جلسه اول

مقدمه

بخش مجزای دیگری که در طراحی صفحات وب از آن استفاده می شود، زبان CSS است.

CSS یا همان Sheet Style Cascading (سبک نامه آبشاری) روشی است برای قالب بندی اجزای صفحه از قبیل متن، تصویر، زمینه صفحه، کادرها و تقریبا هر چیزی که در طراحی صفحه HTML استفاده می شود.
با استفاده از CSS می توان از دوباره نویسی کدهای HTML که باعث مشکل شدن کدنویسی و همچنین کندشدن سرعت بارگذاری صفحه در اینترنت می شود، جلوگیری کرد. برای مثال فرض کنید ما ۳۰ صفحه HTML داریم و در هرکدام از ۱۰ تگ پاراگراف استفاده کرده ایم، که در مجموع می شود ۳۰۰ تگ پاراگراف، حال اگر بخواهیم برای همه این پاراگراف ها یک استایل را تعریف کرد باید از ۳۰۰ تگ فونت دیگر نیز استفاده کرد، اما با استفاده از CSS می توان این ۳۰۰ تگ فونت را در یک خط خلاصه کرد!! که هم باعث کم شدن حجم صفحات و نیز افزایش سرعت بارگذاری صفحات خواهد شد.
یکی دیگر از مزیت های CSS قابلیت های بسیار زیاد آن است، که با رسیدن CSS3 حتی در بعضی مواقع می توان از آن بجای JavaScript و jQuery نیز استفاده کرد. در این سری آموزشی سعی خواهد شد شما را با این زبان شگفت انگیزکه یکی از لازمه های اجتناب ناپذیر برای تبدیل شدن به یک طراح وب است، آشنا کنیم. اما یک پیش نیازی که برای یادگیری CSS لازم دارید، این است که شما حداقل اطلاعات پایه ای با زبان HTML را دارا، و به خوبی با تگ های HTML آشنایی داشته باشید.
این سری آموزشی با پوشش آخرین نسخه از این زبان، یعنی CSS3 این قول را به شما خواهد داد که پس از به پایان رساندن آموزش ها، شما قادر خواهید بود بدون کمک دیگران، یک وبسایت را طراحی کنید.یک نکته دیگر اینکه برای کدنویسی زبان css از هر ویرایشگری می توان استفاده کرد اما پیشنهاد من نرم افزار ripid css است که می توانید آن را از لینک زیر دریافت کنید

دانلود نرم افزار ripid css

این سری آموزشی در ۱۵ فصل ارائه خواهد شد ، و هر سه روز یک بار مطلب جدید افزوده می شود.
با ما باشید.....

جلسه دوم

روش های استفاده از CSS در اسنادHTML

برای وارد کردن کدهای CSS در یک صفحه HTML چهار روش وجود دارد :

1.قرار دادن دستورات CSSدر داخل فایل HTML(استایل داخلی)

```
<html>

<head>

<style type="text/css">

...
```

```
.....

</style

</head>

<body>

</body>

</html>
```

در این روش باید کد فوق را در بالای سند HTML و در داخل تگ head قرار بگیرد و بجای نقطه چین ها نیز دستورات Css قرار می گیرند.

.2قرار دادن دستورات cssدر یک فایل جداگانه با پسوند .cssو استفاده از تگ <link /> برای ارتباط دادن این فایل به فایل (HTMLاستایل خارجی)

```
1  <html>
2   <head>
3     <link rel="stylesheet" type="text/css" href="style.css" />
4                                                    </head>
5   <body>
6   </body>
7 </html>
```

در کد بالا یک فایل css خارجی با نام Style.css به صفحه لینک شده است. کد فوق باید در بالای سند Html و در داخل تگ head قرار بگیرد. دقت کنید که مقدار صفت href باید آدرس فایل باشد.

.3دستورات cssمی توانند به طور مستقیم در داخل عناصر HTMLو با استفاده از صفت عمومی styleدر تگ های HTMLاستفاده شوند.

```
1  <html>
2   <head>
3                                  </head>
```

```
4    <body style="font: 12px Tahoma;">
5    </body>
6   </html>
```

4.با استفاده از دستور @importمی توان دستورات css را به هر دو صورت استایل داخلی و استایل خارجی در یک فایل وارد کرد:

```
1   <style type="text/css">
2    @import url(style.css)
3 </style>
```

اضافه کردن توضیح (کامنت) درCSS

نحوه نوشتن comment در CSS شبیه به زبان C++ بوده و به این صورت است که با علامت (/*) شروع و با علامت (*/) به پایان می رسد :

```
1 /* This is a Comment */
2        /* This is Another
3 Comment */
```

اضافه کردن کامنت ها در css به طراح وب کمک می کند که سریعتر کدهای هر بخش را پیدا کند.

قسمت های یک دستورCSS

در زبان CSS برای نسبت دادن استایل خاصی به یک قسمت از اجزای صفحه وب، اول باید آن را انتخاب کرد و سپس به تعریف استایل نمود. به طور مثال :

```
1              H2 {
2 font-size: 24px;
3 }
```

با توجه به مثال بالا به طور کلی هر دستور css دارای ۳ بخش است :

۱. **گزینشگر (یا همان selectors) :** که در اصل همان تگ های html هستند و یا کلاس ها (class) و یا شناسه ها(id) یی هستند که شما برای یک تگ در صفحه html تعریف کرده اید. در مثال فوق h2 گزینشگر بوده و مشخص می کند که این استایل موردنظر باید به تمامی تگ های <h2>اعمال شود.

۲. **خصوصیات (یا همان : (prorerties**که مشخص می کنند که چه سبکی باید به برچسبهای انتخاب شده اعمال شود.یا به عبارتی دیگر کدام خصوصیت آن تگ، مورد تغییر قرار گیرد. در مثال فوق خصوصیت font-size اندازه فونت تگ h2 را تغییر می دهد.

۳. **مقادیر (یا همان : (value**سومین قسمت از هر دستور مقدار است. که مشخص می کند مقدار خصوصیت انتخاب شده چه باشد. اگر به مثال بالا دقت کنید می بینید که ۲٤ pxمقدار خصوصیت انتخاب شده می باشد، که تعیین می کند اندازه فونت همه ی تگ های <h2>بیست و چهار پیکسل باشد.

دستور css قسمت چهارمی نیز دارد آن هم علامت ; (سمیکولن) است که در انتهای هر دستور می آید.

نکته : اگر می خواهید بیش از یک خصوصیت از یک تگ را تغییر دهید، حتما باید برای فاصله انداختن بین propertyها از این علامت استفاده کرد.

درضمن به یاد داشته باشیدکه خصوصیت ها حتما باید ما بین علامت {} قرار بگیرند.

برای درک بهتر مطلب به تصویر زیر دقت کنید :



بسیار خب تا اینجا با اصول اولیه زبان CSS، طریقه افزودن کدها، و اجزای یک دستور CSS آشنا شدید. در جلسه بعدی شما را با انواع گزینشگرها آشنا خواهیم کرد.

جلسه سوم

گزینشگرهایCSS

در زبان CSS از ٦ طریق می توان یک تگ را انتخاب کرد یا به عبارتی ٦ نوع گزینشگر وجود دارد :

۱. **گزینشگرهای برچسب(Tag Selectors)**
۲. **گزینشگرهای کلاس کاذب(Pseudo-Class Selectors)**
۳. **گزینشگرهای کلاس(Class Selectors)**
٤. **گزینشگرهای مفهومی(Contextual Selectors)**
٥. **گزینشگرهای خصوصی(ID Selectors)**
٦. **گزینشگر عمومی(Universal Selector)**

حال به تشریح هر کدام از این گزینشگرها به همراه مثال می پردازیم :

۱. گزینشگرهای برچسب : این گزینشگرها همان نام تگ های HTML می باشند. در این حالت نام تگ را انتخاب می کنیم و استایل مورد نظرمان را برای آن تعریف می کنیم، در این روش استایل بر همه ی تگ های از این نوع در کل صفحه اعمال می شود. و بیشتر در مواقعی استفاده میشود که بخواهیم یک استایل پیشفرض برای تگ ها تعریف کنیم :

```
h1 , h2 , h3 , h4 , h5 , h6 {
                color: #000;
font-family: Tahoma;
                }
```

2.گزینشگرهای کلاس کاذب : این سیلکتورها برای نمایش رویدادهای پویا مورد استفاده قرار می گیرند ؛ این رویدادها می توانند مثل حرکت موس یا کلیک بر روی عنصر باشند. کلاس های کاذب به طراح، آزادی عمل می دهند که چگونگی رفتار عنصر در شرایط مختلف را تعیین کند.

طریقه استفاده از کلاس های کاذب به اینگونه است که پس از نام گزینشگر یک علامت : (دو نقطه) و سپس نام یکی از کلاس های کاذب می آید. کلاس های کاذب مختلف به همراه توضیحاتشان را می توانید در تصویر زیر مشاهده کنید:

کلاس کاذب	توضیحات
:link	اشاره به لینک های بازدید نشده دارد.
:visited	اشاره به لینک های بازدید شده دارد.
:hover	اشاره به عناصری دارد که در حال حاضر اشاره گر موس کاربر روی آن ها شناور است.
:focus	اشاره به عناصری دارد که focus روی آن ها است.
:active	اشاره به عنصری که کاربر روی آن کلیک کرده است.

توجه داشته باشید که دو کلاس link و visited فقط برای تگ های <a> استفاده می شوند و بقیه کلاسهای کاذب را می توان برای همه ی تگ ها استفاده کرد.

درضمن ترتیب قرارگیری این کلاسها نیز بسیار مهم است و اگر طبق ترتیبی در ادامه خواهیم گفت قرار نگیرند، استایل های تعریف شده به درستی کار نخواهند کرد. ترتیب آن ها به صورت زیر باید باشد:

```
Selector:link
Selector:visited
Selector:hover
Selector:active
Selector:focus
```

نکته : یک روش برای به خاطر سپردن ترتیب کلاسهای کاذب، به خاطر داشتن کلمه ی LoVe HAte است. (به ترتیب حروف بزرگ در کلمه دقت کنید) این کلمه به آن معناس که اول link سپس visited پس از آن hover سپس active و در آخر نیز focus قرار می گیرد.

به مثال زیر دقت کنید :

```
a:link { color: #0000ff; }
a:visited { color: #ff00ff; }
a:hover { color: #ff0000;
}
a:active { color: #00ff00; }
```

3.گزینشگرهای کلاس : تا به اینجا شما یاد گرفتید که چگونه برای هر تگی از تگ های HTML استایل مورد نظر را تعریف کنید، اما اگر خواستید به دو تگ یکسان دو استایل مختلف را اختصاص دهید باید چه کنید ؟؟؟؟؟؟ فرض کنید ما کد HTML زیر را داریم:

```
<p> 7Learn : </p>
<p> JavaScript and Web Design Tutorials </p>
```

حال می خواهیم learn۷ به رنگ قرمز و javascript and web design tutorials به رنگ آبی باشد:

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

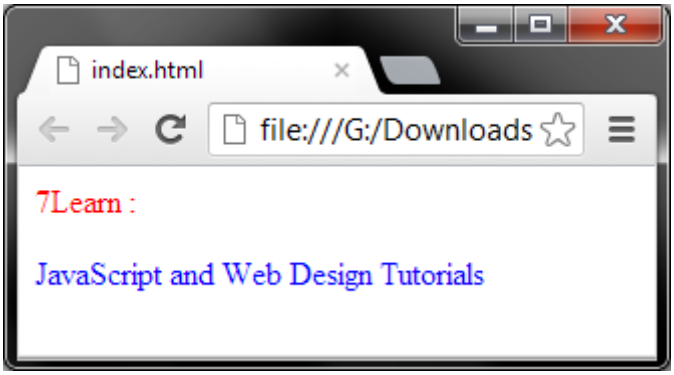
اما این کد هر دو کلمه را به رنگ آبی در می آورد؛ اینجاست که باید از کلاس استفاده کرد تا بتوانیم به هر یک از تگ ها، حتی اگر یکسان باشند، استایل مختلفی را اختصاص داد. مانند کد زیر :

```
<p class="red"> 7Learn : </p>
<p class="blue"> JavaScript and Web Design Tutorials </p>
```

حال می توانیم برای هر یک از کلمه ها استایل مورد نظر را تعریف کرد :

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

نتیجه کد در مرورگر :



:: چند نکته در استفاده از کلاس ها :

۱. در انتخاب نام کلاس ها باید دقت داشته باشید که اولین حرف نباید عدد یا نقطه و یا علامت ؟ باشد مثلا کلاس learn ۷اشتباه است و هنگام تعریف استایل کار نخواهد کرد.

۲. هنگام نوشتن نام کلاس در زبان CSS قبل از نام باید یک . (نقطه) قرار دهید مانند مثالی که در بالا زده شد.

۳. شما می توانید بیش از یک کلاس برای یک تگ تعریف کنید به این صورت که بین هر نام یک فاصله (space) قرار دهید، مانند مثال زیر از ۳ کلاس استفاده شده:

```
<p class="red big right"> 7Learn.com </p>
```

4.گزینشگرهای مفهومی :هنگام تعریف یک کلاس برای یک تگ، اگر تگ دیگری (همسان یا غیر همسان) همان نام کلاس را نیز داشته باشد استایل تعریف شده به آن تگ نیز اختصاص داده خواهد شد. به مثال زیر دقت کنید:

```
<div>
<a href="#"> 7Learn.com </a>
<a href="#"> 7Learn.com </a>
<a class="sidebar" href="#"> 7Learn.com </a>
<a class="sidebar" href="#"> 7Learn.com </a>
</div>
```

فرض کنید استایل زیر را نیز برای آن ها تعریف کرده باشیم :

```
.sidebar {
color: red;
font-size: 14px;
}
```

همانطور که می بینید اینکار بسیار خسته کننده خواهد بود اگر تعداد زیادی لینک را داشته باشیم و بخواهیم به همه ی آنها کلاس sidebar را اضافه کنیم. برای صرفه جویی هم در وقت و هم در کدنویسی می توان کدهای بالا را به شکل زیر تغییر داد:

```
<div class="sidebar">
<a href="#"> 7Learn </a>
<a href="#"> 7Learn </a>
<a href="#"> 7Learn </a>
<a href="#"> 7Learn </a>
</div>
```

سپس در CSS از این گزینشگر استفاده کرد.

```
div.sidebar a {
color: red;
font-size: 14px;
}
```

این گزینشگر به آن معناس که همه ی تگ های a که در divی با کلاس sidebar هستند را انتخاب کن.

به اینگونه گزینشگر ها، گزینشگرهای مفهومی گفته می شود.

۵. **گزینشگرهای خصوصی :** اینگونه گزینشگرها همانند گزینشگرهای کلاس هستند با این تفاوت که برای تعریف آنها باید از صفت id در html استفاده کرد. مانند زیر:

```
1 <p id="red"> 7Learn: JavaScript and Web design Tutorials </p>
```

و در هنگام تعریف استایل برای انتخاب نام id که تعریف شده، قبل از نام id باید یک علامت # قرار داده شود:

```
1 #red { color: red; }
```

٦ . **گزینشگر عمومی :** این گزیشگر موقعی استفاده می شود که بخواهید همه ی تگ های صفحه html تان دارای خصوصیت هایی مشابه باشند، به طور مثال اگر زبان صفحه html ما فارسی و راست باشد، و بخواهیم فونت همه ی تگ ها Tahoma باشد بجهی اینکه نام تک تک تگ ها را بنویسیم از این گزینشگر استفاده می کنیم که علامت آن * (ستاره) می باشد :

```
1          * {
2 Color: #000;
3 Font-family: Tahoma;
4                    }
```

این کد باعث می شود که فونت همه ی تگ ها Tahoma و رنگ متن همه ی تگ ها نیز سیاه باشد.

بسیار خب، تا اینجا با طریقه ی انتخاب یک تگ با استفاده از این ٦ گزینشگر ، آشنا شدید در جلسه بعدی که فصل دو خواهد بود ، شما را با خصوصیت های قالب بندی متن آشنا خواهیم کرد.

جلسه چهارم

خصوصیتColor

با استفاده از این خاصیت می توان رنگ متن عناصر را تغییر داد. و به پنج طریق می توان آن را مقدار دهی کرد.

- **استفاده از کلمات کلیدی :** می توان از ١٦ رنگ استاندارد برای مقدار دهی به color استفاده کرد عبارتند از:

aque	black	blue	fuchsia
gray	green	lime	maroon
navy	olive	purple	red
silver	teal	white	yellow

مثال:

```
1 P { color: lime; }
```

- **استفاده از مبنای هگزا دسیمال :** رنگ ها را می توان بر مبنای عبارت متناظر آنها در مبنای ۱۶ به کار برد. در این عبارت می توان سه عدد در مبنای ۱۶ نوشت. عدد اول معرف رنگ قرمز، عدد دوم معرف رنگ سبز و عدد سوم معرف رنگ آبی است. فرم کلی این عبارت ها به شکل #rrggbbاست. برای مثال #000000معرف رنگ سیاه #ffffff معرف رنگ سفید و #۰۰۰۰ ffمعرف رنگ آبی است.

مثال :

```
1 P { color: #ff00aa; }
```

- **استفاده از کوتاه نویسی هگزادسیمال :** در این روش زوج مقادیر تکراری را می توان به صورت تکی نوشت. برای مثال fff معادل ffffff ، و ۰۰۰ معادل ۰۰۰۰۰۰ و ccddaa معادل cda می باشد.

مثال :

```
1 P { color: #af0; }
```

- **استفاده از تابع rgb :** تابع rgb به ترتیب سه رنگ قرمز (red) ، سبز (green) ، آبی (blue) را به عنوان مقدار می پذیرد. این مقادیر یا اعداد، صحیح هستند. (از ۰ تا ۲۵۵) و یا به صورت درصد می باشند. (از ۰ تا ۱۰۰ درصد).

مثال :

```
1 P { color: rgb(120,200,40); }
```

- **استفاده از تابع rgba :** این تابع که در css3 معرفی شد، همانند تابع rgb است با این تفاوت که پس از اختیار سه مقدار اول که برای تعیین رنگ به کار می روند، مقدار چهارمی نیز را باید اختیار کرد که مقدار شفافیت (opacity) رنگ را تعیین می کند. و مقداری بین ۰ تا ۱ می پذیرد. که مقدار ۰ به معنای ۰ درصد و ۱ به معنای ۱۰۰ درصد است.

مثال:

```
1 P {color: rgba(120,140,130,.25); }
```

برای دیدن همه رنگ های استاندارد و مقدار هگزادسیمال و rgb آنها به لینک زیر بروید

لینک صفحه ی رنگ های استاندارد

خصوصیتbackground-color

با استفاده از این خاصیت می توان رنگ پس زمینه عناصر را تغییر داد. و مقداری که می گیرد همانند مقداری است که خصوصیت color می گیرد، یعنی همان، تابع rgb و rgba، هگزادسیمال و رنگ های استاندارد.

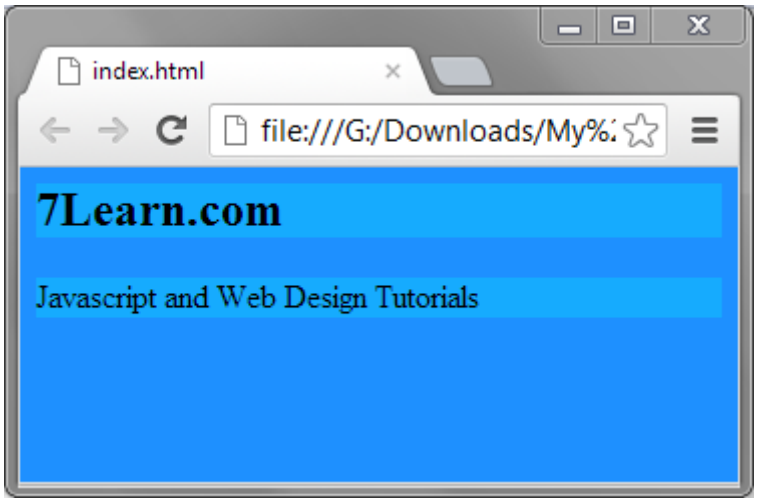
مثال : استفاده از خاصیت color و background-color

```
1                <h2> 7Learn.com </h2>
2 <p> Javascript and Web Design Tutorials </p>
```

و کد css

```
1 Body { background-color: #1e90ff; }
2                                 H2 , p {
3                                   color: #000;
4    background-color: rgba(0,255,255,.25);
5 }
```

نتیجه در مرورگر:



خصوصیتbackground-image

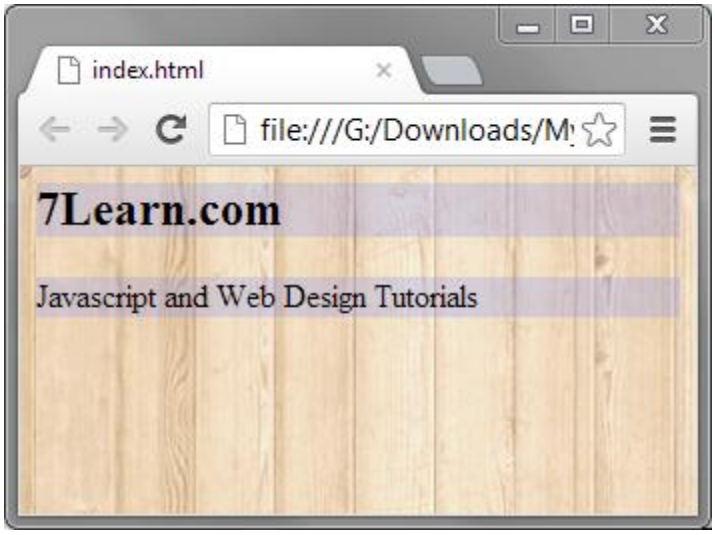
با استفاده از این خاصیت می توان برای عناصر پس زمینه تعریف کرد. و قاعده کلی آن به صورت زیر است :

```
1 Background-image: url('Image Address');
```

و بجای کلمه Image Address آدرس تصویر قرار می گیرد.

به طور مثال:

```
1  Body { background-image: url('images/bg.png'); }
2 P , h2 { background-color: rgba(120,140,250,.25); }
```



خصوصیتbackground repeat

با استفاده از این خاصیت می توان نحوه تکرار عکس در پس زمینه را کنترل کرد و یکی از چهار مقدار زیر را می پذیرد:

- Repeat : تکرار عکس در دو جهت افقی و عمودی
- Repeat-x : تکرار عکس در جهت افقی(horizental)
- Repeat-y : تکرار عکس در جهت عمودی(vertical)
- No-repeat : عدم تکرار عکس

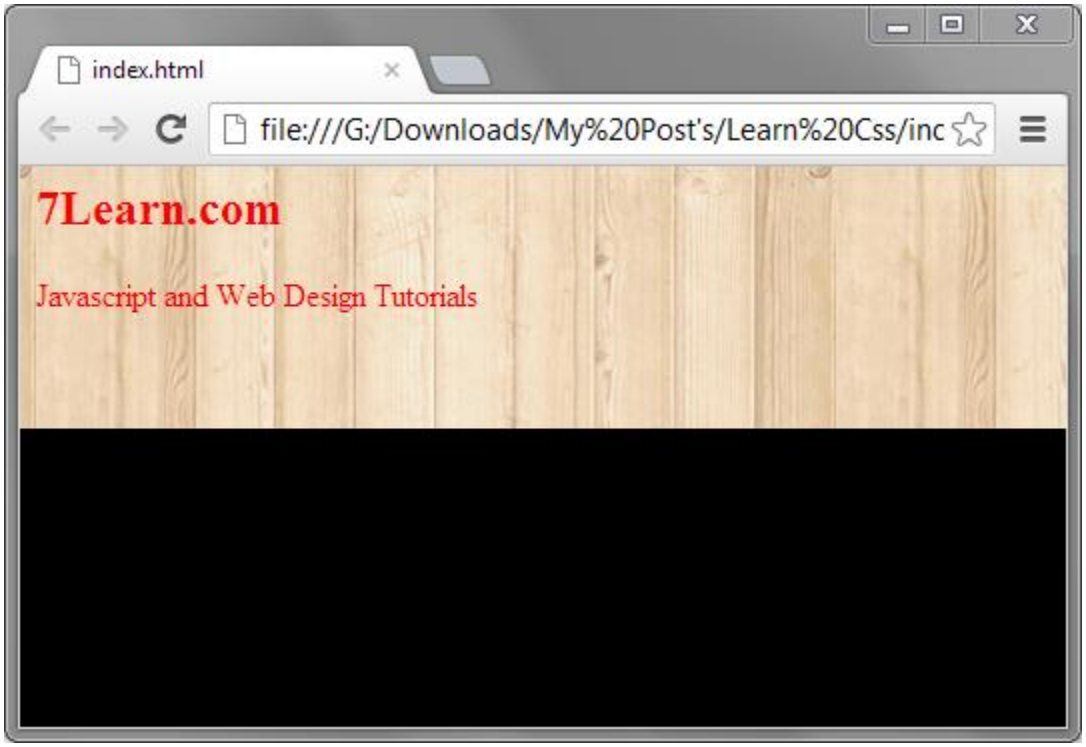
مثال:

```
1                    <h2> 7Learn.com </h2>
2 <p> Javascript and Web Design Tutorials </p>
```

و کد css :

```
1                                Body {
2   Background-image: url('images/bg.png');
3   Background-repeat: repeat-x;
4       Background-color: #000;
5   Color: #f00;
6               }
```

نتیجه در مرورگر:



می بینید که تصویر فقط در جهت افقی (عرض صفحه) تکرار شده...

خصوصیتbackground-attachment

با استفاده از این خاصیت می توان مشخص کرد که با اسکرول شدن محتوای صفحه عکس پس زمینه ی آن نیز اسکرول شود یا اینکه ثابت بماند، و یکی از دو مقدار زیر را می گیرد :

- Fixed : با اسکرول شدن محتوا، عکس پس زمینه در جای خود ثابت بماند.
- Scroll : با اسکرول شدن محتوا، عکس پس زمینه نیز اسکرول شود. (مقدار پیشفرض)

مثال :

```
<h2> 7learn.com </h2> * <h2> 7learn.com </h2> * <h2> 7learn.com </h2> * <h2> 7learn.com </h2> * <h2> 7learn.com </h2> …..
```

و کد سی اس اس:

```
Body {
Background-color: blue;
Background-image: url('images/bg.png');
Background-repeat: repeat-x;
Background-attachment: fixed;
}
```

نتیجه کد

خصوصیتbackground position

با استفاده از این خاصیت می توان موقعیت قرار گرفتن عکس در پس زمینه ی عنصر را مشخص کرد. و یکی از سه مقدار کلیدی ، درصدی و عددی را می پذیرد:

- کلمات کلیدی:

مقادیر افقی: left , center , right

مقادیر عمودی : top , center , bottom

مثال :

```
1 Background-position : left center;
```

- مقادیر درصدی : که در اول باید موقعیت افقی و سپس موقعیت عمودی ذکر شود.

مثال:

```
1 Background-position : 20% 65%;
```

دستور فوق مشخص می کند که تصویر پس زمینه باید ۲۰% به سمت داخل در جهت افقی(به سمت راست) و ٦٥% به سمت داخل در جهت عمودی (به سمت پایین) قرار بگیرد.

- مقادیر عددی : این مقادیر نیز عددی بر حسب یکی از واحد های محاسباتی در زبان css هستند:

مثال :

```
1 Background-position : 5px 10px;
```

دستور فوق مشخص می کند که تصویر پس زمینه باید از گوشه ی بالا و سمت چت ٥ پیکسل به طرف راست ، و ۱۰ پیکسل به طرف پایین قرار بگیرد.

ترکیب مقادیر عددی و درصدی و همچنین استفاده از مقادیر منفی برای این خصوصیت مجاز است، اما مقادیر عددی و درصدی نمی توانند با مقادیر کلیدی ترکیب شوند:

مثال :

```
1 Background-position : 20% 30px;
```

کوتاه نویسی خصوصیت های خاصیتbackground

همه ی خاصیت های فوق که مربوط به خصوصیت background هستند را می توان با هم در یک خط ادغام کرد.

قاعده کلی :

```
1 Background: background-color | background-image | background-repeat | background-attachment | background-position;
```

هر پنج مقدار ممکن برای background را به طور همزمان می توان استفاده کرد. درضمن ترتیب قرار گرفتن خاصیت ها نیز مهم نمی باشد. و اگر مقداری برای یکی از خاصیت ها ذکر نشود مقدار پیشفرض آن خاصیت در نظر گرفته می شود.

مثال:

```
1 Body {
2 Background: #0c0 url("images/bg.png") no-repeat fixed bottom right;
```

3 }

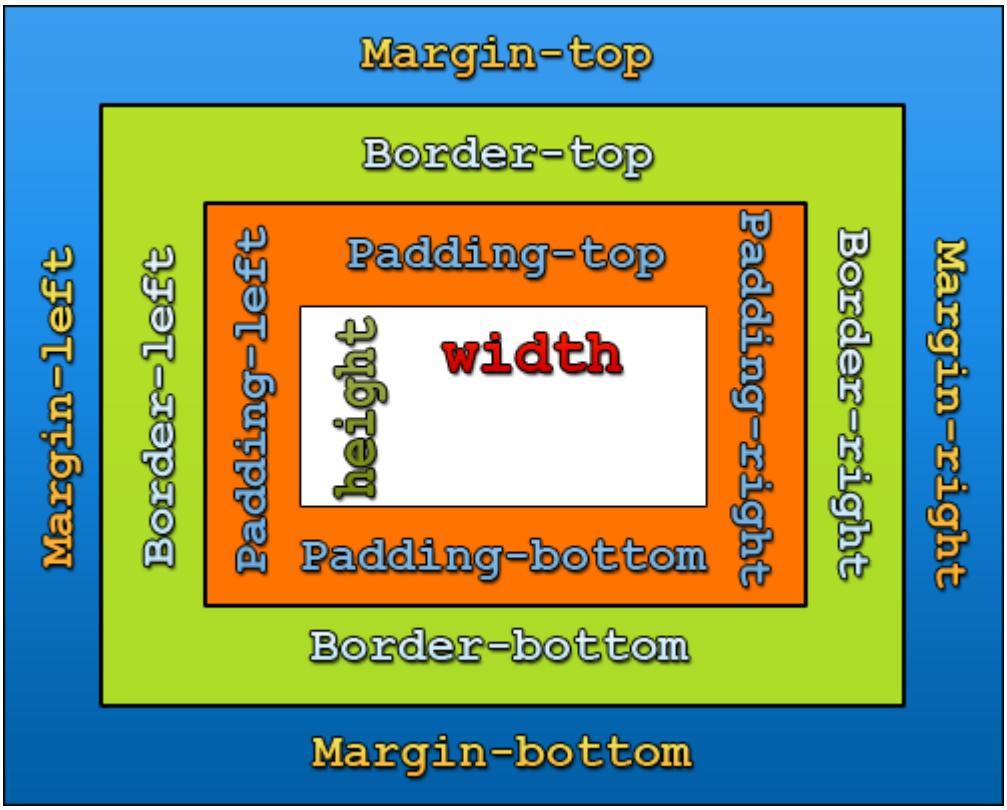
بسیار خب تا به اینجا کار ما با کار با پس زمینه به پایان رسید، در قسمت بعدی به کار با فضاهای موجود اطراف عناصر خواهیم پرداخت.

فصل پنجم

سلام دوستان. امیدوارم که حالتون خوب باشه، مدتی بود سرگرم کاری بودم به همین دلیل آماده کردن این آموزش کمی طول کشید و یک عذر خواهی به شما عزیزان که این آموزش ها رو دنبال می کنید بدهکارم. اما به امید خدا دیگه همچین اتفاقی نمیوفته و مطالب سر وقت خودشون یعنی هر سه روز آماده می شن . خب حالا بریم سر اصل مطلب :

در این قسمت از آموزش css با پنج ویژگی دیگر آشنا خواهیم شد، که به فضا های اطراف عنصر مربوط می شوند و عبارتنداز : margin و padding و border و height و width .

برای آشنایی بهتر با این خصوصیات و مکان قرارگیری هر کدام در اطراف یک عنصر به تصویر زیر دقت کنید:



خب اگر گیج شدید کمی درمورد تصویر توضیح میدهیم : در وسط قسمت سفید رنگ عنصر ما قرار دارد که هر عنصری نیز ارتفاع (height) و عرض (width) مشخصی دارد. برای تعیین فاصله متن درون عنصر از حاشیه (border)، از padding استفاده میشود. Border هم که حاشیه دور عنصر است، و margin نیز فاصله عنصر از عناصر اطراف آن را مشخص می کند.

حال به تعریف دقیق تر به همراه مثال برای هر خاصیت می پردازیم :

خصوصیتmargin

این خصوصیت فضای خالی بیرونی عناصر را مشخص می کند، یعنی فضای خالی بین عنصر و پنجره ی مرورگر یا فضای خالی بین دو عنصر را مشخص می کند. یکی از سه مقدار عددی، درصدی یا auto را می پذیرد:

- Margin-top : فاصله بالای عنصر را مشخص می کند.
- Margin-right : فاصله سمت راست عنصر را مشخص می کند.
- Margin-bottom : فاصله پایینی عنصر را مشخص می کند.
- Margin-left : فاصله سمت چپ عنصر را مشخص می کند.

مقدار پیشفرض auto می باشد در ضمن استفاده از مقادیر منفی نیز مجاز است.

مثال :

```
1 Body { margin-top: 0; }
```

دستور فوق حاشیه بالای سند را حذف می کند.

درضمن می توان بجای نوشتن هر چهار خصوصیت از حالت کوتاه نویسی نیز استفاده کرد. در حالت کوتاه نویسی می توان از یک تا چهار به مقدار به خصوصیت margin داد.

کوتاه نویسی خصوصیتMargin

در صورت استفاده از یک مقدار در کوتاه نویسی این مقدار به هر چهار فاصله بیرونی تخصیص خواهد یافت.

مثال :

```
1 Body { margin : 5px; }
```

دستور فوق تمامی فاصله های بیرونی عنصر body را (top , right , bottom , left) را برابر px۵ قرار می دهد.

هنگام استفاده از دو مقدار در کوتاه نویسی مقدار اول bottom , top و مقدار دوم right , left را مشخص می کند.

مثال :

```
1 P { margin : 2px 5px; }
```

دستور فوق فاصله بالا و پایین (top , bottom) را برابر px۲ و حاشیه چپ و راست را (right , left) را برابر px۵ تعریف می کند.

در صورت استفاده از سه مقدار در کوتاه نویسی مقدار اول top، مقدار دوم left، مقدار سوم right و مقدار چهارم bottom را مشخص می کند.

مثال :

```
1 Body {margin: 20px 30px 40px; }
```

دستور فوق top را برابر px۲۰، خصوصیت left و right را برابر px۳۰ و مقدار سوم bottom را برابر px٤۰ قرار می دهد.

در صورت استفاده از چهار مقدار در کوتاه نویسی مقدار اول top مقدار دوم right مقدار سوم bottom و مقدار چهارم left را مشخص می کند.

مثال :

```
1 Body { margin : 2px 3px 5px 4px; }
```

دستور فوق top را برابر px۲، مقدار right را برابر px۳، مقدار bottom را برابر px۵ و مقدار left را برابر px٤ قرار می دهد.

حال برای آشنایی با کاربرد این خصوصیت به مثال زیر دقت کنید :
کد html زیر را در نظر بگیرید :

```
<div>
<a href="#">Link 1 </a>
<a href="#">Link 2 </a>
<a href="#">Link 3 </a>
<a href="#">Link 4 </a>
<a href="#">Link 5 </a>
</div>

<a href="#">Link 1 </a>
<a href="#">Link 2 </a>
<a href="#">Link 3 </a>
<a href="#">Link 4 </a>
<a href="#">Link 5 </a>
```

حال میخواهیم بین لینک هایی که در داخل div قرار دارند، و بین لینک های داخل div لینک هایی در زیر آن ها قرار دارند فاصله ایجاد کنیم یعنی باید از دو خصوصیت margin-bottom و margin-right استفاده کنیم

```
div a {
Margin-right: 10px;
}
div {
margin-bottom: 20px;
}
```

نتیجه در مرورگر :



در margin عمودی یعنی marign-top و margin-bottom امکان فروپاشی یا تداخل حاشیه ها وجود دارد؛ این تداخل زمانی اتفاق می افتد که margin-top یا margin-bottom از یک عنصر با margin-bottom یا margin-top از عنصر دیگر برخورد داشته باشد. اگر تداخل حاشیه ها اتفاق بیافتد فاصله بین دو عنصر بسیار بیشتر از چیزی می شود که انتظار داشتیم، هنگامی که چنین اتفاقی بیافتد فقط حاشیه های عنصر با مقدار margin بزرگتر اعمال خواهند شد و حاشیه های عنصر با margin کوچکتر در نظر گرفته نشده و حذف می شود.

اگر عنصری فرزند عنصر دیگری باشد و به هر دو عنصر margin-top یا margin-bottom اختصاص داده باشیم در این صورت نیز تداخل فاصله های بیرونی پیش خواهد آمد و margin عنصر پدر صرف نظر از کوچکتر بودن یا بزرگتر بودن اعمال خواهد شد و margin عنصر فرزند در نظر گرفته نخواهد شد و حذف می شود.

می توان با تعریف padding یا border متناسب با محل تداخل برای عنصر پدر از برخورد دو margin در عناصر پدر و فرزند جلوگیری کرد.

نکته:

۱. این قاعده های کوتاه نویسی در خصوصیت های padding و border نیز به همین شکل به کار می روند.
۲. معمولا در حالت کوتاه نویسی از دو یا چهار مقدار استفاده می شود و استفاده از سه مقدار چون گیج کننده است توصیه نمی شود.
۳. در حالت کوتاه نویسی چهار مقدار در جهت عقربه های ساعت تخصیص می یابند. یعنی همان ۱۲، ۳ ، ٦، ۹ به این صورت که اول top سپس right سپس bottom و در آخر.left

استفاده از مقدار auto در margin نیز کاربرد زیادی دارد و برای چینش عناصر در وسط، چپ، یا راست استفاده می شود به مثال زیر دقت کنید :

```
1 Margin : 0 auto 0 0;      /* Left-Align */
2
3 Margin : 0 auto;          /* Center Align */
4
5 Margin : 0 0 0 auto;      /* Right Align */
```

در مثال فوق خط اول باعث چپ چین شدن عنصر، خط دوم باعث وسط چین شدن عنصر و خط سوم باعث راست چین شدن عنصر می شود.

خصوصیتBorder

Border یا حاشیه، در واقع بین margin و padding قرار گرفته و یک خط به دور عنصر مورد نظر می کشد و خود به چهار قسمت تقسیم می شود.

- Border-width : پهنای خط دور عنصر را مشخص می کند
- Border-color : رنگ خط دور عنصر را مشخص می کند.
- Border-style : طرح خط دور عنصر را مشخص می کند.

حال به تعریف دقیق این سه قسمت وآشنا شدن با همه ی اجزای آن ها می پردازیم :

Border Width

از خاصیت border-width برای مشخص کردن پهنا یا عرض کادر یک عنصر استفاده می شود که یکی از مقادیر زیر را میگیرد :

- Thin : باریک، برابر با ۱px
- Medium : متوسط برابر با ۳px
- Thick : پهن، برابر با ۵px
- مقدار عددی دلخواه برحسب یکی از واحدهای محاسباتی که معمولا px است.

و مقدار پیشفرض این خصوصیت medium می باشد.

نکته:

۱. مقادیر منفی برای این خصوصیت مجاز نیست
۲. مقادیر درصدی نیز برای این خصوصیت قابل استفاده نیست، کلمات کلیدی نیز به ندرت استفاده میشود. معمولا از واحد pixel برای مقدار دهی به این خاصیت استفاده می شود.
۳. همانند margin در کوتاه نویسی این خصوصیت می توان از یک تا چهار مقدار استفاده کرد و قواعد گفته شده در کوتاه نویسی margin در اینجا نیز صدق می کنند.

مثال :

```
1 <h1>7learn : Javascript and Web Design Tutorials </h1>
```

```
1                    H1 {
2 border-width: 2px 5px 7px 9px;
3 }
```



خاصیت border-color

این خاصیت برای مشخص کردن رنگ border استفاده می شود. و یکی از مقادیر هگزادسیمال rgba , rbg ، کلمات کلیدی رنگ ها یا transparent(شفاف - بدون رنگ) را به عنوان مقدار می پذیرد

مقدار پیشفرض برای این خصوصیت برابر با مقدار تعیین شده در خصوصیت color از عنصر مورد نظر می باشد.

نکته:

۱. در حالت کوتاه نویسی می توان چهار طرف border را در یک خط و به صورت جداگانه مقدار دهی کرد) .همانند کوتاه نویسی خصوصیت(margin
۲. در ie6 و ie7 کلمه کلیدی transparent برای border-color پذیرفته نمی شود.
۳. در ie 7 کلمه کلیدی transparent به صورت black (سیاه) تفسیر می شود.

مثال :

```
1 <h1>7learn : Javascript and Web Design Tutorials </h1>
```

```
1                                H1 {
2 border-width: 2px 5px 7px 9px;
3 border-color: black green red yellow;
4                                    }
```



خصوصیتborder-style

این خاصیت برای یک border باید تخصیص یابد تا border به نمایش در آید، و طرح خط دور عنصر را مشخص می کند که یکی از مقادیر زیر را می گیرد :

```
1 None , dotted , dashed , solid , double , hidden , groove , ridge , inset , outset
```

مقدار پیشفرض : none (بدون خط حاشیه)

مثال : کد html

```
1 <h1> 7Learn: Javascript and Web Design Tutorials </h1>
```

و کد css زیر :

```
1                                   h1 {
2 border-width: 2px 5px 7px 9px;
3         border-color: black red green pink;
4 border-style: solid dashed dotted double;
5 }
```

نتیجه در مرورگر :



کوتاه نویسی خاصیتborder

قاعده کلی

```
1 Border: border-width border-style border-color;
```

ترتیب مقدار دهی به این خاصیت مهم بوده و جا به جایی در آن باعث از کار افتادن کد می شود.

مثال :

```
1                    P {
2 Border: 3px dashed red;
3 }
```

این دستور border هر چهار طرف عنصر را مشخص می کند اما شما می توانید از این از این کوتاه نویسی برای هر از چهار طرف استفاده کنید :

مثال :

```
1                        P {
2 Border-top: 3px dashed red;
3     Border-right: 2px solid green;
4 Border-bottom: 5px dotted black;
5 Border-left 1px groove yellow;
6                        }
```

بسیار خب تا به اینجا شما با دو فضا از فضای اطراف هر عنصر آشنا شدید در جلسه بعدی شما را با سه خصوصیت دیگر یعنی padding و height و width آشنا خواهیم کرد.

در جلسه قبل شما را با دو خصوصیت margin و border آشنا کردیم، گفتیم که برای ایجاد فاصله بین یک عنصر و عناصر اطراف آن باید از خصوصیت margin استفاده کرد و برای ایجاد خط حاشیه دور عنصر نیز از خاصیت border استفاده می کنیم، گفتیم که با استفاده از border-width عرض خط حاشیه، border-color رنگ خط حاشیه و با استفاده از border-style طرح خط حاشیه را می توانیم تعیین کنیم.

در این جلسه با یکی از دیگر خصوصیات border که در css3 معرفی شده است آشنا می شویم و آن هم border-radius است و برای خمیده کردن حاشیه ها استفاده می شود. و نیز با خصوصیات padding و height و width شما را آشنا خواهیم کرد.

خصوصیتborder-radius

قبل از آمدن css3 برای گرد کردن حاشیه های یک عنصر مجبور بودیم که خودمان عناصری مانند تصاویر را گوشه گرد کنیم یا که برای باقی عناصر از تصاویر اضافی برای انجام اینکار استفاده کنیم، اما با آمدن css3 و معرفی خصوصیت جدید border-radius این کار بسیار ساده شده می توان با یک خط کد گوشه های عناصر را به شکلی زیبا گرد کرد. این خصوصیت را میتوان برای هر چهار گوشه با یک مقدار متفاوت تعریف کرد، و برای مقدار دهی از مقدار عددی که معمولا بر حسب pxاست استفاده می کنیم، و بیشترین مقداری که میگیرد ۵۰px است.

مثال:

```
1 <img src="images/img1.png" />
```

```
1                          Img {
2 Border-top-left-radius: 50px;
3 Border-bottom-left-radius: 40px;
4   Border-top-right-radius: 30px;
5 Border-bottom-right-radius: 0px;
6                                  }
```

نتیجه در مرورگر:



کوتاه نویسی خاصیت border-radius

می توان این خاصیت را در یک خط نیز کوتاه نویسی کرد و قاعد کلی آن به شکل زیر است.

```
1 border-radius: top-left top-right bottom-right bottom-left;
```

مثال :

```
1 Img {Border-radius: 50px 5px 50px 5px; }
```

نتیجه در مرورگر



خصوصیتpadding

خصوصیت padding مقدار فضای خالی بین محتوای یک عنصر و border آن عنصر را تعیین می کند. یکی از دو مقدار عددی یا درصدی را می پذیرد. (قواعد مقدار دهی آن همانند قواعد خصوصیت margin است).

تفاوت های بین margin وpadding

۱. Paddingفضای خالی بین محتوا و border یک عنصر می باشد اما margin فضای خالی بین borderو عناصر دیگر می باشد.
۲. کلمه کلیدی auto هیچ تاثیری روی خاصیت padding ندارد.
۳. خاصیت margin می تواند مقادیر منفی بگیرد اما خاصیت padding نمی تواند.
٤. در خاصیت padding تداخل و فروپاشی اتفاق نمی افتد، فقط در margin این اتفاق می افتد.

کوتاه نویسی خاصیتpadding

همانطور که گفتیم قوعد مقداردهی به خصوصیت padding همانند خصوصیت marginاست است: و قاعده کلی کوتاه نویسی آن ها نیز شبیه به هم است:

```
Padding: top right bottom left;
```

مثال :

```
P {
Padding: 3px 5px 2px 7px;
}
```

دستور بالا padding-top را برابر ۳پیکسل ، padding-right را برابر ۵ پیکسل، padding-bottom را برابر ۲ پیکسل و padding-left را برابر ۷ پیکسل تعریف می کند.

خصوصیتwidth

خاصیت width عرض محتوای یک عنصر را تعیین می کند. در نهایت عرض کلی یک عنصر برابر با عرض متحتوا (تعیین شده توسط width) بعلاوه ی عرض padding و عرض border آن عنصر می باشد. این خصوصیت را می توان با یکی از سه مقدار عددی، درصدی یا auto مقداردهی کرد. و مقدار پیشفرض آن auto می باشد. مقادیر منفی نیز برای این خصوصیت مجاز نمی باشد.

از خاصیت width می توان برای تخصیص یک مقدار کلی برای برخی عناصر input مثل دکمه های reset و submit استفاده کرد.

مثال :

```
1 Input.button {
2 Width: 10em;
3 }
```

دستور فوق عرض تمامی عناصر input با کلاس button را em۱۰ در نظر می گیرد.

خصوصیتheight

در کنار خصوصیت width خاصیت height وجود دارد که ارتفاع محتوای یک عنصر را مشخص می کند. البته استفاده از خاصیت height برای عناصر چندان متداول نیست و بیشتر همان مقدار پیشفرض یعنی auto را برای آن در نظر گرفته می شود و صفحات وب معمولا با scrool عمودی طراحی می شوند. این خصیوصیت نیز مانند width یکی از سه مقدار عددی، درصدی و یا auto را به عنوان مقدار می پذیرد.

همانند خاصیت width از خاصیت height می توان برای مشخص کردن ابعاد یک تصویر استفاده کرد.

مثال :

```
1   Img.footer {
2 Width: 440px;
3 Height: 130px;
4                 }
```

ابعاد min و max برای width وheight

طراحی با عرض متغییر هنگامی استفاده می شود که میخواهید محتویات را برای مطابقت با resolution های مختلف صفحه نمایش طراحی کنید. همچنین هنگامی مفید می باشد که میخواهید عرض عنصر منبسط یا منقبض نشود.

هنگامی که شما صفحه ای را اصولا برای نمایش در رزولوشن ۸۰۰*۶۰۰ یا ۱۰۲٤*۷٦۸ طراحی می کنید، اگر کاربری سایت شما را با رزولوشن ۱٦۰۰*۱۲۰۰ مشاهده کند، صفحه را به شکلی نه چندان زیبا خواهد دید، یا هنگام استفاده از کلمه کلیدی auto یا مقادیر درصدی ممکن است مشاهده با رزولوشن های مختلف نتایج متفاوتی داشته باشد. در این مواقع استفاده از مقادیر min و max برای width و height می تواند مانع از تغییر ظاهر صفحه در رزولوشن های مختلف شود.

خصوصیتmin

خاصیت min یکی از دو مقدار عددی یا درصدی را می پذیرد. و مقدار پیش فرض آن ۰ است، این خصوصیت در ie6 و قبل از آن پشتیبانی نمی شود.

خصیوصیت min-width و mn-height محدودیت کوچکترین اندازه را برای یک عنصر مشخص می کنند، و از منقبض شدن عرض یا ارتفاع یک عنصر هنگام تغییر سایز پنجره مرورگر یا رزولوشن جلوگیری می کند.

مثال :

```
1 Body {
2 Min-width: 1024px;
3 }
```



در مثال فوق تعریف min-wdith:1024px; برای عنصر body مانع از منقبض شدن محتوای صفحه شده و صفحه scroll افقی پیدا می کند.

خصوصیتmax

خصوصیت max محدودیت بالاتر و حداکثر اندازه را برای یک عنصر مشخص می کند و هنگامی تعریف می شوند که ابعاد یک عنصر ممکن است افزایش یابند که با استفاده از این خصوصیات می توان مانع از منبسط شد و افزایش ابعاد عناصر شد. این خصوصیت یکی از سه مقدار عددی، درصدی یا auto را می پذیرد.

مثال :

```
1 Body {
2 Max-width: 1440;
3 Max-height: auto;
4 }
```

خصوصیتoverflow

این خصوصیت هنگامی استفاده می شود که محتویات یک عنصر به دلیل محدودیت در ابعاد تعریف شده برای آن، امکان سرریز دارند، خاصیت overflow نحوه ی برخورد با این سرریز را مشخص می کند.

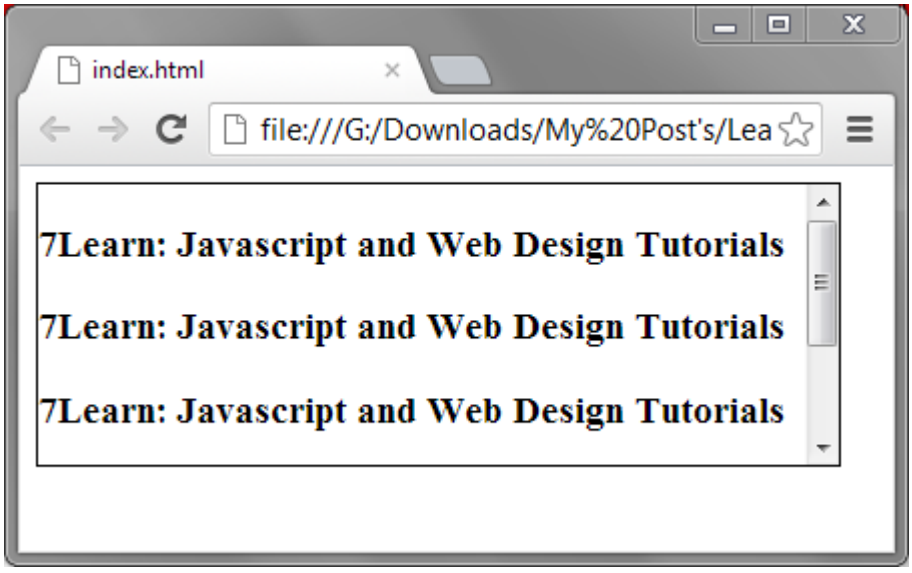
و یکی از ٤ مقدار زیر را می پذیرد:

- Visible : مقدار پیشفرض و باعث نمایش محتوای سرریز شده می شود.
- Hidden : محتوای سرریز شده پنهان می شود.
- Scroll : محتوای سرریز شده را پنهان می کند و قابلیت اسکرول شدن برای دیدن محتوای سرریز شده را فعال می کند
- Auto : اگر کم بودن عرض عنصر باعث سر ریز شدن باشد، اسکرول افقی و اگر کم بودن ارتفاع عنصر باعث سرریز شدن باشد اسکرول عمودی را برای عنصر فعال می کند.

مثال :

```
1                                                    <div>
2 <h3> 7Learn: Javascript and Web Design Tutorials </h3>
3 <h3> 7Learn: Javascript and Web Design Tutorials </h3>
4 <h3> 7Learn: Javascript and Web Design Tutorials </h3>
5 <h3> 7Learn: Javascript and Web Design Tutorials </h3>
6 <h3> 7Learn: Javascript and Web Design Tutorials </h3>
7 </div>
```

```
1   div {
2 border: 1px solid #000;
3                 width: 400px;
4  height: 140px;
5 overflow: auto;
6 }
```



overflow-yوoverflow-x

این خصوصیات مشابه overflow می باشند و اجازه ی کنترل overflow را با scrollbar افقی یا عمودی به طور جداگانه در اختیار می گذارند.

بسیار خب تا به اینجا با همه ی فضاهای اطراف یک عنصر و خاصیت overflow نیز آشنا شدیم در فصل بعدی با خاصیت float یا شناور ساختن عناصر و خاصیت clear، و تراز افقی شما را آشنا خواهیم کرد.

خصوصیتfont-family

می توان نوع فونت مورد استفاده برای متن را با استفاده از خاصیت font family تعیین کرد. به طور مثال فرض کنید که ما کد html زیر را داریم :

```
1                    <h2> 7Learn.com </h2>
2 <h4> Javascript and Web Design Tutorials </h4>
```

حال می خواهیم که متن تگ h2 را با فونت Courier New و متن تگ p را با فونت Tahoma نمایش دهیم، کد css ما چنین خواهد شد :

```
1 h2 { font-family: "courier new"; }
2       h4 {font-family: tahoma; }
```

نتیجه در مرورگر :



حال حتما می پرسید که چرا نام فونت courier new را داخل کوتیشن (" ") گذاشته ام اما نام فونت Tahoma را نه. این کار، به این دلیل است که نام فونت courier new دارای فضای خالی (space) نیز هست، هرگاه نام فونتی دارای فضای خالی بود باید نام آن را داخل کوتیشن قرار داد. درضمن می توان از چندین فونت برای یک تگ استفاده کرد به طور مثال :

```
1 h2 { font-family: times , "times new roman" , serif; }
```

هنگامی که از چندین فونت استفاده می شود، این فونت ها باید با علامت (,) از یکدیگر جدا شوند. در این حالت اگر فونت اول بر روی سیستم کاربر موجود نباشد دومین فونت اعمال خواهد شد و اگر فونت دوم نیز بر روی سیستم کاربر موجود نباشد سومین فونت اعمال خواهد شد و الی آخر.

استایل تعریف شده در مثال فوق فونت تمامی تگ های h2 را به فونت times تغییر خواهد داد و اگر فونت times بر روی سیستم کاربر موجود نباشد از فونت times new roman استفاده خواهد شد و اگر .....

خصوصیتfont-style

با استفاده از این خاصیت می توان متن را به صورت italic یا مورب نمایش داد. و سه مقدار را می پذیرد:

- Normal : مقدار پیشفرض و حالت عادی فونت می باشد.
- Italic : متن را به صورت مورب نمایش می دهد.
- Oblique : متن را به صورت مورب نمایش می دهد و در اکثر فونت ها تفاوت ظاهری چندانی با italic ندارد.

به یاد داشته باشید که همه ی فونت ها شکل italic ندارند.

مثال :

```
1  h2 { font-style: italic; }
2 h4 { font-style: oblique; }
```

خصوصیتfont-variant

این خصوصیت حروف را به صورت uppercase (حروف بزرگ) نمایش می دهد. اما در این حالت حروف کمی از حالت capitalized کوچکترند. این خصوصیت دو مقدار زیر را می گیرد:

- Normal : مقدار پیشفرض و حالت عادی حروف
- Small-caps : نمایش حروف به صورت uppercase

مثال :

```
1 h2 , h4 { font-variant: small-caps; }
```

نتیجه در مرورگر :



خصوصیتfont-weight

با استفاده از این خاصیت می توان مقدار ضخامت متن را مشخص کرد.

و یکی از مقادیر زیر را می گیرد.

```
1 normal , bold , bolder , lighter , 100 , 200 , 300 , 400 , 500 , 600 , 700 , 800 , 900
```

مقدار پیشفرض این خصوصیت normal است.

در عمل فقط سه مقدار normal و bold و lighter در اکثر مرورگرها به درستی کار می کنند.

به طور معمول مقدار normal معادل مقدار ٤٠٠ و مقدار bold معادل مقدار ٧٠٠ می باشد.

خصوصیتtext-indent

این خاصیت مقدار تو رفتگی اولین خط از عنصر را تعریف می کند و به طور معمول برای ایجاد تو رفتگی در اولین خط یک پاراگراف استفاده می شود

خصوصیتtext-align

این خاصیت مشابه صفت align در تگ های HTML می باشد و برای تراز بندی افقی متن استفاده می شود، و یکی از چهار زیر را می پذیرد:

- Left : مقدار پیشفرض و متن را از چپ تراز می کند.
- Right : متن را از سمت راست تراز می کند.
- Center : متن را از وسط تراز می کند.
- Justify : متن را از دو طرف تراز می کند.

مثال: استفاده از خاصیت های text-indent و text-align

کد html :

```
1 <h2> 7Learn.com </h2>
2 <p> Javascript and Web Design Tutorials/ Javascript and Web Design Tutorials/ Javascript and Web Design Tutorials</p>
```

و کد css زیر :

```
1 h2 { text-align: center; }
2 p {
3 text-indent: 20px;
4 text-align: justify;
5 }
```



خصوصیتtext-decoration

از این خاصیت برای underline (زیرخط دار) کردن متون و یا برداشتن underline از لینک ها، استفاده می شود و از ۵ مقدار زیر را می پذیرد:

- Underline : یک خط زیر متن می کشد.
- Overline : یک خط بالای متن می کشد.
- Line-through : یک خط روی متن می کشد.
- Blink : متن را به صورت چشمک زن نمایش می دهد و معادل تگ <blink> در HTML است.
- None : مقدار پیشفرض و برای از بین بردن خط ها.

درضمن می توان به طور همزمان از ٤ مقدار اول، استفاده کرد.

مثال :

```
1                    <h2> 7Learn.com </h2>
2 <p> Javascript and Web Design Tutorials </p>
```

و کد css :

```
1 h2 { text-decoration: underline overline; }
2          p { text-decoration: line-through; }
```



بسیار خب تا به اینجا با ٦ خصوصیت قالب بندی متن آشنا شدید، در جلسه دوم شما را با بقیه خصوصیات قالب بندی متن آشنا خواهیم کرد.

خصوصیتdirection

این خاصیت مشابه صفت dir در تگ های HTML می باشد و برای تراز متن استفاده می شود، و یکی از ۲ مقدار زیر را می پذیرد:

- ltr : مقدار پیشفرض و متن را چپ چین می کند و معمولا برای متون انگلیسی است
- rtl : متن را راست چین می کند. و معمولا برای متون فارسی است

خاصیتline-height

با استفاده از این خاصیت می توان فاصله عمودی بین خطوط یک متن را کنترل کرد. و مقداری که می پذیرد، عدد اس و واحد آن em می باشد. در ضمن اندازه ای آن به اندازه فونت بستگی دارد.

مثال:

```
1 p { line-height: 2; }
```

همانطور که گفتیم اندازه ی line-height به اندازه فونت بستگی دارد، به طور مثال اگر اندازه فونت تگ p در مثال بالا px۱۰ باشد مقدار ۲ برای خصوصیت line-height برابر با px۲۰ خواهد شد..

نکته : توصیه می شود که از خصوصیت line-height فقط برای تگ body استفاده شود، زیرا در صورت استفاده آن برای دیگر تگ ها، در صورت تغییر اندازه فونت، اندازه line- height نیز تغییر خواهد کرد، اما در تگ body این صور نیست و با تغییر اندازه ی فونت، line-heightهمه ی عناصر یکسان خواهد بود.

با تخصیص مقدار به line-height مقدار به دو قسمت مساوی تقسیم شده و نصف آن به فضای خالی بالای خط و نصف دیگر به فضای خالی پایین خط اختصاص داده می شود.

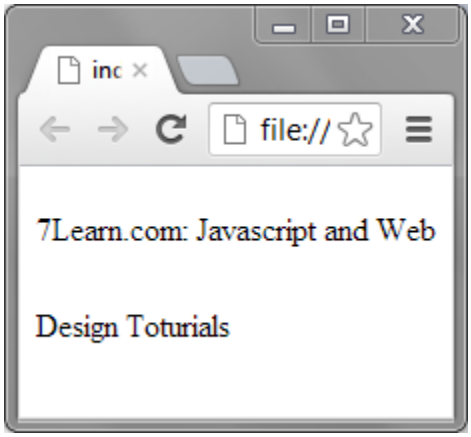
مثال :

```
1 <p> 7Learn.com: Javascript and Web Design Tutorials </p>
```

و کد css :

```
1 p { line-height: 3; }
```

نتیجه در مرورگر:



خصوصیتletter-spacing

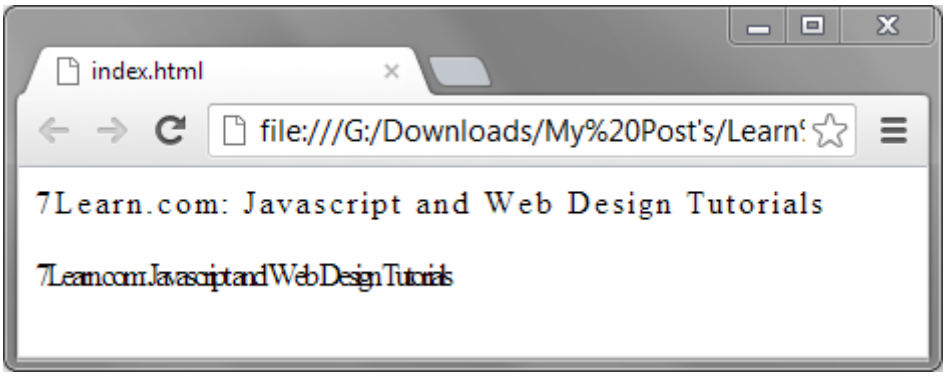
خاصیت letter spacing یک فضای خالی اضافی بین کاراکتر ها ایجاد می کند. برای این خاصیت دو مقدار عددی مثبت و منفی مجاز می باشد که استفاده از مقادیر منفی باعث تو رفتگی و به هم ریختگی حروف می شود.

مثال:

```
1 <p class="p1"> 7Learn.com: Javascript and Web Design Tutorials </p>
2 <p class="p2"> 7Learn.com: Javascript and Web Design Tutorials </p>
```

و کد css زیر:

```
1  .p1 { letter-spacing: 2px; }
2 .p2 { letter-spacing: -2px; }
```



خصوصیتword-spacing

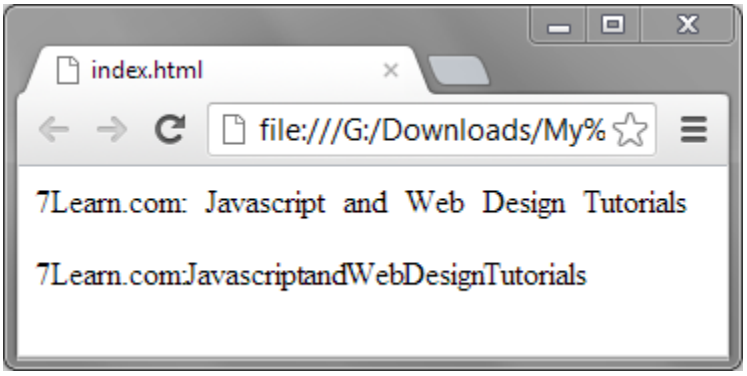
خاصیت word-spacing یک مقدار فضای خالی اضافی برای فاصله بین لغات تعیین می کن. این مقدار نیز می تواند هم منفی و هم مثبت باشد.

مثال:

```
1 <p class="p1"> 7Learn.com: Javascript and Web Design Tutorials </p>
2 <p class="p2"> 7Learn.com: Javascript and Web Design Tutorials </p>
```

و کد css زیر:

```
1 .p1 { word-spacing: 5px; }
2 .p2 { word-spacing: -5px; }
```



خصوصیتtext transform

با استفاده از این خاصیت می توان کلمات (انگلیسی) را به حروف بزرگ یا حروف کوچک تبدیل کرد، و یکی از چهار مقدار زیر را می گیرد:

- Capitalize : اولین کاراکتر از هر کلمه را با حروف بزرگ نمایش می دهد.
- Uppercase: تمامی کاراکترهای یک کلمه را با حروف بزرگ نمایش می دهد.
- Lowercase : تمامی کاراکترهای یک کلمه را با حروف کوچک نمایش می دهد.
- None : مقدار پیشفرض را برای کاراکتر ها در نظر می گیرد.

مثال :

```
<h2 class="capitalize"> 7Learn.com </h2>
<h2 class="uppercase"> 7Learn.com </h2>
<h2 class="lowercase"> 7Learn.com </h2>
```

و کد css زیر :

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



خصوصیت white space

مقدار پیشفرض: normal

- Pre : از این مقدار برای نمایش متن به همان شکلی که در عنصر مورد نظر نوشته شده با در نظر گرفتن تمامی فضاهای خالی و شکست های خط استفاده می شود. از مقدار pre معمولا برای عنصر <code>استفاده می شود. مقدار pre معادل تگ pre در HTML می باشد.
- Nowrap : استفاده از مقدار nowrap از شکست خط در متن جلوگیری می کند می کند، مگر اینکه از تگ
استفاده شده باشد. این مقدار تقریبا هیچ وقت استفاده نمی شود زیرا استفاده از این مقدار آرایش صفحه را به هم می زند.

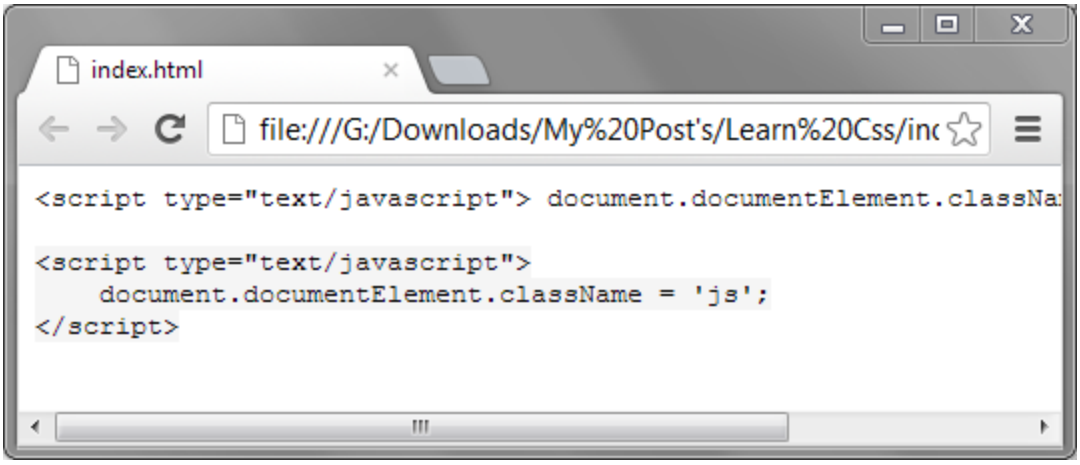
مثال:

```
<code class="code1">
<script type="text/javascript">
document.documentElement.className = 'js';
</script>
</code>
```

```
6 <code class="code2">
7                         <script type="text/javascript">
8       document.documentElement.className = 'js';
9    </script>
10    </code>
```

و کد CSS :

```
1 .code1 { white-space: nowrap; }
2       .code2 { white-space: pre; }
```



پیشوند مرورگرها

پیشوند مرورگرها را می توانید در تصویر زیر مشاهده کنید. :

WebKit	-webkit-
Mozilla	-moz-
Opera	-o-
Konqueror	-khtml-
Microsoft	-ms-
Chrome	-chrome-

::. طبق تجربیات، مرورگر گوگل کروم نیز با افزودن پیشوند -webkit- خصوصیت مورد نظر را تاثیر می دهد.

::. استفاده از هر پیشوند کد را فقط در آن مرورگر نمایش خواهد داد و در دیگر مرورگرها کد از کار خواهد افتاد. کار جالبی که می توان با استفاده از این پیشوند ها انجام داد، تاثیر تنضیمات خاصی هنگام استفاده کاربر از هر مرورگر است.

مثال:

```
1                    P {
2 text-shadow: 2px 2px 2px blue;
3 -webkit-text-shadow: 2px 2px 2px green;
4    -moz-text-shadow: 2px 2px 2px black;
5 -o-text-shadow: 2px 2px 2px yellow;
6  -ms-text-shadow: 2px 2px 2px red;
7 }
```

کد بالا به این معناس که سایه متن تگ p را در همه ی مرورگرها به رنگ آبی (خط ۲) اما در مرورگر گوگل کروم و سافاری، به رنگ سبز (خط ۳) ، در مرورگر فایرفاکس به رنگ سیاه (خط ٤)، در مرورگر اوپرا به رنگ زرد (خط ٥) و در مرورگر اینترنت اکسپلورر به رنگ قرمز (خط ٦) نشان دهد.

[note/]

خصوصیت@font-face

در بعضی از مواقع دلمان می خواهد از فونت های مورد نظرمان استفاده کنیم، اما به دلیل اینکه شاید این فونت بر روی سیستم کاربر نصب نشده باشد، از آن فونت صرف نظر می کنیم و باز به دنبال انتخاب فونتی از میان فونت های پیشفرض می رویم. اما با آمدن css3 و معرفی خصوصیت font-face شما می توانید از هر فونتی حتی اگر بر روی سیستم کاربر نصب نشده باشد نیز استفاده کنید.

استفاده از این خاصیت به صورت زیر است :

```
1 @font-face {
2 font-family: FONT NAME;
3 src: url("FONT ADDRESS") format("FONT TYPE");
```

به طور مثال

```
1 <h2> 7Learn.com </h2>
2 <p> Javascript and Web Design Tutorials </p>
```

و کد css :

```
1 @font-face {
2 font-family: "douglas adams hand";
3 src: url(fonts/dnahand.ttf) format("truetype");
4 }
5 @font-face {
6 font-family: "typographer rotunda";
7 src: url(fonts/typographerrotunda.otf) format(opentype");
8 }
9 h2 { font: 40px "typographer rotunda"; }
10 p { font: 20px "douglas adams hand"; }
```

نتیجه در مرورگر :



نکته : همه ی مرورگرها از همه ی فرمت های فونت پشتیبانی نمی کنند. به تصویر زیر دقت کنید تا با پشتیبانی مرورگرها از فرمت های فونت ها آشنا شوید:

فرمت های TTF , OTF و SVG	
فرمت های TTF , OTF و WOFF	
فرمت های EOT و WOFF	
فرمت های WOFF TTF , OTF , SVG	
فرمت های WOFF TTF , OTF , SVG	

نکته : این خصوصیت را باید در فصل دوم که مربوط به فونت ها بود معرفی می کردم، اما چونکه یادم رفته بود، اینجا براتون توضیحش دادم.

بسیار خب تا به اینجا شما با همه ی خصوصیات قالب بندی متون آشنا شدید، در فصل چهارم شما را با خصوصیات پس زمینه آشنا خواهیم کرد.

خاصیتFloat

از این خاصیت برای شناور کردن عناصر و قرار دادن آن ها در کنار یکدیگر استفاده می شود و یکی از سه مقدار را می پذیرد :

- None : مقدار پیشفرض (هیچ)

- Left : عنصری که این خاصیت برای آن تخصیص یابد در سمت چپ صفحه و یا در سمت چپ عنصر پدر خود قرار خواهد گرفت و عناصر پس از آن نیز در کنار این عنصر شناور خواهند شد.
- Right : عنصری که این خاصیت برای آن تخصیص یابد در سمت راست صفحه و یا در سمت راست عنصر پدر خود قرار خواهد گرفت و عناصر پس از آن نیز در کنار این عنصر شناور خواهند شد.

خاصیت float برای ایجاد و تنظیم بخش navigation (منوی سایت) مناسب است

مثال : (ایجاد بخش منوی بالای سایت با استفاده از float)

کد html :

```
1 <h1> 7Learn.com </h1>
2          <ul class="navi">
3 <li> <a href="#"> Home Page </a> </li>
4    <li> <a href="#"> Tutorials </a> </li>
5 <li> <a href="#"> About Us </a> </li>
6                                    </ul>
```

و کد Css زیر :

```
1       H1 {
2 Float: left;
3 Margin-top: 0;
4                  }
5
6 .navi {
7 Float: right;
8  Margin: 0;
 9 List-style: none;
10                  }
11
12 .navi li {
13 Float: left;
14             }
15
```

```
16 .navi a {
17     Display: block;
18 Margin-left: 15px;
19           Padding: 15px;
20 Border: 1px solid #ccc;
21               Color: #233;
22 Text-decoration: none;
23 }
24
25 .navi a:focus , .navi a:hover {
26               Background: #aaa;
27 Color: #fff;
28               }
```

نتیجه در مرورگر :



خاصیتClear

خاصیت clear وابسته به خاصیت float بوده و برای کنترل خاصیت float به کار می رود. و یکی از چهار مقدار زیر را می گیرد:

- مقدار پیشفرضnone :
- Left : تاثیر خاصیت float:left را از بین می برد؛ یعنی باعث می شود که تاثیر left floated element) عنصری که خاصیت float left به آن دادیم (بر روی عناصر دیگر از بین برود.
- Right : تاثیر خاصیت float: right را از بین می برد.

- Both : تاثیر هر دو خاصیت float:left و float:right را بین می برد.

یک توضیح ساده برای خاصیت clear این است که این خاصیت برای cancel کردن تاثیر یک یا تعداد بیشتری از عناصر floated بر روی دیگر عناصراستفاده می شود. تاثیر یک عنصر floated بر روی دیگر عناصر این است که عناصر همجوار این عنصر به صورت شناور در کنار این عنصر قرار خواهند گرفت. به منظور از بین بردن تاثیر عنصر floated بر روی عناصر همجواراز خاصیت clear استفاده می کنند.

اگر خاصیت clear برای عنصری تعریف شود این عنصر در کنار عنصر floated شناور نخواهد شد.

مثال :

```
<h1> 7Learn.com </h1>
        <ul class="navi">
<li> <a href="#"> Home Page </a> </li>
  <li> <a href="#"> Tutorials </a> </li>
<li> <a href="#"> About Us </a> </li>
                                  </ul>
<h2> Javascript and Web Design Tutorials </h2>
```

و کد Css زیر :

```
      H1 {
Float: left;
Margin-top: 0;
                }

.navi {
Float: right;
 Margin: 0;
List-style: none;
                }

.navi li {
Float: left;
            }

.navi a {
```

```
17     Display: block;
18 Margin-left: 15px;
19             Padding: 15px;
20 Border: 1px solid #ccc;
21                 Color: #233;
22 Text-decoration: none;
23 }
24
25 .navi a:focus , .navi a:hover {
26                 Background: #aaa;
27 Color: #fff;
28                 }
29
30 H2 {
31 Clear: both;
32                 }
```

نتیجه در مرورگر :



اگر در مثال فوق از خاصیت clear استفاده نکنیم، صفحه به هم ریخته خواهد شد. همانند شکل زیر:



خصوصیتVertical Align

این خاصیت به منظور تراز عمودی عناصر inline مثل <a> , <img> , و ... استفاده می شود...

خب اول باید فرق عناصر inline و عناصر block را بدانید تا تعریف این خاصیت قابل فهم شود.

- عناصر : block عناصری هستند که قبل و بعد از آنهایک خط خالی وجود دارد؛ مانند , <p> <hn> , و ... درHTML
- عناصر inline (خطی) : عناصر خطی عناصری هستند که قبل و بعد از انها خط خالی وجود ندارد؛ مانند <img>, <a>و... درHTML

و همان طور که گفتیم خاصیت vertical align برای تراز عناصر inline استفاده می شود؛ این خاصیت یکی از سه مقدار عددی، درصدی و کلمات کلیدی را به عنوان مقدار می پذیرد.

کلمات کلیدی:

- **Baseline :**این مقدار خطی است فرضی که هم تراز با قسمت پایینی حروف یک کلمه یا جمله می باشد. مثلا خط فرضی که کلمه Learn ۷رو آن می نشیند.
- **Middle :**تراز با میانه ی خط
- **Sub :**تراز بالای خط مثل اعداد توان در ریاضی؛ این مقدار معادل تگ <sub>در HTML است.
- **Super :**تراز با پایین خط و مثل اعداد اندیس در ریاضی ؛ این مقدار معادل تگ <sup>در HTMLاست.
- **Text-top** : متن را با بالای بلندترین حروف اطراف تراز می کند.
- **Text-bottom** : متن را با کوتاه ترین یا پایین ترین حروف تراز می کند، برای مثال g یا y زیر خط اصلی (baseline) قرار می گیرند و متن با پایین ترین قسمت این حروف تراز می شود.
- **Top :**قسمت بالای عنصر را بلند ترین عنصر موجود در خط تراز می کند.
- **Bottom :**قسمت پایین عنصر را با کوتاه ترین عنصر موجود در خط تراز می کند.

استفاده از مقادیر درصدی و عددی بستگی به خاصیت line-height تعریف شده برای عنصر مورد نظر دارد.

مثال : استفاده از vertical-align: 50% متن را در وسط خط تراز می کند.

مثال :

```
P {
Line-height: 10px;
Vertical-align: 5px;
}
```

در مثال فوق line-height برابر ۱۰ پیکسل و از آنجاییکه vertical-align برابر ۵ پیکسل و نصف مقدار line-height می باشد، متن در وسط خط تراز می شود.
خاصیت vertical-align به طور خاص برای تراز بندی تصاویر مفید است.

مثال :
کد html :

```
<h1>
<span class="top"> Top </span>
<span class="middle"> Middle </span> 7Learn
<span class="bottom"> bottom </span> 7Learn
</h1>
```

و کد css :

```
H1 span { font-size: 10px;}
.top {vertical-align: top;}
.bottom {vertical-align: bottom;}
.middle {vertical-align: middle;}
```

نتیجه در مرورگر :



بسیار خب تا اینجا با شناور ساختن عناصر و طریقه از بین بردن شناور بودن عناصر بعد از آن ها، و تراز عمودی آشنا شدیم، در فصل بعدی به کار با لیست های ul و ol و عناصر آن ها یعنی تگ li می پردازیم.

لیست ها درCSS

لیست ها عناصری با کاربرد های بسیار متنوع هستند. از جمله برای بخش navigation سایت و برای ایجاد کنترل های tab استفاده می شوند. Navigation معمولا شامل لیستی از لینک ها است، که دارای ویژگی های بصری می باشد و ثابت و خشک نیست، ما در این فصل به طریقه استایل دادن و معرفی خصوصیت هایی که برای کار با لیست ها در CSS استفاده میشود می پردازیم.

خاصیتlist-style-type

با استفاده از این خاصیت می توان برای لیست های مرتب و نامرتب bullet تعریف کرد. و یکی از ۱۴ مقدار زیر را می پذیرد.

```
1  disc , circle , square , decimal , decimal-leading-zero , lower-roman , upper-roman , lower-greek , lower-latin , upper-latin , armenian , georgian , none
```

مقدار پیشفرض برای لیست های نا مرتب (ul) : مقدار disc

مقدار پیشفرض برای لیست های مرتب (ol) : مقدار decimal

نکته : ممکن است در هنگام تعریف خصوصیات فوق، bulletلیست های مرتب را برای لیست های نامرتب در نظر بگیریم و بر عکس. شدیدا توصیه می شود که این کار انجام نشود و برای هر نوع از لیست ها bullet مخصوص به همان لیست را با توجه به معنای آن در نظر گرفت.

Bullet های مخصوص لیست های نامرتب (ul) عبارتنداز : square، circle و disc و none و بقیه bullet های مربوط به لیت های مرتب (ol) می باشد.

مثال:

کد HTML :

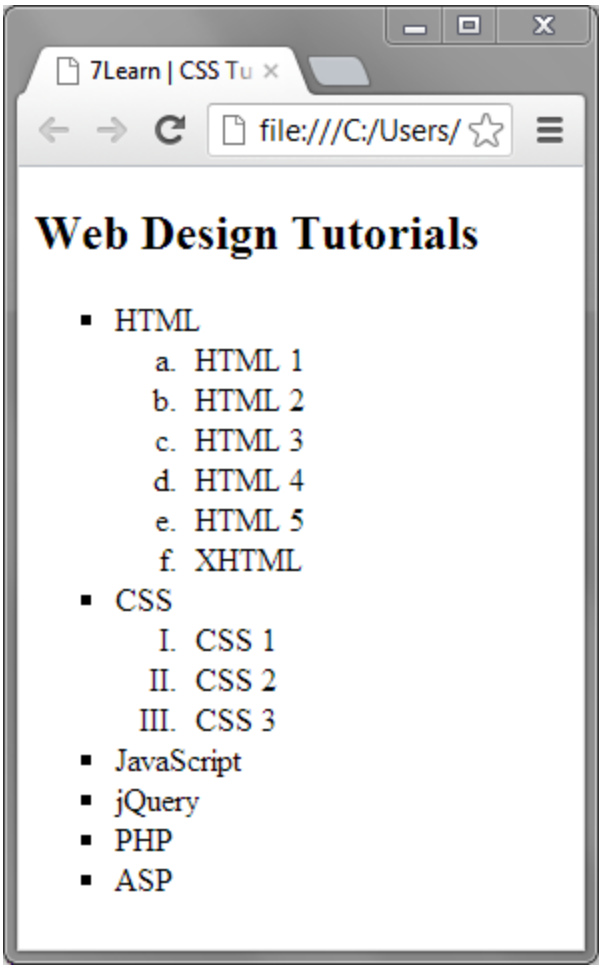
```
1 <h2> Web Design Tutorials </h2>
2                                   <ul>
3                 <li>HTML
4          <ol class="html">
5                   <li>HTML 1</li>
6                   <li>HTML 2</li>
7                   <li>HTML 3</li>
8                   <li>HTML 4</li>
9                                 <li>HTML 5</li>
```

```
10              <li> XHTML</li>
11      </ol>
12      </li>
13              <li>CSS
14      <ol class="css">
15          <li>CSS 1</li>
16          <li>CSS 2</li>
17          <li>CSS 3</li>
18                  </ol>
19                  </li>
20  <li>JavaScript</li>
21  <li>jQuery</li>
22    <li>PHP</li>
23  <li>ASP</li>
24          </ul>
```

و کد CSS :

```
1               ul li {list-style-type: square;}
2 ul ol.html li {list-style-type: lower-alpha;}
3 ul ol.css li {list-style-type: upper-roman;}
```

نتیجه در مرورگر:



خاصیت list style image

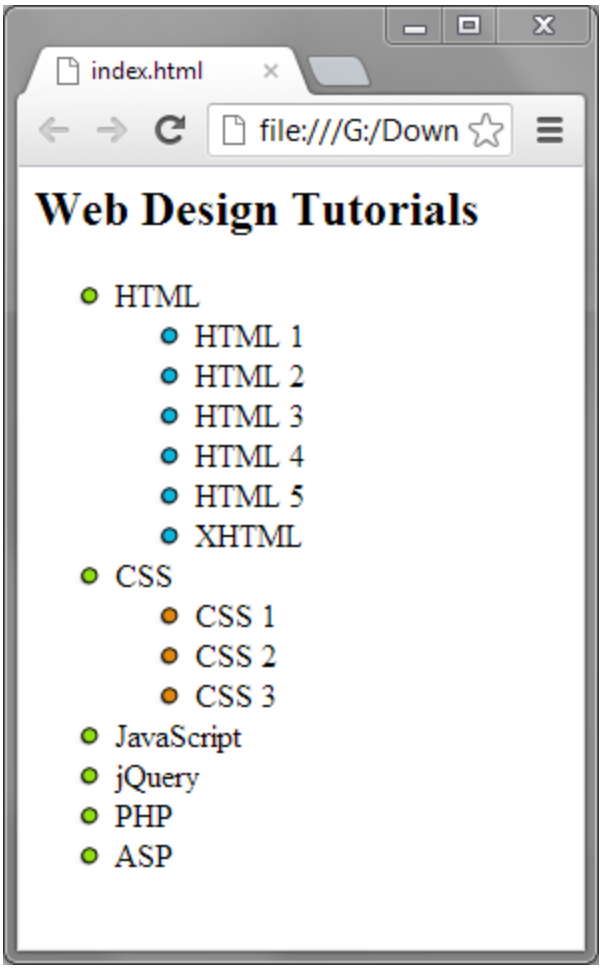
این خاصیت یک عکس را برای استفاده به عنوان نشانه bullet در کنار عنصر لیست تعریف می کند. این عکس با نشانه ی تعریف شده در خاصیت list-style-type جایگزین می شود.

مثال :

```
1 ul li {list-style-image: url('images/green-bullet.png');}

2 ul ol.html li {list-style-image: url('images/blue-bullet.png');}

3 ul ol.css li {list-style-image: url('images/brown-bullet.png');}
```



list-style-position

این خاصیت موقعیت و نحوه ی قرار گرفتن bullet در کنار آیتم های لیست را مشخص می کند. و یکی از دو مقدار زیر را می گیرد :

- Inside : مشخص می کند که bullet لیست باید در داخل آیتم های لیست قرار گیرد.
- Outside : مشخص که bullet لیست باید در خارج از آیتم های لیست قرار گیرد.

مثال :

```
1 ul li {list-style-image: url('images/arrow.png');}

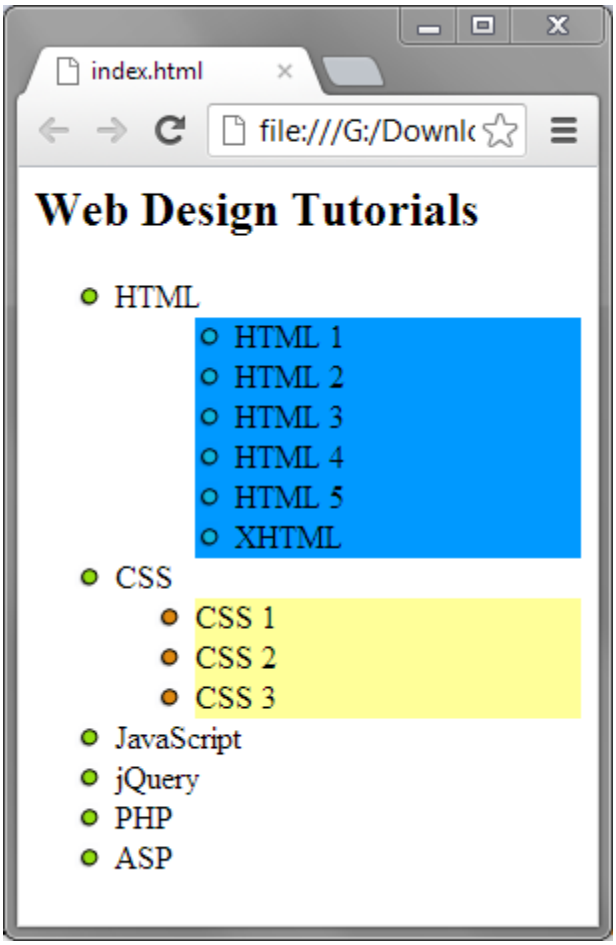
2                                   ul ol.html li {

3    list-style-image: url('images/arrow1.png');

4                           list-style-position: inside;

5    background: #09f;
```

```
6                    }
7                                ul ol.css li {
8   list-style-image: url('images/arrow2.png');
9    list-style-position: outside;
10              background: #ff9;
11 }
```



می بینید که bullet زیر مجموعه ی گزینه ی HTML که به آن مقدار inside داده ایم در کنار داخل کادر قرار گرفته، اما bullet زیر مجموعه CSS که مقدار outside دارد، خارج از کادر قرار گرفته است.

کوتاه نویسی خاصیت list style

برای کوتاه نویسی خاصیت list-style از قاعده زیر استفاده می شود.

```
1 list-style: type position url('ImageAddress');
```

مثال :

```
1 li { list-style: square inside url('bullet1.png'); }
```

بسیار خب، تا اینجا به خصوصیات مربوط به استایل دهی به list ها آشنا شدیم، در جلسه بعد به positioning یا همان موقعیت عناصر در صفحه می پردازیم.