، کدی استاندارد است که هم در نوارهای پیمایش افقی و هم عمودی بکار می رود.

نوار پیمایش عمودی

برای ساختن نوار پیمایش عمودی، فقط به استایل کردن عنصرها نیاز داریم، بعلاوه کد بالا

```
a
{
Display:block;
Width:60px;
}
```

مثال بالا توضیح می دهد

**display:block** — لینک ها را بصورت عنصرهای **block** نمایش می دهد و کل ناحیه لینک را قابل کلیک می کند.

**width:60px** — عنصرهای **block**، کل پهنای موجود پیش فرض را اشغال می کند. ما قصد داریم **60px** پهنای تعیین کنیم.

نکته مهم همیشه پهنای عنصرها را در نوار پیمایش عمودی تعیین کنید. اگر پهنای حذف کنید، **IE** می تواند نتایج غیر منتظره ای تولید کند.

نوار پیمایش افقی

دو راه برای ایجاد نوار پیمایش افقی وجود دارد. استفاده از **list item**های **inline** و **floating**.



هر دو متد بخوبی کار می کند، اما اگر می خواهید لینک ها دارای یک سایز باشند، باید از متد **floating** استفاده کنید.

## Inline List Item

یک راه ساختن نوار پیمایش افقی، تعیین عنصرها بصورت **inline** است، بعلاوه کدهای استاندارد بالا

```
li
{
  Display:inline;
}
```

مثال بالا توضیح می دهد که

**display:inline** — بطور پیش فرض ، عنصرهایی **block** هستند. در اینجا ما **break** های خط را قبل و بعد از هر **list item** حذف می کنیم تا آنها را روی یک خط نمایش دهیم.

## Floating list item

در مثا بالا، لینک ها پهناهای متفاوتی دارند. برای همه لینک هایی که قرار است پهناي مساوی داشته باشند، عنصر را حرکت دهید و پهناي بر ای عنصر تعیین کنید

```
li
{
  Float:left;
}
a
{
  Display:block;
  Width:60px;
}
```

مثال بالا توضیح داد که

**floatleft** — از **float** بر ای حرکت دادن عنصرهای **block** در کنار یکدیگر استفاده کنید

**display:block** — لینک ها را بصورت عنصرهایی نمایش می دهد که کل ناحیه لینک را قابل کلیک می کند، و به ما اجازه تعیین پهنا را می دهد

**width60px** — از انجاییکه عنصرهای block، کل پهنای موجود را اشغال می کنند، نمی توانند در کنار هم حرکت کنند. ما پهنای لینک ها را **60 px** تعیین می کنیم.

## آموزش گالری عکس

با استفاده از **CSS** می توان یک گالری یا آلبوم تصویر ایجاد کرد .



### آلبوم تصویر

گالری تصویری که مشاهده می کنید با **CSS** ایجاد شده است.

### ایجاد تصاویر شفاف در CSS

ایجاد تصاویر شفاف (**transparent**) در **CSS** امری بسیار آسان است.

خاصیت **opacity** در **CSS** یکی از ویژه گی ها و امکانات جدید ویرایش **CSS3** می باشد.

خاصیتی که برای شفاف سازی تصاویر در **CSS3** تعبیه شده، **opacity** می باشد.

ابتدا نحوه ی ایجاد تصویر شفاف در **CSS** را به شما خواهیم آموخت.

### تصویر در حالت عادی آن



تصویر فوق که شفاف سازی به آن اعمال شده



حال به کد CSS در مثال زیر توجه کنید

مثال

```
img {  
  opacity: 0.4;  
  filter: alpha(opacity=40); /* For IE8 and earlier */  
}
```

IE9، Firefox، Chrome، Opera و Safari نیز از خاصیت **Opacity** به منظور شفاف سازی تصویر بهره می گیرند. خاصیت **opacity** می تواند مقداری از 0.0 تا 1.0 - بگیرد. مقدار بیشتر باعث می شود، المان شفاف تر گردد.

IE8 و ویرایش های قدیمی تر آن همگی از **filter:alpha(opacity=x)** بهره می گیرند. **x** می تواند مقداری از 0 تا 100 بگیرد. هرچه مقدار بیشتر باشد عنصر مربوطه شفاف تر می گردد.

### – شفافیت تصویر – Hover Effect

در این بخش منظور از **Hover Effect** این است که در صورت قرار دادن اشاره گر موس روی تصویر، تصویر شفاف تر شود.

کد CSS آن به ترتیب زیر نوشته می شود

```
img {
  opacity: 0.4;
  filter: alpha(opacity=40); /* For IE8 and earlier */
}
imghover {
  opacity: 1.0;
  filter: alpha(opacity=100); /* For IE8 and earlier */
}
```

قطعه کد (block) اول در این مثال درست مشابه کدی است که در مثال 1 بکار برده شد. علاوه بر آن در این مثال، کد را چنان تعریف کرده ایم که اگر کاربر مکان نمای موس را روی تصویر مربوطه قرار دهد، تصویر مزبور از حالت **transparent** یا شفاف بیرون آید.

کد **CSS** برای این امر **opacity=1** می باشد.

در **IE8** و نسخه های قدیمی تر (**filter:alpha(opacity=100)**) هنگامی که اشاره گر موس از روی تصویر برداشته می شود، تصویر مجدداً به حالت شفاف باز می گردد.

مثال 3 – جای گذاری نوشته در کادر شفاف داخل تصویر

کد منبع (**Source code**) این مثال به صورت زیر می باشد

مثال

```
<html>
<head>
<style>
div.background {
  background: url(klematis.jpg) repeat;
  border: 2px solid black;
}
div.transbox {
  margin: 30px;
  background-color: #ffffff;
  border: 1px solid black;
  opacity: 0.6;
  filter: alpha(opacity=60); /* For IE8 and earlier */
}
div.transbox p {
  margin: 5%;
  font-weight: bold;
  color: #000000;
}
</style>>
</head>>
body>>
```

```

<div class="background">
  <div class="transbox">
    <p>This is some text that is placed in the transparent box.</p>
  </div>
</div>
<body>
<html>

```

در مرحله ی اول، یک المان `<div>` ایجاد می کنیم (`class="background"`) و به دنبال آن یک تصویر پس زمینه و خط حاشیه (`border`). سپس، یک `<div>` دیگر (`class="transbox"`) داخل `<div>` اول ایجاد می کنیم. `<div class="transbox">` دارای یک رنگ پس زمینه و خط حاشیه (`border`) است. `div`، شفاف یا `transparent` خواهد بود. درون `<div>` شفاف، یک نوشته (`text`) داخل تگ `<p>` اضافه می کنیم.

## Image Sprite چیست؟

عبارت است از مجموعه ی از تصاویر که در یک تصویر گنجانده شده است. `sprite` خود یک تصویر دو بعدی (تصویر کوچک) است که در صحنه نمایش بزرگتری جای داده شده و مستقل از دیگر تصاویر پس زمینه قبل جا به جایی می باشد.

یک صفحه وب که دربردارنده ی تصاویر زیادی است، طبیعتاً زمانی زیادی برای بارگذاری و تولید درخواست های متعدد به سرور نیاز دارد.

با بهره گیری از `image sprite` می توان تعداد درخواست های سرور (`server request`) را کاهش داد و در مصرف پهنای باند صرف جویی کرد.

## Image Sprites – مثالی ساده

به جای استفاده از سه تصویر جداگانه، یک تصویر واحد مورد استفاده قرار می دهیم که هر سه را دربر گیرد ("`img_navsprites.gif`")



در مثال زیر `CSS` مشخص می کند کدام بخش از تصویر "`img_navsprites.gif`" نمایش داده شود

مثال

```
#home {
  width: 46px;
  height: 44px;
  background: url(img_navsprites.gif) 0 0;
}
```

## تشریح مثال

 صرفاً یک تصویر شفاف کوچک تعریف می کند، به این خاطر که خصیصه **src** خالی نیست. تصویری که نمایش داده می شود همان تصویر پس زمینه خواهد بود که در **CSS** مشخص می کنیم.

**width 46px; height 44px;** آن بخشی از تصویر را که می خواهیم مورد استفاده قرار دهیم، تعیین می کند.

**background url(img\_navsprites.gif) 0 0;** تصویر پس زمینه و موقعیت آن را مشخص می کند . (**top 0px** و **left 0px**)

این روش آسان ترین راه برای استفاده از **image sprite** است. اکنون می خواهیم با استفاده از لینک و **hover effects** آن را بسط دهیم.

## Image Sprites – ایجاد لیست پیمایشی (navigation list)

می خواهیم یک (**"img\_navsprites.gif"**) **image sprite** جهت ایجاد لیست پیمایشی بکار ببریم.

یک لیست **HTML** بکار می بریم، زیرا می تواند مثل لینک عمل کرده و همچنین از تصویر پس زمینه پشتیبانی کند.

## مثال

```
#navlist {
  position: relative;
}
#navlist li {
  margin: 0;
  padding: 0;
  list-style: none;
  position: absolute;
  top 0;
}
#navlist li #navlist a {
  height :44px;
```

```

display: block;
}
#home {
left: 0px;
width: 46px;
background: url('img_navsprites.gif') 0 0;
}
#prev {
left: 63px;
width: 43px;
background: url('img_navsprites.gif') -47px 0;
}
#next {
left: 129px;
width: 43px;
background: url('img_navsprites.gif') -91px 0;
}

```

شرح مثال

```
#navlist { position: relative; }:
```

موقعیت دهی به صورت نسبی (**relative**) تنظیم و مقداردهی می شود تا بدین وسیله امکان موقعیت دهی از نوع مطلق (**absolute**) داخل آن ممکن شود

```
#navlist li { margin: 0; padding: 0; list-style: none; position: absolute; top: 0; }:
```

**margin** و **padding** هر دو 0 مقدار دهی شده، **list-style** حذف گردیده و تمامی آیتم های لیست به صورت نسبی موقعیت دهی شده اند .

```
#navlist li #navlist a { height: 44px; display: block; }:
```

ارتفاع یا طول کلیه ی تصاویر 44px می باشد .

حال سبک دهی (**styling**) و موقعیت دهی (**positioning**) را برای هر بخش معین آغاز می کنیم .

```
#home { left: 0px; width: 46px; } :
```

در چپ ترین موقعیت (دورترین موقعیت سمت چپ) قرار داده شده و عرض یا پهنای تصویر 46px می

باشد .

```
#home {background: url('img_navsprites.gif') 0 0; } :
```

تصویر پس زمینه و موقعیت آن را تعریف می کند (**top px, left 0px**) .

```
#prev { left: 63px; width: 43px; background: url('img_navsprites.gif') -47px 0; } :
```

به اندازه ی 63px در سمت راست قرار داده می شود( کمی فضای اضافه بر سازمان بین آیتم ها **#home**

، عرض نیز 43px می باشد.

```
#prev {background: url('img_navsprites.gif') -47px 0; } :
```

تصویر پس زمینه را به اندازه ی **47px** در سمت راست قرار می دهد (**#home width 46px + 1px line divider**).

```
#next { left :129px; width: 43px; background: url('img_navsprites.gif') -91px 0; } :
```

به اندازه ی **129px** در سمت راست موقعیت دهی می شود (ابتدای **#prev** حدوداً **63px** است + **#prev width 43px** + فضای اضافه بر سازمان)، و پهنا **43px** می باشد.

```
#next {background: url('img_navsprites.gif') -91px 0; }:
```

تصویر پس زمینه را **91px** به سمت راست مقدار دهی می کند .

```
(#home width: 46px + 1px line divider + #prev width: 43px + 1px line divider ):
```

قرار گرفتن مکان نمای موس روی تصویر (**Image Sprites - Hover Effect**):  
حال می خواهیم یک رویداد **hover** به لیست پیمایشی خود اضافه کنیم.

انتخابگر (**hover(selector)** زمانی بکار می رود که نشانگر موس به منظور انتخاب آیتم مورد نظر روی آن قرار می گیرد.

**نکته** انتخابگر **hover** را می توان نه تنها روی لینک ها بلکه روی کلیه ی آیتم های بکار برد.

تصویر جدیدمان ("**img\_navsprites\_hover.gif**") حاوی سه تصویر پیمایشی (**navigation image**) و سه تصویر ویژه ی رویدادهای **hover** می باشد



به منظور افزودن رویداد **hover** کافی است تنها سه خط کد بنویسیم

مثال

```
#home ahover {  
background: url('img_navsprites_hover.gif') 0 -45px;
```



```

}

#prev ahover {
  background: url('img_navsprites_hover.gif') -47px -45px;
}

#next ahover {
  background: url('img_navsprites_hover.gif') -91px -45px;
}

```

## تشریح مثال

```
#home ahover { background: url('img_navsprites_hover.gif') 0 -45px; }
```

برای هر سه تصویری که رویداد **hover** به آن ها اعمال شده همان موقعیت پس زمینه را تعریف می کنیم، فقط **45px** پایین تر.

## انواع رسانه ها در CSS

با بهره گیری از انواع **media**، یک صفحه می تواند یک طرح ویژه ی صفحه نمایش، یکی برای چاپ، یکی مختص دستگاه های دستی و ... داشته باشد.

برخی از خصوصیت ها تنها ویژه ی یک نوع خاص **media** تعبیه شده اند. به عنوان مثال می توان به **voice-family** اشاره کرد که به عنوان **user agent** (به نرم افزاری گفته می شود که به نمایندگی از طرف کاربر عمل می کند) شنیداری عمل می کند.

برخی دیگر از خصوصیت های **CSS** را می توان برای دیگر **media type** ها مورد استفاده قرار داد. برای مثال، خاصیت **font-size** را می توان برای هر دو رسانه های مربوط به صفحه نمایش (**screen**) و چاپی (**print**) مورد استفاده قرار داد اما با مقادیری متفاوت. یک سند معمولاً بر روی صفحه نمایش در مقایسه با کاغذ به **font-size** بزرگ تری نیاز دارد. قلم نوشته ها (فونت) **sans-serif** را بهتر می توان روی صفحه نمایش خواند، این در حالی است که فونت های **serif** را بهتر می توان روی کاغذ خواند.

دستور @media امکان تعریف قوانین گوناگون سبک دهی (style rule) برای انواع مختلف media در یک صفحه ی سبک دهی واحد (style sheet) را برای طراح وب ایجاد می کند .

کد CSS که در مثال زیر بکار گرفته شده، به مرورگر دستور می دهد نوشته را با فونت Verdana در اندازه یا با مقیاس 17 پیکسل بر روی صفحه نمایش نشان دهد، ولی در صورت چاپ شدن صفحه بر روی کاغذ نوشته را با رنگ آبی و فونت Georgia به اندازه ی 14 پیکسل نمایش دهد .

مثال

```
@media screen {
  p {
    font-family: verdana, sans-serif;
    font-size: 17px;
  }
}
@media print {
  p {
    font-family: georgia, serif;
    font-size: 14px;
    color: blue;
  }
}
```

جدول زیر دیگر انواع media که در CSS کاربرد دارند را فهرست کرده

نوع media	شرح
all	برای تمامی نوع های مختلف media بکار می رود
aural	ویژه ی دستگاه های ترکیب کننده ی صدا (sound Synthesizer) بکار می رود
braille	برای دستگاه های بازخورد لمسی خط بریل مورد استفاده قرار می گیرد
embossed	برای چاپگرهایی که صفحاتی با خط بریل پرینت می کنند بکار می رود
handheld	ویژه ی دستگاه های کوچک دستی و قابل حمل مورد استفاده قرار می گیرد

print	برای چاپگرهای معمولی استفاده می شود
projection	برای دستگاه های ارائه پروژکتوری ، مانند اسلایدها مورد استفاده قرار می گیرد
screen	ویژه ی صفحه نمایش کامپیوتر طراحی شده
tty	برای رسانه های استفاده کننده از شبکه fixed-pitch ، مانند تله تایپ و ترمینال بکار گرفته می شود
tv	برای دستگاه های تلویزیون گونه کاربرد دارد

### گزینه‌گر های صفت

سبک دهی به عناصر HTML با استفاده از خصیصه های ویژه

این امکان وجود دارد که المان های HTML ی که دارای صفت مشخصه (Attribute) یا مقادیر attribute خاص هستند را سبک دهی کرد (به آن ها style اعمال کرد) .

**توجه IE7 و IE8** هر دو زمانی توانایی پشتیبانی از گزینه‌گرهای صفت را دارند که یک **DOCTYPE!** برای آن تعریف شده باشد .

### گزینه‌گر [ صفت ] CSS

از این گزینه‌گر (**[attribute] Selector**) برای انتخاب عناصری استفاده می شود که دارای خصیصه (Attribute) خاصی باشند .

به عنوان مثال، نمونه ی زیر تمامی المان های **<a>** را که دارای خصیصه ی مورد نظر هستند، انتخاب می کند

### مثال

```
a[target] {
background-color: yellow;
}
```

## گزینشگر [مفت = مقدار] CSS

این گزینشگر به منظور انتخاب المان هایی با خصیصه و مقدار مشخص شده بکار می رود .

مثال ذیل تمامی تگ های **<a>** که خصیصه ی **target="\_blank"** را دارند گزینش می کند

مثال

```
a[target="_blank"] {  
  background-color: yellow;  
}
```

## گزینشگر [مقدار =~ خصیصه] CSS

گزینشگر ذکر شده به منظور انتخاب المان هایی بکار می رود که مقدار **attribute** آن دربردارنده ی کلمه ی مشخص شده باشد .

مثال زیر تمامی المان هایی که خصیصه ی **title** آن دربردارنده ی لیستی از کلمات جدا شده توسط خط فاصله باشند (که یکی از آن ها **"flower"** می باشد) را گزینش می کند

مثال

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

مثال فوق با المان هایی که عنوان (**title**) آن **title="flower"** ، **title="summer flower"** ، **title="flower"** **new** و نه **title="my-flower"** **title="flowers"** است مطابقت داشته و مچ می شود .

## گزینشگر [خصیصه=مقدار] در CSS

گزینشگر فوق برای انتخاب آن دسته از عناصری بکار می رود که خصیصه ی مشخص شده آن با مقدار مشخص شده و مورنظر آغاز گردد.

مثال زیر تمامی المان هایی که مقدار **attribute** کلاس آن با واژه ی **"top"** آغاز می گردد را انتخاب می کند

**توجه** دقت داشته باشید که مقدار مورد نظر باید حتماً یک واژه ی کامل باشد، یا تنها و به خودی خود مانند `class="top"` باشد و یا یک خط ربط (-) به دنبال آن قرار گیرد، درست مانند این مثال `class="top-text"`

مثال

```
[class="top"] {  
  background: yellow;  
}
```

گزینشگر [خصیصه^=مقدار] در CSS

این انتخابگر به منظور گزینش عناصری بکار می رود که مقدار `attribute` یا صفت مشخصه ی آن با مقدار تعریف شده آغاز گردد .

مثال زیر تمامی المان هایی که مقدار `attribute` کلاس آن با `"top"` آغاز می گردد را انتخاب می کند **توجه** مقدار نباید لزوماً یک واژه ی یا کلمه ی کامل باشد .

مثال

```
[class^="top"] {  
  background: yellow;  
}
```

گزینشگر [attribute\$=value] در CSS

این انتخابگر برای گزینش المان هایی بکار می رود که مقدار `attribute` آن با مقدار مشخصی پایان می یابد .

به عنوان مثال نمونه ی زیر کلیه ی المان هایی که مقدار `attribute` کلاس آن با واژه ی `"test"` خاتمه می یابد را انتخاب می کند .

**توجه** مقدار لزوماً نباید یک واژه ی کامل باشد .

مثال

```
[class$="test"] {  
  background: yellow;  
}
```

## گزینشگر [attribute\*=value] و کاربرد آن در CSS

انتخابگر [خصیصه\*=مقدار] ویژه ی گزینش آن دسته از مقادیری بکار می رود که مقدار **attribute** آن دربردارنده ی مقدار مشخصی باشد .

مثال زیر کلیه ی المان هایی که مقدار **attribute** کلاس آن دربردارنده ی "te" می باشد را گزینش می کند .

**توجه** لازم نیست مقدار یک کلمه یا واژه ی کامل باشد .

مثال

```
[class*="te"] {  
  background: yellow;  
}
```

سبک دهی به فرم

گزینشگر صفت می تواند در سبک دهی به فرم بدون کلاس یا شناسه (ID) بسیار کارآمد عمل کند

مثال

```
input[type="text"] {  
  width: 150px;  
  display: block;  
  margin-bottom: 10px;  
  background-color: yellow;  
}  
input[type="button"] {  
  width: 120px;  
  margin-left: 35px;  
  display: block;  
}
```

## سلکتورهای Grouping

در **style sheet** ها، اغلب عناصری با یک استایل وجود دارند.

```
h1  
{  
  Color:green;  
}  
h2  
{  
  Color:green;  
}
```

```
p
{
Color:green;
}
```

برای کوتاه کردن کد، می توان سلکتورها را **group** کرد.

هر سلکتور را باید با یک علامت ویرگول جدا کرد.

در مثلا زیر، سلکتورهای کد بالا با هم **group** شده اند

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

Nest کردن سلکتورها

اعمال کردن استایل برای سلکتور درون سلکتوری دیگر امکان پذیر است.

در مثال زیر، یک استایل برای همه عنصرهای **p**، تعیین شده است، و استایلی دیگر نیز برای عنصرهای **p**، که درون کلاس "**marked**" تودرتو شده اند، تعیین شده است.

```
p
{
Color:blue;
text-align:center;
}
.marked
{
background-color:blue;
}
.marked p
{
Color:white;
}
```

تراز کردن عنصرهای **block**

عنصر **block**، عنصری است که کل پهنای موجود را اشغال می کند، و دارای فاصله ای (**line break**) قبل و بعد از کنترل است.

نمونه هایی از عنصر **block**

برای تراز کردن متن به قسمت اول مقاله مراجعه کنید.

در این بخش، به شما نشان می دهیم چگونه عنصر **block** را برای استفاده در **layout** تراز کنید.

تراز کردن در مرکز با استفاده از پراپرتی **margin**

عنصرهای **block** را می توان با تنظیم مارجین های چپ و راست روی **"auto"** تراز کرد.

نکته مهم استفاده از **margin:auto** در **IE** کار نمی کند مگر اینکه یک **DOCTYPE** تعریف شود.

تنظیم **margin** های چپ و راست روی **auto**، تعیین می کند که آنها باید **margin** موجود را بطور مساوی تقسیم کنند. نتیجه این کار، عنصری در مرکز خواهد بود

```
.center
{
margin-left:auto;
margin-right:auto;
width:70%;
background-color:#b0e0e6;
}
```

اگر پهنا **100%** باشد، تراز کردن اثر نخواهد داشت.

در **IE**، یک **margin handling bug** برای عنصرهای **block** وجود دارد. برای اینکه مثال های بالا در **IE5** کار کنند، باید کدهای بیشتری وارد کنید.

تراز چپ و راست با استفاده از پراپرتی **Position**

یکی از متدهای تراز کردن عنصرها، استفاده از موقعیت یابی مطلق (**absolute positioning**) است

```
.right
{
Position:absolute;
Right:0px;
Width:300px;
background-color:#b0e0e6;
}
```

نکته مهم عنصرهای موقعیت مطلق، از جریان نرمال حذف می شوند، و ممکن است با عناصر دیگر تداخل داشته باشند.



هنگام تراز کردن عناصری مانند این، از پیش تعریف کردن **margin** و **padding** برای عنصر ایده خوبی است. این کار به معنای پرهیز از تفاوت‌های **visual** در مرورگرهای مختلف است.

**IE** هنگام استفاده از پراپرتی **position** مشکل دیگری نیز دارد. اگر عنصر دربرگیرنده (در مثال ما، **<div class="container">**)

دارای پهنایی مشخص است، و هیچ **DOCTYPE**! تعریف نشده است؛ **IE** یک مارجین **17 px** در سمت راست اضافه خواهد کرد. ظاهراً این فضا برای **scrollbar** رزرو شده است. هنگام استفاده از پراپرتی **position**، **DOCTYPE**! را تعریف کنید

```
body
{
  Margin:0;
  Padding:0;
}
.container
{
  Position:relative;
  Width:100%;
}
.right
{
  Position:absolute;
  Right:0px;
  Width:300px;
  background-color:#b0e0e6;
}
```

تراز چپ و راست با استفاده از پراپرتی **float**

یکی از متدهای تراز کردن عناصرها، استفاده از پراپرتی **float** است

```
.right
{
  Float:right;
  Width:300px;
  background-color:#b0e0e6;
}
```

## مسئله سازگاری Crossbrowser

هنگام تراز کردن عناصر ، از پیش تعریف کردن **padding** و **margin** برای عنصر ایده خوبی است. این کار به معنای پرهیز از تفاوت‌های **visual** در مرورگرهای مختلف است.

**IE** هنگام استفاده از از پراپرتی **float** مشکل دیگری نیز دارد. اگر هیچ **DOCTYPE** تعریف نشده باشد، **IE** یک مارجین **px 17** در سمت راست اضافه خواهد کرد. ظاهراً این فضا برای **scrollbar** رزرو شده است. هنگام استفاده از پراپرتی **float** ، **DOCTYPE** را تعریف کنید

```
body
{
  Margin:0;
  Padding:0;
}
.right
{
  Float:right;
  Width:300px;
  background-color:#b0e0e6;
}
```

در پایان ضمن تشکر از انتخاب شما، امیدواریم مطالب این کتاب برای شما مفید بوده باشد.

علاوه بر این می‌توانید پیشنهادات و انتقادات خود را از طریق رایانامه [Book.tahlildadeh@gmail.com](mailto:Book.tahlildadeh@gmail.com) با ما در میان بگذارید.