

• آموزش CSS

یک دستور CSS از سه قسمت تشکیل شده است: سلکتور یا انتخابگر (selector)، ویژگی (property) و مقدار (value). این سه قسمت در CSS به صورت زیر نوشته می شوند:

```
selector {property: value}
```

سلکتور معمولاً یکی از تگهای HTML یا سایر عناصر صفحه است که می خواهیم ویژگیهایی را برای آن تعریف کنیم. **ویژگی ها** مشخصاتی هستند که ما می خواهیم آنها را تغییر دهیم و **مقادیر** هم مقدار آن ویژگی را تعیین می کنند که ممکن است یک اندازه برای تعیین ارتفاع، یک رنگ، آدرس یک تصویر و ... باشند. برای جدا کردن ویژگی ها از مقادیر آنها از علامت دو نقطه (:) استفاده می کنیم و ویژگیها و مقادیر را درون آکولاد ({ }) قرار می دهیم تا از سلکتور مجزا باشند. برای مثال دستور زیر برای مشخص کردن رنگ متن در صفحه استفاده می شود:

```
body {color: black}
```

تذکر: اگر مقداری که استفاده می کنید از چند کلمه مجزا تشکیل شده است از علامت دابل کوت (" ") در اطراف آن استفاده کنید:

```
body {font-family: "Times New Roman"}
```

تذکر: اگر از چند ویژگی برای یک سلکتور استفاده می کنید باید ویژگیها را با یک نقطه ویرگول (;) از هم جدا کنید. برای نمونه در مثال زیر یک پاراگراف با متن وسط چین و آبی رنگ را تعریف می کند:

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

برای اینکه استایلی که می نویسید قابل فهم تر باشد می توانید هر ویژگی را در یک سطر جداگانه بنویسید. مانند زیر:

```
p {  
font-family: Tahoma;  
font-size: 10pt;  
color: black  
}
```

• گروه بندی سلکتورها

شما می توانید از یک ویژگی برای چندین سلکتور استفاده کنید بدون آنکه نیازی باشد برای هر سلکتور کد جداگانه ای بنویسید. برای این کار باید نام همه سلکتورها را به دنبال هم بنویسید اما به خاطر داشته باشید که

سلکتور ها باید به وسیله یک کاما از هم جدا شده باشند. در مثال زیر رنگ متن را برای ه تگ متفاوت HTML مشخص می کنیم:

```
p,h1,h2,div,table {  
color: blue  
}
```

نکته: نباید بعد از آخرین سلکتور از کاما استفاده شود. این باعث می شود تا بعضی مرورگرها از استایل چشم پوشی کنند.

برای مثال نمونه زیر **غلط** است:

```
p, a, h1, { color: #333333 }
```

ولی نمونه زیر صحیح است:

```
p, a, h1 { color: #333333 }
```

چند کلاس و id نیز قابلیت گروه بندی دارند که بعداً به نحوه گروه بندی آنها می پردازیم.

کامنتها در CSS

کامنت شامل توضیحاتی است که در مورد کد داده می شود و هنگامی که بعد از اینکه مدتی از نوشتن یک کد گذشت قصد ویرایش آنرا داشته باشید به شما در فهمیدن کدها کمک می کند. کامنتها توسط مرورگر خوانده نمی شوند و در صفحه نمایش داده نمی شوند. یک کامنت در CSS با علامت (/*) آغاز می شود و با علامت (/*) پا می یابد. مثال زیر یک کامنت را در CSS نشان می دهد:

```
/* این یک کامنت است */
```

ساده ترین راه برای افزودن استایل به یک صفحه استفاده از شناسه **style** در تگهای HTML است. برای افزودن استایل به یک تگ باید عبارت " **style="** را وارد تگ مربوطه کنیم و ویژگیهای مورد نیاز خود را بین دو دابل کوت (") وارد می کنیم. مثلاً فرض کنید می خواهیم به یک تگ DIV استایل اضافه کنیم. در این صورت تگ DIV به صورت زیر در می آید.

```
<div style="property:value"
```

برای نمونه وقتی بخواهیم متن داخل یک DIV به رنگ سبز نمایش داده شود می توانیم به صورت زیر تگ DIV را بنویسیم:

<div style="color:green">متن داخل DIV </div>

توجه کنید که بین ویژگی color و رنگ سبز (green) از علامت دو نقطه (:) استفاده شده نه از علامت مساوی که در تگهای HTML استفاده می شود. همچنین لازم نیست مقادیر مورد نظر را بین علامت های نقل قول (") قرار دهیم. می توانید نتیجه به کارگیری کد بالا را در سطر زیر مشاهده کنید:

متن داخل DIV

به خاطر داشته باشید که برای اینکه خاصیت این استایل از بین برود باید تگی را که استایل در آن به کار رفته است ببندید. در اینجا تا زمانی که تگ DIV بسته نشده باشد متن ما به رنگ قرمز نمایش داده خواهد شد.

این امکان نیز وجود دارد که به یک تگ HTML بیش از یک ویژگی افزوده شود. تنها کافی است بین ویژگیهای مختلف از یک نقطه ویرگول (;) استفاده کنیم. برای مثال اگر بخواهیم متنی که در یک DIV قرار دارد به رنگ قرمز و به صورت *ایتالیک* باشد می توانیم به صورت زیر کد DIV را بنویسیم:

<div style="color:red; font-style:italic">قرمز رنگ و ایتالیک است این متن.</div>

نتیجه به صورت زیر خواهد بود:

این متن قرمز رنگ و ایتالیک است.

در این روش می توانیم به هر تعداد که لازم باشد از ویژگیهای مختلف درون یک تگ HTML استفاده کنیم. به مثال زیر توجه کنید:

<div style="color:blue; font-style:italic; font-weight:bold; font-size:12pt; text-align:center">می گیرد متن مورد نظر اینجا قرار.</div>

در این مثال ما متنی را با رنگ آبی، ایتالیک، ضخیم، اندازه فونت ۱۲pt و در وسط صفحه خواهیم داشت:

متن مورد نظر اینجا قرار می گیرد.

در قسمتهای بعدی لیستی را از ویژگیهای مختلفی که با استایل می توانیم آنها را کنترل کنیم خواهیم آورد. در اینجا فقط قصد آموزش نحوه قرارگیری استایل در تگهای HTML را داشتیم.

استفاده از استایلها می تواند به ما کمک کند تا از دوباره نویسی فرمانها جلوگیری کنیم. برای انجام این کار ما می توانیم یک Style را در بخش HEAD در متن HTML خود تعریف کنیم. یک استایل در HEAD با تگ <STYLE> آغاز می شود و با </STYLE> به پایان می رسد. در قسمت زیر می توانید یک نمونه از استایل را که مشخصاتی را برای تگ تعریف می کند مشاهده کنید که در بخش HEAD در متن قرار گرفته است.

<head>

<style>

<!--

```
span { color:red; font-style:italic }
-->
</style>
</head>
```

همانطور که می بینید استایل با تگ **<style>** آغاز می شود. سپس یک کامنت را شروع می کنیم تا مرورگرهایی که از CSS پشتیبانی نمی کنند استایل را نادیده بگیرند و در صفحه به نمایش در نیاورند. در قسمت بعد این خط را مشاهده می کنید:

```
span { color:red; font-style:italic }
```

این خط مشخص می کند که هر بار که در صفحه از تگهای **<STYLE>** **</STYLE>** استفاده می شود. متن بین آنها به رنگ قرمز و به صورت ایتالیک نمایش داده می شود. در نوشتن استایل باید در نظر داشته باشید که نباید از علامتهای کوچکتر و بزرگتر (< و >) که در اطراف تگهای HTML استفاده می شود، استفاده کنید. همچنین به جای استفاده از علامتهای نقل قول (" ") که در متن HTML استفاده می شود از آکولاد ({ }) در اطراف خواص استایل استفاده می کنیم. خواص مختلف یک استایل هم به وسیله یک نقطه ویرگول (;) از هم جدا می شوند.

وقتی ما استایلی را در بخش HEAD متن خود برای یک تگ خاص تعریف می کنیم برای محتویات همه تگهایی که استایل برای آن تعریف شده در صفحه مورد استفاده قرار می گیرد. برای مثال وقتی که ما کد بالا را در HEAD متن خود قرار دادیم، هر تگ **** که در صفحه استفاده شود دارای متنی قرمز و ایتالیک است و نیاز ندارید هر بار دوباره مشخصات متن مورد نظر را تعریف شود.

برای مثال کد زیر را وارد متن HTML خود می کنیم:

```
<span> است و &lt;span&gt; این متن خارج تگ اما </span> این متن قرمز و ایتالیک است،</span>
<br> ندارد مشخصات آنرا
</span>، این متن نیز قرمز و ایتالیک است</span>
```

نتیجه را به صورت زیر مشاهده خواهید کرد:

*این متن قرمز و ایتالیک است، اما این متن خارج تگ است و مشخصات آنرا ندارد.
این متن نیز قرمز و ایتالیک است.*

با این روش ما می توانیم تقریباً خواص همه تگهای HTML را کنترل کنیم، اما این کافی نیست. در قسمتهای بعدی روشهایی را معرفی می کنیم که یک استایل مشخص به وسیله تگهای مختلفی مورد استفاده قرار بگیرد. این کار با استفاده از CLASS و ID انجام می دهیم که به آن خواهیم پرداخت.

فرض کنید می خواهیم برای چندین صفحه از یک سری استایل استفاده کنیم که در همه صفحات مشابه هستند. در این مورد می توانیم از یک فایل استایل خارجی استفاده کنیم تا نیازی به چندین بار نوشتن استایلهای نباشد.

برای شروع با استفاده از یک ویرایشگر متن مانند Notepad در ویندوز یک فایل متنی ایجاد کنید. سپس اطلاعات زیر را در این فایل وارد کنید:

```
div { font-family:Arial }
```

```
.redtext { color:red }
```

حالا این متن را با نام example.css ذخیره کنید (شما می توانید از هر نامی که می خواهید برای ذخیره کردن فایل استفاده کنید، فقط به خاطر داشته باشید که پسوند فایل باید CSS باشد.)

برای پیوند دادن یک فایل استایل خارجی به صفحه HTML باید از تگ <LINK> استفاده کنیم که در بخش HEAD در متن HTML قرار می گیرد. یک نمونه از این تگ را می توانید در اینجا ببینید:

```
<link rel="stylesheet" type="text/css" href="URL">
```

شناسه rel="stylesheet" مشخص می کند که فایل مورد نظر شما یک استایل است.

شناسه type="text/css" مشخص کننده نوع متن فایل برای مرورگر است.

شناسه href=" " مشخص کننده محل قرار گرفتن فایل استایل شماست. شما می توانید از آدرس کامل فایل استفاده کنید و با اگر فایل شما در همان پوشه ای است که متن HTML شما قرار دارد می توانید تنها از نام فایل استفاده کنید.

حالا متن HTML مورد نظر که می خواهید استایل به آن افزوده شود را باز کنید . خط زیر را بین تگ <HEAD> و </HEAD> وارد کنید. مطمئن شوید که آدرس فایل CSS را درست وارد کرده اید:

```
<link rel="stylesheet" type="text/css" href="http://www.neopersia.org/css/example.css">
```

شما باید این کد را در همه صفحاتی که می خواهید از استایل تعریف شده در فایل قبل استفاده کنید وارد کنید وگرنه صفحات آنگونه که طراحی شده اند نمایش داده نمی شوند.

حالا هر تگ DIV که در صفحات مورد نظر شما وجود داشته باشد محتوای با فونت Arial خواهد داشت. برای مثال فایل example.css به همین صفحه پیوند شده است و در صورتی که در این صفحه از DIV استفاده شود محتوای آن با فونت Arial نمایش داده می شوند:

```
<div>شد نمایش داده خواهد Arial این متن با فونت</div>
```

این هم نتیجه:

این متن با فونت Arial نمایش داده خواهد شد.

همچنین وقتی که از کلاس redtext در تگها استفاده شود متن به رنگ قرمز نمایش داده می شود:

```
<p class="redtext">خواهد شد این متن به رنگ قرمز نمایش داده</p>
```

نتیجه به این صورت خواهد بود:

این متن به رنگ قرمز نمایش داده خواهد شد.

سلکتور کلاس

با استفاده از کلاس می توانیم برای یک تگ HTML استایل‌های متفاوتی را تعریف کنیم. برای نمونه فرض کنیم می خواهیم در صفحه دو نوع پاراگراف داشته باشیم: یکی با متن مشکی و دیگری با متن قرمز. برای مشخص کردن این دو نوع پاراگراف می توانیم از سلکتور کلاس استفاده کنیم. ابتدا استایل را به صورت زیر می نویسیم:

```
p.black {color: black}
```

```
p.red {color:red}
```

در مرحله بعد باید برای هر پاراگرافی که می خواهیم به رنگ مشکی باشد از شناسه `class="black"` در تگ `<p>` استفاده کنیم و برای پاراگراف قرمز از شناسه `class="red"`. البته نام کلاسها اختیاری است ولی بهتر است آنها را طوری انتخاب کنید که مفهوم داشته باشند تا وقتی که حجم استایل شما بیشتر شد به گنگ نباشد. در مثال زیر می توانید کد مربوط به نوشتن دو پاراگراف بالا را در متن HTML ملاحظه کنید:

```
<p class="black">شود این متنی است که به رنگ مشکی نمایش داده می</p>
```

```
<p class="red">شود این متن به رنگ قرمز نمایش داده می</p>
```

نکته : شما می توانید بیش از یک کلاس برای یک تگ HTML تعریف کنید اما برای این کار نباید دو بار از شناسه `class` استفاده کرد. بلکه باید در یک شناسه `class` نام دو کلاس مورد نظر را با یک فاصله بین آنها وارد کرد. برای مثال برای پاراگراف زیر از دو کلاس فرضی `red` و `center` استفاده شده است:

```
<p class="red center">قرار دارد center و red دو کلاس این متن پاراگرافی است که تحت تأثیر</p>
```

در نوشتن استایل می توان از نوشتن نام تگ در سلکتور کلاس چشم پوشی کرد. در این صورت ویژگیهای تعریف شده در آن کلاس بر هر تگ HTML که از کلاس مربوطه در شناسه `class` آن استفاده شده باشد اثر می گذارد. برای مثال در کلاس زیر ویژگی وسط چین را تعریف می کنیم و از نام تگ در سلکتور کلاس استفاده نمی کنیم:

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

اکنون برای هر تگی که احتیاج به وسط چین داشته باشد می توانیم از کلاس بالا استفاده کنیم. به چند نمونه در زیر توجه کنید:

```
<p class="center">داده می شود متن این پاراگراف به صورت وسط چین نمایش</p>
```

```
<div class="center">نمایش داده می شود متن این قسمت به صورت وسط چین</div>
```

```
<h2 class="center">وسط چین نمایش داده می شود متن این سرفصل به صورت</h2>
```

تذکر : هیچ وقت نام کلاس را با یک عدد شروع نکنید چون در این صورت مرورگرهای Mozilla و Firefox نمی توانند از آن کلاس استفاده کنند.

ما می توانیم از سلکتور کلاس برای تگی که حاوی شناسه `class` است استفاده کنیم که در قسمتهای قبلی در این مورد توضیح داده شد. علاوه بر این می توانیم از ویژگیهای تعریف شده در سلکتور کلاس برای تگهایی که در

یک تگ محتوی شناسه class قرار دارند نیز استفاده کنیم. مثلا برای لینکهایی که در یک DIV قرار دارند و برای DIV کلاس تعریف شده است. برای این کار در نوشتن استایل مربوطه باید پس از نقطه که مشخص کننده سلکتور کلاس است نام کلاس را وارد کنیم و پس از وارد کردن یک فضای خالی نام تگی که می خواهیم ویژگیهای بر روی آن اعمال شود. به مثال زیر توجه کنید:

در این مثال از استایل زیر در استفاده می کنیم:

```
.test { border: #F00 1px double }  
.test a { color: green }  
.test a:hover { border: yellow 2px solid }
```

کد HTML زیر برای این مثال مورد استفاده قرار می گیرد:

```
<div class="test">  
<a href="#">لینک آزمایشی شماره ۱</a>  
</div>  
<a href="#">لینک آزمایشی شماره ۲</a>
```

می توانید نتیجه را در اینجا مشاهده کنید:

[لینک آزمایشی شماره ۱](#)

[لینک آزمایشی شماره ۲](#)

همانطور که در مثال بالا می بینید خواص لینک آزمایشی شماره ۱ تحت تأثیر استایل مورد استفاده برای تگ DIV قرار دارد ولی لینک آزمایشی شماره ۲ مانند سایر لینکهای خارج از DIV نمایش داده می شود.

• افزودن استایل به نوع مشخصی از عناصر

همانطور که می دانید با وجود اینکه برخی از عناصر در صفحه با هم تفاوت دارند، از یک تگ HTML برای ایجاد آنها استفاده می شود. برای مثال برخی از این عناصر را که از تگ <input> در به وجود آوردن آنها استفاده می شود می توانید در زیر مشاهده کنید:

- **<input type="text">**
کادرهای متنی که در فرمها استفاده می شود.
- **<input type="checkbox">**
چک باکس هم با استفاده از تگ <input> به وجود می آید.
- **<input type="radio">**
در ساختن دکمه های رادیو هم از همان تگ استفاده می شود.

- **<input type="button">** این تگ در ساختن هم استفاده می شود.
- **<input type="submit">** برای ساختن دکمه ارسال فرم هم از تگ <input> استفاده می شود.

حالا فرض کنید ما در یک صفحه از چند نمونه از عناصر بالا استفاده کرده ایم ولی فقط می خواهیم ویژگیهای یکی از آنها را تغییر دهیم. دو روش برای این کار وجود دارد: یکی استفاده از کلاس است که در این روش اگر تعداد زیادی از عنصر مورد نظر در صفحه وجود داشته باشد باید به همان تعداد از شناسه class در تگهای آنها استفاده کنیم و ممکن است این کار باعث افزایش حجم صفحه یا موجب سختی در نوشتن کد شود. روش دوم تعریف کردن سلکتور input برای عناصر مورد نظر است که به صورت زیر انجام می گیرد:

فرض کنید می خواهیم فقط عرض کادرهای متنی را در صفحه تغییر دهیم. در صورتی که در استایل از سلکتور input به تنهایی استفاده کنیم عرض همه چک باکسها ، دکمه ارسال، دکمه های رادیو و سایر عناصری که از تگ <input> استفاده کرده اند نیز تغییر می کند. برای انجام این کار می توانید سلکتور را به صورت زیر تعریف کنیم:

```
input[type="text"] {
width: 200px
}
```

کد استایلی که در بالا نوشتیم فقط بر روی کادرهای متنی که در آنها از شناسه **type="text"** استفاده شده است اثر می گذارد.

البته این روش ممکن است در مرورگر Internet Explorer پشتیبانی نشود اما مرورگرهای Firefox و Opera از آن پشتیبانی می کنند.

سلکتور id

روش دیگری که برای تعریف استایل وجود دارد استفاده از id است. شناسه id احتمالاً در آینده جایگزین شناسه name در تگهای HTML می شود بنا بر این در هرصفحه نباید بیش از یک بار از هر ID استفاده شود. اگر بیش از یک بار از یک id استفاده شود کارایی صفحه تغییری نمی کند ولی صفحه اعتبار خود را از دست می دهد. سلکتور id با علامت # مشخص می شود. با استایل زیر همه عناصر که شناسه **id="border"** دارند با کادری سبز رنگ نمایش داده می شوند:

```
#border { border: green thin solid }
```

برای مثال ما از استایل بالا در این صفحه استفاده کرده ایم. حالا کد زیر را وارد صفحه می کنیم:

```
<h3 id="border">۳</h3>
<div id="border"> است این قسمت یک</div>
```


در قسمت زیر می توانید نتیجه کد بالا را مشاهده کنید:

این یک سرفصل با اندازه ۳ است

این قسمت یک div است

همانطور که کلاس را می توانستیم برای یک عنصر مشخص تعریف کنیم در مورد id نیز می توانیم این کار را انجام دهیم. به استایل زیر توجه کنید:

```
p#border { border: green thin solid }
```

استایل بالا فقط روی پاراگرافی که شناسه "id=green" در آن به کار رفته اثر می گذارد.

نکته : هرگز نام یک id را با یک عدد آغاز نکنید چون در این صورت مرورگرهای Mozilla و Firefox نمی توانند از آن استفاده کنند.

عرض و ارتفاع

width

این مشخصه به ما امکان می دهد تا عرض یک بخش از صفحه را تعیین کنیم. برای مثال می توانیم با آن عرض یک DIV را تعیین کنیم. در مثال زیر برای DIV عرضی معادل ۲۰۰ پیکسل تعیین می کنیم:

```
<div style="width:200px">این جمله پس از اینکه عرض آن</div>  
</div>
```

نتیجه به صورت زیر خواهد بود:

این جمله پس از اینکه عرض آن از ۲۰۰ پیکسل بیشتر شد در سطرهای بعدی ادامه پیدا می کند.

مقادیری که برای پارامتر width قرار می دهید می تواند بر اساس پیکسل باشد که در این صورت باید پس از مقدار مربوطه حروف px که نشانه پیکسل است را بنویسید. مقادیر ممکن است بر اساس درصد باشند که در آن صورت باید علامت درصد (%) را پس از مقدار قرار دهید.

height

این پارامتر به ما اجازه می دهد تا ارتفاع بخش مشخصی را تعیین کنیم. برای مثال وقتی که ما این پارامتر را برای یک DIV استفاده می کنیم و مقدار آن را ۱۰۰ پیکسل قرار می دهیم تا وقتی که متن موجود در DIV از ۱۰۰ پیکسل کمتر ارتفاع دارد ارتفاع 100 DIV پیکسل خواهد بود. ولی وقتی که متن داخل DIV زیاد باشد و از ۱۰۰

پیکسل بیشتر شود در نتیجه ارتفاع DIV هم از ۱۰۰ تجاوز خواهد کرد تا همه متن موجود قابل خواندن باشد. به مثال زیر توجه کنید:

۱. ابتدا هنگامی که ارتفاع متن از ارتفاع DIV کمتر است:

```
<div style="height:100px">۱۰۰</div>
```

ارتفاع این متن از ۱۰۰ پیکسل کمتر است.

۲. حالا هنگامی که ارتفاع متن از ارتفاع DIV بیشتر است:

```
<div style="height:100px">ارتفاع<br>این<br>از متن<br>100<br>پیکسل<br>کمتر<br>است.</div>
```

ارتفاع
این
متن از
۱۰۰
پیکسل
کمتر
است.

اگر بخواهیم ارتفاع DIV همان ۱۰۰ پیکسل بماند و تغییر نکند باید از پارامتر overflow در استایل استفاده کنیم. این پارامتر به ما این امکان را می دهد که تعیین کنیم وقتی محتویات DIV از عرض یا ارتفاع DIV بزرگتر می شود چگونه نمایش داده شود. در اینجا برای پارامتر overflow از مقدار hidden استفاده می کنیم که تعیین می کند هر چیزی بزرگتر از عرض یا ارتفاع DIV نمایش داده نشود. به مثال زیر توجه کنید:

```
<div style="height:100px;overflow:hidden">ارتفاع<br>این<br>از متن<br>100<br>پیکسل<br>کمتر<br>است.</div>
```

ارتفاع
این
متن از
۱۰۰
پیکسل
کمتر
است.

مقادیری که برای پارامتر height قرار می دهید می تواند بر اساس پیکسل باشد که در این صورت باید پس از مقدار مربوطه حروف px که نشانه پیکسل است را بنویسید. مقادیر ممکن است بر اساس درصد باشند که در آن صورت باید علامت درصد (%) را پس از مقدار قرار دهید.

overflow

وقتی محتویات قسمتی از صفحه طول یا عرضی بزرگتر از آن قسمت در بر گیرنده داشته باشد، با پارامتر overflow می توانیم مشخص کنیم که آن قسمت زیادی نمایش داده شود یا مخفی شود. به صورت پیش فرض قسمتهای اضافی نمایش داده می شوند ولی با این پارامتر ما می توانیم آنها را مخفی کنیم. به مثال زیر توجه کنید:

```
<div style="width:200px; overflow:hidden"><nobr>عرض بیشتر از عرض</nobr></div>
```

طول این سطر بیشتر از عرض DIV است ولی با استفاده از overflow قسمتهای بیشتر از عرض DIV دیده نمی شوند.

به طور کلی چند مقدار را می توانیم برای پارامتر overflow استفاده کنیم:

۱. visible

یا قرار دادن این مقدار برای overflow همه محتویاتی که طول یا عرض بیش از اندازه داشته اند نمایش داده خواهند شد.

۲. hidden

این مقدار باعث خواهد شد تا محتویات با طول یا عرض بیش از اندازه نمایش داده نشوند.

۳. scroll

این مقدار باعث خواهد شد تا در صورتی که محتویات اندازه ای بیش از طول و عرض لازم دارند با استفاده از اسکرول بار قابل مشاهده باشند.

به یک مثال توجه کنید:

```
<div style="width:200px; overflow:scroll; border:dashed 1px #3399FF; white-space:nowrap">
```

DIV قسمتهای بیشتر از عرض overflow استفاده از است ولی با DIV طول این سطر بیشتر از عرض

شوند دیده نمی

```
</div>
```

این هم نتیجه:

طول این سطر بیشتر از عرض DIV است ولی با استفاده از overflow قسمتهای بیشتر از عرض DIV دیده نمی شوند.

border-style

با استفاده از استایل ما می توانیم برای اجزای یک صفحه کادر ایجاد کنیم. با استفاده از پارامتر border-style می توانیم مشخص کنیم که آیا می خواهیم کادر داشته باشیم یا نه، و اگر می خواهیم نوع کادری را که می

خواهیم ایجاد کنیم مشخص کنیم. در اینجا می توانید برخی از مقادیری را که می توانیم برای border-style استفاده کنیم را ببینید:

none .۱

با استفاده از این مقدار تعیین می کنیم که نمی خواهیم کادر داشته باشیم. در صورتی که مقداری غیر از none تعیین شود کادر خواهیم داشت.

solid .۲

DIV have a solid border This

dashed .۳

DIV have a dashed border This

dotted .۴

DIV have a dotted border This

double .۵

DIV have a double border This

groove .۶

DIV have a groove border This

hidden .۷

This DIV have a hidden border

inset .۸

DIV have a inset border This

outset .۹

DIV have a outset border This

ridge .۱۰

DIV have a ridge border This

همانطور که می بینید عرض این کادرها بیش از آن چیزی است که ما نیاز داریم چون ما برای این کادرها عرضی مشخص نکردیم. اگر بخواهیم عرض این کادرها به اندازه مشخصی باشد باید از پارامتر width که قبلاً با آن آشنا شدیم استفاده کنیم. به مثال زیر توجه کنید:

```
<div style="width:300px; border-style:solid">این کادر عرض ۳۰۰ پیکسل تعیین شده برای /div> است.
```

نتیجه به این صورت خواهد بود:

برای این کادر عرض ۳۰۰ پیکسل تعیین شده است.

border-width

این پارامتر به ما اجازه می دهد تا میزان ضخامت کادرها را تعیین کنیم. این پارامتر می تواند مقادیر زیر را داشته باشد:

مقدار عددی بر حسب پیکسل .۱

در این روش می توانیم مقدار ضخامت کادر را بر حسب پیکسل تعیین کنیم. به مثال زیر توجه کنید:

`<div style="border-style:solid; border-width:10px">` این کادر ۱۰ پیکسل ضخامت دارد

این هم کادر به دست آمده:

این کادر ۱۰ پیکسل ضخامت دارد

۲. thin

این مقدار باعث خواهد شد تا کادر ما باریک باشد. مانند زیر:

`<div style="border-style:solid; border-width:thin">` کادر thin ضخامت این کادر شده است.

این هم کادر به دست آمده:

ضخامت این کادر thin در نظر گرفته شده است.

۳. thick

این مقدار به ما یک کادر ضخیم خواهد داد. به مثال زیر توجه کنید:

`<div style="border-style:solid; border-width:thick">` کادر thick ضخامت این کادر شده است.

نتیجه به صورت زیر خواهد بود:

ضخامت این کادر thick در نظر گرفته شده است.

border-color

شما می توانید در صورت تمایل رنگ کادرهای خود را تغییر دهید. برای این کار باید از پارامتر `border-color` استفاده کنید. برای مقدار این پارامتر می توانید از نام رنگها استفاده کنید. در صورتی که از نام رنگها استفاده کنید فقط می توانید از تعداد کمی از رنگها استفاده کنید. راه دیگری که برای انتخاب رنگ وجود دارد استفاده از معادل هگزادسیمال رنگها است که نسبت به نام رنگها انعطاف پذیری بیشتری دارد. اگر جدول معادل هگزادسیمال رنگها

را می خواهید [اینجا را کلیک کنید](#).
به یک نمونه برای تغییر رنگ کادر توجه کنید:

استفاده از نام رنگها:

<div style="border-style:solid; border-color:green">این کادر به رنگ سبز خواهد بود.</div>

این کادر به رنگ سبز خواهد بود.

استفاده از معادل هگزاسیما رنگها:

<div style="border-style:solid; border-color:#008000">این کادر به رنگ سبز خواهد بود.</div>

این کادر به رنگ سبز خواهد بود.

margin-left

این گزینه به ما اجازه می دهد تا برای هر یک از اجزای صفحه حاشیه چپ مشخصی را تعیین کنیم. در حقیقت این گزینه مقدار فاصله اجزا را از سمت چپ صفحه مشخص می کند. به یک مثال در این مورد توجه کنید:

<div style="margin-left:200px; border:double">دویست پیکسل از سمت چپ صفحه DIV این</div> فاصله دارد

همانطور که می بینید کد بالا یک DIV را با کادر مضاعف و ۲۰۰ پیکسل حاشیه از چپ نشان می دهد:

این DIV دویست پیکسل از سمت چپ صفحه فاصله دارد.

margin-right

این گزینه هم مانند گزینه بالا عمل می کند با این تفاوت که مقدار حاشیه را از سمت راست تعیین می کند:

<div style="margin-right:200px; border:double">دویست پیکسل از سمت راست صفحه DIV این</div> فاصله دارد

همانطور که می بینید کد بالا یک DIV را با کادر مضاعف و ۲۰۰ پیکسل حاشیه از راست نشان می دهد:

این DIV دویست پیکسل از سمت راست صفحه فاصله دارد.

margin-top

این گزینه هم مانند گزینه های بالا کار می کند با این تفاوت که میزان حاشیه را از بالای صفحه تعیین می کند:

`<div style="margin-top:100px; border:double">` پیکسل از بالای صفحه فاصله صد DIV این `</div>` دارد.

این DIV صد پیکسل از بالای صفحه فاصله دارد.

margin-bottom

این گزینه هم مانند گزینه های بالا کار می کند با این تفاوت که میزان حاشیه را از پایین صفحه تعیین می کند:

`<div style="margin-bottom:100px; border:double">` پیکسل از پایین صفحه فاصله صد DIV این `</div>` دارد.

این DIV صد پیکسل از پایین صفحه فاصله دارد.

Padding

padding-right

پارامتر padding در CSS مشابه شناسه cellpadding در جدولهاست که با این تفاوت که می توان آنرا برای بالا، پایین، چپ و راست به صورت جداگانه تعریف کرد به جای اینکه به صورت یکجا تعیین شود. در اینجا می توانید یک نمونه از padding را که برای راست تعریف شده است را ببینید:

`<div style="padding-right:100px; border-style:double">` معادل ۱۰۰ پیکسل DIV این برای `</div>` padding شده است

می توانید نتیجه را در اینجا مشاهده کنید:

برای این DIV معادل ۱۰۰ پیکسل padding تعریف شده است.

padding-top

مانند قسمت قبل است با این تفاوت که به جای راست برای بالا تعریف می شود:

`<div style="padding-top:30px; border-style:double">` padding معادل ۳۰ پیکسل DIV این برای `</div>` تعریف شده است

می توانید نتیجه را در اینجا مشاهده کنید:

برای این DIV معادل ۳۰ پیکسل padding تعریف شده است.

padding-left

مانند padding-right است با این تفاوت که برای چپ تعیین می شود.

padding-bottom

مانند padding-top است با این تفاوت که برای پایین تعیین می شود.

به یاد داشته باشید که در صورتی که پارامتر padding بدون top, left, right یا bottom به کار رود مقدار وارد شده برای هر چهار جهت مورد استفاده قرار می گیرد، البته در مورد margin هم به همین صورت است:

بر روی padding پارامتر DIV در این `<div style="padding:35px; width:250px; border-style:solid ">` میگذارد هر چهار جهت تأثیر

این هم نتیجه:

در این DIV پارامتر padding بر روی هر چهار جهت تأثیر میگذارد.

position

ویژگی position به ما اجازه می دهد تا تعیین کنیم که آیا می خواهیم موقعیت یک عنصر در صفحه مشخص شود یا نه. اگر می خواهیم آن عنصر موقعیت مشخصی داشته باشد موقعیت عنصر را در صفحه تعیین می کند. چندین مقدار را می توان برای این ویژگی قرار داد. در اینجا می توانید این مقادیر را با کمی توضیح در مورد هر کدام ببینید:

• **static** :

این مقدار پیش فرضی است که در صورت تعریف نشدن position در صفحه استفاده می شود. با این مقدار اجزا در همان جایی که کد آنها را می نویسید دیده می شوند.

• **absolute** :

این مقدار به ما اجازه می دهد تا یک عنصر از صفحه را در هر موقعیتی نسبت به بالا، راست، پایین یا چپ صفحه که بخواهیم نمایش دهیم.

• fixed :

این گزینه به ما اجازه می دهد تا موقعیت یک عنصر صفحه را در پنجره مرورگر مشخص کنیم. در صورتی که برای یک عنصر از این ویژگی استفاده کنیم، موقعیت آن عنصر حتی در صورتی که به بالا و پایین صفحه برویم یا اندازه پنجره مرورگر را تغییر دهیم به همان صورت قبلی باقی می ماند. البته اینترنت اکسپلورر در ویندوز از این ویژگی پشتیبانی نمی کند ولی اینترنت اکسپلورر در مکینتاش همچنین مرورگرهای نت اسکایپ از این ویژگی پشتیبانی می کنند.

static

این ویژگی در حقیقت مشخص می کند که عنصر مربوطه باید در محلی که کد آن نوشته شده نمایش داده شود. وقتی که از این ویژگی استفاده می کنیم ویژگیهای right, top, left, bottom کاربردی ندارند.

این ویژگی به صورت پیش فرض از طرف مرورگر برای صفحه مورد استفاده قرار می گیرد.

fixed

چون اینترنت اکسپلورر بیشتر از سایر مرورگرها مورد استفاده قرار می گیرد و از ویژگی fixed در CSS پشتیبانی نمی کند زیاد به این ویژگی نمی پردازیم. اما در هر صورت ویژگی جالبی است. در اینجا فقط به ذکر یک مثال بسنده می کنیم:

در این مثال از ویژگی position:fixed برای تعیین موقعیت یک DIV استفاده می کنیم:

یا مرورگرهایی مثل netscape اگر با مرورگر <div style="position:fixed; top:50%; float:right"> در میانه صفحه و سمت راست آن به نمایش div صفحه بازدید کنید محتویات این از این opera و firefox </div>.آیند در می

اگر می خواهید ببینید این ویژگی چگونه کار می کند لینک زیر را با یکی از مرورگرهای Opera, Netscape یا Firefox مشاهده کنید و صفحه را با استفاده از اسکرول بار به بالا و پایین ببرید:

[استفاده از ویژگی position:fixed \(لینک در پنجره جدید باز می شود \)](#).

absolute

ویژگی position:absolute مشخص می کند که عنصر مربوطه باید در محل تعیین شده به نمایش در آید نه محلی که کد آن نوشته شده است.

در اینجا می توانید یک مثال را در مورد استفاده از ویژگی position:absolute در صفحه ببینید:

<div style="position:absolute; top:65px; right:20px; border:double #0033cc; width:350px; color:#ffff00; background-color:#33cccc; height: 40px;">این div در کد در موقعیت تعیین شده در کد </div> شود آن نمایش داده می

همانطور که در کد بالا می بینید برای DIV موقعیتی با فاصله ۶۵ پیکسل از بالای صفحه و ۲۰ پیکسل از سمت راست صفحه تعیین کرده ایم. با کلیک کردن بر روی لینک زیر می توانید نتیجه را مشاهده کنید:

[استفاده از absolute برای ویژگی position \(لینک در یک پنجره جدید باز خواهد شد\).](#)

top

این ویژگی موقعیت یک عنصر را از بالای صفحه تعیین می کند. (ویژگیهای top, left, right, bottom بیشتر زمانی مورد استفاده قرار می گیرند که برای یک عنصر ویژگی position تعیین شده باشد.)

در مثال قبل همانطور که ملاحظه می کنید از ویژگی top:65px استفاده شده است که این ویژگی معین می کند که DIV با فاصله ۶۵ پیکسل از بالای پنجره مرورگر به نمایش در می آید.

مقداری که برای ویژگیهای top, bottom, left, right در نظر گرفته می شود به صورت می تواند باشد. یا به صورت عددی در واحد پیکسل که در این صورت باید پس از عدد مربوطه از حروف اختصاری px استفاده شود یا به صورت درصد که پس از عدد با id از علامت درصد (%) استفاده شود.

bottom

کاربردی همانند ویژگی بالا دارد با این تفاوت که میزان فاصله عناصر را از پایین پنجره مرورگر مشخص می کند. در مثال بعدی از این ویژگی استفاده خواهیم کرد.

right

این ویژگی فاصله عناصر را از سمت راست صفحه تعیین می کند. همانطور که در مثال قبل می بینید، از این ویژگی به صورت right:20px استفاده شده است یعنی DIV مربوطه با فاصله ۲۰ پیکسل از سمت راست پنجره مرورگر قرار خواهد گرفت.

left

این ویژگی موقعیت عنصر را از سمت چپ پنجره مرورگر تعیین می کند.

در مثال زیر از ویژگیهای bottom و left برای همان DIV مثال قبل استفاده می کنیم:

```
<div style="position:absolute; bottom:65px; left:20px; border:double #0033cc; width:350px; color:#ffff00; background-color:#33cccc; height: 40px;">این div در کد در موقعیت تعیین شده در کد این div است.</div>
```

برای اینکه نتیجه را مشاهده کنید لینک زیر را کلیک کنید:

[استفاده از ویژگیهای bottom و left \(لینک در پنجره جدید باز می شود \).](#)

در مثال ذکر شده در این قسمت برخی از ویژگیهای متن را که در CSS استفاده می شوند به شما معرفی می کنیم. با کمک این ویژگیها می توانیم نحوه قرار گرفتن متن در صفحه، فاصله بین حروف و بسیاری از خواص یک متن را تغییر دهیم.

letter-spacing

این ویژگی به ما اجازه می دهد تا فاصله میان حروف یک کلمه را در یک متن تعیین کنیم. این ویژگی می تواند مقادیری را در مقیاس پیکسل به خود بگیرد یا از مقدار normal استفاده کند. گزینه پیش فرض برای این ویژگی همان normal است. در اینجا به ذکر یک مثال در مورد این ویژگی می پردازیم:

فرض کنید می خواهیم از این ویژگی برای متن موجود در یک تگ DIV استفاده کنیم:

```
<div style="letter-spacing:5px">
```

فاصله ۵ پیکسل از یکدیگر نمایش داده خواهند شد حروف این متن با

```
</div>
```

نتیجه را به صورت زیر مشاهده خواهید کرد:

حروف این متن با فاصله ۵ پیکسل از یکدیگر نمایش داده خواهند شد.

text-align

این ویژگی به ما اجازه می دهد تا نحوه قرارگیری قسمتی از متن را از لحاظ راست چین، چپ چین یا وسط چین بودن تنظیم کنیم.

این ویژگی می تواند مقادیر زیر را به مورد استفاده قرار دهد:

• **: left**

با استفاده از این مقدار می توانیم محتویات یک تگ را در سمت چپ صفحه نمایش دهیم.

• **: right**

این مقدار معین می کند که محتویات یک تگ باید در سمت راست صفحه نمایش داده شوند.

• **: center**

این مقدار محتویات یک تگ را به صورت وسط چین تعریف میکند.

برای مثال اگر می خواهید متنی که در یک DIV قرار دارد به صورت چپ چین نمایش داده شود می توانیم از این ویژگی به مانند زیر استفاده کنیم:

در سمت چپ DIV محتویات این `<div style="text-align:left; border:thin dashed #00CCFF">این</div>` صفحه نمایش داده می شوند

نتیجه به صورت زیر خواهد بود:

محتویات این DIV در سمت چپ صفحه نمایش داده می شوند.

text-decoration

این ویژگی به ما اجازه می دهد تا برای قسمت خاصی از متن خاصیت‌های از قبیل زیرخط تعریف کنیم. مثلاً فرض کنید می خواهیم یک متن را با خطی بالای آن به وجود بیاوریم. در اینجا می توانیم از این ویژگی به صورت زیر استفاده متین:

!!توجه کنید به این متن `<div style="text-decoration: underline">!!توجه کنید به این متن</div>`

این هم نتیجه:

به این متن توجه کنید!!

این ویژگی چندین مقدار دارد که در اینجا به آنها می پردازیم:

• **: none**

این گزینه به صورت پیش فرض بر روی متن اعمال می شود (البته برای لینکها به صورت پیش فرض underline

اعمال می شود) و در نتیجه آن متنها به صورت ساده و معمولی نمایش داده می شوند. در صورتی که بخواهیم لینکهایی در متن ما بدون زیر خط نشان داده شوند می توانیم از این گزینه برای تگ لینک استفاده کنیم.



• **: underline**

از این گزینه می توان برای زیر خط دار کردن متنها استفاده کرد.



• **: overline**

این گزینه برای نمایش خط بالای متن استفاده می شود.



• **: line-through**

این گزینه یک خط را در میانه متن ایجاد می کند و آنرا به صورت خط خورده نشان می دهد.



• **: blink**

از این گزینه می توانیم برای چشمک زن کردن متن استفاده کنیم. البته این گزینه توسط مرورگر اینترنت اکسپلورر حمایت نمی شود و فقط در مرورگرهای Netscape نمایش داده می شود.

در اینجا یک لینک را می بینید که با استفاده از ویژگی بالا بدون زیر خط نمایش داده می شود:

[مورد استفاده در لینک متن](#)

این هم لینک بدون زیر خط:

[متن مورد استفاده در لینک](#)

DIV به اندازه ۶۵ پیکسل از پایین و ۲۰ پیکسل از سمت چپ پنجره مرورگر فاصله دارد.

در این قسمت به ادامه ویژگیهایی که بر روی متن ها اثر می گذارند می پردازیم.

text-transform

این ویژگی مشخص می کند که متن با استفاده از حروف بزرگ نوشته شود یا حروف کوچک و یا به صورت معمولی البته بدون آنکه در حروفی که تایپ شده است تغییری به وجود آید. با این روش اگر خواستید قسمتی از

متن مثلاً برای تأکید بیشتر با حروف بزرگ مشاهده شود نیازی نیست که دوباره آن قسمت را از اول تایپ کنید. البته این ویژگی در زبان فارسی چندان استفاده ای ندارد ولی در زبانهای مثل انگلیسی می توانید آنرا به کار ببرید.

به یک مثال در مورد این ویژگی توجه کنید:

```
<div style="text-transform: uppercase">This is an example of uppercase in text-transform property.</div>
```

این هم نتیجه:

THIS IS AN EXAMPLE OF UPPERCASE IN TEXT-TRANSFORM PROPERTY.

همانطور که مشاهده می کنید بدون استفاده از حروف بزرگ در نوشتن کد، متن ما با حروف بزرگ نمایش داده می شود.

این ویژگی چند مقدار دارد که در اینجا مشاهده می کنید:

• **: none**
با انتخاب این گزینه متن به همان صورتی که تایپ شده است نمایش داده می شود.

• **: lowercase**
با انتخاب این گزینه متن با حروف کوچک نمایش داده می شود.

• **: uppercase**
این گزینه متن را با حروف بزرگ نمایش می دهد.

• **: capitalize**
این گزینه حروف اول کلماتی را که با حروف کوچک نوشته شده اند با حروف بزرگ و بقیه حروف را با حروف کوچک نمایش می دهد. به مثال زیر توجه کنید:

```
<div style="text-transform: capitalize">this is an example of capitalize value for text-transform propety</div>
```

متن به صورت زیر نمایش داده می شود:

this is an example of capitalize value for text-transform property

line-height

این ویژگی به ما امکان می دهد تا ارتفاع سطرها را مشخص کنیم. این ارتفاع می تواند به صورت درصد یا در مقیاس پیکسل باشد و فاصله بین سطرها را مشخص می کند.

برای مثال اگر بخواهیم سطرها از هم ۵۰ پیکسل فاصله داشته باشند می توانیم به صورت زیر عمل کنیم:

```
<div style="line-height: 50px">سطر و سطر<br>20 پیکسل فاصله وجود دارد.</div>
```

نتیجه به صورت زیر خواهد بود:

بین این سطر و سطر بعدی

۲۰ پیکسل فاصله وجود دارد.

text-indent

این ویژگی مقدار تو رفتگی متن را در سطر اول هر قسمت (مثلاً سطر اول هر پاراگراف) مشخص می کند. این مقدار می تواند بر حسب پیکسل یا درصد باشد.

در مثال زیر این مقدار تو رفتگی را ۳۰ پیکسل در نظر می گیریم:

```
<div style="text-indent: 30px"><br>با ۳۰ پیکسل تو رفتگی نمایش داده می شود سطر اول این قسمت<br>به صورت عادی و در حالی که بقیه سطرها به<br>بدون تو رفتگی نمایش داده می شوند</div>
```

این هم نتیجه:

سطر اول این قسمت با ۳۰ پیکسل تو رفتگی نمایش داده می شود.
در حالی که بقیه سطرها به صورت عادی و بدون تو رفتگی نمایش داده می شوند.

در این قسمت به روش تنظیم فونت متن با استفاده از استایل می پردازیم. بسیاری از اعمالی را که قبلاً با تگ انجام می دادیم می توانیم با استایل انجام دهیم.

font-family

ویژگی font-family مشخص می کند که متن با چه فونتی باید نشان داده شود. این ویژگی در حقیقت جانشین شناسه face که در تگ font استفاده می شود است. گزینه پیش فرض برای فونت ممکن است در مرورگرهای مختلف تفاوت داشته باشد، اما معمولاً Roman Times New است.

در مثال زیر از ویژگی font-family برای تغییر فونت متن داخل یک div به Arial استفاده می کنیم:

<div style="font-family: Arial">فونت این متن با فونت</div>

نتیجه را به صورت زیر مشاهده می کنید:

این متن با فونت Arial نمایش داده می شود.

البته باید به خاطر داشته باشید که اگر بازدید کننده فونت مورد نظر شما را نداشته باشد، مرورگر به جای آن از فونت پیش فرض خود استفاده می کند. برای جلوگیری از بروز این مشکل باید از فونتهایی استفاده کنید که معمولاً به همراه سیستم عامل نصب می شوند و در اکثر سیستمها موجود هستند.

font-size

این ویژگی مشخص کننده اندازه فونت در صفحه می باشد. گزینه پیش فرض برای این ویژگی medium است.

فرض کنید می خواهیم متن درون یک DIV با اندازه ۳۰ پیکسل نمایش داده شود:

<div style="font-size: 30px">۳۰ پیکسل نمایش داده می شود این متن با اندازه</div>

نتیجه را ملاحظه کنید:

این متن با اندازه ۳۰ پیکسل نمایش داده می شود.

font-style

کار این ویژگی تقریباً شبیه تگ <I> در HTML است، اما چون استفاده از تگهایی نظیر , , <U> و <I> در حال کمرنگ شدن است بهتر است از استایل برای

در این قسمت به روش تنظیم فونت متن با استفاده از استایل می پردازیم. بسیاری از اعمالی را که قبلاً با تگ انجام می دادیم می توانیم با استایل انجام دهیم.

font-family

ویژگی font-family مشخص می کند که متن با چه فونتی باید نشان داده شود. این ویژگی در حقیقت جانشین شناسه face که در تگ font استفاده می شود است. گزینه پیش فرض برای فونت ممکن است در مرورگرهای مختلف تفاوت داشته باشد، اما معمولاً Roman Times New است.

در مثال زیر از ویژگی font-family برای تغییر فونت متن داخل یک div به Arial استفاده می کنیم:

<div style="font-family: Arial">نمایش داده می شود Arial این متن با فونت</div>

نتیجه را به صورت زیر مشاهده می کنید:

این متن با فونت Arial نمایش داده می شود.

البته باید به خاطر داشته باشید که اگر بازدید کننده فونت مورد نظر شما را نداشته باشد، مرورگر به جای آن از فونت پیش فرض خود استفاده می کند. برای جلوگیری از بروز این مشکل باید از فونتهایی استفاده کنید که معمولاً به همراه سیستم عامل نصب می شوند و در اکثر سیستمها موجود هستند.

font-size

این ویژگی مشخص کننده اندازه فونت در صفحه می باشد. گزینه پیش فرض برای این ویژگی medium است.

فرض کنید می خواهیم متن درون یک DIV با اندازه ۳۰ پیکسل نمایش داده شود:

<div style="font-size: 30px">۳۰ پیکسل نمایش داده می شود این متن با اندازه</div>

نتیجه را ملاحظه کنید:

این متن با اندازه ۳۰ پیکسل نمایش داده می شود.

font-style

کار این ویژگی تقریباً شبیه تگ <I> در HTML است، اما چون استفاده از تگهایی نظیر , , <U> و <I> در حال کمرنگ شدن است بهتر است از استایل برای مورب نشان دادن متن استفاده شود. این ویژگی می تواند مقادیر زیر را داشته باشد:

- normal •
- italic •
- oblique •

به یک مثال در مورد این ویژگی توجه کنید:

<div style="font-style:oblique">مشاهده می شود این متن به صورت مورب</div>

نتیجه به این صورت خواهد بود:

این متن به صورت مورب مشاهده می شود.

font-weight

این ویژگی به ما امکان می دهد تا میزان ضخامت متن را تعیین کنیم. می توانیم از مقادیر زیر برای این ویژگی استفاده کنیم:

- normal •
- bold •
- bolder •
- lighter •
- ۱۰۰ •
- ۲۰۰ •
- ۳۰۰ •
- ۴۰۰ •

- ۵۰۰
- ۶۰۰
- ۷۰۰
- ۸۰۰
- ۹۰۰

در اینجا به ذکر یک مثال می پردازیم:

<div style="font-weight: 700">مشاهده می شود این متن تقریباً ضخیم</div>

نتیجه به این صورت خواهد بود:

این متن تقریباً ضخیم مشاهده می شود.

از ویژگی **color** برای تغییر رنگ متن قسمت‌های مختلف صفحه استفاده می شود. برای مثال می توانید رنگ متن یک جدول را تغییر دهیم و یا قسمتی از متن صفحه را که با تگ SPAN مشخص کرده ایم با رنگ متفاوتی با بقیه متن نمایش دهیم. این تگ از آن جهت که کاربرد تگ و در نتیجه شناسه color که در تگ فونت استفاده می شود، در HTML4 در حال محدود شدن است اهمیت بیشتری پیدا می کند.

البته در قسمت‌های قبلی هم از این ویژگی استفاده کرده ایم اما در باره آن توضیحی داده نشده است.

مانند COLOR در تگ FONT رنگها در این ویژگی به دو صورت تعیین می شوند. شکل کلی این ویژگی به صورت زیر است:

نام رنگ مورد نظر: color:

معادل هگزادسیمال رنگ: color:#

اگر می خواهید از رنگهای معمولی که با نام شناخته می شوند استفاده کنید کافی است تا نام رنگ مورد نظر خود را در این ویژگی استفاده کنید. این روش ساده تر است ولی انعطاف پذیری کمتری هم دارد چون تنها می توانید رنگ مورد نظر خود را از بین تعداد اندکی نام رنگ انتخاب کنید.

در صورتی که می خواهید از رنگهای بیشتری استفاده کنید و بهتر بتوانید رنگ مورد نظر خود را انتخاب کنید می توانید از معادل هگزادسیمال رنگها استفاده کنید. در این صورت می توانید از ۲۱۶ رنگ مورد استفاده در وب رنگ مورد نظر خود را انتخاب کنید.

اگر جدول مربوط به معادل هگزادسیمال رنگها را می خواهید [اینجا را کلیک کنید](#). (لینک در پنجره جدید باز می شود)

در اینجا به ذکر چند مثال برای استفاده از ویژگی color می پردازیم. ما در اینجا از شناسه style برای تعریف کردن این ویژگی استفاده می کنیم اما به خاطر داشته باشید که می توانیم این ویژگی را برای یک تگ مشخص یا به صورت class و id در بخش HEAD متن یا استایل خارجی نیز به کار ببریم.

برای مثال در استفاده از color برای تغییر رنگ متن یک جدول می توانیم کد جدول را به صورت زیر بنویسیم:

```
<table border="1" style="color: red">
  <tr>
    <td>
      قرمز نمایش داده می شود این متن به رنگ
    </td>
  </tr>
</table>
```

نتیجه کد بالا با کدی که در پایین آمده است یکسان است:

```
<table border="1" style="color: #FF0000">
  <tr>
    <td>
      رنگ قرمز نمایش داده می شود این متن به
    </td>
  </tr>
</table>
```

نتیجه هر دو کد بالا به صورت زیر است:

این متن به رنگ قرمز نمایش داده می شود .

به یاد داشته باشید که می توانید از ویژگی color برای هر کدام از سطرها یا سلولهای یک جدول به صورت جداگانه استفاده کنید. فقط کافیست تا **style="color: XXX"** را در هر قسمت قرار دهید و رنگ مورد نظر خود را به جای XXX قرار دهید.

background-color

این ویژگی به ما اجازه می دهد تا تقریباً برای هر چیزی در صفحه رنگ زمینه مشخص کنیم. برای این کار باید این ویژگی را به همراه رنگ مورد نظر وارد تگ مربوط به قسمت مورد نظر کنیم. اگر بخواهیم زمینه قسمتی از یک متن را مشخص کنیم می توانیم از این ویژگی در یک تگ SPAN در اطراف متن مورد نظر استفاده کنیم.

در مثال زیر زمینه قسمتی از یک متن را به رنگ زرد در می آوریم:

```
<p>این قسمت زمینه زرد <span style="background-color: yellow">. متن بدون زمینه</span></p>
```

این هم متن نهایی:

متن بدون زمینه. این قسمت زمینه زرد دارد.

همینطور می توانیم از این ویژگی برای تغییر زمینه یک جدول یا DIV هم استفاده کنیم:

```
<table border="1" style="background-color: #FFC0CB">
<td>سلول اول</td>
<td>سلول دوم</td>
</table>
```

این هم جدول به دست آمده:

سلول اول	سلول دوم
----------	----------

شما حتی می توانید از این ویژگی برای سلولهای یک جدول به صورت جداگانه استفاده کنید.

یک نمونه دیگر از موارد استفاده این ویژگی در طراحی فرمهاست. به مثال زیر توجه کنید:

```
<form action="some_script">
نام : <input type="text" size="30" style="background-color: #009900">
</form>
```

این هم فرم آزمایشی ما:

نام :

توجه داشته باشید که در این ویژگی هم می توانیم از نام رنگ ها و هم از [معادل هگزادسیمال](#) آنها استفاده کنیم.

شما می توانید با استفاده از این ویژگی در تگ body رنگ زمینه صفحه را هم تغییر دهید. در مورد تصویر زمینه هم که در قسمت بعد به آن می پردازیم می توان به همین صورت عمل کرد.

background-image

این ویژگی برای مشخص کردن تصویر زمینه به کار می رود. کاربرد **background-image** همانند **background-color** است با این تفاوت که به جای مشخص کردن رنگ باید یک تصویر را تعیین کنیم. این ویژگی به صورت زیر نوشته می شود.

background-image:url(آدرس تصویر مورد نظر)

بعد از نام ویژگی باید از url برای مشخص کردن آدرس تصویر استفاده کنیم. آدرس تصویری که می خواهیم به عنوان زمینه استفاده کنیم نیازی نیست بین علامتهای نقل قول (") قرار گیرد ولی باید بین دو پرانتز قرار گیرد. به یک مثال در این رابطه توجه کنید:

```
<span style="background-image: url(http://www.neopersia.org/css/example1.gif)"> برای این </span>  
</span> استفاده شده است از تصویر زمینه
```

می توانید نتیجه کد بالا را مشاهده کنید:

برای این span از تصویر زمینه استفاده شده است.

از این ویژگی می توان تقریباً برای همه اجزای صفحه استفاده کرد. در اینجا برای یک جدول از آن استفاده کرده ایم:

```
<table border="1" style="background-image:  
url(http://www.neopersia.org/css/example1.gif)">  
<td>اول محتویات سلول</td>  
<td>محتویات سلول دوم</td>  
</table>
```

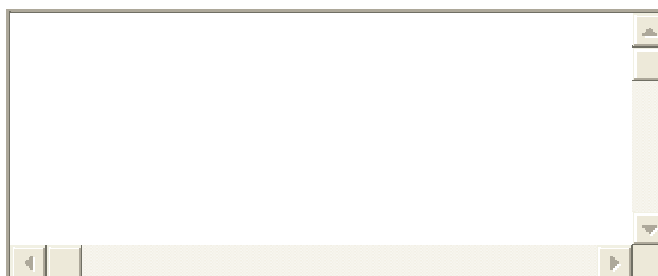
جدول به دست آمده مانند زیر است:

محتویات سلول اول	محتویات سلول دوم
------------------	------------------

همانطور که برای text box از رنگ زمینه استفاده کردیم برای آن از تصویر زمینه هم می توانیم استفاده کنیم. در اینجا از این ویژگی برای یک textarea استفاده می کنیم:

```
<form>  
<textarea rows="8" cols="60" style="background-  
image:url(http://www.neopersia.org/css/example1.gif)">  
</textarea>  
</form>
```

نتیجه را ملاحظه کنید:



ما با استفاده از شناسه **background** در تگ **BODY** می توانستیم برای صفحه تصویر زمینه مشخص کنیم. اما با استفاده از استایل می توانیم کنترل بیشتری بر این تصویر زمینه داشته باشیم چه این تصویر زمینه برای کل صفحه استفاده شود چه برای قسمت مشخصی از صفحه. برای مثال می توانیم تصویر زمینه را به گونه ای تعریف کنیم که تکرار نشود، ثابت باشد، یا برای آن نقطه شروع تعیین کنیم. برخی از این خواص را می توانید در جدول زیر ملاحظه کنید:

ویژگی	مقادیر قابل استفاده	توضیحات
background-image: url(url)	آدرس محل قرار گرفتن تصویر	مشخص کننده محل قرار گرفتن تصویر در اینترنت است
background-repeat: مقدار ویژگی	repeat, no-repeat, repeat-x, repeat-y	نوع تکرار تصویر را مشخص می کند
background-attachment: مقدار ویژگی	scroll, fixed	مشخص می کند که آیا تصویر زمینه با اسکرول کردن صفحه یا در جای خود حرکت می کند ثابت می ماند
background-position: مقدار مقدار اول دوم	مقدار اول: top, center, bottom, پیکسل، عددی در مقیاس در مقیاس درصد مقدار دوم: left, center, right, عددی در مقیاس پیکسل، عددی در مقیاس درصد	موقعیت تصویر زمینه در صفحه را مشخص می کند

برای استفاده از این ویژگیها و کاربرد آنها برای تگ BODY می توانید یک استایل در قسمت HEAD متن تعریف کنید یا از استایل خارجی استفاده کنید. مثلاً برای تصویر زمینه صفحات سایت ما می توان در قسمت HEAD متن یک استایل به صورت زیر تعریف کرد:

```
<style type="text/css">
<!--
body {
background-image:url(http://www.neopersia.org/images/index.jpg);
background-repeat:repeat-y }
-->
</style>
```

در این صورت تصویر زمینه ای مشابه سایت ما خواهید داشت. می توانید آنرا امتحان کنید.

background-repeat

همانطور که در درس قبل ملاحظه کردید در مورد [background-image](#) توضیحاتی داده شد. در اینجا به سایر ویژگیهای مربوط به تصویر زمینه می پردازیم.

این ویژگی تعیین کننده نوع تکرار شدن تصویر زمینه است. در اینجا مقادیری را که می توان به این ویژگی اختصاص داد بیان می کنیم:

- **repeat**
این مقدار تعیین می کند که تصویر زمینه در صورتی که از اندازه پنجره کوچکتر بود هم به صورت افقی و هم عمودی تکرار شود تا همه زمینه پنجره را بپوشاند.
- **no-repeat**
این گزینه مشخص می کند که تصویر زمینه نباید تکرار شود حتی اگر از اندازه پنجره کوچکتر بود.
- **repeat-x**
این گزینه مشخص می کند که تصویر باید به صورت افقی تکرار شود نه عمودی.
- **repeat-y**
این گزینه مشخص می کند که تصویر زمینه باید به صورت عمودی تکرار شود.
-

background-attachment

از این ویژگی برای تعیین موقعیت تصویر زمینه در هنگام بالا و پایین بردن صفحه به وسیله اسکرول بار در پنجره مرورگر است. مقادیر زیر را می توان برای این ویژگی استفاده کرد:

- **scroll**
اگر این گزینه را انتخاب کنید تصویر زمینه به همراه محتویات صفحه به بالا و پایین می رود.
- **fixed**
با انتخاب این گزینه مشخص می کنید که تصویر زمینه در هنگام حرکت کردن محتویات صفحه باید ثابت بماند و حرکت نکند.
کد زیر را می توان برای مشخص کردن تصویر زمینه ثابت در بخش HEAD متن استفاده کرد. در این مثال لوگوی سایت ما به عنوان زمینه ثابت در نظر گرفته شده است. برای مشاهده یک نمونه از این تصویر زمینه [اینجا را کلیک کنید](#).

```
<style type="text/css">
<!--
body {
background-image:url(http://www.neopersia.org/images/neopersia_logo2.gif);
background-attachment:fixed;
background-repeat:no-repeat;
background-position:center center }

```

-->

</style>

این مثال را به یاد داشته باشید. دوباره از آن استفاده خواهیم کرد. در درس بعدی به سایر ویژگیهای مرتبط با تصویر زمینه می پردازیم.

background-position

این ویژگی جهت تعیین موقعیت تصویر زمینه در صفحه استفاده می شود. این ویژگی بیشتر زمانی مورد استفاده قرار می گیرد که برای تصویر زمینه از **no-repeat :background-repeat** استفاده شده باشد. این ویژگی می تواند دو مقدار را همزمان به خود اختصاص دهد. با این دو مقدار می توانیم موقعیت عمودی و افقی تصویر در صفحه را تعیین کنیم. توجه داشته باشید که این دو مقدار باید با یک فاصله خالی از هم جدا شوند نه کاما یا نقطه ویرگول.

دو مقداری که می توان به این ویژگی اختصاص داد از مقادیر زیر انتخاب می شوند:

- **top, center, or bottom**
- **left, center, or right**

برای این ویژگی باید از هر دسته از مقادیر بالا فقط یکی انتخاب شود. در اینجا می توانید برخی از موارد قابل استفاده را مشاهده کنید:

- **background-position:top right**
این ویژگی تصویر زمینه را در بالا و سمت راست صفحه نمایش می دهد.
- **background-position:center right**
این ویژگی تصویر را در سمت راست وسط صفحه نمایش می دهد.
- **background-position:bottom center**
این ویژگی تصویر زمینه را در پایین مرکز صفحه نمایش می دهد.

می توانیم به جای گزینه های بالا از مقادیری در مقیاس پیکسل یا درصد قرار دهیم تا تصویر زمینه را در هر فاصله ای از بالا و سمت چپ صفحه که بخواهیم قرار دهیم. در این روش فقط کفایت مقدار فاصله از سمت چپ صفحه را نوشته و مقیاس آن را مشخص کنیم و با یک فاصله مقدار فاصله از بالای صفحه همراه با مقیاس آنرا بنویسیم. به مثال زیر توجه کنید:

در این مثال از لوگوی سایت ما به عنوان زمینه صفحه استفاده شده است که در فاصله ۲۰ پیکسل از بالا و ۴۰ پیکسل از چپ صفحه قرار می گیرد. در ضمن از ویژگی **fixed :background-attachment** استفاده شده است تا در صورت حرکت دادن صفحه زمینه به صورت ثابت قرار بگیرد. کد زیر را به منظور تعیین تصویر زمینه صفحه در بخش HEAD صفحه قرار می دهیم:

```
<style type="text/css">
```

```
<!--
```

```
body {
```

```
background-image:url(http://www.neopersia.org/images/neopersia_logo2.gif);
```

```
background-attachment:fixed;
background-position:40px 20px }
-->
</style>
```

برای مشاهده نتیجه [اینجا را کلیک کنید](#).

با ترکیب کردن ویژگی‌هایی که توضیح داده شد می‌توانیم انواع تصویر زمینه مختلف و جالبی را به وجود بیاوریم. می‌توانید این کار را انجام دهید و تنوع آنرا ملاحظه کنید.

با استفاده از استایل این امکان وجود دارد که شکل نشانگر ماوس را وقتی روی قسمت مشخصی از صفحه قرار می‌گیرد تغییر دهیم. به این منظور می‌توانیم از ویژگی زیر در استایل مربوط به تگ مورد نظر استفاده کنیم:

cursor: نوع نشانگری که مورد نظر ماست

برای مثال فرض کنید می‌خواهیم وقتی نشانگر ماوس روی یک قسمت متن قرار می‌گیرد به شکل ساعت شنی در آید. می‌توانیم به صورت زیر عمل کنیم (ما برای مشخص شدن متن از رنگ قرمز در تگ span استفاده کرده ایم):

```
<span style="cursor: wait; color: red">روی این متن قرار دهید ماوس خود را</span>
```

نتیجه را می‌توانید اینجا مشاهده کنید:

ماوس خود را روی این متن قرار دهید.

اگر بخواهید وقتی ماوس کاربر روی یک قسمت مشخص از صفحه مثلاً جدولها قرار می‌گیرد شکل خاصی داشته باشد می‌توانید در بخش HEAD صفحه این ویژگی را برای جدول تعریف کنید تا در همه جدولهای صفحه به کار گرفته شود. از class و id هم می‌توانید استفاده کنید. روش ساده تر هم استفاده از شناسه style در تگهای HTML است.

در جدول زیر برخی از مقادیر مختلف این ویژگی که شکل نشانگر ماوس را عوض می‌کنند آورده شده است. می‌توانید ماوس خود را بر روی توضیحات این گزینه ها نگره دارید تا ببینید به چه شکلی در می‌آید.

مقدار قابل استفاده	توضیحات
auto	این گزینه نشانگر ماوس را با توجه به تنظیمات مرورگر کاربر تغییر می‌دهد.
crosshair	نشانگر ماوس به صورت به علاوه در می‌آید.
default	نشانگر ماوس به صورت همان فلش همیشگی در می‌آید.
move	نشانگر به شکلی در می‌آید که در هنگام جا به جا کردن چیزی به خود می‌گیرد.
pointer	نشانگر به شکل دست در می‌آید.
help	یک علامت سؤال کنار پیکان نشانگر ماوس قرار می‌گیرد.

به شکلی در می آید که بر روی متنها به خود می گیرد.	text
به شکل ساعت شنی در می آید.	wait
یک فلش به سمت شمال	n-resize
یک فلش به سمت جنوب	s-resize
یک فلش به سمت مشرق	e-resize
یک فلش به سمت مغرب	w-resize
یک فلش به سمت شمال شرقی	ne-resize
یک فلش به سمت شمال غربی	nw-resize
یک فلش به سمت جنوب شرقی	se-resize
یک فلش به سمت جنوب غربی	sw-resize

با استفاده از CSS می توانیم نحوه نمایش دادن لینکها را در صفحه تنظیم کنیم. به صورتی که استفاده از جاوا اسکریپت در صفحه تا حد زیادی کم می شود. از طریق استفاده از استایل می توانیم رنگ لینکها، اندازه فونت آنها، زیر خط لینکها، رنگ زمینه هر لینک و بسیاری دیگر از خواص لینکها را تغییر دهیم.

برای اینکه بتوانیم از همه خواصی که برای لینکها در CSS وجود دارد استفاده کنیم بهتر است تا استایل را در بخش HEAD متن خود یا در فایلها CSS خارجی که به صفحه پیوند می شوند استفاده کنیم. در هر دو صورت مراحل انجام کار شبیه هم است. ما در اینجا به استایل را در بخش HEAD متن تعریف می کنیم.

برای کار با لینکها در CSS می توانیم از مشخصات یک لینک را به چهار بخش تقسیم کنیم که در زیر می توانید آنها را مشاهده کنید:

• **a:link** و **a**

این مشخصه ها نحوه نمایش لینکهایی را که در صفحه وجود دارند و هنوز روی آنها کلیک نشده است تنظیم می کنند.

• **a:hover**

این مشخصه وضعیت لینک را وقتی که نشانگر ماوس روی آن قرار دارن تنظیم می کند.

• **a:active**

این قسمت مشخص کننده وضعیت لینکی است که بر روی آن کلیک می شود.

• **a:visited**

این مشخصه برای لینکهایی که در بازدیدهای قبلی مورد استفاده قرار گرفته اند استفاده می شود.

برای کار کردن با لینکها ما در قسمت HEAD متن از مشخصه های بالا استفاده می کنیم. یعنی کد زیر را در بخش HEAD متن وارد می کنیم اگر می خواهید از استایل خارجی استفاده کنید می توانید [نحوه فراخوانی استایل خارجی به صفحه](#) را مطالعه کنید.

```
<style type="text/css">
<!--
a:link { مشخصات مورد نظر }
a:hover { مشخصات مورد نظر }
a:active { مشخصات مورد نظر }
a:visited { مشخصات مورد نظر }
-->
</style>
```

برای هر ویژگی بالا می توانید مشخصات خاص خود را تعریف کنید برای مثال می تواند رنگ از مشخصه background-color برای تعیین رنگ زمینه لینک وقتی که ماوس روی آن قرار می گیرد استفاده کنید یا از color برای تعیین رنگ متن لینکها. به طور کلی می توانید از بیشتر مشخصه هایی که برای متن ها کارایی دارند در مورد لینکها هم استفاده کنید برای مثال font-family و font-size و border و ...

در اینجا یک مثال مطرح می کنیم که در آن لینکهای صفحه به رنگ سبز نمایش داده می شوند با اندازه فونت 12pt و بدون زیرخط، وقتی که ماوس روی آنها قرار می گیرد رنگ آنها قرمز می شود و رنگ زمینه متفاوتی پیدا می کنند همچنین یک کادر نیز اطراف آنها به وجود می آید. در موقع کلیک کردن رنگ لینک سفید می شود و لینکهایی که در دفعات قبلی کلیک شده اند با زیر خط نمایش داده می شوند. برای ساختن لینکهایی با مشخصاتی که بیان شد می توانیم از کد زیر در قسمت HEAD متن یا استایل خارجی استفاده کنیم:

```
<style type="text/css">
<!--
a:link { color:green; font-size:12pt; text-decoration:none }
a:hover { color:red; background-color:#6699CC; border:#006699 2px solid }
a:active { color: white }
a:visited { text-decoration: underline }
-->
</style>
```

در قسمت زیر می توانید لینکهایی را که برای آنها از کد بالا استفاده شده است مشاهده کنید:

این لینکها با استفاده از کد بالا تنظیم شده اند.

ماوس خود را روی آنها قرار دهید.

می توانید بر روی آنها کلیک کنید.

همانطور که می بینید کدهایی که در قسمت HEAD این صفحه قرار دارند بر روی لینکهای بالا تأثیر گذاشته اند اما لینکها در سایر نقاط صفحه ویژگیهای متفاوتی دارند. دلیل این تفاوت این است که ما لینکهای بالا را در یک DIV قرار دادیم و برای آن یک ID در بخش HEAD صفحه تعریف کردیم و ویژگیهای مربوط به لینکهای بالا در قالب ID مشخص کردیم. در اینجا می توانید کدی را که ما مورد استفاده قرار داده ایم ببینید:

```
#pre a:link,#pre a { color:green; font-size:10pt; text-decoration:none }
#pre a:hover { color:red; background-color:#6699CC; border:#006699 2px solid; text-
decoration:none }
#pre a:active { color: White }
#pre a:visited { text-decoration: underline; color:#003399 }
```

کد مربوط به DIV را نیز به صورت زیر نوشتیم:

```
<div align="center" id="pre">
لینکها کدهای مربوط به
</div>
```