

اصول طراحی صفحات وب

هدف: طراحی وب سایت‌های ایستا

HTML

نسخه: ۱.۱۵

kishuniv@montajab.com
<http://montajab.com/kishuniv>

HTML

Hypertext Markup Language

HTML یک زبان نشانه‌گذاری است که برای ساخت و ارائه محتوا در شبکه جهانی وب مورد استفاده قرار می‌گیرد. HTML5 در ۲۸ اکتبر ۲۰۱۴، توسط کنسرسیوم شبکه جهانی وب (W3C) نهایی و منتشر شد.

Version	Year
HTML	1991
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML	2000
HTML5	2014

- یک زبان نشانه گذاری، مجموعه ای از Tagهاست.
- اسناد HTML با استفاده از Tagهای HTML توصیف می شوند.
- این Tag (برچسب)ها، کلمات کلیدی هستند که بین دو علامت < و > قرار می گیرند:

<tagname>

- تگ های HTML معمولا به صورت جفت می آیند:

<tagname >Content</tagname>

opening tag

closing tag

<tagname >Content</tagname>

opening tag

closing tag

- به توصیه W3C تگ‌ها را در HTML5 با حروف کوچک می‌نویسیم.
- تگ پایانی با یک کاراکتر / مشخص می‌گردد.
- در صورت فراموش کردن تگ پایانی، ممکن است نتیجه مورد نظر حاصل نشود.



ساختار یک فایل HTML

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

Metadata

```
</head>
```

```
<body>
```

Content

```
</body>
```

```
</html>
```

ساختار یک فایل HTML

```
<!DOCTYPE html>
```

<!DOCTYPE> یک تگ HTML نیست.

<!DOCTYPE> اولین تگی است که در فایل HTML قرار می‌گیرد (قبل از

<html>) و نسخه HTML را برای مرورگر وب مشخص می‌کند.

این تگ در HTML5 به صورت زیر تعریف می‌گردد:

```
<!DOCTYPE html>
```

لیست DOCTYPE ها:

<http://www.w3.org/QA/2002/04/valid-dtd-list.html>

ساختار یک فایل HTML

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

Metadata

```
</head>
```

Metadata: مجموعه ای از داده است که سایر داده‌ها را توصیف کرده و اطلاعاتی در مورد آن‌ها ارائه می‌کند.

Metadata در HTML شامل اطلاعاتی در مورد سند HTML است و توسط تگ‌های زیر توصیف می‌شود:

- **title:** عنوان صفحه HTML را تعیین می‌کند.
- **style:** استایل صفحه HTML را تعریف می‌کند.
- **meta:** شامل اطلاعاتی مانند کلمات کلیدی صفحه، نام مولف و ... باشد.
- **link:** منابع خارجی (مانند فایل‌های CSS) را به صفحه HTML لینک می‌کند.
- **script:** اسکریپت‌های سمت کلاینت در این تگ قرار می‌گیرند.
- **base:** آدرس پیش‌فرض تمام لینک‌های صفحه را تعیین می‌کند.

```
</html>
```

مثال:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>My Title</title>
```

```
<base href="http://www.name.com/images/" target="_blank">
```

```
</head>
```

```
<body>
```

```

```

```
</body>
```

```
</html>
```

عنوان صفحه در Tab مرورگر
عنوان صفحه در هنگام اضافه کردن به Favorites
عنوان صفحه در نتیجه‌ی موتورهای جستجو

تگ base، آدرس پیش‌فرض تمام
لینک‌های صفحه HTML را تعیین می‌کند.

مقدار پیش‌فرض صفت target را برای تمام
لینک‌های صفحه HTML تعیین می‌کند.

آدرس کامل که با استفاده از تگ base تعیین شده است:
http://www.name.com/images/html5.gif

تگ meta می‌تواند شامل توضیحاتی در مورد صفحه وب، کلمات کلیدی صفحه، نام مولف سند، نوع رمزگذاری کاراکترهای سند و مانند آن‌ها باشند.

Name	Value
charset	<i>Character Set</i>
http-equiv	content-type default-style refresh
Name	application-name author description generator keywords
content	<i>Text</i>

صفت‌های تگ meta:

باعث می‌شود صفحه هر ۳۰ ثانیه یکبار Refresh شود.

```
<meta http-equiv="refresh" content="30">
```

کلمات کلیدی مرتبط با این صفحه را مشخص می‌کند.
موتورهای جستجو از این کلمات استفاده می‌کنند.

```
<meta name="keywords" content="HTML,CSS">
```

مقدار مرتبط با http-equiv و یا Name را تعیین می‌کند.

Unicode: مجموعه‌ای از کاراکترهاست که هر کدام با یک عدد یکتای در مبنای شانزده (Code Points) مشخص می‌شوند:

Hex:	A: 41	B:42	C:43	...	I:49	J:4A	...
Dec:	A: 65	B:66	C:67	...	I:73	J:74	...

بین ۱ تا ۴ بایت

UTF-8: نوعی رمزگذاری (چگونگی تبدیل اعداد به باینری) است.

رمزگذاری کاراکترها در HTML5 به صورت پیش فرض UTF-8 است.

```
<head>
  <meta charset="UTF-8">
</head>
```

Character Encoding

HTML 4.01: `<meta http-equiv="content-type" content="text/html; charset=UTF-8">`

HTML5: `<meta charset="UTF-8">`

ساختار یک فایل HTML

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

Metadata

```
</head>
```

```
<body>
```

شامل محتوای صفحه، مانند متن، تصاویر و لینک‌هاست.
محتوایی که در این بخش قرار داده شود، نمایش داده می‌شود.

```
</body>
```

```
</html>
```

بر اساس استاندارد **HTML5** می‌توان تگ‌های **html**، **head** و **body** را حذف نمود.

مرورگرهای وب تگ‌های HTML را نمایش نمی‌دهند، بلکه با استفاده از آن‌ها چگونگی نمایش محتوا را تشخیص می‌دهند.

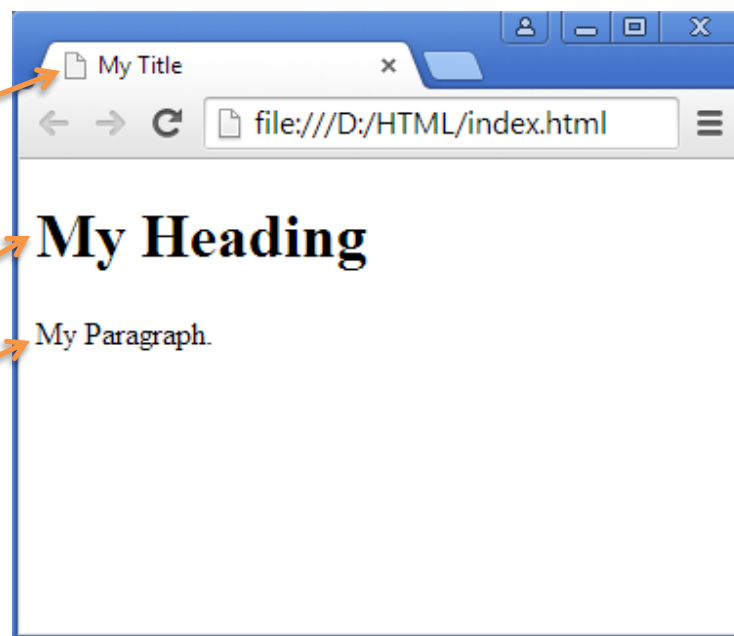
در صورت امکان از حروف کوچک برای نام فایل استفاده نمایید.

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Title</title>
  </head>
  <body>

    <h1>My Heading</h1>
    <p>My Paragraph.</p>

  </body>
</html>
```



استفاده از پسوند **.html** توصیه می‌شود.

تفاوت پسوند **.html** و **.htm** ؟



عناصر قالب بندی متن

Bold:	<code><p>Text1 Text2</p></code>
Strong:	<code><p>Text1 Text2</p></code>
Italic:	<code><p>Text1 <i>Text2</i></p></code>
Emphasized:	<code><p>Text1 Text2</p></code>
Small:	<code><p>Text1 <small>Text2</small></p></code>
Marked:	<code><p>Text1 <mark>Text2</mark></p></code>
Deleted:	<code><p>Text1 ~~Text2~~</p></code>
Inserted:	<code><p>Text1 <ins>Text2</ins></p></code>
Subscript:	<code><p>Text1 <sub>Text2</sub></p></code>
Superscript:	<code><p>Text1 <sup>Text2</sup></p></code>
Variable:	<code><p><var>E = m c<sup>2</sup></var></p></code>

Text1 **Text2**

Text1 **Text2**

Text1 *Text2*

Text1 *Text2*

Text1 Text2

Text1 Text2

Text1 ~~Text2~~

Text1 Text2

Text1 Text2

Text1 Text2

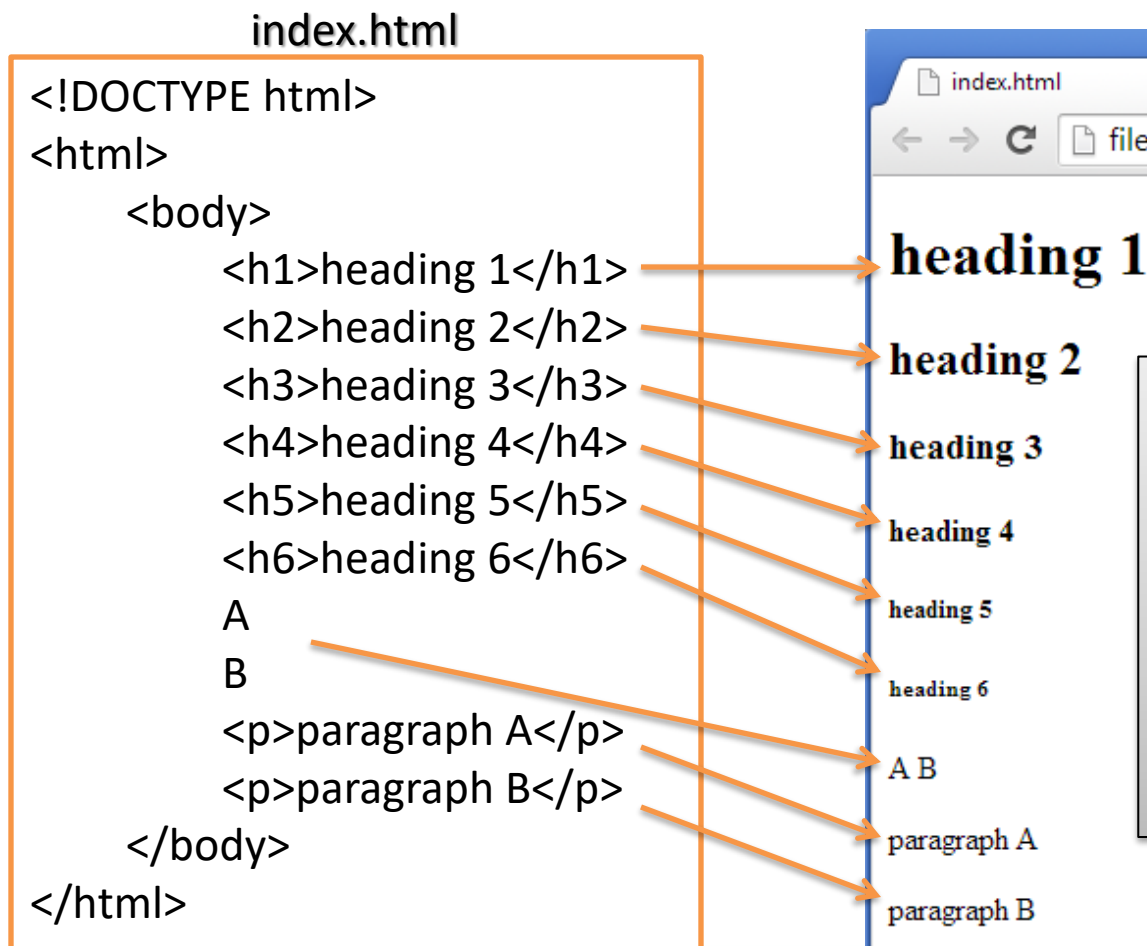
$E = m c^2$

Mathematical Variable

تفاوت تگ `<b>` و `<strong>` ؟ تفاوت تگ `<i>` و `<em>` ؟



سرتیترها و پاراگراف:



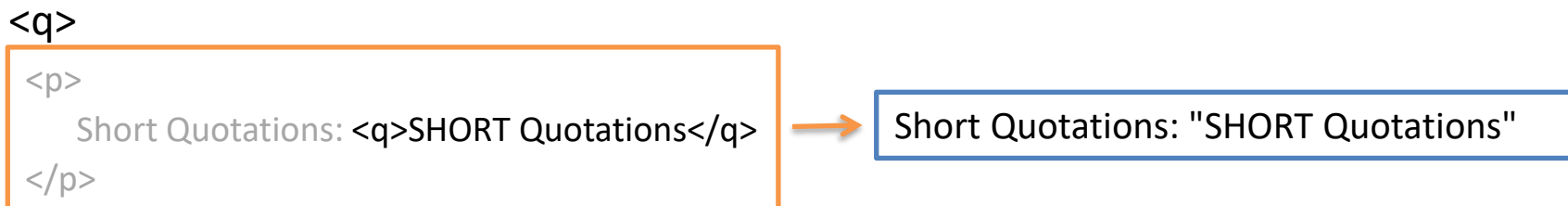
نکته: برای تغییر اندازه متن نباید از تگهای سرتیتر استفاده کرد، زیرا موتورهای جستجو از آنها برای ایندکس کردن ساختار و محتوای صفحات وب استفاده می‌کنند. در واقع برای نمایش ساختار صفحه از این تگها استفاده می‌کنیم (اهمیت این تگها به ترتیب از H1 تا H6 است).

Empty HTML Elements:

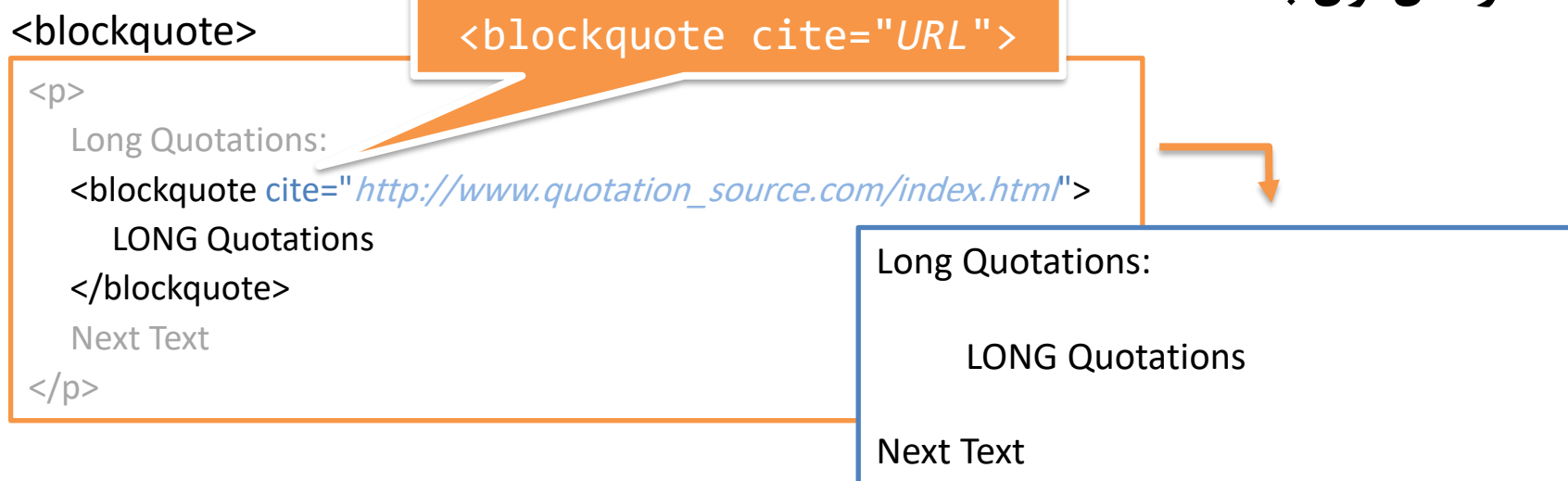
این عناصر هیچ محتوایی ندارند، مانند:

- **
**: این عنصر یک line break تعریف می‌کند، یعنی باعث می‌شود ادامه محتوا از سطر بعد نمایش داده شود.
- **<hr>**: این عنصر یک thematic break تعریف می‌کند و برای جدا کردن موضوعات محتوا از هم بکار می‌رود؛ این عنصر باعث می‌شود یک خط افقی در صفحه برای جدا کردن محتوا نمایش داده شود.

عنصر نقل قول کوتاه:



عنصر نقل قول بلند:



می‌توان با استفاده از صفت `cite` مرجع نقل قول (کوتاه/بلند) را تعیین نمود؛ مرجع تعیین شده نمایش داده نمی‌شود، اما این کار باعث می‌شود تا موتورهای جستجو اطلاعات بیشتری در مورد نقل قول مورد نظر بدست آورند.

عنصر مخفف:

<abbr>

<p>

With <abbr title="Hyper Text Markup Language"> HTML </abbr> you can create your own Web site.

</p>

abbreviation



With HTML you can create your own Web site.



With HTML Hyper Text Markup Language Web site.

بدون این تگ نیز می‌توان مخفف یک عبارت اختصاری را به صورت tooltip (با استفاده از تگ title) نمایش داد. اما این تگ، با مشخص کردن مخفف برای مرورگرها، سیستم‌های ترجمه و موتورهای جستجو به آن‌ها کمک می‌کند.

عنصر آدرس (اطلاعات تماس):

اطلاعات تماس نویسنده/مالک یک سند یا مقاله را مشخص می‌کند. این اطلاعات می‌تواند آدرس ایمیل، URL، آدرس فیزیکی، شماره تلفن، آدرس شبکه‌های اجتماعی و غیره باشد.

```
<p>HTML,...</p>  
<address>  
  email: <a href="mailto:kishuniv@montajab.com">kishuniv@montajab.com</a><br />  
</address>  
<p>HTML,...</p>
```



```
HTML,...  
  
email: kishuniv@montajab.com  
  
HTML,...
```

عنصر معکوس نویسی:

<bdo>

```
<p>
  <bdo dir="rtl">right to left</bdo>
</p>
```

bi-directional override

tfel ot thgir

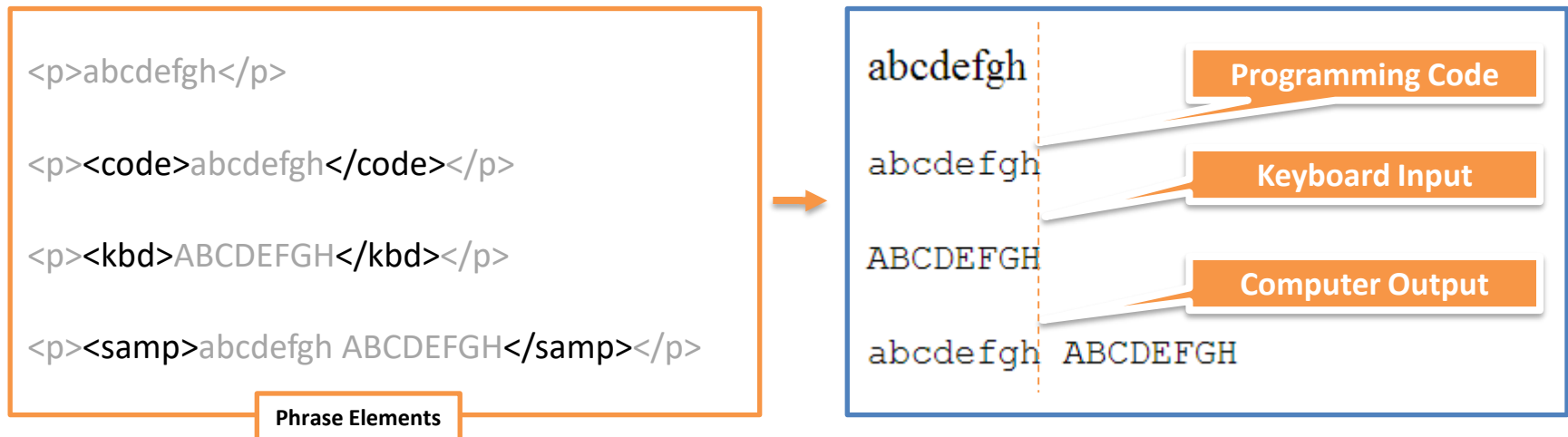
عناصر کدهای کامپیوتری:

در عناصر زیر، اندازه کارکترها و فاصله متن بین این تگ‌ها یکسان است:

`<code>`: Programming Code

`<kbd>`: Keyboard Input

`<samp>`: Computer Output



Phrase Elements

Programming Code

Keyboard Input

Computer Output

pre: از این تگ برای تعریف متن‌های قالب‌بندی شده استفاده می‌گردد.

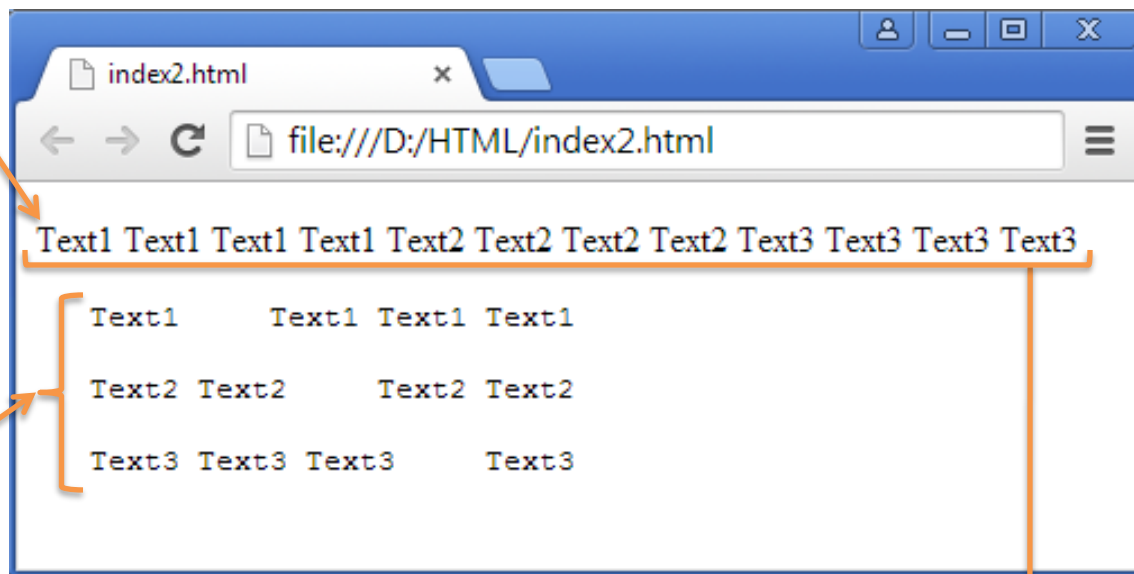
```
<p>
  Text1   Text1 Text1 Text1

  Text2 Text2   Text2 Text2

  Text3 Text3 Text3   Text3
</p>
<pre>
  Text1   Text1 Text1 Text1

  Text2 Text2   Text2 Text2

  Text3 Text3 Text3   Text3
</pre>
```



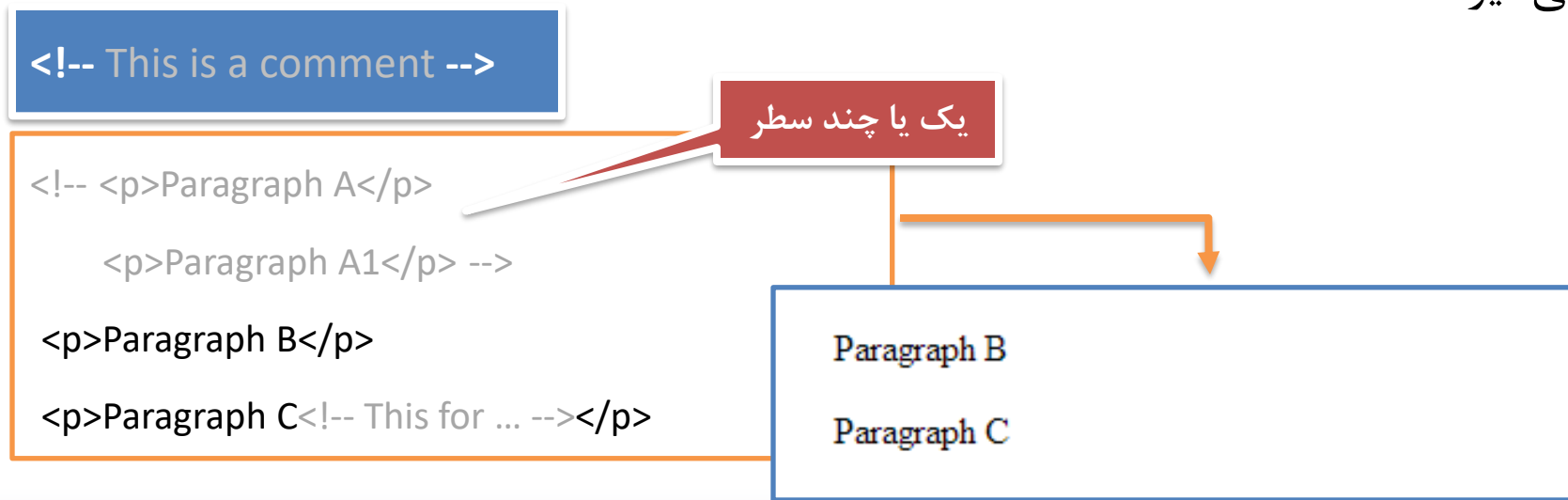
Preformatted Text

متن درون این تگ با همان شکل و ساختاری که در فایل HTML آورده شده است، نمایش داده می‌شود.

در حالت عادی، تمام فاصله‌ها و خطوط بین کلمات به اندازه یک فاصله در نظر گرفته می‌شوند.

توضیحات:

اگر بخواهیم توضیحی، برای بخش خاصی، در کد قرار دهیم و یا جلوی اعمال بخشی از کد را بدون پاک کردن آن بگیریم، از امکان توضیحات (Comments) استفاده می‌کنیم، که با `<!--` شروع و با `-->` پایان می‌پذیرد. در واقع مرورگر این بخش از کد را نادیده می‌گیرد:



توضیحات شرطی:

فقط در صورتی که شرط برقرار باشد، عبارت بین دو تگ نمایش داده خواهد شد. 

`<!--[if Condition]> Conditional Comment <![endif]-->`

```
<!--[if IE]> Internet Explorer (HTML Code) <![endif]-->  
<!--[if IE 8]> Internet Explorer 8.0 (HTML Code) <![endif]-->  
<!--[if lt IE 8]> Internet Explorer 7... (HTML Code) <![endif]-->  
<!--[if gt IE 8]> Internet Explorer 9... (HTML Code) <![endif]-->
```

lt: Lesser Than
gt: Greater Than
lte: Lesser Than or Equal
gte: Greater Than or Equal

Attributes

(صفت‌ها)

عناصر HTML می‌تواند دارای صفاتی باشند، که این صفات ویژگی‌های بیشتری در مورد عناصر ارائه می‌کنند. صفات همیشه در تگ شروع اضافه می‌شوند:

<tagname name="value"> ... </tagname>

نام صفت

مقدار صفت

<tagname name='value'> ... </tagname>

<tagname name=value> ... </tagname>

اگر مقدار صفت فاقد کاراکتر فاصله باشد (وگرنه فقط عبارت قبل از فاصله لحاظ می‌گردد).

به توصیه W3C، صفت‌ها را در HTML5 با حروف کوچک می‌نویسیم.

تصویر:

تگ `<img>`، فضایی برای تصویر در صفحه ایجاد می‌کند.

از نظر فنی تصویر در صفحه اضافه نمی‌شود، بلکه به صفحه لینک می‌شود.

```

```

صفت **src**: URL تصویر است.

صفت **alt**: در صورت عدم نمایش تصویر توسط مرورگر، این متن جایگزین (alternative) تصویر می‌شود؛ علت عدم نمایش می‌تواند سرعت پایین ارتباط، اشکال در **src** و یا استفاده کاربر از نرم‌افزارهای صفحه‌خوان باشد.

صفت **height**: ارتفاع فضایی است که تصویر در آن قرار می‌گیرد.

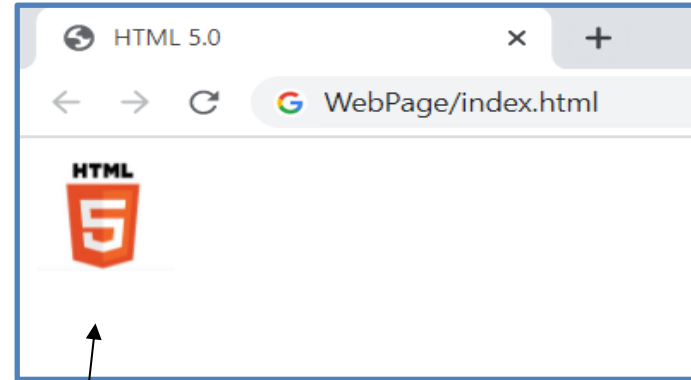
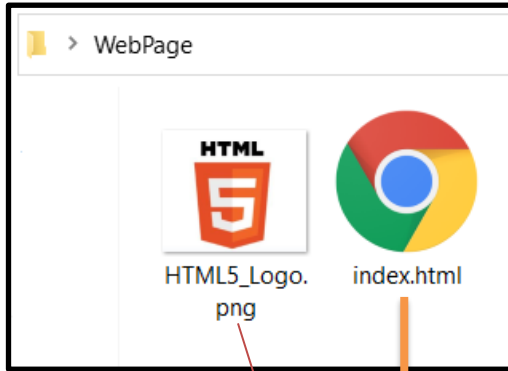
صفت **width**: عرض فضایی است که تصویر در آن قرار می‌گیرد.

صفت **title**: tooltip محدوده این تگ را مشخص می‌کند.

صفت **id**: یک شناسه منحصر به فرد برای یک عنصر تعیین می‌کند.

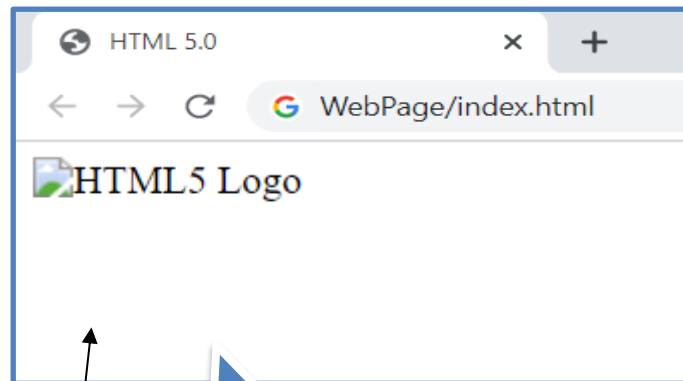
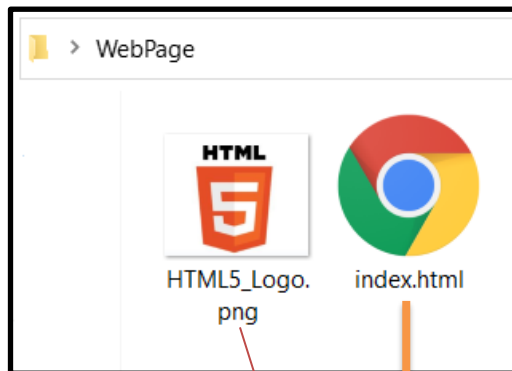
استفاده از ویژگی‌های **height** و **width** در CSS مقدم‌تر از صفات **height** و **width** است.

تصویر:



```
<!DOCTYPE html>
<html>
  <head>
    <title> HTML 5.0 </title>
  </head>
  <body >
    
  </body>
</html>
```

تصویر:



```
<!DOCTYPE html>
<html>
  <head>
    <title> HTML 5.0 </title>
  </head>
  <body >
    
  </body>
</html>
```

در این مثال، متن صفت **alt** جایگزین تصویر شده است، علت عدم نمایش تصویر، اشکال در صفت **src** است.

تصویر:

```
<body >  
  <figure>  
      
    <figcaption>Fig.1 - Kish Island</figcaption>  
  </figure>  
</body>
```



Fig.1 - Kish Island

لینک:

لینک‌های HTML، هایپرلینک هستند و با استفاده از تگ `<a>` تعریف می‌شوند. یک هایپرلینک، متن و یا تصویری است که با کلیک بر روی آن به سند دیگری منتقل می‌شویم.

```
<a href="url">Link text</a>
```

صفت `href`: آدرس مقصد را تعیین می‌کند.

```
<a href="http://www.montajab.com/index.html" target="_blank">Montajab</a>
```

آدرس مطلق
Absolute Address

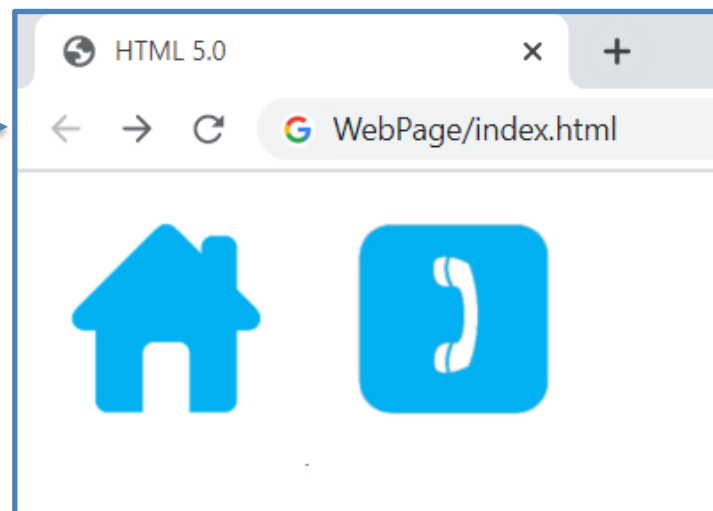
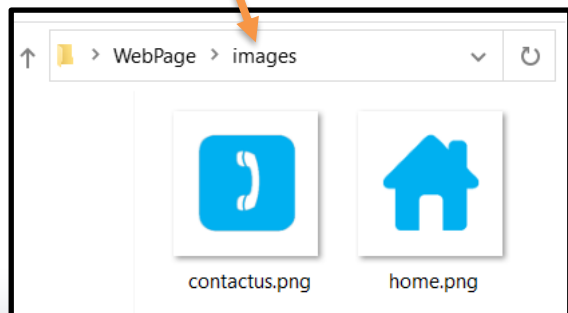
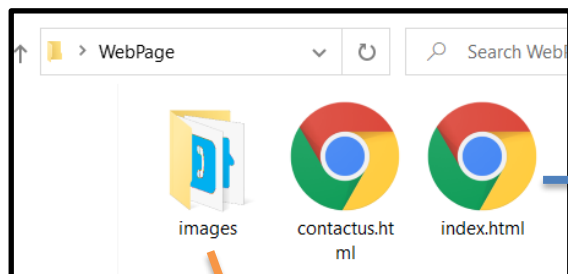
صفت `target`: تعیین می‌کند که سند لینک، در کجا باز شود:

- `_self`: سند لینک، در همان جایی که کلیک شده باز می‌شود (پیش فرض).
- `_blank`: سند لینک، در یک پنجره یا تب جدیدی باز می‌شود.
- `_parent`: سند لینک، در فریم والد باز می‌شود.
- `_top`: سند لینک، بدون در نظر گرفتن فریم‌ها، به صورت کامل در صفحه باز می‌شود.
- `framename`: سند لینک، در فریمی با این نام باز می‌شود.

لینک تصویری:

آدرس نسبی
Relative Address

```
<a href="index.html">
  
</a>
<a href="contactus.html">
  
</a>
```



:Bookmark

با استفاده از Bookmark می‌توان به بخش خاصی از همان صفحه و یا بخش خاصی از صفحه‌ای دیگر منتقل (اسکرول) شد؛ که عملاً در صفحات طولانی کاربرد دارد.

```
...  
<h1 id="header1">HEADER 1</h1>  
...
```

اسکرول به موقعیت عنصری که صفت id آن header1 است:

```
<a href="#header1">Visit HEADER 1</a>
```

```
<a href="index.html#header1">Visit HEADER 1</a>
```

استایل HTML

برای توصیف نحوه نمایش اسناد HTML از زبان CSS (Cascading Style Sheets) استفاده می‌شود.
برای استفاده از CSS سه روش وجود دارد:

1. External style sheet
2. Internal style sheet
3. Inline style

صفت **style**:

`style="CSS_property:CSS_value"`

مثال:

`<p style="color:blue">Text</p>`

`<p style="color:blue; font-size:14px; text-align:center">Text</p>`



استایل HTML

برای توصیف نحوه نمایش اسناد HTML از زبان CSS (Cascading Style Sheets) استفاده می‌شود.
برای استفاده از CSS سه روش وجود دارد:

1. External style sheet
2. Internal style sheet
3. Inline style

CSS Selectors

element

id

class

تگ style:

```
<head>
  <style>
    Selector {CSS_property:CSS_value;}
  </style>
</head>

<style>
  p {color:blue; font-family:tahoma; }
</style>
```

مثال:

استایل HTML

برای توصیف نحوه نمایش اسناد HTML از زبان CSS (Cascading Style Sheets) استفاده می‌شود.
برای استفاده از CSS سه روش وجود دارد:

1. External style sheet
2. Internal style sheet
3. Inline style

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

استایل‌های تعریف شده در یک فایل متنی
CSS، با پسوند CSS ذخیره می‌شوند.

برای استفاده از یک فایل CSS، باید آن را به صفحه مورد نظر، توسط یک لینک ارجاع داد.

عناصر درون خطی و بلوک:

`<span>inline</span>`

`<div>block-level</div>`

Inline Element

Inline `<span>Element</span>`

`<div>Block-Level Element</div>`

`<span>Inline Element</span>`

`<br><br>`

`<div style="background-color:black; color:orange;">Block-Level Element</div>`

`<br>`

text`<span style="background-color:black; color:orange;">Inline Element</span>`text

`<div>`

`<h1> - <h6>`

`<p>`

`<form>`

`<span>`

`<a>`

`<img>`

Inline Element Inline Element

Block-Level Element

Inline Element

Block-Level Element

text**Inline Element**text

ظاهر لینک:

به صورت پیش فرض، یک لینک در حالت های مختلف رنگ های مختلفی دارد که می توان آن ها را تغییر داد.

```
<style>
a:link {color:blue;
        background-color:transparent;
        text-decoration:underline}

a:visited {color:Purple;
           background-color:transparent;
           text-decoration:underline}

a:hover {color:blue;
         background-color:transparent;
         text-decoration:underline}

a:active {color:red;
          background-color:transparent;
          text-decoration:underline}
</style>
```

بعد از کلیک لینک

هنگام رفتن موس
بر روی لینک

هنگام کلیک موس
بر روی لینک

text-decoration

none
(پیش فرض)

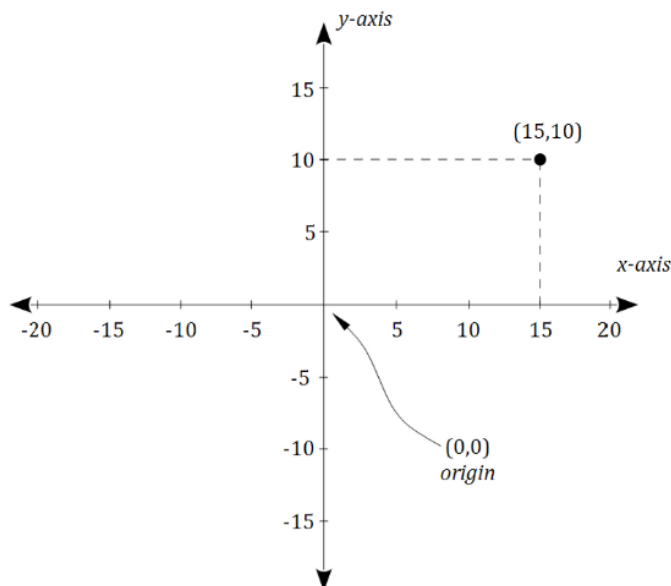
underline

overline

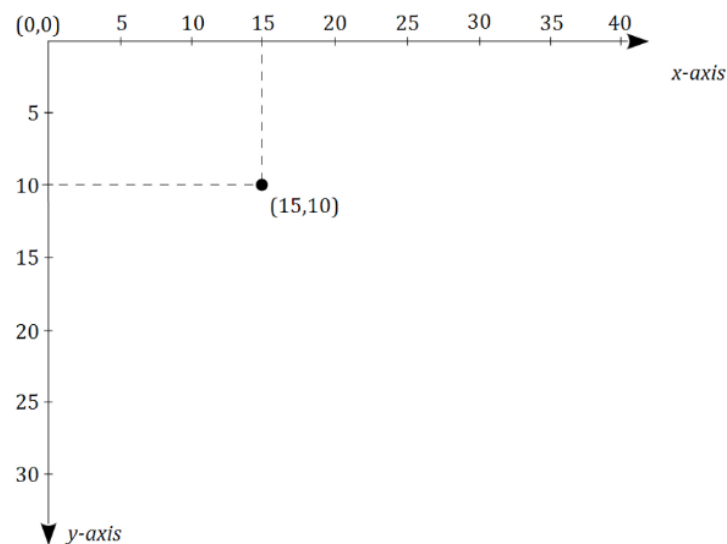
~~line-through~~

سیستم‌های مختصات:

مبدأ در سیستم مختصات صفحه نمایش، با سیستم مختصات دکارتی متفاوت بوده و در گوشه بالا سمت چپ است.



Cartesian coordinate system



screen coordinate system

:Image Map

با تگ `map` تعریف می‌شود و این امکان را می‌دهد تا بخش‌هایی از یک تصویر، قابل کلیک کردن باشند؛ یعنی قسمت‌های تعیین شده مانند لینک عمل می‌کنند.

در تگ `img` با استفاده از صفت `usemap`، رابطه بین تصویر و `map` ایجاد می‌گردد؛ از طرفی در تگ `map`، برای تعریف محدوده‌های قابل کلیک از تگ `area` استفاده می‌کنیم.

:Image Map

در تگ area، از تگ‌های shape و coords برای تعیین محدوده‌ها استفاده می‌شود:

– تگ shape: نوع شکل محدوده را مشخص می‌کند.

– تگ coords: مختصات محدوده را بر اساس شکل تعیین شده، مشخص می‌نماید.

	Shape	Coords	
Rectangle:	rect	$x1,y1,x2,y2$	مستطیل:
Circle:	circle	$x,y,radius$	دایره:
Polygon:	poly	$x1,y1,x2,y2,...,xn,yn$	چند ضلعی:
	default	-	کل محدوده تصویر:

:Image Map

```

```

```
<map name="samplemap">
```

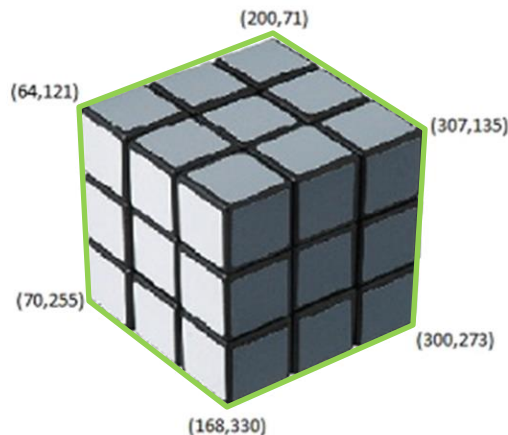
```
<area shape="rect" coords="739,114,1168,312" alt="Keyboard" href="keyboard.html">
```

نحوه استفاده از جاوا اسکریپت در href

```
<area shape="circle" coords="510,198,115" alt="Ball" href="javascript:alert('BALL');">
```

```
<area shape="poly" coords="200,71,307,135,300,273,168,330,70,255,64,121" alt="Cube" href="cube.html">
```

```
</map>
```



:Image Map

صفت download، تگ area:

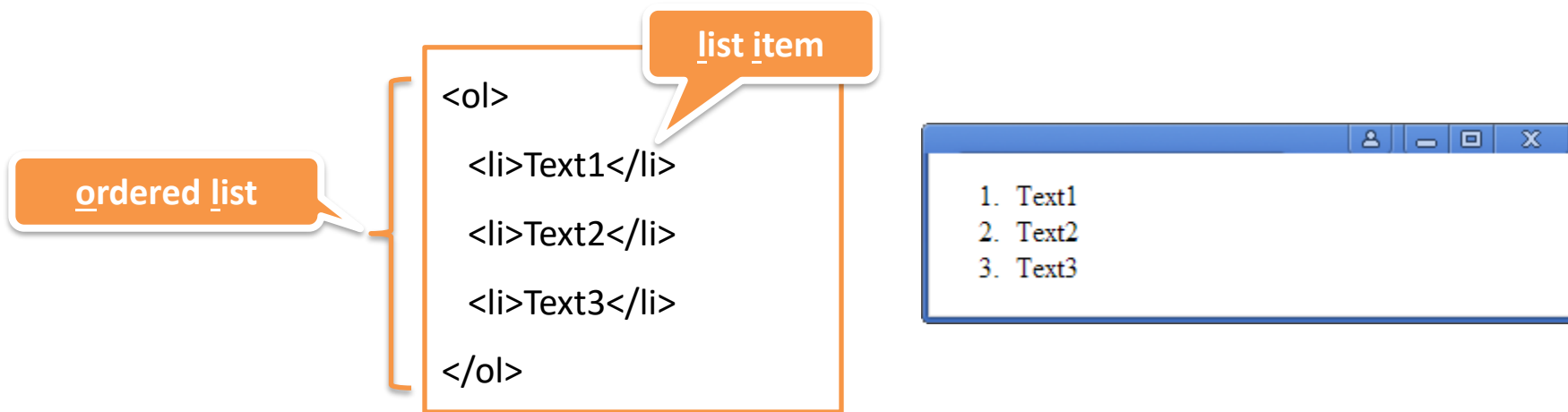
`download="filename"`

با اضافه کردن صفت download بدون مقدار، هنگام کلیک بر روی لینک، فایل مشخص شده در صفت href دانلود می‌شود. حال اگر مقدار *filename* نیز تعیین گردد، فایل با این نام، اما با همان پسوند قبلی دانلود خواهد شد (صفت download در صورتی قابل استفاده است که صفت href نیز تعیین شده باشد).

در این مثال فایل keyboard.jpg با نام KB.jpg دانلود خواهد شد:

```
<area shape="rect" coords="739,114,1168,312" alt="Keyboard" href="keyboard.jpg" download="KB" >
```

لیست‌های مرتب



به صورت پیش فرض یک عدد هر آیتم را مشخص می کند، که این عدد توسط صفت **type** به مقادیر دیگر قابل تغییر است:

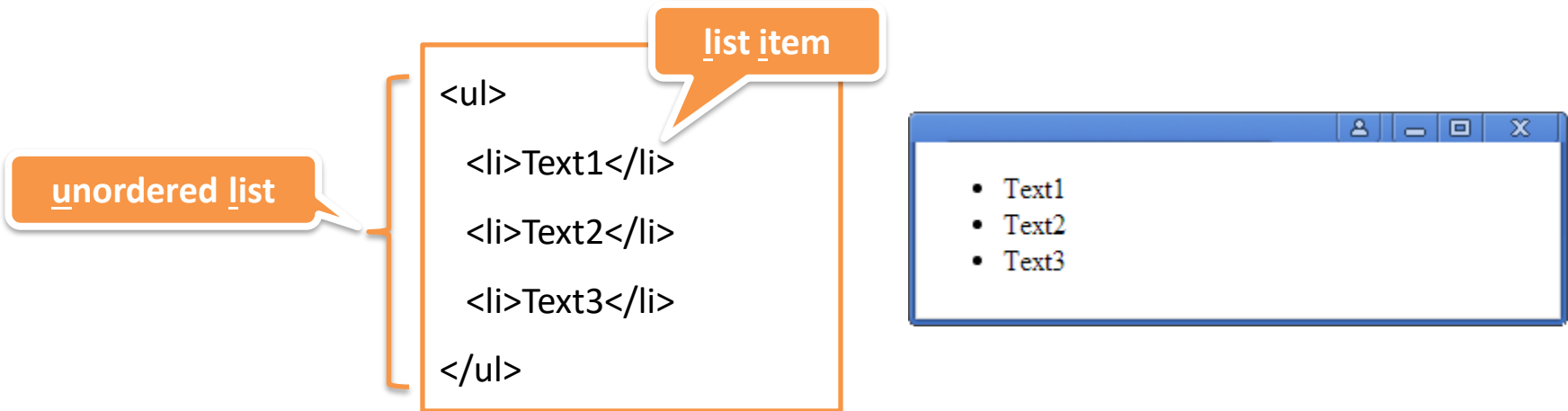
type Attribute
<u>1</u>
A
a
I
i

The diagram also shows the HTML code for an ordered list with a custom starting value:

```
<ol type="1">  
<li>Text1</li>  
<li>Text2</li>  
<li>Text3</li>  
</ol>
```

Two callout boxes point to specific parts of the code: one labeled **مقدار پیش فرض** (Default value) points to the `type="1"` attribute, and another labeled **اعداد رومی** (Roman numerals) points to the `I` value in the table.

لیست‌های نامرتب

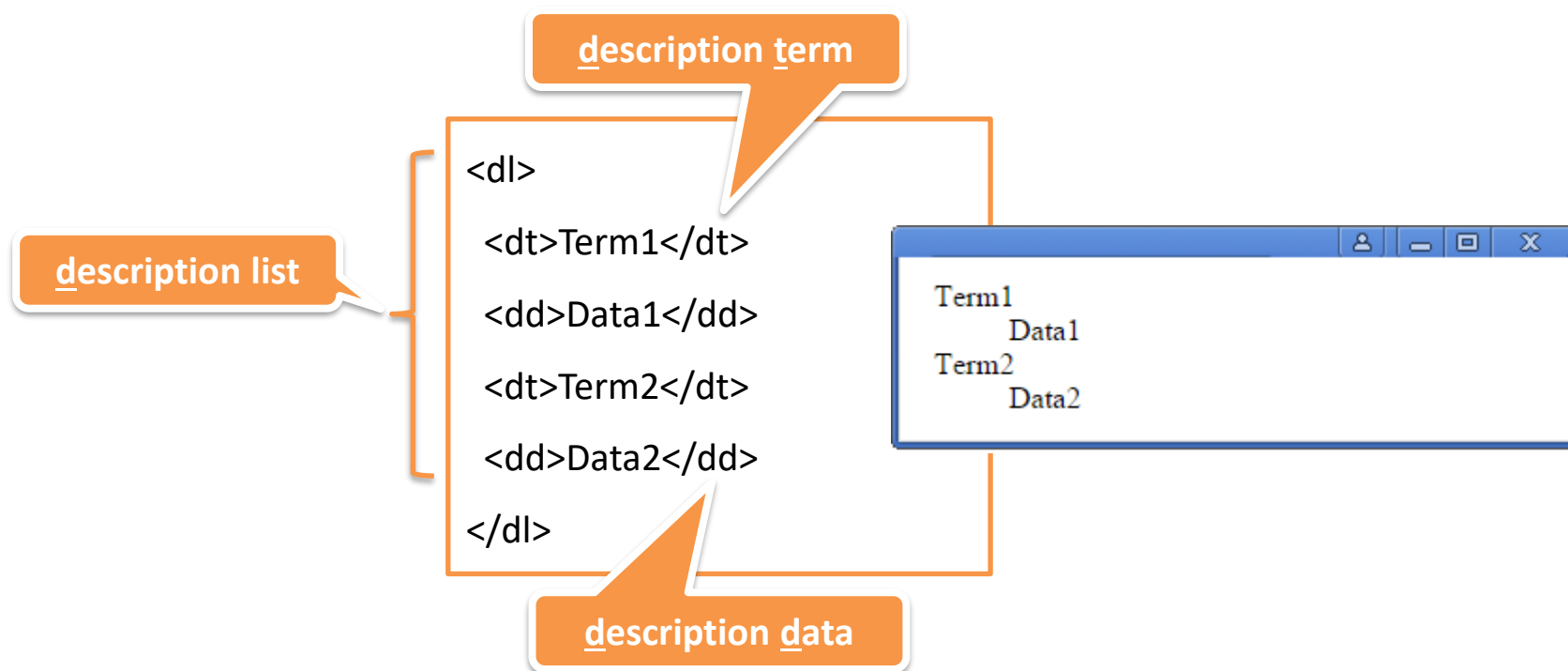


به صورت پیش‌فرض یک دایره کوچک توپر هر آیت‌م را مشخص می‌کند، که این کاراکتر توسط ویژگی `list-style-type` قابل تغییر است:

<pre><ul style="list-style-type:disc"> Text1 Text2 Text3 </pre>	list-style-type Property
	<u>disc</u>
	circle
	square
	none

لیست‌های توصیفی

یک لیست توصیفی، لیستی از اصطلاحات، با یک توضیح برای هر اصطلاح است.



لیست‌های تودرتو

لیست‌ها می‌توانند به صورت تودرتو نیز بکار روند؛ هر آیتم می‌تواند یک لیست دیگر و یا یک عنصر HTML مانند لینک و ... باشد:

```
<h2>HTML Colors</h2>
<ol>
  <li>Black</li>
  <li style="color:Blue">Blue</li>
  <ul>
    <li style="color:AliceBlue">AliceBlue</li>
    <li style="color:BlueViolet">BlueViolet</li>
    <li style="color:CadetBlue">CadetBlue</li>
    <li style="color:CornflowerBlue">CornflowerBlue</li>
    <li style="color:DarkBlue">DarkBlue</li>
    <li style="color:DarkSlateBlue">DarkSlateBlue</li>
    <li style="color:DeepSkyBlue">DeepSkyBlue</li>
    <li style="color:Blue"> ... </li>
  </ul>
  <li style="color:Green">Green</li>
  <li style="color:Red">Red</li>
</ol>
```

HTML Colors

1. Black

2. Blue

- AliceBlue
- BlueViolet
- CadetBlue
- CornflowerBlue
- DarkBlue
- DarkSlateBlue
- DeepSkyBlue
- ...

3. Green

4. Red

لیست‌های افقی

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      ul { padding: 0; }
      ul li { display: inline; }
      ul li a {
        color: white;
        background-color: CornflowerBlue;
        border-radius: 7px 7px 0 0;
        text-decoration: none;
        padding: 9px 15px;
        font-weight: bold;}
      ul li a:hover {
        background-color: Navy; }
    </style>
  </head>
  <body>
    <ul>
      <li><a href="Lesson1.html">Lesson 1</a></li>
      <li><a href="Lesson2.html">Lesson 2</a></li>
      <li><a href="Lesson3.html">Lesson 3</a></li>
    <hr>
    </ul>
  </body>
</html>
```

باعث می‌شود، آیتم‌های لیست در یک سطر نمایش داده شوند.

border-radius

باعث گرد شدن گوشه‌های عنصر می‌شود.

رنگ پس‌زمینه هنگام رفتن نشانگر موس بر روی لینک را تعیین می‌کند.

Lesson 1

Lesson 2

Lesson 3

آیا تگ `<hr>` در جای مناسبی قرار دارد؟



لیست‌های افقی

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      ul#topmenu { padding: 0; }
      ul#topmenu li { display: inline; }
      ul#topmenu li a {
        color: white;
        background-color: CornflowerBlue;
        border-radius: 7px 7px 0 0;
        text-decoration: none;
        padding: 9px 15px;
        font-weight: bold;}
      ul#topmenu li a:hover {
        background-color: Navy; }
    </style>
  </head>
  <body>
    <ul id="topmenu">
      <li><a href="Lesson1.html">Lesson 1</a></li>
      <li><a href="Lesson2.html">Lesson 2</a></li>
      <li><a href="Lesson3.html">Lesson 3</a></li>
    <hr>
    </ul>
  </body>
</html>
```

این استایل فقط بر لیست‌های نامرتبی که صفت id آن‌ها topmenu است، تاثیر می‌گذارد.

Lesson 1

Lesson 2

Lesson 3



جدول

table

عنوان

table_row

table_row

table_row

table_row

```
<table style="width:100%;text-align:center" border="3" caption-side: top;>
  <caption>List of students</caption>
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
  </tr>
  <tr>
    <td>Firstname1</td>
    <td>Lastname1</td>
  </tr>
  <tr>
    <td>Firstname2</td>
    <td>Lastname2</td>
  </tr>
  <tr>
    <td>Firstname3</td>
    <td>Lastname3</td>
  </tr>
</table>
```

table headings

table data

بالا و یا پایین بودن، مکان
عنوان جدول را تعیین می نماید.
مقادیر: top | bottom

بهتر است از CSS استفاده شود.
table, th, td {
border: 1px solid black;
}

Firstname	Lastname
Firstname1	Lastname1
Firstname2	Lastname2
Firstname3	Lastname3

صفت `:border-collapse``border-collapse: separate;`

مقدار پیش فرض

List of students

Firstname	Lastname
Firstname1	Lastname1
Firstname2	Lastname2
Firstname3	Lastname3

`empty-cells: hide;` /* مقدار پیش فرض: show */در حالت `separate` باعث می شود، تا در صورت خالی بودن سلول، حاشیه و پس زمینه آن نمایش داده نشود.`border-collapse: collapse;`

List of students

Firstname	Lastname
Firstname1	Lastname1
Firstname2	Lastname2
Firstname3	Lastname3

صف `colspan`:

List of students

Firstname	Middlename	Lastname
Firstname1	Middlename1	Lastname1
Firstname2		Lastname2
Firstname3	Middlename3	
Firstname4		Text4

`<th colspan="number">`
`<td colspan="number">`

`<tr>`
`<td colspan="2">Firstname2</td>`
`<td>Lastname2</td>`
`</tr>`

`<tr>`
`<td colspan="3">Firstname4</td>`
`<td>Text4</td>`
`</tr>`

صف `rowspan`:

List of students

Firstname	Middlename	Lastname
Firstname1	Middlename1	Lastname1
Firstname2	Middlename2	Lastname2
Firstname3		Lastname3
Firstname4	Middlename4	

`<th rowspan="number">`
`<td rowspan="number">`

`<tr>`
`<td>Firstname2</td>`
`<td rowspan="2">Middlename2</td>`
`<td>Lastname2</td>`
`</tr>`

`<tr>`
`<td>Firstname3</td>`
`<td>Lastname3</td>`
`</tr>`

صفت border:

با استفاده از این صفت می‌توان سبک، اندازه و رنگ حاشیه عناصر را تعیین کرد. تا زمانی که سبک تعیین نشود، تعیین اندازه و رنگ تاثیری ندارد.



```
<style>
p {
  border-width: 1px;
  border-style: solid;
  border-color: black;
}
</style>
```

border-X: AllBorders;
 border-X: top&bottom right&left;
 border-X: top right&left bottom;
 border-X: top right bottom left;

```
<style>
p {
  border: 1px solid black;
}
</style>
```

خلاصه نویسی

border: width style color;

تگ iframes

با استفاده از این تگ می‌توان محتوای یک صفحه وب را در صفحه وب جاری نمایش داد:

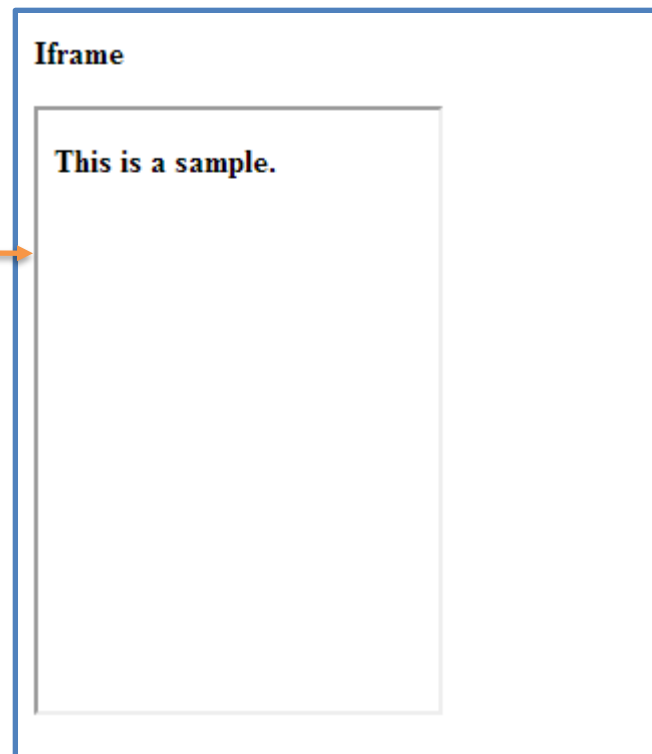
`<iframe src="URL"></iframe>`

```
<!DOCTYPE html>
<html>
  <body>
    <p><b>Iframe</b></p>
    <iframe src="sample.html" width="200" height="350">
    </iframe>
  </body>
</html>
```

پیش فرض:
width="300"
height="150"

Sample.html

```
<!DOCTYPE html>
<html>
  <body>
    <p><b>This is a sample.</b></p>
  </body>
</html>
```



تگ iframe:

می‌توان در لینک‌ها نیز از iframe استفاده کرد، تا با کلیک بر روی آن لینک، نتیجه در iframe مورد نظر نمایش داده شود. برای این کار باید با استفاده از صفت name، یک نام برای iframe مربوطه تعیین کرده و سپس در تگ a لینک مورد نظر، در صفت target، نام آن iframe را قرار دهیم:

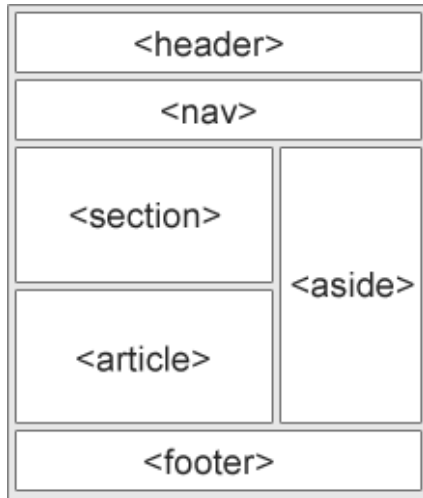
```
<p>  
  <a href="http://montajab.com" target="my_iframe">MONTAJAB.COM</a> |  
  <a href="http://montajab.com/kishuast" target="my_iframe">Kish UAST</a>  
</p>  
<iframe src="sample.html" name="my_iframe" width="550" height="110" style="border:5px double black">  
</iframe>
```



HTML Layout

صفحه آرایی

اغلب وبسایت‌ها محتوای صفحات را در چندین ستون نمایش می‌دهند. در HTML5 عناصر جدیدی جهت تعریف بخش‌های مختلف یک صفحه ارائه شده است، این عناصر معنایی (Semantic) بوده و عبارتند از:



<header>	تعریف یک هدر برای یک سند یا بخش
<nav>	تعریف یک محدوده برای لینک‌ها
<section>	تعریف یک بخش در یک سند
<article>	تعریف یک بخش مستقل
<aside>	تعریف محتوای کنار محتوای اصلی مانند Sidebar
<footer>	تعریف یک فوتر برای یک سند یا بخش
<details>	تعریف جزئیات بیشتر
<summary>	تعریف عنوان برای عنصر <details>

عناصر معنایی، عناصری هستند که می‌توان از نام آن‌ها به وضوح به محتوایشان پی‌برد.

فرم‌های HTML:

از فرم‌های HTML برای دریافت اطلاعات ورودی کاربران استفاده می‌گردد. برای ایجاد یک فرم HTML از تگ form استفاده می‌گردد:

```
<form>
```

```
...
```

```
form elements
```

```
...
```

```
</form>
```

عناصر form انواع مختلفی دارند، که با توجه به چگونگی ورود اطلاعات در بخش‌های مختلف مورد استفاده قرار می‌گردند.

فرم‌های HTML:

عنصر `input`: مهمترین عنصر فرم‌های HTML است، که با استفاده از صفت `type` آن، نوع آن را تعیین می‌کنیم. انواع `input`:

Type	Type Attribute
Text Input	text
Radio Button Input	radio
Checkbox Input	checkbox
The Submit Button	Submit
Password Input	password
...	...
Types: color date datetime datetime-local email file hidden image month number range reset search tel text time url week	

فرم‌های HTML:

عنصر input:

Text Input

type: "text"

یک فیلد ورودی تک سطری تعریف می‌کند:

```
<form>  
First name: <input type="text" name="firstname">  
Last name: <input type="text" name="lastname">  
</form>
```

نام فیلد را تعیین می‌کند.

First name: Last name:

پیش فرض: ۲۰ کاراکتر

فرم‌های HTML:

عنصر input:

Text Input

type: "text"

برخی از صفات عنصر input عبارتند از:

readonly:

عنصر جاری را فقط خواندنی می‌کند و امکان تایپ در آن وجود نخواهد داشت:

```
<input type="text" name="age" value="25" readonly>
```

disabled:

عنصر جاری را غیرفعال می‌کند:

```
<input type="text" name="age" value="25" disabled>
```

فرم‌های HTML:

عنصر input:

Text Input

type: "text"

برخی از صفات عنصر input عبارتند از:

maxlength:

حداکثر تعداد کاراکتر قابل تایپ را مشخص می‌کند:

```
<input type="text" name="age" value="25" maxlength="2">
```

پیش فرض: 524288

placeholder:

عبارت مشخص شده، تا قبل از تایپ اولین کاراکتر، در عنصر نمایش داده می‌شود:

```
<input type="text" name="age" placeholder="Age">
```

فرم‌های HTML:

عنصر input:

The Submit Button

type: "submit"

A rectangular orange button with the word "Submit" in white text.

یک دکمه برای ارسال فرم، به یک Form-Handler، تعریف می‌کند.

Form-Handler معمولاً یک صفحه سمت سرور، با اسکریپتی برای پردازش داده‌های ورودی است، که با استفاده از صفت action در تگ form تعیین می‌گردد. استفاده از دکمه submit روش رایجی برای ارسال فرم به سرور است؛ به طور معمول، فرم به یک صفحه وب (در وب سرور) ارسال می‌گردد.

```
<form action="action_page.asp">  
...  
<input type="submit" value="Submit Form">  
</form>
```

An orange callout box with a white border and a shadow, pointing to the submit button in the code example.

اسکریپت سمت سرور
(مسئول رسیدگی به فرم ارسالی)

فرم‌های HTML:

عنصر input:

The Submit Button

index.html

```
<form action="action_page.asp">  
  First name: <input type="text" name="firstname">  
  Last name: <input type="text" name="lastname">  
  
  <input type="submit" value="Submit Form">  
</form>
```

جهت ارسال صحیح اطلاعات، برای هر فیلد ورودی باید یک نام تعیین کرد؛ در هنگام Submit مقدار فیلد، با این نام، ارسال می‌گردد.

اگر تعیین نگردد، به صورت پیش فرض مقدار Submit بر روی دکمه نمایش داده خواهد شد.

First name: Last name:

با زدن دکمه Submit، مقادیر دو فیلد تعریف شده، به فرم action_page.asp، به شکل زیر ارسال می‌گردند:

... /action_page.asp?firstname=Ali&lastname=M

Get Method (پیش فرض): از طریق URI

در صورت عدم تعریف action، مقدار این صفت، نام فرم جاری شده، و مقادیر فیلدها به همین فرم باز می‌گردند:

... /index.html?firstname=Ali&lastname=M

فرم‌های HTML:

عنصر input:

The Submit Button

Attribute	Description
accept-charset	Specifies the charset used in the submitted form (default: the page charset).
action	Specifies an address (url) where to submit the form (default: the submitting page).
autocomplete	Specifies if the browser should autocomplete the form (default: on).
enctype	Specifies the encoding of the submitted data (default: is url-encoded).
method	Specifies the HTTP method used when submitting the form (default: GET).
name	Specifies a name used to identify the form (for DOM usage: document.forms.name).
novalidate	Specifies that the browser should not validate the form.
target	Specifies the target of the address in the action attribute (default: _self).

فرم‌های HTML:

Form:

صفت method: متد HTTP مورد استفاده در هنگام ارسال فرم را تعیین می‌کند:

GET: متد پیش‌فرض است و در مواقعی که داده ارسالی حاوی اطلاعات حساس و مهمی نباشد، مورد استفاده قرار می‌گیرد، چون این اطلاعات در بخش آدرس صفحه قابل دیدن هستند. این روش برای ارسال داده با مقدار کم مناسب است (محدودیت اندازه‌ی آن به مرورگر بستگی دارد).

پس از ارسال؛ در بخش آدرس، ابتدا نام فایلی که داده را دریافت می‌کند، در ادامه یک علامت سوال و سپس فیلدها که با علامت & از هم جدا شده‌اند می‌آیند؛ هر فیلد با دو بخش نام فیلد و مقدار آن مشخص می‌شود:

.../filename?FieldName1=FiledValue1&FieldName2=FiledValue2

مثال:

.../action_page.asp?firstname=Ali&lastname=M

فرم‌های HTML:

Form:

صفت method: متد HTTP مورد استفاده در هنگام ارسال فرم را تعیین می‌کند:

POST: در مواقعی که داده‌ای ارسال می‌شود، حاوی اطلاعات حساس و مهمی باشد (مانند کلمه عبور)، مورد استفاده قرار می‌گیرد، چون این اطلاعات در بخش آدرس صفحه نمایش داده نمی‌شوند، در نتیجه امنیت بیشتری خواهند داشت.

```
<form action="action_page.asp" method="post" >  
...  
</form>
```

تفاوت‌های متد GET و POST را بیان نمایید. 

فرم‌های HTML:

عنصر input:

Password

type: "password"

یک فیلد برای کلمه عبور ایجاد می‌کند، در نتیجه مقدار تایپ شده به صورت ستاره و یا دایره توپر نمایش داده می‌شود، تا توسط افراد دیگر قابل مشاهده نباشد:

```
<form action="login.asp">  
  User name: <input type="text" name="username">  
  Password: <input type="password" name="pass">  
  <input type="submit" value="Login">  
</form>
```

User name: Password:

فرم‌های HTML:

عنصر input:

Radio Button Input

type: "radio"

امکان انتخاب یک گزینه از بین گزینه‌های تعریف شده را می‌دهد:

<form>

First name: <input type="text" name="firstname">

Last name: <input type="text" name="lastname">

Gender:

<input type="radio" name="gender" value="male" checked> Male

<input type="radio" name="gender" value="female"> Female

</form>

صفت name گزینه‌های یک گروه از انتخاب‌ها باید یکسان باشد.

انتخاب پیش فرض

First name: Last name: Gender: ☒ Male ☐ Female

مقدار هر گزینه با استفاده از صفت value تعیین می‌شود، این مقدار نمایش داده نمی‌شود.

فرم‌های HTML:

عنصر input:

Checkbox Input

type: "checkbox"

امکان انتخاب یک (یا چند) گزینه را می‌دهد و مقدار آن بولین است:

<form>

...

Webskills:

<input type="checkbox" name="webskills" value="HTML" checked>HTML

<input type="checkbox" name="webskills" value="CSS" checked>CSS

<input type="checkbox" name="webskills" value="JavaScript">JavaScript

</form>

مقدار هر گزینه با استفاده از صفت value تعیین می‌شود، این مقدار نمایش داده نمی‌شود.

انتخاب پیش فرض

First name: Last name: Gender: ☒ Male ☐ FemaleWeb Skills: ☒ HTML ☒ CSS ☐ JavaScript

Submit:

...?webskills=HTML&webskills=CSS

فرم‌های HTML:

عنصر input:

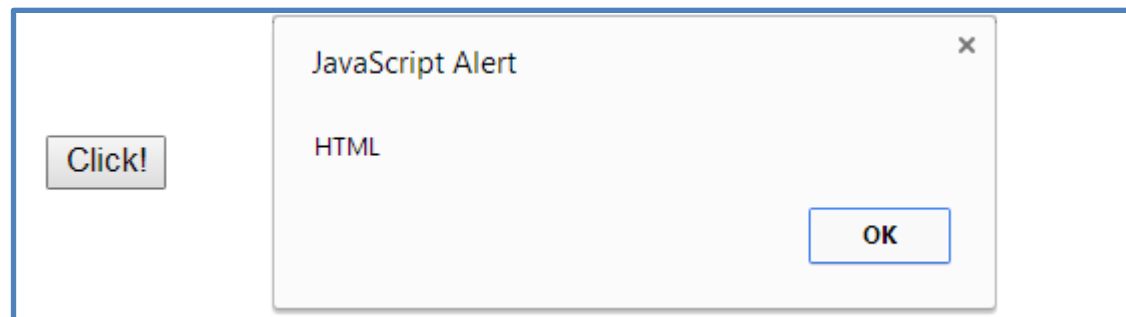
Button

type: "button"

یک دکمه ایجاد می‌کند:

onclick رخدادی است که هنگام کلیک شدن عنصر (دکمه)، فراخوانی می‌شود. در اینجا بعد از کلیک دکمه دستور جاوا اسکریپتی **alert** اجرا می‌گردد.

```
<input type="button" onclick="alert('HTML')" value="Click!">
```



فرم‌های HTML:

عنصر input:

type:	<u>button</u>	<u>password</u>
	<u>checkbox</u>	<u>radio</u>
	color	 range
	date	reset
	datetime	 search
	datetime-local	<u>submit</u>
	email	 tel
	file	<u>text</u>
	hidden	 time
	image	 url
	month	 week
	number	

فرم‌های HTML:

عنصر select:

Drop-Down List

هر مقدار در لیست، با تگ option تعریف می‌گردد.

Country Code:

```
<select name="countrycode">
  <option value="55">Brazil</option>
  <option value="1">Canada</option>
  <option value="98" selected>Iran</option>
  <option value="39">Italy</option>
  <option value="34">Spain</option>
</select>
```

به عنوان نمونه، پس از انتخاب Iran، مقداری که در صفت value برای آن تعریف شده (یعنی ۹۸) در هنگام ارسال فرم، با نام countrycode ارسال می‌گردد:
...?countrycode=98

Country Code:

Iran ▼
Brazil
Canada
Iran
Italy
Spain

به صورت پیش فرض اولین مقدار به صورت انتخاب شده نمایش داده می‌شود، مگر این که با صفت selected مقدار دیگری به صورت پیش فرض انتخاب شده باشد.

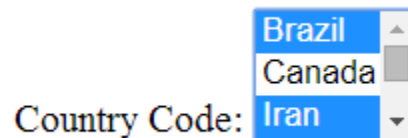
فرم‌های HTML:

عنصر select:

Drop-Down List

Country Code:

```
<select name="countrycode" size="3" multiple>
  <option value="55">Brazil</option>
  <option value="1">Canada</option>
  <option value="98">Iran</option>
  <option value="39">Italy</option>
  <option value="34">Spain</option>
</select>
```



تعداد سطر قابل مشاهده‌ی لیست را تعیین می‌کند:
مقدار پیش‌فرض در حالتی که multiple وجود داشته باشد ۴ سطر و در غیراین صورت، یک سطر خواهد بود.

امکان انتخاب چند گزینه را فراهم می‌کند:
در Windows، با نگاه داشتن کلید Ctrl و انتخاب گزینه‌ها
در Mac، با نگاه داشتن کلید Command و انتخاب گزینه‌ها

به عنوان نمونه، پس از انتخاب Iran و Brazil، در هنگام ارسال فرم:

...?countrycode=55&countrycode=98

فرم‌های HTML:

عنصر **datalist**: لیستی از مقادیر از پیش تعریف شده را برای عنصر **input** تعیین می‌کند.

Data List

```
<form method="get">
```

```
Country: <input list="countries" name="country">
```

```
<datalist id="countries">
```

```
<option value="Brazil ">
```

```
<option value="Canada">
```

```
<option value="Iran">IR</option>
```

```
<option value="Italy">
```

```
<option value="Spain">
```

```
</datalist>
```

```
<input type="submit">
```

```
</form>
```

صفت **list** ارتباط با **datalist** را
از طریق مقدار **id** برقرار می‌کند.

...?country=Iran

Country:

- Brazil
- Canada
- Iran
- Italy
- Spain

Autocomplete

Country:

- Iran
- Italy

فرم‌های HTML:

عنصر textarea:

Text Area

یک فیلد ورودی چند سطرری تعریف می‌کند، که تعداد کاراکتر آن محدودیتی نداشته و عرض کاراکترهای آن به صورت پیش‌فرض یکسان است (معمولا فونت Courier):

Learn:

<textarea rows="4" cols="50">

Learning HTML HTML & CSS, Learning HTML HTML & CSS
Learning HTML HTML & CSS, Learning HTML HTML & CSS
Learning HTML HTML & CSS, Learning HTML HTML & CSS

</textarea>

تعداد سطرها و ستون‌ها را تعیین می‌کند.

بهتر است از CSS (ویژگی‌های height و width) استفاده شود.

Learn:

Learning HTML HTML & CSS, Learning HTML HTML & CSS
Learning HTML HTML & CSS, Learning HTML HTML & CSS
Learning HTML HTML & CSS, Learning HTML HTML & CSS

فرم‌های HTML:

عنصر textarea:

Text Area

صفت `placeholder`، با یک متن کوتاه، مقداری که باید در `textarea` نوشته شود را توصیف می‌کند. بعد از شروع کردن به ورود اطلاعات، این مقدار حذف می‌شود.

Learn:

<textarea placeholder="Learning ..." maxlength="50">

</textarea>

حداکثر تعداد کاراکتر

Learn:

Learning ...

فرم‌های HTML:

گروه‌بندی عناصر فرم:

عنصر `fieldset` داده مرتبط در یک فرم را گروه‌بندی می‌کند، همچنین با استفاده از عنصر `legend` می‌توان یک عنوان برای آن تعریف کرد:

```
<fieldset>  
  <legend>Information</legend>  
  First name: <input type="text" name="firstname" value="First">  
  Last name: <input type="text" name="lastname" value="Last">  
  <input type="submit" value="Submit">  
</fieldset>
```

Information

First name: Last name:

فایل‌های چند رسانه‌ای در HTML5:



فایل‌های صوتی، تصویری (Multimedia) پشتیبانی شده در استاندارد HTML5 عبارتند از:

Sound Format	Type
MP3	audio/mpeg
WAV	audio/wav
Ogg	audio/ogg

Video Format	Type
MPEG-4	video/mp4
WebM	video/webm
Ogg	video/ogg

از عناصر audio و video جهت پخش فایل‌های صوتی و ویدیویی استفاده می‌گردد. صفت controls، کنترل‌های مانند Play، Pause و Volume را اضافه می‌کند:

```
<audio controls>  
...  
</audio>
```

```
<video controls>  
...  
</ video>
```

فایل‌های چند رسانه‌ای در HTML5:

عنصر `source`، با استفاده از صفت `src`، URI فایل مورد نظر و با استفاده از صفت `type`، نوع آن را تعیین کند:

Audio

```
<audio controls autoplay loop>
  <source src="sample.mp3" type="audio/mpeg">
  <source src="sample.ogg" type="audio/ogg">
  Your browser does not support the audio element.
</audio>
```

Video

```
<video width="320" height="240" autoplay>
  <source src="sample.mp4" type="video/mp4">
  <source src="sample.ogg" type="video/ogg">
  Your browser does not support the video element.
</video>
```

می‌توان چندین عنصر `source` تعریف نمود، اما مرورگر اولین فرمتی که تشخیص دهد را در نظر می‌گیرد. در تگ‌های `audio` و `video`، می‌توان یک متن قرار داد، این متن در زمانی که مرورگر این تگ‌های را شناسد به کاربر نمایش داده می‌شود.

صفت `autoplay` باعث می‌شود تا فایل (صوتی/ویدیویی) به صورت خودکار پخش شود. صفت `loop` باعث می‌شود تا فایل (صوتی/ویدیویی) تکرار شود.

برخی از کاراکترهای خاص در HTML:

Char	Entity Number	Entity Name	Description
	 	 	non-breaking space
<	<	<	less than
>	>	>	greater than
&	&	&	ampersand
¢	¢	¢	cent
£	£	£	pound
ریال	﷼		rial
€	€	€	euro
©	©	©	copyright
®	®	®	registered trademark

برخی از نمادهای ریاضی در HTML:

Char	Entity Number	Entity Name	Description
$\forall$	∀	∀	for All
∂	∂	∂	partial differential
$\exists$	∃	∃	there exists
$\emptyset$	∅	∅	empty sets
$\int$	∫	∫	integral
$\in$	∈	∈	element of
$\notin$	∉	∉	not an element of
$\ni$	∋	∋	contains as member
$\prod$	∏	∏	n-ary product
$\sum$	∑	∑	n-ary summation

برخی از Emoji ها در HTML:

Char	Entity Number	Char	Entity Number
	😃		🌲
	😄		🍎
	⌚		🎀
	☕		🎖
	⚓		🏳
	⛅		💭
	✔		💻
	✖		🛒
	🆓		🛠
	🌎		📽

اعتبارسنجی کدهای Markup:

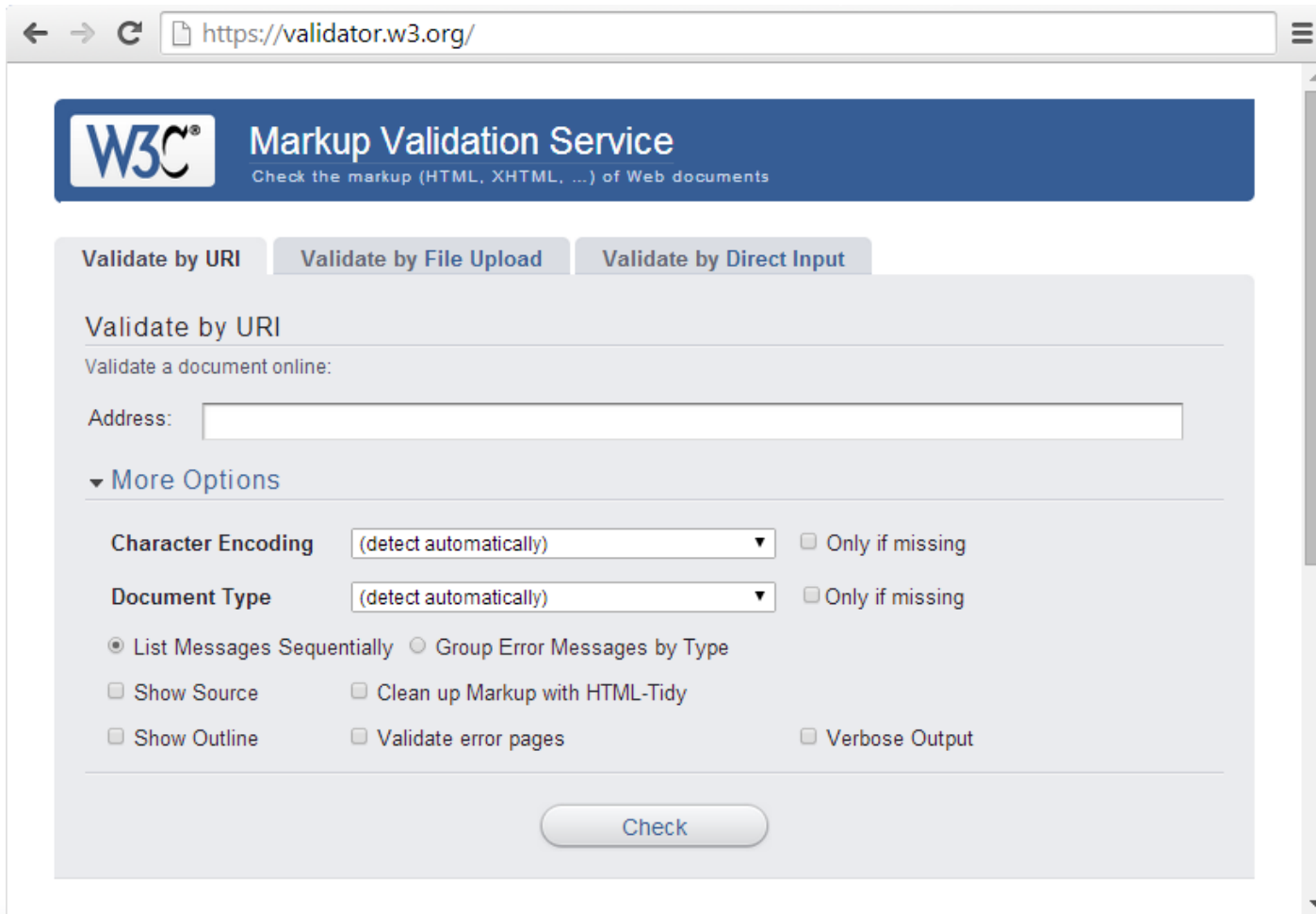
جهت بررسی درستی کدهای HTML، می‌توان از سرویس اعتبارسنجی کنسرسیوم جهانی وب (W3C) که در آدرس <https://validator.w3.org> قرار دارد، استفاده نمود. بعد از وارد کردن کدهای HTML به یکی از سه روش موجود (از طریق URI، بارگذاری فایل و یا به صورت مستقیم)، و در صورت نیاز انجام تنظیمات، بررسی انجام شده و نتیجه به صورت مفصل ارائه می‌گردد.

امکان اعتبارسنجی کدهای CSS نیز در آدرس <https://jigsaw.w3.org/css-validator> توسط W3C فراهم آمده است. امکانات دیگری در ابزار توسعه W3C وجود دارد، که از طریق آدرس زیر قابل دستیابی هستند:

W3C Developer Tools:

<https://www.w3.org/developers/tools>

اعتبارسنجی کدهای Markup:



The screenshot shows the W3C Markup Validation Service interface in a web browser. The address bar displays `https://validator.w3.org/`. The page header features the W3C logo and the title "Markup Validation Service" with the subtitle "Check the markup (HTML, XHTML, ...) of Web documents".

There are three tabs for validation: "Validate by URI" (selected), "Validate by File Upload", and "Validate by Direct Input".

Under the "Validate by URI" tab, the text "Validate a document online:" is followed by an "Address:" label and a text input field.

Below the input field is a section titled "More Options" with a dropdown arrow. It contains several settings:

- Character Encoding:** A dropdown menu set to "(detect automatically)" with a checkbox for "Only if missing".
- Document Type:** A dropdown menu set to "(detect automatically)" with a checkbox for "Only if missing".
- Message Display:** Radio buttons for "List Messages Sequentially" (selected) and "Group Error Messages by Type".
- Additional Options:** Checkboxes for "Show Source", "Clean up Markup with HTML-Tidy", "Show Outline", "Validate error pages", and "Verbose Output".

At the bottom of the form is a "Check" button.