

مترجم:
عبدالصمد کرامت فر
طراح : ویدا غلامی



HTML

آموزش

کاري اختصاصي از کتابخانه ی:

گویا آیتي

مقاوت- ترين مجله اینترنتی ...



book.gooyait.com

آموزش HTML

عنوان کتاب

عبدالصمد کرامت فر

مترجم

Page | 2

ویدا غلامی

طراح جلد و صفحه آرا

محمدرضا رشیدی

ویراستار



هرگونه دخل و تصرف در محتوای کتاب و همچنین سوء استفاده از مطالب، شرعا و قانونا جایز نیست.



فهرست مطالب

۷..... ۱.۱ تاریخچه

۸..... ۱.۲ شروع به کار

۹..... ۲. ساختار اساسی

۹..... ۲.۱ عناصر و صفات

۱۰..... ۲.۲ ساختار یک سند HTML:

۱۱..... ۲.۳ سرفصل‌ها و پاراگراف‌ها

۱۲..... ۲.۴ برخی از صفات عمومی

۱۲..... ۲.۵ افزودن تاکید

۱۳..... ۲.۶ مدل لایه

۱۴..... ۲.۷ شکست خطوط

۱۴..... ۳. لیست‌ها

۱۴..... ۳.۱ لیست‌های مرتب و نامرتب

۱۶..... ۳.۲ لیست‌های تعریف

۱۷..... پیوندهای

۱۷..... ۴.۱ منبع لینک

۱۸..... ۴.۲ ریز مقصدها^۱

۱۹..... ۴.۳ عنصر base

۱۹..... ۵. عناصر بلوکی دیگر

۱۹..... ۵.۱ ترسیم خط افقی

۲۰..... ۵.۲ عنصر عمومی div

۲۰..... ۶. عناصر درون خطی دیگر

۲۰..... ۶.۱ عناصر عبارتی مهم

۲۲..... ۶.۲ توان و اندیس

۲۲..... ۶.۳ عنصر span

۲۳..... ۷. تصاویر

۲۳..... ۷.۱ عنصر img

۲۳..... ۷.۲ عنصر object

۲۴..... ۷.۳ نقشه تصویری

¹ . Fragment Destinations

۲۶.....جدولها ۸

۲۶.....مقدمه ۸.۱

۲۶.....ساختار ۸.۲

Page | 5

۲۹.....فرمها ۹

۲۹.....مقدمه ۹.۱

۳۰.....پردازش فرم ۹.۲

۳۳.....عنصر input ۹.۳

۳۷.....selection ها ۹.۴

۴۰.....ناحیه متنی ۹.۵

۴۱.....عنصر button ۹.۶

۴۱.....سررسند^۲ ۱۰

۴۱.....مقدمه ۱۰.۱

². Document Head

۴۱.....عنوان ۱۰.۲

۴۱.....meta ۱۰.۳

۴۴.....اسکرپت نویسی ۱۰.۴

۴۶.....style ۱۰.۵

۴۶.....link ۱۰.۶

۴۷.....link ۱۰.۷

۴۸.....base ۱۰.۸

۱.۱ تاریخچه

زبان نشانه‌گذاری فرامتن (HTML) - Hyper Text Markup Language - زبان نشانه‌گذاری اصلی صفحات وب است که در سال ۱۹۹۰ توسط تیم‌برنرزیلی^۳ با عنوان روش نشانه‌گذاری صفحات اطلاعات به نحوی که با مرورگر قابل مشاهده باشند، معرفی شد. نسخه‌های مختلف آن طی سالیان گذشته ارائه شده‌است:

- ۱) HTML 2.0: اولین تعریف واقعی با ویژگی‌های اصلی HTML کنونی در سال ۱۹۹۴.
- ۲) HTML 3.2: اولین پیشنهاد W3C^۴ برای HTML در سال ۱۹۹۶. ویژگی‌های این نسخه افزوده شدن جدول‌ها، اپلت‌ها، متون جاری پیرامون تصاویر، بالا و پایین‌نگاشت‌ها و سازگاری آن با نسخه قبلی بود.
- ۳) HTML 4.0: با سازوکار صفحه‌شيوه‌ها، اسکریپت‌ها، فریم‌ها، تعبیه آبجکت‌ها توسعه پیدا کرد. پشتیبانی بهتری از متون راست به چپ و متونی که هم از چپ به راست و هم راست به چپ هستند، جدول‌های بهتر، توسعه فرم‌ها فراهم کرد.
- ۴) HTML 4.01: در بیست و چهارم دسامبر ۱۹۹۹ و با رفع برخی از اشکالات نسخه قبلی ارائه شد.
- ۵) XHTML 1.0: شکل جدیدی از HTML 4 در XML که در سال ۲۰۰۰ ارائه شد.
- ۶) XHTML 1.1: تعریف دیگری از XHTML 1.0 به عنوان مجموعه‌ای از مدل‌ها که در سال ۲۰۰۱ پیشنهاد شد. در این نسخه استواری و آینده‌نگری استاندارد وجود داشت و ایرادات نسخه‌های قبلی برطرف شده بود. توسعه‌ای از نسخه ۴.۰۱ بوده و بخشی از XHTML 1.0 است.
- ۷) Xhtml Basic: در سال ۲۰۰۰ ارائه و شامل یک مجموعه کوچک از XHTML 1.0 (در اصطلاح مدل‌های XHTML) که بوسیله طیفی از مرورگرهای تلفن همراه و پی‌دی‌ای‌ها پشتیبانی می‌شد.
- ۸) XHTML 2.0: نسخه بعدی HTML که در سال ۲۰۰۳ ارائه شد.

³ . Tim Berners-Lee

⁴ . World Wide Web Consortium

۹) HTML 5: در ژانویه ۲۰۰۸ کنسرسیوم جهانی وب پیش‌نویس کار بر روی این نسخه را منتشر کرد. برخی از ویژگی‌های HTML 5 تعبیه عواملی برای صدا، فیلم، گرافیک، ذخیره‌سازی داده در سمت کاربر و وجود اسناد تعاملی هستند.

۱.۲ شروع به کار

برنامه‌نویسان وب برای طراحی صفحات وب آن‌چه را که می‌خواهند در صفحه نمایش دهند، در میان تگ‌های این زبان قرار می‌دهند. مرورگر کدهای اچ‌تی‌ام‌ال را از سرور دریافت، ترکیب و تفسیر کرده و در نهایت به شکل یک صفحه کامل حاوی تصاویر و سایر اجزا در اختیار کاربر قرار می‌دهد. یکی از اشتباهات رایج در معرفی این زبان، آنست که اغلب از آن به عنوان یک زبان برنامه‌نویسی تعبیر می‌کنند، اما در واقع HTML یک زبان برنامه‌نویسی نیست بلکه همان‌طور که از نام آن نیز برمی‌آید یک زبان نشانه‌گذاری است یعنی برای تمایز اجزای مختلف صفحه از تگ‌های مشخصی استفاده می‌کند. برای ایجاد صفحات وب پویا و برنامه‌هایی که تحت وب عمل می‌کنند، برنامه‌نویسان وب زبان‌های برنامه‌نویسی شامل برنامه‌های سمت سرور و سمت کاربر، که در میان کدهای اچ‌تی‌ام‌ال استفاده می‌شوند را به کار می‌برند.

به عنوان مثال برای استفاده از کدهای زبان PHP که یک زبان برنامه‌نویسی سمت سرور است، تگ شناخته شده توسط اچ‌تی‌ام‌ال به این شکل است:

```
<?php
```

PHP Codes

```
?>
```

تگ شروع و پایان نوشته شده به مرورگر می‌گوید که از اینجا به بعد کدهای پی‌اچ‌پی آمده و باید اقدامات لازم را انجام دهد.

برای ایجاد صفحات اچ تی ام ال نیاز به یک ویرایشگر ساده متن مانند NotePad دارید، اما اگر بخواهید صفحات حرفه‌ای‌تری ایجاد کنید می‌توانید از ⁵WISYWIG ها استفاده کنید. این نرم‌افزارها با ارائه یک واسط گرافیکی و امکانات دیگر سرعت و دقت ایجاد صفحات را افزایش می‌دهند. از بهترین نمونه‌های آن‌ها می‌توان به FrontPage و DreamViwer اشاره کرد.

همین حالا NotePad را باز کرده و بنا بر این مقاله شروع به نوشتن صفحات وب کنید. پس از نوشتن کدها، صفحه را با پسوند .html ذخیره کنید، فایل ایجاد شده را بوسیله مرورگر باز کنید؛ این اولین صفحه وب ساخته شده به دست شماست!

لازم به ذکر است که این دستنامه بر مبنای XHTML 2.0 نوشته شده‌است.

۲. ساختار اساسی

۲.۱ عناصر و صفات

اطلاعات در HTML با افزودن نشانه متمایز می‌شوند که این نشانه‌ها ساختار آن را تشکیل می‌دهد. اطلاعات نشانه‌گذاری شده، متشکل از مجموعه‌ای از عناصر هستند که دارای شکل عمومی زیرند:

<xyz>Some content</xyz>

این بند یک عنصر فرضی با نام **xyz** تعریف می‌کند. تگ شروع **<xyz>** و تگ پایان **</xyz>** است.

محتوای عنصر متنی است که باید نشانه‌گذاری شود، البته خود می‌تواند شامل عناصر دیگری باشد:

<xyz>Some

<abc>more</abc>content<def>added</def></xyz>

تگ‌ها باید به شکل صحیح در هم ترکیب شوند، مثال زیر یک شکل نادرست است:

⁵ .What You See Is What You Get

```
<xyz>Some <abc>more content<def>added</abc> </def>  
</xyz>
```

عنصر **def** یا باید به طور کامل درون **abc** قرار بگیرد یا به طور کامل خارج از آن.

عناصر ممکن است دارای صفات باشند که بخشی از تگ شروع را تشکیل می‌دهند:

```
<xyz attr1="sometext" attr2='somemoretext'>Some  
content</xyz>
```

هر صفت متشکل است از نام صفت، علامت مساوی و مقداری که معمولاً یک رشته متنی است و داخل گیومه تک یا دوگانه قرار می‌گیرد.

۲.۲ ساختار یک سند HTML:

تمامی کدها بین `<html>` و `</html>` قرار می‌گیرد، کل صفحه به دو قسمت سرصفحه و بدنه تقسیم شده، که عنوان و فراداده‌ها - اطلاعاتی که سند را توصیف می‌کنند- در قسمت سر و محتوای سند در قسمت بدنه قرار می‌گیرند، هر سند دارای یک بخش ثابت است که وظیفه بیان نوع سند را بر عهده دارد و کمک می‌کند تا سند بهتر اجرا شود:

```
<?xml version="1.0" encoding="iso-8859-1" ?>  
  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  
  
<html>  
  
<head>  
  
<title>  
  
Title of Page Come Here
```

</title>

</head>

<body>

Other Elements Come Here

</body>

</html>

۲.۳ سرفصل‌ها و پاراگراف‌ها

h1 h2 h3 h4 h5 h6 p

سرفصل‌ها در HTML بوسیله h1-6 ایجاد می‌شوند:

- <h1>Heading1</h1>
- <h2>Heading2</h2>
- <h3>Heading3</h3>
- <h4>Heading4</h4>
- <h5>Heading5</h5>
- <h6>Heading6</h6>

و پاراگراف‌ها:

<p>This is a Paragraph.</p>

۲.۴ برخی از صفات عمومی

`<h1 id= class= . . . onclick= onmousedown= . . . >`

صفات عمومی صفاتی هستند که تقریباً در تمام عناصر کاربرد دارند؛ Page | 12

- **id:** نام منحصر به فردی که بر یک عنصر گذارده می‌شود، دقت کنید که این نام تنها و تنها بر یک عنصر تعلق می‌گیرد. کاربرد آن در ایجاد لینک‌ها، شیوه‌ها، و تعامل‌هاست. بعداً در رابطه به موارد مذکور بحث خواهد شد.
- **Class:** کاربرد اصلی، تخصیص شیوه‌ها از طریق آن‌ها به عناصر است. می‌توان بیش از یک `class` به هر عنصر افزود.
- از دیگر صفات عمومی می‌توان به **Style** و **Title** اشاره کرد.

۲.۵ افزودن تاکید

em strong

گاهی اوقات در متون نکاتی وجود دارد که می‌خواهیم بر آن‌ها تاکید کنیم. در HTML عناصری وجود دارند که مسئولیت این کار را بر عهده دارند.

- **em:** این عنصر متن داخل خود را مورد تاکید قرار می‌دهد.
- **Strong:** تاکید بیش‌تری بر متن محصور ایجاد می‌کند.

```
<p>This is a <em>Point</em> and This is more  
<strong>Important.</strong></p>
```

This is a *Point* and This is more **Important**.

در HTML چندین روش برای نحوه نمایش عناصر در اسناد وجود دارد. سرفصل‌ها باید در یک خط جداگانه ظاهر شوند. اگر دو پاراگراف داشته باشیم، باید به طور طبیعی پاراگراف اول قبل از پاراگراف دوم ظاهر شده و باید انتهای پاراگراف اول و ابتدای پاراگراف دوم مشخص باشد. برای تاکید، باید مشخص باشد که کلمات موکد، پاراگراف مستقل نیستند بلکه کلمات موکد درون پاراگراف هستند.

این مهم در HTML با اعلام نوع عنصر به مرورگر محقق می‌شود. عناصر به دسته تقسیم می‌شوند:

- درون خطی^۶: عناصری که فقط شامل متن یا دیگر عناصر درون خطی هستند. عناصر `em` و `strong` درون خطی هستند.

بلوکی: عناصری که دارای اهمیت بیش‌تری هستند، می‌توانند دامنه‌ی خود را تعیین کرده و می‌توانند شامل عناصر درون خطی باشند. عناصر `p` و `h1` نمونه‌های عناصر بلوکی هستند. بعضی عناصر بلوکی هستند که بیش‌تر شبیه به جعبه^۷ هستند تا عناصر بلوکی منفرد که می‌توانند محتوی عناصر بلوکی دیگر باشند. در این مورد ساختار درونی می‌تواند پیچیده‌تر باشد.

از مرورگر انتظار می‌رود که این دو دسته را متمایز نمایش دهد. بر خلاف عناصر درون خطی، عناصر بلوکی باید از یک خط جدید شروع شوند. هنگام بحث در شیوه‌ها، در این زمینه مسائل پیچیده‌تری مطرح می‌شود اما در اینجا توصیف نحوه نمایش عناصر توسط مرورگر کافی است. پیچیدگی‌ها مربوط به زمانی است که عناصر بلوکی درون خود عناصر بلوکی دیگری را جای دهند. می‌توان گفت که عنصر `body` یک عنصر بلوکی است و تا این‌جا دیده‌ایم که عناصر بلوکی دیگر می‌توانند در آن قرار گیرند.

⁶ . Inline

⁵ . Block-Level

⁷ . Container

عنصری وجود دارد که برای تعریف نحوه نمایش دیگر عناصر به کار می‌روند. این عناصر از عناصر بلوکی نبوده و بنابراین در بحث مدل لایه قرار نمی‌گیرند. همچنین عناصری وجود دارد که می‌توانند در هر جا ظاهر شوند (مانند اسکریپت‌ها) اما کاربرد سراسری دارند. نوشتن آن‌ها در یک مکان خاص سند فقط برای تسهیل خوانایی است.

۲.۷ شکست خطوط

{
}

اگر در بین یک پاراگراف نیاز به انتقال به خط بعد باشد از عنصر درون خطی **br** استفاده می‌شود. برای مثال:

<p>It is a Paragraph with
Line Breaks</p>

It is a Paragraph with Line Breaks.

۳. لیست‌ها

۳.۱ لیست‌های مرتب و نامرتب

ul ol li

دو نوع از لیست‌های معمول در اسناد، لیست‌های مرتب و نامرتب هستند.

- **ul** (Unorder Lists): لیست‌های نامرتب؛ که در آن‌ها همه‌ی موارد دارای اهمیت بوده و ترتیب در آن‌ها اهمیت ندارد.

- **ol** (Order Lists): مجموعه‌ای از موارد که در آن‌ها ترتیب اهمیت دارد.

li: تنها عنصری است که امکان استفاده از آن درون عناصر **ul** و **ol** وجود دارد و برای تعریف هر یک از آیتم‌ها در لیست به کار می‌رود.

Teams in Group 1

```
<ul>
```

```
<li>Iran</li>
```

```
<li>Brazil</li>
```

```
<li>Italy</li>
```

```
<li>Africa</li>
```

```
</ul>
```

Qualified Teams from Group 1

```
<ol>
```

```
<li>Iran</li>
```

```
<li>Brazil</li>
```

```
</ol>
```

Teams in Group 1

- Iran
- Brazil
- Italy
- Africa

Qualified Teams from Group 1

- 1) Iran
- 2) Brazil

dl dt dd

dl (Defintion List): نوع دیگری از لیست‌ها در HTML ، لیست هایی هستند که به منظور تعریف موارد بکار می‌روند، یعنی حاوی عنوان آیتم و تعریف آن هستند. از دو عنصر dt لغت مورد نظر و dd تعریف عنصر تشکیل شده است.

```
<p>The following definitions apply:</p>
```

```
<dl>
```

```
<dt>Unordered List</dt>
```

```
<dd>List in any order that has a bullet in front of  
it</dd>
```

```
<dt>Ordered List</dt>
```

```
<dd>List in a precise order that are numbered</dd>
```

```
</dl>
```


۴.۱ منبع لینک

<h1 id= >

مهم‌ترین ویژگی وب امکان لینک دادن بین صفحات و حتی بخش‌های مختلف یک صفحه است. عنصر **a** وظیفه ایجاد این ارتباط را برعهده دارد. صفت **href** منبع لینک را به عنوان مقدار دریافت می‌کند. یک عنصر در درون یک صفحه از طریق صفت **id** به عنوان مقصد لینک تعریف می‌شود. یک لینک توسط مرورگرها به شیوه‌های مختلف متمایز می‌شود. به عنوان مثال در زیر لینک به صورت آندرلاین نمایش داده شده است.

For More Information See Our Page

[For More Information See Our Page](#)

مقدار دریافتی آن **URL (Uniform Resource Locator)** صفحه‌ای است که می‌خواهیم با آن پیوند برقرار کنیم که می‌تواند یک مقدار نسبی یا یک مقدار مطلق باشد. مقدار مطلق به صورت کامل دربردارنده آدرس است در حالی که مقدار نسبی به نسبت موقعیت ما در دایرکتوری سایت تعیین می‌شود.

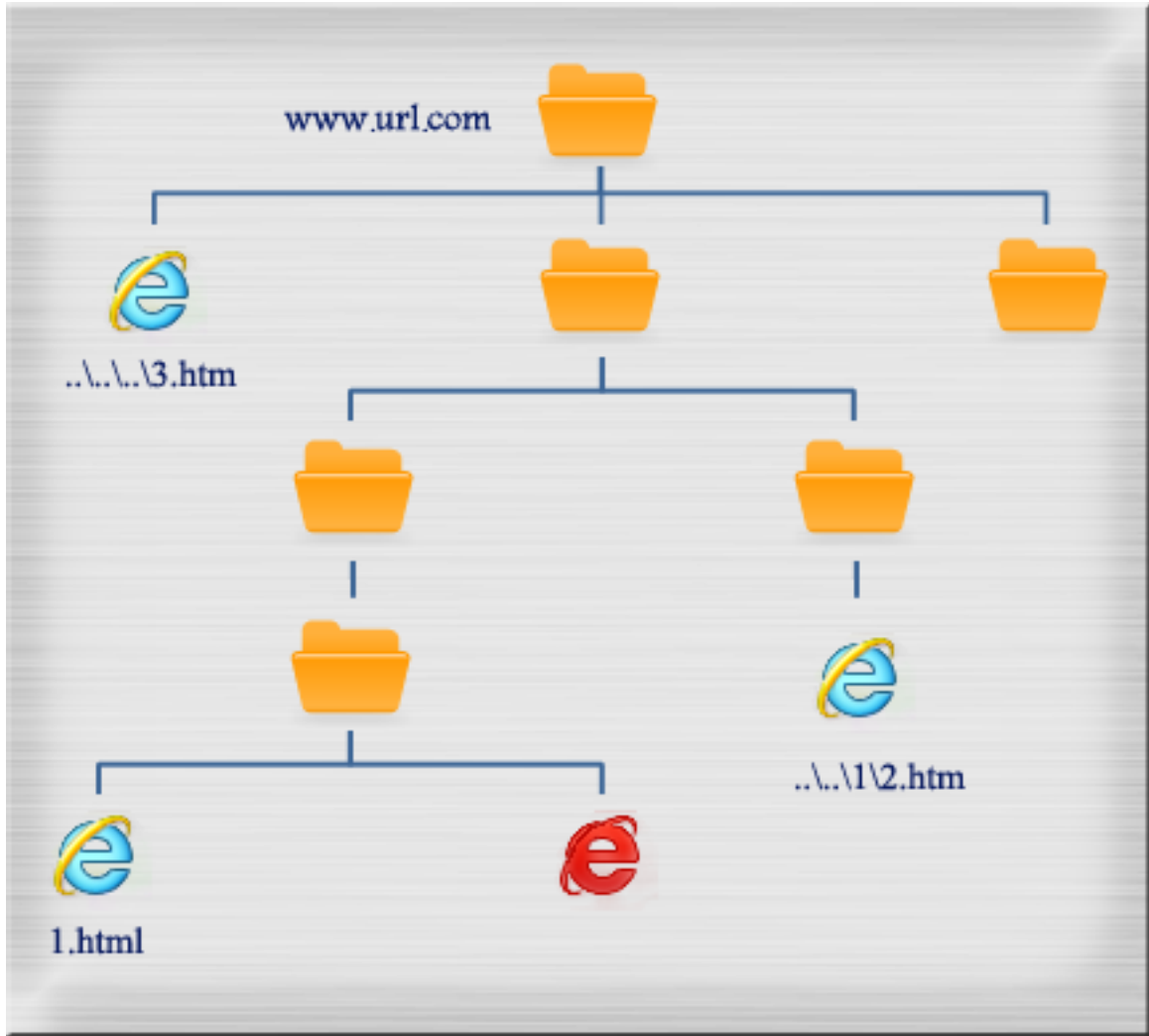
یک آدرس مطلق:

<http://www.monadianghadir.blogfa.com/post1.aspx#Myid>

http: پروتکل ارتباط با سرور، www.monadianghadir.blogfa.com: نام سرور، **post1.aspx**: نام

منبع موجود بر روی سرور، **#Myid**: تعیین کننده عنصر خاصی از صفحه که مقدار **id** آن برابر مقدار مذکور باشد.

Post.aspx



در شکل بالا آدرس صفحاتی از دایرکتوری سایت به صورت نسبی نشان داده شده است. قسمت قرمز رنگ صفحاتی است که لینکها درون آن قرار دارند.

۴.۲ ریز مقصدها

وقتی مقصد لینک بخشی از صفحه حاوی لینک باشد، از صفت **id** که باید در صفحه یکتا باشد، استفاده می کنیم.

<p>More information will be found in

Section
3.

Please visit it.</p>

. . .

<h1 id="sec3">3. Section Three of Document</h1>

۴.۳ عنصر base

< base href= />

عنصر **base** در قسمت سر صفحه قرار گرفته و باید یکتا باشد. تنها دارای یک صفت **href** است که مبنای تمامی آدرس‌های نسبی موجود در سند را تعیین می‌کند.

۵. عناصر بلوکی دیگر

۵.۱ ترسیم خط افقی

<hr />

ترسیم خط افقی به منظور مشخص کردن بخش‌های اصلی در صفحه را برعهده‌دارد. هیچ پارامتر اضافی ندارد و به صورت زیر به کار می‌رود:

this is a section

<hr/>

this is an other

this is a section

this is an other

۵.۲ عنصر عمومی div

div

یک عنصر بلوکی که می‌تواند هر نوعی از عناصر درون‌خطی یا بلوکی را درون خود جای داده و هیچ صفات اضافی ندارد. گروهی از عناصر را به گونه‌ای جمع می‌کند که می‌توانند به عنوان یک مجموعه عمل کنند. در زمینه‌ی شیوه‌ها این عنصر بسیار کاربردی‌ست زیرا می‌تواند یک صفت **class** داشته‌باشد. اگر کاربر نیاز به تعامل با بخش خاصی از سند داشته باشد، می‌توان آن بخش را بوسیله یک **div** محصور کرده و برای آن یک **id** تعیین نمود.

۶. عناصر درون‌خطی دیگر

۶.۱ عناصر عبارتی مهم

code samp var abbr acronym cite defn

این عناصر حاوی اطلاعات معمولی که در پارگراف‌ها می‌آیند نیستند. HTML شامل تعدادی از عناصر است که معنای عبارت درون خود را به دقت تعریف می‌کنند، این عناصر در ارائه بهتر توسط مرورگر، موتورهای جستجو و غلط‌یاب‌ها کاربرد دارند. بعضی از آن‌ها:

```
<p> As <cite>Bob Hopgood</cite> points out, <dfn>HTML stands for Hypertext Markup Language</dfn>, which is a <abbr title="World Wide Web Consortium">W3C</abbr> Recommendation used by <acronym title="North Atlantic
```

Treaty Organisation">NATO</acronym> to markup documents.

Something like `<code>class="classified"</code>` is used to indicate that some text like `<samp>In Confidence</samp>` is added to all the classified paragraphs. If the `<var>class</var>` has a value of `<var>unclassified</var>`, no additional text is added.</p>

As Bob Hopgood points out, *HTML stands for Hypertext Markup Language*, which is a W3C

Recommendation used by NATO to markup documents. Something like `class="classified"` is used to indicate that some text like `In Confidence` is added to all the classified paragraphs. If the `class` attribute has a value of `unclassified` no additional text is added.

Cite: استناد یا ارجاعی به منبع دیگر.

Dfn: مکان تعریف یک لغت. در موتور جستجوی گوگل، فرمول جستجوی:

Difine: Some Thing

تعریف عبارت مورد نظر را ارائه می کند، به نظر می رسد که عملکرد خزنده این موتور شناسایی عبارت dfn در کد صفحات باشد.

Code: ثبت تکه کدها.

Samp: ارائه مثال.

Var: آرگومان یا متغیر.

Abbr: اختصارات، صفت `title`، حاوی عنوان کامل اختصار است. عنوان کامل به هنگام نگاهداشتن ماوس روی اختصار نمایش داده می‌شود.

Acronym: سرنام، صفت `title`، حاوی عناوین کامل سرنام‌هاست. عنوان کامل به هنگام نگاهداشتن ماوس روی سرنام نمایش داده می‌شود.

۶.۲ توان و اندیس

`sub`، `sup`

```
<p> x<sub>i</sub> = y<sub>i</sub> + 2<sup>a + b</sup> +  
b<sup>2</sup></sup></p>
```

$$x_i = y_i + 2^{a+b}$$

۶.۳ عنصر `span`

`span`

مجموعه‌ای از عناصر و متون درون خطی را گرد هم می‌آورد که می‌توانند به صورت واحد عمل کنند. با داشتن یک صفت `class` می‌تواند برای مجموعه یک شیوه متفاوت از سایر سند ایجاد کند. با انتساب یک `id` به آن، امکان تعامل کاربر با مجموعه فراهم می‌شود.

۷. تصاویر

۷.۱ عنصر `img`

```
<img src= alt= height= width= longdesc= usemap= />
```

عنصر درون خطی نمایش تصاویر است و صفات اصلی آن:

- **Src:** حاوی آدرس تصویر.
- **Alt:** نوشته توصیفی تصویر. اگر مرورگر کاربر موفق به اجرای تصویر مورد نظر نشود، به جای تصویر نمایش داده می شود.
- **height, Width:** طول و عرض تصویر مورد نظر را تعیین می کنند.
- **Usemap:** برای تعیین نقشه مرتبط در نقشه تصویری. (توضیحات در ادامه خواهد آمد).

```

```

۷.۲ عنصر `object`

```
<object data= type= standby= height= width= >
```

شبيه به عنصر قبلی است و به دلیل انعطاف بالا در نمایش کاربردی تر است. عنصری درون خطی بوده و برای قراردادن انواع فایل ها با قالب های مختلف در صفحات وب به کار می رود.

- **Data:** حاوی آدرس تصویر مورد نظر است.
- **Type:** نوع تصویر را معین می کند، مرورگر قبل از دانلود تصویر می تواند نوع فایل را بررسی کند.
- **height, Width:** طول و عرض تصویر مورد نظر را تعیین می کنند.
- **Standby:** متنی که در هنگام دانلود تصویر نمایش داده می شود، برای مثال: تصویر در حال بارگذاری است.

```
< img usemap= />
```

```
<map name= >
```

```
<area href= alt= shape= coords= />
```

بوسیله صفت **usemap** از عنصر **img** ایجاد می‌گردد. کاربرد آن ایجاد لینک در نقاط مختلف یک تصویر است، فرض کنید می‌خواهیم تصویری از نقشه ایران در یک صفحه وب بگنجانیم، به گونه‌ای که کاربر بتواند با کلیک روی تصویر هر یک از استان‌ها به صفحه‌ی مخصوص آن استان منتقل شود. کافی‌ست بخش‌هایی که هنگام کلیک کردن بر روی آن‌ها فعال خواهند شد، را تعریف کنیم.

```
<a href="#dummy">
```

```

```

```
</a>
```

```
</p>
```

```
<map name="maprcmm">
```

```
<area
```

```
href="http://cern.web.cern.ch/CERN/Divisions/ETT/WPE/People/RobertCailli
```

```
au/"
```

```
shape="rect" coords="0, 0, 138, 127" />
```



```
<area href="http://www.muzmo.com/Murray/" shape="rect" coords="139, 0,  
277, 127" />  
</map>
```

map: این عنصر در هر جای سند می‌تواند به کار برده‌شود، عنصر **img** از طریق صفت **usemap** خود و تطابق مقدار آن با صفت **name** از عنصر **map** نقشه مرتبط را شناسایی می‌کند. عنصر **map** شامل عناصری به نام **area** است که بخش‌های قابل کلیک تصویر و مقصد لینک را مشخص می‌کنند. هر عنصر **area** دارای صفت **coords** است که مختصات بخش موردنظر را مشخص می‌کند. دیگر عنصر آن **shape** است که شکل هندسی بخش موردنظر تعیین می‌کند. مقادیر قابل استفاده:

- **Default**: تمام شکل را تعیین می‌کند.
- **Rect**: مستطیلی شکل؛ (minx miny maxx maxy).
- **Circle**: دایره شکل (centrex centrey radius).
- **Poly**: چندضلعی (x1 y1 x2 y2 x3 y3).

از آن‌جا که ساخت نقشه‌های تصویری با دست و استفاده از کدنویسی طاقت‌فرساست و نیاز به دقت زیادی دارد می‌توان از نرم‌افزارهایی که دارای رابط گرافیکی هستند استفاده نمود؛ از بهترین نمونه‌های آن می‌توان به Adobe Image Ready اشاره نمود.

۸. جدول‌ها

۸.۱ مقدمه

هر جدول از سه قسمت، مطلع، بدنه، و انتها تشکیل شده‌است، همچنین دارای یک عنوان اختیاری است. در کنار سه بخش اصلی، جدول به ردیف‌هایی تقسیم می‌شود که خود دارای تعدادی ستون هستند. جدول‌ها از سلول‌هایی تشکیل شده‌اند که می‌توانند محتوی هر تعداد عنصر بلوکی باشند. در صورت نیاز در هر سلول می‌توان جدول دیگری ایجاد کرد.

۸.۲ ساختار

```
<table summary= width= >  
  
<caption>. . .</caption>  
  
<colgroup align= span= width= />  
  
<col align= span= width= />  
  
<thead>  
  
. . .  
  
</thead>  
  
<tfoot>  
  
. . .  
  
</tfoot>  
  
<tbody>  
  
. . .  
  
</tbody>
```

</table>

```
<table title="group1" summary="5 column table with header row and first column titles." width="90%">
```

```
<caption>Group 1</caption>
```

```
<thead>
```

```
<tr>
```

```
<th>country</th><th>Iran</th><th>Germany</th><th>Brazil</th><th>Africa</th>
```

```
</tr>
```

```
</thead>
```

```
<tbody>
```

```
<tr>
```

```
<th>Games</th><td>2</td><td>2</td><td>2</td><td>2</td>>
```

```
</tr>
```

```
<tr>
```

```
<th>Wins</th><td>1</td><td>2</td><td>0</td><td>0</td>
```

```
</tr>
```

```
<tr>
```

```

<th>loses</th><td>0</td><td>0</td><td>1</td><th>2</th>
>
</tr>
<tr>
<th>draws</th><td>1</td><td>0</td><td>1</td><th>0</th>
>
</tr>
</tbody>
</table>

```

Group 1				
country	Iran	Germany	Brazil	Africa
Games	2	2	2	2
Wins	1	2	0	0
loses	0	0	1	2
draws	1	0	1	0

بخش‌های جدول از سطرها **tr** و تقسیمات آن **th** و **td** تشکیل شده‌است. مرورگر اطلاعات مطلع را به صورت **bold** نمایش می‌دهد. در این مثال جدول عرضی برابر ۹۰ درصد عرض صفحه دارد.

Summary برای درک بهتر جدول توسط کاربر به کار می‌رود.

۹. فرم‌ها

۹.۱ مقدمه

فرم‌های HTML شبیه به فرم‌هایی هستند که ما در زندگی روزمره با آن‌ها سروکار داریم. فرم‌ها در اشکال و اندازه‌های متنوعی وجود دارند. بعضی از آن‌ها دارای فیلدهای متنی هستند که کاربر باید متنی -مانند نام یا نام‌خانوادگی- را در آن‌ها وارد کند، بعضی دیگر چند گزینه در اختیار کاربر قرار داده و می‌خواهند تا یکی از گزینه‌ها را انتخاب کند -مانند پرسش‌های کنکور- و بعضی دیگر به کاربر امکان قراردادن تیک در کنار مواردی را می‌دهند. تقریباً تمام اشکال استاندارد فرم توسط HTML پشتیبانی می‌شوند. کاربر باید تعدادی فیلد را پر کرده و آن‌ها را به سرور ارسال کند. فرم‌ها در سرور بوسیله یک زبان برنامه‌نویسی تجزیه و تحلیل می‌شوند. ممکن است فرم تنها برای تعامل صفحه با کاربر ایجاد شده باشد که در این صورت از یک اسکریپت در کنار فرم استفاده می‌شود. در اینجا ما بر طراحی ظاهر فرم تمرکز داریم نه ارسال و پردازش آن.

form: تمام اجزای فرم را دربرمی‌گیرد.

input: اغلب فیلدها را ایجاد کرده و صفات آن بیان‌گر نوع فرم هستند.

textarea: برای فیلدهایی طراحی شده که کاربر می‌خواهد متنی طولانی را وارد نماید.

Select: برای ایجاد لیست‌های لغزنده‌ست، برای ایجاد گزینه‌های آن از عنصر **option** استفاده می‌شود.

Button: برای ایجاد دکمه‌ها -نظیر ارسال- به کار می‌رود.

```
<form method= action= onsubmit= >  
  
<input name= type= value= size= maxlength= checked=  
src= accesskey= >  
  
<form method= "post" action  
="mailto:bhop@mailsara.ac.uk">  
  
<p>Please type the mail to us:  
  
<input name="MyMessage" type="text" >  
  
<input type="submit" value="submit" name="submit">  
  
</p>  
  
</form>
```

Please type the mail to us:

عنصر فرم دو عنصر اصلی دارد.

- **Method:** پروتکل تحویل اطلاعات فرم را مشخص می کند. دارای دو نوع مقدار get و post می باشد.
 - get: حالت پیش فرض است. اگر یک URL به عنوان مقدار صفت action تعیین شود، یک ؟ که به دنبال آن مقدار ورودی می آید به سرور فرستاده خواهد شد.
 - post: در این مورد مقدار ورودی، درون فرم به سرور فرستاده می شود.

• **Action**: مکانی را که اطلاعات فرم باید فرستاده شوند، مشخص می کند.

اگر کاربر در کادر **Hello world** وارد کند، این عبارت به آدرس فرستاده می شود:

MyMessage=Hello+world

اگر خط اول عبارت را به این صورت تغییر دهیم و کاربر همان عبارت را وارد کند:

```
<form method= "get" action =  
"www.monadianghadir.blogfa.com">
```

این صفحه درخواست می شود:

```
http:// www.monadianghadir.blogfa.com  
/?MyMessage=Hello+world
```

اگر فرم چند مقدار **input** داشته باشد، هر یک از این مقادیر با **&** از هم جدا خواهد شد:

```
http://www.w3.org/?MyMessage=Hello+world&MySecondMess  
age=From+me
```

اگر بخواهیم از طریق فرم با صفحه تعامل برقرار شود:

```
<form onsubmit="process () ">
```

با فشردن دکمه تحویل توسط کاربر، تابع **process** عمل خواهد کرد. بیش تر در این زمینه مربوط به زبان های

اسکریپتی است.

```
<form method= action= onsubmit= >
```

```
<input name= type= value= size= maxlength= checked=  
src= accesskey= >
```

دارای سه صفت اصلی **name**، **type** و **value** و چند صفت دیگر به شرح زیر است:

- **Name**: مقداری است که به ورودی افزوده می‌شود و بیان‌گر آن‌ست که اطلاعات از کجا آمده است.
- **Type**: نوع ورودی را مشخص می‌کند. دارای انواع مقادیر: text، submit، radio، checkbox، button، rest، password، file، hidden و image است.
- **Text**

```
<input name ="MyMessage" type="text" value="Some  
text:" size="40"  
maxlength="10" />
```

• **Submit**

وظیفه تحویل فرم را بر عهده دارد و یک دکمه به این منظور ایجاد می‌کند.

```
<form method= "post" action =  
"mailto:monadianghadir@gmail.com">
```

```
<p>Please type the mail to send message:
```



```
<input name="MyMessage" type="text" />
```

```
<input type = "submit" name="Mail" value = "Send Mail" /></p>
```

```
</form>
```

Please type the mail to send message:

:Radio •

```
<p>Indicate your choice of language:</p>
```

```
<p>Persian: <input type ="radio" name="language" value="persian"
```

```
checked="checked" /></p>
```

```
<p>English: <input type ="radio" name="language" value="English"></p>
```

```
<p>Arabic: <input type ="radio" name="language" value="Arabic" /></p>
```

Indicate your choice of language:

Persian:

English:

Arabic:

از آن جا که تمام موارد دارای صفت **Name** یکسانی هستند، به عنوان یک ابزار واحد عمل می کند و تنها یکی از گزینه ها در هر زمان می تواند انتخاب شده باشد. مقدار اولیه Persian تعیین شده است.

• **Checkbox:**

Page | 34

```
<p>Indicate all the choices of languages you Know:</p>
```

```
<p>Persian: <input type = "checkbox" name="language" value="Persian" /></p>
```

```
<p>Arabic: <input type="checkbox" name="language" value="Arabic" /></p>
```

```
<p>English: <input type="checkbox" name="language" value="English" /></p>
```

```
<input type="submit" name="Choice" value="Choices Made" />
```

Indicate all the choices of languages you Know:

Persian:

Arabic:

English:

Choices Made

اگر دو گزینه اول انتخاب شوند، مقدار ارسالی به سرور:

language=Persian&language=Arabic&Choice=Choices+Made

• **:Button**

اجرای اسکریپت‌های محلی را ممکن می‌سازد.

```
<input type="button" value="Click Me"
onclick="processclick()" />
```

• **:Reset**

اگر کاربر در ورود اطلاعات اشتباه کرده باشد، بوسیله این دکمه می‌تواند تمام قسمت‌های فرم را به حالت اولیه برگرداند.

• **:Password** در مواردی که اطلاعات ورودی محرمانه بوده و نمی‌خواهیم که قابل مشاهده باشند، کاربرد دارد. هر کاراکتر ورودی توسط کاربر به صورت * نمایش داده می‌شود.

• **:Value** در اکثر موارد مقدار اولیه فرم را مشخص می‌کند، نوع radio تنها نوعی است که مقدار اولیه برای آن ضروری است.

• **:File**

گاهی اوقات ورودی خواسته شده فایلی از رایانه کاربر می‌باشد. در این صورت به جای تایپ نام فایل و آدرس آن، کاربر می‌تواند به فایل مزبور از طریق بررسی⁸ رایانه دسترسی داشته باشد.

<p>Input name of picture to be processed:

```
<input type="file" name="imagefile"
```

```
value="C:\Files\test\photo.jpg" /></p>
```

⁸. Browes

Input name of picture to be processed:

• **:Hidden**

گاهی اوقات نیاز است که اطلاعاتی به سرور فرستاده شود که طراح صفحه نمی‌خواهد کاربر آن‌ها را وارد کند.

```
<input type="hidden" name="from" value="\ " />
```

چیزی در صفحه نمایش داده نمی‌شود اما هنگامی که فرم به سرور فرستاده شود یکی از مقادیر، `from=1` خواهد بود.

• **Value:** این صفت مقدار اولیه را تعیین می‌کند. **radio** تنها نوعی است که در آن تعیین مقدار اولیه

اجباری است. اگر فیلدهای فرم توسط کاربر پرنشوند، این مقدار به سرور ارسال خواهد شد.

• **Size:** عرض فرم را یا به پیکسل و یا کاراکتر مشخص می‌کند.

• **Maxlength:** برای نوع **text** و **password** کاربرد داشته و حداکثر تعداد کاراکترها را برای

ورودی تعیین می‌کند.

• **Checked:** مخصوص انواع **text** و **checkbox** و تعیین‌کننده مقدار اولیه دکمه‌هاست.

۹.۴ selection ها

```
<select name= size= multiple= >
```

```
<option value= label= >
```

```
<optgroup label= >
```

بسیاری از فرم‌ها مجموعه‌ای از گزینه‌ها را ایجاد کرده و از کاربر می‌خواهند یکی از آن‌ها را انتخاب کند. عنصر

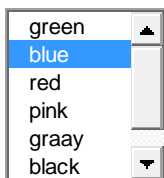
`select` لیستی از گزینه‌ها برای انتخاب در اختیار کاربر قرار می‌دهد.

```
<p>Please make your choice of color:</p>
```



```
<select name="colors" size="6" multiple="multiple">
<option value = "green">green</option>
<option value = "blue">blue</option>
<option value = "red">red</option>
<option value = "pink">pink</option>
<option value = "gray">graay</option>
<option value = "black"
selected="selected">black</option>
<option value = "yellow">yellow</option>
</select>
```

Please make your choice of color:



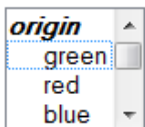
صفت **name**، نام اطلاعاتی که به سرور فرستاده می‌شوند را مشخص می‌کند و **size** تعداد گزینه‌هایی که در آن نمایش داده‌شوند را تعیین می‌کند. صفت **Multiple** تعیین می‌کند که یک یا چندگزینه قابل انتخاب باشند. اگر امکان انتخاب چندگزینه فراهم شود کاربر با نگه‌داشتن **ctrl** می‌تواند چندگزینه را انتخاب کند. عناصر **option** درون عنصر **select** قرار گرفته و هر کدام از آن‌ها یکی از گزینه‌ها را تعیین می‌کند. اگر صفت **select** برای گزینه‌ای تنظیم شود، آن گزینه در هنگام نمایش صفحه در حال انتخاب خواهد بود. در مثال بالا **black** تنظیم شده‌است. اگر **green** توسط کاربر انتخاب شود، مقدار فرستاده شده به سرور چنین خواهد بود:

color=green&color=black

برای منوهای با تعداد گزینه بالا می‌توان گزینه‌ها را تقسیم‌بندی کرد. برای این کار از عنصر **optgroup**

استفاده می‌شود.

```
<p>Please make your choice of color:</p>
<select name="music" size="4" multiple="multiple">
  <optgroup label="origin">
    <option value = "green">green</option>
    <option value = "red">red</option>
    <option value = "blue">blue</option>
  </optgroup>
  <optgroup label="other">
    <option value = "pink">pink</option>
    <option value = "yellow"
      selected="selected">yellow</option>
  </optgroup>
</select>
```



<textarea name= rows= cols= readonly= />

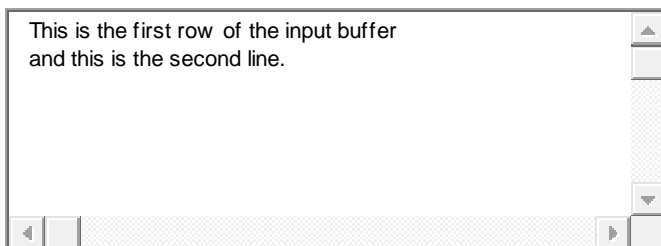
به منظور ایجاد ناحیه ورود اطلاعات با چندین خط به کار می‌رود. Page | 39

<textarea name="para" rows="7" cols="60" ></textarea>



یک ناحیه هفت خطی که هر خط گنجایش ۶۰ کاراکتر دارد، ایجاد می‌کند. البته از لحاظ تعداد خطوط ورودی محدودیت ندارد بلکه تعداد خطوطی که نمایش داده می‌شود را تعیین می‌کند. برای تعیین مقدار اولیه:

```
<textarea name="para" rows="7" cols="60"
readonly="readonly">This is the first row of the
input buffer
and this is the second line.</textarea>
```



با تنظیم صفت **readonly**، متن ابتدایی غیر قابل تغییر خواهد بود.

۹.۶ عنصر button

`<button name= value= type= >`

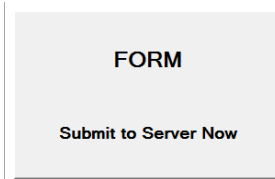
به دلیل توان کم دکمه‌هایی که با input ایجاد می‌شوند دکمه‌های دیگری ایجاد شده که شبیه به همان دکمه‌ها هستند و جانشین انواع button, submit و reset هستند. تفاوت عمده این دکمه‌ها امکان قرار گرفتن هر محتوایی درون آن‌هاست.

```
<button name="form" value="submit" type="submit" >
```

```
<h2>FORM</h2>
```

```
<h3>Submit to Server Now</h3>
```

```
</button>
```



به این ترتیب شما تسلط کاملی بر محتوای دکمه خواهید داشت. می‌توان به دکمه‌ها تصاویر، رنگ‌های زمینه و هر افکت موردنیازی افزود.

۱۰.۱ مقدمه

عصر head شامل مجموعه‌ای از عناصر است که از محتوا محسوب نمی‌شوند، شامل فراداده‌ها، اطلاعات مربوط به شیوه‌ها و نحوه ارتباط صفحه با سایر صفحه‌هاست. مجموعه عناصری که در قسمت سر می‌آیند:

۱۰.۲ عنصر title

`<title >`

حاوی عنوان صفحه است. حضور این عنصر برای استاندارد بودن یک سند HTML الزامی است. وظیفه تعیین هویت سند را برعهده دارد و معمولاً این عنوان در قسمت بالای مرورگرها نمایش داده می‌شود. وقتی که کاربر صفحه‌ای را در لیست علاقه‌مندی‌ها یا نشانه‌ها^{۱۰} ذخیره می‌کند به طور پیش فرض بوسیله عنوان تعیین هویت می‌شود.

۱۰.۳ عنصر meta

`<meta name= content= http-equiv= scheme= />`

راه اصلی ست که صفحات، توسط موتورهای جستجو ردیابی می‌شوند. صفات name و content یک ویژگی و مقدار آن را تعیین می‌کنند.

مجموعه‌ای از فراداده‌ها -اطلاعات درباره اطلاعات- است، مورد استفاده مرورگرها، موتورهای جستجو، ویرایشگرهاست.

`<meta name= content= http-equiv= scheme= />`

⁹ . Document Head

¹⁰ .bookmark

بعضی از موتورهای جستجو با استفاده از خزنده در وب جستجو می‌کنند و بعضی دیگر از فهرست‌نویسی انسانی استفاده می‌کنند، نوع سوم آن‌ها از ترکیب روش‌های فوق استفاده می‌کند. خزنده صفحات وب را مشاهده و لینک‌های آن‌ها را دنبال می‌کند. نوع بهتر آن‌ها ممکن است تمام متن صفحه را بررسی کند. نوع ضعیف آن‌ها ممکن است تنها قسمت سر را بررسی کند. عنصر **meta** یکی از عناصری است که بعضی موتورهای جستجو به آن توجه دارند. اما تضمینی وجود ندارد که با استفاده از تعداد زیادی از آن‌ها سند حتما شناخته خواهد شد. بعضی طراحان صفحات برای افزایش رتبه خود در موتورهای جستجو، به استفاده از کلمات غیرمرتبط می‌پردازند، به این عمل **spamming** می‌گویند. برای مقابله با این مشکل موتورهای جستجو فراداده‌ها را با محتوای صفحات مقایسه می‌کنند. مهم‌ترین مقادیری که برای صفت **name** به کار می‌روند **description** و **keywords** هستند. اگر موتور جستجو به وب سایت اعتماد پیدا کند، به جای روش‌های دیگر از **description** صفحه استفاده خواهد کرد. مقدار **keywords** کلیدواژه‌هایی را که مردم برای پیدا کردن صفحه نیاز دارند، شامل می‌شود.

مقدار **robots** توسط نویسنده برای درخواست تحوه بررسی از خزنده -مثل **index, follow** درخواست نمایه کردن صفحه و بررسی لینک‌ها- به کار می‌رود. اگر مقدار **noindex** تعیین شود خزنده از صفحه بازدید نخواهد کرد.

این وب‌سایت آموزشی از نحوه `<meta name="description" content="`

`>` ایجاد صفحات وب است.

`<meta name="keywords" content="html, ساخت صفحات وب ,`

`>` اچ‌تی‌ام‌ال , طراحی وب

`<meta name="robots" content="index, follow">`

سایر مقادیر صفت **name** عبارتند از:

Author: نام نویسنده صفحه

Generator: ابزار مورد استفاده برای ایجاد صفحه.

Formatter: شبیه به گزینه قبل.

Copyright: عبارت حق مولف.

Owner: حق مولف.

Subject: شبیه به عنوان.

صفت **http-equiv** یک **http** سرصفحه تعریف می کند که فعالیت مرورگری که صفحه را دریافت می کند را کنترل کرده و برای تصفیه اطلاعات مهیاشده بوسیله سرصفحه های فعال در **http** کاربرد دارد. بعضی سرورها به طور خودکار فرااطلاعات را به سرصفحه های **http** فعال ترجمه می کنند. دیگر مقادیر ممکن:

```
<meta http-equiv="expires" content=" Wed 28 Feb 2002  
23:59:59 GMT">
```

```
<meta http-equiv="Refresh"  
content="0 ;URL=http://www.newurl.com">
```

```
<meta http-equiv="Set-Cookie"  
content="cookievalue=xxx;expires=Wednesday, 28-Feb-02  
18:12:20 GMT;
```

path="/">

خط اول زمانی که صفحه دیگر معتبر نیست یا به عبارتی تاریخ انقضا صفحه را تعیین می‌کند. خط دوم زمانی که صفحه باید به‌روزرسانی شود و آدرس صفحه جدید را معین می‌کند. خط سوم یک کوکی را تعیین می‌کند.

صفت **scheme** به عنوان توصیف‌گر فرااطلاعات به کار می‌رود.

```
<meta scheme="ISBN" name="identifier" content="0-201-17805-2">
```

که شاپک کتاب Raggett on HTML 4 را بیان می‌کند.

۴.۱۰ اسکریپت‌نویسی

```
<script type= src= >
```

```
<noscript >
```

شامل اسکریپتی مرتبط با صفحه است. اسکریپت‌نویسی خارج از دامنه این مقاله قرار دارد اما همین قدر کافی است که بدانیم اسکریپت‌ها را هم در قسمت سر و هم بدنه می‌توان استفاده کرد. اسکریپت‌ها بیش‌تر در قسمت سر قرار می‌گیرند اما بهترین مکان نیست زیرا موتورهای جستجو معمولاً قسمت سر را بررسی می‌کنند و اگر شما اسکریپتی طولانی در این قسمت قرار دهید، ممکن است موتورهای جستجو آن‌ها را هجو تلقی کنند.

```
<script type="text/vbscript" src="http://.org/script/script.js" />
```

یک اسکریپت خارجی - در صفحه‌ی دیگری نوشته شده -، صفت `type`، نوع آن که وی‌بی اسکریپت است و `src`، آدرس صفحه حاوی کد را مشخص می‌کند.

```
<script type="text/javascript">
```

```
Javascript Codes </script>
```

از اسکریپت داخلی که بین تگ‌های شروع و پایان قرار می‌گیرد، استفاده شده‌است.

```
<body onload="processscript()">
```

```
<body onunload="processscript()">
```

عنصر `body` صفتی دارد که سبب می‌شود که در هنگام بارگذاری یا ترک صفحه اسکریپت اجرا شود.

اکثر عناصر صفاتی برای احضار اسکریپت‌ها در هنگام وقایع مانند کلیک کردن یا جابجایی ماوس دارند.

```
<h1 onclick="ps()">ABC</h1>
```

```
<p onmousedown="ps()">ABC</p>
```

وقایع عبارتند از:

Onload: اختصاص به عنصر **body** داشته و هنگامیکه مرورگر صفحه را به طور کامل بارگذاری کند اتفاق می‌افتد.

- **Onunload**: اختصاص به عنصر **body** داشته و هنگامیکه مرورگر صفحه را ترک کند اتفاق می‌افتد.
- **OnClick**: هنگامی که ماوس روی یک عنصر کلیک کند، اتفاق می‌افتد.
- **Ondblclick**: هنگامی که ماوس روی یک عنصر دابل کلیک کند، اتفاق می‌افتد.
- **Onmousedown**: هنگامی که دکمه ماوس روی عنصری نگاه‌داشته شود.
- **Onmouseup**: هنگامی که دکمه ماوس روی عنصری رها شود.
- **Onmouseover**: هنگامی که ماوس از عنصر عبور کند.
- **Onmousemove**: هنگامی که ماوس روی عنصر حرکت کند.
- **Onmouseout**: هنگامی که ماوس به خارج از عنصر حرکت کند.
- **Onfocus**: هنگامی که روی یک عنصر مثلاً یک فرم تمرکز شود.

- **Onblur**: هنگامی که عنصری از حالت تمرکز^{۱۱} خارج شود.
- **Onkeypress**: هنگامی که یک کلید روی عنصری زده شود.
- **Onkeydown**: هنگامی که یک کلید روی عنصری نگاه داشته شود.
- **Onkeyup**: هنگامی که کلیدی روی یک عنصر رها شود.
- **Onsubmit**: هنگامی که یک فرم تحویل شود.
- **Onreset**: هنگامی که یک فرم پاک شود^{۱۲}.
- **Onselect**: هنگامی که متنی در عنصر **input** یا **textarea** انتخاب شود.
- **Onchange**: هنگامی که یک فرم از حالت تمرکز خارج شود و مقدارش پس از تمرکز تغییر کرده است.

۱۰.۵ عنصر style

`<style type= >`

این عنصر صفحه شیوه^{۱۳} ای که بوسیله سند استفاده می شود را تعیین می کند.

`<style type="text/css" media="print"> . . . </style>`

صفت **type** نوع شیوه و صفت **media** نوع رسانه را تعیین می کند.

`<link rel="stylesheet" href="new_page_1.css"
media="all" type="text/css" />`

۱۰.۶ عنصر link

`<link href= type= media= hreflang= charset=/>`

¹¹. focus

¹². reset

¹³. stylesheet

هدف آن ارائه اطلاعات مرتبط با تحوه ارتباط سند با دیگر صفحات است. وقتی دو صفحه به طور متقابل به هم لینک می‌دهند مشخص نیست که کدام یک پدر و کدام فرزند است، کاربرد این عنصر در حل این مشکل و کمک به موتورهای جستجو برای اطلاعات لینک‌ها و افزایش کیفیت ارتباط جستجوهاست.

```
<link href="../../../primerstyle.css" rel="stylesheet"
type="text/css"
media="print"/>
<link rel="alternate" hreflang="pt" type="text/html"
/>
<link rel="alternate" hreflang="ar" charset="ISO-
8859-6"
type="text/html" />
```

خط اول لینک به یک صفحه شیوه خارجی برای زمانی که سند پرینت می‌شود.

خط دوم یک لینک به صفحه جایگزین پرتغالی ارائه می‌دهد.

خط سوم به صفحه جایگزین عربی لینک می‌دهد. در این مورد مجموعه کاراکتر نیز، نیاز به تعریف دارد. موتورهای جستجو ممکن است نسخه جایگزین را در صورت نیاز کاربر به زبانی دیگر ارائه دهند.

۱۰.۷ عنصر base

```
<base href= />
```

در هر سند یکتاست و تنها دارای یک صفت -href- است که مبنای تمامی آدرس‌های نسبی موجود در سند را معین می‌کند.

```
<base href= />
```

```
<base href="http://www.w3.org/nuts/index.htm" />
```

```
. . . < a href="../brazil/origin.htm" />
```

URL مطلق آن چنین است: <http://www.w3.org/brazil/origin.htm>