

کتاب آموزش HTML

نت پارادیس

www.netparadis.com

فهرست

10	آموزش مقدمات HTML
10	پیش نیازها
11	مروری بر HTML
11	Html چیست؟
11	داکیومنت پایه ی HTML
12	برچسب های HTML
13	ساختار داکیومنت HTML
13	برچسب <!DOCTYPE>
14	برچسب های پایه ای در HTML
14	برچسب های تیتل
15	برچسب پاراگراف
15	برچسب شکست لینک
16	مرکزگذاری متن
17	خطوط افقی
18	حفظ طراحی
18	فضاهای غیرشکست
19	آموزش عناصر در HTML
20	برچسب HTML در مقابل عنصر
20	عناصر تو در توی HTML
21	ویژگی های زبان HTML
22	ویژگی های اصلی
23	ویژگی title
23	ویژگی class
24	ویژگی style
24	درونی کردن ویژگی ها
24	ویژگی dir
26	ویژگی xml:lang
26	ویژگی های عمومی
27	قالب بندی صفحات HTML
27	بولد کردن متن

28	ایتالیک کردن متن
28	آندرلاین کردن متن
29	خط کشیدن روی متن
29	فونت monospaced
30	متن superscript (چاپ شده در بالا)
30	متن subscript (چاپ شده در زیر)
31	متن مندرج
31	متن حذف شده
31	متن بزرگتر
32	متن کوچکتر
32	گروه بندی محتوا
34	آموزش HTML در Meta Tag
34	افزودن برجسب های متا به داکيومنت ها
35	مشخص کردن کلمات کلیدی
36	توصیف داکيومنت
36	بازبینی تاریخ داکيومنت
37	تازه سازی داکيومنت
37	Redirect کردن صفحه
38	تنظیم cookies
38	تنظیم نام نگارنده
39	تعیین تنظیم کاراکتر
40	آموزش استفاده از comment در html
40	کامنت های معتبر در مقابل کامنت های نامعتبر
41	کامنت های چندخطی
42	کامنت های شرطی
43	استفاده از برجسب کامنت
43	کد اسکریپت کامنت
44	کامنت صفحات طراحی
44	وارد کردن تصاویر در صفحات HTML
46	تنظیم موقعیت تصویر
47	تنظیم طول و عرض تصویر

48	تنظیم حاشیه ی تصویر
49	تنظیم همترازی تصویر
49	آموزش Table در HTML
50	تیتزر جدول
50	ویژگی های cellpadding و cellspacing
51	ویژگی های colspan و rowspan
52	پس زمینه ی جدول
53	طول و عرض جدول
54	تیتزر، بدنه و پاورقی جدول
55	جدول های تودرتو
56	آموزش List
56	لیست های بدون ترتیب HTML
57	ویژگی type
58	لیست های منظم HTML
60	ویژگی start
61	لیست های تعریف HTML
61	لینک های متن HTML
62	لینک کردن داکيومنت ها
62	ویژگی target
64	استفاده از مسیر پایه
65	لینک شدن به بخشی از صفحه
65	تنظیم رنگ های لینک
66	HTML tahlildadeh
66	دانلود کردن لینک ها
66	Download PDF File
66	آموزش Image Link
67	تصاویر حساس به ماوس
69	سیستم مختصات
70	لینک ایمیل HTML
70	برچسب HTML Email
71	تنظیمات پیش فرض

71	آموزش HTML Layout
72	HTML Layout – استفاده از جدول ها
73	Layout چند ستونی – استفاده از جدول ها
74	Layout های HTML – استفاده از DIV و Span
75	آموزش Iframes
76	ویژگی های برچسب <iframe>
78	background در html
78	زمینه ی HTML با رنگها
79	زمینه ی HTML با تصاویر
80	زمینه های طرح دار و شفاف
81	رنگهای HTML
81	روش های کدگذاری رنگ HTML
82	رنگ های HTML – نام رنگ
82	16 رنگ استاندارد W3C
83	رنگ های HTML – کدهای Hex
84	رنگ های HTML – مقادیر RGB
88	فونت ها
89	تنظیم اندازه ی فونت
90	تنظیم ظاهر فونت
90	تعیین ظاهر فونت جایگزین
91	تنظیم رنگ فونت
91	عنصر <basefont>
92	فرم های html
92	ویژگی های فرم
93	کنترل های فرم HTML
94	کنترل های ورودی متن
94	کنترل های ورودی تک خطی
96	کنترل های پسورد ورودی
97	کنترل های متن ورودی چند خطی
98	کنترل چک باکس
99	کنترل دکمه ی رادیو

100	کنترل Select Box
102	فایل آپلود باکس
103	کنترل های دکمه
104	کنترل های مخفی شده ی فرم
104	آموزش چند رسانه ای در HTML
105	ویژگی های برچسب <embed>
106	انواع ویدیوهای پشتیبانی شده
107	صدای زمینه
109	آموزش HTML Marquee
112	سربرگ HTML
112	برچسب <title> در HTML
113	برچسب <meta> در HTML
114	برچسب <base> در HTML
114	برچسب <link> در HTML
115	برچسب <style> در HTML
116	برچسب <script> در HTML
116	آموزش Cascading style sheet
118	Style sheet خارجی
119	Style sheet داخلی
120	Inline Style Sheet
121	آموزش javascript
121	جاوا اسکریپت خارجی
122	جاوا اسکریپت درونی
123	گرداندگان رویدادها (EventHandler())
123	مخفی کردن اسکریپت ها از مرورگرهای قدیمی
124	عنصر <noscript>

آموزش مقدمات HTML

HTML مخفف عبارت **Hyper Text Markup Language** می باشد که در نایع گسترده ترین زبان استفاده شده در صفحات وب می باشد.

HTML توسط برنزی لی در اواخر سال **1991** ایجاد شد، اما **html 2.0** اولین نوع استاندارد آن بود که در سال **1995** منتشر شد. **HTML 4.01** ورژن مهمی از **HTML** بود که در اواخر **1999** منتشر شد. گرچه **HTML 4.01** ورژنی است که به طور گسترده استفاده می شود، اما معمولاً **HTML 5** را داریم که توسعه ی یافته **html 4.01** است و در سال **2012** منتشر شد.

این آموزش برای آشنا کردن طراحان و توسعه دهندگان وب با نیاز برای درک **HTML** با جزئیات کافی به همراه یک نظر کلی ساده و مثال های عملی، طراحی شده است. این آموزش به شما محتوای کافی برای شروع با **HTML** از جایی که می توانید تخصصی با سطح بالاتر داشته باشید، ارائه می دهد.

پیش نیازها

قبل از پیش رفتن با این آموزش شما باید یک دانش پایه برای کار با ویندوز و سیستم عملیاتی **Linux** داشته باشید. علاوه بر این شما باید با موارد زیر آشنا باشید

تجربه کار با هر ویرایشگری مانند **Notepad ++**، **Notepad** و یا **Editplus** و غیره.

چگونگی ایجاد مسیرها و فایل ها روی کامپیوتر خود.

چگونگی مسیریابی از طریق مسیرهای مختلف.

چگونگی تایپ کردن محتوا در یک فایل و ذخیره ی آنها روی کامپیوتر.

درک تصاویر با فرمت های مختلف مانند **JPEG** و **PNG**.

امتحان کردن **HTML** روی خط

مثال

```
<!DOCTYPE html>
<html>
<body>
  <h1>Hello World!</h1>
```

```
</body>  
</html>
```

مروری بر HTML

Html چیست؟

HTML مخفف **Hypertext Markup Language** می باشد که گسترده ترین زبان مورد استفاده برای نوشتن صفحات وب است.

HTML اشاره به روشی دارد که در آن صفحات وب (داکیومنت های **HTML**) به هم لینک می شوند. بنابراین لینک موجود روی یک صفحه ی وب **Hypertext** نامیده می شود. همانطور که از اسم آن پیداست **HTML** یک **Markup language** (زبان نشانه گذاری) می باشد، که بدین معناست که شما می توانید از **HTML** برای نشانه گذاری متن داکیومنت استفاده کنید با برچسب هایی که به مرورگر چگونگی ساخت آن را برای نمایش بیان می کنند. در واقع **HTML** به نیت تعریف ساختار داکیومنت هایی مانند تیترها، پاراگراف ها، لیست ها و غیره توسعه پیدا کرد تا اشتراک گذاری اطلاعات علمی بین محققان را آسان تر کند. اکنون **HTML** به طور گسترده ای مورد استفاده قرار می گیرد تا صفحات وب را به کمک برچسب های موجود در زبان **HTML** تنظیم کند.

داکیومنت پایه ی HTML

به ساده ترین شکل خود، در زیر مثالی را از یک داکیومنت **HTML** مشاهده می کنید

```
<!DOCTYPE html>  
<html>  
<head>  
<title>This is document title</title>  
</head>  
<body>  
<h1>This is a heading</h1>  
<p>Document content goes here.....</p>  
</body>  
</html>
```

برای چک کردن نتیجه این کد **HTML** هم می توانید با استفاده از ویرایشگر مورد علاقه ی خود، آن را در یک فایل **HTML** و با نام **test.htm** ذخیره کنید. در انتها آن را با استفاده از یک مرورگر مانند اینترنت اکسپلورر یا گوگل کروم یا فایرفاکس و غیره باز کنید.

برچسب های HTML

همانطور که قبلا گفته شد HTML یک زبان نشانه گذاری است و برای طراحی محتوا از برچسب های مختلفی

استفاده می کند. این برچسب ها در داخل علامت $\langle \rangle$ قرار گرفته اند، به این شکل $\langle \text{نام برچسب} \rangle$.

به جز تعداد کمی از برچسب ها، بسیاری از آنها برچسب متناظر بستن را نیز دارند. به عنوان مثال $\langle \text{html} \rangle$

دارای برچسب بستن $\langle \text{/html} \rangle$ و $\langle \text{body} \rangle$ دارای برچسب بستن $\langle \text{/body} \rangle$ می باشند.

مثال بالا از داکيومنت HTML از برچسب های زیر استفاده می کند

Tag	Description
$\langle \text{!DOCTYPE...} \rangle$	این برچسب نوع داکيومنت و ورژن HTML را تعریف می کند.
$\langle \text{html} \rangle$	این برچسب داکيومنت کامل HTML را احاطه می کند و اساسا شامل تیتیر داکيومنت می شود که توسط $\langle \text{head} \rangle \dots \langle \text{/head} \rangle$ نمایش داده می شود و بدنه ی داکيومنت نیز به وسیله ی برچسب های $\langle \text{body} \rangle \dots \langle \text{/body} \rangle$ نمایش داده می شوند.
$\langle \text{head} \rangle$	این برچسب تیتیر داکيومنتی را نشان می دهد که می تواند دیگر برچسب های HTML از مانند $\langle \text{link} \rangle$, $\langle \text{title} \rangle$ را در خود نگه دارد.
$\langle \text{title} \rangle$	برچسب $\langle \text{title} \rangle$ در داخل برچسب $\langle \text{head} \rangle$ استفاده می شود تا تیتیر داکيومنت را ذکر کند.
$\langle \text{body} \rangle$	این برچسب بدنه ی داکيومنتی را نشان می دهد که دیگر برچسب های HTML مانند $\langle \text{p} \rangle$, $\langle \text{div} \rangle$, $\langle \text{h1} \rangle$ را در خود دارد.
$\langle \text{h1} \rangle$	این برچسب تیتیر را نمایش می دهد.

<code><p></code>	این برچسب یک پاراگراف را نمایش می دهد.
------------------------	--

برای یادگیری **HTML** لازم است برچسب های مختلفی را مطالعه کنید و درک کنید که در هنگام طراحی یک داکيومنت متنی چگونه رفتار می کنند. یادگیری **HTML** ساده است چرا که یوزرها باید مورد استفاده ی برچسب های مختلف را برای طراحی متن و یا تصاویر و ایجاد یک صفحه وب زیباتر یاد بگیرند.

وب جهانی **W3C Consortium** پیشنهاد می دهد که در شروع با **HTML** برچسب ها را با حروف کوچک به کار ببرید.

ساختار داکيومنت HTML

یک نمونه از داکيومنت **HTML** دارای ساختار زیر خواهد بود

Document declaration tag

```
<html>  
<head>  
  Document header related tags  
</head>  
<body>  
  Document body related tags  
</body>  
</html>
```

تمام برچسب های مربوط به بدنه م تیترا را در بخش های بعدی مطالعه خواهیم کرد، اکنون اجازه بدهید در مورد برچسب اطلاعیه ی داکيومنت بیشتر بدانیم.

برچسب <!DOCTYPE>

برچسب اطلاعیه ی **<!DOCTYPE>** توسط مرورگرهای وب برای پی بردن به ورژن **HTML** استفاده شده در داکيومنت مورد استفاده قرار می گیرد. ورژن حال حاضر **HTML** ورژن 5 می باشد که از اطلاعیه ی زیر استفاده می کند

```
<!DOCTYPE html>
```

انواع دیگری اطلاعات وجود دارند که می توانند در داکيومنت HTML، متناسب با ورژن HTML مورد استفاده قرار بگیرند. جزئیات بیشتری در این مورد در زمان بحث در مورد برچسب های دیگر HTML فرا خواهیم گرفت.

برچسب های پایه ای در HTML

برچسب های تیتر

هر داکيومنت با یک تیتر آغاز می شود. شما می توانید از اندازه های مختلف برای تیترهای خود استفاده کنید. HTML دارای شش سطح می باشد که از `<h5>`, `<h4>`, `<h3>`, `<h2>`, `<h1>` و `<h6>` عناصر استفاده می کند. در هنگام نمایش هر تیتر مرورگر یک خط قبل و یک خط بعد از تیتر اضافه می کند.

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Heading Example </title>
</head>
<body>
  <h1>This is heading 1 </h1>
  <h2>This is heading 2 </h2>
  <h3>This is heading 3 </h3>
  <h4>This is heading 4 </h4>
  <h5>This is heading 5 </h5>
  <h6>This is heading 6 </h6>
</body>
</html>
```

این مثال نتیجه ی زیر را به دنبال دارد

This is heading 1

This is heading 2

This is heading 3

This is heading 4

This is heading 5

This is heading 6

برچسب پاراگراف

برچسب `<p>` به روش طراحی متن شما در پاراگراف های مختلف اشاره دارد. هر پاراگراف متن باید بین برچسب بازکننده ی `<p>` و بسته ی `</p>` قرار بگیرد، همانطور که در مثال زیر مشاهده می کنید

```
<!DOCTYPE html>
<html>
<head>
  <title>Paragraph Example </title>
</head>
<body>
  <p>Here is a first paragraph of text. </p>
  <p>Here is a second paragraph of text. </p>
  <p>Here is a third paragraph of text. </p>
</body>
</html>
```

برچسب شکست لینک

هرزمان که شما از `
` عنصر استفاده کنید، هر چیزی که آن را دنبال می کند از خط بعد شروع خواهد شد. این برچسب نمونه ای از یک عنصر `empty` می باشد، زمانی که لازم نیست برچسبی را باز کنید یا ببندید چرا که چیزی برای رفتن بین آنها وجود ندارد.

در برچسب `
`، یک فضای خالی بین کاکترهای `br` و اسلش جلوی آن وجود دارد. اگر شما این فضا را حذف کنید، مرورگرهای قدیمی تر در اجرای خط شکست مشکل خواهند داشت، در حالیکه اگر اسلش را حذف کنید برچسب `
` باقیمانده در HTML معتبر نمی باشد.

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Line Break Example </title>
</head>
<body>
  <p>
    Hello <br />
    You delivered your assignment ontime. <br />
    Thanks <br />
    Mahnaz
  </p>
</body>
</html>
```

مثال بالا نتیجه ی زیر را تولید خواهد کرد

```
Hello
You delivered your assignment ontime.
Thanks
Mahnaz
```

مرکزگذاری متن

می توانید با استفاده از برچسب `<center>` می توانید هر محتوایی را در مرکز صفحه یا در مرکز هر سلول از یک جدول قرار دهید.

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Centring Content Example </title>
</head>
<body>
  <p>This text is not in the center. </p>
  <center>
    <p>This text is in the center. </p>
  </center>
</body>
</html>
```

مثال بالا نتیجه ی زیر را تولید خواهد کرد

This text is not in the center.

This text is in the center.

خطوط افقی

خطوط افقی برای بخش های شکست بصری یک داکيومنت استفاده می شوند. برچسب `<hr>` خطی از موقعیت کنونی داکيومنت به حاشیه ی سمت راست ایجاد کرده و خط را طبق آن می شکنند.

به عنوان مثال ممکن است تمایل داشته باشید بین دو پاراگراف خطی قرار دهید، همانطور که در مثال زیر ارائه شده است

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Horizontal Line Example </title>
</head>
<body>
  <p>This is paragraph one and should be on top </p>
  <hr />
  <p>This is paragraph two and should be at bottom </p>
</body>
</html>
```

این مثال نتیجه ی زیر را تولید خواهد کرد

This is paragraph one and should be on top

This is paragraph two and should be at bottom

و باز برچسب `<hr />` مثالی از **Empty** می باشد که نیازی به باز کردن یا بستن برچسب ندارید زیرا چیزی برای رفتن بین آنها وجود ندارد.

در عنصر `<hr />` یک فضای خالی بین کاراکترهای `hr` و اسلش مقابل آن وجود دارد. اگر این فضا را حذف کنید مرورگرهای قدیمی تر در اجرای خط افقی مشکل خواهند داشت. در حالیکه اگر اسلش مقابل آن را حذف کنید عنصر باقیمانده `<hr>` می باشد که در `HTML` فاقد اعتبار می باشد.

حفظ طراحی

گاهی اوقات تمایل دارید که متن فرمت دقیق خود در `HTML` را دنبال کند، در این موارد می توانید از برچسب پریفرمت `<pre>` استفاده کنید.

هر متن بین برچسب باز کننده `<pre>` و برچسب بستن `</pre>` طراحی متن منبع را حفظ خواهد کرد.

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Preserve Formatting Example </title>
</head>
<body>
<pre>
function testFunction( strText ){
  alert (strText)
}
</pre>
</body>
/html>
```

مثال بالا نتیجه ی زیر را تولید خواهد کرد

```
function testFunction( strText ){
  alert (strText)
}
```

سعی کنید از همان کد بدون نگهداری آن بین برچسب های `<pre>...</pre>` استفاده کنید.

فضاهای غیرشکست

فرض کنید می خواهید از عبارت "`12 Angry Men`" استفاده کنید. در اینجا از مرورگر نمی خواهید عبارت را بین دو خط به صورت `12 Angry` و `Men` بشکنند.

An example of [this](#) technique appears in the movie "`12 Angry Men`."

در مواردی که نمی خواهید مرورگر متن را بشکند باید به جای یک فضای عادی از فضای غیر شکست ** ** استفاده کنید. برای مثال وقتی "12 Angry Men" را در یک پاراگراف کدگذاری می کنید باید از کدی مانند زیر استفاده کنید

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Nonbreaking Spaces Example</title>
</head>
<body>
  <p>An example of this technique appears in the movie "12&nbsp;Angry&nbsp;Men."</p>
</body>
</html>
```

آموزش عناصر در HTML

یک عنصر HTML توسط یک برچسب شروع کننده تعریف می شود. اگر عنصر دارای محتوای دیگری باشد، با یک برچسب بسته کننده تمام می شود در حالیکه جلوی نام عنصر یک علامت اسلش قرار گرفته است که می توانید در جدول زیر برخی از این برچسب ها را مشاهده کنید

Start Tag	Content	End Tag
<p>	این برچسب محتوای پاراگراف می باشد.	</p>
<h1>	این برچسب تیتر محتوا می باشد.	</h1>
<div>	این برچسب تقسیم محتوا می باشد.	</div>

بنابراین در اینجا `<p>....</p>` یک عنصر HTML و `<h1>...</h1>` عنصر دیگری از HTML می باشد. عناصری از HTML وجود دارند که نیازی به بسته شدن ندارند مانند `<img.../>`, `<hr />` و `
`. این عناصر با عنوان **Void elements** (عناصر خالی) شناخته می شوند.

داکیومنت های HTML دارای درختی از این عناصر می باشند و مشخص می کنند که چگونه داکیومنت ها باید ساخته شوند و چه نوع محتوایی باید در چه بخشی از داکیومنت HTML قرار بگیرد.

برچسب HTML در مقابل عنصر

یک عنصر HTML به وسیله ی یک برچسب شروع کننده تعریف می شود. اگر عنصر دارای محتوای دیگری باشد با یک برچسب بستن تمام می شود.

برای مثال `<p>` برچسب شروع کننده ی یک پاراگراف می باشد و `</p>` برچسب بستن همان پاراگراف می باشد، اما `<p>This is paragraph</p>` عنصر یک پاراگراف می باشد.

عناصر تو در توی HTML

نگهداری یک عنصر HTML در داخل عنصر دیگر بسیار متداول می باشد.

```
<!DOCTYPE html>
<html>
<head>
  <title>Nested Elements Example </title>
</head>
<body>
  <h1>This is <i>italic</i> heading</h1>
  <p>This is <u>underlined</u> paragraph </p>
</body>
</html>
```

مثال بالا نتیجه ی زیر را دنبال خواهد کرد

This is *italic* heading

This is underlined paragraph

ویژگی های زبان HTML

برخی از برچسب های HTML مانند برچسب های تیترو برچسب های پاراگراف، و موارد استفاده ی آنها را مشاهده کردیم. تاکنون از آنها به ساده ترین شکل خود استفاده کرده ایم، اما بیشتر برچسب های HTML می توانند ویژگی هایی داشته باشند که مقداری اطلاعات اضافه می باشد.

یک **attribute** برای تعریف ویژگی های عنصر HTML استفاده می شود و در داخل برچسب بازکننده ی عنصر قرار می گیرد. همه ی ویژگی ها از دو بخش تشکیل شده اند **name** و **value**.

Name ویژگی مورد نظر شما برای تنظیم میباشد، به عنوان مثال عنصر پاراگراف **<p>** در مثال ارائه شده دارای ویژگی می باشد که نام آن **align** می باشد و شما می توانید از آن برای تنظیم پاراگراف در صفحه استفاده کنید.

Value همان است که شما می خواهید مقدار ویژگی تنظیم شود و همیشه در داخل گیومه قرار می دهید. مثال زیر سه مقدار ممکن از یک ویژگی تراز را نشان می دهد چپ، مرکز و راست.

ویژگی نام ها و ویژگی مقادیر غیرهوشمند می باشند. به هر حال وب جهانی **W3C Consortium** (مقادیر ویژگی ها را در **HTML 4** با حروف کوچک پیشنهاد می دهد).

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Align Attribute Example</title>
</head>
<body>
  <p align="left">This is left aligned</p>
  <p align="center">This is center aligned</p>
  <p align="right">This is right aligned</p>
</body>
</html>
```

این مثال نتیجه ی زیر را دنبال خواهد کرد

This is left aligned

This is center aligned

This is right aligned

ویژگی های اصلی

چهار ویژگی اصلی که می توانند در اکثر عناصر HTML مورد استفاده قرار بگیرند عبارتند از

id

title

class

style

ویژگی id یک برچسب HTML می تواند برای تشخیص یک عنصر در یک صفحه HTML مورد استفاده قرار بگیرد. دو دلیل اصلی برای تمایل شما به استفاده از ویژگی id در یک عنصر وجود دارد

اگر یک عنصر یک ویژگی را به عنوان تشخیص دهنده ی منحصر به فرد استفاده می کند شناخت تنها آن عنصر و محتوای مربوط به آن ممکن می باشد.

اگر دارای دو عنصر هم نام در یک صفحه ی وب می باشید، می توانید از عنصر id برای تشخیص این عناصر هم نام استفاده کنید.

در مورد طراحی صفحه در آموزش های مجزا بحث خواهیم کرد، اکنون اجازه بدهید از ویژگی id برای تشخیص عناصر بین دو پاراگراف استفاده کنیم، مانند مثال زیر

`<p id="html">This para explains what is HTML</p>`

`<p id="css">This para explains what is Cascading Style Sheet</p>`

ویژگی title

این ویژگی یک تیتل پیشنهادی برای عنصر ارائه می دهد. ترکیب مربوط به ویژگی **title** شبیه به ترکیب توضیح داده شده برای ویژگی id می باشد. رفتار این ویژگی بستگی به عنصری دارد که آن را حمل می کند، گرچه اغلب اوقات وقتی مکان نما روی عنصر قرار می گیرد یا عنصر در حال بارگذاری می باشد، با عنوان یک راهنمای ابزار (**tooltip**) نمایش داده می شود.

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>The title Attribute Example</title>
</head>
<body>
  <h3 title="Hello HTML!">Titled Heading Tag Example</h3>
</body>
</html>
```

مثال بالا نتیجه ی زیر را تولید خواهد کرد

Titled Heading Tag Example

اکنون سعی کنید مکان نما را روی **"Titled Heading Tag Example"** بیاورید، خواهید دید که هر تیتلی که در کد خود استفاده کرده اید، مانند یک راهنمای ابزار نمایش داده خواهد شد.

ویژگی class

این ویژگی برای برقراری ارتباط بین یک عنصر با یک صفحه ی طراحی استفاده می شود و گروه عنصر را مشخص می کند. وقتی که را یاد بگیرید (**Cascading Style Sheet (CSS)**) در مورد این ویژگی بیشتر فرا خواهید گرفت. اکنون تا همین حد کافیست.

مقدار ویژگی نیز ممکن است لیستی از فضاها ی مجزای نام های گروه باشد، برای مثال

```
class="className1 className2 className3"
```

ویژگی style

این ویژگی به شما اجازه می دهد تا قوانین CSS را در داخل عنصر مشخص کنید.

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>The style Attribute</title>
</head>
<body>
  <p style="font-family:arial; color:#FF0000;">Some text...</p>
</body>
</html>
```

این مثال نتیجه ی زیر را تولید می کند

Some text...

اکنون CSS را یاد نمیگیریم بنابراین اجازه دهید بدون اینکه بیشتر از این خود را در مورد CSS اذیت کنیم، به جلوتر برویم. در اینجا لازم است در مورد ویژگی های HTML بدانید و اینکه چگونه در هنگام طراحی متن مورد استفاده قرار می گیرند.

درونی کردن ویژگی ها

سه ویژگی درونی وجود دارند که برای اکثر عناصر HTML استفاده می شوند

dir

lang

xml:lang

ویژگی dir

این ویژگی به شما اجازه می دهد تا مسیری را به مرورگر نشان دهید که متن در آن باید جریان داشته باشد. ویژگی dir می تواند یکی از دو مقدار باشد، همانطور که در جدول زیر نشان داده شده است

Value	Meaning
ltr	از چپ به راست (مقدار پیش فرض)
rtl	راست به چپ (برای زبان هایی مانند هبرو یا عربی که از راست به چپ خوانده می شوند.)

مثال

```
<!DOCTYPE html>
<html dir="rtl">
<head>
  <title>Display Directions</title>
</head>
<body>
  This is how IE 5 renders right-to-left directed text.
</body>
</html>
```

این مثال نتیجه ی زیر را تولید خواهد کرد

This is how IE 5 renders right-to-left directed text.

وقتی که ویژگی **dir** در داخل برچسب **<html>** استفاده می شود، مشخص می کند که چگونه متن در کل داکيومنت نمایش داده می شود. وقتی در برچسب دیگری مورد استفاده قرار بگیرد، مسیر متن را برای محتوای مربوط به آن برچسب کنترل می کند.

ویژگی **lang** این ویژگی به شما کمک می کند تا زبان اصلی استفاده شده در یک داکيومنت را نشان دهید، اما این ویژگی فقط برای سازگاری معکوس با ورژن های قدیمی تر **HTML** در **HTML** حفظ شده است. این ویژگی به وسیله ی ویژگی **xml:lang** در داکيومنت های جدید **HTML** جایگزین شده است.

مقادیر ویژگی **lang** کدهای دو کاراکتری زبان استاندارد **ISO-639** می باشند. **HTML Language Codes** **ISO 639** را برای لیست کاملی از کدهای زبان چک کنید.

مثال

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>English Language Page</title>
</head>
<body>
  This page is using English Language
</body>
</html>
```

ویژگی xmlang

این ویژگی جایگزین **xhtml** برای ویژگی **lang** می باشد. مقدار ویژگی **xmlang** باید یک کد ISO-639 می باشد، همانطور که در بخش قبل بیان شد.

ویژگی های عمومی

در اینجا جدولی را می بینید که برخی دیگر از ویژگی هایی که در برچسب های **HTML** مفید می باشند، را نشان می دهد.

Attribute	Options	Function
align	راست، چپ، مرکز	برچسب ها را به طور افقی تنظیم می کند.
valign	بالا، وسط، پایین.	برچسب ها را به طور عمودی در یک عنصر HTML تنظیم می کند.
bgcolor	numeric, hexadecimal, RGB values	پشت یک عنصر یک رنگ پس زمینه قرار می دهد.
background	URL	پشت یک عنصر یک تصویر پس زمینه قرار می دهد.

id	User Defined	عنصری را برای استفاده با Cascading Style Sheets (CSS) نام می برد.
class	User Defined	عنصری را برای استفاده با Cascading Style Sheets طبقه بندی می کند.
width	Numeric Value	عرض جدول ها، تصاویر و یا سلول های جدول ها را مشخص می کند.
height	Numeric Value	طول جدول ها، تصاویر و یا سلول های جدول ها را مشخص می کند.
title	User Defined	تیتر عناصر را پاپ آپ می کند.

قالب بندی صفحات HTML

اگر با پردازشگر **word** کار می کنید باید با بولد کردن (**bold**)، ایتالیک کردن و آندرلاین کردن متن آشنا باشید. این ها فقط سه گزینه از ده گزینه موجود برای چگونگی ظاهر شدن متن در **HTML** و **XHTML** می باشند.

بولد کردن متن

هرچیزی که بین عناصر **...**

قرار می گیرد به صورت بولد ظاهر می شود، مانند مثال زیر

مثال

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



```
<title>Bold Text Example</title>
</head>
<body>
  <p>The following word uses a <b>bold</b> typeface.</p>
</body>
</html>
```

مثال بالا نتیجه ی زیر را در بر خواهد داشت

The following word uses a bold typeface.

ایتالیک کردن متن

هر چیزی که بین عناصر `<i>...</i>` قرار بگیرد

قرار بگیرد به صورت ایتالیک ظاهر می شود، مانند مثال زیر

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Italic Text Example </title>
</head>
<body>
  <p>The following word uses a <i>italicized</i> typeface.</p>
</body>
</html>
```

مثال بالا نتیجه زیر را تولید خواهد کرد

The following word uses a *italicized* typeface.

آندرلاین کردن متن

هر چیزی که بین عناصر `<u>...</u>` قرار بگیرد به صورت آندرلاین ظاهر می شود، مانند مثال زیر

```
<!DOCTYPE html>
<html>
<head>
  <title>Underlined Text Example</title>
</head>
<body>
  <p>The following word uses a <u>underlined</u> typeface.</p>
</body>
</html>
```

مثال بالا نتیجه ی زیر را تولید خواهد کرد

The following word uses a underlined typeface.

خط کشیدن روی متن

هر چیزی که بین گزینه های `<strike>...</strike>` قرار بگیرد، با یک **strikethrough** نمایش داده می شود که خط باریکی می باشد که روی متن کشیده می شود، همانطور که در مثال زیر مشاهده می کنید

```
<!DOCTYPE html>
<html>
<head>
  <title>Strike Text Example</title>
</head>
<body>
  <p>The following word uses a <strike>strikethrough</strike> typeface.</p>
</body>
</html>
```

مثال بالا نتیجه ی زیر را تولید خواهد کرد

The following word uses a ~~strikethrough~~ typeface.

فونت monospaced

محتوای عنصر `<tt>...</tt>` به فونت **monospaced** نوشته می شود. اکثر فونت ها با عنوان فونت هایی با عرض متغیر شناخته شده اند، زیرا حروف مختلف دارای عرضهای مختلف هستند (به عنوان مثال حرف m عریض تر از حرف l می باشد.) به هر حال در فونت **monospaced** تمام حروف دارای عرض یکسان می باشند.

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Monospaced Font Example</title>
</head>
<body>
  <p>The following word uses a <tt>monospaced</tt> typeface.</p>
</body>
</html>
```

این مثال نتیجه ی زیر را تولید خواهد کرد

The following word uses a monospaced typeface.

متن superscript (چاپ شده در بالا)

محتوای عنصر `^{...}` در بالا نوشته می شود، فونت استفاده شده برای آن همان فونت کاراکترهای اطراف آن می باشد، اما به اندازه ی نصف یک کاراکتر بالاتر از دیگر کاراکترها نمایش داده می شود.

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Superscript Text Example</title>
</head>
<body>
  <p>The following word uses a <sup>superscript</sup> typeface.</p>
</body>
</html>
```

این مثال نتیجه ی زیر را تولید خواهد کرد

The following word uses a ^{superscript} typeface.

متن subscript (چاپ شده در زیر)

محتوای عنصر `_{...}` در زیر نوشته می شود. اندازه ی فونت استفاده شده برای آن به اندازه ی فونت کاراکترهای اطراف می باشد اما به اندازه ی نصف ارتفاع یک کاراکتر زیر کاراکترهای دیگر نمایش داده می شود.

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Subscript Text Example</title>
</head>
<body>
  <p>The following word uses a <sub>subscript</sub> typeface.</p>
</body>
</html>
```

این مثال نتیجه ی زیر را تولید خواهد کرد

The following word uses a subscript typeface.

متن مندرج

هر چیزی که بین عنصر `<ins>...</ins>` قرار بگیرد به عنوان متن مندرج نمایش داده می شود.

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Inserted Text Example</title>
</head>
<body>
  <p>I want to drink <del>cola</del> <ins>wine</ins></p>
</body>
</html>
```

این مثال نتیجه ی زیر را تولید خواهد کرد

I want to drink cola wine

متن حذف شده

هر چیزی که بین عنصر `...` ظاهر شود با عنوان یک متن حذف شده نمایش داده می شود.

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Deleted Text Example</title>
</head>
<body>
  <p>I want to drink <del>cola</del> <ins>wine</ins></p>
</body>
</html>
```

این مثال نتیجه ی زیر را تولید می کند

I want to drink ~~cola~~ wine

متن بزرگتر

محتوای عنصر `<big>...</big>` اندازه فونت را بزرگتر از متن اطراف نشان می دهد. مانند مثال زیر

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Larger Text Example</title>
</head>
<body>
  <p>The following word uses a <big>big</big> typeface.</p>
</body>
</html>
```

این مثال نتیجه زیر را تولید می کند

The following word uses a big typeface.

متن کوچکتر

محتوای عنصر `<small>...</small>` متن را یک سایز کوچکتر از متن اطراف آن نشان می دهد، مانند مثال زیر

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Smaller Text Example</title>
</head>
<body>
  <p>The following word uses a <small>small</small> typeface.</p>
</body>
</html>
```

این مثال نتیجه ی زیر را تولید خواهد کرد

The following word uses a small typeface.

گروه بندی محتوا

عناصر `<div>` و `` به شما اجازه می دهند تا برای ایجاد بخش ها و یا زیرمجموعه های یک صفحه، عناصر زیادی را با یکدیگر گروه بندی کنید.

برای مثال ممکن است تمایل داشته باشید که تمام پاورقی ها را در یک صفحه در داخل عنصر `<div>` قرار دهید تا نشان دهید که تمام عناصر موجود در آن عنصر مربوط به پاورقی می باشند. پس از آن ممکن است طرحی را به عنصر `<div>` ضمیمه کنید، طوری که با استفاده از مجموعه ای از قوانین طراحی ظاهر شوند.

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Div Tag Example</title>
</head>
<body>
  <div id="menu" align="middle">
    <a href="/index.htm">HOME</a> |
    <a href="/about/contact_us.htm">CONTACT</a> |
    <a href="/about/index.htm">ABOUT</a>
  </div>
  <div id="content" align="left" bgcolor="white">
    <h5>Content Articles</h5>
    <p>Actual content goes here.....</p>
  </div>
</body>
</html>
```

این مثال نتیجه ی زیر را تولید خواهد کرد

[HOME](#) | [CONTACT](#) | [ABOUT](#)

CONTENT ARTICLES

Actual content goes here.....

از طرف دیگر عنصر `` فقط می تواند برای گروه بندی داخل خطی عناصر استفاده شود. بنابراین اگر بخشی از جمله یا پاراگراف را دارید که می خواهید با هم در یک گروه قرار دهید، می توانید از عنصر `` مانند زیر استفاده کنید

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Span Tag Example</title>
</head>
<body>
```

```
<p>This is the example of <span style="color:green">span tag</span> and the <span style="color:red">div tag</span> alongwith  
CSS</p>  
</body>  
</html>
```

این مثال نتیجه ی زیر را تولید خواهد کرد

This is the example of **span tag** and the **div tag** alongwith CSS

این برچسب ها عموما با **CSS** استفاده می شوند تا به شما اجازه دهند طرحی را به بخشی از یک صفحه ضمیمه کنید.

آموزش HTML در Meta Tag

HTML علاوه بر مشخص کردن اطلاعات مهم به روش های مختلف در مورد یک داکيومنت، به شما اجازه می دهد تا متادیتا را نیز مشخص کنید. عناصر **META** می توانند برای وارد کردن جفت مقدار/نام استفاده شوند، این جفت ویژگی های داکيومنت **HTML** مانند نویسنده، تاریخ اتمام، لیست کلمات کلیدی، داکيومنت نویسنده و غیره را ارائه میدهد.

برچسب **<meta>** برای ارائه ی چنین اطلاعات اضافه ای استفاده می شود. این برچسب یک عنصر خالیست و دارای برچسب بستن نیست اما اطلاعاتی را با ویژگی آن در خود دارد. شما می توانید بر اساس اطلاعاتی که می خواهید در داکيومنت خود نگهداری کنید، یک برچسب یا بیشتر از یک برچسب متا وارد داکيومنت خود کنید. اما به طور کل برچسب های متا وضعیت ظاهری داکيومنت را تحت تاثیر قرار نمی دهند، بنابراین از لحاظ ظاهری استفاده کردن یا نکردن از آنها مشخص نمی شود.

افزودن برچسب های متا به داکيومنت ها

شما می توانید با قرار دادن برچسب های **<meta>** در داخل تیتیر داکيومنت که با برچسب های **<head>** و **</head>** مشخص می شوند، متا دیتا را به صفحات وب خود اضافه کنید. یک برچسب متا علاوه بر ویژگی های مرکزی، می تواند دارای ویژگی های زیر نیز باشد

Attribute	Description
Name	نام برای پراپرتی، می تواند هر چیزی از جمله کلمات کلیدی، توصیفات، نویسنده، تولید کننده و غیره باشد.
content	مقدار ویژگی را مشخص می کند.
scheme	نموداری را برای توضیح مقدار ویژگی مشخص می کند. (همانطور که در محتوی ویژگی مشاهده کردید.)
http-equiv	برای تیتروهای پیام پاسخ http استفاده می شود. برای مثال http-equiv می تواند برای تازه سازی صفحات یا تنظیم یک cookie استفاده شود. مقادیری مانند نوع محتوا، اتمام، تازه سازی و تنظیم cookie.

مشخص کردن کلمات کلیدی

می توانید از برچسب **<meta>** برای مشخص کردن کلمات کلیدی مربوط به داکيومنت استفاده کنید، و پس از آن این کلمات توسط موتورهای جستجو استفاده می شوند، و صفحه ی وب شما را به هدف جستجو ایندکس می کنند.

مثال

در این مثال برچسب های متا، متادیتا و **HTML** را با عنوان کلمات کلیدی در مورد داکيومنت وارد می کنیم.

```
<!DOCTYPE html>
<html>
<head>
  <title>Meta Tags Example</title>
  <meta name="keywords" content="HTML, Meta Tags, Metadata" />
</head>
<body>
  <p>Hello HTML5!</p>
</body>
```



```
</html>
```

این مثال نتیجه ی زیر را به دنبال دارد

Hello HTML5!

توصیف داکيومنت

شما می توانید از برچسب **<meta>** برای توصیف داکيومنت استفاده کنید. این برچسب نیز می تواند توسط موتورهای مختلف جستجو مورد استفاده قرار بگیرد، در حالیکه صفحه ی وب شما را به هدف جستجو ایندکس می کند.

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Meta Tags Example</title>
  <meta name="keywords" content="HTML, Meta Tags, Metadata" />
  <meta name="description" content="Learning about Meta Tags." />
</head>
<body>
  <p>Hello HTML5!</p>
</body>
</html>
```

بازبینی تاریخ داکيومنت

شما می توانید از برچسب **<meta>** برای ارائه ی اطلاعات در مورد زمان آخرین آپدیت داکيومنت استفاده کنید. این اطلاعات می توانند توسط مرورگرهای مختلفی استفاده شوند، در حالیکه صفحه ی وب شما را تازه سازی می کنند.

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Meta Tags Example</title>
  <meta name="keywords" content="HTML, Meta Tags, Metadata" />
  <meta name="description" content="Learning about Meta Tags." />
  <meta name="revised" content="Tahlildadeh, 3/7/2014" />
</head>
<body>
```

```
<p>Hello HTML5!</p>
</body>
</html>
```

تازه سازی داکيومنت

یک برچسب **<meta>** می تواند برای مشخص کردن دوره ای که پس از آن صفحه ی وب شما به طور خودکار بازسازی می شود، استفاده شود.

مثال

اگر می خواهید صفحه ی وب شما پس از هر 5 ثانیه ریفرش شود، از ترکیب زیر استفاده کنید

```
<!DOCTYPE html>
<html>
<head>
<title>Meta Tags Example</title>
<meta name="keywords" content="HTML, Meta Tags, Metadata" />
<meta name="description" content="Learning about Meta Tags." />
<meta name="revised" content="Tahlildadeh, 3/7/2014" />
<meta http-equiv="refresh" content="5" />
</head>
<body>
<p>Hello HTML5!</p>
</body>
</html>
```

Redirect کردن صفحه

می توانید از برچسب **<meta>** برای **Redirect** کردن صفحه ی خود استفاده کنید، همچنین می توانید دوره ای را مشخص کنید که پس از آن صفحه به طور خودکار **Redirect** شود.

مثال

در این مثال صفحه ی جاری پس از 5 ثانیه به صفحه ی دیگر **Redirect** می شود. اگر می خواهید صفحه فوراً **Redirect** شود، هیچ محتوایی برای آن مشخص نکنید.

```
<!DOCTYPE html>
<html>
<head>
<title>Meta Tags Example</title>
<meta name="keywords" content="HTML, Meta Tags, Metadata" />
<meta name="description" content="Learn." />
```

```
<meta name="revised" content="Tahlildadeh, 3/7/2014" />
<meta http-equiv="refresh" content="5" url=http://www.tahlildadeh.com" />
</head>
<body>
  <p>Hello HTML5!</p>
</body>
</html>
```

تنظیم cookies

Cookies داده هایی هستند که در یک فایل کوچک متن روی کامپیوتر شما ذخیره شده اند و بین مرورگر وب و سرور وب ردو بدل می شود تا مسیر اطلاعات مختلف را براساس نیاز برنامه ی وب شما حفظ کنند.

شما می توانید از برچسب **<meta>** برای ذخیره ی **cookies** در بخش کاربری استفاده کنید و پس از آن این اطلاعات می توانند توسط سرور وب استفاده شوند تا بازدیدکننده ی سایت را پیگیری کنند.

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Meta Tags Example</title>
  <meta name="keywords" content="HTML, Meta Tags, Metadata" />
  <meta name="description" content="Learning about Meta Tags." />
  <meta name="revised" content="Tahlildadeh, 3/7/2014" />
  <meta http-equiv="cookie" content="userid=xyz; expires=Wednesday, 08-Aug-15 235959 GMT;" />
</head>
<body>
  <p>Hello HTML5!</p>
</body>
</html>
```

اگر تاریخ و زمان انقضا را مشخص نکرده اید، **cookie** یک **session cookie** می باشد و وقتی که یوزر از مرورگر خارج شود، پاک خواهد شد.

تنظیم نام نگارنده

می توانید با استفاده از **meta tag** نام یک نگارنده را برای صفحه ی وب خود تنظیم کنید. یک مثال در این رابطه را در زیر مشاهده می کنید.

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Meta Tags Example</title>
  <meta name="keywords" content="HTML, Meta Tags, Metadata" />
  <meta name="description" content="Learning about Meta Tags." />
  <meta ame="author" content="Mahnaz Mohtashim" />
</head>
<body>
  <p>Hello HTML5!</p>
</body>
</html>
```

تعیین تنظیم کاراکتر

می توانید از برچسب **<meta>** برای تعیین تنظیم کاراکتر مربوط به صفحه ی وب استفاده کنید.

مثال

به طور پیش فرض مرورگرها و سرورهای وب از رمزگذاری **ISO-8859-1** برای پردازش صفحات وب استفاده می کنند. در زیر مثالی را می بینید برای تنظیم رمزگذاری **UTF-8**

```
<!DOCTYPE html>
<html>
<head>
  <title>Meta Tags Example</title>
  <meta name="keywords" content="HTML, Meta Tags, Metadata" />
  <meta name="description" content="Learning about Meta Tags." />
  <meta ame="author" content="Mahnaz Mohtashim" />
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>
<body>
  <p>Hello HTML5!</p>
</body>
</html>
```

برای ارائه ی کاراکترهای چینی سنتی به یک صفحه ی استاتیک، صفحه ی وب باید دارای یک برچسب **<meta>** باشد تا رمزگذاری **Big5** را تنظیم کند.

```
<!DOCTYPE html>
<html>
<head>
  <title>Meta Tags Example</title>
  <meta name="keywords" content="HTML, Meta Tags, Metadata" />
  <meta name="description" content="Learning about Meta Tags." />
  <meta ame="author" content="Mahnaz Mohtashim" />
  <meta http-equiv="Content-Type" content="text/html; charset=Big5" />
</head>
```

```
<body>
  <p>Hello HTML5!</p>
</body>
</html>
```

آموزش استفاده از comment در html

کامنت قطعه ای از کد می باشد که توسط مرورگرها نادیده گرفته می شود. افزودن کامنت به کد HTML خود، تمرین خوبی است، به ویژه در داکيومنت های پیچیده برای نشان دادن بخش هایی از یک داکيومنت و یادداشت هایی دیگر به هرکس که کد را نگاه می کند. کامنت ها به شما و دیگران کمک می کنند تا کد خود را متوجه شوید و قابلیت خواندن آن را افزایش می دهد.

کامنت های HTML بین برچسب های `<!-- ... -->` واقع می شوند. بنابراین هر محتوایی که بین این برچسب ها واقع شود، مثل کامنت با آن رفتار خواهد شد و توسط مرورگرها به طور کامل نادیده گرفته خواهد شد.

مثال

```
<!DOCTYPE html>
<html>
  <head>
    <!-- Document Header Starts -->
    <title>This is document title</title>
  </head> <!-- Document Header Ends -->
  <body>
    <p>Document content goes here.....</p>
  </body>
</html>
```

این مثال نتیجه ی زیر را بدون نمایش محتوای ارائه شده به عنوان بخشی از کامنت ها، به دنبال دارد.

Document content goes here.....

کامنت های معتبر در مقابل کامنت های نامعتبر

کامنت ها تودرتو نمی شوند، یعنی اینکه یک کامنت نمی تواند در داخل کامنت دیگری قرار بگیرد. خط تیره ی (dash) دوتایی `--` ممکن نیست در داخل یک کامنت ظاهر شود، به جز در مواردی که بخشی از برچسب closing باشد. شما باید مطمئن باشید که هیچ فضایی در ابتدای رشته ی کامنت وجود ندارد.

مثال

در اینجا کامنت ارائه شده یک کامنت معتبر می باشد و توسط مرورگر پاک می شود.

```
<!DOCTYPE html>
<html>
<head>
  <title>Valid Comment Example</title>
</head>
<body>
  <!-- This is valid comment -->
  <p>Document content goes here.....</p>
</body>
</html>
```

اما خط زیر یک کامنت معتبر نیست و توسط مرورگر نمایش داده خواهد شد. این به این خاطر است که فضایی بین حاشه ی چپ پرانتز و علامت تعجب وجود دارد.

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Invalid Comment Example</title>
</head>
<body>
  <!-- This is not a valid comment -->
  <p>Document content goes here.....</p>
</body>
</html>
```

این مثال نتیجه ی زیر را به دنبال خواهد داشت

```
<!-- This is not a valid comment -->
Document content goes here.....
```

کامنت های چندخطی

تاکنون فقط کامنت های تک خطی را مشاهده کردیم، اما **HTML** کامنت های چندخطی را نیز پشتیبانی می کند.

شما می توانید کامنت های چند خطی داشته باشید با استفاده از برچسب آغازگر **<!-->** و پایان دهنده ی **-->** که قبل از اولین خط و در پایان آخرین خط قرار دارد، همانطور که در مثال زیر می بینید.

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Multiline Comments</title>
</head>
<body>
  <!--
  This is a multiline comment and it can
  span through as many as lines you like.
  -->
  <p>Document content goes here.....</p>
</body>
</html>
```

این مثال نتیجه ی زیر را ایجاد خواهد کرد

Document content goes here.....

کامنت های شرطی

کامنت های شرطی تنها در اینترنت اکسپلورر روی ویندوز کار می کنند، اما توسط مرورگرهای دیگر نادیده گرفته می شوند. این کامنت ها در اکسپلورر 5 به بالاتر پشتیبانی می شوند و می توانید از آنها برای دادن دستورات شرطی به ورژن های مختلف IE استفاده کنید.

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Conditional Comments</title>
  <!--[if IE 6]>
    Special instructions for IE 6 here
  <![endif]-->
</head>
<body>
  <p>Document content goes here.....</p>
</body>
</html>
```

در جایی که نیاز خواهید داشت یک صفحه ی طراحی متفاوت بر اساس ورژن های مختلف اینترنت اکسپلورر به کار بگیرید، به راه حلی نیاز خواهید داشت. در این مواقع چنین کامنت های شرطی مفید خواهند بود.

استفاده از برچسب کامنت

مرورگرهای کمی هستند که استفاده از برچسب `<comment>` را برای کامنت بخشی از کد HTML استفاده می کنند.

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Using Comment Tag</title>
</head>
<body>
  <p>This is <comment>not</comment> Internet Explorer.</p>
</body>
</html>
```

اگر از IE استفاده می کنید، نتیجه ی زیر حاصل خواهد شد

This is Internet Explorer.

اما اگر در حال استفاده از IE نمی باشد، نتیجه ی زیر حاصل می شود.

This is Internet Explorer.

کد اسکریپت کامنت

گرچه شما جاوا اسکریپت را با HTML یاد خواهید گرفت، اما در یک آموزش مجزا. در اینجا باید دقت کنید که اگر در حال استفاده از `java script` یا `vb script` در کد `html` خود هستید، بنابراین توصیه می شود که آن کد اسکریپت را در داخل کامنت های مناسب HTML قرار دهید طوری که مرورگرهای قدیمی بتوانند به درستی کار کنند.

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Commenting Script Code</title>
  <script>
<!--
  document.write("Hello World!")
```



```
//-->
</script>
</head>
<body>
  <p>Hello , World!</p>
</body>
</html>
```

این مثال نتیجه ی زیر را در پی خواهد داشت

Hello World!

Hello , World!

کامنت صفحات طراحی

گرچه شما **HTML** را در یک آموزش مجزا با صفحات طراحی فرا میگیرید، اما در اینجا به باید دقت داشته باشید که اگر از **CSS** در کد **HTML** خود استفاده می کنید، بنابراین توصیه می شود که کد صفحه ی طراحی را داخل کامنت های مناسب **HTML** قرار دهید، طوری که مرورگرهای قدیمی بتوانند کار کنند.

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Commenting Style Sheets</title>
  <style>
    <!--
    .example {
      border : 1px solid #4a7d49;
    }
    <!-->
  </style>
</head>
<body>
  <div class="example">Hello , World!</div>
</body>
</html>
```

این مثال نتیجه ی زیر را تولید می کند

Hello , World!

وارد کردن تصاویر در صفحات HTML

در این آموزش چگونگی استفاده از تصاویر در صفحات HTML را فرا می‌گیریم. زیبا سازی تصاویر و همچنین ترسیم بسیاری از مفاهیم پیچیده به یک روش ساده روی صفحه ی وب شما، بسیار مهم می باشد. این آموزش مراحل ساده ی استفاده از تصاویر در صفحات وب را به شما آموزش خواهد داد.

وارد کردن تصویر

شما می توانید با استفاده از برچسب `` هر تصویری را وارد صفحه ی وب خود کنید. در زیر ترکیب ساده ی استفاده از این برچسب را می بینید

```

```

برچسب `` یک برچسب خالی می باشد، یعنی تنها می تواند دارای لیستی از ویژگی ها باشد و دارای برچسب closing نمی باشد.

مثال برای امتحان کردن مثال، زیر اجازه بدهید فایل `html` خود یعنی `test.htm` و فایل تصویر خود یعنی `test.png` را در یک مسیر قرار دهیم.

```
<!DOCTYPE html>
<html>
<head>
  <title>Using Image in Webpage</title>
</head>
<body>
  <p>Simple Image Insert</p>
  
</body>
</html>
```

این مثال نتیجه ی زیر را به دنبال خواهد داشت

Simple Image Insert

Simple Image Insert



شما می توانید از فایل تصویر **JPEG**، **PNG** یا **GIF** متناسب با راحتی خود استفاده کنید، اما مطمئن شوید که در ویژگی **src** نام فایل تصویر را به درستی وارد کردید. نام تصویر همیشه یک مورد هوشمند می باشد. ویژگی **alt** یک ویژگی می باشد که اگر تصویر نمایش داده نشود، یک متن جایگزین را برای آن مشخص می کند.

تنظیم موقعیت تصویر

معمولا ما تمام تصاویر خود را در یک مسیر مجزا قرار می دهیم. بنابراین اجازه بدهید فایل **test.htm** مربوط به **HTML** را در مسیر اصلی نگه داشته و یک مسیر فرعی **images** داخل مسیر اصلی، جایی که تصویر **test.png** را نگهداری می کنیم، ایجاد کنیم.

با فرض اینکه موقعیت تصویر ما **"image/test.png"** می باشد، مثال زیر را امتحان کنید

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Using Image in Webpage</title>
</head>
<body>
  <p>Simple Image Insert</p>
  
</body>
</html>
```

این مثال نتیجه ی زیر را به دنبال دارد

Simple Image Insert

Simple Image Insert



تنظیم طول و عرض تصویر

می توانید طول و عرض تصویر را براساس نیاز خود و با استفاده از ویژگی های **width** و **height** تنظیم کنید.
شما می توانید طول و عرض تصویر را به پیکسل یا براساس درصد اندازه ی واقعی آن تنظیم کنید.

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Set Image Width and Height</title>
</head>
<body>
  <p>Setting image width and height</p>
  
</body>
</html>
```

این مثال نتیجه ی زیر را در برخواهد داشت

Setting image width and height

Setting image width and height



تنظیم حاشیه ی تصویر

به طور پیش فرض تصویر حاشیه ای در اطراف خود خواهد داشت، شما می توانید ضخامت این حاشیه را با استفاده از ویژگی **border** و به واحد پیکسل تنظیم کنید. ضخامت 0 یعنی هیچ حاشیه ای در اطراف تصویر وجود ندارد.

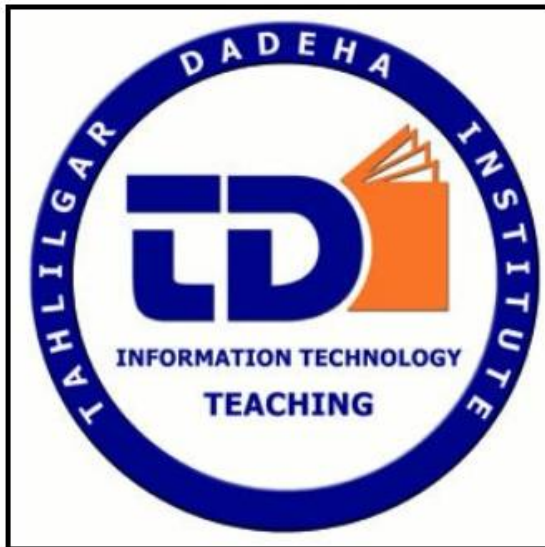
مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Set Image Border</title>
</head>
<body>
  <p>Setting image Border</p>
  
</body>
</html>
```

این مثال نتیجه ی زیر را تولید خواهد کرد

Setting image Border

Setting image Border



تنظیم همترازی تصویر

به طور پیش فرض تصویر در سمت چپ صفحه تنظیم می شود، اما می توانید از ویژگی **align** برای تنظیم تصویر در سمت راست یا مرکز صفحه استفاده کنید.

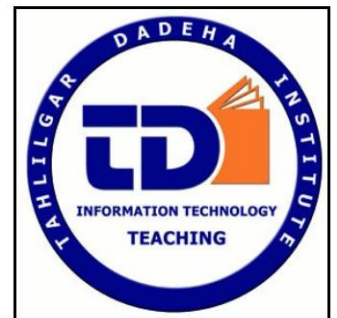
مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Set Image Alignment</title>
</head>
<body>
  <p>Setting image Alignment</p>
  
</body>
</html>
```

این مثال نتیجه ی زیر را به دنبال دارد

Setting image Alignment

Setting image Alignment



آموزش Table در HTML

جدول های **HTML** به نویسندگان وب اجازه می دهند تا داده هایی مانند متن، تصاویر، لینک ها، جدول های دیگر و غیره در ردیف ها و ستون ها تنظیم کنید. جدول های **HTML** با استفاده از برچسب **<table>** ایجاد می شوند که در آن برچسب **<tr>** برای ایجاد ردیف ها و برچسب **<td>** برای ایجاد داده های سلول ها استفاده می شوند.

مثال

```

<!DOCTYPE html>
<html>
<head>
  <title>HTML Tables</title>
</head>
<body>
  <table border="1">
    <tr>
      <td>Row 1, Column 1</td>
      <td>Row 1, Column 2</td>
    </tr>
    <tr>
      <td>Row 2, Column 1</td>
      <td>Row 2, Column 2</td>
    </tr>
  </table>
</body>
</html>

```

در اینجا **border** یک ویژگی از برچسب **<table>** می باشد و برای قرار دادن حاشیه در پهنای همه ی سلول ها استفاده می شود. اگر به حاشیه احتیاج نداشته باشید می توانید از **border="0"** استفاده کنید.

تیتر جدول

تیتر جدول می تواند با استفاده از برچسب **<th>** تعریف شود. این برچسب برای جایگزینی با برچسب **<td>** استفاده می شود که برچسب **<td>** برای نمایش داده ی حقیقی سلول استفاده می شود. طبیعتا شما ردیف های بالای جدول را به عنوان تیتر قرار می دهید.

ویژگی های **cellpadding** و **cellspacing**

دو ویژگی به نام های **Cellpadding** و **Cellspacing** وجود دارند که برای تنظیم محیط سفید در سلول های جدول استفاده می شوند. ویژگی **cellspacing** عرض حاشیه را تعریف می کند، در حالیکه **cellpadding** فاصله ی بین حاشیه ی سلول و محتوای آن را تعریف می کند.

مثال تیتر جدول و **cellpadding** و **cellspacing**

```

<!DOCTYPE html>
<html>
<head>

```

```

<title>HTML Table Cellpadding</title>
</head>
<body>
  <table border="1" cellpadding="5" cellspacing="5">
    <tr>
      <th>Name</th>
      <th>Salary</th>
    </tr>
    <tr>
      <td>Ramesh Raman</td>
      <td>5000</td>
    </tr>
    <tr>
      <td>Shabbir Hussein</td>
      <td>7000</td>
    </tr>
  </table>
</body>
</html>

```

ویژگی های colspan و rowspan

اگر بخواهید دو یا بیشتر از دو ستون را با هم تلفیق کنید از ویژگی **colspan** استفاده می کنید. به طور مشابه اگر بخواهید دو یا بیشتر از دو ردیف را در یک ردیف تلفیق کنید از **rowspan** استفاده کنید.

مثال

```

<!DOCTYPE html>
<html>
<head>
  <title>HTML Table Colspan/Rowspan</title>
</head>
<body>
  <table border="1">
    <tr>
      <th>Column 1</th>
      <th>Column 2</th>
      <th>Column 3</th>
    </tr>
    <tr>
      <td rowspan="2">Row 1 Cell 1</td><td>Row 1 Cell 2</td><td>Row 1 Cell 3</td></tr>
    <tr>
      <td>Row 2 Cell 2</td><td>Row 2 Cell 3</td></tr>
    <tr>
      <td colspan="3">Row 3 Cell 1</td></tr>
  </table>
</body>
</html>

```


پس زمینه ی جدول

می توانید به دو روش زیر برای جدول خود زمینه ای تنظیم کنید.

ویژگی **Bg color** می توانید رنگ زمینه را برای همه ی جدول و یا تنها برای یک سلول تنظیم کنید.

ویژگی **background** می توانید یک تصویر را برای کل جدول یا تنها یک سلول تنظیم کنید.

همچنین می توانید با استفاده از ویژگی **bordercolor** رنگ حاشیه را نیز تنظیم کنید.

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Table Background</title>
</head>
<body>
  <table border="1" bordercolor="green" bgcolor="yellow">
    <tr>
      <th>Column 1</th>
      <th>Column 2</th>
      <th>Column 3</th>
    </tr>
    <tr><td rowspan="2">Row 1 Cell 1</td><td>Row 1 Cell 2</td><td>Row 1 Cell 3</td></tr>
    <tr><td>Row 2 Cell 2</td><td>Row 2 Cell 3</td></tr>
    <tr><td colspan="3">Row 3 Cell 1</td></tr>
  </table>
</body>
</html>
```

در اینجا مثالی در رابطه با استفاده از ویژگی **background** می بینید. در این مثال ما از تصویری موجود در

directory image استفاده کرده ایم.

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Table Background</title>
</head>
<body>
  <table border="1" bordercolor="green" background="/images/test.png">
    <tr>
      <th>Column 1</th>
      <th>Column 2</th>
      <th>Column 3</th>
    </tr>
    <tr><td rowspan="2">Row 1 Cell 1</td><td>Row 1 Cell 2</td><td>Row 1 Cell 3</td></tr>
    <tr><td>Row 2 Cell 2</td><td>Row 2 Cell 3</td></tr>
  </table>
```

```
<tr><td colspan="3">Row 3 Cell 1</td></tr>
</table>
</body>
</html>
```

طول و عرض جدول

شما می توانید طول و عرض جدولی را با استفاده از ویژگی های **width** و **height** تنظیم کنید. شما می توانید طول و عرض جدول را به پیکسل و یا متناسب با درصد صفحه ی جاری تنظیم کنید.

مثال

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Table Width/Height</title>
</head>
<body>
<table border="1" width="400" height="150">
<tr>
<td>Row 1, Column 1</td>
<td>Row 1, Column 2</td>
</tr>
<tr>
<td>Row 2, Column 1</td>
<td>Row 2, Column 2</td>
</tr>
</table>
</body>
</html>
```

شرح جدول

برچسب **caption** یک توضیح یا یک تیترا برای جدول می باشد که در بالای جدول نمایش داده می شود. این برچسب در ورژن های جدید **HTML/XHTML** توصیه می شود.

مثال

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Table Caption</title>
</head>
<body>
<table border="1" width="100%">
<caption>This is the caption</caption>
<tr>
<td>row 1, column 1</td>
```

```

<td>row 1, column 2</td>
</tr>
<tr>
<td>row 2, column 1</td>
<td>row 2, column 2</td>
</tr>
</table>
</body>
</html>

```

تیترا، بدنه و پاورقی جدول

جدول ها می توانند به سه بخش تقسیم شوند تیترا، بدنه و پاورقی جدول. عنوان و فوت (**tfoot**) در واقع شبیه به تیترا و پاورقی در یک فایل پردازش **word** می باشند که برای هرصفحه یکی می باشد، در حالیکه بدنه همان نگه دارنده ی محتوای اصلی جدول می باشد.

سه عنصر برای مجزا کردن عنوان، بدنه و فوت در یک جدول عبارتند از

<thead> برای ایجاد یک تیترا مجزا

<tbody> برای نشان دادن بدنه ی اصلی جدول

<tfoot> برای ایجاد یک پاورقی مجزا در یک جدول

یک جدول برای نشان دادن صفحات و گروه های مختلف یک داده، ممکن است دارای عناصر **<tbody>** مختلفی باشد، اما ظاهر شدن برچسب های **<thead>** و **<tfoot>** قبل از **<tbody>** زیاد مهم نیست.

مثال

```

<!DOCTYPE html>
<html>
<head>
<title>HTML Table</title>
</head>
<body>
<table border="1" width="100%">
<thead>
<tr>
<td colspan="4">This is the head of the table</td>
</tr>
</thead>
<tfoot>
<tr>
<td colspan="4">This is the foot of the table</td>

```

```

</tr>
</tfoot>
<tbody>
<tr>
<td>Cell 1</td>
<td>Cell 2</td>
<td>Cell 3</td>
<td>Cell 4</td>
</tr>
</tbody>
</table>
</body>
</html>

```

جدول های تودرتو

شما می توانید از یک جدول در داخل جدولی دیگر استفاده کنید. نه تنها جدول ها، بلکه می توانید تمام برچسب ها را در داخل برچسب داده ی `<td>` استفاده کنید.

مثال

در زیر **مثالی** را از استفاده ی یک جدول و برچسب های دیگر در داخل یک سلول مشاهده می کنید.

```

<!DOCTYPE html>
<html>
<head>
<title>HTML Table</title>
</head>
<body>
<table border="1" width="100%">
<tr>
<td>
<table border="1" width="100%">
<tr>
<th>Name</th>
<th>Salary</th>
</tr>
<tr>
<td>Ramesh Raman</td>
<td>5000</td>
</tr>
<tr>
<td>Shabbir Hussein</td>
<td>7000</td>
</tr>
</table>
</td>
</tr>
</table>
</body>
</html>

```

آموزش List

HTML سه روش برای مشخص کردن لیست هایی از اطلاعات ارائه می دهد. تمام لیست ها باید شامل یک یا بیشتر از یک عنصر باشند. لیست ها ممکن است شامل موارد زیر شوند.

**** - لیست بدون ترتیب. این لیست آیتم ها را با استفاده از **bullet** های ساده لیست می کند.

**** - لیست منظم. این لیست از نمودارهای عددی مختلف برای لیست کردن آیتم های شما استفاده می کند.

<dl> - لیست تعریف. این لیست آیتم های شما را به همان روشی که در دیکشنری منظم شده اند، منظم می کند.

لیست های بدون ترتیب HTML

لیست بدون ترتیب مجموعه ای از آیتم های مربوط به هم می باشد که هیچگونه نظم و ترتیب خاصی ندارند. این لیست با استفاده از برچسب **** در **HTML** ایجاد می شود. هر آیتم در لیست با یک **bullet** مشخص می شود.

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Unordered List</title>
</head>
<body>
  <ul>
    <li>Beetroot</li>
    <li>Ginger</li>
    <li>Potato</li>
    <li>Radish</li>
  </ul>
</body>
</html>
```

ویژگی type

می توانید از ویژگی **type** برای برچسب **** استفاده کنید تا نوع **bullet** خود را مشخص کنید، که به طور پیش فرض یک دیسک می باشد. در زیر گزینه های ممکن را مشاهده می کنید.

```
<ul type="square">  
  <ul type="disc">  
    <ul type="circle">
```

در زیر **مثالی** را میبینید که در آن از **<ul type="square">** استفاده می کنیم.

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>HTML Unordered List</title>  
</head>  
<body>  
  <ul type="square">  
    <li>Beetroot</li>  
    <li>Ginger</li>  
    <li>Potato</li>  
    <li>Radish</li>  
  </ul>  
</body>  
</html>
```

در زیر **مثالی** را می بینید که در آن از **<ul type="disc">** استفاده کرده ایم.

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>HTML Unordered List</title>  
</head>  
<body>  
  <ul type="disc">  
    <li>Beetroot</li>  
    <li>Ginger</li>  
    <li>Potato</li>  
    <li>Radish</li>  
  </ul>  
</body>  
</html>
```

در زیر **مثالی** را می بینید که در آن از **<ul type="circle">** استفاده کرده ایم.

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>HTML Unordered List</title>  
</head>  
<body>
```

```

<ul type="circle">
  <li>Beetroot</li>
  <li>Ginger</li>
  <li>Potato</li>
  <li>Radish</li>
</ul>
</body>
</html>

```

لیست های منظم HTML

اگر تمایل دارید آیتم های خود را به جای قرار دادن در یک لیست دارای **bullet**، در یک لیست عددگذاری شده قرار دهید، می توانید از لیست منظم HTML استفاده کنید. این لیست با استفاده از برچسب **** ایجاد می شود. شماره گذاری از یک شروع شده و برای هر لیست منظم عنصر بعدی با اضافه شدن یک عدد و به همراه برچسب **** اضافه می شود.

می توانید از ویژگی **type** برای برچسب **** استفاده کنیم تا نوع شماره گذاری مورد نظر خود را مشخص کنید. به طور پیش فرض شماره گذاری به وسیله ی عدد انجام می شود. در زیر گزینه های ممکن را مشاهده می کنید.

```

<ol type="1">
  - Default-Case Numerals.
<ol type="I">
  - Upper-Case Numerals.
<ol type="i">
  - Lower-Case Numerals.
<ol type="a">
  - Lower-Case Letters.
<ol type="A"> - Upper-Case Letters.

```

در زیر مثالی را می بینید که در آن از **<ol type="1">** استفاده کرده ایم.

```

<!DOCTYPE html>
<html>
<head>
  <title>HTML Ordered List</title>
</head>
<body>
  <ol type="1">
    <li>Beetroot</li>
    <li>Ginger</li>
    <li>Potato</li>
    <li>Radish</li>
  </ol>
</body>
</html>

```

در زیر مثالی را می بینید که در آن از **<ol type="I">** استفاده می کنیم.

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Ordered List</title>
</head>
<body>
  <ol type="">
    <li>Beetroot</li>
    <li>Ginger</li>
    <li>Potato</li>
    <li>Radish</li>
  </ol>
</body>
</html>
```

در زیر مثالی را میبینید که در آن از `<ol type="">` استفاده کرده ایم.

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Ordered List</title>
</head>
<body>
  <ol type="i">
    <li>Beetroot</li>
    <li>Ginger</li>
    <li>Potato</li>
    <li>Radish</li>
  </ol>
</body>
</html>
```

در زیر مثالی را می بینید که در آن از `<ol type="A">` استفاده کرده ایم.

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Ordered List</title>
</head>
<body>
  <ol type="A">
    <li>Beetroot</li>
    <li>Ginger</li>
    <li>Potato</li>
    <li>Radish</li>
  </ol>
</body>
</html>
```

در زیر مثالی را می بینید که در آن از `<ol type="a">` استفاده کرده ایم.

```
<!DOCTYPE html>
```



```
<html>
<head>
  <title>HTML Ordered List</title>
</head>
<body>
  <ol type="a">
    <li>Beetroot</li>
    <li>Ginger</li>
    <li>Potato</li>
    <li>Radish</li>
  </ol>
</body>
</html>
```

ویژگی start

شما می توانید از ویژگی **start** برای برچسب **** استفاده کنید تا نقطه ی شروع شماره گذاری خود را مشخص کنید. در زیر گزینه های ممکن را مشاهده می کنید.

```
<ol type="1" start="4">
- Numerals starts with 4.
<ol type="I" start="4">
- Numerals starts with IV.
<ol type="i" start="4">
- Numerals starts with iv.
<ol type="a" start="4">
- Letters starts with d.
<ol type="A" start="4"> - Letters starts with D.
```

در زیر مثالی را می بینید که در آن از **<ol type="i" start="4">** استفاده می کنیم.

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Ordered List</title>
</head>
<body>
  <ol type="i" start="4">
    <li>Beetroot</li>
    <li>Ginger</li>
    <li>Potato</li>
    <li>Radish</li>
  </ol>
</body>
</html>
```

لیست های تعریف HTML

HTML و **XHTML** لیست هایی به نام لیست های تعریف را پشتیبانی می کنند که در این لیست ها ورودی ها مانند ترتیب لغات در دیکشنری قرار می گیرند. این لیست یک روش ایده آل برای ارائه یک فهرست از معانی یا لیستی از اصطلاحات یا لیستی از نام ها و مقادیر می باشد.

لیست تعریف از سه برچسب زیر استفاده می کند.

<dl> - شروع لیست را تعریف می کند.

- یک عبارت

- تعریف عبارت

- پایان لیست را تعریف می کند.

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Definition List</title>
</head>
<body>
  <dl>
    <dt><b>HTML</b></dt>
    <dd>This stands for Hyper Text Markup Language</dd>
    <dt><b>HTTP</b></dt>
    <dd>This stands for Hyper Text Transfer Protocol</dd>
  </dl>
</body>
</html>
```

لینک های متن HTML

یک صفحه ی وب می تواند لینک های متنوعی داشته باشد که شما را مستقیماً به صفحات دیگر یا حتی بخش هایی خاص از یک صفحه ی ارائه شده می برد. این لینک ها هایپرلینک نامیده می شوند.

هایپرلینک ها به بازدیدکنندگان اجازه می دهند تا با کلیک کردن روی لغات، اصطلاحات و تصاویر بین صفحات وب مسیریابی کنند. شما می توانید هایپرلینک ها را روی صفحه ی وب با استفاده از متن یا تصاویر موجود ایجاد کنید.

لینک کردن داکيومنت ها

یک لینک با استفاده از برچسب `<a>` در HTML تعیین می شود. این برچسب `anchor tag` نامیده می شود و هر چیزی بین برچسب آغازین `<a>` و پایانی `` بخشی از لینک می شود و یک یوزر می تواند آن بخش را کلیک کرده تا به داکيومنت لینک شده برسد. در زیر ترکیب ساده ی استفاده از برچسب `<a>` را می بینید.

```
<a href="Document URL" ... attributes-list>Link Text</a>
```

مثال

اجازه دهید مثال زیر را امتحان کنیم که <http://www.tahlildadeh.com> را در صفحه ی شما لینک می کند.

```
<!DOCTYPE html>
<html>
<head>
<title>Hyperlink Example</title>
</head>
<body>
<p>Click following link</p>
<a href="http://www.tahlildadeh.com" target="_self">tahlildadeh </a>
</body>
</html>
```

این مثال نتیجه ی زیر را تولید خواهد کرد که شما می توانید روی لینک تولید شده ی `tahlildadeh` کلیک کنید تا به صفحه ی اصلی `tahlildadeh` برسید.

Click following link

[Tahlildadeh](#)

ویژگی target

ما از ویژگی `target` در مثال قبلی خود استفاده کردیم . این ویژگی برای مشخص کردن موقعیتی که داکيومنت لینک شده باز می شود، مورد استفاده قرار می گیرد. در زیر گزینه های ممکن را مشاهده می کنید.

Option	Description
_blank	داکیومنت لینک شده را در یک پنجره یا تب جدید باز می کند.
_self	داکیومنت لینک شده را در همان چارچوب باز می کند.
_parent	داکیومنت لینک شده را در چارچوب اصلی باز می کند.
_top	داکیومنت لینک شده در کل بدنه ی پنجره باز می شود.
targetframe	داکیومنت لینک شده را در یک targetframe نام گذاری شده باز می کند.

مثال

برای درک تفاوت اصلی در برخی گزینه های ارائه شده در ویژگی **target** مثال زیر را امتحان کنید.

```
<!DOCTYPE html>
<html>
<head>
  <title>Hyperlink Example</title>
  <base href="http://www.tahlildadeh.com/">
</head>
<body>
  <p>Click any of the following links</p>
  <a href="http://www.tahlildadeh.com" target="_blank">Opens in New</a> |
  <a href="http://www.tahlildadeh.com" target="_self">Opens in Self</a> |
  <a href="http://www.tahlildadeh.com" target="_parent">Opens in Parent</a> |
  <a href="http://www.tahlildadeh.com" target="_top">Opens in Body</a></body>
</html>
```

این مثال نتیجه ی زیر را تولید خواهد کرد که می توانید روی لینک های مختلف کلیک کنید تا تفاوت بین گزینه های مختلف ارائه شده برای ویژگی **target** را درک کنید.

Click any of the following links

[Opens in New](#) | [Opens in Self](#) | [Opens in Parent](#) | [Opens in Body](#)

استفاده از مسیر پایه

وقتی که شما داکيومنت های HTML را متناسب با همان وب سایت لینک می کنید، ارائه ی یک URL کامل برای هر لینک ضروری نیست. اگر از برچسب `<base>` در تیتیر داکيومنت HTML خود استفاده می کنید، می توانید از دست آن خلاص شوید. این برچسب برای ارائه ی یک مسیر پایه برای همه ی لینک ها استفاده می شود. بنابراین مرورگر شما مسیر ارائه شده ی مرتبط را به مسیر پایه پیوند خواهد داد و یک URL کامل ایجاد خواهد کرد.

مثال

مثال زیر از برچسب `<base>` برای مشخص کردن URL پایه استفاده می کند و پس از آن ما می توانیم به جای ارائه ی URL کامل برای هر لینک از مسیرهای مرتبط استفاده کنیم.

```
<!DOCTYPE html>
<html>
<head>
  <title>Hyperlink Example</title>
  <base href="http://www.tahlildadeh.com/">
</head>
<body>
  <p>Click following link</p>
  <a href="/html/index.htm" target="_blank">HTML tahlildadeh</a>
</body>
</html>
```

این مثال نتیجه ی زیر را تولید خواهد کرد که می توانید روی لینک تولید شده ی HTML tahlildadeh کلیک کنید تا به آموزش HTML برسید.

اکنون URL ارائه شده ی `` باعنوان

`` در نظر گرفته می شود.

Click following link

HTML tahlildadeh

لینک شدن به بخشی از صفحه

می توانید برای بخش خاصی از صفحه ی وب ارائه شده، با استفاده از ویژگی **name** یک لینک ایجاد کنید. این امر یک فرایند دو مرحله ای می باشد.

نخست اینکه در مکانی که می خواهید به داخل صفحه ی وب برسید یک لینک ایجاد کنید و آن را با استفاده از برچسب **<a...>** نام گذاری کنید.

```
<h1>HTML Text Links <a name="top"></a></h1>
```

مرحله ی دوم ایجاد یک هایپر لینک می باشد برای لینک کردن داکيومنت و قرار دادن در مکانی که می خواهید برسید.

```
<a href="/html/html_text_links.htm#top">Go to the Top</a>
```

و این مثال لینک زیر را تولید خواهد کرد که می توانید در آن روی لینک تولید شده ی **Go to the Top** کلیک کنید تا به نقطه ی بالای آموزش **HTML Text Link** برسید.

Go to the Top

تنظیم رنگ های لینک

شما می توانید رنگ لینک های خود، لینک های فعال و لینکهای مشاهده شده را با استفاده از ویژگی های **link** و **alink** و **vlink** از برچسب **<body>** تنظیم کنید.

مثال

مثال زیر را در **test.htm** ذخیره کنید و سپس می توانید آن را در هر مرورگری باز کنید تا ببینید که چگونه ویژگی های **link**, **alink** و **vlink** کار می کنند.

```
<!DOCTYPE html>
<html>
<head>
<title>Hyperlink Example</title>
<base href="http://www.tahlildadeh.com/">
</head>
<body alink="#54A250" link="#040404" vlink="#F40633">
<p>Click following link</p>
```

```
<a href="/html/index.htm" target="_blank">HTML tahlildadeh</a>
</body>
</html>
```

این مثال نتیجه ی زیر را تولید می کند. فقط رنگ لینک را قبل از کلیک کردن روی آن چک کنید، سپس رنگ آن را در هنگام فعال کردن و بازدید آن چک کنید.

Click following link

[HTML tahlildadeh](#)

دانلود کردن لینک ها

شما می توانید لینک متنی ایجاد کنید تا فایل های قابل دانلود **PDF**، **ZIP** و **DOC** خود را بسازید. این کار بسیار ساده می باشد، فقط کافیست یک **URL** کامل از فایل قابل دانلود ارائه بدهید.

```
<!DOCTYPE html>
<html>
<head>
<title>Hyperlink Example</title>
</head>
<a href="http://tahlildadeh.com/Files/Articles/04GL07.pdf">Download PDF File</a>
</body>
</html>
```

این مثال لینک زیر را تولید خواهد کرد و برای دانلود یک فایل استفاده می شود.

[Download PDF File](#)

آموزش Image Link

تاکنون مشاهده کردیم که چگونه با استفاده از متن، یک لینک هایپر تکست ایجاد کنیم و همچنین فرا گرفته ایم که چگونه از تصاویر در صفحات وب خود استفاده کنیم. اکنون فرا خواهیم گرفت که چگونه با استفاده از تصاویر هایپرلینک ایجاد کنیم.

مثال

استفاده از تصویر به عنوان هایپرلینک بسیار ساده می باشد. لازم است که یک تصویر را در داخل هایپرلینک در محل تصویر قرار دهیم، همانطور که در زیر نشان داده شده است.

```
<!DOCTYPE html>
```

```

<html>
<head>
  <title>Image Hyperlink Example</title>
</head>
<body>
  <p>Click following link</p>
  <a href="http://www.tahliidadeh.com" target="_self">
    
  </a>
</body>
</html>

```

این ساده ترین راه ایجاد هایپرلینک با استفاده از تصاویر می باشد.

تصاویر حساس به ماوس

استانداردهای HTML و XHTML یک ویژگی ارائه می دهند که به شما اجازه می دهد لینک های مختلفی را در داخل یک تصویر اجرا کنید. شما می توانید براساس مختصات مختلف موجود روی تصویر، لینک های مختلفی را روی یک تصویر مجزا ایجاد کنید. زمانی که لینک های متفاوت به مختصات متفاوت ضمیمه می شود، می توانیم برای باز کردن داکيومنت های تارگت روی بخش های مختلف تصویر کلیک کنیم. چنین تصاویری که به ماوس حساس می باشند، تصاویر نقشه نامیده میشوند. دو روش برای ایجاد چنین تصاویری وجود دارد.

Server-side image maps – توسط ویژگی **ismap** از برچسب **** فعال می شود و دستیابی به یک سرور و پردازش برنامه های مربوط به تصویر نقشه لازم می باشد.

Client-side image maps – با استفاده از ویژگی **usemap** از برچسب **** همراه با برچسب های متناظر **<map>** و **<area>** ایجاد می شود.

Server side image maps

در اینجا به سادگی تصویر خود را داخل یک هایپرلینک قرار داده و از ویژگی **ismap** استفاده کنید که آن را یک تصویر خاص می سازد و وقتی یوزر در بخشی از تصویر کلیک می کند، مرورگر مختصات اشاره گر ماوس را همراه با URL مشخص شده در برچسب **<a>** به سرور وب منتقل می کند. سرور از مختصات اشاره گر ماوس استفاده می کند تا تعیین کند کدام داکيومنت باید به مرورگر بازگردانده شود.

زمانی که **ismap** استفاده می شود، ویژگی **href** از برچسب **<a>** باید **URL** یک برنامه ی سرور را مانند یک **cgi** یا اسکریپت **PHP** و غیره در برداشته باشد، تا درخواست ورودی را براساس مختصات انتقال داده شده پردازش کند. مختصات موقعیت ماوس پیکسل های صفحه میباشند که از گوشه ی بالای سمت چپ تصویر شمرده می شوند و با **(0,0)** شروع می شوند. مختصات دنبال شده با یک علامت سوال، به انتهای **URL** اضافه می شوند.

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>ISMAP Hyperlink Example</title>
</head>
<body>
  <p>Click following link</p>
  <p>Click following link</p>
  <a href="Exp1.html" target="_self">
    
  </a>
</body>
</html>
```

سپس مرورگر پارامترهای جستجوی زیر را به سرور می فرستد که می تواند توسط اسکریپت **ismap.cgi** یا فایل **map** پردازش شود و شما می توانید هر داکيومنتی را که دوست داشته باشید به این مختصات لینک کنید.

</cgi-bin/ismap.cgi?20,30>

از این طریق شما می توانید لینک های مختلفی را به مختصات متفاوت یک تصویر اختصاص دهید، و وقتی این مختصات کلیک می شوند، می توانید داکيومنت های لینک شده ی متناظر را باز کنید.

توجه

شما برنامه نویسی **CGI** را زمانی فراخواهید گرفت که برنامه نویسی **perl** را مطالعه کنید. می توانید اسکریپت خود را بنویسید تا این مختصات انتقال داده شده را با استفاده از **PHP** یا هر اسکریپت دیگری پردازش کنید. فعلا اجازه بدهید روی یادگیری **HTML** تمرکز کنیم، می توانید این فصل را بعدا بازبینی کنید.

Client side image maps

این تصاویر به وسیله ی ویژگی **usemap** از برچسب **** فعال می شوند و به وسیله ی برچسب های ضمیمه ی **<map>** و **<area>** تعریف می شوند.

نقشه ای که قرار است نقشه را طراحی کند، به وسیله ی **** به عنوان یک تصویر عادی وارد صفحه می شود، به جز اینکه این مورد ویژگی اضافه ای به نام **usemap** همرا خود دارد. مقدار ویژگی **usemap** مقداری می باشد که در یک برچسب **<map>** استفاده می شود تا برچسب های تصویر و نقشه را لینک کند. **<map>** همراه با برچسب های **<area>** همه ی مختصات تصویر و لینک های مربوطه را تعریف می کند.

برچسب **<area>** در داخل برچسب نقشه مختصات و شکل حاشیه های قابل کلیک در داخل تصویر را تعریف می کند. در اینجا **مثالی** از تصویر نقشه می بینید.

```
<!DOCTYPE html>
<html>
<head>
<title>USEMAP Hyperlink Example</title>
</head>
<body>

<map name="planetmap">
<area shape="rect" coords="0,0,40,126" href="02.jpg">
<area shape="circle" coords="90,58,20" href="04.jpg">
<area shape="circle" coords="124,78,20" href="05.jpg">
</map>
</body>
</html>
```

سیستم مختصات

مقدار حقیقی مختصات کانلا به شکل در سوال وابسته است. در اینجا خلاصه ای می بینید که قرار است با **مثال** های مفصل دنبال شوند.

$$\text{rect} = x_1, y_1, x_2, y_2$$

x1 و **y1** مختصات گوشه ی بالای سمت چپ از مستطیل می باشد. **x2** و **y2** مختصات گوشه ی سمت راست پایین می باشند.

$$\text{circle} = x_c, y_c, \text{radius}$$

x_c و y_c مختصات مرکز دایره و **radius** شعاع دایره می باشد. دایره ای به مرکز **200,50** با شعاع **25** دارای ویژگی **coords="200,50,25"** خواهد بود.

$poly = x_1, y_1, x_2, y_2, x_3, y_3, \dots, x_n, y_n$

جفت های مختلف **x-y** رئوس چند ضلعی می باشند، با یک خط که از یک نقطه به نقطه ی دیگر کشیده شده. یک چندضلعی لوزی شکل با بالاترین راس آن در نقطه ی **20,20** و **40** پیکسل، در عریض ترین نقطه ی خود دارای ویژگی **coords="20,20,40,40,20,60,0,40"** می باشد.

تمام مختصات مربوط به بالاترین گوشه ی سمت چپ تصویر می باشند. هر شکل دارای یک **URL** مربوطه می باشد. می توانید از هر نرم افزار تصویری برای دانستن مختصات موقعیت های مختلف استفاده کنید.

لینک ایمیل HTML

قرار دادن یک لینک ایمیل **HTML** روی صفحه ی وب خود کار سختی نیست، اما این کار ممکن است باعث بروز مشکل اسپم های غیرضروری در اکانت ایمیل شما شود. افرادی هستند که می توانند برنامه هایی را برای جمع آوری چنین ایمیل هایی اجرا کنند و سپس آنها را با راه های مختلف برای اسپم کردن استفاده کنند.

می توانید از گزینه دیگری استفاده کنید تا افراد به راحتی بتوانند به شما ایمیل ارسال کنند. یک گزینه می تواند استفاده از فرم های **HTML** باشد برای جمع آوری داده های یوزر و سپس استفاده از اسکریپت **PHP** یا **CGI** برای ارسال ایمیل.

برای یک **مثال** ساده، فرم **Contact Us** را چک کنید. با استفاده از این فرم یک فیدبک از یوزر گرفته و سپس از یک برنامه ی **CGI** استفاده می کنیم که در حال جمع آوری این اطلاعات و ارسال ایمیل به یک **email ID** ارائه شده می باشد.

برچسب HTML Email

برچسب **<a>** در **HTML** برای مشخص کردن یک آدرس ایمیل و ارسال ایمیل، گزینه هایی را به شما ارائه می دهد. در حالیکه از برچسب **<a>** به عنوان یک **email tag** استفاده می کنید، از **mailtoemail address** همراه با ویژگی **href** نیز استفاده خواهید کرد. در زیر ترکیب استفاده از **mailto** را به جای استفاده از **http** می بینید.

[Send Email](mailto:abc@example.com)

این کد لینک زیر را تولید خواهد کرد که می توانید به آن ایمیل ارسال کنید

Send Email

اکنون اگر یوزر روی این لینک کلیک کند، **email client** آغاز به کار خواهد کرد (مانند **lotus notes** و **outlook express** و غیره). که روی کامپیوتر شما نصب شده است. ریسک دیگری در استفاده از این گزینه برای ارسال ایمیل وجود دارد، زیرا اگر یوزر **email client** نصب شده روی کامپیوتر خود نداشته باشد، ارسال ایمیل ممکن نخواهد بود.

تنظیمات پیش فرض

شما می توانید یک موضوع ایمیل و یک بدنه ی ایمیل پیش فرض همراه با آدرس ایمیل خود ایجاد کنید. در زیر **مثال** استفاده از بدنه و موضوع پیش فرض را مشاهده می کنید.

[Send Feedback](mailto:abc@example.com?subject=Feedback&body=Message)

این کد نیز لینک زیر را تولید خواهد کرد که می توانید ایمیل ارسال کنید.

Send Feedback

آموزش HTML Layout

Layout برای یک صفحه ی وب بسیار مهم است، چرا که دید بهتری نسبت به وب سایت شما ارائه می دهد. طراحی **Layout** خوب با منظره و حس خوب برای یک وب سایت زمان فوق العاده زیادی می گیرد. این روزها همه ی وب سایت های مدرن از چارچوب هایی بر اساس جاوااسکریپت و **CSS** استفاده میکنند تا به وب سایت های پاسخگو و دینامیک پیروز شوند، اما کار یک **Layout** در صفحه ی وب شما، صرفا از **HTML** و ویژگی های آن استفاده می کند.

HTML Layout – استفاده از جدول ها

ساده ترین و محبوب ترین راه برای ایجاد **Layout** ها، استفاده از برچسب **<table>** در **HTML** می باشد. این جدول ها در ردیف ها و ستون ها منظم می شوند که شما می توانید از این ردیف ها و ستون ها به هر طریقی که می خواهید استفاده کنید.

مثال

برای **مثال**، نمونه **HTML Layout** زیر از طریق استفاده ی یک جدول با سه ردیف و دو ستون به دست می آید، اما عنوان و پاورقی ستون هر دو ستون را با استفاده از ویژگی **colspan** احاطه می کند.

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Layout using Tables</title>
</head>
<body>
  <table width="100%" border="0">
    <tr>
      <td colspan="2" bgcolor="#b5dcb3">
        <h1>This is Web Page Main title</h1>
      </td>
    </tr>
    <tr valign="top">
      <td bgcolor="#aaaa" width="50">
        <b>Main Menu</b><br />
        HTML<br />
        PHP<br />
        PERL...
      </td>
      <td bgcolor="#eeee" width="100" height="200">
        Technical and Managerial Tutorials
      </td>
    </tr>
    <tr>
      <td colspan="2" bgcolor="#b5dcb3">
        <center>
          Copyright © 2007 Tahlildadeh.com
        </center>
      </td>
    </tr>
  </table>
</body>
</html>
```

این **مثال** نتیجه ی زیر را تولید خواهد کرد.

This is Web Page Main title		
Main	Menu	Technical and Managerial Tutorials
HTML		
PHP		
PERL...		
Copyright © 2007 Tahlildadeh.com		

Layout چند ستونی – استفاده از جدول ها

شما می توانید صفحه ی وب خود را طوری طراحی کنید تا محتوای وب خود را در چند صفحه قرار دهید. می توانید محتوای خود را در ستون وسط قرار دهید و از ستون سمت چپ برای قرار دادن منو و از ستون سمت راست برای تبلیغ یا موارد دیگر استفاده کنید. این **Layout** بسیار شبیه آنچه می باشد که ما در tahlildadeh.com داشتیم.

در اینجا **مثالی** را از ایجاد یک **Layout** سه ستونی می بینید.

```
<!DOCTYPE html>
<html>
<head>
  <title>Three Column HTML Layout</title>
</head>
<body>
  <table width="100%" border="0">
    <tr valign="top">
      <td bgcolor="#aaaa" width="20%">
        <b>Main Menu</b><br />
        HTML<br />
        PHP<br />
        PERL...
      </td>
      <td bgcolor="#b5dcb3" height="200" width="60%">
        Technical and Managerial Tutorials
      </td>
      <td bgcolor="#aaaa" width="20%">
        <b>Right Menu</b><br />
        HTML<br />
        PHP<br />
      </td>
    </tr>
  </table>
</body>
</html>
```

```

PERL...
</td>
</tr>
</table>
</body>
</html>
    
```

این مثال نتیجه ی زیر را تولید خواهد کرد.

Main	Menu	Technical and Managerial Tutorials	Right	Menu
HTML			HTML	
PHP			PHP	
PERL...			PERL...	

Layout های HTML – استفاده از DIV و Span

عنصر **<div>** یک عنصر **block level** می باشد که برای گروه بندی عناصر **HTML** استفاده می شود. در حالیکه این عنصر یک عنصر **block level** است، عنصر **** برای گروه بندی عناصر در یک سطح درون خطی استفاده می شوند.

گرچه می توانیم با جدول های **HTML** ، **Layout** های بسیار زیبایی به دست آوریم، اما جدول ها در واقع به عنوان ابزار **Layout** طراحی نشده اند، و بیشتر برای نمایش داده های جدولی استفاده می شوند.

توجه

این مثال از **CSS** استفاده می کند، بنابراین قبل از درک این مثال، بهتر است درک بهتری از چگونگی کار **CSS** داشته باشید.

مثال

در اینجا سعی می کنیم با استفاده از برچسب `<div>` همراه با **CSS** همان نتیجه ای را به دست آوریم که هنگام استفاده از برچسب `<table>` در مثال قبل به دست آوردیم.

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Layouts using DIV, SPAN</title>
</head>
<body>
  <div style="width:100%">
    <div style="background-color:#b5dcb3; width:100%">
      <h1>This is Web Page Main title</h1>
    </div>
    <div style="background-color:#aaa; height:200px;width:100px;float:left;">
      <div><b>Main Menu</b></div>
      HTML<br />
      PHP<br />
      PERL...
    </div>
    <div style="background-color:#eee; height:200px;width:350px;float:left;">
      <p>Technical and Managerial Tutorials</p>
    </div>
    <div style="background-color:#aaa; height:200px;width:100px;float:right;">
      <div><b>Right Menu</b></div>
      HTML<br />
      PHP<br />
      PERL...
    </div>
    <div style="background-color:#b5dcb3;clear:both">
      <center>
        Copyright © 2007 Tahlildadeh.com
      </center>
    </div>
  </div>
</body>
</html>
```

شما می توانید با استفاده از **DIV** و **SPAN** به همراه **Layout, CSS** های بهتری طراحی کنید. برای درک بیشتر از **CSS** لطفاً به [CSS Tutorial](#) مراجعه کنید.

آموزش iframes

شما می توانید یک **frame** درون خطی را با استفاده از برچسب `<iframe>` مربوط به **HTML** تعریف کنید. این برچسب به برچسب `<frameset>` ارتباطی ندارد، در عوض می تواند در هر جایی در داکيومنت شما ظاهر شود.

برچسب `<iframe>` یک محدوده ی مستطیلی را در داخل داکيومنت تعريف می کند که در آن مرورگر می تواند یک داکيومنت مجزا را ارائه دهد، مانند نوارهای اسکرول و حاشیه ها.

ویژگی `src` برای مشخص کردن URL مربوط به داکيومنتی استفاده می شود که حاوی `frame` درون خطی می باشد.

مثال

در زیر مثالی را می بینید که چگونگی استفاده از `<iframe>` را توضیح می دهد.

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Iframes</title>
</head>
<body>
  <p>Document content goes here...</p>
  <iframe src="/html/menu.htm" width="555" height="200">
    Sorry your browser does not support inline frames.
  </iframe>
  <p>Document content also go here...</p>
</body>
</html>
```

ویژگی های برچسب `<iframe>`

بسیاری از ویژگی های برچسب `<iframe>` شامل نام، گروه، حاشیه، `id`، `longdesk`، طول حاشیه، عرض حاشیه، نام، اسکرول کردن، روش و تیتیر، درست مانند ویژگی های متناظر با برچسب `<frame>` رفتار می کنند.

Attribute	Description
src	این ویژگی برای نام گذاری فایلی استفاده می شود که باید در <code>frame</code> بارگذاری شود. مقدار آن می تواند هر URL باشد. برای مثال <code>src="/html/top_frame.htm"</code> فایل HTML موجود در مسیر <code>html</code> را بارگذاری خواهد کرد.

name	<p>این ویژگی به شما اجازه می دهد تا یک frame را نامگذاری کنید. این ویژگی نشان می دهد که یک داکيومنت در کدام frame باید بارگذاری شود. هنگامی که می خواهید لینک هایی را در یک frame ایجاد کنید که صفحاتی را در یک frame دیگر بارگذاری می کند، که در این مورد دومین frame برای تشخیص خود به عنوان هدف لینک به نام نیاز دارد، در اینجا این ویژگی بسیار مهم است.</p>
frameborder	<p>این ویژگی مشخص می کند که آیا حاشیه های frame نشان داده شوند یا خیر. این ویژگی مقدار داده شده به ویژگی frameborder روی برچسب <frameset> را می گیرد، البته اگر مقداری مشخص شده باشد، این مقدار می تواند 1 (بله) و یا 0 (خیر) باشد.</p>
marginwidth	<p>این ویژگی به شما اجازه می دهد تا عرض فضای بین حاشیه های چپ و راست frame و محتوای آن را مشخص کنید. مقدار به پیکسل داده می شود. برای مثال</p> <p style="text-align: right;">.marginwidth="10"</p>
marginheight	<p>این ویژگی به شما اجازه می دهد تا طول فضای بین بالا و پایین حاشیه ی frame و محتوای آن را مشخص کنید. مقدار به پیکسل داده می شود. برای مثال</p> <p style="text-align: right;">marginheight="10"</p>
noresize	<p>به طور پیش فرض هر frame را با کلیک کردن و درگ کردن روی حاشیه های آن می توانید تغییر اندازه دهید. ویژگی noresize مانع تغییر اندازه ی frame توسط یوزر می شود. برای مثال</p> <p style="text-align: right;">.noresize="noresize"</p>
scrolling	<p>این ویژگی ظاهر نارهای اسکرول نمایان شده در frame را کنترل می کند. این ویژگی مقادیر "no", "yes", یا "auto" را می گیرد. برای مثال scrolling="no" به معنای نبودن نوارهای اسکرول می باشد.</p>

longdesc	این ویژگی به شما اجازه می دهد تا یک لینک به صفحه ای حاوی یک توصیف طولانی از محتوای frame، ارائه بدهید. برای مثال "longdesc="framedescription.htm"
----------	---

html در background

به طور پیش فرض رنگ زمینه ی صفحه ی وب شما سفید می باشد. ممکن است این زمینه را دوست نداشته باشید، اما نگرانی وجود ندارد. **HTML** دو روش مناسب زیر را ارائه می دهد تا زمینه ی صفحه وب خود را به دلخواه بیاراییید.

زمینه ی **HTML** با رنگ ها.

زمینه ی **HTML** با تصاویر.

اکنون اجازه بدهید هر دو روش را یکی یکی و با استفاده از **مثال** های مناسب بررسی کنیم.

زمینه ی HTML با رنگها

ویژگی **bgcolor** برای کنترل زمینه ی یک عنصر **HTML**، به ویژه بدنه ی صفحه و زمینه ی جدول، استفاده می شود. در زیر ترکیب استفاده از **bgcolor** را با هر برچسب **HTML** می بینید.

```
<tagname bgcolor="color_value" ...>
```

این **color-value** می تواند به هرکدام از فرمت های زیر ارائه شود.

```
<!-- Format 1 - Use color name -->
```

```
<table bgcolor="lime">
```

```
<!-- Format 2 - Use hex value -->
```

```
<table bgcolor="#f1f1f1">
```

```
<!-- Format 3 - Use color value in RGB terms -->
```

```
<table bgcolor="rgb(0,0,120)">
```

مثال

در اینجا **مثالی** را می بینید از تنظیم زمینه ی یک برچسب **HTML**.

```

<!DOCTYPE html>
<html>
<head>
  <title>HTML Background Colors</title>
</head>
<body>
  <!-- Format 1 - Use color name -->
  <table bgcolor="yellow" width="100%">
    <tr>
      <td>
        This background is yellow
      </td>
    </tr>
  </table>
  <!-- Format 2 - Use hex value -->
  <table bgcolor="#6666FF" width="100%">
    <tr>
      <td>
        This background is sky blue
      </td>
    </tr>
  </table>
  <!-- Format 3 - Use color value in RGB terms -->
  <table bgcolor="rgb(255,0,255)" width="100%">
    <tr>
      <td>
        This background is green
      </td>
    </tr>
  </table>
</body>
</html>

```

زمینه ی HTML با تصاویر

ویژگی **background** همچنین می تواند برای کنترل زمینه ی یک عنصر **HTML**، به ویژه بدنه ی صفحه و زمینه های جدول، استفاده شود. شما می توانید یک تصویر را به عنوان زمینه ی صفحه و یا جدول **HTML** خود استفاده کنید. در زیر ترکیب استفاده از ویژگی **background** را با هر عنصر **HTML** می بینید.

توجه ویژگی **background** به عنوان ویژگی خوبی تلقی نمی شود و توصیه می شود از **style sheet** برای تنظیم زمینه استفاده کنید.

```
<tagname background="Image URL" ...>
```

متداول ترین فرمت های مورد استفاده ی تصویر عبارتند از **JPEG**، **PNG** و **GIF**.

مثال

در اینجا مثال هایی را از تنظیم تصویر به عنوان زمینه ی جدول مشاهده می کنید.

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Background Images</title>
</head>
<body>
  <!-- Set table background -->
  <table background="/images/html.jpg" width="100%" height="400">
<tr>
  <td>
    This background is filled up with HTML image.
  </td>
</tr>
</table>
</body>
</html>
```

زمینه های طرح دار و شفاف

شما ممکن است الگوها و زمینه های شفاف بسیاری را در وب سایت های مختلف دیده باشید. این امر به راحتی و با استفاده از تصاویر طرح دار و شفاف در زمینه قابل دستیابی می باشد. توصیه می شود در هنگام ایجاد تصاویر طرح دار یا شفاف GIF یا PNG، از کوچکترین ابعاد ممکن آنها استفاده کنید، حتی به کوچکی 1x1 برای جلوگیری از بارگذاری آهسته.

مثال

در اینجا مثالی از تنظیم یک الگوی زمینه برای یک جدول را مشاهده می کنید.

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Background Images</title>
</head>
<body>
  <!-- Set a table background using pattern -->
  <table background="/images/pattern1.jpg" width="20%" height="100">
<tr>
  <td>
    This background is filled up with a pattern image.
  </td>
</tr>
</table>
  <!-- Another example on table background using pattern -->
  <table background="/images/pattern2.jpeg" width="20%" height="100">
<tr>
  <td>
    This background is filled up with a pattern image.
```

```
</td>  
</tr>  
</table>  
</body>  
</html>
```

رنگهای HTML

رنگ ها برای دادن یک ظاهر و احساس خوب به وب سایت شما بسیار مهم می باشند. شما می توانید رنگ ها را روی لایه ی صفحه با استفاده از برچسب **<body>** مشخص کنید، یا می توانید رنگ ها را برای برچسب های مجزا با استفاده از ویژگی **bgcolor** مشخص کنید.

برچسب **<body>** دارای ویژگی های زیر می باشد که می تواند برای تنظیم رنگ های مختلف استفاده شود.

bgcolor رنگی را برای زمینه ی صفحه تنظیم می کند.

text رنگی را برای متن تنظیم می کند.

alink رنگی را برای لینک های فعال یا انتخاب شده تنظیم می کند.

link رنگی را برای متن لینک شده تنظیم می کند.

رنگی را برای لینک های بازدید شده تنظیم می کند۔ یعنی برای متن لینک شده که روی آن کلیک کرده اید.

روش های کدگذاری رنگ HTML

در زیر سه روش متفاوت برای تنظیم رنگ در صفحه ی وب خود مشاهده می کنید

Color names – می توانید نام رنگ ها را به طور مستقیم تعیین کنید، به عنوان **مثال** سبز، قرمز، آبی و غیره.

Hex codes – یک کد شش رقمی که نشان دهنده ی مقدار آبی، قرمز و سبز سازنده ی رنگ، می باشد.

Color decimal or percentage values – این مقدار با استفاده از ویژگی **rgb()** تعیین می شود.




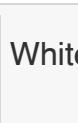


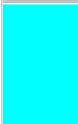









اکنون این روش های رنگ گذاری را یکی یکی بررسی خواهیم کرد.

رنگ های HTML – نام رنگ

می توانید برای تنظیم رنگ متن یا زمینه به طور مستقیم از نام رنگ استفاده کنید. **W3C** لیستی از **16** رنگ پایه را دارد که توسط یک اعتبار سنج **HTML** ارزیابی می شود، اما نام بیشتر از **200** رنگ مختلف وجود دارد که توسط مرورگرهای مهم پشتیبانی می شوند.

16 رنگ استاندارد W3C

در اینجا نام **16** رنگ استاندارد **W3C** را مشاهده می کنید، پیشنهاد می شود که از آنها استفاده کنید.

	Black		Gray		Silver		White
	Yellow		Lime		Aqua		Fuchsia
	Red		Green		Blue		Purple
	Maroon		Olive		Navy		Teal

مثال


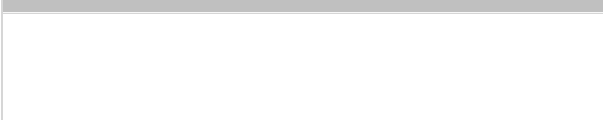
در اینجا **مثال** هایی را می بینید از تنظیم زمینه ی یک برچسب **HTML** به وسیله ی نام رنگ.

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Colors by Name</title>
</head>
<body text="blue" bgcolor="green">
  <p>Use different color names for for body and table and see the result.</p>
  <table bgcolor="black">
    <tr>
      <td>
        <font color="white">This text will appear white on black background.</font>
      </td>
    </tr>
  </table>
</body>
</html>
```

رنگ های HTML – کدهای Hex

هگزادسیمال یک نمایش 6 رقمی از یک رنگ می باشد. دو رقم اول یعنی **RR** نمایانگر رنگ قرمز (**Red**) می باشند، دو رقم بعدی (**GG**) مقدار سبز (**green**) را نشان می دهند و آخرین دو رقم (**BB**) نیز مقدار آبی (**blue**) را نشان می دهند.

هر کد هگزادسیمال به وسیله ی یک علامت **#** دنبال می شود. در ادامه رنگ های مورد استفاده در نشانه گذاری هگزادسیمال را مشاهده می کنید.

Color	Color HEX
	#000000
	#FF0000
	#00FF00
	#0000FF
	#FFFF00
	#00FFFF
	#FF00FF
	#C0C0C0
	#FFFFFF

در زیر مثال هایی را می بینید از تنظیم زمینه ی یک برچسب **HTML** به وسیله ی کد رنگ در هگزادسیمال.

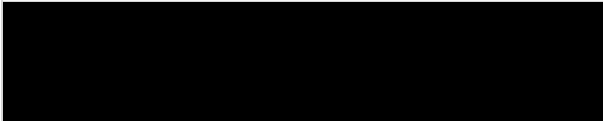


```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Colors by Hex</title>
</head>
<body text="#0000FF" bgcolor="#00FF00">
  <p>Use different color hexa for for body and table and see the result.</p>
  <table bgcolor="#000000">
    <tr>
      <td>
        <font color="#FFFFFF">This text will appear white on black background.</font>
      </td>
    </tr>
  </table>
</body>
/html>>
```







رنگ های HTML – مقادیر RGB

مقدار این رنگ با استفاده از ویژگی (**rgb**) مشخص می شود. این ویژگی سه مقدار می گیرد که هر کدام برای سبز، قرمز و آبی می باشد. مقدار می تواند عددی بین **0** و **255** یا یک درصد باشد.

توجه همه ی مرورگرها ویژگی (**rgb**) را پشتیبانی نمی کنند، بنابراین توصیه می شود از آن استفاده نکنید.

در زیر لیستی از رنگ ها با مقادیر **RGB** را مشاهده می کنید.

Color	Color RGB
	rgb(0,0,0)
	rgb(255,0,0)
	rgb(0,255,0)

	rgb(0,0,255)
	rgb(255,255,0)
	rgb(0,255,255)
	rgb(255,0,255)
	rgb(192,192,192)
	rgb(255,255,255)

در اینجا مثال هایی را مشاهده می کنید از تنظیم زمینه ی یک برچسب HTML به وسیله ی برچسب rgb() با کد رنگ.

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Colors by RGB code</title>
</head>
<body text="rgb(0,0,255)" bgcolor="rgb(0,255,0)">
  <p>Use different color code for for body and table and see the result.</p>
  <table bgcolor="rgb(0,0,0)">
    <tr>
      <td>
        <font color="rgb(255,255,255)">This text will appear white on black background.</font>
      </td>
    </tr>
  </table>
</body>
/html>>
```

در زیر لیستی از 216 رنگ را مشاهده می کنید که تصور می شود سالم ترین و مستقل ترین رنگ های کامپیوتر باشند. این رنگ ها از کد هگزای 000000 تا FFFFFFFF متفاوت می باشند و توسط همه ی کامپیوترهای دارای پالت رنگ 256 تایی پشتیبانی می شوند.

000000	000033	000066	000099	0000CC	0000FF
003300	003333	003366	003399	0033CC	0033FF
006600	006633	006666	006699	0066CC	0066FF
009900	009933	009966	009999	0099CC	0099FF
00CC00	00CC33	00CC66	00CC99	00CCCC	00CCFF
00FF00	00FF33	00FF66	00FF99	00FFCC	00FFFF
330000	330033	330066	330099	3300CC	3300FF
333300	333333	333366	333399	3333CC	3333FF
336600	336633	336666	336699	3366CC	3366FF
339900	339933	339966	339999	3399CC	3399FF
33CC00	33CC33	33CC66	33CC99	33CCCC	33CCFF
33FF00	33FF33	33FF66	33FF99	33FFCC	33FFFF
660000	660033	660066	660099	6600CC	6600FF
663300	663333	663366	663399	6633CC	6633FF

666600	666633	666666	666699	6666CC	6666FF
669900	669933	669966	669999	6699CC	6699FF
66CC00	66CC33	66CC66	66CC99	66CCCC	66CCFF
66FF00	66FF33	66FF66	66FF99	66FFCC	66FFFF
990000	990033	990066	990099	9900CC	9900FF
993300	993333	993366	993399	9933CC	9933FF
996600	996633	996666	996699	9966CC	9966FF
999900	999933	999966	999999	9999CC	9999FF
99CC00	99CC33	99CC66	99CC99	99CCCC	99CCFF
99FF00	99FF33	99FF66	99FF99	99FFCC	99FFFF
CC0000	CC0033	CC0066	CC0099	CC00CC	CC00FF
CC3300	CC3333	CC3366	CC3399	CC33CC	CC33FF
CC6600	CC6633	CC6666	CC6699	CC66CC	CC66FF
CC9900	CC9933	CC9966	CC9999	CC99CC	CC99FF

CCCC00	CCCC33	CCCC66	CCCC99	CCCCCC	CCCCFF
CCFF00	CCFF33	CCFF66	CCFF99	CCFFCC	CCFFFF
FF0000	FF0033	FF0066	FF0099	FF00CC	FF00FF
FF3300	FF3333	FF3366	FF3399	FF33CC	FF33FF
FF6600	FF6633	FF6666	FF6699	FF66CC	FF66FF
FF9900	FF9933	FF9966	FF9999	FF99CC	FF99FF
FFCC00	FFCC33	FFCC66	FFCC99	FFCCCC	FFCCFF
FFFF00	FFFF33	FFFF66	FFFF99	FFFFCC	FFFFFF

فونت ها

فونت ها نقش مهمی در زیبا ساختن و خواناتر کردن وب سایت بازی می کنند. ظاهر و رنگ فونت به طور کامل بستگی به کامپیوتر و مرورگری دارد که استفاده می شود، اما شما می توانید از برچسب `` در HTML برای افزودن استایل، سایز و رنگ به متن خود در وب سایت استفاده کنید. می توانید از یک برچسب `<basefont>` برای تنظیم تمام متن خود به اندازه، ظاهر و رنگ یکسان استفاده کنید.

برچسب فونت دارای سه ویژگی به نام های `color`، `size` و `face` می باشد که فونت شما را به دلخواه در می آورد. برای تغییر هر کدام از ویژگی های فونت در هر زمانی در صفحه ی وب خود، به سادگی از برچسب `` استفاده کنید. متنی که دنبال می کند، تغییر یافته باقی می ماند تا زمانی که شما آن را با `` برچسب ببندید. شما می توانید یکی از ویژگی ها یا همه ی ویژگی های داخل برچسب `` را تغییر دهید.

نکته

برچسب های **font** و **basefont** استفاده نمی شوند و احتمال می رود که در ورژن های بعدی **HTML** حذف شوند. بنابراین نباید مورد استفاده قرار بگیرند، پیشنهاد می شود که برای اجرای فونت های خود از استایل های **CSS** استفاده کنید. اما برای رسیدن به هدف این فصل برچسب های **font** و **basefont** را با جزئیات توضیح می دهد.

تنظیم اندازه ی فونت

شما می توانید با استفاده از ویژگی **size** اندازه فونت محتوا را تنظیم کنید. دامنه ی مقادیر قابل قبول از 1 (کوچکترین) تا 7 (بزرگترین) می باشد. اندازه ی فونت پیش فرض 3 می باشد.

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Setting Font Size</title>
</head>
<body>
  <font size="1">Font size="1"</font><br />
  <font size="2">Font size="2"</font><br />
  <font size="3">Font size="3"</font><br />
  <font size="4">Font size="4"</font><br />
  <font size="5">Font size="5"</font><br />
  <font size="6">Font size="6"</font><br />
  <font size="7">Font size="7"</font>
</body>
</html>
```

اندازه ی فونت مربوط

شما می توانید مشخص کنید چه تعداد از فونت ها بزرگتر و چه تعداد کوچکتر از اندازه ی فونت حاضر باشند. می توانید آن را مانند **** or **** مشخص کنید.

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Relative Font Size</title>
</head>
<body>
  <font size="-1">Font size="-1"</font><br />
  <font size="+1">Font size="+1"</font><br />
```

```
<font size="+2">Font size="+2"</font><br />  
<font size="+3">Font size="+3"</font><br />  
<font size="+4">Font size="+4"</font>  
</body>  
</html>
```

تنظیم ظاهر فونت

شما می توانید با استفاده از ویژگی **face** ظاهر فونت را تنظیم کنید، اما باید بدانید که اگر کاربر بازدیدکننده ی صفحه، فونت را نصب نکرده باشد، قادر به دیدن آن نخواهد بود. در عوض کاربر ظاهر فونت پیش فرض را می بیند که برای کامپیوترش مناسب می باشد.

مثال

```
<!DOCTYPE html>  
<html>  
<head>  
<title>Font Face</title>  
</head>  
<body>  
<font face="Times New Roman" size="5">Times New Roman</font><br />  
<font face="Verdana" size="5">Verdana</font><br />  
<font face="Comic sans MS" size="5">Comic Sans MS</font><br />  
<font face="WildWest" size="5">WildWest</font><br />  
<font face="Bedrock" size="5">Bedrock</font><br />  
</body>  
</html>
```

تعیین ظاهر فونت جایگزین

یک بازدید کننده فقط قادر خواهد بود فونت شما را ببیند، اگر آن را نصب شده روی کامپیوتر خود داشته باشد. بنابراین امکان تعیین دو یا بیشتر از دو ظاهر جایگزین با ارائه ی نام های مربوط به فونت ها وجود دارد.

```
<font face="arial, helvetica">  
<font face="Lucida Calligraphy, Comic Sans MS, Lucida Console">
```

اگر هیچکدام از فونت های ارائه شده نصب نشده باشند، بنابراین فونت پیش فرض *Times New Roman* نمایش داده خواهد شد.

تنظیم رنگ فونت

شما می توانید با استفاده از ویژگی color هر رنگی را برای فونت تنظیم کنید. شما می توانید رنگ مورد نظر خود را یا با استفاده از نام رنگ و یا با استفاده از کد هگزادسیمال برای آن رنگ تعیین کنید.

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Setting Font Color</title>
</head>
<body>
  <font color="#FF00FF">This text is in pink</font><br />
  <font color="red">This text is red</font>
</body>
</html>
```

عنصر <basefont>

انتظار می رود که عنصر <basefont> یک اندازه، رنگ و ظاهر پیش فرض برای هر بخشی از داکيومنت که در برچسب قرار نمی گیرند، تنظیم کند. شما می توانید از عناصر استفاده کنید تا تنظیمات <basefont> را انجام دهید.

برچسب <basefont> همچنین ویژگی های رنگ، اندازه و ظاهر را می گیرد و با دادن مقدار بیشتر از +1 برای فونت های بزرگتر و کمتر از -2 برای فونت های کوچکتر، تنظیمات فونت مربوطه را پشتیبانی می کند.

مثال

```
<!DOCTYPE html>
<html>
<head>
  <title>Setting Basefont Color</title>
</head>
<body>
  <basefont face="arial, verdana, sans-serif" size="2" color="#ff0000">
  <p>This is the page's default font.</p>
  <h2>Example of the <basefont&gt; Element</h2>
  <p>
    <font size="+2" color="darkgray">
      This is darkgray text with two sizes larger
    </font>
  </p>
  <p>
```



```
<font face="courier" size="-1" color="#000000">
  It is a courier font, a size smaller and black in color.
</font>
</p>
</body>
</html>
```

فرم های html

وقتی که می خواهید داده هایی را از سایت بازدیدکننده جمع آوری کنید، به فرم های **HTML** احتیاج خواهید داشت. به عنوان مثال در هنگام ثبت کاربر، اطلاعاتی مانند نام، آدرس ایمیل و کارت اعتباری و غیره را جمع آوری می کنید. یک فرم داده ها را از بازدیدکننده ی سایت می گیرد و سپس آن را به یک برنامه ی پایانی مانند **CGI**، اسکریپت **ASP** یا اسکریپت **PHP** باز می گرداند. برنامه ی پایانی فرایندهای مورد نیاز را بر اساس کار تعریف شده در برنامه، روی داده ی منتقل شده انجام می دهد.

عناصر متفاوتی برای فرم در دسترس میباشند، مانند فیلدهای متن، فیلدهای **textarea**، منوهای رو به پایین، دکمه های رادیو، چک باکس ها و غیره.

برچسب **<form>** مربوط به **HTML**، برای ایجاد یک فرم **HTML** استفاده می شود و دارای ترکیب زیر می باشد.

```
<form action="Script URL" method="GET|POST">
  form elements like input, textarea etc.
</form>
```

ویژگی های فرم

علاوه بر ویژگی های متداول، در زیر لیستی از متداول ترین ویژگی های مربوط به فرم را مشاهده می کنید .

Attribute	Description
action	اسکریپت backend برای پردازش داده ی انتقال شده ی شما آماده میباشد.

method	روش‌های بارگذاری داده. متداولترین روش‌های مورد استفاده GET و POST می‌باشند.
target	پنجره‌ی هدف یا چارچوب را، جایی که نتیجه‌ی اسکریپت نمایش داده خواهد شد، مشخص می‌کند. این ویژگی مقادیر <code>_parent</code> , <code>_self</code> , <code>_blank</code> و غیره را به خود می‌گیرد.
enctype	<p>You can use the enctype attribute to specify how the browser encodes the data before it sends it to the server. Possible values are</p> <p>شما می‌توانید از ویژگی <code>enctype</code> برای مشخص کردن چگونگی برنامه نویسی توسط مرورگر قبل از ارسال به سرور، استفاده کنید. مقادیر ممکن عبارتند از</p> <p><code>application/x-www-form-urlencoded</code> – این روش استاندارد است که بیشتر فرم‌ها در سناریوهای ساده استفاده می‌کنند.</p> <p><code>multipart/form-data</code> – این روش زمانی استفاده می‌شود که شما می‌خواهید داده‌های دوتایی را به شکل فایل‌هایی مانند تصویر بارگذاری کنید.</p>

کنترل‌های فرم HTML

انواع مختلفی از کنترل‌های فرم وجود دارند که می‌توانید برای جمع‌آوری داده با استفاده از فرم **HTML** از آنها استفاده کنید.

کنترل‌های ورودی متن

کنترل‌های چک باکس

کنترل های رادیو باکس

کنترل های انتخاب باکس

باکس های انتخاب فایل

کنترل های مخفی

دکمه های قابل کلیک شدن

ثبت و تنظیم مجدد

کنترل های ورودی متن

سه نوع ورودی متن وجود دارد که در فرم ها استفاده می شوند .

کنترل های ورودی متن تک خطی این کنترل برای آیتم هایی استفاده می شود که فقط یک خط از ورودی

کاربر را لازم دارد، مانند باکس های جستجو و نام ها. این کنترل ها با استفاده از برچسب `<input>`

مربوطه **HTML** ایجاد می شوند.

کنترل ورودی گذرواژه این نیز یک متن ورودی تک خطی می باشد، اما به محض اینکه کاربر کاراکتر را وارد

می کند، این کنترل آن را می پوشاند.

کنترل های متن ورودی چند خطی زمانی استفاده می شود که لازم است کاربر جزئیاتی را وارد کند که بیشتر

از یک جمله می باشند. کنترل های ورودی چند خطی با استفاده از برچسب `<textarea>` استفاده می

شوند.

کنترل های ورودی تک خطی

این کنترل ها برای آیتم هایی استفاده می شوند که کاربر فقط یک خط ورودی احتیاج دارد، مانند باکس های

جستجو و نام ها و با استفاده از برچسب `<input>` مربوط به **HTML** ایجاد می شوند.

مثال

در اینجا مثال پایه ای می بینید از یک ورودی تک خطی که برای گرفتن نام کوچک و نام خانوادگی به کار می رود .

```
<!DOCTYPE html>
<html>
<head>
  <title>Text Input Control</title>
</head>
<body>
  <form>
    First name <input type="text" name="first_name" />
    <br>
    Last name <input type="text" name="last_name" />
  </form>
</body>
</html>
```

ویژگی ها

در زیر لیستی از ویژگی های برچسب **<input>** را برای ایجاد فیلد متن می بینید.

Attribute	Description
type	نوع کنترل ورودی را نشان می دهد و برای کنترل متن ورودی برای text تنظیم خواهد شد.
name	برای دادن نام به کنترلی استفاده می شود که قرار است برای تشخیص به سرور ارسال شود و مقدار بگیرد.
value	می تواند برای ارائه ی یک مقدار اصلی در داخل کنترل استفاده شود.
size	اجازه می دهد تا عرض کنترل متن ورودی را متناسب با کاراکترها تعیین کنید.
maxlength	اجازه می دهد تا حداکثر تعداد کاراکترهایی را که یک کاربر می تواند در یک تکست باکس وارد کند، مشخص کنید.

کنترل های پسورد ورودی

این کنترل نیز یک کنترل تک خطی می باشد، اما به محض اینکه کاربر کاراکترها را وارد می کند، آنها را می پوشاند. این ها نیز با استفاده از برچسب `<input>` مربوط به HTML ایجاد می شوند، اما نوع ویژگی با عنوان `password` تنظیم میشود.

مثال

در اینجا مثالی از ورودی تک خطی پسورد می بینید که برای گرفتن پسورد کاربر استفاده می شود.

```
<!DOCTYPE html>
<html>
<head>
  <title>Password Input Control</title>
</head>
<body>
  <form>
    User ID <input type="text" name="user_id" />
    <br>
    Password <input type="password" name="password" />
  </form>
</body>
</html>
```

ویژگی ها

در زیر لیست مربوط به ویژگی های برچسب `<input>` را برای ایجاد فیلد پسورد می بینید.

Attribute	Description
type	نوع کنترل ورودی را نشان می دهد و برای کنترل ورودی پسورد، برای <code>password</code> تنظیم خواهد شد.
name	برای نام گذاری کنترلی استفاده می شود که برای تشخیص و گرفتن مقدار به سرور فرستاده می شود.

value	برای ارائه ی یک مقدار اولیه در داخل کنترل استفاده می شود.
size	اجازه می دهد تا عرض کنترل متن ورودی را با کاراکترها تعیین کنید.
maxlength	اجازه می دهد تا حداکثر تعداد کاراکترهایی را که یک کاربر می تواند در یک تکست باکس وارد کند، تعیین کنید.

کنترل های متن ورودی چند خطی

زمانی استفاده می شود که یک کاربر باید جزئیاتی را وارد کند که بیشتر از یک جمله می باشند. کنترل های ورودی چند خطی با استفاده از برچسب `<textarea>` ایجاد میشوند.

مثال

در اینجا مثالی را می بینید از یک ورودی متن چند خطی که برای ارائه ی توصیفات آیتم استفاده می شود.

```
<!DOCTYPE html>
<html>
<head>
  <title>Multiple-Line Input Control</title>
</head>
<body>
  <form>
    Description <br />
    <textarea rows="5" cols="50" name="description">
      Enter description here...
    </textarea>
  </form>
</body>
</html>
```

ویژگی ها

در زیر لیستی از ویژگی های برچسب `<textarea>` ارائه شده اند.

Attribute	Description
name	برای نامگذاری کنترل استفاده می شود که به سرور ارسال می شود تا تشخیص داده شده و مقدار بگیرد.
rows	تعداد ردیف های area box مربوط به متن را نشان می دهد.
cols	تعداد ستون های area box مربوط به متن را نشان می دهد.

کنترل چک باکس

چک باکس ها زمانی استفاده می شوند که بیشتر از یک گزینه قرار است انتخاب شود. آنها نیز با استفاده از برچسب `<input>` ایجاد می شوند، اما نوع ویژگی به `checkbox` تنظیم می شود.

مثال

در اینجا مثالی از کد HTML را مشاهده می کنید برای یک فرم با دو چک باکس.

```

<!DOCTYPE html>
<html>
<head>
  <title>Checkbox Control</title>
</head>
<body>
  <form>
    <input type="checkbox" name="maths" value="on"> Maths
    <input type="checkbox" name="physics" value="on"> Physics
  </form>
</body>
</html>

```

در زیر لیستی از ویژگی های برچسب `<checkbox>` را می بینید

Attribute	Description
type	نوع کنترل ورودی را نشان می دهد و برای کنترل ورودی چک باکس با checkbox کنترل خواهد شد.
name	برای نامگذاری کنترلی استفاده می شود که برای تشخیص و گرفتن مقدار به سرور ارسال می شود.
value	مقداری که اگر چک باکس انتخاب شود، استفاده خواهد شد.
checked	اگر بخواهید آن را به طور پیش فرض انتخاب کنید، با checked تنظیم می شود.

کنترل دکمه ی رادیو

دکمه های رادیو زمانی استفاده میشوند که بین چندین گزینه تنها یک گزینه باید انتخاب شود. این کنترل ها نیز با برچسب `<input>` ایجاد میشوند، اما نوع ویژگی با `radio` تنظیم می شود.

مثال

در اینجا مثالی از کد HTML می بینید برای یک فرم با دو دکمه ی رادیو.

```
<!DOCTYPE html>
<html>
<head>
  <title>Checkbox Control</title>
</head>
<body>
  <form>
    <input type="radio" name="maths" value="on"> Maths
    <input type="radio" name="physics" value="on"> Physics
  </form>
</body>
</html>
```

ویژگی ها

در زیر لیستی از ویژگی های دکمه ی رادیو را می بینید.

Attribute	Description
type	نوع کنترل ورودی را نشان می دهد و برای کنترل ورودی چک باکس با عنوان radio تنظیم می شود.
name	برای نامگذاری کنترلی استفاده می شود که برای تشخیص و گرفتن مقدار به سرور ارسال می شود.
value	مقداری که radio box انتخاب شود، استفاده خواهد شد.
checked	اگر می خواهید آن را به عنوان پیش فرض استفاده کنید، به checked تنظیم کنید.

کنترل Select Box

یک **Select Box** که منوی رو به پایین نیز نامیده می شود، گزینه ای را برای ارائه ی گزینه های مختلف به شکل یک لیست رو به پایین ارائه می دهد، که کاربر می تواند از آن یک یا بیشتر از یک گزینه را انتخاب کند.

مثال

در اینجا مثالی از کد **HTML** برای یک فرم با یک جعبه ی رو به پایین می بینید.

```

<!DOCTYPE html>
<html>
<head>
  <title>Select Box Control</title>
</head>
<body>
  <form>
    <select name="dropdown">
      <option value="Maths" selected>Maths</option>
      <option value="Physics">Physics</option>
    </select>
  </form>
</body>
</html>

```

```
</form>
</body>
</html>
```

در زیر لیستی از ویژگی‌های برچسب `<select>` را می‌بینید.

Attribute	Description
name	برای نامگذاری کنترلی استفاده می‌شود که برای تشخیص و گرفتن مقدار به سرور انتقال می‌شود.
size	برای نمایش یک لیست باکس اسکروالینگ استفاده می‌شود.
multiple	اگر روی <code>multiple</code> تنظیم شده باشد، به کاربر اجازه می‌دهد تا چند آیتم را از منو انتخاب کند.

در زیر لیستی از ویژگی‌های مهم برچسب `<option>` ارائه شده است.

Attribute	Description
value	اگر در سلکت باکس یک گزینه انتخاب شده باشد، این مقدار استفاده خواهد شد.
selected	مشخص می‌کند که این گزینه رد زمان بارگذاری صفحه، باید گزینه‌ی انتخابی باشد.
label	یک روش جایگزین برای برچسب دار کردن گزینه‌ها.

فایل آپلود باکس

اگر می خواهید به کاربر اجازه دهید تا فایلی را روی وب سایت شما آپلود کند، به یک کنترل فایل آپلود احتیاج خواهید داشت، که **Select Box** نیز نامیده می شود. این ویژگی با استفاده از برچسب **<input>** نیز ایجاد می شود، اما نوع ویژگی با **file** تنظیم می شود.

مثال

در اینجا مثالی را می بینید از کد **HTML** برای یک فرم با یک فایل آپلود باکس.

```
<!DOCTYPE html>
<html>
<head>
  <title>File Upload Box</title>
</head>
<body>
  <form>
    <input type="file" name="fileupload" accept="image/*" />
  </form>
</body>
</html>
```

ویژگی ها

در زیر لیستی از ویژگی های فایل آپلود باکس را می بینید.

Attribute	Description
name	برای نامگذاری کنترلی استفاده می شود که برای تشخیص و گرفتن مقدار به سرور انتقال می شود.
accept	نوع فایل هایی را که سرور می پذیرد، تعیین می کند.

کنترل های دکمه

راه های مختلفی برای ایجاد دکمه های قابل کلیک شدن وجود دارد. شما با استفاده از برچسب **<input>** و با تنظیم نوع ویژگی آن به **button** دکمه های قابل کلیک شدن ایجاد کنید. نوع ویژگی می تواند مقادیر زیر را بگیرد.

Type	Description
submit	دکمه ای را ایجاد می کند که به طور خودکار یک فرم را می پذیرد.
reset	دکمه ای را ایجاد می کند که به طور خودکار کنترل های فرم را به مقادیر اولیه ی خود بازمی گرداند.
button	دکمه ای را ایجاد می کند که برای اجرای اسکریپت کاربر، زمانی که کاربر آن دکمه را کلیک می کند، استفاده می شود.
image	یک دکمه ی قابل کلیک شدن ایجاد می کند، اما ما می توانیم از یک تصویر به عنوان زمینه ی دکمه استفاده کنیم.

مثال

در اینجا مثالی از کد HTML برای یک فرم با سه نوع دکمه را می بینید.

```

<!DOCTYPE html>
<html>
<head>
  <title>File Upload Box</title>
</head>
<body>
  <form>
    <input type="submit" name="submit" value="Submit" />
    <input type="reset" name="reset" value="Reset" />
    <input type="button" name="ok" value="OK" />
  </form>

```

```
<input type="image" name="imagebutton" src="/html/images/logo.png" />
</form>
</body>
</html>
```

کنترل های مخفی شده ی فرم

کنترل های مخفی شده فرم برای پنهان کردن داده در داخل صفحه ای استفاده می شوند که بعدا می توانند به سرور ارسال شوند. این کنترل در داخل کد مخفی شده و روی صفحه ی حقیقی ظاهر نمی شود. برای مثال فرم مخفی شده ی زیر برای حفظ شماره ی صفحه ی جاری استفاده می شود. وقتی که کاربر روی **next page** کلیک کند، مقدار کنترل مخفی شده به سرور وب ارسال شده و در آنجا تصمیم خواهد گرفت که بر اساس انتقال صفحه ی جاری، کدام صفحه نمایش داده خواهد شد.

مثال

در اینجا مثالی از کد **HTML** برای نمایش کاربرد کنترل مخفی شده می بینید.

```
<!DOCTYPE html>
<html>
<head>
<title>File Upload Box</title>
</head>
<body>
<form>
<p>This is page 10</p>
<input type="hidden" name="pagename" value="10" />
<input type="submit" name="submit" value="Submit" />
<input type="reset" name="reset" value="Reset" />
</form>
</body>
</html>
```

آموزش چند رسانه ای در HTML

گاهی اوقات تمایل دارید که موسیقی یا ویدئو به صفحه ی وب خود اضافه کنید. ساده ترین راه برای افزودن صدا یا ویدئو به وب سایت، وارد کردن برچسب خاص **HTML** به نام **<embed>** می باشد. این برچسب باعث می شود که مرورگر به طور خودکار کنترل هایی را برای مولتی مدیا وارد کند که مرورگر ارائه شده برچسب **<embed>** و نوع مدیای ارائه شده را پشتیبانی می کند.

همچنین می توانید یک برچسب **<noembed>** برای مرورگرهایی وارد کنید که برچسب **<embed>** را نمی شناسند. به عنوان **مثال** می توانید از **<embed>** برای نمایش یک فیلم به انتخاب خود استفاده کنید و اگر مرورگر **<embed>** را پشتیبانی نمی کند، از

<noembed>، برای نمایش یک تصویر **JPG** مجزا استفاده کنید.

مثال

در اینجا **مثال** ساده ای می بینید از اجرای یک فایل جاسازی شده ی **.midi**.

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML embed Tag </title>
</head>
<body>
  <embed src="/html/yourfile.mid" width="100%" height="60">
  <noembed></noembed>
</embed>
</body>
</html>
```

ویژگی های برچسب **<embed>**

در زیر لیستی از ویژگی های مهمی را می بینید که توسط برچسب **<embed>** استفاده می شود.

Attribute	Description
align	تعیین می کند که چگونه یک آبجکت را تنظیم کرد، که می تواند هم در مرکز، هم در راست و یا چپ تنظیم شود.
autostart	این ویژگی Boolean نشان می دهد که آیا مدیا باید به طور خودکار شروع به کار کند. شما می توانید آن را با true یا false تنظیم کنید.

loop	تعیین می کند که آیا صدا متداولا پشت سرهم تکرار شود (لوپ را روی true تنظیم کنید)، یا چند دفعه باید تکرار شود (یک مقدار مثبت) و یا اصلا تکرار نشود (false).
playcount	تعداد دفعاتی را که یک صدا باید تکرار شود تعیین می کند. اگر در حال استفاده از IE هستید، این گزینه جایگزینی برای loop می باشد.
hidden	مشخص می کند که آیا آبجکت مولتی مدیا باید روی صفحه نمایش داده شود. مقدار false یعنی نه و مقدار true یعنی بله.
width	عرض آبجکت به پیکسل.
height	عرض آبجکت به پیکسل.
name	نامی که برای اشاره به آبجکت استفاده می شود.
src	URL مربوط به آبجکت که باید جاسازی شود.
volume	میزان صدا را کنترل می کند که می تواند از 0 (صدا قطع شده) تا 100 (آخرین حد صدا) باشد.

انواع ویدیوهای پشتیبانی شده

می توانید از انواع مدیاهای مختلفی مانند فایل های **flash movies**، **AVI** و **MOV** در داخل برچسب **embed** استفاده کنید.

فایل های **swf** – فایل هایی هستند که با برنامه ی **macromedia's flash** تولید می شوند.

فایل های **wmv** – انواع فایل های تصویری ویندوز میکروسافت می باشند.

فایل های mov – فرمت Quick time movie در اپل.

فایل های mpeg – فایل های تصویری هستند که توسط گروه تخصصی تصاویر متحرک (Moving Pictures Expert Group) ایجاد می شوند.

```
<!DOCTYPE html>
<html>
<head>
<title>HTML embed Tag</title>
</head>
<body>
<embed src="/html/yourfile.swf" width="200" height="200">
<noembed></noembed>
</embed>
</body>
</html>
```

صدای زمینه

شما می توانید از برچسب **<bgsound>** در **html** برای اجرای صدا در زمینه ی صفحه ی وب خود استفاده کنید. این برچسب فقط توسط اینترنت اکسپلورر پشتیبانی می شود و دیگر مرورگرها آن را نادیده می گیرند. زمانی که سند اصلی در ابتدا توسط کاربر دانلود شده و نمایش داده می شود، این برچسب یک فایل صدا را دانلود کرده و اجرا می کند. همچنین هر وقت کاربر مرورگر را تازه سازی کند، صدای زمینه نیز دوباره اجرا خواهد شد.

این برچسب دارای فقط دو ویژگی می باشد، **loop** و **src**، که همانطور که در بالا توضیح داده شد هر دوی این ویژگی ها دارای یک معنا می باشند.

در اینجا **مثال** ساده ای از اجرای فایل کوچک **midi** را می بینید.

```
<!DOCTYPE html>
<html>
<head>
<title>HTML embed Tag</title>
</head>
<body>
<bgsound src="/html/yourfile.mid">
<noembed></noembed>
</bgsound>
</body>
</html>
```


این مثال یک صفحه ی خالی تولید خواهد کرد. این برچسب هیچ مولفه ای را اجرا نمی کند و مخفی باقی می ماند.

اینترنت اکسپلورر نیز تنها سه فرمت صدا را اجرا می کند **wav** فرمت داخلی کامپیوترها، **au** فرمت داخلی برای کار ابزارهای **Unix** و **MIDI** یک کد برنامه گذاری جهانی برای موسیقی.

برچسب آبجکت در HTML

HTML 4 عنصر **<object>** را معرفی می کند که یک راه حل چند منظوره برای وارد کردن آبجکت عمومی ارائه می دهد. عنصر **<object>** به نویسندگان **HTML** اجازه می دهد تا هر چیز لازم را با یک آبجکت برای ارائه ی آن توسط یک کاربر، تعیین کند. در اینجا چند مثال در این رابطه می بینید.

مثال 1

شما می توانید یک سند **HTML** را در خود سند **HTML** اجرا کنید.

```
<object data="data/test.htm" type="text/html" width="300" height="200">  
  alt <a href="data/test.htm">test.htm<a>  
</object>
```

در اینجا اگر مرورگر برچسب **object** را پشتیبانی نکند، ویژگی **alt** وارد تصویر می شود.

مثال 2

شما می توانید یک سند **PDF** را در یک سند **HTML** اجرا کنید.

```
<object data="data/test.pdf" type="application/pdf" width="300" height="200">  
  alt <a href="data/test.pdf">test.htm<a>  
</object>
```

مثال 3

شما می توانید با استفاده از برچسب **<param>** برخی پارامترهای متناسب با سند را مشخص کنید. در اینجا مثالی از اجرای فایل **wav** را می بینید.

```
<object data="data/test.wav" type="audio/x-wav" width="200" height="20">
  <param name="src" value="data/test.wav">
  <param name="autoplay" value="false">
  <param name="autoStart" value="0">
  alt <a href="data/test.wav">test.wav<a>
</object>
```

مثال 4

شما می توانید یک سند **flash** مانند زیر اجرا کنید.

```
<object classid="clsidD27CDB6E-AE6D-11cf-96B8-444553540000" id="penguin"
  codebase="someplace/swflash.cab" width="200" height="300">
  <param name="movie" value="flash/penguin.swf" />
  <param name="quality" value="high" />
  
</object>
```

مثال 5

شما می توانید یک سند **java applet** وارد سند **HTML** کنید.

```
<object classid="clsid8ad9c840-044e-11d1-b3e9-00805f499d93"
  width="200" height="200">
  <param name="code" value="applet.class">
</object>
```

ویژگی **classid** تعیین می کند که چه نسخه ای از **java plug-in** استفاده شود. شما می توانید از ویژگی انتخابی **codebase** برای مشخص کردن چگونگی دانلود **JRE** استفاده کنید.

آموزش HTML Marquee

Marquee در **HTML** یک قطعه اسکرولینگ می باشد که یا به صورت افقی در عرض متن و یا به صورت عمودی در پایین صفحه ی وب شما نمایش داده می شود، بستگی به تنظیمات دارد و توسط برچسب **<Fmarquees>** مربوط به **HTML** ایجاد می شود.

توجه ممکن است برچسب `<marquees>` در HTML توسط مرورگرهای زیادی پشتیبانی نشود، بنابراین توصیه می شود که به این برچسب تکیه نکنید، در عوض می توانید از javascript و CSS برای ایجاد چنین تاثیراتی استفاده کنید.

ترکیب ساده برای استفاده از برچسب `<marquees>` در HTML مانند زیر می باشد

```
<marquee attribute_name="attribute_value" ....more attributes>
  One or more lines or text message or image
</marquee>
```

ویژگی های برچسب `<marquee>`

در زیر لیست مهمی از ویژگی های برچسب `<marquee>` را می توانید مشاهده کنید.

Attribute	Description
width	این ویژگی عرض marquee را مشخص می کند. می تواند مقداری مانند 10 یا 20 درصد باشد.
height	این ویژگی طول marquee را مشخص می کند. می تواند مقداری مانند 10 یا 20 درصد باشد.
direction	این ویژگی مسیری را که marquee باید در آن اسکرول شود، تعیین می کند. می تواند مقداری مانند left، down، up و یا right داشته باشد.
behavior	این ویژگی نوع اسکرول marquee را تعیین می کند. می تواند مقداری مانند scroll، side و alternate داشته باشد.
scrolldelay	این ویژگی میزان تاخیر بین دو پرش را تعیین می کند. می تواند مقداری مانند 10 داشته باشد.
scrollamount	سرعت متن marquee را تعیین می کند. می تواند مقداری مانند 10 داشته باشد.

loop	تعداد دفعات loop را تعیین می کند. مقدار پیش فرض INFINITE می باشد که به این معناست که در <code>marquee</code> به طور پایان ناپذیری loop انجام می شود.
bgcolor	این ویژگی رنگ زمینه را به شکل نام رنگ یا مقدار شش تایی رنگ مشخص می کند.
hspace	این ویژگی فضای افقی اطراف <code>marquee</code> را تعیین می کند. می تواند مقداری مانند 10 یا 20 درصد داشته باشد.
vspace	این ویژگی فضای عمودی اطراف <code>marquee</code> را تعیین می کند. می تواند مقداری مانند 10 یا 20 درصد داشته باشد.

در ادامه چند مثال از استفاده ی برجسب `marquee` می بینید.

مثال 1

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML marquee Tag</title>
</head>
<body>
  <marquee>This is basic example of marquee</marquee>
</body>
</html>
```

مثال 2

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML marquee Tag</title>
</head>
<body>
  <marquee width="50%">This example will take only 50% width</marquee>
</body>
</html>
```

مثال 3

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML marquee Tag</title>
</head>
<body>
  <marquee direction="right">This text will scroll from left to right</marquee>
</body>
</html>
```

مثال 4

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML marquee Tag</title>
</head>
<body>
  <marquee direction="up">This text will scroll from bottom to up</marquee>
</body>
</html>
```

سربرگ HTML

یاد گرفتیم که یک نمونه داکيومنت HTML دارای ساختار زیر می باشد.

Document declaration tag

```
<html>
```

```
<head>
```

Document header related tags

```
<head>
```

```
<body>
```

Document body related tags

```
<body>
```

```
</html>
```

این فصل جزئیات بیشتری در مورد بخش سربرگ در HTML ارائه می دهد که به وسیله ی برچسب `<head>`

نمایش داده می شود. برچسب `<head>` حاوی برچسب های مهمی می باشد که عبارتند از `<title>`, `<meta>`,

`<script>`, `<style>`, `<base>`, `<link>` و `<noscript>` tags.

برچسب `<title>` در HTML

این برچسب برای تعیین تیتتر داکيومنت HTML استفاده می شود. در زیر مثالی می بینید از ارائه ی تیتتر به

داکيومنت HTML.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>HTML Title Tag Example</title>
<head>
<body>
  <p>Hello, World!</p>
</body>
</html>
```

این مثال نتیجه ی زیر را تولید خواهد کرد.

Hello, World!

برچسب <meta> در HTML

این برچسب برای ارائه ی متادیتا در مورد داکيومنت HTML استفاده می شود که اطلاعاتی از قبیل انقضا صفحه، گردآورنده ی صفحه، لیست کلمات کلیدی، توصیف صفحه و غیره ارائه می دهد.

در ادامه استفاده های مهم برچسب <meta> در داکيومنت HTML ارائه شده اند.

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Meta Tag Example</title>
  <!-- Provide list of keywords -->
  <meta name="keywords" content="C, C++, Java, PHP, Perl, Python">
  <!-- Provide description of the page -->
  <meta name="description" content="Simply Easy Learning by Tutorials Point">
  <!-- Author information -->
  <meta name="author" content="Tutorials Point">
  <!-- Page content type -->
  <meta http-equiv="content-type" content="text/html; charset=UTF-8">
  <!-- Page refreshing delay -->
  <meta http-equiv="refresh" content="30">
  <!-- Page expiry -->
  <meta http-equiv="expires" content="Wed, 21 June 2006 142527 GMT">
  <!-- Tag to tell robots not to index the content of a page -->
  <meta name="robots" content="noindex, nofollow">
</head>
<body>
  <p>Hello, World!</p>
</body>
</html>
```

این مثال نتیجه ی زیر را تولید می کند.

Hello, World!

برچسب <base> در HTML

این برچسب برای تعیین URL پایه برای همه ی URL های وابسته در صفحه استفاده می شود، که به این معناست که همه ی URL های دیگر هنگامی که برای آیتم ارائه شده قرار می گیرند، دیگر URL ها به URL پایه زنجیر خواهند شد.

به عنوان مثال تمام صفحات و تصاویر ارائه شده، پس از پیشوند دار کردن URL های ارائه شده با URL پایه مسیر prefixing جستجو خواهند شد.

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Base Tag Example</title>
  <base href="http://www.tahlildadeh.com/" />
</head>
<body>
  
  <a href="/html/index.htm" title="HTML Tutorial" />HTML Tutorial</a>
</body>
</html>
```

این مثال نتیجه ی زیر را تولید خواهد کرد.



برچسب <link> در HTML

این برچسب ارتباط بین داکيومنت جاری و منبع خارجی را تعیین می کند. در ادامه مثالی را می بینید از لینک یک فایل style sheet خارجی موجود در مسیر CSS با یک web root.

```
<!DOCTYPE html>
```

```
<html>
<head>
  <title>HTML link Tag Example</title>
  <base href="http://www.tahlildadeh.com/" />
  <link rel="stylesheet" type="text/css" href="/css/style.css">
</head>
<body>
  <p>Hello, World!</p>
</body>
</html>
```

این مثال نتیجه ی زیر را تولید خواهد کرد.

Hello, World!

برچسب <style> در HTML

این برچسب برای تعیین **style sheet** برای داکيومنت جاری HTML استفاده می شود. در ادامه **مثالی** را می بینید از تعریف برخی قوانین **style sheet** در داخل برچسب **<style>**.

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML style Tag Example</title>
  <base href="http://www.tahlildadeh.com/" />
  <style type="text/css">
    .myclass {
      background-color: #aaa;
      padding : 10px;
    }
  </style>
</head>
<body>
  <p class="myclass">Hello, World!</p>
</body>
</html>
```

این مثال نتیجه ی زیر را تولید خواهد کرد.

Hello, World!

توجه برای فراگیری چگونگی کار **Cascading Style Sheet** یک آموزش مجزای موجود در [اینجا](#) را کلیک کنید.

برچسب <script> در HTML

این برچسب برای وارد کردن فایل اسکریپت خارجی و یا تعریف فایل اسکریپت داخلی برای داکيومنت HTML استفاده می شود. در زیر مثالی را می بینید که در آن از جاوا اسکریپت برای تعریف عملکرد ساده ی جاوا اسکریپت استفاده می کنیم.

```
<!DOCTYPE html>
<html>
<head>
<title>HTML script Tag Example</title>
<base href="http://www.tahlildadeh.com/" />
<script type="text/javascript">
function Hello() {
    alert("Hello, World");
}
</script>
</head>
<body>
<input type="button" onclick="Hello();" name="ok" value="OK" />
</body>
</html>
```

آموزش Cascading style sheet

Cascading style sheet (CSS) بیان می کند که داکيومنت ها روی صفحه و در چاپ چگونه نشان داده می شوند، یا شاید چگونه بیان می شوند. از زمانی که کنسرسیوم در سال 1994 تاسیس شد، W3C به طور فعالانه ای در استفاده از **Style sheet** روی وب پیشرفت کرده است.

CSS جایگزین های ساده و موثری را برای تعیین ویژگی های مختلف برچسب های HTML ارائه می دهد. با استفاده از **CSS** شما می توانید تعدادی از ویژگی های طراحی را برای یک HTML ارائه شده تعیین کنید. هر ویژگی دارای نام مقدار میباشد که به وسیله ی علامت () از هم جدا شده اند. اطلاعاتی مربوط به هر ویژگی نیز توسط علامت نقطه ویرگول (;) از یکدیگر جدا شده اند.

مثال

ابتدا اجازه بدهید که **مثالی** از داکيومنت HTML را مورد بررسی قرار دهیم که برای تعیین رنگ و اندازه ی فونت از برچسب استفاده می کند.

```
<!DOCTYPE html>
```

```
<html>
<head>
  <title>HTML CSS</title>
</head>
<body>
  <p><font color="green" size="5">Hello, World!</font></p>
</body>
</html>
```

می‌توانیم با استفاده از **style sheet** مثال بالا را مانند زیر بازنویسی کنیم.

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML CSS</title>
</head>
<body>
  <p style="color:green;font-size:24px;">Hello, World!</p>
</body>
</html>
```

این مثال نتیجه‌ی زیر را تولید خواهد کرد.

Hello, World!

شما می‌توانید به سه روش از **CSS** در داکيومنت **HTML** خود استفاده کنید.

Style sheet خارجی – قوانین **style sheet** را در یک فایل **css** مجزا تعریف می‌کند، و سپس آن فایل را با استفاده از برچسب **<link>** در **HTML** وارد داکيومنت **HTML** شما می‌کند.

Style sheet داخلی – قوانین **style sheet** را در با استفاده از برچسب **<style>** در بخش سربرگ داکيومنت **HTML** تعریف می‌کند.

Inline style sheet – قوانین **style sheet** را مستقیماً و به همراه عناصر **HTML** با استفاده از ویژگی **style** تعریف می‌کند.

اجازه بدهید هر سه مورد را یکی یکی و با استفاده از مثال‌های مناسب بررسی کنیم.

Style sheet خارجی

اگر می خواهید از **style sheet** خود در صفحات مختلف استفاده کنید، توصیه می شود که یک **style sheet** متداول در یک فایل مجزا تعریف کنید. یک فایل **cascading style sheet** دارای ضمیمه هایی مانند **css** می باشد و با استفاده از برچسب **<link>** وارد فایل های **HTML** خواهد شد.

مثال

توجه داشته باشید که ما یک فایل **style sheet** به نام **style.css** که دارای قوانین زیر می باشد، تعریف می کنیم.

```
.red{
  color:red;
}
.thick{
  font-size:20px;
}
.green{
  Color:green;
}
```

در اینجا سه قانون **CSS** را تعریف کردیم که برای سه گروه متفاوت تعریف شده در برچسب های **HTML** مناسب می باشند. پیشنهاد می کنم در مورد چگونگی تعریف این قوانین خود را اذیت نکنید، زیرا هنگام مطالعه ی **CSS** آنها را فرا خواهید گرفت. اکنون اجازه بدهید از فایل **CSS** خارجی بالا در داکيومنت **HTML** زیر استفاده کنیم.

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML External CSS</title>
  <link rel="stylesheet" type="text/css" href="/html/style.css">
</head>
<body>
  <p class="red">This is red</p>
  <p class="thick">This is thick</p>
  <p class="green">This is green</p>
  <p class="thick green">This is thick and green</p>
</body>
</html>
```

این **مثال** نتیجه ی زیر را تولید خواهد کرد.

This is red

This is thick

This is green

This is thick and green

Style sheet داخلی

اگر می خواهید قوانین **style sheet** را برای یک داکيومنت مجزا به کار ببرید، فقط آن موقع است که می توانید این قوانین را با استفاده از برچسب **<style>** وارد بخش سربرگ داکيومنت **HTML** کنید.

قوانین تعریف شده در **style sheet** داخلی، قوانین تعریف شده در فایل **CSS** خارجی را نیز در بر می گیرد.

مثال

اجازه بدهید **مثال** بالا را یک بار دیگر بازنویسی کنیم، اما در اینجا قوانین **style sheet** را در همان داکيومنت **HTML** و با استفاده از برچسب **<style>** می نویسیم.

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Internal CSS</title>
  <style type="text/css">
    .red {
      Color: red;
    }
    .thick {
      font-size :20px;
    }
    .green {
      Color: green;
    }
  </style>
</head>
<body>
  <p class="red">This is red</p>
  <p class="thick">This is thick</p>
  <p class="green">This is green</p>
  <p class="thick green">This is thick and green</p>
</body>
</html>
```

این مثال نتیجه ی زیر را تولید خواهد کرد.

This is red

This is thick

This is green

This is thick and green

Inline Style Sheet

شما می توانید با استفاده از ویژگی **style** از برجسب مربوطه، قوانین **style sheet** را مستقیماً برای هر عنصر **HTML** به کار برید. این امر فقط زمانی می تواند انجام شود که شما علاقمند به ایجاد تغییرات خاص در هر کدام از عناصر **HTML** می باشید.

قوانین تعریف شده با عنصر درون خطی، قوانین تعریف شده در یک فایل **CSS** خارجی و نیز قوانین تعریف شده در یک عنصر **<style>** را در برمی گیرد.

اجازه بدهید **مثال** بالا را یک بار دیگر بازنویسی کنیم، اما این بار قوانین **style sheet** را همراه با قوانین **HTML** و با استفاده از ویژگی **style** در عناصر خواهیم نوشت.

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Inline CSS</title>
</head>
<body>
  <p style="color:red;">This is red</p>
  <p style="font-size:20px;">This is thick</p>
  <p style="color:green;">This is green</p>
  <p style="color:green;font-size:20px;">This is thick and green</p>
</body>
</html>
```

این مثال نتیجه ی زیر را تولید خواهد کرد.

This is red

This is thick

This is green

This is thick and green

آموزش javascript

یک اسکریپت قطعه ی کوچکی از برنامه است که می تواند به وب سایت شما تعامل اضافه کند. به عنوان مثال یک اسکریپت می تواند یک هشدار پاپ آپ مربوط به باکس پیغام تولید کند، یا یک منوی رو به پایین ایجاد کند. این اسکریپت توسط **javascript** یا **VBScript** نوشته می شود.

شما می توانید با استفاده از هر زبان اسکریپتی، کارکردهای متنوع کوچکی به نام گردانندگان رویداد بنویسید و سپس می توانید آن عملکردها را با استفاده از ویژگی های **HTML** اجرا کنید.

این روزها فقط **javascript** و چارچوب های متناظر با آن توسط بسیاری از توسعه دهندگان وب استفاده می شوند. **VBScript** حتی توسط برخی مرورگرها پشتیبانی هم نمی شود.

شما می توانید کد **javascript** را در یک فایل مجزا نگهداری کرده و سپس هرزمان که لازم بود آن را وارد کنید، یا می توانید قابلیت را در داخل خود داکيومنت **HTML** تعریف کنید. اجازه بدهید هر دو مورد را یکی یکی با **مثال** های مناسب بررسی کنیم.

جاوااسکریپت خارجی

اگر قصد دارید قابلیت را تعریف کنید که در انواع داکيومنت های **HTML** استفاده می شود، بهتر است آن قابلیت را در یک فایل مجزای جاوااسکریپت حفظ کرده و آن فایل را وارد داکيومنت های **HTML** خود کنید. یک فایل جاوااسکریپت دارای ضمیمه هایی مانند JS می باشد که با استفاده از برچسب **<script>** وارد فایل های **HTML** می شود.

مثال

تصور کنید که با استفاده از جاوااسکریپت در **script.js** یک عملکرد کوچک را تعریف می کنیم که دارای کد زیر می باشد.

```
function Hello()
{
  alert("Hello, World");
}
```

اکنون اجازه بدهید از فایل جاوااسکریپت خارجی بالا در داکيومنت HTML خود استفاده کنیم، مانند زیر

```
<!DOCTYPE html>
<html>
<head>
  <title>Javascript External Script</title>
  <script src="/html/script.js" type="text/javascript" /></script>
</head>
<body>
  <input type="button" onclick="Hello();" name="ok" value="Click Me" />
</body>
</html>
```

این مثال نتیجه ای را به دنبال خواهد داشت که می توانید با کلیک کردن بر روی دکمه ی ارائه شده توسط این مثال آن را امتحان کنید.

جاوا اسکریپت درونی

شما می توانید کد جاوااسکریپت را مستقیماً روی داکيومنت HTML خود بنویسید. معمولاً کد اسکریپت را با استفاده از برچسب **<script>** در سربرگ داکيومنت نگهداری می کنیم در غیر این صورت هیچ محدودیتی وجود نداشته و می توانید کد خود را در هر جایی در داکيومنت به جز داخل برچسب **<script>** نگه دارید.

```
<!DOCTYPE html>
<html>
<head>
  <title>Javascript Internal Script</title>
  <script type="text/javascript">
    function Hello() {
      alert("Hello, World");
    }
  </script>
</head>
<body>
  <input type="button" onclick="Hello();" name="ok" value="Click Me" />
</body>
</html>
```

این مثال نتیجه ای را به دنبال خواهد داشت که می توانید با کلیک کردن بر روی دکمه ی ارائه شده توسط این مثال آن را امتحان کنید.

گردانندگان رویدادها (EventHandler())

گردانندگان رویدادها عملکردهای تعریف شده ی ساده ای می باشند که می توانند برعلیه هر یک از عملکردهای ماوس یا صفحه کلید فراخوانده شوند. شما می توانید عملکرد خود را در یک گرداننده ی رویداد تعریف کنید که می تواند یک خط تا 1000 خط کد باشد.

در زیر مثالی را می بینید که چگونگی نگارش یک گرداننده ی رویداد را نشان می دهد. اجازه بدهید یک تیتتر ساده ی *EventHandler()* را در تیتتر داکيومنت بنویسیم. هر زمان که کاربر ماوس را روی یک پاراگراف بیاورد، این عملکر را فرا خواهیم خواند.

```
<!DOCTYPE html>
<html>
<head>
<title>Event Handlers Example</title>
<script type="text/javascript">
function EventHandler() {
    alert("I'm event handler!!");
}
</script>
<head>
<body>
<p onmouseover="EventHandler();">Bring your mouse here to see an alert</p>
</body>
</html>
```

این مثال نتیجه ی زیر را تولید خواهد کرد، ماوس خود را روی آن آورده تا نتیجه را ببینید.

Bring your mouse here to see an alert

مخفی کردن اسکریپت ها از مرورگرهای قدیمی

گرچه بیشتر مرورگرها این روزها جاوا اسکریپت را پشتیبانی می کنند، اما هنوز مرورگرهایی قدیمی تری هستند که این کار را نمی کنند. اگر مرورگری جاوا اسکریپت را پشتیبانی نکند، به جای اجرای اسکریپت کدی را به کاربر نمایش می دهد. برای جلوگیری از این مسئله می توانید کامنت های **HTML** را در اطراف جاوا اسکریپت قرار دهید، همانطور که در زیر می بینید.

JavaScript Example

```
<script type="text/javascript">
<!--
document.write("Hello Javascript!");
//-->
</script>
VBScript Example
<script type="text/vbscript">
<!--
document.write("Hello VBScript!")
'-->
</script>
```

عنصر <noscript>

می توانید یک سری اطلاعات جایگزین به کاربری ارائه دهید که مرورگر وی اسکریپت ها را پشتیبانی نمی کند، همچنین برای کاربرهایی که گزینه ی **script** مرورگر خود را غیرفعال کرده اند. شما می توانید اینکار را با استفاده از برچسب **<noscript>** انجام دهید.

JavaScript Example

```
<script type="text/javascript">
<!--
document.write("Hello Javascript!");
//-->
</script>
<noscript>Your browser does not support Javascript!
</noscript>
```

درمان ضمن شکر از انتخاب شما، امیدواریم مطالب این کتاب برای شما مفید بوده باشد.

از دوره حرفه ای طراحی که مسایل و اکنشنگرایی با **flexbox** و همچنین پروژه طراحی یک فروشگاه را دارد دیدن کنید

<http://netparadis.com/tutorials/web-design/>