

## آموزش کار با استایل (CSS)

CSS یا همان **Sheet Style Cascading** روشی است برای قالب بندی و طراحی اجزای صفحه از قبیل متن، تصویر، زمینه صفحه، کادرها و تقریباً هر چیزی که در طراحی صفحه استفاده می شود بدون استفاده از کدهای HTML. از طریق این استایل شیتها می توانیم از دوباره نویسی کدهای HTML که باعث مشکل شدن کد نویسی و همچنین کند شدن سرعت بارگذاری صفحه در اینترنت می شود جلوگیری کنیم. برای مثال می توانیم یک بار نوع فونت مورد استفاده در یک صفحه را تعیین کنیم و دیگر مجبور نباشیم کد مربوط به فونت را در صفحه به کار ببریم. همچنین می توانیم از این ابزار قدرتمند در مواردی به جای جاوا اسکریپت نیز استفاده کنیم و چون مشکلات جاوا اسکریپت مثلاً غیر فعال بودن جاوا اسکریپت در مرورگر را ندارد با خیالی آسوده تر می توانیم از آن استفاده کنیم.

CSS هم مانند جاوا اسکریپت این امکان را به ما می دهد که از یک فایل استایل خارجی برای تعیین خواص بخشهای مختلف صفحه های یک سایت استفاده کنیم. در این صورت می توانیم ظاهر همه صفحه ها را با ایجاد یک تغییر کوچک در فایل استایل خارجی تغییر دهیم. مثلاً فرض کنید که رنگ متن همچنین نوع فونت متن صفحه هایمان را در یک فایل استایل تعیین کرده ایم و از آن در طراحی چندین صفحه استفاده کرده ایم، حالا اگر بخواهیم رنگ متن همه این صفحات را تغییر دهیم تنها کافی است تا رنگ متن را در همان فایل استایل تغییر دهیم و نیازی به تغییر دادن صفحات نداریم.

در این قسمت به معرفی قابلیت های مختلف CSS خواهیم پرداخت. چون این قسمت در سایت ما به تازگی راه اندازی شده است هنوز کامل نشده و به مرور زمان سعی می کنیم آنرا کاملتر کنیم.

اگر انتقاد یا پیشنهادی برای بهتر شدن این قسمت دارید می توانید از طریق «فرم تماس با ما» ما را مطلع کنیم. ما نیز خوشحال می شویم از نظرات شما در جهت بهبود این بخش استفاده کنیم.

## استفاده از کلاس و id در طراحی صفحات با CSS

### • سلکتور کلاس

با استفاده از کلاس می توانیم برای یک تگ HTML استایلهای متفاوتی را تعریف کنیم. برای نمونه فرض کنیم می خواهیم 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 کادر های 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 است این یک سرفصل با اندازه</h3>  
<div id="border"> است این div 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.

## وارد کردن CSS در تگهای HTML با استفاده از شناسه Style

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.

### تعریف کردن استایل در بخش head صفحه HTML

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 <SPAN> 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 <SPAN> 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:

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

### استفاده از استایل خارجی برای طراحی صفحات HTML

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 یا 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 محتویاتی 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 محتویات 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">شد این متن به رنگ قرمز نمایش داده خواهد</p>
```

نتیجه به این صورت خواهد بود:

این متن به رنگ قرمز نمایش داده خواهد شد.

## تنظیم عرض، ارتفاع، و overflow با CSS

### 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 این پیکسل بیشتر شد در این جمله پس از اینکه عرض آن</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 DIV 100 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 پیکسل کمتر است ارتفاع این متن از 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>متن<br>100<br>پیکسل<br>کمتر<br>است.</div>

از

ارتفاع  
این  
متن  
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 محتوای 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  
۱۰۰  
پیکسل  
کمتر  
است.

از

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>قسمتهای بیشتر از overflow از است ولی با استفاده DIV طول این سطر بیشتر از عرض<nobr></div> شونده دیده نمی DIV عرض

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">  
شونده دیده نمی DIV قسمتهای بیشتر از عرض overflow از است ولی با DIV طول این سطر بیشتر از عرض  
</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.

نحوه ایجاد کادر برای عناصر صفحه با استفاده از استایل

## کادر (Boder)

## 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:

۱. **none**  
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** .٢

DIV have a solid border This

**dashed** .٣

DIV have a dashed border This

**dotted** .٤

DIV have a dotted border This

**double** .٥

DIV have a double border This

**groove** .٦

DIV have a groove border This

**hidden** .٧

This DIV have a hidden border

**inset** .٨

DIV have a inset border This

**outset** .٩

DIV have a outset border This

**ridge** .١٠

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 است برای</div>

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:

۱. مقدار عددی بر حسب پیکسل تعیین کنیم. به مثال زیر توجه کنید:

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

|     |    |      |    |     |       |
|-----|----|------|----|-----|-------|
| 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">این کادر thin ضخامت این کادر</div> است در نظر گرفته شده

|     |    |      |    |     |       |
|-----|----|------|----|-----|-------|
| 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">این کادر thick ضخامت این کادر</div> است در نظر گرفته شده

|       |    |      |     |       |      |
|-------|----|------|-----|-------|------|
| 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 هگزادسیمال رنگها:

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

این کادر به رنگ سبز خواهد بود.

تنظیم حاشیه و فاصله از لبه در عناصر صفحه

## حاشیه ها

## 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 فاصله دارد DIV این</div>

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 فاصله دارد DIV این</div>

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 دارد DIV این</div>

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 دارد صد DIV این</div>

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">تعریف شده است padding معادل ۱۰۰ پیکسل DIV این برای</div>

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">تعریف شده است padding معادل ۳۰ پیکسل DIV این برای</div>

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:

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

این هم نتیجه:

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

تغییر موقعیت عناصر صفحه با استفاده از CSS

## 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 اسکپ 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:

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

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 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 می شود در موقعیت تعیین شده در کد آن نمایش داده می این</div>

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 left و bottom 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 می شود در موقعیت تعیین شده در کد آن نمایش داده می این</div>

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

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

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

---

### ویژگیهای متن در CSS

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

بن 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:

|                                                                               |               |   |
|-------------------------------------------------------------------------------|---------------|---|
| •                                                                             | <b>left</b>   | : |
| با استفاده از این مقدار می توانیم محتویات یک تگ را در سمت چپ صفحه نمایش دهیم. |               |   |
|                                                                               |               |   |
|                                                                               | <b>right</b>  | : |
| این مقدار معین می کند که محتویات یک تگ باید در سمت راست صفحه نمایش داده شوند. |               |   |
|                                                                               |               |   |
| •                                                                             | <b>center</b> | : |
| 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:

```
<div style="text-align:left; border:thin dashed #00CCFF">محتویات این DIV</div>
```

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:

<div style="text-decoration: overline">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:

|                                                                                                                                                                                                                                                                                                                                                                     |   |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
| <ul style="list-style-type: none"><li>• <b>none</b></li></ul> <p>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.</p> | : |
|                                                                                                                                                                                                                                                                                   |   |
| <ul style="list-style-type: none"><li>• <b>underline</b></li></ul> <p>از این گزینه می توان برای زیر خط دار کردن متنها استفاده کرد.</p>                                                                                                                                                                                                                              | : |
|                                                                                                                                                                                                                                                                                  |   |
| <ul style="list-style-type: none"><li>• <b>overline</b></li></ul> <p>این گزینه برای نمایش خط بالای متن استفاده می شود.</p>                                                                                                                                                                                                                                          | : |
|                                                                                                                                                                                                                                                                                 |   |
| <ul style="list-style-type: none"><li>• <b>line-through</b></li></ul> <p>این گزینه یک خط را در میانه متن ایجاد می کند و آنرا به صورت خط خورده نشان می دهد.</p>                                                                                                                                                                                                      | : |
|                                                                                                                                                                                                                                                                                 |   |
| <ul style="list-style-type: none"><li>• <b>blink</b></li></ul> <p>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.</p>                                                                                                                          | : |

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

<a href="#example" style="text-decoration:none">مورد استفاده در لینک</a>

این هم لینک بدون زیر خط:

[متن مورد استفاده در لینک](#)

### ویژگیهای مربوط به متن در CSS

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 آورد. 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:

|                                                                                                                                          |                   |   |
|------------------------------------------------------------------------------------------------------------------------------------------|-------------------|---|
| •                                                                                                                                        | <b>none</b>       | : |
| با انتخاب این گزینه متن به همان صورتی که تایپ شده است نمایش داده می شود.                                                                 |                   |   |
| <div></div>                                                                                                                              |                   |   |
| •                                                                                                                                        | <b>lowercase</b>  | : |
| با انتخاب این گزینه متن با حروف کوچک نمایش داده می شود.                                                                                  |                   |   |
| <div></div>                                                                                                                              |                   |   |
| •                                                                                                                                        | <b>uppercase</b>  | : |
| این گزینه متن را با حروف بزرگ نمایش می دهد.                                                                                              |                   |   |
| <div></div>                                                                                                                              |                   |   |
| •                                                                                                                                        | <b>capitalize</b> | : |
| 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 propety



## 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 و سطر<br>20 بعدی بین این سطر و سطر</div>

نتیجه به صورت زیر خواهد بود:

بین این سطر و سطر بعدی

۳۰ 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 در صد 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>صورت عادی و در حالی که بقیه سطرها به  
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.

### تنظیم فونت صفحه با استفاده از استایل

در این قسمت به روش تنظیم فونت متن با استفاده از استایل می پردازیم. بسیاری از اعمالی را که قبلاً با تگ <font> 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 ۳۰</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 <FONT>, <B>, <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.

## نحو تغییر رنگ متن با استفاده از CSS

### color

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 <FONT> 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

## تغییر رنگ و تصویر زمینه عناصر HTML با CSS

### 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 دارد</span> <span style="background-color: yellow">متن بدون زمینه</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:

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

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>
```

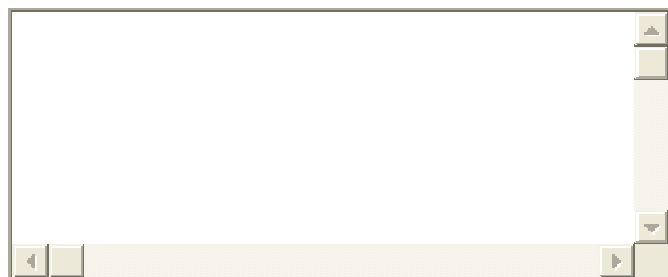
جدول به دست آمده مانند زیر است:

|                  |                  |
|------------------|------------------|
| 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:



### کار با تصویر زمینه در طراحی صفحات با استایل

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: <a href="#">url(url)</a>                                               | آدرس محل قرار گرفتن تصویر                                                                                                                                                                      | مشخص کننده محل قرار گرفتن تصویر در اینترنت است                                            |
| background-repeat: <a href="#">مقدار ویژگی</a>                                           | repeat, no-repeat, repeat-x, repeat-y                                                                                                                                                          | نوع تکرار تصویر را مشخص می کند                                                            |
| background-attachment: <a href="#">مقدار ویژگی</a>                                       | scroll, fixed                                                                                                                                                                                  | مشخص می کند که آیا تصویر زمینه با یا در اسکرول کردن صفحه حرکت می کند جای خود ثابت می ماند |
| background-position: <a href="#">مقدار</a> <a href="#">مقدار دوم</a> <a href="#">اول</a> | <a href="#">مقدار اول</a> :<br>top, center, bottom,<br>پیکسل، عددی در مقیاس<br>مقیاس درصد<br><a href="#">مقدار دوم</a> :<br>left, center, right,<br>عددی در مقیاس پیکسل، عددی در<br>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/cssBGtest.jpg);
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.

## تنظیمات موقعیت عکس زمینه عناصر صفحه با CSS

### 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 **background-repeat: no-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 مقداری 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 ۳۰ پیکسل از بالا و ۴۰ پیکسل از چپ صفحه قرار می گیرد. در ضمن از ویژگی **background-attachment: fixed** 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/cssBGtest.jpg);
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.

#### تغییر شکل نشانگر ماوس با استفاده از استایل

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:

نوع نشانگری که مورد نظر ماست: **cursor:**

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 id و class 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 نگه 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.                      |
| text               | به شکلی در می‌آید که بر روی متن‌ها به خود می‌گیرد.                      |
| wait               | به شکل ساعت شنی در می‌آید.                                              |
| n-resize           | یک فلش به سمت شمال                                                      |
| s-resize           | یک فلش به سمت جنوب                                                      |
| e-resize           | یک فلش به سمت مشرق                                                      |
| w-resize           | یک فلش به سمت مغرب                                                      |
| ne-resize          | یک فلش به سمت شمال شرقی                                                 |
| nw-resize          | یک فلش به سمت شمال غربی                                                 |
| se-resize          | یک فلش به سمت جنوب شرقی                                                 |
| sw-resize          | یک فلش به سمت جنوب غربی                                                 |

#### اعمال تغییر بر روی لینکها در صفحه با استفاده از استایل

با استفاده از CSS می‌توانیم نحوه نمایش دادن لینکها را در صفحه تنظیم کنیم. به صورتی که استفاده از جاوا اسکریپت در صفحه تا حد زیادی کم می‌شود. از طریق استفاده از استایل می‌توانیم 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 فایل‌های 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  
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 ۱۲pt 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>
```

#### نحوه استفاده از کدهای جاوا اسکریپت در صفحه

به طور کلی به سه طریق می توانیم از جاوا اسکریپت در یک صفحه استفاده کنیم:

۱. وارد کردن اسکریپت در بخش **<head>** صفحه
۲. وارد کردن جاوا اسکریپت در بخش **<body>** صفحه
۳. استفاده از فایل اسکریپت خارجی

به یاد داشته باشید که می توانید از همه روشهای بالا به طور همزمان برای وارد کردن جاوا اسکریپت در صفحه استفاده کنید.

در دو روش اول کدهای جاوا اسکریپت به طور مستقیم وارد صفحه HTML می شوند. برای وارد کردن جاوا اسکریپت در این دو روش از تگ **script** استفاده می شود. تمامی کدهای جاوا اسکریپتی که می خواهید به طور مستقیم وارد صفحه کنید باید در تگ **script** قرار داشته باشند وگرنه مرورگر به جای به کار بردن آنها را به صورت متن در صفحه نمایش می دهد. معمولاً کدهای جاوا اسکریپت با ساختار زیر وارد صفحه می شوند :

```
<script language="javascript" type="text/javascript">
دارید در این قسمت وارد می شوند دستورات جاوا اسکریپتی که قصد استفاده از آنها را
</script>
```

در تگ بالا برای مشخص کردن نوع موتور اسکریپتی که مرورگر باید برای تفسیر کردن کد استفاده کند از دو شناسه **language** و **type** استفاده می کنیم.

## وارد کردن اسکریپت در بخش <head>

در هنگام باز شدن یک صفحه وب عناصر موجود در بخش **<head>** قبل از سایر اجزای صفحه لود می شوند. معمولاً هنگامی کدهای جاوا اسکریپت در بخش **<head>** متن استفاده می شود که بخواهیم مطمئن شویم به هنگام نیاز کدهای ما لود شده اند.

در اینجا یک مثال از استفاده از جاوا اسکریپت در **<head>** را مشاهده می کنید:

```
<html dir="rtl">

<head>
<script language="javascript" type="text/javascript">
function message() {
این <br> صفحه قبل از سایر اجزای صفحه لود می شوند &lt;head&gt; کدهای جاوا اسکریپت موجود در<div align="center">
document.write('<div align="center">');
}
}
</script>
</head>

<body>
```

```
<p align="center"><a href="javascript:message();">کنید روی این متن کلیک</a></p>
</body>

</html>
```

[برای مشاهده صفحه ساخته شده با کد بالا اینجا را کلیک کنید.](#)

---

## وارد کردن جاوا اسکریپت در قسمت بدنه متن HTML

این امکان وجود دارد که کدهای جاوا اسکریپت را در قسمت `<body>` صفحه وارد کنیم. در این صورت لود شدن آنها تا هنگامی که آن بخش از صفحه لود شود به عقب می افتد و اگر صفحه به صورت کامل باز نشود ممکن است اسکریپت ما کار نکند. مثال زیر یک نمونه از وارد کردن جاوا اسکریپت در `<body>` صفحه HTML را نشان می دهد:

```
<html dir="rtl">

<head>
<title>صفحه جاوا اسکریپت در بدنه</title>
</head>

<body>
<script language="javascript" type="text/javascript">
document.write('<br><h1 align="center">است این متن توسط جاوا اسکریپت نوشته شده</h1>');
</script>
</body>

</html>
```

[برای مشاهده صفحه مربوط به کد بالا اینجا را کلیک کنید](#)

---

## فرا خوانی اسکریپت خارجی به صفحه

گاهی اوقات ممکن است بخواهید از یک سری از کدهای جاوا اسکریپت برای چندین صفحه استفاده کنید و نمی خواهید همه آنها را به صورت جداگانه در هر صفحه بنویسید. در این مواقع می توانید کدهای مورد نظر خود را در یک فایل متنی وارد کنید و آنرا با پسوند `.js` ذخیره کنید. در هر صفحه ای که بخواهید از این فایل استفاده کنید کافی است آنرا در آن صفحه فرا خوانی کنید. ( به یاد داشته باشید که در این فایلها خارجی نباید از تگهای `<script>` که مخصوص HTML هستند استفاده کنید )

برای فرا خوانی این فایلها به صفحه از همان تگ **script** که در موارد قبل استفاده می کردیم کمک می گیریم با این تفاوت که در اینجا به آن یک شناسه **src** که آدرس اسکریپت ما به عنوان مقدار آن قرار گرفته است اضافه می کنیم.

برای مثال کد جاوا اسکریپت زیر را در نظر بگیرید :

```
var text = '<br><h1 align="center" dir="rtl" >متن با استفاده از جاوا اسکریپت نوشته شده است این</h1>';

document.write(text);
```

ما این کد را در فایلی با نام `external.js` در همان پوشه ای که صفحه ما قرار دارد ذخیره کرده ایم. حالا با وارد کردن کد زیر در صفحه می توانیم فایل بالا را فراخوانی کنیم:

```
<script language="javascript" type="text/javascript" src="external.js"></script>
```

[برای مشاهده صفحه ای که فایل ذخیره شده ما را فرا خوانی می کند اینجا را کلیک کنید.](#)

## معرفی عملگرهای مورد استفاده در JavaScript

در جاوا اسکریپت برای جمع، تفریق، مقایسه و بسیاری از عملیات دیگر از عملگرها استفاده می کنیم. عملگرها به چند دسته تقسیم می شوند : عملگرهای محاسباتی، مقایسه ای، جایگزینی، منطقی، رشته ای، و شرطی. در ادامه به ارائه توضیحاتی برای هر نوع از این عملگرها خواهیم پرداخت.

## عملگرهای محاسباتی

از این نوع عملگرها برای انجام عملیات ریاضی نظیر جمع و ضرب استفاده می شود. می توانید این عملگرها را در اینجا مشاهده کنید ( در مثال زیر متغیر x را برابر ۸ و متغیر y را برابر ۴ در نظر بگیرید ) :

| نام       | عملگر | مثال     | نتیجه |
|-----------|-------|----------|-------|
| جمع       | +     | $x + y$  | 12    |
| تفریق     | -     | $x - y$  | 4     |
| ضرب       | *     | $x * y$  | 32    |
| تقسیم     | /     | $x / y$  | 2     |
| باقیمانده | %     | $9 \% y$ | 1     |
| افزایش    | ++    | $x++$    | 9     |
| کاهش      | --    | $x--$    | 7     |

از عملگرهای افزایش و کاهش به ترتیب برای افزودن ۱ به مقدار قبلی و کم کردن ۱ از مقدار قبلی استفاده می شود.

## عملگرهای جایگزینی

این عملگرها ترکیبی از عملگرهای دیگر هستند و به همین دلیل آنها را عملگرهای ترکیبی هم می نامند. می توانید این عملگرها را در جدول زیر مشاهده کنید ( در مثال زیر متغیر x را برابر ۸ و متغیر y را برابر ۴ در نظر بگیرید ) :

| نام              | عملگر | مثال       | عبارت معادل  |
|------------------|-------|------------|--------------|
| انتساب           | =     | $x = 8$    | $x = 8$      |
| انتساب جمع       | +=    | $x += y$   | $x = x + y$  |
| انتساب تفریق     | -=    | $x -= y$   | $x = x - y$  |
| انتساب ضرب       | *=    | $x *= y$   | $x = x * y$  |
| انتساب تقسیم     | /=    | $x /= y$   | $x = x / y$  |
| انتساب باقیمانده | %=    | $x \% = y$ | $x = x \% y$ |

## عملگرهای مقایسه ای

از این عملگرها برای مقایسه دو مقدار استفاده می شود که شامل عملگرهای زیر هستند.

در جدول زیر مقادیر  $x$  و  $y$  و  $z$  را مطابق زیر در نظر بگیرید و توجه داشته باشید که مقدار متغیرهای  $x$  و  $y$  عددی و مقدار متغیر  $z$  از نوع رشته ای است :

$x = 4;$   
 $y = 8;$   
 $z = "4";$

| نام             | عملگر              | مثال                   | نتیجه |
|-----------------|--------------------|------------------------|-------|
| تساوی           | <code>==</code>    | <code>x == z</code>    | True  |
| همانی           | <code>===</code>   | <code>x === z</code>   | False |
| نامساوی         | <code>!=</code>    | <code>x != y</code>    | True  |
| بزرگتر از       | <code>&gt;</code>  | <code>x &gt; y</code>  | False |
| کوچکتر از       | <code>&lt;</code>  | <code>x &lt; y</code>  | True  |
| بزرگتر یا مساوی | <code>&gt;=</code> | <code>x &gt;= z</code> | True  |
| کوچکتر یا مساوی | <code>&lt;=</code> | <code>x &lt;= y</code> | True  |

تفاوت عملگر تساوی با عملگر همانی در این است که در عملگر تساوی نوع متغیرها در نظر گرفته نمی شود و اگر مقدار متغیرها با هم برابر باشد حتی اگر از یک نوع هم نباشند نتیجه True خواهد بود. ولی در عملگر همانی باید متغیرها مقداری مساوی باشند و از یک نوع هم باشند تا نتیجه True باشد.

## عملگرهای منطقی

این عملگرها برای انجام عملیات منطقی بر روی دو عبارت به کار می روند ( در مثال زیر متغیر  $x$  را برابر ۸ و متغیر  $y$  را برابر ۴ در نظر بگیرید ) :

| نام  | عملگر | مثال                                 | نتیجه |
|------|-------|--------------------------------------|-------|
| نقیض | !     | <code>!(x==y)</code>                 | True  |
|      |       | <code>!(y==3)</code>                 | False |
| و    | &&    | <code>(x=4 &amp;&amp; y=8)</code>    | True  |
|      |       | <code>(x&gt;4 &amp;&amp; y=8)</code> | False |
| یا   |       | <code>(x&gt;4    y=8)</code>         | True  |
|      |       | <code>(x&gt;4    y&gt;8)</code>      | False |

## عملگر رشته ای

از این عملگر برای چسباندن دو رشته به هم استفاده می شود. مثلاً برای اتصال دو متغیر از نوع رشته از این عملگر استفاده می شود.

| نام            | عملگر | مثال                                                  |
|----------------|-------|-------------------------------------------------------|
| عملوند رشته ای | +     | a = 'طراحی وب';<br>b = 'با جاوا اسکریپت';<br>c = a+b; |

در مثال بالا مقدار متغیر c این است : 'طراحی وب با جاوا اسکریپت'  
برای ایجا فاصله بین کلمه وب و کلمه با می توانیم یک فاصله خالی به یکی از متغیرهای a یا b اضافه کنیم یا با استفاده از عملوند رشته ای یک فضای خالی به متغیر c اضافه کنیم :

```
a = 'طراحی وب';
b = 'با جاوا اسکریپت';
c = a+b;
```

```
a = 'طراحی وب';
b = 'اسکریپت با جاوا';
c = a+' '+b;
```

در نتیجه کد بالا مقدار متغیر c به این صورت خواهد بود : 'طراحی وب با جاوا اسکریپت'

## عملگر شرطی

در جاوا اسکریپت نوعی عملگر وجود دارد که بسته به شرایط مقداری را به یک متغیر تخصیص می دهد به این عملگر عملگر شرطی می گویند. ساختار این عملگر به صورت زیر است :

```
VAR_NAME=(شرط)?value1:value2
```

این عملگر در صورتی که شرط برقرار باشد مقدار اول را به متغیر می دهد و در صورت برقرار نبودن شرط مقدار دوم را به متغیر تخصیص می دهد. به مثال زیر توجه کنید :

```
age_group=(age>18)?'بزرگسال':'زیر سن بلوغ'
```

در این مثال اگر متغیر age بزرگتر از ۱۸ باشد مقدار 'بزرگسال' به متغیر age\_group تخصیص داده می شود و در صورتی که متغیر age کوچکتر از ۱۸ باشد مقدار 'زیر سن بلوغ' به عنوان مقدار متغیر age\_group قرار می گیرد.

### نمایش پیام هشدار در صفحه با استفاده از جاوا اسکریپت

آیا مایلید یکی از آن کادر هایی که ناگهان ظاهر می شوند و معلوم نیست از کجا آمده اند را به صفحه خود اضافه کنید؟ بسیار خوب بیایید با کادر هشدار شروع کنیم که به بیننده چیزهایی را که شما می خواهید می گوید.

این فرمان هشدار است:

```
alert('متنی که می خواهید به بازدید کننده نشان داده شود')
```

حالا برای استفاده از آن ما باید آنرا در یک فرمان دیگر قرار دهیم. بیایید آنرا در یک فرمان آشنا مثل فرمان onMouseover استفاده کنیم. اسکریپت مقابل باعث ظاهر شدن یک کادر هشدار وقتی که ما نشانگر ماوس خود را روی لینک به وجود آمده قرار می دهیم می شود. مثلاً در لینکی که نمی خواهیم بیننده کلیک کند:

```
<a href="alert.php" onmouseover="alert('Insert your message here')">
ماوس خود را روی این لینک قرار دهید
</a>
```

آنرا امتحان کنید. فقط ماوس خود را روی آن قرار دهید:

[ماوس خود را روی این لینک قرار دهید](#)

در اینجا به ارائه توضیحاتی در مورد دستورات جاوا اسکریپت استفاده شده در کد بالا می پردازیم :

**onmouseover** :

این رویداد مسئولیت اجرای کد جاوا اسکریپت موجود در خود را وقتی که ماوس روی آن قرار می گیرد بر عهده دارد.

**alert** :

این تابع متنی را که به عنوان مقدار آن قرار گیرد در یک کادر محاوره ای با یک دکمه OK نمایش می دهد.

بسیار خوب، یک روش دیگر هم وجود دارد که بیشتر باعث گیج شده بیننده می شود. آن هم کادر هشدار به محض باز شدن پنجره است. شما با این روش می توانید مطالبی را که می خواهید به بیننده خود بگویید قبل از باز شدن صفحه به او نشان دهید.

برای ساختن این نوع کادر هشدار فرمان هشدار را بین <head> و </head> قرار دهید. برای این کار باید از تگهای <script> که در اول فصل با آن آشنا شدیم استفاده کنیم تا کدهای جاوا اسکریپت بین بقیه کدها تشخیص داده شوند:

```
<head>
<script language="javascript">
<!-- hide from old browsers
alert('به این صفحه خوش آمدید');
//-->
</script>
</head>
```

فرمان بالا یک کادر محاوره ای را وقتی که بیننده صفحه شما را باز می کند نشان می دهد. وقتی که OK را کلیک کنید بارگذاری صفحه ادامه پیدا می کند.

آیا می خواهید آنرا امتحان کنید؟ لینک زیر را کلیک کنید و به صفحه آزمایشی که ما برای این فرمان ساخته ایم بروید:

[پیام هشدار به محض باز شدن صفحه!](#)

آیا می خواهید چند گام جلو تر برویم؟ چندین فرمان هشدار را در تگ SCRIPT قرار دهیم تا چندین کادر هشدار پشت سر هم به محض باز شدن صفحه نمایش داده شوند و البته برای هر پیام باید دکمه OK را کلیک کنیم تا به پیام بعدی برویم. خودتان این را امتحان کنید :

[چندین پیام هشدار به محض باز شدن صفحه!](#)

این هم کدی که مسئول این پیامها است:

```
<head>
<title>javascript example 2</title>
<script language="javascript">
<!--
alert('لطفاً سایت ما را به منوی favorite اضافه کنید خود');
alert('از منوی favorite گزینه add to favorite را انتخاب کنید');
alert('میانبر هم انجام دهید می توانید این کار را با کلیدهای');
alert('کلید فقط کلید کنترل را نگهدارید و d را فشار دهید');
//-->
</script>
</head>
```

شما می توانید برای ظاهر شدن کادر هشدار از دکمه ها هم استفاده کنید. فقط کد را در جایی که می خواهید دکمه نشان داده شود قرار دهید. در این حالت شما نیازی به تگهای <script> هم ندارید. این هم یک نمونه از این نوع پیام هشدار:

این هم کد این اسکریپت:

```
<input type="button" value="اینجا را کلیک کنید اگر این صفحه را مطالعه کرده اید"
onclick="alert('باز هم سری به ما بزنید از اینکه به سایت ما مراجعه کردید متشکریم')">
```

ما در این دکمه از رویداد **onclick** استفاده کردیم. کار این رویداد هم مانند **onmouseover** است با این تفاوت که کد مربوطه را در هنگام کلیک کردن روی آن اجرا می کند.

#### نوشتن متن با استفاده از جاوا اسکریپت

در جاوا اسکریپت این امکان وجود دارد که متنی را نوشته و در صفحه نمایش بدهیم. متن مورد نظر را می توانیم قالب بندی کنیم و از تگهای HTML نیز در آن استفاده کنیم. برای این کار از فرمان **document.write()** استفاده می کنیم که یک فرمان جاوا اسکریپت مخصوص چاپ کردن خروجی در صفحه است. روش کار به صورت زیر است :

#### نوشتن متنهای ساده

```
<script language="JavaScript" type="text/javascript">
document.write('بنویسید متن مورد نظر خود را اینجا');
</script>
```

البته نقطه ویرگولی که در آخر فرمان نوشته شده تنها در صورتی الزامی است که بیش از یک فرمان در اسکریپت خود داشته باشیم.

برای نوشتن متن به این صورت کافی است تا فرمان بالا را در جایی از صفحه HTML که می خواهید متن به نمایش در آید بنویسید. در مثال زیر متنی با استفاده از جاوا اسکریپت در صفحه نمایش داده می شود، برای مشاهده مثال لینک زیر را کلیک کنید :

[نوشتن متنهای ساده با استفاده از جاوا اسکریپت](#)

#### نوشتن متنهای قالب بندی شده

با استفاده از فرمان **document.write** امکان نوشتن متنهای قالب بندی شده با استفاده از تگهای HTML هم وجود دارد. برای اینکار فقط کافیه متن HTML مورد نظر خود را در تابع **document.write()** قرار دهید. برای مثال به کد زیر توجه کنید :

```
<script language="JavaScript" type="text/javascript">
document.write('<h1>جاوا اسکریپت نوشته شده است این متن با استفاده از</h1>');
</script>
```

برای مشاهده نتیجه این مثال لینک زیر را کلیک کنید :

[نوشتن متنهای قالب بندی شده با HTML توسط جاوا اسکریپت](#)

#### تابع document.write() با مقدار چند سطری

اگر بخواهیم متنی که با استفاده از جاوا اسکریپت متنی را بنویسیم که در صفحه در چند سطر نمایش داده شود به راحتی می توانیم با وارد کردن **<br>** در متن مورد نظر یا با سایر تگهای HTML این کار را انجام دهیم. اما اگر بخواهیم مقداری را که به تابع **document.write()** می دهیم در چند سطر وارد کنیم باید مقدار مورد نظر را در هر سطر از هم جدا کرده و با استفاده از عملوند رشته ای + آنها را در چند سطر به هم پیوند بزنیم.

**نکته :** نباید بدون استفاده از عملگرهای مربوطه مقدار تابع بالا را در بیش از یک سطر بنویسید. در صورتی که متن مورد نظر خود را به این گونه در چند سطر بنویسید تابع کار نمی کند و احتمالاً با پیغام خطا مواجه می شوید.

اگر کد خود را به صورت زیر بنویسید با خطا مواجه خواهید شد :

```
<script language="JavaScript" type="text/javascript">
document.write('First part of text
```

```
Second part of text');  
</script>
```

برای اینکه با خطا مواجه نشوید و کد شما کار خود را انجام دهد باید آنرا به صورت زیر بنویسید :

```
<script language="JavaScript" type="text/javascript">  
document.write('First part of text'  
+  
'Second part of text');  
</script>
```

[برای آشنایی بیشتر با عملگرهای جاوا اسکریپت اینجا را کلیک کنید.](#)

### معرفی دستور شرطی if...else و کاربردهای مختلف آن

از دستورات شرطی برای انجام دستوراتی به شرط برقراری یک رابطه دیگر استفاده می شود. در طراحی صفحات وب بسیار پیش می آید که می خواهیم در شرایط خاصی صفحه یک رفتار مشخص داشته باشد و در موارد دیگر رفتار صفحه تفاوت داشته باشد. در این مواقع می توانیم از دستور شرطی **if** استفاده کنیم. با این دستور مشخص می کنیم اگر شرط خاصی برقرار بود یک عمل خاص انجام شود. ساختار این دستور به شکل زیر است :

### دستور شرطی if

```
if (condition)  
{  
    در اینجا قرار می گیرد دستوری که در صورت درستی شرط اجرا می شود  
}
```

به یک مثال برای دستور **if** توجه کنید :

```
var test = confirm('ای می شود اگر این پیام را تأیید کنید دستور شرطی اجرا');  
  
if (test == true) {  
    alert('و این پیام به نمایش در آمد شما دکمه مورد نظر را کلیک کردید');  
}
```

برای اینکه کد بالا را آزمایش کنید لینک زیر را کلی کنید و در پیامی که نمایش داده می شود دکمه OK را کلیک کنید. اگر دکمه Cancel را کلیک کنید شرط بالا برقرار نمی شود و بدون اجرا شدن دستور شرطی صفحه مثال باز می شود.

[برای مشاهده مثال بالا در عمل اینجا را کلیک کنید.](#)

### دستور شرطی if...else

برخی مواقع ممکن است بخواهیم دستور شرطی یک شرط را چک کند تا در صورت برقرار بودن آن شرط یک فرمان خاص را اجرا کند و در صورتی که شرط برقرار نبود یک فرمان دیگر را به اجرا در آورد. در این موارد می توانیم از دستور **if** به همراه **else** استفاده کنیم. ساختار این دستور به شکل زیر است :

```
if (condition) {  
    .شوند دستوراتی که در صورت برقراری شرط اجرا می  
}  
else {  
    .دستوراتی که در صورت عدم برقراری شرط اجرا می شوند  
}
```

در کد بالا ابتدا شرط بررسی می شود و در صورت برقراری آن دستورات مربوطه اجرا می شوند ولی در صورتی که شرط برقرار نباشد دستورات مربوط به قسمت **else** اجرا می شود.

```
var yourmark = prompt('یک شماره از ۰ تا ۲۰ وارد کنید');
```

```
if (yourmark > 10) {  
    alert('است شماره ای که وارد کرده ای از ۱۰ بزرگتر');  
}  
else {  
    alert('!نیست عبارتی که شما وارد کرده اید از ۱۰ بزرگتر');  
}
```

برای مشاهده نتیجه مثال بالا لینک زیر را کلیک کنید و در کادری که نمایش داده می شود یک عدد از ۰ تا ۲۰ وارد کنید و دکمه OK را کلیک کنید.

[مثال استفاده از دستور if...else](#)

## استفاده از دستور if...else if...else (دستورات شرطی تو در تو)

ممکن است در نظر داشته باشیم تا در صورت برقراری یک شرط یک دستور خاص اجرا شود و در صورتی که شرط برقرار نبود شرط دیگری بررسی شود و دستور مربوط به آن اجرا شود و در صورتی که هیچ یک از شرطهای قبل برقرار نبود دستور دیگری اجرا شود. در این مواقع می توانیم از دستور if...else در دستور قبلی استفاده کنیم. به صورت زیر :

```
if (condition 1) {  
    شوند دستوراتی که در صورت درستی شرط اول اجرا می  
}  
else if (condition 2) {  
    می شوند دستوراتی که در صورت درستی شرط دوم اجرا  
}  
else {  
    می شوند دستوراتی که زمانی که هیچکدام از شرطها صحیح نباشد اجرا  
}
```

در این روش محدودیتی از نظر تعداد شرطها وجود ندارد و می توانیم به تعداد نامحدود از دستور else if استفاده کنیم ولی در مواردی که تعداد این شرطها خیلی زیاد می شود بهتر است به جای این روش از دستور switch که بعداً توضیح داده خواهد شد استفاده کنیم.

در اینجا مثال قبل را با کمی تغییر برای این روش استفاده می کنیم :

```
var yourmark = prompt('یک شماره به دلخواه وارد کنید');
```

```
if (yourmark > 0 && yourmark < 100) {  
    alert('وارد شده بین ۰ تا ۱۰۰ است شماره');  
}  
else if (yourmark > 100) {  
    alert('شماره وارد شده از ۱۰۰ بزرگتر است');  
}  
else {  
    alert('وارد شده یا ۰ است یا به جای عدد از حروف استفاده کرده اید عبارت');  
}
```

[برای مشاهده نتیجه مثال دستور if...else if...else اینجا را کلیک کنید.](#)

دستور شرطی switch در جاوا اسکریپت

با استفاده از دستور if امکان این وجود دارد که برنامه را حداکثر به یکی از دو جهت ممکن هدایت کنیم. یعنی اگر شرط مورد نظر برقرار بود اسکریپت یک سری کد را اجرا می کند و در صورت عدم برقراری شرط یک سری کد دیگر اجرا می شود. اگر بخواهیم برای اسکریپت خود مسیرهای بیشتری در نظر بگیریم یک راه موجود استفاده از دستورات if تو در تو است. این روش ممکن است در موارد جزئی کارایی داشته باشد اما اگر تعداد شرطها زیاد شود ما را مجبور می کند از تعداد زیادی if و else در اسکریپت خود استفاده کنیم که این مسئله هم باعث طولانی شدن برنامه و هم باعث گیج کننده شدن آن می شود از طرفی امکان اشتباه را هم بالا می برد.

راه حل دیگر استفاده از دستور شرطی **switch** است. این دستور را می توان در هنگامی که می خواهیم یک متغیر را با چندین مقدار مقایسه کنیم و مقدار متناسب با آنرا پیدا کنیم کار برد دارد. ساختار این دستور به صورت زیر است :

**switch(x)**

{

**مقدار اول case:**

با مقدار اول اجرا می شودXاین کد در صورت برابری متغیر

**break**

**مقدار دوم case:**

با مقدار دوم اجرا می شودXاین کد در صورت برابری متغیر

**break**

**default:**

با هیچکدام از مقادیر Xاین کد در صورتی اجرا می شود که مقدار متغیر

ذکر شده برابر نباشد

}

در این دستور مقدار متغیر معرفی شده در قسمت **switch** با مقادیر آورده شده در هر **case** مقایسه می شود. در صورت مساوی بودن مقادیر کد همان **case** اجرا می شود و با استفاده از دستور **break** آورده شده در هر قسمت برنامه پس از اجرای کد آن قسمت به خارج از دستور **switch** هدایت می شود. در غیر این صورت برنامه به سراغ **case** بعدی می رود و در صورتی که مقدار متغیر مورد نظر با هیچکدام از **case** ها برابر نبود کد قسمت **default** اجرا خواهد شد.

قابل ذکر است که استفاده از **default** اختیاری است و شما بنا به نیاز خود می توانید از آن استفاده کرده یا آنرا حذف کنید. در دستور **switch** محدودیتی در تعداد **case** ها وجود ندارد و می توانید به هر تعدادی که نیاز داری از این مورد استفاده کنید.

در این قسمت به ذکر یک نمونه برای دستور **switch** می پردازیم :

به کدی که در اینجا آورده شده است توجه کنید. پس از اجرای این کد کادری باز می شود که از شما میخواهد عددی را وارد کنید. با وارد کردن عدد در این کادر و تأیید آن پیامی متناسب با روز معادل آن عدد در هفته ظاهر می شود. البته ما این کار را فقط برای اعداد ۱ تا ۳ انجام دادیم، شما در صورت تمایل می توانید آنرا تکمیل کنید :

**var x=prompt('لطفاً یک عدد بین ۱ تا ۷ وارد کنید');**

**switch (x) {**

**case '1':**

**alert('شماره وارد شده مربوط به روز « شنبه » است');**

**break**

**case '2':**

**alert('شماره وارد شده مربوط به روز « یکشنبه » است');**

**break**

**case '3':**

```

alert('شماره وارد شده مربوط به روز « دوشنبه » است');
break
case '4':
alert('شماره وارد شده مربوط به روز « سه شنبه » است');
break
case '5':
alert('شماره وارد شده مربوط به روز « چهارشنبه » است');
break
case '6':
alert('شماره وارد شده مربوط به روز « پنجشنبه » است');
break
case '7':
alert('شماره وارد شده مربوط به روز « جمعه » است');
break
default:
alert('روز مربوط به شماره وارد شده در این برنامه وجود ندارد');
}

```

[برای مشاهده مثال بالا در عمل اینجا را کلیک کنید.](#)

#### رویداد onmouseover

وقت اون رسیده که نوشتن اولین اسکریپت خود را شروع کنید. این یکی راحت‌تر چون مجبور نیستیم درگیر اضافه کردن تگهای SCRIPT شویم. این اسکریپت کوچک به دلخواه شما وقتی نشانگر ماوس روی یک لینک قرار می‌گیرد متنی را در نوار وضعیت مرورگر نشان می‌دهد. به مثال نگاه کنید:

```

<a href="JavaScript_2.htm"
onmouseover="window.status='« آموزش جاوا اسکریپت »';return true">
!!!دهید ماوس خود را روی این لینک قرار
</a>

```

شما هم امتحان کنید و ببینید چگونه کار می‌کند. نشانگر ماوس خود را روی لینک زیر قرار دهید ولی کلیک نکنید (اگر کلیک کنید، چون در کد بالا آدرس لینک، آدرس همین صفحه است پس از کلیک کردن دوباره همین صفحه باز می‌شود). حالا پایین پنجره مرورگر خود به نوار وضعیت نگاه کنید باید این جمله را ببینید: « آموزش جاوا اسکریپت »

ماوس خود را روی این لینک قرار دهید!!!

بسیار خوب، این چیزی است که با فرمان **onmouseover** اتفاق می‌افتد:

۱. **onmouseover="**

این شکل فرمان **onmouseover** است. مرورگر انتظار فرمان دیگری بین دو علامت دابل کوت ( " ) را دارد.

۲. **» 'window.status='**

این فرمان به مرورگر می‌گوید که جمله ای را در نوار وضعیت بنویسد. شما می‌توانید جمله بین دو علامت ( ' ) را با آنچه که می‌خواهید در نوار وضعیت به نمایش در آید جایگزین کنید.

**return**

**true** .۳

عبارت را تأیید کرده و مرورگر آن را نمایش می دهد.

دلیل استفاده از علامت سینگل کوت ( ' ) در فرمان window.status این است که window.status خود قسمتی از فرمان onMouseover است که از ( " ) استفاده کرده است. اگر دوباره از علامت دابل کوت ( " ) استفاده کرده بودیم ممکن بود مرورگر در مورد اینکه چه کاری باید انجام دهد گیج می شد. چون ممکن بود علامت دابل کوت دوم را به منزله پایان فرمان onMouseover تلقی کند و فرمان جدیدی را با window.status شروع کند: onMouseover=""window.statuse

مشکلی که ممکن است در اینجا به وجود آید پاک نشدن متنی است که می خواستید در نوار وضعیت نمایش داده شود. دو راه برای از بین بردن این مشکل وجود دارد. یکی استفاده از فرمان **onMouseout** است، و دیگری فرا خواندن تابعی که متن را پس از مدتی پاک می کند. راه دوم نیازمند استفاده از توابع و اسکرپت تگها است. در اینجا آسانترین راه را به شما نشان می دهیم: استفاده از فرمان onMouseout:

```
<a href="JavaScript_2.htm"
onmouseover="window.status='متن خود را اینجا قرار دهید'; return true"
onmouseout="window.status=''; return true">
قرار دهید لینک خود را اینجا
</a>
```

این فرمان متن نوار وضعیت را وقتی که نشانگر ماوس روی آن نیست ناپدید می کند. مانند زیر:

[لینک خود را اینجا قرار دهید](#)

در حقیقت ما همان کار قبل را تکرار کردیم ولی به جای متن از فضای خالی در فرمان **window.status** استفاده کردیم.

این اسکرپت جالب و مفید است و می تواند به بازدید کنندگان صفحه شما کمک کند تا بهتر بدانند لینک آنها را به چه صفحه ای می برد. و می توانند توضیحات شما را در مورد لینک ببینند.

#### دکمه ها و موارد استفاده آنها

ما برای نوشتن اسکرپت با استفاده از دکمه ها، ابتدا باید بدانیم چگونه می توان یک دکمه را در یک صفحه قرار داد. برای این کار شما باید از تگهای <FORM> در اطراف تگ دکمه استفاده کنید. در مثال زیر نمونه یک HTML که یک دکمه را در یک صفحه قرار می دهد آورده شده است:

```
<FORM>
<INPUT type="button" value="اینجا کلیک کنید" name="button1">
</FORM>
```

این فرمان یک دکمه را در صفحه شما قرار می دهد، اما اگر روی آن کلیک کنید اتفاقی نمی افتد...

در زیر کار هر قسمت از فرمان بالا را می بینیم:

۱. <FORM>

این قسمت یک فرم را ایجاد می کند بنابر این ما می توانیم یک دکمه بسازیم.

۲. <INPUT>

این تگ به ما اجازه می دهد نوعی ناحیه ورودی بسازیم.

۳. type="button"

این فرمان ناحیه ورودی ما را به عنوان یک دکمه تعریف می کند.

۴. value="اینجا کلیک کنید"

این متنی است که افراد روی دکمه می بینند. شما می توانید هر چیزی را که می خواهید بینندگان روی دکمه ببینند اینجا بنویسید.

۵. name="button1"

شما می توانید برای مراجعات بعدی یا احتمالاً استفاده در یک اسکرپت به دکمه خود یک نام بدهید.

احتمالاً شما نمی خواهید یک دکمه بسازید که کاری انجام نمی دهد. در زیر یک فرمان جاوا اسکرپت آورده شده است که مرورگر را مجبور می کند وقتی دکمه را کلیک کردید عکس العمل نشان دهد:

```
onClick="javascript command"
```

فقط این فرمان را در تگ INPUT قرار دهید که دکمه را با آن ساختید، مانند زیر:

```
<INPUT type="button" value="اینجا کلیک کنید" name="button1" onClick="" >
```

برای مثال ما می توانیم مانند فرمان onMouseover در این دکمه نیز استفاده کنیم و متنی را در نوار وضعیت پنجره چاپ کنیم!

```
<FORM>
<INPUT type="button" value="اینجا کلیک کنید" name="Button2"
onClick="window.status='متن خود را اینجا بنویسید'; return true">
</FORM>
```

حالا دکمه را فشار دهید و به متن نوار وضعیت نگاه کنید:

شما می توانید با استفاده از دکمه ها به بینندگان خود اجازه دهید رنگ زمینه صفحه را برای خودشان تغییر دهند. فقط از فرمان زیر به جای فرمان window.status استفاده کنید:

```
document.bgColor='رنگ مورد نظر به زبان انگلیسی نوشته شود'
```

فرمان زیر را به عنوان تمرین فرمان onClick وارد کنید:

```
<FORM>
<INPUT type="button" value="Change Background Color" name="button3" onClick="document.bgColor='yellow'">
</FORM>
```

حالا شما دکمه ای مانند زیر دارید که با کلیک کردن بر روی آن زمینه صفحه به رنگ زرد در می آید!

شما می توانید به هر تعداد که می خواهید از این دکمه ها اضافه کنید، فقط مطمئن شوید که آنها گزینه بازگشت به حالت اولیه را نیز داشته باشند. اسکریپت زیر به شما چهار گزینه می دهد : سفید، زرد، قرمز و آبی . می توانید از دکمه «بازگشت به رنگ اولیه!» برای بازگرداندن تصویر زمینه این صفحه استفاده کنید. کدی که در این صفحه از آن استفاده شده است به دلیل وجود تصویر زمینه کمی با کد زیر متفاوت است ولی کارایی آن تفاوتی ندارد.

```
<FORM>
<INPUT type="button" value="به سفید تغییر رنگ زمینه"
name="button6" onClick="document.bgColor='white'"><br>
<INPUT type="button" value="تغییر رنگ زمینه به زرد"
name="button3" onClick="document.bgColor='yellow'"><br>
<INPUT type="button" value="تغییر رنگ زمینه به قرمز"
name="button4" onClick="document.bgColor='red'"><br>
<INPUT type="button" value="تغییر رنگ زمینه به آبی"
name="button5" onClick="document.bgColor='#66CCFF'">
</FORM>
```

برای دکمه چهارم به جای نام رنگ از معادل هگزادسیمال رنگها استفاده شده است. برای یافتن معادل هگزادسیمال رنگها و توضیح درباره آنها می توانید به [این صفحه](#) مراجعه کنید.

حالا شما سه دکمه دارید، آنها را امتحان کنید!

در اسکریپت آخری که در این قسمت بررسی می کنیم، اجازه می دهیم تا یک دکمه به عنوان یک لینک عمل کند. به این منظور فقط فرمان زیر را برای فرمان onClick استفاده کنید:

```
window.location='URL'
```

برای مثال این هم اسکریپتی که شما را به صفحه آموزش جاوا اسکریپت ما می برد:

```
<FORM>
<INPUT type="button" value="اسکریپت آموزش جاوا" name="button6"
onClick="window.location='.//'">
</FORM>
```

این هم دکمه، آن را امتحان کنید تا به سرفصل آموزش جاوا اسکریپت بروید:

### آموزش کار با تاریخچه مرورگر با استفاده از جاوا اسکریپت

مرورگر ها به صورت پیش فرض آدرس و حتی خود صفحاتی را که کاربر مشاهده می کند ذخیره می کنند. این آدرسها به ترتیبی که کاربر آنها را مشاهده کرده است در **تاریخچه (history)** مرورگر ثبت می شوند و دکمه های Back و Forward پنجره مرورگر بر اساس این تاریخچه کار می کنند. با استفاده از جاوا اسکریپت می توانیم به این تاریخچه دسترسی داشته باشیم و با کمک آن مسیر کاربران را هدایت کنیم.

ما برای کار با تاریخچه مرورگر در اینجا از دکمه های HTML استفاده می کنیم که با کمک رویداد onclick در هنگام کلیک کردن روی این دکمه ها دستورات مورد نظر ما اجرا خواهند شد. در اینجا می توانید کد HTML مربوط به ساخت این دکمه ها را مشاهده کنید (البته مقدار مربوط به شناسه onclick مربوط به دستورات جاوا اسکریپت است که بعداً وارد خواهد شد):

```
<input type="button" value="متن روی دکمه" onclick="">
```

برای آشنایی بیشتر با رویداد های HTML می توانید از صفحه [رویدادهای مربوط به صفحه کلید و ماوس در HTML](#) استفاده کنید. (لینک در صفحه جدید باز خواهد شد)

دستورات مختلفی برای کار با تاریخچه مرورگر وجود دارد که همه آنها از شیء **history** استفاده می کنند. در ادامه می توانید تعدادی از این دستورات را که با بیشتر مرورگرها کار می کنند مشاهده کنید

## history.back()

از این فرمان می توانید جهت هدایت کاربران به صفحه ای که قبل از صفحه کنونی در پنجره مرورگر مشاهده کرده اند استفاده کنید. این دستور دقیقاً مانند دکمه **Back** در پنجره مرورگر عمل می کند و در صورتی که صفحه کنونی اولین صفحه ای باشد که از زمان باز شدن پنجره نمایش داده شده باشد این دستور کاربردی ندارد. برای مشاهده یک نمونه از کاربرد این دستور می توانید لینک را کلیک کنید :

[برای مشاهده کاربرد دستور history.back\(\) اینجا را کلیک کنید](#)

در اینجا می توانید کد مورد استفاده در مثال بالا را مشاهده کنید :

```
<input type="button" value="کلیک کنید برای بازگشت به صفحه قبل اینجا را" onclick="history.back()">
```

## history.forward()

در صورتی که بازدید کننده قبلاً از دکمه Back در مرورگر خود استفاده کرده باشد یا با استفاده از دستور history.back() به صفحه قبلی باز گشته باشد می توانیم از دستور **history.forward()** برای هدایت بازدید کننده به صفحه ای که پس از صفحه کنونی باز کرده است استفاده کنیم. این دستور کاربردی مشابه دکمه **Forward** در پنجره مرورگر دارد و کاربرد آن عکس کاربرد history.back() است.

دکمه زیر با استفاده از دستور **history.forward()** شما را به صفحه ای که برای مثال مربوط به بازگشت به صفحه قبلی مشاهده کرده اید هدایت می کند :

می توانید کد استفاده شده در ساخت دکمه بالا را در اینجا مشاهده کنید :

```
<input type="button" value="نمایش صفحه بعدی" onclick="history.forward()">
```

## history.go()

با استفاده از دستورهای `history.back()` و `history.forward()` فقط امکان این وجود دارد که کاربران را یک صفحه به عقب یا جلو هدایت کنیم. اگر بخواهیم بازدید کننده را به چندین صفحه عقب یا جلو هدایت کنیم باید از دستور `history.go()` استفاده کنیم.

### ۱- استفاده از مقداری مثبت برای `history.go()`

اگر برای دستور `history.go()` از مقداری مثبت استفاده شود کاربر به همان تعداد صفحه در تاریخچه مرورگر به جلو هدایت می شود. به کد زیر توجه کنید :

```
<input type="button" value="متن روی دکمه" onclick="history.go(3)">
```

کد بالا دکمه ای را می سازد که کلیک کردن بر روی آن همان کاری را انجام می دهد که سه بار فشار دادن دکمه Forward در پنجره مرورگر انجام می دهد.

در صورتی که دستور به صورت `history.go(۱)` نوشته شود مانند دستور `history.forward()` عمل می کند.

### ۲- استفاده از مقدار صفر برای `history.go()`

در صورتی که این دستور به صورت `history.go(۰)` نوشته شود باعث باز بارگذاری مجدد همان صفحه خواهد شد. برای مثال به کد زیر و دکمه ساخته شده با آن توجه کنید :

```
<input type="button" value="بارگذاری مجدد صفحه" onclick="history.go(0)">
```

این هم دکمه ساخته شده :

### ۳- استفاده از مقادیر منفی برای `history.go()`

اگر بخواهیم کاربر را چند صفحه در تاریخچه مرورگر به عقب هدایت کنیم می توانیم برای دستور `history.go()` از مقادیر منفی استفاده کنیم. برای مثال به کد زیر توجه کنید :

```
<input type="button" value="متن روی دکمه" onclick="history.go(-3)">
```

این کد دکمه ای را می سازد که وقتی کاربر آنرا کلیک کند به سومین صفحه ای که قبل از صفحه کنونی در پنجره مرورگر بازدید کرده است هدایت می شود. کاربرد این دکمه در اینجا مشابه سه بار فشار دادن دکمه Back در پنجره مرورگر است.

## history.length

با استفاده از دستور `history.length` می توانید تعداد صفحاتی را که در تاریخچه مرورگر وجود دارد پیدا کنید. یعنی با استفاده از این دستور می توانیم به تعداد صفحاتی که از زمان باز شدن پنجره مرورگر در آن نمایش داده شده است پی ببریم. برای مثال می توانید با کلیک کردن بر روی لینک زیر تعداد صفحات ذخیره شده در تاریخچه پنجره مرورگر خود را پیدا کنید :

[نمایش تعداد صفحات موجود در تاریخچه مرورگر](#)

در اینجا می توانید کد مورد استفاده در ساخت لینک بالا را مشاهده کنید :

```
<a href="javascript:alert(history.length)">
صفحات موجود در تاریخچه مرورگر نمایش تعداد
</a>
```

#### درخواست اطلاعات از کاربر

بیا فرض کنیم می خواهیم نام فردی را قبل از ورود به صفحه بهرسم و سپس نام آنها را در صفحه برای خودشان نمایش دهیم. این کار را با استفاده از جاوااسکریپت می توان انجام داد. فرمانی که برای این کار استفاده می شود به این صورت است:

```
prompt('سؤال شما', '');
```

این فرمان یک پنجره را قبل از باز شدن صفحه نمایش می دهد که سؤال شما در آن پرسیده می شود و بیننده می تواند جواب خود را در کادری که برای او در نظر گرفته شده بنویسد. دو سینگل کوت بعدی جواب پیش فرض شما برای سؤال است. اگر شما چیزی بین این دو سینگل کوت ننویسد، بیننده با یک لیست خالی، آماده برای نوشتن مواجه می شود. این پیغام قبل از بارگذاری صفحه نمایش داده می شود پس شما می توانید پاسخ بیننده را در صفحه مثال نام شما پرسیده می شود و پس از وارد کرده آن، نام شما در صفحه جدید نمایش داده می شود:

[لینک صفحه آزمایشی](#)

حالا به سراغ اسکریپتی که این کار را انجام می دهد می رویم. توجه داشته باشید که دستورات prompt و if/else در بخش HEAD صفحه وارد می شوند ولی نام کاربر در بخش BODY صفحه مشاهده می شود.

```
<html dir="rtl" >

<head>
<script language="javascript">
<!--
var yourname=prompt('لطفاً نام خود را وارد کنید',' ');
if ( (yourname=="") || (yourname==null) )
{
yourname="ناشناس";
}
//-->
</script>
</head>

<body>
<script language="javascript">
<!--
document.write("<center><h1>سلام "+yourname+" به سایت ما خوش آمدید "</h1></center>");
//-->
</script>
</body>

</html>
```

اولین چیزی که اتفاق می افتد این است که متغیر yourname مقداری را که کاربر برای آن وارد می کند به خود می گیرد. پس مقدار متغیر yourname معادل استرینگ است که نام کاربر را می سازد.

دستور if/else نام « ناشناس » را در صورتی که کاربر نامی را وارد نکند به عنوان مقدار متغیر yourname قرار می دهد. این دستور چک می کند که اگر نامی وارد نشود (null) و یا اگر فضای خالی وارد شود، مقدار « ناشناس » را به متغیر yourname می دهد.

در بخش body متن دوباره از تگ script استفاده می کنیم تا کدهای جاوا اسکریپت از بقیه متن html متمایز شود. در این قسمت ما از یک فرمان جدید یعنی document.write(" "); استفاده کرده ایم. این همان چیزی است که به متغیر yourname اجازه نوشته شدن در صفحه html را می دهد. ما در این پیغام دو استرینگ را به علاوه متغیر yourname در صفحه می نویسیم. متغیر yourname در این فرمان بین دو دابل کوت قرار نمی گیرد چون یک مقدار است نه یک استرینگ، (البته مقدار متغیر yourname یک استرینگ است). به دلیل اینکه متغیر yourname یک استرینگ نیست ما از دو علامت بعلاوه در ابتدا و انتهای آن استفاده می کنیم. این سری نوشته ها مرورگر را وادار می کند ابتدا یک استرینگ سپس مقدار متغیر و

سپس استرینگ دوم را بنویسد. چون ما در حال نوشتن یک کد جاوا اسکریپت هستیم برای اینکه بتوانیم متنی را در صفحه بنویسیم نمی توانیم مانند نوشتن متن در html عمل کنیم، و باید متن مورد نظر را بین دابل کوت و در قالب استرینگ در صفحه بنویسیم.

شاید متوجه شده باشید تگهای پایانی h1 و center به گونه ای متفاوت با قبل نوشته شده اند ( </h1> ). این تفاوت بدین دلیل است که اسلش در جاوا اسکریپت به معنای تقسیم است و برای جلوگیری از اشتباه شدن آن با علامت تقسیم از یک بک اسلش قبل از آن استفاده می کنیم. در نتیجه استفاده از یک بک اسلش ( \ ) و بلافاصله بعد از آن یک اسلش ( / )