

# فهرست جلسات دوره آموزش HTML و HTML5

## آموزش HTML

- جلسه ۰۱ : مقدمه ای بر html
- جلسه ۰۲ : آشنایی با ویرایشگرهای html
- جلسه ۰۳ : تگ های اصلی html
- جلسه ۰۴ : ساختار عناصر html
- جلسه ۰۵ : صفات در HTML
- جلسه ۰۶ : تگ heading در HTML
- جلسه ۰۷ : پاراگراف در html
- جلسه ۰۸ : style در html
- جلسه ۰۹ : قالب بندی متن در html
- جلسه ۱۰ : تگ های Citation و Quotation
- جلسه ۱۱ : کامنت در html
- جلسه ۱۲ : رنگ ها در HTML
- جلسه ۱۳ : CSS در HTML
- جلسه ۱۴ : لینک ها در html
- جلسه ۱۵ : تگ img در HTML
- جلسه ۱۶ : کار با Table در HTML
- جلسه ۱۷ : تگ های List در html
- جلسه ۱۸ : ویژگی های Block و Inline در html
- جلسه ۱۹ : صفت class در html
- جلسه ۲۰ : صفت id در html
- جلسه ۲۱ : تگ Iframe در html
- جلسه ۲۲ : جاوااسکریپت در HTML
- جلسه ۲۳ : آدرس دهی فایل ها در HTML
- جلسه ۲۴ : تگ head در html
- جلسه ۲۵ : قالب وب سایت
- جلسه ۲۶ : ویژگی Responsive در html
- جلسه ۲۷ : کدهای کامپیوتری در html
- جلسه ۲۸ : موجودیت ها در html
- جلسه ۲۹ : نمادها در html
- جلسه ۳۰ : ویژگی Encoding در ( htmlتنظیمات کاراکترها )
- جلسه ۳۱ : منابع مکانی در html
- جلسه ۳۲ : ویژگی های HTML و XHTML

## فرم در HTML

- جلسه ۳۳ : فرم در کد HTML
- جلسه ۳۴ : عناصر فرم در HTML
- جلسه ۳۵ : انواع ورودی در HTML
- جلسه ۳۶ : صفات input در html

## HTML5

- جلسه ۳۷ : مقدمه ای بر HTML5
- جلسه ۳۸ : پشتیبانی HTML5 از مرورگرها
- جلسه ۳۹ : عناصر جدید در html5
- جلسه ۴۰ : عناصر نوع Semantic در html5
- جلسه ۴۱ : مهاجرت از HTML4 به HTML5
- جلسه ۴۲ : کد نویسی صحیح در html5

## گرافیک در HTML

- جلسه ۴۳ : آموزش گرافیک در HTML
- جلسه ۴۴ : کار با تگ گرافیکی SVG در html5

## چند رسانه ای در html

- جلسه ۴۵ : چند رسانه ای در html
- جلسه ۴۶ : اضافه کردن ویدئو در HTML
- جلسه ۴۷ : فایل صوتی در HTML
- جلسه ۴۸ : آموزش کار با Plug-ins در html
- جلسه ۴۹ : ویدئوهای YouTube در html

## HTML در APIs

- جلسه ۵۰ : موقعیت جغرافیای در html5
- جلسه ۵۱ : کار با Google Maps در html
- جلسه ۵۲ : قابلیت درگ کردن در HTML
- جلسه ۵۳ : ذخیره سازی وب در html
- جلسه ۵۴ : آشنایی با Web Workers در html5
- جلسه ۵۵ : چگونگی ارتباط با سرور در HTML

# فصل اول : آموزش مقدماتی HTML

## جلسه ۰۱ : مقدمه ای بر html

### مقدمه

با عرض سلام و وقت بخیر خدمت کاربران سایت [پی وی لرن](#) . و کاربرانی که [دوره کامل آموزش html](#) را دنبال می کنند. HTML که مخفف عبارت **Hypertext Markup Language** می باشد یک زبان نشانه گذاری برای طراحی ساختار صفحات وب سمت کاربر است. از html تاکنون نسخه های متعددی منتشر شده است که html 5 آخرین نسخه از آن می باشد و ما در طی این دوره کامل آموزش شما را با ساختار کلی و تگ های html و html5 آشنا خواهیم کرد. در ابتدای دوره ، در قالب مبحث **مقدمه ای بر html** شما را با ساختار کلی و طرز کار با html آشنا خواهیم کرد.

### مقدمه ای بر html

در ادامه با ارائه ی مفاهیم کلی html و چگونگی تایپ و تست تگ ها، مقدمه ای بر html خواهیم داشت.

### مثال های دوره

در طی دوره ی جاری در هر کدام از جلسات html نمونه کدهایی از html برای درک بهتر مطالب گنجانده شده است.

مثال : ساختار کلی صفحات وب

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>Page Title</title>
5. </head>
6. <body>
7.
8. <h1>This is a Heading</h1>
9. <p>This is a paragraph.</p>
10.
11. </body>
12. </html>
```

### خودتان امتحان کنید

البته مثال های ارائه شده به همراه یک ادیتور آنلاین در دسترس است تا شما به راحتی بتوانید خودتان کدهای html را تست کنید.

### html چیست

HTML زبان استاندارد نشانه گذاری برای ایجاد صفحات وب است.

- HTML مخفف عبارت Hyper Text Markup Language است.
- HTML ساختار صفحات وب را با نشانه گذاری توصیف می کند.
- عناصر HTML بخش های مختلف صفحات وب را می سازند.
- عناصر HTML در قالب تگ ها پیاده سازی می شوند.
- عنوان برچسب تگ های html بیانگر کاربرد آن است مانند: " " "table", "paragraph", "heading"...
- مرورگر تگ های HTML را نمایش نمی دهد، بلکه از آنها برای رندر محتوای صفحه استفاده می کند.

یک صفحه ی ساده html به صورت زیر است:

تشریح تگ های مثال فوق:

- اعلامیه `<!DOCTYPE html >` این سند را HTML5 تعریف می کند.
- عنصر `<html >` عنصر ریشه یک صفحه HTML است.
- عنصر `<head >` شامل متا اطلاعاتی در مورد سند است.
- عنصر `<title >` یک عنوان را برای سند مشخص می کند.
- عنصر `<body >` دارای محتوای صفحه قابل مشاهده است.
- عنصر `<h1 >` یک عنوان بزرگ را تعریف می کند.
- عنصر `<p >` پاراگراف را تعریف می کند.

## تگ های html

تگ های HTML در قالب کاراکترهای " `<...>` " که داخل کارکترهای باز و بسته نام عناصر و داخل تگ محتوای آن قرار می گیرد.

ساختار:

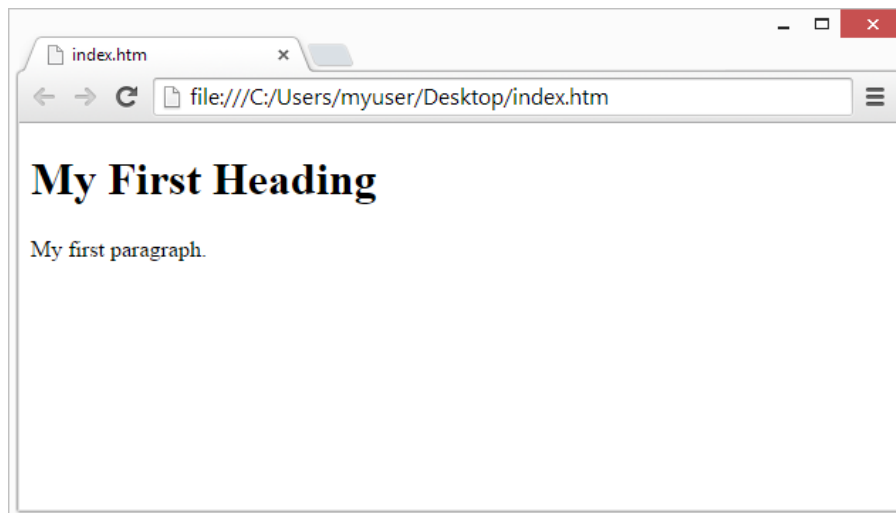
1. `<tagname>content goes here...</tagname>`

- تگ های HTML معمولاً به صورت جفتی مانند `<p >` و `</p >` قرار می گیرند.
- اولین تگ در یک جفت تگ شروع است، برچسب دوم تگ پایان است.
- تگ پایان مانند تگ شروع بوده، اما قبل از اسم تگ وارد شده است.

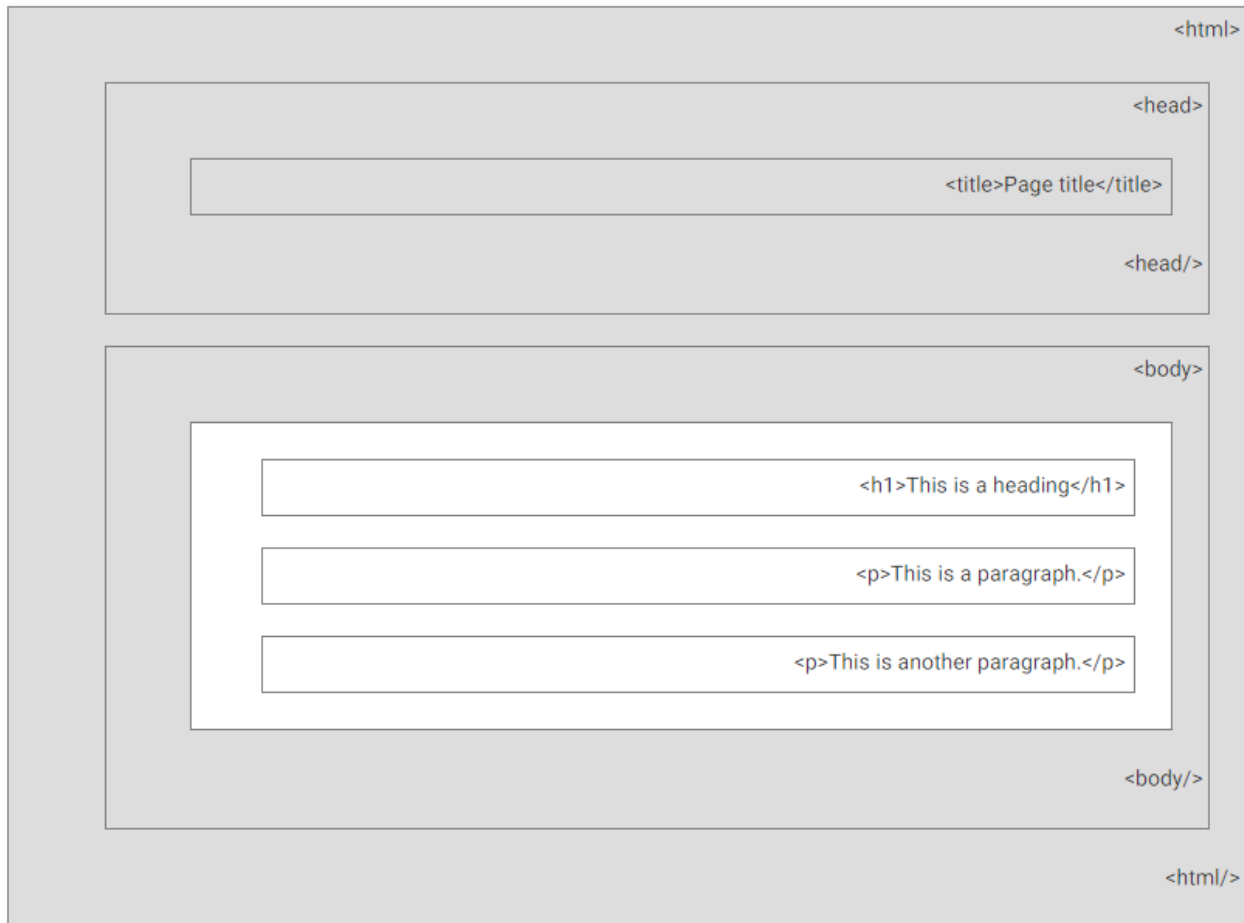
## مرورگرهای وب

هدف مرورگرهای وب (کروم، IE، فایرفاکس، سافاری و...) خواندن اسناد HTML و نمایش آنها است.

مرورگر تگ های HTML را نمایش نمی دهد، بلکه از آنها برای تعیین نحوه نمایش سند استفاده می کند:



مقدمه ای بر html



**نکته :** فقط محتوا در قسمت `<body>` (قسمت سفید بالای بالا) در مرورگر نمایش داده می شود.

### اعلامیه `<!DOCTYPE>`

اعلام `<!DOCTYPE>` نشان دهنده نوع سند است و به مرورگر کمک می کند تا صفحات وب به درستی نمایش داده شوند.

این باید فقط یک بار در بالای صفحه ظاهر شود (قبل از هر تگ HTML).

اعلام `<!DOCTYPE>` نسبت به حروف حساس نیست.

ساختار اعلامیه `<!DOCTYPE>` برای HTML5:

1. `<!DOCTYPE html>`

Year	Version
۱۹۹۱	HTML
۱۹۹۵	HTML 2.0
۱۹۹۷	HTML 3.2
۱۹۹۹	HTML 4.01
۲۰۰۰	XHTML
۲۰۱۴	HTML5

### کلام آخر

قبل از شروع آموزش های html ابتدا در این بخش سعی کردیم تا با ارائه ی **مقدمه ای بر html** ، شما را با مفاهیم و ساختار کلی تگ های html آشنا کنیم تا درک سایر مباحث ساده تر باشد.

در بخش بعدی شما را با ادیتورها و ویرایشگرهای html آشنا می کنیم.

## جلسه 02 : آشنایی با ویرایشگرهای HTML

### مقدمه

با عرض سلام و وقت بخیر خدمت کاربران سایت **پی وی لرن** . و کاربرانی که **دوره کامل آموزش html** را دنبال می کنند. تاکنون با مفاهیم و ساختار کلی تگ های html و طرز تایپ کردن آن ها آشنا شدیم، اما برای شروع هر گونه طراحی html شما نیاز به محیطی برای نوشتن تگ ها و طراحی ساختار html صفحات وب دارید. از این رو **آشنایی با ادیتورهای html** حائز اهمیت است. در ساده ترین حالت شما می توانید با یک ویرایشگر متن ساده نظیر notpad، تگ های html را نوشته و در قالب صفحه وب html ذخیره کنید، سپس می توانید نتیجه کار را در مرورگرهای وب مختلف بررسی کنید. همچنین نرم افزارهایی نظیر **dreamweaver** با ارائه ی ابزارهای کاملی کار طراحی صفحات وب را بسیار ساده تر می کنند. برای **آشنایی با ویرایشگرهای html** در ادامه کار با ما همراه باشید.

### آشنایی با ادیتورهای html

در ادامه ی مباحث برای آشنایی با ویرایشگرهای html مراحل تایپ و اجرای کد html در یک ویرایشگر ساده را ارائه داده ایم.

### نوشتن تگ های html با استفاده از Notepad یا Textedit

صفحات وب را می توان با استفاده از ویراستاران حرفه ای HTML از جمله dreamweaver، ایجاد و اصلاح کرد.

با این حال، برای یادگیری HTML، یک ویرایشگر متن ساده مانند Notepad (در ویندوز) یا TextEdit (در Mac) را توصیه می کنیم.

ما معتقدیم استفاده از یک ویرایشگر متن ساده راه خوبی برای یادگیری HTML است.

حالا چهار مرحله زیر را دنبال کنید تا اولین صفحه وب خود را با Notepad یا TextEdit ایجاد کنید:

### مرحله ۱: باز کردن Notepad (در ویندوز)

در ویندوز ۸ یا جدید تر:

Start Screen را باز کنید (نماد ویندوز در پایین و سمت چپ صفحه قرار دارد) سپس در کادر جستجو **Notepad** را تایپ کنید.

در ویندوز ۷ یا قدیمی تر:

مسیر **Start > Programs < Accessories < Notepad** دنبال کنید.

### مرحله ۱ ویندوز : Mac باز کردن برنامه TextEdit

مسیر **Finder > Applications > TextEdit** را دنبال کنید.

همچنین برخی از تنظیمات را تغییر دهید تا برنامه به درستی ذخیره شود. (از مسیر **Preferences > Format** گزینه **Plain Text** را انتخاب کنید).

سپس از زیر عنوان **Open and Save** تیک گزینه ی **Display HTML files as HTML code instead of formatted text** را نیز بزنید.

سپس یک سند جدید برای قرار دادن تگ های html ایجاد کنید.

## مرحله ۲ : تایپ یک کد HTML ساده

کد HTML زیر را در Notepad نوشته و یا به آنجا کپی کنید:

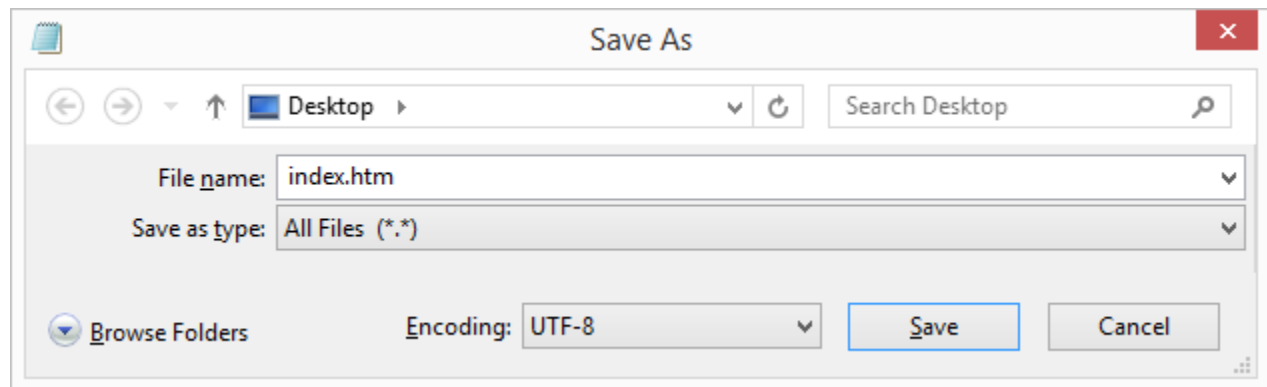
مثال : html ساده

```
1. <html>
2. <body>
3.
4. <h1>My First Heading</h1>
5.
6. <p>My first paragraph.</p>
7.
8. </body>
9. </html>
```

## مرحله ۳ : ذخیره کردن صفحه HTML

با انتخاب مسیر **File > Save as** از منوی notpad فایل ایجاد شده را در کامپیوتر خود ذخیره کنید.

در پنجره ی **Save as** نام فایل را "index.htm" و encoding را **UTF-8** تنظیم کنید.



آشنایی با ویرایشگرهای html

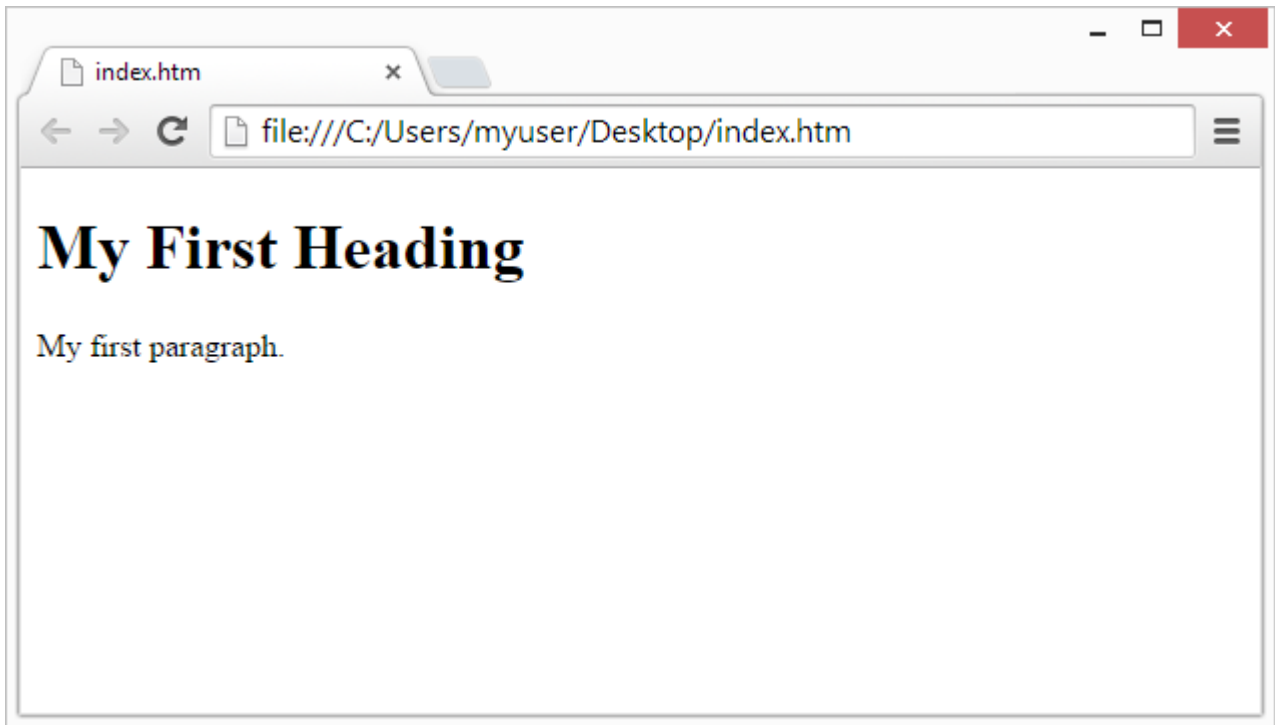
شما می توانید از .htm یا .html به عنوان پسوند فایل استفاده کنید. هیچ تفاوتی وجود ندارد.

## مرحله ۴ : مشاهده ی صفحه HTML در مرورگر

فایل HTML ایجاد شده را در بخش favorite مرورگر خود باز کنید (روی فایل html راست کلیک کرده و از زیر منوی "Open with" مرورگر خود را انتخاب نمایید).



نتیجه ی کار مشابه زیر خواهد بود:



آشنایی با ویرایشگرهای html

## ادیتور آنلاین پی وی لرن

ما یک ادیتور آنلاین برای تست و اجرای کدهای HTML که در قالب مثال های HTML در طول جلسات آموزش ارائه می شود. قرار داده ایم.

در این ادیتور شما می توانید کدهای HTML مثال ها را خودتان تغییر داده و اجرا کنید.

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>Page Title</title>
5. </head>
6. <body>
7.
8. <h1>This is a Heading</h1>
9. <p>This is a paragraph.</p>
10.
11. </body>
12. </html>
```

## [خودتان امتحان کنید](#)

### کلام آخر

همانطور که اشاره شد برای ایجاد صفحات وب ابتدا باید کدهای html را در ویرایشگرهای html تایپ نموده و پس از ذخیره آن ها را در مرورگرهای خود تست کنید. از این رو ما مباحث این بخش را به **آشنایی با ادیتورهای html** اختصاص دادیم.

## جلسه ۰۳ : تگ های اصلی html

### مقدمه

با عرض سلام و وقت بخیر خدمت کاربران سایت [پی وی لرن](#) . و کاربرانی که [دوره کامل آموزش html](#) را دنبال می کنند. تاکنون در بخش های قبلی شما را با مفاهیم و ساختار کلی تگ ها و کاربرد html آشنا کردیم، همچنین مراحل کامل ایجاد و تست یک صفحه ی ساده ی html را در ویرایشگرهای ساده ای مانند notpad را ارائه کردیم. در ادامه ی آموزش ها به آشنایی با تگ های مختلف html و کاربرد هر یک می پردازیم. و در این جلسه از [آشنایی با تگ های اصلی html](#) شروع می کنیم.

### آشنایی با تگ های اصلی html

در ادامه ی مباحث این جلسه ما به بررسی چند نمونه از تگ های اصلی html و ساختار آن ها می پردازیم.

### اسناد HTML

تمام اسناد و صفحات html باید با تگ توصیفی `<!DOCTYPE html>` شروع شوند. هر کدام از اسناد html با تگ `<html>` شروع و با تگ `</html/>` پایان می یابد. بخش قابل مشاهده ی هر سند html که کاربر آن را می بیند بین تگ های `<body>` و `</body/>` قرار می گیرد.

```
1. <!DOCTYPE html>
2. <html>
3. <body>
4.
5. <h1>My First Heading</h1>
6.
7. <p>My first paragraph.</p>
8.
9. </body>
10. </html>
```

### [خودتان امتحان کنید](#)

### heading ها در html

heading های html با تگ های `<h1>` تا `<h6>` تعریف می شوند. از تگ `<h1>` برای تعریف هدینگ های بسیار مهم و برجسته و از تگ `<h6>` نیز برای مشخص کردن هدینگ های کم اهمیت استفاده می شود.

```
1. <!DOCTYPE html>
2. <html>
3. <body>
4.
5. <h1>This is heading 1</h1>
6. <h2>This is heading 2</h2>
7. <h3>This is heading 3</h3>
8. <h4>This is heading 4</h4>
9. <h5>This is heading 5</h5>
10. <h6>This is heading 6</h6>
11.
12. </body>
13. </html>
```

## خودتان امتحان کنید

### پاراگراف های HTML

پاراگراف های html با تگ `<p>` تعریف می شوند.

```
1. <!DOCTYPE html>
2. <html>
3. <body>
4.
5. <p>This is a paragraph.</p>
6. <p>This is another paragraph.</p>
7.
8. </body>
9. </html>
```

## خودتان امتحان کنید

### لینک های HTML

لینک های html با تگ `<a>` تعریف می شوند:

```
1. <!DOCTYPE html>
2. <html>
3. <body>
4.
5. <h2>HTML Links</h2>
6. <p>HTML links are defined with the a tag:</p>
7.
8. <a href="https://www.pvlearn.com">This is a link</a>
9.
10. </body>
11. </html>
```

## خودتان امتحان کنید

صفت `href` آدرسی که با آن لینک می شود را مشخص می کند. صفات برای ارائه اطلاعات اضافی در مورد عناصر HTML استفاده می شود.

### تصاویر HTML

تصاویر HTML با تگ `<img>` تعریف می شوند. فایل منبع (`src`)، متن جایگزین (`alt`) و `width` و `height` به عنوان ویژگی ها ارائه می شود:

```
12. 
```

## خودتان امتحان کنید

### button در HTML

button در HTML با تگ `<button>` تعریف می شوند.

```
1. <button>Click me</button>
```

## لیست های HTML

لیست های html با تگ `<ul>` (لیست های نامرتب/لیست علامتدار) و تگ `<ol>` (لیست های نامرتب/لیست شماره گذاری شده) و داخل آن ها آیتم های لیست با تگ `<li>` تعریف می شوند.

```
1. <ul>
2.   <li>Coffee</li>
3.   <li>Tea</li>
4.   <li>Milk</li>
5. </ul>
```

## [خودتان امتحان کنید](#)

### کلام آخر

برای طراحی بخش های مختلف یک صفحه وب تگ های متعدد و گسترده ای ارائه شده که از میان آن ها چند تگ به عنوان تگ های اصلی و پایه در هر صفحه وب وجود دارد از این رو در این بخش مباحث را به **آشنایی با تگ های اصلی html** اختصاص دادیم.

### مقدمه

با عرض سلام و وقت بخیر خدمت کاربران سایت [پی وی لرن](#) . و کاربرانی که [دوره کامل آموزش html](#) را دنبال می کنند. تاکنون شما را با تگ های اصلی و پایه در html که در تمام صفحات وب استفاده می شود و تا حدودی با ساختار آن ها آشنا کردیم. در این بخش نیز قصد داریم تا شما را با ساختار عناصر html و نکات مهمی که در پیاده سازی تگ های مختلف html باید رعایت کنید آشنا کنیم. از این رو برای آشنایی با این نکات مهم در ادامه ی مباحث [ساختار عناصر html](#) با ما همراه باشید.

### ساختار عناصر html

در ادامه ی این جلسه در راستای آشنایی با ساختار عناصر html شما را با ساختار و نکات مهمی در این زمینه آشنا می کنیم.

### عناصر html

یک عنصر HTML معمولا شامل یک تگ شروع و تگ پایان ، با محتوای وارد شده بین تگ شروع و پایان است:

`<tagname/>...محتوا<tagname>`

تمام عناصر html که شامل محتوای متن و... هستند دارای یک تگ شروع و پایان می باشند.

`<p/>My first paragraph.<p>`

پایان تگ	محتوای عنصر	شروع تگ
<code>&lt;h1/&gt;</code>	My First Heading	<code>&lt;h1&gt;</code>
<code>&lt;p/&gt;</code>	My first paragraph.	<code>&lt;p&gt;</code>
		<code>&lt;br&gt;</code>

**نکته :** عناصر HTML بدون محتوا، Empty elements (عناصر خالی) نامیده می شوند.

عناصر خالی تگ پایان ندارند، مانند عنصر `<br>` (که نشان دهنده یک شکست خطی به خط بعد است).

## عناصر html تودرتو

عناصر html را می توان به صورت تو در تو تعریف کرد.

تمام صفحات html دارای عناصر تو در تو می باشند.

مثال زیر از ۴ عنصر html تشکیل شده است:

شرح مثال فوق:

```
1. <!DOCTYPE html>
2. <html>
3. <body>
4.
5. <h1>My First Heading</h1>
6. <p>My first paragraph.</p>
7.
8. </body>
9. </html>
```

### خودتان امتحان کنید

تگ <html> کل سند html را در بر می گیرد. تگ <html> شامل تک شروع <html> و تگ پایان </html> است.

سایر محتوا که قابل مشاهده است در داخل تگ <body> قرار می گیرد:

```
1. <html>
2. <body>
3.
4. <h1>My First Heading</h1>
5. <p>My first paragraph.</p>
6.
7. </body>
8. </html>
```

عصر <body> در واقع بدنه ی سند html را توصیف می کند.

که با شامل تگ شروع <body> و تگ پایان </body> می باشد.

این صفحه وب شامل دو عنصر محتوایی <h1> و <p> است:

```
1. <body>
2.
3. <h1>My First Heading</h1>
4. <p>My first paragraph.</p>
5.
6. </body>
```

عصر <h1> یک هدینگ یا سر فصل را تعریف می کند.

این عنصر شامل تگ شروع <h1> و تگ پایان </h1> می باشد.

محتوای این عنصر My First Heading است:

```
1. <h1>My First Heading</h1>
```

عصر <p> یک پاراگراف را توصیف می کند.

این عنصر شامل تگ شروع `<p>` و تگ پایان `</p>` می باشد.

محتوای این عنصر My first paragraph است:

1. `<p>My first paragraph.</p>`

## تگ پایان را فراموش نکنید

برخی از عناصر HTML به درستی نمایش داده می شود، حتی اگر تگ پایان را فراموش کنید:

```
1. <html>
2. <body>
3.
4. <p>This is a paragraph
5. <p>This is a paragraph
6.
7. </body>
8. </html>
```

## خودتان امتحان کنید

مثال فوق در تمام مرورگرها کار می کند، زیرا برچسب بسته بودن اختیاری است.

**نکته:** هرگز در این مورد عادت نکن، چراکه اگر تگ پایان را فراموش کنید ممکن است نتایج غیر منتظره و / یا اشتباه ایجاد کنید.

## عناصر html خالی

عناصر html بدون محتوا به عنوان عناصر خالی شناخته می شوند.

مثلا تگ `<br>` یک نمونه تگ خالی است چرا که محتوایی ندارد.  
تگ های خالی نیز می توانند بسته باشند مانند: `<br/>`

در نظر داشته باشید که HTML5 عناصر خالی را نمی بندد.

اما اگر می خواهید اعتبار دقیقتری داشته باشید یا اگر بخواهید سند خود را با تجزیه کننده XML قابل خواندن کنید، باید تمام عناصر HTML را به درستی ببندید.

## از حروف کوچک در ایجاد تگ استفاده کنید.

تگ های HTML حساس به حروف بزرگ نیستند: `<P>` به معنی همان `<p>` است.

استاندارد HTML5 نیازی به تگ های کوچک و بزرگ نیست اما بهتر است که از حروف کوچک استفاده کنید چراکه برخی فرمت ها مانند XHTML در این مورد سخت گیر تر عمل می کنند.

## کلام آخر

برای شروع طراحی یک صفحه ی html بعد از انتخاب ویرایشگر و ایجاد فایل ، نخستین چیزی که باید بدانید و آن را رعایت کنید نکات اصلی کار با عناصر html و آشنایی با **ساختار عناصر html** است که ما در این بخش به آن پرداختیم.

### مقدمه

با عرض سلام خدمت کاربران سایت [پی وی لرن](#) . در این پست آموزشی قصد داریم به آموزش چگونگی **استفاده از صفات در html** یا کاربرد **صفات در html**، بپردازیم. صفات ( **Attributes** ) ، در html برای توصیف یک عنصر html بکار می رود. مواردی از جمله رنگ، اندازه، فونت ، **تصویر پس زمینه** و... توسط صفات در html برای یک عنصر html استفاده می شود. در html ، صفات هر عنصر در بخش شروع تعریف هر تگ قرار می گیرد.

البته توصیف عناصر در html را با استفاده از ابزاری نظیر [جاوااسکریپت](#) و ابزار [css در html](#) ، می توان انجام داد. استفاده از این ابزاری مزایایی نسبت به تعریف صفات در داخل تگ ها دارد.

در ادامه ی این مطلب به آموزش استفاده از صفات ( **Attributes** ) در html ، می پردازیم.



Attributes

### استفاده از صفات در html

در html صفات همیشه در قالب نام صفت و مقدار آن به صورت زیر تعریف می شود:

**name="value"**

اکنون به بررسی برخی از صفات پرکاربرد html می پردازیم:



## صفت href

لینک های html در داخل تگ <a> تعرف می شوند. و آدرس آنها در صفت href قرار می گیرد.

مثال : صفت href

1. `<a href="https://www.pvlearn.com">This is a link</a>`

## [خودتان امتحان کنید](#)

### صفت src

تصاویر در html با استفاده از تگ <img> شناخته می شوند. و آدرس دهی تصاویر به سایت توسط صفت src در این تگ انجام می شود.

مثال : صفت src

1. ``

## [خودتان امتحان کنید](#)

### صفت width و height

این دو صفت در html ، اندازه یک عنصر را توصیف می کنند.

مثال : اندازه عکس

1. ``

## [خودتان امتحان کنید](#)

### صفت alt

این صفت در html در واقع یک متن جایگزین را برای تصاویر ، تعریف می کند، در صورتی تصویر به هر علتی نمایش داده نشود متن جایگزین به جای تصویر قرار می گیرد.

مثال : صفت alt

1. ``
2. : برای توصیف تصاویری که نمایش داده نشده اند نیز استفاده می شود alt از صفت

## [خودتان امتحان کنید](#)

مثال : کاربرد صفت alt

1. ``

## صفت style

از این صفت در html در واقع برای استفاده از کدهای CSS به منظور تغییر دادن و ایجاد جلوه های خاص در ظاهر عناصر html به کار می رود.

مثال: قرمز کردن رنگ یه عنصر:

مثال : صفت style

1. `<p style="color:red">I am a paragraph</p>`

## [خودتان امتحان کنید](#)

## صفت lang

از این صفت برای توصیف کردن زبان وب سایت در داخل تگ html استفاده می شود.

مثال: تعریف زبان انگلیسی برای وب سایت:

مثال :

```
1. <!DOCTYPE html>
2. <html lang="en-US">
3. <body>
4. ...
5. </body>
6. </html>
```

## صفت title

از این صفت برای بیان عنوان در یک عنصر html استفاده می شود.

مثال: تعریف عنوان برای تگ <p>:

مثال : صفت title

```
1. <p title="I'm a tooltip">
2. This is a paragraph.
3. </p>
```

## [خودتان امتحان کنید](#)

### توصیه ی ما : استفاده از حروف کوچک

شما در کدنویسی html در استفاده از حروف کوچک یا بزرگ مختارید، اما ما توصیه می کنیم که برای راحتی کار از حروف کوچک بخصوص در XHTML استفاده کنید.

### نوشتن مقادیر صفات در دابل کوتیشن

در استاندارد HTML5 نوشتن مقادیر صفات بدون استفاده از دابل کوتیشن (") نیز پشتیبانی می شود.

اما بهتر است به دلایلی از جمله افزایش خوانایی مقادیر صفات عناصر html را بین دابل کوتیشن (") قرار دهید.

مثال : حالت غیر استاندارد

```
1. <a href=https://www.pvlearn.com>
```

## [خودتان امتحان کنید](#)

مثال : حالت استاندارد

```
1. <a href="https://www.pvlearn.com">
```

## [خودتان امتحان کنید](#)

### تک کوتیشن و دابل کوتیشن

در اکثر موارد بویژه در مقادیر صفات html از دابل کوتیشن استفاده می شود، اما تک کوتیشن هم می تواند استفاده شود.

در مواردی که مقدار صفت خود شامل کوتیشن می شود، باید از تک کوتیشن در مقدار صفت استفاده کرد.

مثال :

```
1. <p title='John "ShotGun" Nelson'>
```

و یا :

مثال : حالت صحیح

```
1. <p title="John 'ShotGun' Nelson">
```

### خلاصه ی این بخش :

تمام عناصر HTML می توانند دارای صفات باشند. ویژگی `title` دارای اطلاعات " `tool-tip` اضافی است. ویژگی `href` اطلاعات آدرس را برای لینک ها فراهم می کند. ویژگی های `width` و `height` اطلاعات اندازه را برای تصاویر ارائه می کنند. ویژگی `alt` متن جایگزین را برای خوانندگان صفحه فراهم می کند. در آموزش های این دوره، ما همیشه از حروف کوچک استفاده می کنیم. در آموزش های این دوره، ما همیشه مقادیر ویژگی با نقل قول ها را نقل می کنیم.

### سخن پایانی

صفات ( Attributes ) ، مهمترین و اصلی ترین بخش های هر عنصر html است. و تصور یک سایت کارآمد بدون استفاده از صفات به هر شکلی، تقریباً غیر ممکن است. از این رو پی وی لرن ، اقدام به انتشار این مطلب مفید و کارآمد در خصوص آشنایی به طریقه استفاده از صفات در html کرده است.

### مقدمه

با عرض سلام خدمت کاربران سایت [پی وی لرن](#) . در آموزشهای قبلی اشاره ای داشتیم به [کاربرد صفات در html](#) . همچنین با نحوه ی [استفاده از فرم ها در html](#) ، آشنا شدید. در این بخش از سری آموزشهای کامل html قصد داریم به چگونگی [کار با سربرگها \( Headings در html](#) ، بپردازیم. سربرگ ها ( Headings ) در واقع مشخص کننده ی یک سر فصل یا یک بخش مجزا در یک صفحه وب است. و خود سرفص ممکن است دارای زیر مجموعه های کوچکتری باشد.

سربرگ ها در html در بازه ای بین تگ های **<h1>** تا **<h6>** ، تعریف می شوند. تگ **<h1>** بزرگترین سربرگ و **<h6>** کوچکترین آن هاست.

در ادامه این مطلب به بررسی و تحلیل هر کدام از سربرگ ها ی **<h1>** تا **<h6>** ، می پردازیم.

**<h1>Heading 1</h1>**

**<h2>Heading 2</h2>**

**<h3>Heading 3</h3>**

**<h4>Heading 4</h4>**

**<h5>Heading 5</h5>**

**<h6>Heading 6</h6>**

headings

## کار با سربرگها ( Headings ) در html

در ترتیب استفاده از سربرگ ها <h1> مهمترین سربرگ ها را در یک وب سایت و <h6> نیز کوچکترین و کم اهمیت ترین سربرگ ها را در یک وب سایت مشخص می کند:

مثال : سرفصل ها

```
1. <!DOCTYPE html>
2. <html>
3. <body>
4. <h1>Heading 1</h1>
5. <h2>Heading 2</h2>
6. <h3>Heading 3</h3>
7. <h4>Heading 4</h4>
8. <h5>Heading 5</h5>
9. <h6>Heading 6</h6>
10. </body>
11. </html>
```

### خودتان امتحان کنید

## اهمیت سربرگ ها

موتورهای جستجو در واقع از سربرگ ها ، برای فهرست کردن ساختار و محتوای وب سایت شما ، استفاده می کنند.

کاربران نیز از سربرگ های وب سایت ها، برای رسیدن به مطالب مورد نظر خود استفاده می کنند ، پس این خیلی مهم است، که چگونه از سربرگ ها برای اعلام محتوای بخش های مختلف وب سایت خود استفاده کنید.

## اندازه سربرگ

هر کدام از سربرگ ها در html یک اندازه پیش فرض دارند، با این وجود شما می توانید اندازه سربرگ های خود را با استفاده از [صفت style](#) ، و کد CSS اندازه فونت (`font-size`) ، تغییر دهید.

مثال : اندازه ها

```
1. <h1>Heading 1</h1>
2. <h2>Heading 2</h2>
3. <h3>Heading 3</h3>
4. <h4>Heading 4</h4>
5. <h5>Heading 5</h5>
6. <h6>Heading 6</h6>
```

### خودتان امتحان کنید

## بزرگترین اندازه ی سر فصل

هر heading (عنوان) HTML دارای اندازه پیش فرض است. با این حال، با استفاده از خصوصیت اندازه فونت CSS می توانید برای هر عنوان با ابعاد style مشخص کنید:

مثال : بزرگترین اندازه ی heading

```
1. <h1 style="font-size:60px;">Heading 1</h1>
```

## خودتان امتحان کنید

### محتوای افقی در html

از تگ `<hr>` ، برای شکستن محتوا ، و انتقال به خط بعدی استفاده میشود. و بیشتر برای نمایش افقی محتوا استفاده می شود.

تگ `<hr>` ، در واقع برای جدا کردن بخش های مختلف یک محتوا استفاده می شود.

مثال : محتوای افقی

```
1. <h1>This is heading 1</h1>
2. <p>This is some text.</p>
3. <hr>
4. <h2>This is heading 2</h2>
5. <p>This is some other text.</p>
6. <hr>
```

## خودتان امتحان کنید

### تگ `<head>`

تگ `<head>` ، در واقع هیچ ربطی به سربرگ ها ندارد. این تگ در واقع یک ظرف برای نگهداری تگ های کلیدی از جمله `<title>` ، `<link>` و ... می باشد.

تگ `<head>` ، بین تگ های `<html>` و `<body>` قرار می گیرد.

مثال : تگ head

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>My First HTML</title>
5. <meta charset="UTF-8">
6. </head>
7. <body>
8. .
9. .
10. .
```

## خودتان امتحان کنید

### مشاهده ی سورس کد html صفحات وب

جهت مشاهده ی سورس کد html صفحات وب، در صفحه ی HTML راست کلیک کرده و گزینه ی "View Page Source" (در Chrome) یا گزینه ی "View Source" (در IE) را انتخاب کنید.

### بررسی ساختار و عناصر HTML صفحات وب

در صفحه وب مربوطه روی یک عنصر و یا فضای خالی راست کلیک کرده و گزینه ی "Inspect" یا "Inspect Element" را برای مشاهده ی ساختار و عناصر HTML صفحه وب مربوطه انتخاب کنید.

این قابلیت به شما اجازه ی تغییر کدهای HTML و CSS صفحه وب مربوطه را می دهد.

## کلام آخر

محتوای اکثر صفحات وب از مقالات ، تحقیقات ، آموزش ها و ... تشکیل می شوند که به ایجاد عناوینی در قسمت های مختلف محتوا منجر می شود، برای اینگونه موارد استفاده از **تگ heading در HTML** بهترین گزینه است.



### مقدمه

با عرض سلام و وقت بخیر خدمت کاربران سایت [پی وی لرن](#) . و کاربرانی که [دوره کامل آموزش html](#) را دنبال می کنند. محتوای صفحات وب از مطالبی از جمله متن، تصویر، ویدئو، صوت و... تشکیل شده است. یکی از اصلی ترین بخش های هر صفحه وب متن و پاراگراف است که سریع تر از هر محتوای دیگری بارگذاری شده و کاربر آن را می بیند. همچنین پاراگراف ها مطالب و مفاهیم موضوعات هر صفحه را به کاربران منتقل می کنند از این بابت از اهمیت بسیاری برخوردارند. برای **آشنایی با پاراگراف در html** و انواع تگ های پاراگرافی و نحوه شکستن خطوط در html و ساختار آن در ادامه ی مباحث با ما همراه باشید.

### آشنایی با پاراگراف در html

در ادامه ی مباحث شما را با تگ های تنظیم پاراگراف در html و مطالب مربوط به آن آشنا خواهیم کرد.

### پاراگراف های html

عنصر `<p>` در html یک پاراگراف را توصیف می کند.

مثال : تگ p

1. `<p>This is a paragraph.</p>`
2. `<p>This is another paragraph.</p>`
3. `<strong>نکته</strong>` : مرورگرها به طور خودکار برخی از فضای سفید (حاشیه) را قبل و بعد از یک پاراگراف اضافه می کنند.

### [خودتان امتحان کنید](#)

### نمایش در html

شما نمی توانید مطمئن باشید که HTML نمایش داده خواهد شد. صفحه های بزرگ یا کوچک و تغییر اندازه در پنجره های نتایج مختلفی ایجاد می کنند.

با HTML، شما نمی توانید خروجی را با افزودن فضاهای اضافی یا خطوط اضافی در کد HTML خود تغییر دهید. مرورگر فضاهای اضافی و خطوط اضافی را هنگامی که صفحه نمایش داده می شود حذف می کند.

مثال : نمایش تگ پایان

1. `<p>`
2. This paragraph
3. contains a lot of lines
4. in the source code,
5. but the browser
6. ignores it.
7. `</p>`
8. `<p>`
9. This paragraph
10. contains a lot of spaces
11. in the source code,
12. but the browser
13. ignores it.
14. `</p>`

### [خودتان امتحان کنید](#)

## تگ پایان

بیشتر مرورگرها محتوای html را حتی در صورت عدم بستن تگ، نمایش می دهند:

مثال : تگ پایان

1. `<p>This is a paragraph.`
2. `<p>This is another paragraph.`
3. مثال فوق در اکثر مرورگرها کار خواهد کرد، اما سعی نکنید که به این صورت عمل کنید.

### [خودتان امتحان کنید](#)

## شکستن خطوط در html

تگ `<br>` در html برای شکستن خطوط و یا انتقال به خط بعدی استفاده می شوند.

مثال : شکستن خطوط

1. `<p>This is<br>a paragraph<br>with line breaks.</p>`

### [خودتان امتحان کنید](#)

تگ `<br>` یک عنصر خالی html است به این معنا که نیازی به بستن تگ ندارد.

## مشکل تایپ شعر

شعر زیر در یک خط نمایش داده خواهد شد:

مثال : مشکل تایپ شعر

1. `<p>`
2. `My Bonnie lies over the ocean.`
3. `My Bonnie lies over the sea.`
4. `My Bonnie lies over the ocean.`
5. `Oh, bring back my Bonnie to me.`
6. `</p>`

### [خودتان امتحان کنید](#)

## تگ `<pre>` در html

عنصر `<pre>` در html متن را درست همانطور که تایپ شده نمایش می دهد.

متن داخل عنصر `<pre>` در یک فونت ثابت (معمولا بیگ) نمایش داده می شود و فضای خالی و اندازه خطوط را حفظ می کند.

1. <pre>
2. My Bonnie lies over the ocean.
3. My Bonnie lies over the sea.
4. My Bonnie lies over the ocean.
5. Oh, bring back my Bonnie to me.
6. </pre>

### [خودتان امتحان کنید<>](#)

## تگ های ایجاد مدیریت پاراگراف

تگ های زیر در ایجاد و تنظیم پاراگراف پرکاربرد هستند :

توضیحات	تگ
یک پاراگراف را تعریف می کند.	<p>
خط جاری را به خط بعدی می شکند.	 
متن را در همان شکلی که تایپ شده نمایش می دهد.	<pre>

## کلام آخر

با توجه با اینکه متن و پاراگراف ها بخش قابل توجهی از هر صفحه وب را تشکیل داده و از مهمترین محتوا در هر صفحه وب می باشند از این رو مباحث این بخش را به آشنایی با **پاراگراف در html** و دو تگ پرکاربرد درج پاراگراف اختصاص دادیم.

مباحث بخش بعد را به Style ها در html اختصاص داده ایم.

### مقدمه

با عرض سلام و وقت بخیر خدمت کاربران سایت [پی وی لرن](#) . و کاربرانی که [دوره کامل آموزش html](#) را دنبال می کنند. همانطور که در بخش های قبلی نیز عنوان کردیم عناصر html بخش های مختلف هر صفحه وب را تشکیل می دهند، اما در حالت عادی این عناصر هیچ style خاصی ندارند و اغلب آن ها در حالت ساده دیده نمی شوند. برای دیده شدن عناصر و تعریف ظاهر برای هر عنصر باید با نحوه ی کار با style در html آشنایی داشته و از این طریق برای هر عنصر ظاهر و شکل مورد نظر را تعریف کنید. از این رو style در html از اصلی ترین بخش های هر صفحه وب بوده و از اهمیت ویژه ای برخوردار است. در ادامه ی این آموزش به منظور آشنایی با چگونگی **کار با style در html** با ما همراه باشید.

### کار با style در html

در ادامه ی این مباحث برای داشتن یک آشنایی از نحوه کار با style در html به چند style ساده اما پرکاربرد در html پرداخته ایم.

مثال : استایل متن

1. I am Red
2. I am Blue
3. I am Big

### [خودتان امتحان کنید](#)

### صفات style در html

تنظیم style یک عنصر HTML را می توان با ویژگی style انجام داد.

ساختار تعریف صفت style در: html:

مثال :

1. `<tagname style="property:value;">`

در ساختار فوق *property* یک ویژگی CSS و *value* نیز یک مقدار CSS است.

### رنگ پس زمینه در html

خصوصیت `background-color` رنگ پس زمینه را برای یک عنصر HTML تعریف می کند. این مثال رنگ پس زمینه برای یک صفحه را به powderblue تغییر می دهد:

مثال : رنگ پس زمینه

1. `<body style="background-color:powderblue;">`
2. `<h1>This is a heading</h1>`
3. `<p>This is a paragraph.</p>`
4. `</body>`

### [خودتان امتحان کنید](#)

## رنگ متن در html

صفت color یک رنگ متن را برای متن عناصر html تعریف می کند:

مثال : رنگ متن

1. `<h1 style="color:blue;">This is a heading</h1>`
2. `<p style="color:red;">This is a paragraph.</p>`

## [خودتان امتحان کنید<<](#)

### فونت ها در html

ویژگی `font-family` فونت متن عناصر html را تعریف می کند.

مثال : استایل فونت ها

1. `<h1 style="font-family:verdana;">This is a heading</h1>`
2. `<p style="font-family:courier;">This is a paragraph.</p>`

## [خودتان امتحان کنید<<](#)

### اندازه ی متن در html

ویژگی `font-size` اندازه ی فونت متن عناصر html را توصیف می کند.

مثال : اندازه ی متن

1. `<h1 style="font-size:300%;">This is a heading</h1>`
2. `<p style="font-size:160%;">This is a paragraph.</p>`

## [خودتان امتحان کنید<<](#)

### تراز بندی متن در html

ویژگی `text-align` برای تراز بندی متن های عناصر html بکار می رود.

مثال : تراز بندی متن

1. `<h1 style="text-align:center;">Centered Heading</h1>`
2. `<p style="text-align:center;">Centered paragraph.</p>`

## [خودتان امتحان کنید<<](#)

### کلام آخر

همانطور که در متن جلسه نیز اشاره کردیم ، عناصر html در حالت عادی هیچگونه نمایشی ندارند. برای نشان دادن آن ها و ایجاد حالات مختلف نمایشی باید با چگونگی **کار با style در html** آشنایی داشته و از آن برای ظاهر سازی عناصر html استفاده کنید.

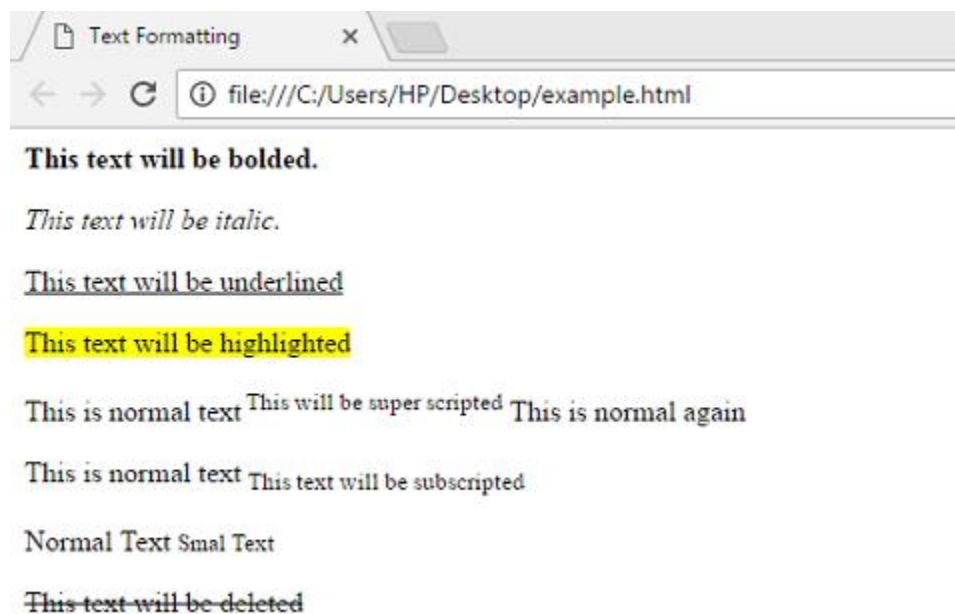
## جلسه ۰۹ : قالب بندی متن در html

### مقدمه

با عرض سلام خدمت کاربران سایت [پی وی لرن](#) . در مطالب قبلی دوره آموزش html ، نگاهی داشتیم. به [کاربرد سربرگها \(Headings\)](#) در [html](#) ، در ادامه آموزشها ، در این بخش به بحث [آموزش قالب بندی متن در html](#) ، می پردازیم. html تگ هایی را برای توصیف متن در جاهای مختلف و با کاربردهای مختلف در اختیار کاربران قرار داده است [تگ های قالب بندی متن](#) یا [tag های قالب بندی متن](#) از جمله تگ هایی است که می توانند ظاهری زیبا و متمایز به متن ما بدهند.

تگ های پر کاربردی نظیر تگ `<i>` و تگ `<b>` ، برای ایجاد افکت های *italic* و **bold** در قسمتی از متن بکار می رود.

در ادامه این مطلب ، به بررسی تگ های قالب بندی متن ، می پردازیم.



html

## آموزش قالب بندی متن در html

مثال : قالب بندی متن

1. This text is bold
- 2.
3. This text is italic
- 4.
5. This is subscript and superscript

[خودتان امتحان کنید <<](#)

تگ هایی برای قالب بندی های مختلف متن در html ارائه شده است. که هر کدام در جاهای مختلفی کاربرد دارد:

- **<b>** : این تگ متن را **Bold** می کند
- **<strong>** : این تگ برای قسمت مهم متن استفاده می شود.
- **<i>** : این تگ برای *italic* کردن متن بکار می رود.
- **<em>** : این تگ برای تاکید کردن روی یک متن بکار می رود.
- **<mark>** : این تگ برای علامت زدن یک متن استفاده می شود.
- **<small>** : این تگ برای کوچک کردن یک متن بکار می رود.
- **<del>** : این تگ برای حذف یک متن بکار می رود.
- **<ins>** : برای درج یک متن کاربرد دارد.
- **<sub>** : برای افزودن زیر مجموعه کاربرد دارد.
- **<sup>** : برای متن های ستاره دار بکار می رود.

تگ های **<b>** و **<strong>**

تگ **<b>** برای Bold کردن متن استفاده می شود.

مثال : ضخیم کردن متن

1. `<b>This text is bold</b>`

[خودتان امتحان کنید<<](#)

تگ **<strong>**

تگ **<strong>** برای متمایز کردن بخشهای مهم متن بکار می رود.

مثال : متن مهم

1. `<strong>This text is strong</strong>`

[خودتان امتحان کنید<<](#)

تگ های **<i>** و **<em>**

تگ **<i>** برای ایتالیک کردن یک متن بکار می رود.

مثال : ایتالیک متن

1. `<i>This text is italic</i>`

[خودتان امتحان کنید<<](#)

تگ `<em>` برای نشان دادن تاکید روی یک متن بکار می رود.

مثال : مشابه ایتالیک

1. `<em>This text is emphasized</em>`

[خودتان امتحان کنید<<](#)

تگ `<small>`

تگ `<small>` برای کوچکتر کردن یک متن بکار می رود.

مثال : متن کوچکتر

1. `<h2>HTML <small>Small</small> Formatting</h2>`

[خودتان امتحان کنید<<](#)

تگ `<mark>`

این تگ برای علامتدار کردن متن بکار می رود.

مثال : متن علامتدار

1. `<h2>HTML <mark>Marked</mark> Formatting</h2>`

[خودتان امتحان کنید<<](#)

تگ `<del>`

این تگ برای حذف یک متن بکار می رود.

مثال : متن حذف شده

1. `<p>My favorite color is <del>blue</del> red.</p>`

[خودتان امتحان کنید<<](#)

تگ `<ins>`

از این تگ برای درج متن در محتوا ، استفاده می شود.

مثال : متن زیر خط دار

1. `<p>My favorite <ins>color</ins> is red.</p>`



## تگ <sub>

از این تگ برای اعمال ویژگی subscripted (متن پایین تر) در یک متن بکار می رود.

مثال : متن پایین نویس

1. `<p>This is <sub>subscripted</sub> text.</p>`

## [خودتان امتحان کنید](#)

## تگ <sup>

از این تگ برای اعمال ویژگی subscripted (متن بالاتر) در یک متن بکار می رود.

مثال : متن بالانویس

1. `<p>This is <sup>superscripted</sup> text.</p>`

## [خودتان امتحان کنید](#)

با کشیدن خطی رو متن مورد نظر آن را حذف شده نشان می دهد.	<del>
با highlight کردن زمینه ی یک متن آن را مشخص می کند.	<mark>

## کلام آخر

در صفحات html متن یکی از اصلی ترین بخش های هر محتوایی را تشکیل می دهد که مطالب وب سایت را به کاربران انتقال می دهد، از این رو **قالب بندی متن در html** از اهمیت ویژه ای برخوردار می باشد.



### مقدمه

با عرض سلام خدمت کاربران سایت [پی وی لرن](#)، در این قسمت، با چگونگی کار با تگ های Citation و Quotation می پردازیم. تگ های Citation و Quotation، برای درج نقل قول و استناد در html در یک متن بکار می رود. زمانیکه می خواهید سخنی از یک شخص را با ذکر نام و نشان آن شخص، در متن بگنجانید، از این تگ می توانید استفاده کنید. از تگ های Citation و Quotation، برای درج استناد یک متن و ارائه مرجع نیز، بکار می رود.

### بررسی تگ های Citation و Quotation

#### مثال : نقل قول ها

1. Here is a quote from WWF's website:
- 2.
3. For 50 years, WWF has been protecting the future of nature.
4. The world's leading conservation organization,
5. WWF works in 100 countries and is supported by 1.2 million members in the United States
6. and close to 5 million globally.

#### [خودتان امتحان کنید](#)

در اینجا با تگ هایی که دارای ویژگی Citation و Quotation هستند با ذکر مثالهایی آشنا می شویم:

#### تگ <q>

تگ <q> برای درج یک نقل قول کوتاه بکار می رود.

#### مثال : تگ q

1. <p>WWF's goal is to: <q>Build a future where people live in harmony with nature.</q></p>

#### [خودتان امتحان کنید](#)

#### تگ <blockquote>

از تگ <blockquote> برای درج بخشی که از یک منبع دیگری نقل قول شده است استفاده می شود.

#### مثال : نقل قول از سایر منابع

1. <p>Here is a quote from WWF's website:</p>
2. <blockquote cite="http://www.worldwildlife.org/who/index.html">
3. For 50 years, WWF has been protecting the future of nature.
4. The world's leading conservation organization,
5. WWF works in 100 countries and is supported by
6. 1.2 million members in the United States and
7. close to 5 million globally.
8. </blockquote>

## [خودتان امتحان کنید](#)

### تگ <abbr>

از این تگ برای درج یک اختصار یا مخفف یک کلمه استفاده می شود.

مثال : افزودن عنوان به متن

1. <p>The <abbr title="World Health Organization">WHO</abbr> was founded in 1948.</p>

## [خودتان امتحان کنید](#)

### تگ <address>

از این تگ برای درج اطلاعات تماس (نویسنده / مالک) ، درباره ی یک سند یا یک متن علمی، استفاده می شود.

مثال : آدرس و مشخصات نویسنده

1. <address>
2. Written by John Doe.<br>
3. Visit us at:<br>
4. Example.com<br>
5. Box 564, Disneyland<br>
6. USA
7. </address>

## [خودتان امتحان کنید](#)

### تگ <cite>

از این تگ نیز برای درج عنوان برای یک کار استفاده می شود. (مرورگر های متنی که در این عنصر قرار بگیرد را بصورت ایتالیک نمایش می دهند).

مثال : عنوان یک اثر

1. <p><cite>The Scream</cite> by Edvard Munch. Painted in 1893.</p>

## [خودتان امتحان کنید](#)

### تگ <bdo> در html

از تگ <bdo> برای تعریف جهت گیری دو طرفه (directional) در متن استفاده می شود.

مثال : تنظیم راست یا چپ چینی متن

1. <bdo dir="rtl">This text will be written from right to left</bdo>

## [خودتان امتحان کنید](#)

## عناصر استناد و نقل قول در html

توضیحات	تگ
اختصار را تعریف می کند	<abbr>
اطلاعات تماس را برای نویسنده / صاحب یک سند تعریف می کند.	<address>
جهت گیری متن را تعریف می کند.	<bdo>
یک بخش را که از منبع دیگری نقل قول شده تعریف می کند.	<blockquote>
برای یک متن عنوانی را تعریف می کند.	<cite>
یک نقل قول خطی کوتاه را تعریف می کند.	<q>

### کلام آخر

در وب صفحاتی ویی که محتوای آن ها شامل مقالات بین المللی ، عملی و یا نوشته هایی از نویسندگان می باشند، باید از قالب نقل قول از صاحب سند و همچنین اطلاعات نویسنده در متن استفاده کند، که **تگ های Citation و Quotation** بهترین گزینه برای اینگونه فرمت هاست.

### مقدمه

با عرض سلام و وقت بخیر خدمت کاربران سایت [پی وی لرن](#) . و کاربرانی که [دوره کامل آموزش html](#) را دنبال می کنند. یکی از بخش هایی که در کدهای html به ویژه در موضوعات آموزشی زیاد کاربرد دارد بخش [توضیحات در html](#) است، این بخش شامل متن و کدهایی است که در یک قالب خاص قرار می گیرد تا مرورگر از کامپایل و اجرای آن ها صرف نظر کند. این ویژگی به شما این امکان را می دهد تا در صورت لزوم در بخش های مختلف از html صفحات وب توضیحات و کامنت های مورد نظر خود را به همراه کد و تگ های مربوطه درج کنید. برای آشنایی با طرز کار با کامنت در html در ادامه این آموزش با ما همراه باشید.

### آشنایی با کامنت در html

در این بخش ما شما را با ساختارهای مختلف توضیحات در html و نحوه ی تنظیم آن ها آشنا خواهیم کرد.

### تگ های کامنت در html

شما می توانید در کد html خود با استفاده از ساختار زیر توضیحات و کامنت مربوطه را اضافه کنید:

مثال :

```
1. <!-- Write your comments here -->
```

توجه داشته باشید که نقطه علامت تعجب (!) در تگ باز است.

نکته :

توضیحات در html توسط مرورگر نمایش داده نمی شوند، اما آنها می توانند سند HTML کد منبع را به شما معرفی کنند. با استفاده از کامنت در html شما می توانید اعلان ها و یادآوری ها را در HTML خود قرار دهید:

مثال : مثال توضیحات تکی

```
1. <!-- This is a comment -->
2.
3. <p>This is a paragraph.</p>
4.
5. <!-- Remember to add more information here -->
```

### [خودتان امتحان کنید<>](#)

یکی از مهمترین کاربردهای کامنت در html درج نکات راهنمایی برای کاربر و ابهام زدایی کدهای مبهم است.

```
1. <!-- Do not display this at the moment  
2.   
3. -->
```

[خودتان امتحان کنید<<](#)

### کلام آخر

اگر به دنبال درج نکات مهم و توضیحاتی در مورد کدها و بخش های مختلف html هستید. باید با چگونگی استفاده از **توضیحات در html** آشنایی داشته باشید، از این رو مباحث این بخش را به آشنایی با ساختار بکارگیری کامنت در html اختصاص دادیم.

## مقدمه

با عرض سلام خدمت کاربران سایت [پی وی لرن](#) . در این پست قصد داریم به آموزش نحوه کار با [رنگ ها در HTML](#) بپردازیم. تعریف و رنگ بندی تگ ها و بخش های یک وب سایت از مباحثی است که در جذابیت و زیبایی صفحه وب نقش بسیار مهمی دارد. صفحه وبی که دارای رنگ های متنوع و مرتبط نباشد، خسته کننده و کسل کننده است. در آموزش های قبلی چگونگی کار با تگ های [Table](#) , [Div](#) و تگ [img](#) را در صفحات html را توضیح دادیم. در ادامه این پست نیز به آموزش تعریف رنگ های html برای بخش های مختلف صفحه وب، می پردازیم.

## رنگ ها در HTML

رنگ های html با استفاده از نام رنگ های از پیش تعریف شده یا RGB، HEX، HSL، RGBA، HSLA مشخص می شوند. رنگ ها در HTML می تواند با استفاده از نام آن رنگ مشخص شود:



### مثال : رنگ ها در html

1. `<h1 style="background-color:Tomato;">Tomato</h1>`
2. `<h1 style="background-color:Orange;">Orange</h1>`
3. `<h1 style="background-color:DodgerBlue;">DodgerBlue</h1>`
4. `<h1 style="background-color:MediumSeaGreen;">MediumSeaGreen</h1>`
5. `<h1 style="background-color:Gray;">Gray</h1>`
6. `<h1 style="background-color:SlateBlue;">SlateBlue</h1>`
7. `<h1 style="background-color:Violet;">Violet</h1>`
8. `<h1 style="background-color:LightGray;">LightGray</h1>`

## [خودتان امتحان کنید<<](#)

### تنظیم رنگ پس زمینه

شما می توانید رنگ پس زمینه برای عناصر و تگ های HTML را تنظیم کنید:

مثال:

### مثال : تنظیم رنگ پس زمینه

1. `<h1 style="background-color:DodgerBlue;">Hello World</h1>`
2. `<p style="background-color:Tomato;">Lorem ipsum...</p>`

## [خودتان امتحان کنید<<](#)



## رنگ متن

شما همچنین می توانید، رنگ متن ها را تعیین کنید:

مثال : رنگ متن

1. `<h1 style="color:Tomato;">Hello World</h1>`
2. `<p style="color:DodgerBlue;">Lorem ipsum...</p>`
3. `<p style="color:MediumSeaGreen;">Ut wisi enim...</p>`

[خودتان امتحان کنید<<](#)

## رنگ Border

می توانید برای border کادر دور عناصر html ، رنگ تعریف کنید

مثال:

مثال : رنگ border

1. `<h1 style="border:2px solid Tomato;">Hello World</h1>`
2. `<h1 style="border:2px solid DodgerBlue;">Hello World</h1>`
3. `<h1 style="border:2px solid Violet;">Hello World</h1>`

[خودتان امتحان کنید<<](#)

this is the paragraph with thin border width

this is the paragraph with medium border width

this is the paragraph with thick border width

– border-color رنگ ها در HTML

## مقادیر رنگ ها در HTML

در HTML رنگها نیز می توانند با استفاده از مقادیر **RGB**، مقادیر **HEX**، مقادیر **HSL**، مقادیر **RGBA** و مقادیر **HSLA** مشخص شوند. همانند نام رنگ "Tomato" :

`rgb(255, 99, 71)`

`#ff6347`

`hsl(9, 100%, 64%)`

رنگ های فوق را می توان با شفافیت بیشتر نیز نشان داد :

`rgba(255, 99, 71, 0.5)`

`hsla(9, 100%, 64%, 0.5)`

مثال : مقادیر رنگ ها

1. `<h1 style="background-color:rgb(255, 99, 71);">...</h1>`
2. `<h1 style="background-color:#ff6347;">...</h1>`
3. `<h1 style="background-color:hsl(9, 100%, 64);">...</h1>`
- 4.
5. `<h1 style="background-color:rgba(255, 99, 71, 0.5);">...</h1>`
6. `<h1 style="background-color:hsla(9, 100%, 64, 0.5);">...</h1>`

## [خودتان امتحان کنید<<](#)

### مقدار RGB

در HTML، یک رنگ را می توان به عنوان یک مقدار **RGB** تعریف کرد، با استفاده از این فرمول:

`blue,green rgb(red,)`

در فرمول بالا هر کدام از پارامترهای **red** , **green** , **blue** می تواند مقادیر بین ۰ تا ۲۵۵ را داشته باشد.

برای مثال مقدار (۰, ۰, ۲۵۵) رنگ **قرمز** را نمایش می دهد.

برای رنگ **سیاه** باید تمام پارامترها را به صفر تنظیم کنید مانند: `rgb(0, 0, 0.)`

برای رنگ **سفید** باید تمام پارامترها را به ۲۵۵ تنظیم کرد مانند: `rgb(255, 255, 255.)`

1. `<h1 style="background-color:rgb(255, 0, 0);">rgb(255, 0, 0)</h1>`
2. `<h1 style="background-color:rgb(0, 0, 255);">rgb(0, 0, 255)</h1>`
3. `<h1 style="background-color:rgb(60, 179, 113);">rgb(60, 179, 113)</h1>`
4. `<h1 style="background-color:rgb(238, 130, 238);">rgb(238, 130, 238)</h1>`
5. `<h1 style="background-color:rgb(255, 165, 0);">rgb(255, 165, 0)</h1>`
6. `<h1 style="background-color:rgb(106, 90, 205);">rgb(106, 90, 205)</h1>`

[خودتان امتحان کنید<<](#)

مثال : مثال دوم از RGB

1. `<h1 style="background-color:rgb(0, 0, 0);">rgb(0, 0, 0)</h1>`
2. `<h1 style="background-color:rgb(60, 60, 60);">rgb(60, 60, 60)</h1>`
3. `<h1 style="background-color:rgb(120, 120, 120);">rgb(120, 120, 120)</h1>`
4. `<h1 style="background-color:rgb(180, 180, 180);">rgb(180, 180, 180)</h1>`
5. `<h1 style="background-color:rgb(240, 240, 240);">rgb(240, 240, 240)</h1>`
6. `<h1 style="background-color:rgb(255, 255, 255);">rgb(255, 255, 255)</h1>`

[خودتان امتحان کنید<<](#)**مقدار HEX**

در html می توان رنگ را بر مبنای Hex تعریف کرد مانند فرمول زیر:

**rrggbb #**

جایی که rr (قرمز)، gg (سبز) و bb (آبی) مقادیر هگزادسیمال بین ۰۰ و ff هستند (مانند دهدهی ۰۰-۲۵۵).

به عنوان مثال، # ff0000 به صورت قرمز نمایش داده می شود، زیرا قرمز به بالاترین مقدار (ff) و دیگران به پایین ترین مقدار (۰۰) تنظیم شده است.

مثال:

**ff0000 #****ff۰۰۰۰#****cb371۳#****ee82ee #****ffa500 #****a5acd۶#**

مثال : مثال HEX

1. `<h1 style="background-color:#ff0000;">#ff0000</h1>`
2. `<h1 style="background-color:#0000ff;">#0000ff</h1>`
3. `<h1 style="background-color:#3cb371;">#3cb371</h1>`
4. `<h1 style="background-color:#ee82ee;">#ee82ee</h1>`
5. `<h1 style="background-color:#ffa500;">#ffa500</h1>`
6. `<h1 style="background-color:#6a5acd;">#6a5acd</h1>`

برای رنگ سفید باید تمام پارامترها را به ۲۵۵ تنظیم کرد مانند: `rgb(255, 255, 255.)`

مثال : رنگ بندی خاکستری

1. `<h1 style="background-color:#000000;">#000000</h1>`
2. `<h1 style="background-color:#3c3c3c;">#3c3c3c</h1>`
3. `<h1 style="background-color:#787878;">#787878</h1>`
4. `<h1 style="background-color:#b4b4b4;">#b4b4b4</h1>`
5. `<h1 style="background-color:#f0f0f0;">#f0f0f0</h1>`
6. `<h1 style="background-color:#ffffff;">#ffffff</h1>`

## خودتان امتحان کنید <<

HSL مقدار

در HTML، رنگ را می توان با استفاده از رنگ، اشباع و سبک (HSL) در فرم مشخص کرد:

**hsl(hue, saturation, lightness )**

- **hue**: بصورت درجه ای بر روی چرخ رنگ ها، از ۰ تا ۳۶۰ است. ۰ قرمز، ۱۲۰ رنگ سبز و ۲۴۰ آبی است.
- **saturation**: یک درصد است، ۰٪ به معنای یک سایه خاکستری است و ۱۰۰٪ رنگ کامل است.
- **lightness**: همچنین درصد، ۰٪ سیاه است، ۵۰٪ نه خیلی سیاه یا سفید است، ۱۰۰٪ سفید است.

مثال : نوع HSL

1. `<h1 style="background-color:hsl(0, 100%, 50%);">hsl(0, 100%, 50%)</h1>`
2. `<h1 style="background-color:hsl(240, 100%, 50%);">hsl(240, 100%, 50%)</h1>`
3. `<h1 style="background-color:hsl(147, 50%, 47%);">hsl(147, 50%, 47%)</h1>`
4. `<h1 style="background-color:hsl(300, 76%, 72%);">hsl(300, 76%, 72%)</h1>`
5. `<h1 style="background-color:hsl(39, 100%, 50%);">hsl(39, 100%, 50%)</h1>`
6. `<h1 style="background-color:hsl(248, 53%, 58%);">hsl(248, 53%, 58%)</h1>`

## خودتان امتحان کنید <<

**Saturation**

Saturation در واقع به معنی شدت و کمیت یک رنگ است.

۱۰۰٪ رنگ خالص است، بدون سایه خاکستری.

۵۰٪ خاکستری است، اما شما هنوز هم می توانید رنگ را ببینید.

۰٪ به طور کامل خاکستری است، شما دیگر نمی توانید رنگ را ببینید.

## مثال : شدت رنگ ها

1. `<h1 style="background-color:hsl(0, 100%, 50%);">hsl(0, 100%, 50%)</h1>`
2. `<h1 style="background-color:hsl(0, 80%, 50%);">hsl(0, 80%, 50%)</h1>`
3. `<h1 style="background-color:hsl(0, 60%, 50%);">hsl(0, 60%, 50%)</h1>`
4. `<h1 style="background-color:hsl(0, 40%, 50%);">hsl(0, 40%, 50%)</h1>`
5. `<h1 style="background-color:hsl(0, 20%, 50%);">hsl(0, 20%, 50%)</h1>`
6. `<h1 style="background-color:hsl(0, 0%, 50%);">hsl(0, 0%, 50%)</h1>`

## [خودتان امتحان کنید<<](#)

### روشنایی (Lightness)

Lightness یک رنگ را می توان به این صورت توضیح داد: مقدار روشنایی رنگ، که % به معنای هیچ نوری (سیاه و سفید)، %۵۰ به معنای %۵۰ نور (نه تاریک و نه نور) %۱۰۰ به معنی نور کامل (سفید) توصیف می شود.

## مثال : روشنایی رنگ ها

1. `<h1 style="background-color:hsl(0, 100%, 0%);">hsl(0, 100%, 0%)</h1>`
2. `<h1 style="background-color:hsl(0, 100%, 25%);">hsl(0, 100%, 25%)</h1>`
3. `<h1 style="background-color:hsl(0, 100%, 50%);">hsl(0, 100%, 50%)</h1>`
4. `<h1 style="background-color:hsl(0, 100%, 75%);">hsl(0, 100%, 75%)</h1>`
5. `<h1 style="background-color:hsl(0, 100%, 90%);">hsl(0, 100%, 90%)</h1>`
6. `<h1 style="background-color:hsl(0, 100%, 100%);">hsl(0, 100%, 100%)</h1>`

## [خودتان امتحان کنید<<](#)

در رنگ های خاکستری:

## مثال : روشنایی رنگ های خاکستری

1. `<h1 style="background-color:hsl(0, 0%, 0%);">hsl(0, 0%, 0%)</h1>`
2. `<h1 style="background-color:hsl(0, 0%, 24%);">hsl(0, 0%, 24%)</h1>`
3. `<h1 style="background-color:hsl(0, 0%, 47%);">hsl(0, 0%, 47%)</h1>`
4. `<h1 style="background-color:hsl(0, 0%, 71%);">hsl(0, 0%, 71%)</h1>`
5. `<h1 style="background-color:hsl(0, 0%, 94%);">hsl(0, 0%, 94%)</h1>`
6. `<h1 style="background-color:hsl(0, 0%, 100%);">hsl(0, 0%, 100%)</h1>`

## [خودتان امتحان کنید<<](#)

### مقدار RGBA

مقادیر رنگ RGBA یک فرمت ارزش رنگی RGB با یک کانال آلفا است که کدورت را برای یک رنگ مشخص می کند.

مقدار رنگ RGBA با فرمت زیر تعیین می شود:

#### `rgba(red, green, blue, alpha)`

پارامتر alpha در فرمت فوق می تواند مقداری بین ۰/۰ (بیشترین شفافیت) و ۱/۰ (بیشترین میزان شفافیت) را داشته باشد.

1. `<h1 style="background-color:rgba(255, 99, 71, 0);">rgba(255, 99, 71, 0)</h1>`
2. `<h1 style="background-color:rgba(255, 99, 71, 0.2);">rgba(255, 99, 71, 0.2)</h1>`
3. `<h1 style="background-color:rgba(255, 99, 71, 0.4);">rgba(255, 99, 71, 0.4)</h1>`
4. `<h1 style="background-color:rgba(255, 99, 71, 0.6);">rgba(255, 99, 71, 0.6)</h1>`
5. `<h1 style="background-color:rgba(255, 99, 71, 0.8);">rgba(255, 99, 71, 0.8)</h1>`
6. `<h1 style="background-color:rgba(255, 99, 71, 1);">rgba(255, 99, 71, 1)</h1>`

[خودتان امتحان کنید<<](#)

## HSLA مقدار

مقدار HSLA شامل مقادیر نوع HSL به همراه یک پارامتر alpha که میزان شفافیت رنگ را مشخص می کند همراه است.

برای استفاده از مقادیر HSLA باید از فرمت زیر استفاده کرد :

### hsla(*hue, saturation, lightness, alpha*)

پارامتر alpha در فرمت فوق می تواند مقداری بین ۰٪ و ۱۰۰٪ (بیشترین میزان شفافیت) و ۱/۰ (بیشترین میزان شفافیت) را داشته باشد.

1. `<h1 style="background-color:hsla(9, 100%, 64%, 0);">hsla(9, 100%, 64%, 0)</h1>`
2. `<h1 style="background-color:hsla(9, 100%, 64%, 0.2);">hsla(9, 100%, 64%, 0.2)</h1>`
3. `<h1 style="background-color:hsla(9, 100%, 64%, 0.4);">hsla(9, 100%, 64%, 0.4)</h1>`
4. `<h1 style="background-color:hsla(9, 100%, 64%, 0.6);">hsla(9, 100%, 64%, 0.6)</h1>`
5. `<h1 style="background-color:hsla(9, 100%, 64%, 0.8);">hsla(9, 100%, 64%, 0.8)</h1>`
6. `<h1 style="background-color:hsla(9, 100%, 64%, 1);">hsla(9, 100%, 64%, 1)</h1>`

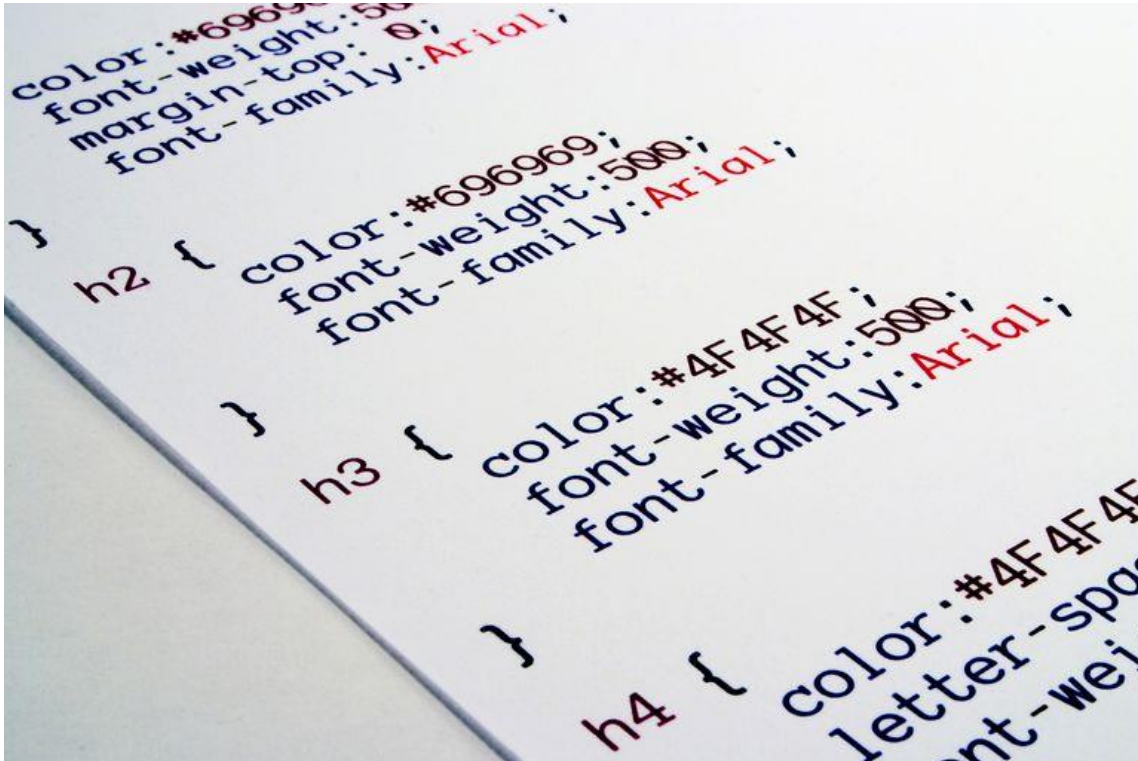
[خودتان امتحان کنید<<](#)

## کلام آخر

رنگ بندی تگ ها و بخش های یک وب سایت از مباحثی است که در جذابیت و زیبایی صفحات وب نقش بسیار مهمی دارد. و به نوعی باعث جذب کاربران بیشتر به وب سایت می شود. از این در این بخش به بررسی فرمت های مختلف **رنگ ها در HTML** پرداختیم.

## مقدمه

با عرض سلام خدمت کاربران سایت [پی وی لرن](#) . در این پست آموزشی قصد داریم به آموزش چگونگی استفاده از کدهای CSS در [html](#) بپردازیم. CSS در واقع افزونه ای برای دادن Style و زیبایی به ظاهر عناصر html بکار رفته در صفحات وب مورد استفاده قرار می گیرد. با استفاده از CSS می توان افکت ها و نماهای زیبا و کاربر پسند به عناصر html اعمال کرد. که در جذب کاربران به وب سایت، نقش بسیار مهمی را ایفا می کند. در این پست آموزشی چگونگی استفاده از کدهای CSS در html را مورد بررسی قرار خواهیم داد.



CSS- کدهای css در html

## css در html

شایع ترین روش اضافه کردن CSS این است که **Style** ها را در فایل های CSS جداگانه نگه دارید. با این حال، در اینجا ما از یک **css** درون خطی در تگ ها و **css** داخلی **html** استفاده خواهیم کرد، برای شما هم راحت تر است که آن ها را امتحان کنید.

## CSS درون خطی

با استفاده از CSS درون خطی شما می توانید به هر کدام از عناصر **html** در همان تگ ، یک **style** بدهید.

مثال:

مثال : درون خطی

1. `<h1 style="color:blue;">This is a Blue Heading</h1>`

## css داخلی html

با استفاده از این روش شما می توانید یک style را به یک صفحه وب html بدهید. این style در قسمت **<head>** صفحه وب html و در داخل تگ **<style>** قرار می گیرد

مثال:

مثال : استایل داخلی

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. body {background-color: powderblue;}
6. h1 {color: blue;}
7. p {color: red;}
8. </style>
9. </head>
10. <body>
11.
12. <h1>This is a heading</h1>
13. <p>This is a paragraph.</p>
14.
15. </body>
16. </html>
```

## [خودتان امتحان کنید](#)

## css خارجی html

با استفاده از این روش می توان یک style را برای چندین صفحه وب html استفاده کرد. در این روش style تعریف شده در داخل یک فایل جداگانه با فرمت **css** قرار گرفته و برای استفاده در بخش **head** فایل html فراخوانی می شود.

مثال:

مثال : استایل خارجی

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <link rel="stylesheet" href="styles.css">
5. </head>
6. <body>
7.
8. <h1>This is a heading</h1>
9. <p>This is a paragraph.</p>
10.
11. </body>
12. </html>
```



در CSS خارجی ابتدا style ها در داخل فایل متنی تایپ شده و سپس در قالب فایل با پسوند CSS مثلا " styles.css ذخیره می شود. این style ها در داخل فایل CSS بصورت نمونه ی زیر باید تایپ شود:

مثال :

```
1. body {
2.     background-color: powderblue;
3. }
4. h1 {
5.     color: blue;
6. }
7. p {
8.     color: red; }
```

## فونت های CSS

خصوصیت color در CSS رنگ متن را تعریف می کند، خصوصیت font-family در CSS نوع فونت را تعریف می کند.

خصوصیت font-size در CSS اندازه ی متن را مشخص می کند.

مثال : استایل متن در CSS

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. h1 {
6.     color: blue;
7.     font-family: verdana;
8.     font-size: 300%;
9. }
10. p {
11.     color: red;
12.     font-family: courier;
13.     font-size: 160%;
14. }
15. </style>
16. </head>
17. <body>
18.
19. <h1>This is a heading</h1>
20. <p>This is a paragraph.</p>
21.
22. </body>
23. </html>
```

## border در CSS

از خصوصیت border در CSS برای استایل border اطراف عناصر HTML استفاده می شود :

مثال : استایل border

```
1. p {  
2.   border: 1px solid powderblue;  
3. }
```

[خودتان امتحان کنید»»](#)

## padding در CSS

از خصوصیت padding در CSS برای مشخص کردن فضای خالی داخلی عناصر HTML استفاده می شود :

مثال : فضای داخلی عناصر HTML

```
1. p {  
2.   border: 1px solid powderblue;  
3.   padding: 30px;  
4. }
```

[خودتان امتحان کنید»»](#)

## margin در خصوصیت

از خصوصیت margin در CSS برای تعیین فاصله یک عنصر HTML از سایر عناصر HTML استفاده می شود:

مثال : ویژگی margin

```
1. p {  
2.   border: 1px solid powderblue;  
3.   margin: 50px;  
4. }
```

[خودتان امتحان کنید»»](#)

## id صفت

برای تعریف style خاص روی یکی از عناصر می توانید از صفت id برای این منظور استفاده کرد.

1. `<p id="p01">I am different</p>`

سپس style مورد نظر خود را برای id فوق به صورت زیر تعریف کنید :

مثال : تعریف استایل id

```
1. #p01 {
2.     color: blue;
3. }
```

### [خودتان امتحان کنید<<](#)

نکته : id یک عنصر باید در یک صفحه html منحصر به فرد باشد، بنابراین انتخاب id برای انتخاب یک عنصر منحصر به فرد استفاده می شود!

### صفت class

برای تعریف یک style خاص برای گروهی از عناصر html می توانید از صفت class در عناصر html استفاده کنید :

مثال :

1. `<p class="error">I am different</p>`

سپس style مورد نظر خود را برای class فوق به صورت زیر تعریف کنید :

مثال : استایل class

```
1. p.error {
2.     color: red;
3. }
```

### [خودتان امتحان کنید<<](#)

### مراجع خارجی

شما می توانید style های عناصر html را در یک فایل جداگانه با فرمت css تعریف کرده و سپس با تگ مخصوص (تگ <link>) در صفحه ی html مربوطه به فایل CSS آدرس دهی کنید:

مثال : استایل خارجی

1. `<link rel="stylesheet" href="https://www.pvlearn.com/html/styles.css">`

## خودتان امتحان کنید<>

این مثال به یک صفحه style در دایرکتوری HTML در وب سایت فعلی پیوند دارد:

مثال : مثال دوم

1. `<link rel="stylesheet" href="/html/styles.css">`

## خودتان امتحان کنید<>

این مثال به یک صفحه style در یک پوشه مشابه صفحه فعلی پیوند دارد:

مثال : لینک به style جاری

1. `<link rel="stylesheet" href="styles.css">`

## خودتان امتحان کنید<>

### خلاصه ی فصل

از ویژگی style در HTML برای سبک درونی استفاده کنید.  
برای تعیین CSS داخلی از عنصر `<style>` استفاده کنید.  
از عنصر `<link>` برای اشاره به یک فایل CSS خارجی استفاده کنید.  
از عنصر `<head>` برای ذخیره عناصر `<style>` و `<link>` استفاده کنید.  
از ویژگی رنگی CSS برای رنگ متن استفاده کنید.  
از فونت CSS فونت برای فونت متن استفاده کنید.  
از ویژگی `font-size` برای اندازه متن استفاده کنید.  
از مرجعیت CSS استفاده کنید.  
از ویژگی `CSS padding` برای فضای داخل مرز استفاده کنید.  
از اموال حاشیه `CSS` برای فضای خارج از مرز استفاده کنید.

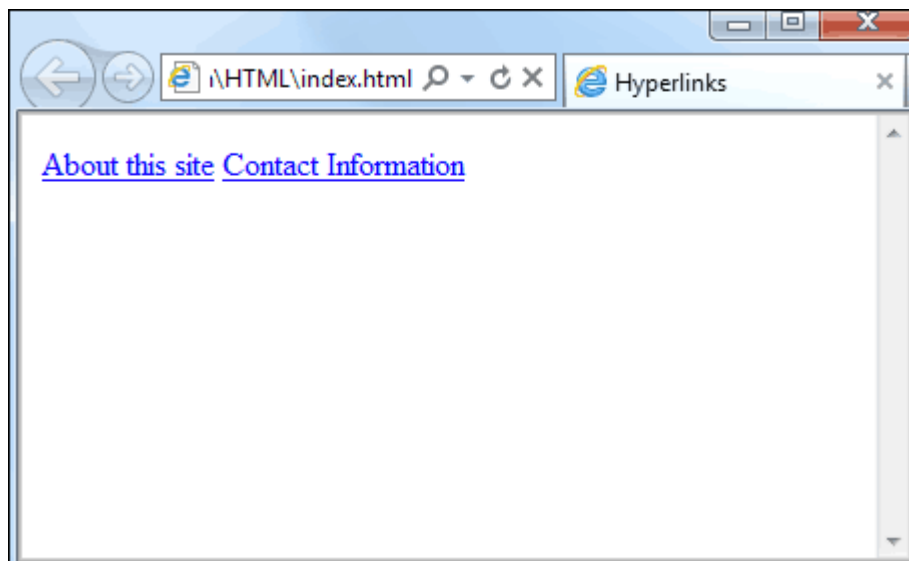
### کلام آخر

تگ های html برای ایجاد ساختار و اسکلت سایت استفاده می شوند و در حالت معمولی هیچگونه شکل و شمایل خاصی ندارند، اما با استفاده از **CSS در HTML** می توان رنگ و style مورد نظر را برای عناصر مختلف html تعریف کرد.

### مقدمه

با عرض سلام خدمت کاربران سایت **پی وی لرن** . در سایر بخش های آموزشهای html ، به کاربرد **تگ های Citation و Quotation** در **html** ، اشاره ای داشتیم. و توضیحاتی در مورد آن ارائه گردید. در این بخش از آموزش، قصد داریم به آموزش **استفاده از لینک ها در html** ، بپردازیم. لینک ها تقریباً، در تمام صفحات وب پیدا می شوند. لینک ها ، به شما اجازه می دهد، تا کاربران را از صفحه ای به صفحه ای دیگر در وب هدایت کنید. و باعث می شود که هر روزه تمام مطالب وب سایت شما مورد بازدید قرار بگیرد.

از این رو یادگیری درست استفاده از **لینک ها** در صفحات وب سایت، از اهمیت ویژه ای برخوردار است. در این پست آموزشی ، ما شما را با ساختار کلی لینک ها و استفاده از آن در صفحات وب آشنا می کنیم.



links

### استفاده از لینک ها در html

لینک ها در html، ابر پیوند ها هستند. شما می توانید فقط با کلیک روی یک لینک از صفحه جاری به صفحه ای دیگر منتقل شوید. زمانیکه شما اشاره گر ماوس را روی یک لینک می برید ، اشاره گر تبدیل به به دست کوچک می شود.

### ساختار لینک در html

در html ، لینک ها با استفاده از تگ **<a>** تعریف می شوند:

مثال :

```
1. <a href="url">link text</a>
```

در html ، لینک ها با استفاده از تگ **<a>** تعریف می شوند:

1. `<a href="https://www.pvlearn.com/html/">Visit our HTML tutorial</a>`

### [خودتان امتحان کنید<<](#)

صفت **href** ، در تگ `<a>` ، آدرس لینک را مشخص می کند.

## لینک محلی

لینک های محلی برای لینک به صفحات وب سایتی که طراحی می کنید، یا طراحی کرده اید. استفاده می شود.

### مثال : لینک های محلی

1. `<a href="html_images.asp">HTML Images</a>`

### [خودتان امتحان کنید<<](#)

## معرفی رنگ لینک ها

می توانید با استفاده از کدهای CSS ، برای وضعیت های مختلف لینک ها (ویزیت شده ، ویزیت نشده و ...) رنگ مشخصی تعریف کنید.

### مثال : رنگ لینک

```

1. <style>
2. a:link {
3.     color: green;
4.     background-color: transparent;
5.     text-decoration: none;
6. }
7.
8. a:visited {
9.     color: pink;
10.    background-color: transparent;
11.    text-decoration: none;
12. }
13.
14. a:hover {
15.     color: red;
16.     background-color: transparent;
17.     text-decoration: underline;
18. }
19.
20. a:active {
21.     color: yellow;
22.     background-color: transparent;
23.     text-decoration: underline;
24. }
25. </style>
    
```

### [خودتان امتحان کنید<<](#)

## صفت target

این صفت تعیین می کند که با کلیک بر روی لینک ، صفحه ای که با آن لینک می شود ، در کجا باز شود. که این صفت مقادیر زیر را می پذیرد:

- **blank\_** : لینک را در صفحه یا تب جدیدی باز می کند.
- **self\_** : لینک را در همان پنجره باز می کند. (پیش فرض)
- **parent\_** : لینک را در همان پنجره در داخل یک قاب کوچک از پنجره باز می کند.
- **top\_** : لینک را در قسمت کامل پنجره ، باز می کند.
- **frameName** : لینک را در یک قاب نامگذاری شده باز می کند.

مثال : صفت target

1. `<a href="https://www.pvlearn.com/" target="_blank">Visit pvlearn!</a>`

[خودتان امتحان کنید<<](#)

نکته: اگر صفحه وب شما در یک frame باشد، می توانید از `target = "_top"` برای خارج شدن از frame استفاده کنید:

مثال : مقدار top

1. `<a href="https://www.w3schools.com/html/" target="_top">HTML5 tutorial!</a>`

[خودتان امتحان کنید<<](#)

## استفاده از تصاویر برای لینک

از تصاویر نیز می توان برای لینک دادن به سایر صفحات استفاده کرد.

مثال : افزودن لینک به تصویر

```
1. <a href="default.asp">
2.   
3. </a>
```

[خودتان امتحان کنید<<](#)

نکته: در کد فوق `border:0;` برای جلوگیری از نمایش border در اطراف تصویر در مرورگرهای (IE9 و قدیمی تر) اضافه شده است.

## لینک کردن title

صفت title ، اطلاعاتی اضافی را در مورد یک عنصر نگهداری می کند، و در واقع یک را توصیف می کند. زمانیکه شما اشاره گر ماوس را روی یک عنصر می برید متن کوچکی در توصیف عنصر مورد نظر ظاهر می شود.

1. `<a href="https://www.w3schools.com/html/" title="Go to W3Schools HTML section">Visit our HTML Tutorial</a>`

## ایجاد bookmark

bookmark ها در HTML اجازه ی پرش از قسمتی از صفحه وب به قسمت مشخص شده را فراهم می کنند.

bookmark برای صفحات وب طولانی و بلند کاربرد بیشتری دارند.

برای ایجاد bookmark ابتدا باید آن را ایجاد کرده و سپس به آن لینک دهید.

سپس زمانیکه روی لینک کلیک شود، صفحه به محلی که با لینک مشخص شده اسکرول می شود.

مثال- ابتدا با استفاده از id یک bookmark برای عنصر مورد نظر ایجاد کنید.

1. `<h2 id="C4">Chapter 4</h2>`

سپس از محل مورد نظر یک لینک به bookmark ایجاد شده ایجاد می کنیم.

1. `<a href="#C4">Jump to Chapter 4</a>`

همچنین می توانید از صفحات دیگر به bookmark به صورت زیر لینک دهید:

1. `<a href="html_demo.html#C4">Jump to Chapter 4</a>`

## خودتان امتحان کنید<<

## آدرس های خارجی

صفحات خارجی را می توان با یک URL کامل یا با مسیری نسبت به صفحه وب فعلی ارجاع داد.

برای مثال لینک به یک آدرس URL کامل به صورت زیر است :

1. `<a href="http://pvlearn.com/product/full-html-and-html5-tutorial/">HTML tutorial</a>`



لینک زیر به فولدر product وب سایت فعلی ارجاع داده شده است :

مثال : لینک به فولدر product

1. `<a href="http://pvlearn.com/product/full-html-and-html5-tutorial/">HTML tutorial</a>`

## خودتان امتحان کنید

لینک زیر به صفحه وبی در مسیر جاری ارجاع دارد:

مثال :

1. `<a href="default.asp">HTML tutorial</a>`

## خلاصه ی فصل

برای تعیین لینک یک عنصر `<a>` استفاده کنید.  
برای تعیین آدرس لینک، از ویژگی `href` استفاده کنید.  
از مشخصه `target` برای تعریف کردن کجا برای باز کردن سند پیوند استفاده کنید.  
از عنصر `<img>` از یک تصویر به عنوان یک لینک استفاده کنید.  
از مشخصه `id` برای تعریف بوک مارک ها در یک صفحه استفاده کنید.  
از ویژگی `href` برای پیوند به نشانه استفاده کنید.

## کلام آخر

یک وب سایت از صفحات وب مختلفی تشکیل شده است که گاهی شما نیاز دارید از بخش های مختلف صفحات وب به یکدیگر لینک بدهید، همچنین ممکن است به لینک به سایر وب سایت ها نیاز باشد، که این اقدامات با استفاده از **لینک ها در html** انجام می شود.

## جلسه ۱۵ : تگ img در کد HTML

### مقدمه

با عرض سلام خدمت کاربران سایت [پی وی لرن](#) . در این پست قصد داریم به آموزش نحوه تعریف و استفاده از تگ **img** در کد **HTML** ، بپردازیم. مهمترین بخش محتوای هر سایت، که در زیبایی و جذابیت سایت بسیار موثر است. تصاویر و عکس های هر سایت می باشد. سایتی که فاقد هر گونه عکس باشد خسته کننده است و کاربران را فراری می دهد. تصاویر می توانند طراحی و ظاهر یک [صفحه وب](#) را بهبود بخشد. که در اسناد [html](#) برای قرار دادن تصاویر در صفحه وب ، تگ **img** ارائه شده است. در پست های قبل، ما نحوه استفاده از [تگ Table](#) و [تگ Div](#) را در **html** توضیح دادیم. و در ادامه این پست، به بررسی تگ **img** ، می پردازیم.



### تگ img در کد HTML

مثال : نمونه تصویر-1

```
1. 
```

[خودتان امتحان کنید<<](#)

مثال : نمونه تصویر-2

```
1. 
```

[خودتان امتحان کنید<<](#)

مثال : نمونه تصویر-3

```
1. 
```

[خودتان امتحان کنید<<](#)

## سینتکس img

در HTML، تصاویر با تگ <img> تعریف می شوند، تگ <img> خالی است، شامل فقط صفات است و تگ بسته ندارد. ویژگی **src** نشانی اینترنتی (آدرس وب) تصویر را مشخص می کند:

مثال :

1. ``

### ویژگی alt:

ویژگی alt یک متن جایگزین برای یک تصویر را فراهم می کند، اگر کاربر به دلایلی نتواند آن تصویر را مشاهده کند. متن جایگزین نمایش داده می شود. مقداری که برای ویژگی alt تعریف می شود، باید تصویر را توصیف کند:

مثال : ویژگی alt

1. ``

### [خودتان امتحان کنید<<](#)

زمانیکه به هر دلیلی مرورگر قادر به نمایش تصویر نباشد، ویژگی alt نمایش داده خواهد شد :

مثال : ویژگی alt

1. ``

### [خودتان امتحان کنید<<](#)

### تنظیم اندازه تصویر:

شما می توانید از ویژگی **Style** برای مشخص کردن عرض (**Width**) و ارتفاع (**Height**) یک تصویر استفاده کنید:

مثال:

مثال : اندازه ی تصویر با style

1. ``

### [خودتان امتحان کنید<<](#)

همچنین می توانید بطور مستقیم از **width** و **height** استفاده کنید (خصوصیات width و ارتفاع همیشه عرض و height تصویر را در پیکسل تعریف می کند):

1. ``

### [خودتان امتحان کنید <<](#)

**نکته :** همیشه عرض و ارتفاع یک تصویر را مشخص کنید. چرا که اگر عرض و ارتفاع مشخص نشده باشند، زمانیکه که تصویر در صفحه وب، بارگیری می شود، صفحه وب ممکن است سوسو زده شود.

## استفاده از Style یا width و height برای اندازه تصاویر؟

هم **Style** و هم **width** و **height** در **HTML 0.5** پشتیبانی می شود. اما ما به شما توصیه می کنیم که از **style** برای تعریف طول و عرض تصویر استفاده کنید. چرا که از تغییرات ناخواسته در اندازه تصاویر جلوگیری می کند.

مثالی برای چگونگی استفاده از **Style** و ویژگی **width** و **height** در **html** :

### مثال : روش های تنظیم عرض و ارتفاع

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. img {
6.     width: 100%;
7. }
8. </style>
9. </head>
10. <body>
11.
12. 
13. 
14.
15. </body>
16. </html>
```

### [خودتان امتحان کنید <<](#)

## فراخوانی تصاویر از سایر فولدر های سیستم

اگر تصویر مورد نظر در فولدر دیگری قرار گرفته باشد و آنرا مشخص نکنید ، مرورگر انتظار دارد که تصویر را در همان پوشه محل قرار گیری فایل HTML، پیدا کند. با این حال، معمول است که تصاویر را در زیر پوشه ها ذخیره کنید. سپس باید نام پوشه را در ویژگی **src** وارد کنید:

### مثال : تصویر در سایر فولدرها

1. ``

### [خودتان امتحان کنید <<](#)

## تصویر در سایر سرورها

برخی از وب سایت ها تصاویر خود را در سرورهای image ذخیره می کنند. در واقع، شما می توانید به تصاویر از هر آدرس وب در جهان دسترسی داشته باشید:

مثال : دسترسی به تصاویر از سایر سرورها

```
1. 
```

[خودتان امتحان کنید<<](#)

## استفاده از تصویر به عنوان لینک

برای استفاده از یک تصویر به عنوان یک لینک، تگ <img> را در داخل برچسب <a> قرار دهید.

مثال :

```
1. <a href="default.asp">  
2.     
3. </a>
```

## انیمیشن تصاویر

HTML اجازه ی تصاویر gif را می دهد :

مثال : تصاویر gif

```
1. 
```

[خودتان امتحان کنید<<](#)

## استفاده از تصویر به عنوان لینک

برای تنظیم تصویر به عنوان لینک کافیست تگ <img> را داخل یک تگ <a> قرار دهید:

مثال : تصویر به عنوان لینک

```
1. <a href="default.asp">  
2.     
3. </a>
```

[خودتان امتحان کنید<<](#)

## خصوصیت float برای تصویر

با استفاده از خصوصیت float می توانید تصاویر مورد نظر را راست چین و یا چپ چین کنید:

مثال : خصوصیت float

1. `<p>`
2. The image will **float** to the right of the text.</p>
- 3.
4. `<p>`
5. The image will **float** to the left of the text.</p>

[خودتان امتحان کنید<<](#)

## نقشه های تصویر

با استفاده از تگ <map> می توانید یک image-map تعریف کنید که با کلیک روی بخش های مختلف تصویر به لینک های جداگانه ارجاع داده شوید:

مثال : تقسیم بندی

1. ``
- 2.
3. `<map name="workmap">`
4. `<area shape="rect" coords="34,44,270,350" alt="Computer" href="computer.htm">`
5. `<area shape="rect" coords="290,172,333,250" alt="Phone" href="phone.htm">`
6. `<area shape="circle" coords="337,300,44" alt="Coffee" href="coffee.htm">`
7. `</map>`

[خودتان امتحان کنید<<](#)

## تنظیم تصویر به عنوان پس زمینه صفحه وب

برای اضافه کردن تصویر پس زمینه بر روی یک عنصر HTML، از ویژگی `background-image` در **CSS** استفاده کنید. برای افزودن یک تصویر پس زمینه در یک صفحه وب، باید ویژگی تصویر پس زمینه را در عنصر **BODY** مشخص کنید.

مثال : پس زمینه

1. `<body style="background-image:url('clouds.jpg')">`
- 2.
3. `<h2>Background Image</h2>`
- 4.
5. `</body>`

[خودتان امتحان کنید<<](#)

برای اضافه کردن یک تصویر پس زمینه بر روی یک پاراگراف، ویژگی پس زمینه را در عنصر P مشخص کنید :

مثال : پس زمینه ی پاراگراف

```
1. <body>
2.
3. <p style="background-image:url('clouds.jpg')">
4. ...
5. </p>
6.
7. </body>
```

[خودتان امتحان کنید<<](#)

## عنصر <picture>

عنصر <picture> در html5 از انعطاف پذیری بیشتری نسبت به <img> برخوردار است.

عنصر <picture> حاوی تعدادی از عناصر <source> است که هر کدام به منابع تصویری اشاره دارند. به این ترتیب مرورگر می تواند تصویری را انتخاب کند که به بهترین شکل متناسب با نمای فعلی و / یا دستگاه باشد.

هر عنصر <source> دارای صفاتی است که توصیف می کند هنگامی که تصویر آنها مناسب تر است.

مرورگر از اولین عنصر <source> با مقادیر تطبیق تطابق استفاده می کند و هر عنصر <source> زیر را نادیده می گیرد.

مثال : عنصر picture

```
1. <picture>
2.   <source media="(min-width: 650px)" srcset="img_pink_flowers.jpg">
3.   <source media="(min-width: 465px)" srcset="img_white_flower.jpg">
4.   
5. </picture>
```

[خودتان امتحان کنید<<](#)

**نکته :** همیشه یک عنصر <img> را بعنوان آخرین عنصر کودک عنصر <picture> تعیین کنید.

عنصر <img> توسط مرورگرهایی که از عنصر <picture> پشتیبانی نمی کند یا اگر هیچ کدام از تگ <source> مطابقت ندارند استفاده می شود.

## screen reader در html

screen reader یک برنامه نرم افزاری است که کد HTML را می خواند، متن را تبدیل می کند و به کاربر اجازه می دهد تا به محتوای آن گوش دهد. خوانندگان صفحه نمایش برای افرادی که دچار اختلال دید و یا معلولیت هستند مفید است.

## خلاصه ی فصل

از المان `<img>` برای تعریف تصویر استفاده کنید.  
از ویژگی `src` برای تعریف URL تصویر استفاده کنید.  
برای تعیین یک متن متناوب برای یک تصویر از ویژگی `alt` HTML استفاده کنید، اگر نمیتوان آن را نمایش داد.  
برای تعیین اندازه تصویر از ابعاد عرض و ارتفاع HTML استفاده کنید.  
با استفاده از خواص عرض و ارتفاع CSS برای تعریف اندازه تصویر (به طور متناوب).  
استفاده از ویژگی `float` برای چیدن تصاویر در سمت راست یا چپ.  
از المان `<map>` برای تعریف نقشه تصویر استفاده کنید.  
از قسمت `<area>` برای تعریف مناطق قابل کلیک در نقشه تصویر استفاده کنید.  
از عنصر `usemap` عنصر `<img>` برای اشاره به یک نقشه تصویر استفاده کنید.  
برای نمایش تصاویر مختلف برای دستگاه های مختلف از عنصر `<picture>` استفاده کنید.

**نکته :** بارگیری تصاویر طول می کشد. تصاویر بزرگ می تواند صفحه شما را کند کند. در انتخاب تصاویر دقت کنید.

## کلام آخر

تصاویر یکی از اصلی ترین بخش ها و محتوای صفحات وب را تشکیل می دهند، و تقریباً در تمام وب سایت های دنیا کم و بیش از تصاویر مختلف استفاده می شود، از این رو انتخاب ، درج و تنظیم تصاویر در وب سایت از اهمیت بالایی برخوردار بوده که در این مبحث در قالب آشنایی با **تگ img در کد HTML** به آن پرداختیم.



### مقدمه

با عرض سلام خدمت کاربران سایت [پی وی لرن](#) . در این پست قصد داریم به آموزش نحوه کار با تگ **table** در کد [html](#) بپردازیم. جداول (Table) از مهمترین و کاربردی ترین تگ های html می باشد که در تمام [صفحات وب](#) جهان استفاده می شود. از مهمترین کاربرد های Table مرتب کردن و نظم دادن به سایر تگ ها ی یک صفحه وب می باشد. در این پست چگونگی ایجاد و کار با tabel ها و کاربرد آن در صفحات وب ، مورد بررسی قرار گرفته است.

Service provider	Data GB	Monthly charge \$	Note
Kimlink	25	75	
Lightwire	200	109	a
Now	20	70	
Orcon	30	75	
Primowireless	30	79	
Snapv	50	75	
Telecom	50	95	
Ubergroup	10	65	
Worldnet	25	70	

Table تگ table –

### تعریف کردن Table

جدول HTML با تگ `<table>` تعریف شده است. هر سطر جدول با تگ `<tr>` تعریف شده است. هدر جدول نیز با تگ `<th>` تعریف می شود. به طور پیش فرض، عناوین جدول درشت و متمرکز هستند. سلول های جدول که حاوی داده ها می باشد، نیز با تگ `<td>` تعریف شده است.

#### مثال : تعریف جدول

```
1. <table style="width:100%">
2.   <tr>
3.     <th>Firstname</th>
4.     <th>Lastname</th>
5.     <th>Age</th>
6.   </tr>
7.   <tr>
8.     <td>Jill</td>
9.     <td>Smith</td>
10.    <td>50</td>
11.  </tr>
12.  <tr>
13.    <td>Eve</td>
14.    <td>Jackson</td>
15.    <td>94</td>
16.  </tr>
17. </table>
```

[خودتان امتحان کنید >>](#)

## اضافه کردن Border در تگ table

Border برای مشخص کردن مرز td ها در تگ tabel استفاده می شود. یک Border را می توان در کد CSS تعریف کرد:

مثال:

مثال : افزودن border به جدول

```
1. table, th, td {
2.     border: 1px solid black;
3. }
```

[خودتان امتحان کنید](#)

## تعریف فضای فاصله داخلی تگ td جداول (Cell padding)

برای تنظیم padding یا فاصله داخلی سلول ها در تگ table، از ویژگی CSS property padding استفاده کنید:

مثال:

مثال : حذف فواصل border جداول

```
1. th, td {
2.     padding: 15px;
3. }
```

[خودتان امتحان کنید](#)

## ادغام چند سلول (td) در یک سلول

برای ادغام سلول ها در بیش از یک ستون، از ویژگی colspan استفاده کنید:

مثال : ادغام سلول های جدول

```
1. <table style="width:100%">
2.   <tr>
3.     <th>Name</th>
4.     <th colspan="2">Telephone</th>
5.   </tr>
6.   <tr>
7.     <td>Bill Gates</td>
8.     <td>55577854</td>
9.     <td>55577855</td>
10.  </tr>
11. </table>
```

[خودتان امتحان کنید](#)

## ترازبندی چپ heading

در حالت پیش فرض heading های جداول در وسط تراز شده اند. برای ترازبندی در چپ یا راست می توانید از صفت `text-align` استفاده کنید :

مثال : ترازبندی heading

```
1. th {
2.     text-align: left;
3. }
```

[خودتان امتحان کنید<<](#)

## افزودن Border spacing

ویژگی Border spacing فواصل بین سلول های جداول را تعیین می کند. برای تعیین آن از ویژگی `border-spacing` به صورت زیر استفاده کنید:

مثال : فواصل بین سلول ها

```
1. table {
2.     border-spacing: 5px;
3. }
```

[خودتان امتحان کنید<<](#)

## ادغام ستون های جدول

برای ادغام چند ستون جدول در یکی از صفت `colspan` به صورت زیر استفاده کنید:

مثال : ادغام ستون های جدول

```
1. <table style="width:100%">
2.   <tr>
3.     <th>Name</th>
4.     <th colspan="2">Telephone</th>
5.   </tr>
6.   <tr>
7.     <td>Bill Gates</td>
8.     <td>55577854</td>
9.     <td>55577855</td>
10.  </tr>
11. </table>
```

[خودتان امتحان کنید<<](#)

## ادغام سطرهای جدول

برای ادغام چند سطر از جدول در یکی از صفت `rowspan` به صورت زیر استفاده کنید:

مثال : ادغام سطرهای جدول

```
1. <table style="width:100%">
2.   <tr>
3.     <th>Name:</th>
4.     <td>Bill Gates</td>
5.   </tr>
6.   <tr>
7.     <th rowspan="2">Telephone:</th>
8.     <td>55577854</td>
9.   </tr>
10.  <tr>
11.    <td>55577855</td>
12.  </tr>
13. </table>
```

[خودتان امتحان کنید<<](#)

## افزودن عنوان در جدول

بای افزودن عنوان در یک تگ `table` ، از تگ `<caption>` استفاده کنید:

مثال : افزودن عنوان

```
1. <table style="width:100%">
2.   <caption>Monthly savings</caption>
3.   <tr>
4.     <th>Month</th>
5.     <th>Savings</th>
6.   </tr>
7.   <tr>
8.     <td>January</td>
9.     <td>$100</td>
10.  </tr>
11.  <tr>
12.    <td>February</td>
13.    <td>$50</td>
14.  </tr>
15. </table>
```

[خودتان امتحان کنید<<](#)

## تعریف کردن یک style برای جدول

برای تعریف یک سبک (style) خاص برای یک جدول ، یک ویژگی id را به جدول اضافه کنید:

مثال :

```
1. <table id="t01">
2.   <tr>
3.     <th>Firstname</th>
4.     <th>Lastname</th>
5.     <th>Age</th>
6.   </tr>
7.   <tr>
8.     <td>Eve</td>
9.     <td>Jackson</td>
10.    <td>94</td>
11.  </tr>
12. </table>
```

اکنون در کد CSS یک Style برای تگ table تعریف کنید:

مثال : تعریف style

```
1. table#t01 {
2.   width: 100%;
3.   background-color: #f1f1c1;
4. }
```

[خودتان امتحان کنید <<](#)

می توانید style های بیشتری را نیز اضافه کنید:

مثال : افزودن style بیشتر

```
1. table#t01 tr:nth-child(even) {
2.   background-color: #eee;
3. }
4. table#t01 tr:nth-child(odd) {
5.   background-color: #fff;
6. }
7. table#t01 th {
8.   color: white;
9.   background-color: black;
10. }
```

[خودتان امتحان کنید <<](#)

## خلاصه ی بخش

از المان `<table>` برای تعریف یک جدول استفاده کنید.  
برای تعیین یک ردیف جدول از عنصر `<tr>` استفاده کنید.  
از المان `<td>` برای تعریف یک جدول استفاده کنید.  
از المان `<th>` برای تعریف یک عنوان جدول استفاده کنید.  
از المان `<caption>` برای تعریف یک عنوان جدول استفاده کنید.  
از `border` برای تعریف مرز استفاده کنید.  
از `border-collapse` برای از بین بردن مرزهای سلولی استفاده کنید.  
از ویژگی `padding` برای اضافه کردن لایه به سلول استفاده کنید.  
از چیدمان CSS متن برای خواندن متن سلول استفاده کنید.  
برای تنظیم فاصله بین سلول ها از ویژگی `border-spacing` استفاده کنید.  
استفاده از خصیصه `colspan` برای ایجاد یک سلول در طول بسیاری از ستونها.  
استفاده از `rowspan` برای ایجاد یک سلول، چندین ردیف را ایجاد می کند.  
از ویژگی `id` برای جداگانه یک جدول استفاده کنید.

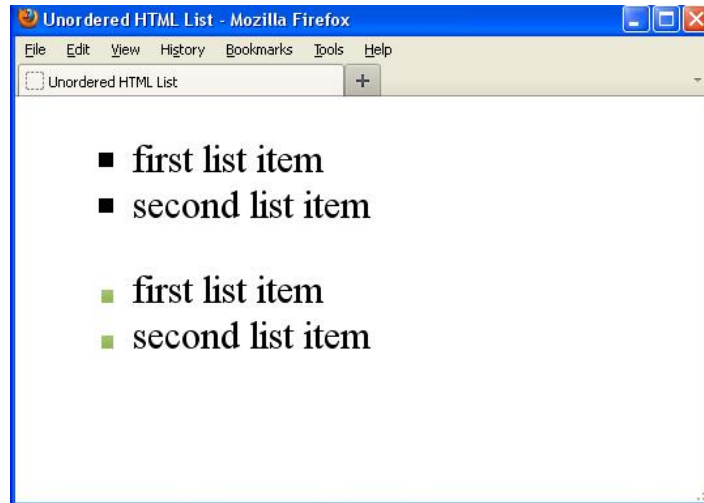
## کلام آخر

**کار با Table در HTML** در گذشته به علت استفاده از آن در مرتب سازی عناصر html اهمیت ویژه ای داشت اما در حال حاضر بیشتر در درج مطالب و محتوای جدولی در صفحات وب از آن استفاده می شود، اما هنوز هم کاربرد فراوانی در وب سایت ها دارد.

## جلسه ۱۷ : تگ های List در html

### مقدمه

با عرض سلام خدمت کاربران سایت [پی وی لرن](#) . در آموزشهایی که تاکنون در مباحث HTML ، ارائه دادیم. به مطالب مهمی از جمله [استفاده از لینک ها در html](#)، اشاره کردیم. در ادامه ی دوره ، در این بخش قصد داریم به آموزش چگونگی کار با تگ های List در html یا لیست ها در html ، بپردازیم. در اکثر وب سایت ها، مخصوصا در محتویاتی نظیر مقالات علمی و آموزشهای کاربردی و ... از لیست ها بسیار استفاده می شود.



کار با تگ های List در html

## کار با تگ های List در html

مثال : لیست ها

```
1. <!DOCTYPE html>
2. <html>
3. <body>
4.
5. <h2>An Unordered HTML List</h2>
6.
7. <ul>
8.   <li>Coffee</li>
9.   <li>Tea</li>
10.  <li>Milk</li>
11. </ul>
12.
13. <h2>An Ordered HTML List</h2>
14.
15. <ol>
16.  <li>Coffee</li>
17.  <li>Tea</li>
18.  <li>Milk</li>
19. </ol>
20.
21. </body>
22. </html>
```

[خودتان امتحان کنید<<](#)

در کد html، تگ های لیست، انواع مختلفی دارد. که در ادامه به بررسی هر کدام می پردازیم.

## لیست های بی نظم در html

لیست بی نظم در html با تگ <ul> شروع می شود و هر کدام از آیتم های آن با تگ <li> شروع می شود. آیتم های این لیست، بطور پیش فرض با bullets علامت گذاری می شود.

مثال : لیست های نامرتب

```
1. <ul>
2.   <li>Coffee</li>
3.   <li>Tea</li>
4.   <li>Milk</li>
5. </ul>
```

### [خودتان امتحان کنید<>](#)

## تغییر دادن علامت آیتم ها در لیست های بی نظم

در اینگونه لیست ها با استفاده از ویژگی **list-style-type** می توانید، نوع علامت آیتم ها را مشخص کنید، مقادیر این ویژگی به شرح زیر است:

- **disc** : علامتدار کردن آیتم ها با bullet (پی ش فرض)
- **circle** : علامتدار کردن آیتم ها با دایره.
- **square** : علامتدار کردن آیتم ها با مربع.
- **none** : بدون علامت.

مثال : تغییر دادن علامت لیست ها

```
1. <ul style="list-style-type:disc">
2.   <li>Coffee</li>
3.   <li>Tea</li>
4.   <li>Milk</li>
5. </ul>
```

### [خودتان امتحان کنید<>](#)

مثال : انتخاب دایره به عنوان علامت

```
1. <ul style="list-style-type:circle">
2.   <li>Coffee</li>
3.   <li>Tea</li>
4.   <li>Milk</li>
5. </ul>
```

### [خودتان امتحان کنید<>](#)



مثال ۳: با استفاده از `square`

مثال : انتخاب مربع به عنوان علامت

```
1. <ul style="list-style-type:square">
2.   <li>Coffee</li>
3.   <li>Tea</li>
4.   <li>Milk</li>
5. </ul>
```

[خودتان امتحان کنید<<](#)

مثال ۴: با استفاده از `none`

مثال : لیست بدون علامت

```
1. <ul style="list-style-type:none">
2.   <li>Coffee</li>
3.   <li>Tea</li>
4.   <li>Milk</li>
5. </ul>
```

[خودتان امتحان کنید<<](#)

## لیست های منظم در html

لیست های منظم در html با تگ `<ol>` شروع می شوند ، هر کدام از آیتم های لیست ها نیز با تگ `<li>` شروع می شود.

مثال : لیست های منظم

```
1. <ol>
2.   <li>Coffee</li>
3.   <li>Tea</li>
4.   <li>Milk</li>
5. </ol>
```

[خودتان امتحان کنید<<](#)

## انواع صفات در لیست های منظم (<ol>)

- `type="1"` : آیتم های لیست ، با اعداد شماره گذاری می شود.
- `type="A"` : آیتم های لیست، با حروف بزرگ علامتگذاری می شود.
- `type="a"` : آیتم های لیست، با حروف کوچک علامتگذاری می شود.
- `type="I"` : آیتم های لیست با اعداد رومی بزرگ شماره می شود.

- **type="i"**: آیتم های لیست با اعداد رومی کوچک شماره می شود.

مثال : شماره گذاری آیتم ها با اعداد:

مثال : صفت type

```
1. <ol type="1">
2.   <li>Coffee</li>
3.   <li>Tea</li>
4.   <li>Milk</li>
5. </ol>
```

[خودتان امتحان کنید<<](#)

مثال ۲: علامتگذاری آیتم های با حروف بزرگ:

مثال : علامت گذاری با حروف بزرگ

```
1. <ol type="A">
2.   <li>Coffee</li>
3.   <li>Tea</li>
4.   <li>Milk</li>
5. </ol>
```

[خودتان امتحان کنید<<](#)

مثال : علامتگذاری با حروف کوچک

```
1. <ol type="a">
2.   <li>Coffee</li>
3.   <li>Tea</li>
4.   <li>Milk</li>
5. </ol>
```

[خودتان امتحان کنید<<](#)

مثال : اعداد رومی بزرگ

```
1. <ol type="I">
2.   <li>Coffee</li>
3.   <li>Tea</li>
4.   <li>Milk</li>
5. </ol>
```

علامتگذاری با اعداد رومی کوچک:

مثال : علامتگذاری با اعداد رومی کوچک

```
1. <ol type="i">
2.   <li>Coffee</li>
3.   <li>Tea</li>
4.   <li>Milk</li>
5. </ol>
```

## خودتان امتحان کنید

### لیست های توصیفی

لیست های توصیفی مجموعه ای از اصطلاحات هستند، که هر کدام از اصطلاحات را توصیف می کنند. تگ <dl> لیست توصیفی را تعریف می کند. تگ <dt> اصطلاحات لیست توصیفی را تعریف می کند. و تگ <dd> برای توصیف همه ی اصطلاحات ، کاربرد دارد.

مثال : لیست های توصیفی

```
1. <dl>
2.   <dt>Coffee</dt>
3.   <dd>- black hot drink</dd>
4.   <dt>Milk</dt>
5.   <dd>- white cold drink</dd>
6. </dl>
```

## خودتان امتحان کنید

### لیست های تودرتو

در html لیست ها، بصورت تودرتو نیز می تواند تعریف شود.

مثال : لیست های تودرتو

```
1. <ul>
2.   <li>Coffee</li>
3.   <li>Tea
4.     <ul>
5.       <li>Black tea</li>
6.       <li>Green tea</li>
7.     </ul>
8.   </li>
9.   <li>Milk</li>
10. </ul>
```

## کنترل شمارشگر لیست

در لیست های منظم ، آیتم ها، از عدد 1 شروع می شوند، شما می توانید با استفاده از صفت start ، عدد شروع شمارش آیتم ها، را تغییر بدهید.

مثال : کنترل شمارشگر لیست

```
1. <ol start="50">
2.   <li>Coffee</li>
3.   <li>Tea</li>
4.   <li>Milk</li>
5. </ol>
```

## لیست های افقی در html

در html ، با استفاده از کدهای CSS (برای آشنایی بیشتر با نحوه استفاده از CSS در HTML می توانید به [آموزشهای کامل CSS](#) ، مراجعه کنید)، می توانید لیست ها را در صفحه وب بصورت افقی نمایش دهید. این ویژگی در موارد بسیاری از جمله ایجاد منو برای وب سایت کاربرد دارد.

مثال : لیست های افقی

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5.   ul {
6.     list-style-type: none;
7.     margin: 0;
8.     padding: 0;
9.     overflow: hidden;
10.    background-color: #333333;
11.  }
12.
13.  li {
14.    float: left;
15.  }
16.
17.  li a {
18.    display: block;
19.    color: white;
20.    text-align: center;
21.    padding: 16px;
22.    text-decoration: none;
```

```

23. }
24.
25. li a:hover {
26.     background-color: #111111;
27. }
28. </style>
29. </head>
30. <body>
31.
32. <ul>
33.   <li><a href="#home">Home</a></li>
34.   <li><a href="#news">News</a></li>
35.   <li><a href="#contact">Contact</a></li>
36.   <li><a href="#about">About</a></li>
37. </ul>
38.
39. </body>
40. </html>

```

## خودتان امتحان کنید

### خلاصه ی بخش

از المان `<ul>` برای تعریف فهرست غیر ارادی استفاده کنید. از ویژگی `style` برای تعیین لیست مارک مورد استفاده استفاده کنید. از المان `<ol>` برای تعریف فهرست دستورالعمل استفاده کنید. از نوع صفت HTML برای تعریف نوع شماره استفاده کنید. از المان `<li>` برای تعریف یک مورد لیست استفاده کنید. از المان `<dl>` برای تعریف یک لیست توضیحات استفاده کنید. برای تعیین اصطلاح توضیح از عنصر `<dt>` استفاده کنید. از عنصر `<dd>` برای توصیف اصطلاح در یک لیست توضیح استفاده کنید. لیست ها را می توان در داخل یکدیگر قرار داد. اقلام لیست می تواند شامل سایر عناصر HTML باشد.

از ویژگی `float:left` یا `display:inline` استفاده کنید: سمت چپ یا صفحه نمایش، درون خطی برای نمایش یک لیست به صورت افقی.

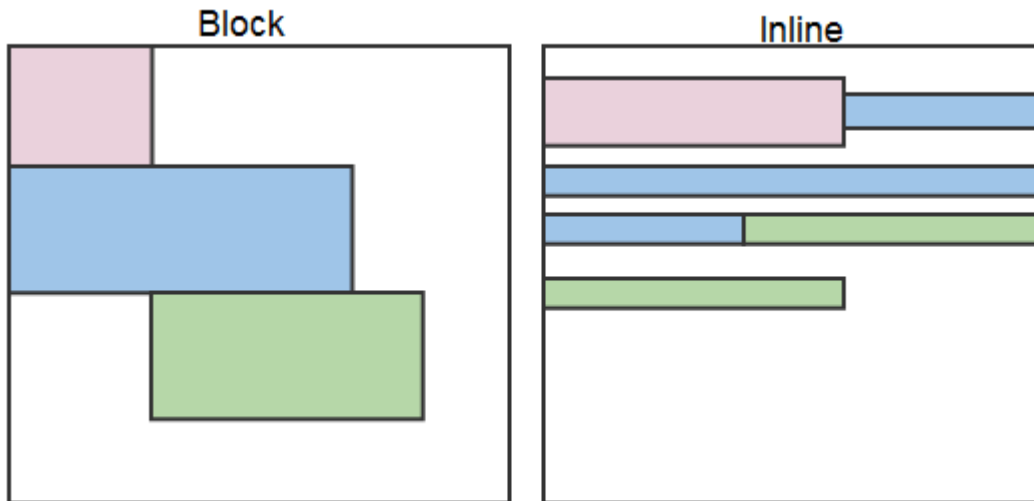
### کلام آخر

در اغلب محتوای وب سایت ها ، خصوصا محتوای علمی و آموزشی از لیست ها به فراوانی استفاده می شود، از این رو آشنایی با طرز کار و تنظیم **تگ های List در html** از مباحث مهم محسوب می شود.

## جلسه ۱۸ : ویژگی های Block و Inline در html

### مقدمه

با عرض سلام خدمت کاربران سایت [پی وی لرن](#) . در آموزشهای مطالب قبلی ، در مورد مباحث [کار با لیست ها](#) و [مدیریت لینک ها](#) در [html](#)، توضیحاتی ارائه شد. در این بخش از آموزشهای html ، قصد داریم به مبحث [آشنایی با ویژگی های Block و Inline در عناصر html](#) ، بپردازیم. هر کدام از عناصر html دارای یک مقدار پیش فرض برای صفت **display** ، می باشد. مقدار پیش فرض **display** ، برای اغلب عناصر html ، معمولا با Block یا Inline ، مقدار دهی می شود.



block

### آشنایی با ویژگی های Block و Inline در عناصر html

ابتدا به بررسی و توضیح ویژگی Block ، در عناصر html می پردازیم:

#### ویژگی Block در عناصر html

ویژگی Block ، در عناصر html ، همیشه باعث می شوند که همیشه عنصر html ، در خط جدیدی شروع شده و بیشترین عرض ( width ) قابل دسترس را اشغال کند. (و تا آنجا که ممکن است سمت چپ و راست را اشغال می کند).

مثال- تگ <div> ، از نمونه تگ های نوع Block ، است:

مثال : سطح block

1. <div>Hello</div>
2. <div>World</div>

[خودتان امتحان کنید>>](#)

تگ هایی که دارای ویژگی پیش فرض **block** هستند:

- <address>
- <article>
- <aside>
- <blockquote>
- <canvas>
- <dd>
- <div>
- <dl>
- <dt>
- <fieldset>
- <figcaption>
- <figure>
- <footer>
- <form>
- <h6>-<h1>
- <header>
- <hr>
- <li>
- <main>
- <nav>
- <noscript>
- <ol>
- <output>
- <p>
- <pre>
- <section>
- <table>
- <tfoot>
- <ul>
- <video>

## ویژگی inline در عناصر html

عناصر html که از نوع inline ، هستند ، در یه خط جدید شروع شده و عرض ( width ) آن ها به اندازه ای که برای محتوای آن ها مورد نیاز است ، تنظیم می شود.

مثال- تگ <span> از نوع تگ هایی با ویژگی پیش فرض inline است:

مثال : عناصر سطح span

1. <span>Hello</span>
2. <span>World</span>

### خودتان امتحان کنید<<

تگ هایی که دارای ویژگی پیش فرض inline هستند:

- <a>
- <abbr>
- <acronym>
- <b>
- <bdo>
- <big>
- <br>
- <button>
- <cite>
- <code>
- <dfn>
- <em>
- <i>
- <img>
- <input>
- <kbd>
- <label>
- <map>
- <object>
- <q>
- <samp>
- <script>
- <select>
- <small>
- <span>
- <strong>
- <sub>
- <sup>
- <textarea>
- <time>
- <tt>



## عنصر <div>

در HTML، تگ <div>، معمولاً به عنوان یک ظرف برای سایر عناصر استفاده می‌شود. تگ <div>، به صفات احتیاج ندارد، اما معمولاً با ویژگی **class**، **id** و **style** توصیف می‌شود.

زمانیکه از کدهای CSS همراه با صفاتی که گفته شد، استفاده کنید، محتوای تگ <div>، نیز می‌تواند ویژگی **block** داشته باشد.

### مثال : عنصر div

```
1. <div style="background-color:black;color:white;padding:20px;">
2.   <h2>London</h2>
3.   <p>London is the capital city of England. It is the most populous city in the United Kingdom, with
      a metropolitan area of over 13 million inhabitants.</p>
4. </div>
```

## [خودتان امتحان کنید <<](#)

### عنصر <span>

عنصر <span> اغلب بعنوان یک ظرف برای برخی از متن‌ها استفاده می‌شود. عنصر <span> دارای صفات مورد نیاز نیست، اما می‌توان از **class**، **id** و **style** استفاده کرد.

هنگامی که با CSS مورد استفاده قرار می‌گیرد، عنصر <span> می‌تواند برای **style** کردن قسمت‌های متن استفاده شود:

### مثال : عنصر span

```
1. <h1>My <span style="color:red">Important</span> Heading</h1>
```

## [خودتان امتحان کنید <<](#)

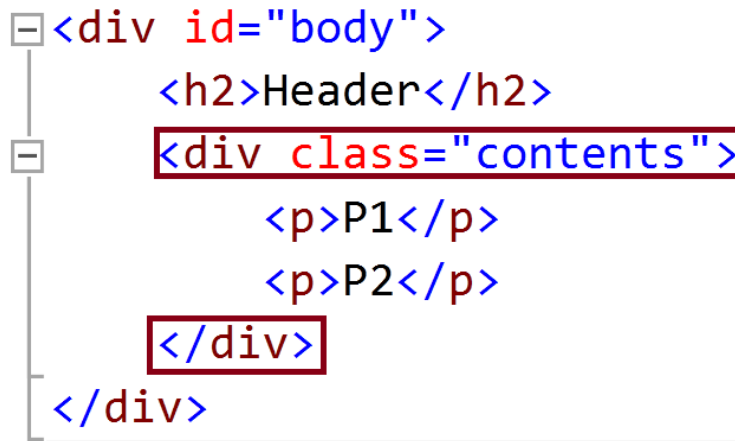
### تگ‌های گروه بندی در HTML

توضیحات	تگ
یک بخش را در صفحه وب تعریف می‌کند (block-level)	<div>
یک بخش را در صفحه وب تعریف می‌کند (inline))	<span>

### کلام آخر

تمام عناصر HTML که تاکنون ارائه شده عموماً شامل یکی از ویژگی‌های **Block** و **Inline** در HTML می‌باشند، به طور خلاصه ویژگی **block** باعث اشغال حداکثر فضای عنصر والد و ویژگی **Inline** بلعکس عمل می‌کند.

با عرض سلام خدمت کاربران سایت [پی وی لرن](http://www.pvlearn.com) . در مطالب آموزشی که تاکنون ارائه گردیده است. در مورد **کاربرد صفات در html** یا **کاربرد صفات کلاس ها در html** ، توضیحاتی ارائه شد. در این بخش از آموزش ، قصد داریم به بحث آموزش استفاده از صفت class در html ، بپردازیم. صفت کلاس در html کاربرد بسیار گسترده ای مخصوصا در ارتباط عناصر با کدهای CSS و جاوااسکریپت دارد. صفت class در واقع یک یا چند نام کلاس را برای یک عنصر html توصیف می کند. که هر کدام از این روشها ، کاربرد متفاوتی دارد.



### آموزش استفاده از صفت class در html

همانطو که اشاره کردیم ، نام صفت **class** یک عنصر html می تواند در کدهای CSS و جاوا اسکریپت استفاده شود. در CSS با استفاده از نام class یک عنصر html می توان style آن عنصر در CSS تغییر داد. و در جاوااسکریپت نیز می توان با استفاده از مشخصه class یک عنصر، یک رفتار خاص را برای آن عنصر تعریف کرد.

در کد زیر از سه <div> استفاده شده که هر سه به یک class اشاره دارند :

مثال : سه div هم class

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. .cities {
6.     background-color: black;
7.     color: white;
8.     margin: 20px;
9.     padding: 20px;
10. }
11. </style>
12. </head>
13. <body>
14.
15. <div class="cities">
16.   <h2>London</h2>
17.   <p>London is the capital of England.</p>
18. </div>
19.
20. <div class="cities">
21.   <h2>Paris</h2>
22.   <p>Paris is the capital of France.</p>

```

```
23. </div>
24.
25. <div class="cities">
26.   <h2>Tokyo</h2>
27.   <p>Tokyo is the capital of Japan.</p>
28. </div>
29.
30. </body>
31. </html>
```

## [خودتان امتحان کنید<<](#)

نتیجه ی کد فوق:

### London

London is the capital of England.

### Paris

Paris is the capital of France.

### Tokyo

Tokyo is the capital of Japan.

## استفاده از class در عناصر inline

مثال : تعریف class در عناصر inline

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. span.note {
6.   font-size: 120%;
7.   color: red;
8. }
9. </style>
10. </head>
11. <body>
12.
13. <h1>My <span class="note">Important</span> Heading</h1>
14. <p>This is some <span class="note">important</span> text.</p>
15.
16. </body>
17. </html>
```

## [خودتان امتحان کنید<<](#)

## انتخاب عناصر با یک کلاس مشخص

در CSS برای انتخاب عناصر با استفاده از ویژگی class آن ها قبل از نام class از یک نقطه (.) استفاده کنید.

مثال: استفاده از کدهای CSS برای تغییر دادن style تمام عناصری که دارای صفت class با نام "city" هستند:

مثال : مثالی از انتخاب عناصر با استفاده از class

```
1. <style>
2. .city {
3.     background-color: tomato;
4.     color: white;
5.     padding: 10px;
6. }
7. </style>
8.
9. <h2 class="city">London</h2>
10. <p>London is the capital of England.</p>
11.
12. <h2 class="city">Paris</h2>
13. <p>Paris is the capital of France.</p>
14.
15. <h2 class="city">Tokyo</h2>
16. <p>Tokyo is the capital of Japan.</p>
```

### [خودتان امتحان کنید](#)

نتیجه ی کد فوق :

London

London is the capital of England.

Paris

Paris is the capital of France.

Tokyo

Tokyo is the capital of Japan.

## استفاده از صفت class در جاوااسکریپت

جاوااسکریپت می تواند به عناصر html با استفاده از صفت class و با استفاده از تابع `getElementsByClassName()` دسترسی پیدا کند.

مثال- زمانیکه کاربر روی یک button کلیک کند تمام عناصر که دارای صفت class با نام "city" هستند. مخفی شوند:

مثال : صفت کلاس در جاوا اسکریپت

```
1. <script>
2. function myFunction() {
3.   var x = document.getElementsByClassName("city");
4.   for (var i = 0; i < x.length; i++) {
5.     x[i].style.display = "none";
6.   }
7. }
8. </script>
```

[خودتان امتحان کنید](#)

## صفت class چندگانه

هر کدام از عناصر html می تواند صفت class با بیشتر از یک نام را داشته باشد. در این صورت هر کدام از نام های باید با یک space از نام قبلی جدا شود.

مثال : صفات چندگانه

```
1. <h2 class="city main">London</h2>
2. <h2 class="city">Paris</h2>
3. <h2 class="city">Tokyo</h2>
```

[خودتان امتحان کنید](#)

## استفاده از یک نام class ، برای تگ های مختلف

می توانید از یک class برای تگ های گوناگون استفاده کنید.

مثال : استفاده از یک نام class در عناصر مختلف

```
1. <h2 class="city">Paris</h2>
2. <p class="city">Paris is the capital of France</p>
```

[خودتان امتحان کنید](#)

## کلام آخر

از صفت class در html برای مشخص کردن یک عنوان که در عناصر مختلف قابل تکرار باشد استفاده می شود، این عنوان قابلیت انتخاب عناصر html را در جاوا اسکریپت و CSS و سپس اعمال تغییرات لازم روی آن ها را می دهد.

### مقدمه

با عرض سلام خدمت کاربران سایت [پی وی لرن](#) . در آموزشهایی که در مطالب قبلی در مورد html ، ارائه گردید. توضیحات مفصلی در مورد [استفاده از صفت class در html](#) ، ارائه شد. در این بخش از آموزش نیز قصد داریم. به موضوع [آموزش استفاده از صفت id در html](#) یا [ویژگی id در html](#) و [کاربرد id در html](#) ، بپردازیم. صفت id ، در واقع یک مقدار منحصر به فرد را برای یک عنصر html را تعریف می کند. صفت id ، همانند صفت class یک نام را برای شناسایی و دستیابی به هر کدام از عنصر .html در کدهای CSS و جاوااسکریپت مشخص می کند. مقدار هر id برای هر کدام از عناصر html باید **یکتا و منحصر به فرد** باشد.

در ادامه به موضوع چگونگی استفاده از صفت id در html ، می پردازیم.

```
<body>
  <div>
    <input type="text" name="myname" id="myname" value="" />
    <br />
    <p>Text box using html helper</p>
    <input id="myname" name="myname" type="text" value="" />
  </div>
```

ویژگی id در html

### استفاده از صفت یا ویژگی id در html

همانطور که اشاره شد. صفت id ، همانند class برای دستیابی به ویژگی های یک عنصر ، در کدهای CSS و جاوااسکریپت ، مورد استفاده قرار می گیرد. اما مقدار id حتما باید منحصر به فرد بوده و در سایر عناصر تکرار نشود.

مثال: استفاده از کدهای CSS برای تغییر دادن محتویات عناصری با id با مقدار "myHeader" :

مثال : صفت id

```
1. <style>
2. #myHeader {
3.     background-color: lightblue;
4.     color: black;
5.     padding: 40px;
6.     text-align: center;
7. }
8. </style>
9.
10. <h1 id="myHeader">My Header</h1>
```

### [خودتان امتحان کنید <<](#)

نتیجه ی کد فوق :

# My Header

## تفاوت بین صفات id و class

تفاوت عمده بین id و class ، این است. که مقدار id ، در هر عنصر html باید یکتا باشد. و تکرار نشود ، اما یک نام class می تواند در چندین عنصر تکرار شود.

### مثال : تفاوت id و class

```
1. <style>
2. /* Style the element with the id "myHeader" */
3. #myHeader {
4.     background-color: lightblue;
5.     color: black;
6.     padding: 40px;
7.     text-align: center;
8. }
9.
10. /* Style all elements with the class name "city" */
11. .city {
12.     background-color: tomato;
13.     color: white;
14.     padding: 10px;
15. }
16. </style>
17.
18. <!-- A unique element -->
19. <h1 id="myHeader">My Cities</h1>
20.
21. <!-- Multiple similar elements -->
22. <h2 class="city">London</h2>
23. <p>London is the capital of England.</p>
24.
25. <h2 class="city">Paris</h2>
26. <p>Paris is the capital of France.</p>
27.
28. <h2 class="city">Tokyo</h2>
29. <p>Tokyo is the capital of Japan.</p>
```

## [خودتان امتحان کنید <<](#)

## bookmark با استفاده از id و link

bookmark ها اغلب برای ایجاد قابلیت پرش از قسمتی از صفحه وب به قسمتی دیگر برای خوانندگان صفحات وب ایجاد می شوند. Bookmark ها معمولا در صفحات وب طولانی و بلند کاربرد دارند، در ابتدا باید Bookmark را ایجاد کرده و سپس یک لینک به آن بدهید. سپس وقتی که کاربر روی آن کلیک کند، مرورگر به جایی که Bookmark ایجاد شده اسکرول می شود. ابتدا یک صفت id به عنصر مربوطه برای ایجاد Bookmark اضافه کنید :

### مثال :

1. `<h2 id="C4">Chapter 4</h2>`

حالا یک لینک به bookmark در همان صفحه وب ایجاد کنید:

مثال :

1. `<a href="#C4">Jump to Chapter 4</a>`

و یا اینکه از صفحات دیگر یک لینک به bookmark آن صفحه بدهید :

مثال : لینک از صفحات دیگر

1. `<a href="html_demo.html#C4">Jump to Chapter 4</a>`

[خودتان امتحان کنید<<](#)

## استفاده از id در جاوااسکریپت

جاوا اسکریپت می تواند. با استفاده از صفت id ، و تابع getElementById() ، به یک عنصر html دسترسی پیدا کند.

مثال: استفاده از جاوااسکریپت برای تغییر دادن محتوای عناصر:

مثال : استفاده از id در جاوا اسکریپت

```
1. <script>
2. function displayResult() {
3.     document.getElementById("myHeader").innerHTML = "Have a nice day!";
4. }
5. </script>
```

[خودتان امتحان کنید<<](#)

## کلام آخر

در مبحث قبلی چگونگی استفاده از صفت class برای انتخاب گروهی از عناصر html را در زبان های CSS و جاوا اسکریپت و ... را توضیح دادیم، اما برای انتخاب یک عنصر منحصر به فرد باید از **صفت id در html** به گونه ای که در مباحث فوق نیز اشاره شده استفاده کنید.



### مقدمه

با عرض سلام و وقت بخیر خدمت کاربران سایت [پی وی لرن](#) . و کاربرانی که [دوره کامل آموزش HTML](#) را دنبال می کنند. گاهی اوقات ممکن است نیاز داشته باشید که یک وب سایت و یا یک صفحه وب را در بخشی از وب سایت خود نمایش داده و به کاربر اجازه دهید تا به یک وب سایت یا بخشی از یک وب سایت دسترسی داشته باشد. برای این منظور باید با طرز کار با تگ **IFRAME** در HTML آشنایی داشته باشید. از این رو در این آموزش به مباحث آشنایی با **تگ IFRAME در HTML** پرداخته ایم.

### آشنایی با تگ IFRAME در HTML

در ادامه ی این آموزش شما را با ساختار کلی و کاربرد تگ IFRAME در HTML آشنا خواهیم کرد. برای نمایش یک صفحه وب در یک صفحه وب دیگر، از تگ IFRAME استفاده می شود:

### ساختار تگ IFRAME

یک عنصر IFRAME در HTML با استفاده از تگ `<iframe >` تعریف می شود:

مثال :

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

در ساختار فوق صفت SRC آدرس صفحه وبی را که در قالب تگ IFRAME نمایش داده می شود را توصیف می کند.

### تنظیم width و height در تگ IFRAME

از صفات height و width برای تنظیم ارتفاع و عرض عنصر IFRAME استفاده کنید. مقدار این ویژگی به طور پیش فرض در پیکسل مشخص می شود، اما همچنین می تواند در درصد (مانند "۸۰٪") باشد.

مثال : تنظیم عرض و ارتفاع IFRAME

```
1. <iframe src="http://pvlearn.com/product/full-html-and-htm15-tutorial/" height="200" width="300"></iframe>
```

### [خودتان امتحان کنید <<](#)

همچنین شما می توانید از CSS برای تنظیم طول و ارتفاع عناصر IFRAME استفاده کنید.

مثال : استفاده از style

```
1. <iframe src="http://pvlearn.com/product/full-html-and-htm15-tutorial/" style="height:200px;width:300px;"></iframe>
```

### [خودتان امتحان کنید <<](#)

### frame

در حالت عادی عنصر IFRAME با یک border مشخص در اطراف عنصر نمایش داده می شود. برای حذف border باید style مربوط به حذف border از عنصر IFRAME را به آن اضافه کنید.

1. `<iframe src="http://pvlearn.com/product/full-html-and-html5-tutorial/" style="border:none;"></iframe>`

### [خودتان امتحان کنید](#)

همچنین شما می توانید از css برای حذف border از iframe استفاده کنید.

1. `<iframe src="http://pvlearn.com/product/full-html-and-html5-tutorial/" style="border:2px solid red;"></iframe>`

### [خودتان امتحان کنید](#)

## لینک در iframe

یک iframe را می توان به عنوان یک target frame برای لینک استفاده کرد. صفت `target` از یک لینک باید به صفت `name` از یک عنصر iframe اشاره کند.

1. `<iframe src="http://pvlearn.com/product/full-html-and-html5-tutorial/" name="iframe_a"></iframe>`
- 2.
3. `<p><a href="https://www.pvlearn.com" target="iframe_a">W3Schools.com</a></p>`

### [خودتان امتحان کنید](#)

## کلام آخر

گاهی اوقات نیاز است که یک وب سایت دیگر یا یک صفحه وب را در قسمتی از وب سایت خود نمایش داده و در دسترس کاربران قرار دهید برای اینگونه موارد **تگ Iframe در html** گزینه ی بسیار مناسبی است.

### مقدمه

با عرض سلام خدمت کاربران سایت [پی وی لرن](#) . در این پست آموزشی قصد داریم به آموزش چگونگی استفاده از کدهای جاوااسکریپت در [html](#) بپردازیم. جاوااسکریپت ( **JavaScript** ) در حقیقت به خودی خود یک زبان سطح بالا محسوب می شود. که زبان سمت سرور آن با نام **Nodjs** در حوزه های گسترده ای در طراحی سایت و برنامه های کاربردی قابل نصب و استفاده بر روی کامپیوتر و گوشی موبایل استفاده می شود. از جاوااسکریپت در html برای دادن رفتارهایی به صفحه وب و ایجاد قابلیت پاسخ دادن به درخواست کاربر، استفاده می شود. صفحه وبی که تنها از html و css تشکیل شده و فاقد کدهای JavaScript است. هیچگونه عکس العملی در برابر درخواست کاربر نشان نمی دهد.

در مطالب آموزشی قبلی در حوزه html ما به آموزش چگونگی کار با تگ های [Table](#) , [Div](#) , تگ [img](#) , [تعریف رنگ ها در html](#) , [تعریف فرم ها در html](#) , پرداختیم. در این پست نیز به چگونگی کار با کدهای جاوااسکریپت در html می پردازیم.

```
<script type="text/javascript">
  function getInspiration() {
    if (morningDay == "depressed")
      start.Coding & get.Awesome;
  }
</script>
```

-JS کدهای جاوااسکریپت در html

### جاوااسکریپت در html

JavaScript صفحات وب را پویا و تعاملی می سازد.

```

1. <!DOCTYPE html>
2. <html>
3. <body>
4.
5. <h1>My First JavaScript</h1>
6.
7. <button type="button"
8. onclick="document.getElementById('demo').innerHTML = Date()">
9. Click me to display Date and Time.</button>
10.
11. <p id="demo"></p>
12.
13. </body>
14. </html>

```

### [خودتان امتحان کنید<>](#)

## تگ <script> در html

تگ <script> برای ایجاد یک اسکریپت سمت سروریس گیرنده استفاده می شود. تگ <script> یا خود حاوی کدهای اسکریپتی می باشد، یا به یک فایل اسکریپت خارجی از طریق ویژگی **src** اشاره می کند.

در جاوااسکریپت برای انتخاب کردن و اعمال رفتار روی هر کدام از عناصر html از دستور :

مثال :

```
1. ()document.getElementById
```

استفاده می شود. در مثال ساده زیر پس از اجرا ، عبارت "Hello JavaScript!" در عنصر html با یک ID با نام "demo" قرار می گیرد.

مثال : مثالی از جاوا اسکریپت

```

1. <script>
2. document.getElementById("demo").innerHTML = "Hello JavaScript!";
3. </script>

```

### [خودتان امتحان کنید<>](#)

## مثال هایی از جاوا اسکریپت

در JavaScript می توان همانند CSS به style های عناصر html دسترسی یافته و آنها را تغییر داد.

مثال : مثال-1

```
1. document.getElementById("demo").innerHTML = "Hello JavaScript!";
```

### [خودتان امتحان کنید<>](#)

```
1. document.getElementById("demo").style.fontSize = "25px";
2. document.getElementById("demo").style.color = "red";
3. document.getElementById("demo").style.backgroundColor = "yellow";
```

[خودتان امتحان کنید >>](#)

```
1. document.getElementById("image").src = "picture.gif";
```

[خودتان امتحان کنید >>](#)

## تگ <noscript>

تگ <noscript> برای ارائه یک محتوای جایگزین برای کاربرانی است که اسکریپت ها را در مرورگر خود غیرفعال کرده اند و یا از مرورگری استفاده می کنند که از اسکریپت های سمت سرور پشتیبانی نمی کند.

مثال : تگ &lt;noscript&gt;

```
1. <script>
2. document.getElementById("demo").innerHTML = "Hello JavaScript!";
3. </script>
4.
5. <noscript>Sorry, your browser does not support JavaScript!</noscript>
```

[خودتان امتحان کنید >>](#)

## تگ های script در html

توضیحات	تگ
یک اسکریپت سمت سرور دهنده را تعریف می کند.	<script>
یک محتوای جایگزین برای کاربرانی که از اسکریپت های سمت سرور پشتیبانی نمی کنند، تعریف می کند.	<noscript>

## کلام آخر

صفحات وب سایت های معمولاً شامل سه بخش html و css و جاوا اسکریپت می باشد، و html و css ظاهر عناصر html را می سازند، اما اسکریپت های **جاوا اسکریپت در HTML** صفحات وب را تعاملی می سازد، و امکان تعریف رفتارهای خاص برای عناصر html را فراهم می کند.

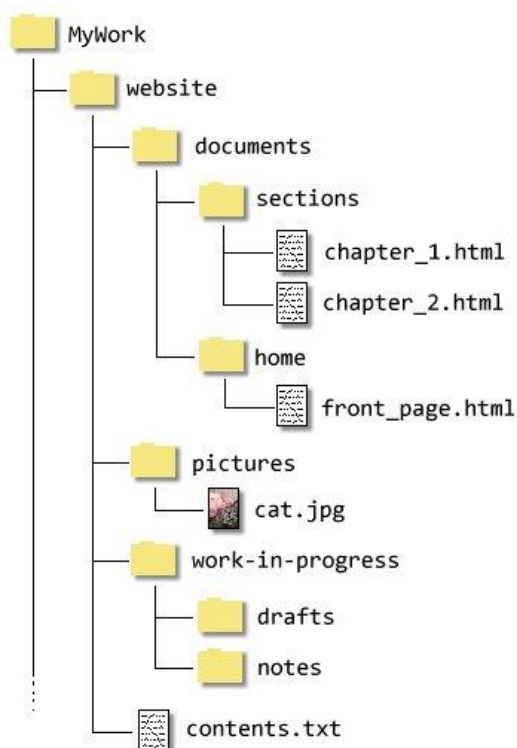
## جلسه ۲۳ : آدرس دهی فایل ها در HTML

### مقدمه

با عرض سلام خدمت کاربران سایت [پی وی لرن](#) . در این پست آموزشی قصد داریم به آموزش روشهای آدرس دهی فایل ها در تگ های نظیر `<img>` در html بپردازیم. تمام تصاویر و فایل هایی که در html بارگزاری می شود. نیاز به آدرس دهی صحیح محل ذخیره سازی فایل، در سیستم دارد. اگر آدرس دهی صحیح نباشد عکس و فایل مورد نظر در وب بارگزاری نخواهد شد. و صفحه وب بدون عکس و سایر فایل ها خسته کننده بوده و کارایی چندانی ندارد. بنابراین آدرس دهی صحیح فایل ها در html از اهمیت ویژه ای برخوردار است.

در مطالب آموزشی قبلی در حوزه html ما به آموزش چگونگی کار با تگ های `Table` , `Div` , `img` , [تعریف رنگ ها در html](#) , [تعریف فرم ها در html](#) و ... پرداختیم. و به [آموزش کار با گرافیک در html](#) نیز اشاره کردیم.

در این پست آموزشی نیز به آموزش روشهای آدرس دهی فایل ها در html می پردازیم.



file-روشهای آدرس دهی

### روشهای آدرس دهی فایل ها در html

مسیر فایل محل قرارگیری یک فایل را در ساختار پوشه وب سایت توصیف می کند. آدرس دهی فایل ها در بخشهای زیر در صفحات وب ، کاربرد دارد:

- Web pages
- Images
- Style sheets
- JavaScripts

## آدرس اینترنتی

آدرس اینترنتی، آدرس یک فایل موجود در یک صفحه وب در اینترنت است. که ما آن را در صفحه وب خود آدرس دهی می کنیم.

مثال :

1. ``

## آدرس نسبی

این آدرس به یک فایل موجود در فولدر های محلی که فایل وب سایت ما در آنجا قرار گرفته، اشاره دارد.

مثال: آدرس دهی عکسی که در فولدری به نام **images** است، که این فولدر در محل قرارگیری وب سایت ما در کامپیوتر قرار دارد:

مثال : آدرس دهی نسبی

1. ``

## آدرس دهی زیر شاخه

در مثال زیر به فولدری به نام **images** در زیر شاخه آدرس دهی شده است :

مثال : آدرس دهی به فولدر زیر شاخه

1. ``

## آدرس دهی سطوح بالاتر

این روش، زمانی که یک فایل در فولدری که یک یا چند سطح بالاتر از محل ذخیره وب سایت ما در کامپیوتر قرار گرفته، استفاده می شود.

مثال:

آدرس دهی عکسی که در فولدری به نام **images** است، که این فولدر یک سطح بالاتر از محل قرارگیری وب سایت ما قرار گرفته است:

مثال : آدرس دهی در زیر شاخه

1. ``

## کلام آخر

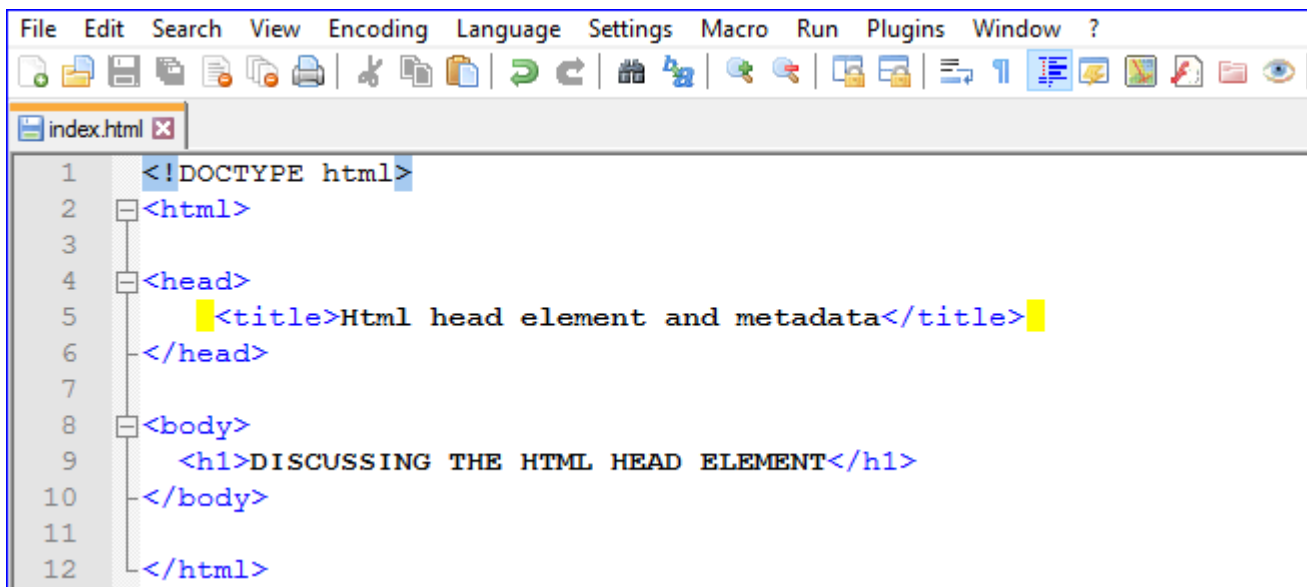
در صفحات وب فایل های مختلف صوتی، تصویری (مانند تصاویر) و... وجود دارد که هر کدام در محل های مختلف هاست ذخیره شده و برای دستیابی به آن ها و نمایش در سایت باید از آدرس دهی فایل ها در **HTML** طبق موارد فوق استفاده کرد.

### مقدمه

با عرض سلام خدمت کاربران سایت [پی وی لرن](#) . در مباحثی که تاکنون در دوره آموزشی html ، ارائه گردیده است. به آموزش برخی از تگ های پرکاربرد ، از جمله تگ div و کاربرد تگ img، پرداختیم. در ادامه ی این آموزشها در این بخش ، قصد داریم به موضوع **آشنایی با کاربرد تگ head در html** یا **عنصر head و کاربرد آن در HTML** ، بپردازیم و کاربردهای این تگ را در ( html تگ هد در )، بررسی کنیم.

همانطور که می دانید تگ <head> ، از تگ های کلیدی و بسیار مهم است که در تمام صفحات وب وجود دارد. ایت تگ همانند یک ظرف برای نگهداری برخی از تگ های مهم و پرکاربرد دیگر مورد استفاده قرار می گیرد.

در ادامه ی این بخش به بررسی کاربرد تگ head در html ، می پردازیم.



```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5     <title>Html head element and metadata</title>
6 </head>
7
8 <body>
9     <h1>DISCUSSING THE HTML HEAD ELEMENT</h1>
10 </body>
11
12 </html>
```

کاربرد تگ head در html

### آشنایی با کاربرد تگ head در html

بسیاری از عناصری که در تگ head ، قرار می گیرند در صفحه وب نمایش داده نمی شوند. اما حاوی اطلاعات بسیاری برای شکل گیری ساختار صفحات وب هستند. از جمله می توان به تگ ها ی <meta> ، <link> و <script> اشاره کرد، برخی از تگ ها نیز مانند تگ <title> نمایش داده می شود.

### تگ title

همانطور که از نامش پیداست، این تگ برای نمایش عنوان سایت در مرورگر ها استفاده می شود. برخی دیگر از کاربردهای این تگ به شرح زیر است:

- زمانی که صفحه وب به بخش علاقه مندی ها ، اضافه شود عنوان آن را در آن بخش مشخص می کند.
- عنوان وب سایت را در لیست نتایج جستجوهای موتورهای جستجوگر مشخص می کند.
- در tab هر مرورگر عنوان سایت را قرار می دهد.



```

1. <!DOCTYPE html>
2. <html>
3.
4. <head>
5.   <title>Page Title</title>
6. </head>
7.
8. <body>
9. The content of the document.....
10. </body>
11.
12. </html>

```

[خودتان امتحان کنید<<](#)**تگ <style>**

این عنصر برای تعریف کردن style های عناصر یک صفحه وب مورد استفاده قرار می گیرد.

```

1. <style>
2.   body {background-color: powderblue;}
3.   h1 {color: red;}
4.   p {color: blue;}
5. </style>

```

[خودتان امتحان کنید<<](#)**تگ <link>**

از این تگ برای لینک دادن به style sheets (بخش کدهای css) ، مورد استفاده قرار می گیرد.

```

1. <link rel="stylesheet" href="mystyle.css">

```

[خودتان امتحان کنید<<](#)**تگ <meta>**

این تگ کاربرد های بسیاری در html دارد. این تگ در واقع مجموعه ای از کاراکتر ها از جمله توصیف صفحات وب ، کلمات کلیدی ، کاراکترهای زبان های مختلف و ... را در صفحه وب مشخص می کند. تگ <meta> در مرورگر ها چگونگی نمایش دادن محتوای سایت را مشخص می کند. در موتورهای جستجو ، کلمات کلیدی سایت را ارائه می دهد.

شناسایی کاراکتر های زبان فارسی در وب:

```

1. <meta charset="UTF-8">

```

## مثال :

```
1. <meta name="keywords" content="HTML, CSS, XML, JavaScript">
```

بروزرسانی صفحه وب در هر ۳۰ ثانیه:

## مثال :

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

## مثال : مثالی از تگ meta

```
1. <meta charset="UTF-8">
2. <meta name="description" content="Free Web tutorials">
3. <meta name="keywords" content="HTML,CSS,XML,JavaScript">
4. <meta name="author" content="John Doe">
```

[خودتان امتحان کنید](#)**تنظیم viewport با تگ <meta>**

در html5 این امکان را به طراحان می دهد که بتوانند viewport (آن بخش از صفحه وب ، که کاربر آن را مشاهده می کند.) صفحات وب خود را به راحتی برای دستگاههای مختلف تنظیم کنند. اینکار به راحتی با تگ <meta> ، قابل اجراست:

## مثال :

```
1. <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

**عنصر <script> در html**

از عنصر <script> برای تعریف کدهای جاوا اسکریپت استفاده می شود. مثال-نمایش عبارت "Hello JavaScript!" در عنصری با "id="demo" :

## مثال : کدهای اسکریپت

```
1. <script>
2. function myFunction {
3.     document.getElementById("demo").innerHTML = "Hello JavaScript!";
4. }
5. </script>
```

## عنصر <base> در html

عنصر <base> آدرس URL پایه را برای تمام URL های مرتبط در یک صفحه مشخص می کند:

مثال : آدرس پایه

```
1. <base href="https://www.pvlearn.com" target="_blank">
```

## [خودتان امتحان کنید](#)

بر طبق استاندارد HTML5؛ <html>، تگ های <body> و <head> را می توان حذف کرد:

مثال : حذف تگ های head و body

```
1. <!DOCTYPE html>
2. <title>Page Title</title>
3.
4. <h1>This is a heading</h1>
5. <p>This is a paragraph.</p>
```

## [خودتان امتحان کنید](#)

## عناصر در head در html

توضیحات	تگ
اطلاعات سند html را نگه می دارد.	<head>
عنوان صفحه را تعریف می کند.	<title>
آدرس url پایه را برای صفحه تعریف می کند.	<base>
برای تعریف آدرس فایل های خارجی مانند CSS و جاوا اسکریپت بکار می رود.	<link>

اطلاعات metadata ها را تعریف می کند.	<meta>
اسکرپت های جاوا اسکرپت را تعریف می کند.	<script>
عناصر html یک صفحه وب را style می کند.	<style>

## کلام آخر

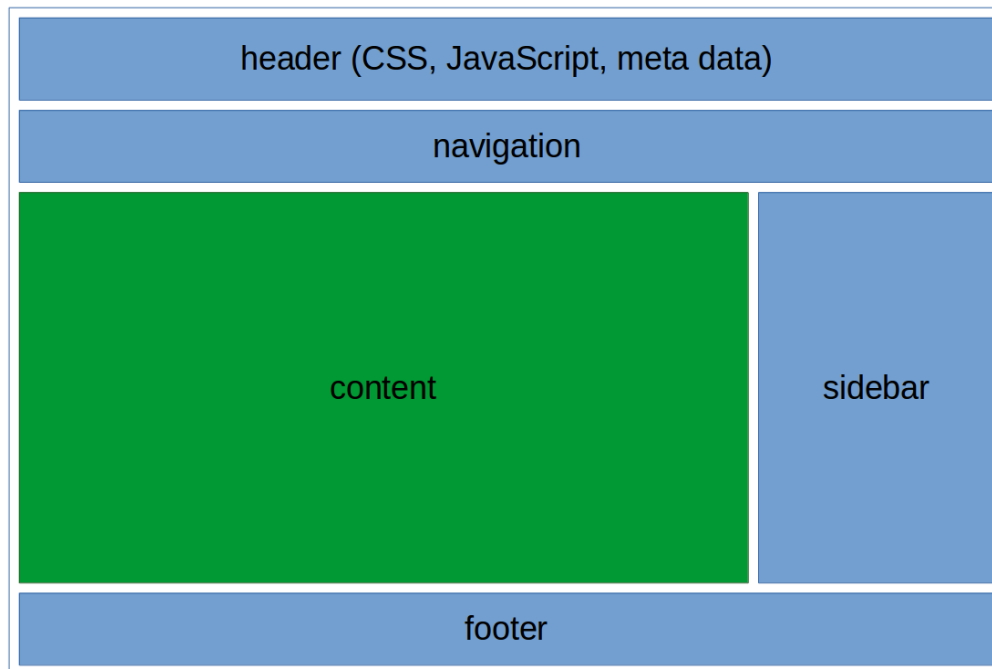
یکی از تگ های اصلی و تشکیل دهنده ی ساختار صفحات وب، **تگ head در html** می باشد که اطلاعات مربوطه به metadata ، آدرس دهی link ها ، عنوان صفحه و... را نگه می دارد، که تعریف آن در نسخه های HTML 4 ضروری است.

## جلسه ۲۵ : قالب وب سایت

### مقدمه

با عرض سلام خدمت کاربران سایت [پی وی لرن](#) . ما در بخش [دوره کامل آموزش CSS](#) ، به طراحی کلی قالب صفحات وب ، اشاره کردیم. و توضیحاتی در این مورد ، در بخش [آموزش CSS](#) ، ارائه کردیم. اکنون در ادامه آموزشهای کامل html ، در این بخش قصد داریم به موضوع [آشنایی با طرح های کلی قالب وب سایت یا طراحی قالب وب سایت](#) ، بپردازیم.

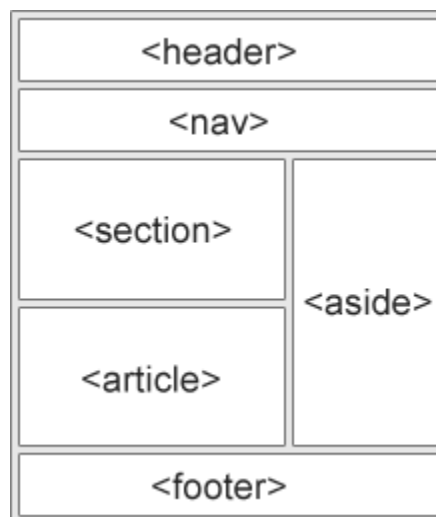
وب سایت های معمولا محتوای خود را در چندین ستون (مانند مجله ها و روزنامه ها) نمایش می دهند. در html5 یک شماتیک کلی برای قالب وب سایت ، ارائه شده است. که در ادامه این آموزش به آن می پردازیم.



Layout در طراحی قالب وب سایت

### آشنایی با طرح های کلی قالب وب سایت

طرح کلی که html5 برای قالب وب سایت ارائه کرده است ، بصورت زیر است:



```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4. <title>CSS Template</title>
5. <meta charset="utf-8">
6. <meta name="viewport" content="width=device-width, initial-scale=1">
7. <style>
8. * {
9.     box-sizing: border-box;
10. }
11.
12. body {
13.     font-family: Arial, Helvetica, sans-serif;
14. }
15.
16. /* Style the header */
17. header {
18.     background-color: #666;
19.     padding: 30px;
20.     text-align: center;
21.     font-size: 35px;
22.     color: white;
23. }
24.
25. /* Create two columns/boxes that floats next to each other */
26. nav {
27.     float: left;
28.     width: 30%;
29.     height: 300px; /* only for demonstration, should be removed */
30.     background: #ccc;
31.     padding: 20px;
32. }
33.
34. /* Style the list inside the menu */
35. nav ul {
36.     list-style-type: none;
37.     padding: 0;
38. }
39.
40. article {
41.     float: left;
42.     padding: 20px;
43.     width: 70%;
44.     background-color: #f1f1f1;
45.     height: 300px; /* only for demonstration, should be removed */
46. }
47.
48. /* Clear floats after the columns */
49. section:after {
50.     content: "";
51.     display: table;
52.     clear: both;
53. }
54.
55. /* Style the footer */
56. footer {
57.     background-color: #777;
58.     padding: 10px;
59.     text-align: center;
60.     color: white;
61. }
```

```

62.
63. /* Responsive layout - makes the two columns/boxes stack on top of each other instead of next to each other, on small screens */
64. @media (max-width: 600px) {
65.     nav, article {
66.         width: 100%;
67.         height: auto;
68.     }
69. }
70. </style>
71. </head>
72. <body>
73.
74. <h2>CSS Layout Float</h2>
75. <p>In this example, we have created a header, two columns/boxes and a footer. On smaller screens, the columns will stack on top of each other.</p>
76. <p>Resize the browser window to see the responsive effect (you will learn more about this in our next chapter - HTML Responsive.)</p>
77.
78. <header>
79.   <h2>Cities</h2>
80. </header>
81.
82. <section>
83.   <nav>
84.     <ul>
85.       <li><a href="#">London</a></li>
86.       <li><a href="#">Paris</a></li>
87.       <li><a href="#">Tokyo</a></li>
88.     </ul>
89.   </nav>
90.
91.   <article>
92.     <h1>London</h1>
93.     <p>London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.</p>
94.     <p>Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.</p>
95.   </article>
96. </section>
97.
98. <footer>
99.   <p>Footer</p>
100. </footer>
101.
102. </body>
103. </html>

```

## خودتان امتحان کنید

- **<header>** : یک سربرگ برای صفحه وب تعریف می کند.
- **<nav>** : ظرفی برای نگهداری لینک های navigation است.
- **<section>** : بخشی از یک سند را تعریف می کند.
- **<article>** : یک مقاله مستقل را در صفحه وب تعریف می کند.
- **<aside>** : محتوای حاشیه ی وب سایت را تعریف می کند.
- **<footer>** : پانوشته ی یک صفحه وب است.
- **<details>** : جزئیات را مشخص می کند.

- **<summary>** : به عنوان یک سربرگ برای جزئیات عمل می کند.

## تکنیک های طرح بندی قالب html

۵ روش کلی برای ایجاد طرح های چند ستونی در قالب وب سایت ، وجود دارد، که هر کدام دارای مزایای و معایبی می باشد:

- HTML tables (توصیه نمی شود)
- CSS float property
- CSS flexbox
- CSS framework

اکنون به بررسی هر کدام از این روشها می پردازیم:

### روش HTML tables

تگ `<table>` در html در واقع برای ایجاد طرح قالب وب سایت ، ارائه نشده است. هدف اصلی استفاده از تگ `<table>` ، در واقع نمایش جدولی داده ها است. بنابراین سعی کنید از جداول برای طراحی قالب وب سایت خود استفاده نکنید! چراکه باعث افزایش حجم کد وب سایت شما شده و اگر بخواهید بعدا طرح وب سایت خود را تغییر بدهید، دچار مشکلاتی می شوید.

### روش CSS framework

اگر شما می خواهید که قالب وب سایت خود را سریع طراحی کنید ، می توانید از فریم ورک های آماده ی وب سایت نظیر [W3.CSS](#) و [Bootstrap](#) استفاده کنید.

### روش CSS float

این روش کلی برای طراحی قالب کلی وب سایت است. این روش برای یادگیری آسان است. شما فقط باید یاد بگیرید که خصوصیات float و clear ، چه طور کار می کنند.

مثال : استفاده از صفت float

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4. <title>CSS Template</title>
5. <meta charset="utf-8">
6. <meta name="viewport" content="width=device-width, initial-scale=1">
7. <style>
8. * {
9.     box-sizing: border-box;
10. }
11.
12. body {
13.     font-family: Arial, Helvetica, sans-serif;
14. }
15.
16. /* Style the header */
17. header {
18.     background-color: #666;
19.     padding: 30px;
20.     text-align: center;
21.     font-size: 35px;
22.     color: white;
23. }
```



```

24.
25. /* Create two columns/boxes that floats next to each other */
26. nav {
27.     float: left;
28.     width: 30%;
29.     height: 300px; /* only for demonstration, should be removed */
30.     background: #ccc;
31.     padding: 20px;
32. }
33.
34. /* Style the list inside the menu */
35. nav ul {
36.     list-style-type: none;
37.     padding: 0;
38. }
39.
40. article {
41.     float: left;
42.     padding: 20px;
43.     width: 70%;
44.     background-color: #f1f1f1;
45.     height: 300px; /* only for demonstration, should be removed */
46. }
47.
48. /* Clear floats after the columns */
49. section:after {
50.     content: "";
51.     display: table;
52.     clear: both;
53. }
54.
55. /* Style the footer */
56. footer {
57.     background-color: #777;
58.     padding: 10px;
59.     text-align: center;
60.     color: white;
61. }
62.
63. /* Responsive layout - makes the two columns/boxes stack on top of each other instead of next to each other, on small screens */
64. @media (max-width: 600px) {
65.     nav, article {
66.         width: 100%;
67.         height: auto;
68.     }
69. }
70. </style>
71. </head>
72. <body>
73.
74. <h2>CSS Layout Float</h2>
75. <p>In this example, we have created a header, two columns/boxes and a footer. On smaller screens, the columns will stack on top of each other.</p>
76. <p>Resize the browser window to see the responsive effect (you will learn more about this in our next chapter - HTML Responsive.)</p>
77.
78. <header>
79.   <h2>Cities</h2>
80. </header>
81.
82. <section>
83.   <nav>
84.     <ul>

```

```

85.     <li><a href="#">London</a></li>
86.     <li><a href="#">Paris</a></li>
87.     <li><a href="#">Tokyo</a></li>
88. </ul>
89. </nav>
90.
91. <article>
92.   <h1>London</h1>
93.   <p>London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.</p>
94.   <p>Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.</p>
95. </article>
96. </section>
97.
98. <footer>
99.   <p>Footer</p>
100. </footer>
101.
102. </body>
103. </html>

```

## [خودتان امتحان کنید](#)

### روش Flexbox

Flexbox به طرح جدیدی است که در CSS3 ایجاد شده است. این روش تضمین می‌کند که رفتار قالب وب سایت، در وضعیت‌های مختلف قابل پیش‌بینی باشد. (زمانیکه صفحه نمایش در دستگاه‌هایی با اندازه‌های مختلف اجرا می‌شود).

مثال : استفاده از flex

```

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4. <title>CSS Template</title>
5. <meta charset="utf-8">
6. <meta name="viewport" content="width=device-width, initial-scale=1">
7. <style>
8. * {
9.   box-sizing: border-box;
10. }
11.
12. body {
13.   font-family: Arial, Helvetica, sans-serif;
14. }
15.
16. /* Style the header */
17. header {
18.   background-color: #666;
19.   padding: 30px;
20.   text-align: center;
21.   font-size: 35px;
22.   color: white;
23. }
24.
25. /* Container for flexboxes */
26. section {
27.   display: -webkit-flex;

```

```

28.     display: flex;
29. }
30.
31. /* Style the navigation menu */
32. nav {
33.     -webkit-flex: 1;
34.     -ms-flex: 1;
35.     flex: 1;
36.     background: #ccc;
37.     padding: 20px;
38. }
39.
40. /* Style the list inside the menu */
41. nav ul {
42.     list-style-type: none;
43.     padding: 0;
44. }
45.
46. /* Style the content */
47. article {
48.     -webkit-flex: 3;
49.     -ms-flex: 3;
50.     flex: 3;
51.     background-color: #f1f1f1;
52.     padding: 10px;
53. }
54.
55. /* Style the footer */
56. footer {
57.     background-color: #777;
58.     padding: 10px;
59.     text-align: center;
60.     color: white;
61. }
62.
63. /* Responsive layout - makes the menu and the content (inside the section) sit on top of each other
   instead of next to each other */
64. @media (max-width: 600px) {
65.     section {
66.         -webkit-flex-direction: column;
67.         flex-direction: column;
68.     }
69. }
70. </style>
71. </head>
72. <body>
73.
74. <h2>CSS Layout Flexbox</h2>
75. <p>In this example, we have created a header, two columns/boxes and a footer. On smaller screens, the
   columns will stack on top of each other.</p>
76. <p>Resize the browser window to see the responsive effect.</p>
77. <p><strong>Note:</strong> Flexbox is not supported in Internet Explorer 10 and earlier versions.</p>
78.
79. <header>
80.   <h2>Cities</h2>
81. </header>
82.
83. <section>
84.   <nav>
85.     <ul>
86.       <li><a href="#">London</a></li>
87.       <li><a href="#">Paris</a></li>
88.       <li><a href="#">Tokyo</a></li>

```

```
89.     </ul>
90. </nav>
91.
92. <article>
93.   <h1>London</h1>
94.   <p>London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.</p>
95.   <p>Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.</p>
96. </article>
97. </section>
98.
99. <footer>
100.   <p>Footer</p>
101. </footer>
102.
103. </body>
104. </html>
```

## [خودتان امتحان کنید](#)

### استفاده از Grid در CSS

ماژول CSS Grid یک سیستم طرح بندی مبتنی بر شبکه با ردیف ها و ستون ها را ارائه می دهد که ساده تر ساختن صفحات وب را بدون نیاز به استفاده از float ها و position ها انجام می دهد.

### کلام آخر

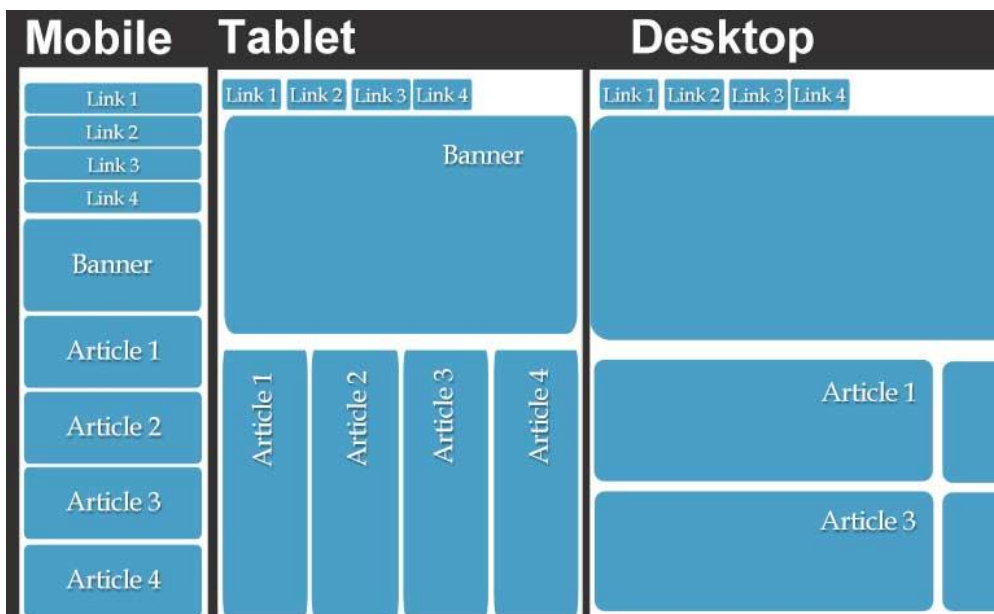
تاکنون شما را با برخی از عناصر مهم HTML ، صفات عناصر ، کدنویسی CSS و کمی جاوا اسکریپت آشنا کردیم، تا اینجا کار شما می توانید همانطور که در مباحث فوق اشاره کردیم قالب کلی و ساده ی صفحات HTML را پیاده سازی کنید.

در بخش های بعدی شما را با مباحث بیشتری از [طراحی قالب وب سایت](#) آشنا خواهیم کرد.

### مقدمه

با عرض سلام خدمت کاربران سایت [پی وی لرن](#) . در آموزشهایی که تاکنون در دوره html ارائه شده است به بحث [طراحی قالب وب سایت](#) و روشهای مختلف طراحی پرداختیم. و اشاره ای کوتاه نیز به بحث ویژگی **Responsive** در html ، داشتیم. البته در [دوره آموزش کامل CSS](#) ، مبحث **Responsive** و [فریم ورک های آماده](#) را ، بصورت کاملتر مورد بررسی و آموزش ، قرار دادیم.

در ادامه آموزشهای html ، در این بخش به موضوع **آشنایی با ویژگی Responsive** یا **طراحی قالب واکنشگرا** در html ، می پردازیم.



RS

### آشنایی با ویژگی Responsive در html ( ریسپانسیو در html )

طراحی قالب وب سایت بصورت Responsive ، در واقع استفاده از کدهای html و css برای تغییر اندازه ، مخفی شدن ، بزرگ کردن نسبی خودکار وب سایت در وضعیت ها و دستگاه های مختلف است.

### تنظیمات viewport

زمانیکه در حال طراحی صفحات وب responsive هستید، تگ <meta> را با تنظیمات زیر برای تمام عناصر اعمال کنید:

مثال : تنظیمات viewport

1. `<meta name="viewport" content="width=device-width, initial-scale=1.0">`

### [خودتان امتحان کنید<<](#)

کد بالا viewport صفحات وب شما را تنظیم می کند. و دستورالعمل های مرورگر را در مورد نحوه کنترل ابعاد صفحه و مقیاس بندی ارائه می دهد.

## تصاویر Responsive

در مورد [Responsive کردن تصاویر در آموزش های CSS](#) ، توضیحاتی ارائه کردیم، اکنون به چگونگی اعمال کردن ویژگی responsive در تصاویر با کد html می پردازیم.

### استفاده از ویژگی width

اگر خصوصیت width یک تصویر با ارزش ۱۰۰٪ ، مقدار دهی شود، عکس responsive خواهد شد.



مثال : رسپانسیو عکس ها

1. ``

## خودتان امتحان کنید»

در مثال بالا ، مشکلی که بوجود می آید این است که در صورت بزرگ شدن اندازه صفحه، اندازه تصویر نیز از اندازه اصلی خود بزرگ تر می شود، و کیفیت تصویر ممکن است نامطلوب شود. برای حرف این مشکل پیشنهاد می شود که از خصوصیت **max-width** استفاده شود.

### خصوصیت **max-width**

اگر از ویژگی **max-width** استفاده کنید، تصویر در شرایط لازم کوچکتر می شود ، اما هیچگاه از اندازه اصلی خود بزرگتر نمی شود.



مثال : خصوصیت **max-width**



1. ``

## [خودتان امتحان کنید<<](#)

### نمایش تصاویر مختلف بر اساس عرض (width) هر مرورگر

تگ `<picture>` اجازه می دهد که تصاویر مختلفی را برای اندازه های مختلف پنجره مرورگر ها تعریف کنید.



مثال : تگ picture

```
1. <picture>
2.   <source srcset="img_smallflower.jpg" media="(max-width: 600px)">
3.   <source srcset="img_flowers.jpg" media="(max-width: 1500px)">
4.   <source srcset="flowers.jpg">
5.   
6. </picture>
```

## [خودتان امتحان کنید<<](#)

### متن Responsive

اندازه متن می تواند با واحد "vw" تنظیم شود. که در این صورت با اندازه viewport تنظیم خواهد شد.

مثال : اندازه ی فونت

1. `<h1 style="font-size:10vw">Hello World</h1>`

## [خودتان امتحان کنید<<](#)

## استفاده از media queries

media queries این اجازه را می دهد که با استفاده از CSS صفحات وب خود را responsive کنید.

در آموزشهای CSS ، توضیحاتی در مورد نحوه [کار با media queries](#) ، ارائه کردیم.

### مثال : استفاده از media query

```
1. <style>
2. .left, .right {
3.   float: left;
4.   width: 20%; /* The width is 20%, by default */
5. }
6.
7. .main {
8.   float: left;
9.   width: 60%; /* The width is 60%, by default */
10. }
11.
12. /* Use a media query to add a breakpoint at 800px: */
13. @media screen and (max-width: 800px) {
14.   .left, .main, .right {
15.     width: 100%; /* The width is 100%, when the viewport is 800px or smaller */
16.   }
17. }
18. </style>
```

### [خودتان امتحان کنید](#)

## مثال کامل responsive صفحه وب

یک صفحه وب responsive باید روی صفحه نمایش بزرگ دسک تاپ و تلفن های همراه کوچک، به خوبی قابل نمایش باشد:

### مثال : صفحه وب responsive

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta name="viewport" content="width=device-width, initial-scale=1.0">
5. <style>
6. * {
7.   box-sizing: border-box;
8. }
9. .menu {
10.  float:left;
11.  width:20%;
12.  text-align:center;
13. }
14. .menu a {
15.  background-color:#e5e5e5;
16.  padding:8px;
17.  margin-top:7px;
18.  display:block;
19.  width:100%;
20.  color:black;
21. }
22. .main {
23.  float:left;
```

```

24. width:60%;
25. padding:0 20px;
26. }
27. .right {
28. background-color:#e5e5e5;
29. float:left;
30. width:20%;
31. padding:15px;
32. margin-top:7px;
33. text-align:center;
34. }
35.
36. @media only screen and (max-width:620px) {
37. /* For mobile phones: */
38. .menu, .main, .right {
39. width:100%;
40. }
41. }
42. </style>
43. </head>
44. <body style="font-family:Verdana;color:#aaaaaa;">
45.
46. <div style="background-color:#e5e5e5;padding:15px;text-align:center;">
47. <h1>Hello World</h1>
48. </div>
49.
50. <div style="overflow:auto">
51. <div class="menu">
52. <a href="#">Link 1</a>
53. <a href="#">Link 2</a>
54. <a href="#">Link 3</a>
55. <a href="#">Link 4</a>
56. </div>
57.
58. <div class="main">
59. <h2>Lorum Ipsum</h2>
60. <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</p>
61. </div>
62.
63. <div class="right">
64. <h2>About</h2>
65. <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
66. </div>
67. </div>
68.
69. <div style="background-color:#e5e5e5;text-align:center;padding:10px;margin-top:7px;">♦ copyright w3schools.com</div>
70.
71. </body>
72. </html>

```

## [خودتان امتحان کنید <<](#)

## استفاده از فریم ورک های آماده

بهترین و سریع ترین راه استفاده از فریم ورک هایی است، که از قبل بصورت responsive آماده شده است، که مهمترین آن ها فریم ورک های W3.CSS و Bootstrap می باشد.

### مثال : فریم ورک w3

```
1. <!DOCTYPE html>
2. <html>
3. <meta name="viewport" content="width=device-width, initial-scale=1">
4. <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
5. <body>
6.
7. <div class="w3-container w3-green">
8.   <h1>W3Schools Demo</h1>
9.   <p>Resize this responsive page!</p>
10. </div>
11.
12. <div class="w3-row-padding">
13.   <div class="w3-third">
14.     <h2>London</h2>
15.     <p>London is the capital city of England.</p>
16.     <p>It is the most populous city in the United Kingdom,
17.     with a metropolitan area of over 13 million inhabitants.</p>
18.   </div>
19.
20.   <div class="w3-third">
21.     <h2>Paris</h2>
22.     <p>Paris is the capital of France.</p>
23.     <p>The Paris area is one of the largest population centers in Europe,
24.     with more than 12 million inhabitants.</p>
25.   </div>
26.
27.   <div class="w3-third">
28.     <h2>Tokyo</h2>
29.     <p>Tokyo is the capital of Japan.</p>
30.     <p>It is the center of the Greater Tokyo Area,
31.     and the most populous metropolitan area in the world.</p>
32.   </div>
33. </div>
34.
35. </body>
36. </html>
```

### [خودتان امتحان کنید<>](#)

## فریم ورک Bootstrap

Bootstrap فریم ورک مشهور دیگری است که برای طراحی responsive قالب سایت ها استفاده می شود.

### مثال : فریم ورک Bootstrap

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4. <title>Bootstrap Example</title>
5. <meta charset="utf-8">
```

```
6. <meta name="viewport" content="width=device-width, initial-scale=1">
7. <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
8. <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js"></script>
9. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
10. </head>
11. <body>
12.
13. <div class="container">
14.   <div class="jumbotron">
15.     <h1>My First Bootstrap Page</h1>
16.   </div>
17.   <div class="row">
18.     <div class="col-sm-4">
19.       ...
20.     </div>
21.     <div class="col-sm-4">
22.       ...
23.     </div>
24.     <div class="col-sm-4">
25.       ...
26.     </div>
27.   </div>
28. </div>
29.
30. </body>
31. </html>
```

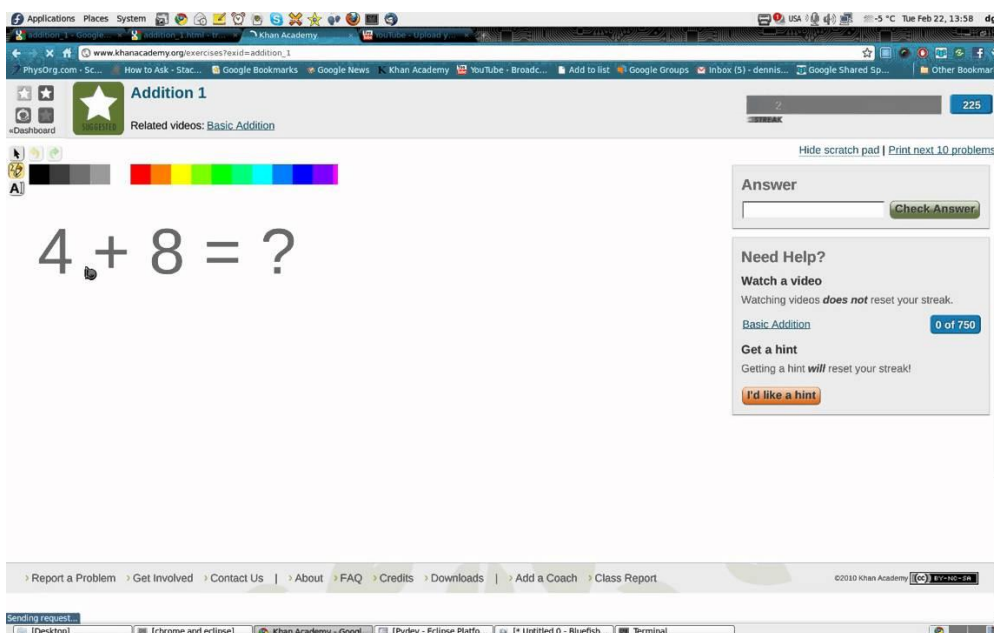
[خودتان امتحان کنید<<](#)

## جلسه ۲۷ : کدهای کامپیوتری در html

### مقدمه

با عرض سلام خدمت کاربران سایت [پی وی لرن](#) . در آموزشهایی که تا کنون در دوره html ارائه کردیم، با مباحث مهمی از جمله [ویژگی Responsive در html](#) ، آشنا شدید. در ادامه این آموزشها ، و در این بخش قصد داریم به بحث [نمایش کدهای کامپیوتری در html](#) ، پردازیم. برای نمایش کد در html و دستورات ، تگ هایی تعریف شده است که هر کدام در مواردی خاصی کاربرد دارد. این تگ ها بیشتر برای وارد کردن محتوای دستوری ، علمی و آموزشی که شامل فرمول های ریاضی و ... هستند بکار می رود.

در ادامه این بخش به موضوع نمایش کدهای کامپیوتر در html ، می پردازیم.



img

### نمایش کدهای کامپیوتری در html

برای نمایش کدهای ساده مانند جمع و تفریق و... می توانید از تگ `<code>` استفاده کنید.

مثال : نمایش کدها

1. `<code>`
2. `x = 5;<br>`
3. `y = 6;<br>`
4. `z = x + y;`
5. `</code>`

[خودتان امتحان کنید<<](#)

## تگ <kbd>

از این تگ برای نمایش کلید های ترکیبی صفحه کلید استفاده می شود.

مثال:

مثال : تگ kbd

1. `<p>Save the document by pressing <kbd>Ctrl + S</kbd></p>`

## [خودتان امتحان کنید<<](#)

### خروجی برنامه ها

تگ <samp> ، برای نمایش خروجی برنامه ها و نرم افزار ها ، استفاده می شود.

مثال : تگ samp

1. `<p>If you input wrong value, the program will return <samp>Error!</samp></p>`

## [خودتان امتحان کنید<<](#)

### استفاده از <code> در HTML

عنصر <code> یک قطعه کد کامپیوتر را تعریف می کند، متنی که توسط برچسب های <code> احاطه شده است معمولا در فونت یکپارچه پیش فرض مرورگر نمایش داده می شود:

مثال : کدهای کامپیوتر

1. `<code>`
2. `x = 5;`
3. `y = 6;`
4. `z = x + y;`
5. `</code>`

## [خودتان امتحان کنید<<](#)

توجه داشته باشید که عنصر <code> فضای خالی اضافی و خط فاصله را حفظ نمی کند. برای رفع این، شما می توانید عنصر <code> را داخل عنصر <pre> قرار دهید:

مثال : تگ pre

1. `<pre>`
2. `<code>`
3. `x = 5;`
4. `y = 6;`
5. `z = x + y;`
6. `</code>`
7. `</pre>`

## توابع و فرمول ها

از تگ <var> نیز برای نمایش تابع و فرمول ها استفاده می شود.

مثال : استفاده از <var> برای تعریف متغیر

1. Einstein wrote: `<var>E</var> = <var>mc</var><sup>2</sup>`.

## عناصر نمایش کدها و دستورات در html

توضیحات	تگ
کدهای برنامه نویسی را مشخص می کند.	<code>
ورودی صفحه کلید را مشخص می کند.	<kbd>
خروجی کامپیوتری را مشخص می کند.	<samp>
برای تعریف متغیر کاربرد دارد.	<var>
متن را به همان شکلی که تایپ شده نمایش می دهد.	<pre>

## کلام آخر

در محتواهای علمی و خصوصا آموزشی در صفحات وب گاهی از کدهای برنامه نویسی ، کامپیوتری و ... استفاده می شود که برای درج **کدهای کامپیوتری در html** نیاز است که با استفاده از تگ های خاصی که در مباحث فوق به آن اشاره کردیم این کدها را تعریف کنید.



### مقدمه

با عرض سلام و وقت بخیر خدمت کاربران سایت [پی وی لرن](#) . و کاربرانی که [دوره کامل آموزش html](#) را دنبال می کنند. موجودیت یا **entities در html** یکسری نمادها و علائم خاصی هستند که از صفحه کلید سیستم شما در دسترس نمی باشند. این نمادها در نمایش مواردی از جمله عملگرهای منطقی و محاسباتی، نماد واحدهای مختلف ارز، نماد برخی واحدهای اندازه گیری و... کاربرد دارند. از این رو برای نمایش اینگونه نمادها در محتوای وب خود باید با چگونگی درج آن در بخش مورد نظر آشنایی داشته باشید. در ادامه ی این آموزش **موجودیت ها در html** ما به شما چگونگی درج اینگونه نمادها در محتوای صفحات وب شما را آموزش خواهیم داد.

### موجودیت ها در html

در ادامه ی این مبحث مواردی از جمله ساختار درج entities در html و طرز استفاده از آن را بررسی کرده ایم. همچنین در انتهای مبحث برخی از پرکاربردترین موجودیت ها در html را برای شما لیست کرده ایم.

### html در entities

برخی از کاراکترها در HTML ذخیره می شوند.

اگر از متن شما کمتر از (>) یا بزرگتر از (<) استفاده کنید، مرورگر ممکن است با برچسب ها آن ها را مخلوط کند.

نهاد های شخصیتی برای نمایش شخصیت های ذخیره شده در HTML استفاده می شوند.

یک کاراکتر موجودیت در html به صورت زیر است:

مثال :

1. &entity\_name;
2. OR
- 3.
4. &#entity\_number;

برای نمایش کاراکتر کمتر با استفاده از علامت (>) ما باید بنویسیم: **&lt;** یا **&#60**

مزیت استفاده از یک نام موجودیت: نام موجودیت ساده بوده و به راحتی قابل یادآوری است.

معایب استفاده از یک نام موجودیت : مرورگرها ممکن است از تمام نامهای موجودیت پشتیبانی نکنند، اما از موجودیت های اعداد به خوبی پشتیبانی می کنند.

### فضای فاصله

کاراکتر موجودیتی که معمولا برای ایجاد فضای فاصله در عناصر استفاده می شود کاراکتر **&nbsp;** است.

کاراکتر فوق یک فضای خالی ایجاد می کند بدون اینکه متن شما را به خط بعدی منتقل کند.

استفاده از کاراکتر فوق ، باعث ایجاد یک فضای فاصله بین کلمات و افزایش خوانایی متن می شود.

مثال ها:

- ۱۰ §
- km/h ۱۰
- PM ۱۰

همچنین اگر شما در متن خود ۱۰ فاصله قرار دهید، مرورگر ۹ مورد از آنها را حذف خواهد کرد. برای اضافه کردن فضاهای واقعی به متن خود، می‌توانید از **&nbsp;** استفاده کنید.

همچنین کاراکتر **&#2098** به شما اجازه می‌دهد تا از یک کاراکتر خط فاصله (-) در متن خود استفاده کنید.

برخی از پرکاربردترین موجودیت‌ها در html :

شماره موجودیت	نام موجودیت	توضیحات	نتایج
	<code>&amp;nbsp;</code>	non-breaking space	
>	<code>&amp;lt;</code>	less than	>
<	<code>&amp;gt;</code>	greater than	<
&	<code>&amp;amp;</code>	ampersand	&
“	<code>&amp;quot;</code>	double quotation mark	“
‘	<code>&amp;apos;</code>	single quotation mark (apostrophe)	‘
¢	<code>&amp;cent;</code>	cent	¢

£	pound;&	pound	£
¥	yen;&	yen	¥
€	euro;&	euro	€
©	copy;&	copyright	©
®	reg;&	registered trademark	®

نکته : نامهای موجودیت حساس به حروف هستند.

## ترکیب علائم دیاکریتیک

یک علامت دیاکریتیک یک "گلیف" است که به یک حرف اضافه شده است. برخی علامت های دیاکریتیک، مانند grave ( ` ) و acute ( ´ )، لهجه می شوند.

علامت های دیاکریتیک می تواند در هر دو قسمت بالا و پایین یک نامه، داخل یک حرف و بین دو حرف ظاهر شود.

علامت های دیاکریتیک می تواند در ترکیب با موجودیت های عددی برای تولید موجودیت که در مجموعه موجودیت (کدینگ) مورد استفاده در صفحه وجود ندارد استفاده شود.

برخی از علائم دیاکریتیک به شرح زیر است:

علامت	کاراکتر	ساختار	نتیجه
`	a	à	à
´	a	á	á

ā	ā	a	^
ã	ã	a	~
o	o	o	`
o	o	o	˘
o	o	o	^
o	o	o	~

























## کلام آخر

در اغلب مواردی که شما نیاز به درج مطالب علمی، پژوهشی و... در وب سایت خود دارید نیاز به درج برخی کاراکترهای خاص نظیر کاراکترهای عملگرهای مختلف منطقی و محاسباتی، نمادهای ارزی و... پیدا می کنید. در اینگونه موارد شما می توانید از **کاراکترهای entities در html** استفاده کنید، از این رو مباحث این بخش را به آشنایی با **موجودیت ها در html** اختصاص دادیم.

### مقدمه

با عرض سلام خدمت کاربران سایت [پی وی لرن](#) . در مطالب آموزشی قبلی ، در مورد مبحث [نمایش کدها و دستورات در html](#) ، توضیحاتی ارائه کردیم. و با کاربرد تگ های مختلف در [نمایش کدها و فرمول ها](#) در صفحات وب آشنا شدید. در ادامه در این بخش ، به آموزش [چگونگی نمایش نمادها در html](#) یا [نمایش symbol ها در html و نمادهای خاص در html](#) ، می پردازیم. بسیاری از نمادهای ریاضی ، تکنیکی ، واحد پول و ... در حالت عادی با کیبورد قابل نمایش نیستند. برای نمایش نماد های خاص در html باید از نام **entity** هر کدام از نمادها ، برای نمایش آنها استفاده کنید.

در ادامه ی این بخش آموزشی به موضوع آموزش چگونگی نمایش نمادها در html ، می پردازیم.

Symbol	HTML code	Symbol	HTML code
 4 Tear-drop asterisk	&#10018;	 4 Balloon asterisk	&#10019;
 Club asterisk	&#10021;	 4 Balloon asterisk heavy	&#10020;
 4-pointed star black	&#10022;	 4-pointed star white	&#10023;
 Stress outlined star	&#10025;	 Circled white star	&#10026;
 Open center black star	&#10027;	 Black center white star	&#10028;
 Black star outlined	&#10029;	 Black star outlined heavy	&#10030;
 Pinwheel star	&#10031;	 White star shadowed	&#10032;
 Asterisk heavy	&#10033;	 Open center asterisk	&#10034;
 8-spoked asterisk	&#10035;	 8-pointed black star	&#10036;
 8-pointed pinwheel star	&#10037;	 6-pointed black star	&#10038;
 8-p. rectilinear	&#10039;	 8-p. rectilinear heavy	&#10040;
 12-p. black star	&#10041;	 16-p. asterisk	&#10042;

symbols

### نمایش نمادها در html

همانطور که اشاره شد برای نمایش هر نماد باید از نام entity ، آن نماد استفاده کنید. اما اگر نام entity نماد مورد نظر در دسترس نبود. می توانید از شماره entity ، که بصورت دهدهی یا مبنای ۱۶ تعریف شده است استفاده کنید.

مثال : مثالی از درج نمادهای خاص در HTML

1. `<p>I will display &euro;</p>`
2. `<p>I will display €</p>`
3. `<p>I will display €</p>`

### [خودتان امتحان کنید<<](#)

نتیجه:

مثال :

1. I will display €  
www.pvlearn.com

2. I will display €
3. I will display €

برخی از نمادهای ریاضی که در html پشتیبانی می شود:

Entity	شماره	کاراکتر
forall;&	$\forall$	$\forall$
part;&	$\partial$	$\partial$
exist;&	$\exists$	$\exists$
empty;&	$\emptyset$	$\emptyset$
nabla;&	$\nabla$	$\nabla$
isin;&	$\ni$	$\ni$
notin;&	$\not\in$	$\not\in$
ni;&	$\in$	$\in$
prod;&	$\prod$	$\prod$

sum;&	$\Sigma$	$\Sigma$
-------	----------	----------

برخی از حروف خاص که در html پشتیبانی می شود:

Entity	شماره	کاراکتر
Alpha;&	A	A
Beta;&	B	B
Gamma;&	$\Gamma$	$\Gamma$
Delta;&	$\Delta$	$\Delta$
Epsilon;&	E	E
Zeta;&	Z	Z

سایر نمادها ی خاصی که در html پشتیبانی می شوند:

Entity	شماره	کاراکتر
copy;&	©	©

reg;&	®	®
euro;&	€	€
trade;&	™	™
larr;&	←	←
uarr;&	↑	↑
rarr;&	→	→
darr;&	↓	↓
spades;&	♠	♠
clubs;&	♣	♣
hearts;&	♥	♥
diams;&	♦	♦

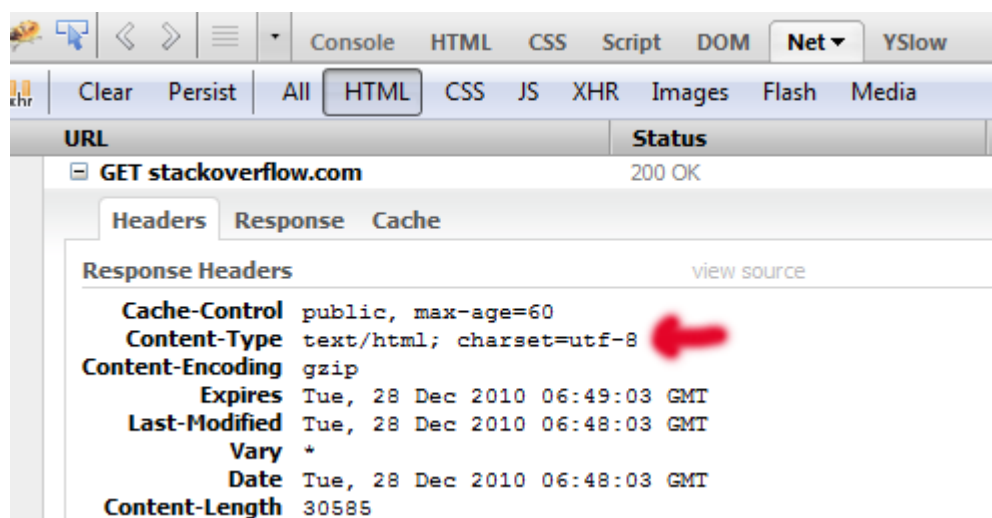


## جلسه ۳۰ : ویژگی Encoding در html ( تنظیمات کاراکترها )

### مقدمه

با عرض سلام خدمت کاربران سایت [پی وی لرن](#) . در مطالب آموزشی که تاکنون ارائه شده ، در موضوع [نمایش کاراکترها و نماد های خاص در html](#) ، به ارائه موضوعاتی در این زمینه پرداختیم. اکنون در این بخش نیز در ادامه موضوعاتی که به آنها اشاره کردیم ، قصد داریم به موضوع آموزش **کار با ویژگی Encoding در html** ( تنظیمات کاراکتر در html ) و کاربرد Encoding در html ، بپردازیم.

برای نمایش صحیح هر وب سائیتی ، مرورگر باید تنظیمات کاراکتر و **Encoding** آن صفحه وب را پشتیبانی کند، تا بتواند از آن استفاده کند. در ادامه آموزش در این بخش ، به ارائه بحث آموزش کار با ویژگی Encoding در html ( تنظیمات کاراکتر ) ، می پردازیم.



diSbx

### کار با ویژگی Encoding در html ( تنظیمات کاراکتر )

**ASCII** (که معمولا character set نامیده می شود) اولین Encoding استاندارد است. ASCII دارای ۱۲۸ نوع کاراکتر از نوع alphanumeric است، که می تواند در اینترنت استفاده شود: اعداد (۰-۹) ، حروف انگلیسی (A-Z) ، و برخی از کاراکترهای خاص نظیر ! \$ % & ' ( ) \* + , - . / : ; < > @ [ \ ] ^ \_ ` { | } ~ .

**ANSI** نیز استاندارد دیگری است که از ۲۵۶ کاراکتر پشتیبانی می کند ، اما استاندارد **UTF-8** تقریبا از تمام کارکترهای جهان ، پشتیبانی می کند.

### صفت charset در html

مرورگرهای با استفاده از تگ <meta> ، تنظیمات کارکتری هر وب سایت را دریافت می کنند.

مثال: در: html4:

مثال :

```
1. <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
```

## مثال :

1. `<meta charset="UTF-8">`

نکته: اگر یک مرورگر استاندارد ISO-8859-1 را در یک صفحه وب شناسایی کند، آن را به صورت ANSI پیش فرض می کند، زیرا ANSI همانند ISO-8859-1 است.

## تفاوت بین مجموعه های کاراکتر

توضیحات	UTF-8	۸۸۵۹	ANSI	ASCII	Numb
space					۳۲
exclamation mark	!	!	!	!	۳۳
quotation mark	"	"	"	"	۳۴
number sign	#	#	#	#	۳۵
dollar sign	\$	\$	\$	\$	۳۶
percent sign	%	%	%	%	۳۷

ampersand	&	&	&	&	۳۸
apostrophe	'	'	'	'	۳۹
left parenthesis	)	)	)	)	۴۰
right parenthesis	(	(	(	(	۴۱
asterisk	*	*	*	*	۴۲
plus sign	+	+	+	+	۴۳
comma	,	,	,	,	۴۴
hyphen-minus	-	-	-	-	۴۵
full stop	.	.	.	.	۴۶
solidus	/	/	/	/	۴۷
digit zero	۰	۰	۰	۰	۴۸

digit one	۱	۱	۱	۱	۴۹
digit two	۲	۲	۲	۲	۵۰
digit three	۳	۳	۳	۳	۵۱
digit four	۴	۴	۴	۴	۵۲
digit five	۵	۵	۵	۵	۵۳
digit six	۶	۶	۶	۶	۵۴
digit seven	۷	۷	۷	۷	۵۵
digit eight	۸	۸	۸	۸	۵۶
digit nine	۹	۹	۹	۹	۵۷
colon	:	:	:	:	۵۸
semicolon	;	;	;	;	۵۹

less-than sign	>	>	>	>	۶۰
equals sign	=	=	=	=	۶۱
greater-than sign	<	<	<	<	۶۲
question mark	?	?	?	?	۶۳
commercial at	@	@	@	@	۶۴
Latin capital letter A	A	A	A	A	۶۵
Latin capital letter B	B	B	B	B	۶۶
Latin capital letter C	C	C	C	C	۶۷
Latin capital letter D	D	D	D	D	۶۸
Latin capital letter E	E	E	E	E	۶۹
Latin capital letter F	F	F	F	F	۷۰

Latin capital letter G	G	G	G	G	٧١
Latin capital letter H	H	H	H	H	٧٢
Latin capital letter I	I	I	I	I	٧٣
Latin capital letter J	J	J	J	J	٧٤
Latin capital letter K	K	K	K	K	٧٥
Latin capital letter L	L	L	L	L	٧٦
Latin capital letter M	M	M	M	M	٧٧
Latin capital letter N	N	N	N	N	٧٨
Latin capital letter O	O	O	O	O	٧٩
Latin capital letter P	P	P	P	P	٨٠
Latin capital letter Q	Q	Q	Q	Q	٨١

Latin capital letter R	R	R	R	R	۸۲
Latin capital letter S	S	S	S	S	۸۳
Latin capital letter T	T	T	T	T	۸۴
Latin capital letter U	U	U	U	U	۸۵
Latin capital letter V	V	V	V	V	۸۶
Latin capital letter W	W	W	W	W	۸۷
Latin capital letter X	X	X	X	X	۸۸
Latin capital letter Y	Y	Y	Y	Y	۸۹
Latin capital letter Z	Z	Z	Z	Z	۹۰
left square bracket	]	]	]	]	۹۱
reverse solidus	\	\	\	\	۹۲

right square bracket	[	[	[	[	۹۳
circumflex accent	^	^	^	^	۹۴
low line	_	_	_	_	۹۵
grave accent					۹۶
Latin small letter a	a	a	a	a	۹۷
Latin small letter b	b	b	b	b	۹۸
Latin small letter c	c	c	c	c	۹۹
Latin small letter d	d	d	d	d	۱۰۰
Latin small letter e	e	e	e	e	۱۰۱
Latin small letter f	f	f	f	f	۱۰۲
Latin small letter g	g	g	g	g	۱۰۳
Latin small letter h	h	h	h	h	۱۰۴
Latin small letter i	i	i	i	i	۱۰۵
Latin small letter j	j	j	j	j	۱۰۶
Latin small letter k	k	k	k	k	۱۰۷



Latin small letter l	l	l	l	l	۱۰۸
Latin small letter m	m	m	m	m	۱۰۹
Latin small letter n	n	n	n	n	۱۱۰
Latin small letter o	o	o	o	o	۱۱۱
Latin small letter p	p	p	p	p	۱۱۲
Latin small letter q	q	q	q	q	۱۱۳
Latin small letter r	r	r	r	r	۱۱۴
Latin small letter s	s	s	s	s	۱۱۵
Latin small letter t	t	t	t	t	۱۱۶
Latin small letter u	u	u	u	u	۱۱۷
Latin small letter v	v	v	v	v	۱۱۸
Latin small letter w	w	w	w	w	۱۱۹
Latin small letter x	x	x	x	x	۱۲۰
Latin small letter y	y	y	y	y	۱۲۱
Latin small letter z	z	z	z	z	۱۲۲
left curly bracket	}	}	}	}	۱۲۳
vertical line					۱۲۴
right curly bracket	{	{	{	{	۱۲۵

tilde	~	~	~	~	۱۲۶
				DEL	۱۲۷
euro sign			€		۱۲۸
NOT USED					۱۲۹
single low-9 quotation mark			,		۱۳۰
Latin small letter f with hook			f		۱۳۱
double low-9 quotation mark			”		۱۳۲
horizontal ellipsis			...		۱۳۳
dagger			†		۱۳۴
double dagger			‡		۱۳۵
modifier letter circumflex accent			^		۱۳۶
per mille sign			‰		۱۳۷
Latin capital letter S with caron			Š		۱۳۸
single left-pointing angle quotation mark			›		۱۳۹
Latin capital ligature OE			Œ		۱۴۰
NOT USED					۱۴۱
Latin capital letter Z with caron			Ž		۱۴۲
NOT USED					۱۴۳

NOT USED			۱۴۴
left single quotation mark		'	۱۴۵
right single quotation mark		'	۱۴۶
left double quotation mark		"	۱۴۷
right double quotation mark		"	۱۴۸
bullet		•	۱۴۹
en dash		–	۱۵۰
em dash		—	۱۵۱
small tilde		~	۱۵۲
trade mark sign		™	۱۵۳
Latin small letter s with caron		š	۱۵۴
single right-pointing angle quotation mark		<	۱۵۵
Latin small ligature oe		œ	۱۵۶
NOT USED			۱۵۷
Latin small letter z with caron		ž	۱۵۸
Latin capital letter Y with diaeresis		ÿ	۱۵۹
no-break space			۱۶۰
inverted exclamation mark	¡	¡	۱۶۱

cent sign	¢	¢	¢	۱۶۲
pound sign	£	£	£	۱۶۳
currency sign	¤	¤	¤	۱۶۴
yen sign	¥	¥	¥	۱۶۵
broken bar				۱۶۶
section sign	§	§	§	۱۶۷
diaeresis	¨	¨	¨	۱۶۸
copyright sign	©	©	©	۱۶۹
feminine ordinal indicator	a	a	a	۱۷۰
left-pointing double angle quotation mark	»	»	»	۱۷۱
not sign	¬	¬	¬	۱۷۲
soft hyphen				۱۷۳
registered sign	®	®	®	۱۷۴
macron	¯	¯	¯	۱۷۵
degree sign	°	°	°	۱۷۶
plus-minus sign	±	±	±	۱۷۷
superscript two	²	²	²	۱۷۸
superscript three	³	³	³	۱۷۹

acute accent	´	´	´	۱۸۰
micro sign	μ	μ	μ	۱۸۱
pilcrow sign	¶	¶	¶	۱۸۲
middle dot	·	·	·	۱۸۳
cedilla	¸	¸	¸	۱۸۴
superscript one	¹	¹	¹	۱۸۵
masculine ordinal indicator	º	º	º	۱۸۶
right-pointing double angle quotation mark	»	»	»	۱۸۷
vulgar fraction one quarter	¼	¼	¼	۱۸۸
vulgar fraction one half	½	½	½	۱۸۹
vulgar fraction three quarters	¾	¾	¾	۱۹۰
inverted question mark	¿	¿	¿	۱۹۱
Latin capital letter A with grave	À	À	À	۱۹۲
Latin capital letter A with acute	Á	Á	Á	۱۹۳
Latin capital letter A with circumflex	Â	Â	Â	۱۹۴
Latin capital letter A with tilde	Ã	Ã	Ã	۱۹۵
Latin capital letter A with diaeresis	Ä	Ä	Ä	۱۹۶
Latin capital letter A with ring above	Å	Å	Å	۱۹۷

Latin capital letter AE	Æ	Æ	Æ	۱۹۸
Latin capital letter C with cedilla	Ç	Ç	Ç	۱۹۹
Latin capital letter E with grave	È	È	È	۲۰۰
Latin capital letter E with acute	É	É	É	۲۰۱
Latin capital letter E with circumflex	Ê	Ê	Ê	۲۰۲
Latin capital letter E with diaeresis	Ë	Ë	Ë	۲۰۳
Latin capital letter I with grave	Ì	Ì	Ì	۲۰۴
Latin capital letter I with acute	Í	Í	Í	۲۰۵
Latin capital letter I with circumflex	Î	Î	Î	۲۰۶
Latin capital letter I with diaeresis	Ï	Ï	Ï	۲۰۷
Latin capital letter Eth	Ð	Ð	Ð	۲۰۸
Latin capital letter N with tilde	Ñ	Ñ	Ñ	۲۰۹
Latin capital letter O with grave	Ò	Ò	Ò	۲۱۰
Latin capital letter O with acute	Ó	Ó	Ó	۲۱۱
Latin capital letter O with circumflex	Ô	Ô	Ô	۲۱۲
Latin capital letter O with tilde	Õ	Õ	Õ	۲۱۳
Latin capital letter O with diaeresis	Ö	Ö	Ö	۲۱۴
multiplication sign	×	×	×	۲۱۵

Latin capital letter O with stroke	Ø	Ø	Ø	۲۱۶
Latin capital letter U with grave	Ù	Ù	Ù	۲۱۷
Latin capital letter U with acute	Ú	Ú	Ú	۲۱۸
Latin capital letter U with circumflex	Û	Û	Û	۲۱۹
Latin capital letter U with diaeresis	Ü	Ü	Ü	۲۲۰
Latin capital letter Y with acute	Ý	Ý	Ý	۲۲۱
Latin capital letter Thorn	ƀ	ƀ	ƀ	۲۲۲
Latin small letter sharp s	ß	ß	ß	۲۲۳
Latin small letter a with grave	à	à	à	۲۲۴
Latin small letter a with acute	á	á	á	۲۲۵
Latin small letter a with circumflex	â	â	â	۲۲۶
Latin small letter a with tilde	ã	ã	ã	۲۲۷
Latin small letter a with diaeresis	ä	ä	ä	۲۲۸
Latin small letter a with ring above	å	å	å	۲۲۹
Latin small letter ae	æ	æ	æ	۲۳۰
Latin small letter c with cedilla	ç	ç	ç	۲۳۱
Latin small letter e with grave	è	è	è	۲۳۲
Latin small letter e with acute	é	é	é	۲۳۳

Latin small letter e with circumflex	ê	ê	ê	۲۳۴
Latin small letter e with diaeresis	ë	ë	ë	۲۳۵
Latin small letter i with grave	ì	ì	ì	۲۳۶
Latin small letter i with acute	í	í	í	۲۳۷
Latin small letter i with circumflex	î	î	î	۲۳۸
Latin small letter i with diaeresis	ï	ï	ï	۲۳۹
Latin small letter eth	ð	ð	ð	۲۴۰
Latin small letter n with tilde	ñ	ñ	ñ	۲۴۱
Latin small letter o with grave	ò	ò	ò	۲۴۲
Latin small letter o with acute	ó	ó	ó	۲۴۳
Latin small letter o with circumflex	ô	ô	ô	۲۴۴
Latin small letter o with tilde	õ	õ	õ	۲۴۵
Latin small letter o with diaeresis	ö	ö	ö	۲۴۶
division sign	÷	÷	÷	۲۴۷
Latin small letter o with stroke	ø	ø	ø	۲۴۸
Latin small letter u with grave	ù	ù	ù	۲۴۹
Latin small letter u with acute	ú	ú	ú	۲۵۰
Latin small letter with circumflex	û	û	û	۲۵۱



Latin small letter u with diaeresis	ü	ü	ü	۲۵۲
Latin small letter y with acute	ý	ý	ý	۲۵۳
Latin small letter thorn	þ	þ	þ	۲۵۴
Latin small letter y with diaeresis	ÿ	ÿ	ÿ	۲۵۵

## کاراکترهای ASCII

ASCII از مقادیر ۰ تا ۳۱ (و ۱۲۷) برای کاراکترهای کنترل استفاده می کند.

ASCII از مقادیر ۳۲ تا ۱۲۶ برای حروف، ارقام و نمادها استفاده می کند.

ASCII از مقادیر ۱۲۸ تا ۲۵۵ استفاده نمی کند.

## ANSI کاراکترهای

ANSI برای مقادیر ۰ تا ۱۲۷ برابر با ASCII است.

ANSI مجموعه ای از کاراکتر اختصاصی برای مقادیر ۱۲۸ تا ۱۵۹ دارد.

ANSI برای مقادیر ۱۶۰ تا ۲۵۵ برابر با UTF-8 است.

## کاراکترهای استاندارد ISO-8859-1

ISO-8859-1 برای مقادیر ۰ تا ۱۲۷ برابر با ASCII است.

ISO-8859-1 از مقادیر ۱۲۸ تا ۱۵۹ استفاده نمی کند.

ISO-8859-1 برای مقادیر ۱۶۰ تا ۲۵۵ برابر با UTF-8 است.

## کاراکترهای UTF-8

UTF-8 برای مقادیر ۰ تا ۱۲۷ برابر با ASCII است.

UTF-8 از مقادیر ۱۲۸ تا ۱۵۹ استفاده نمی کند.

UTF-8 برای مقادیر ۱۶۰ تا ۲۵۵ برابر با هر دو ANSI و ISO-8859-1 است.

UTF-8 همچنان از مقدار ۲۵۶ با بیش از ۱۰ هزار کاراکتر متفاوت ادامه می یابد.

## تعریف استاندارد Encoding، در CSS

شما می توانید از خصوصیت @charset در CSS، برای تعریف Encoding، استفاده کنید.

مثال :

1. @charset "UTF-8";

## کلام آخر

در این مبحث طرز کار با ویژگی **Encoding در html** و صفت charset در html را ارائه کردیم. تا با انواع استانداردهای زبان و کاراکتری در html و نحوه تنظیم آن ها آشنا شوید.

## جلسه ۳۱ : منابع مکانی در html

### مقدمه

با عرض سلام و وقت بخیر خدمت کاربران سایت **پی وی لرن** . و کاربرانی که **دوره کامل آموزش html** را دنبال می کنند. **منابع مکانی** در html یا **URL** یک کلمه دیگر برای تعریف یک آدرس وب است. یک URL می تواند از یک عبارت (مانند [www.pvlearn.com](http://www.pvlearn.com)) یا یک آدرس آی پی (مانند ۱۹۲/۶۸/۲۰/۵۰) تشکیل شده باشد. البته تمام آدرس های اینترنتی که با یک عبارت مشخص شده در واقع یک آدرس ip است. اما به دلیل سخت بودن به خاطر سپردن اعداد ip و برای راحتی کاربران بجای آدرس ip از یک عبارت استفاده می شود. که هر دو روشهای دسترسی به آدرس های وب است. برای آشنایی بیشتر با مباحث **منابع مکانی در html** در ادامه ی مبحث با ما همراه باشید.

### منابع مکانی در html

در ادامه ی مبحث شما را با URL که یکی از موارد منابع مکانی در html می باشد آشنا خواهیم کرد.

#### URL

مرورگرهای وب با استفاده از یک URL از صفحات وب استفاده می کنند.

یک منبع ( URL برای آدرس سندی (یا سایر داده ها) در وب استفاده می شود.

یک آدرس وب مانند <http://www.pvlearn.com> از ساختار زیر تشکیل شده است:

مثال :

1. scheme://prefix.domain:port/path/filename

تشریح ساختار فوق:

- **scheme** – نوع سرویس اینترنت را تعریف می کند (شایع ترین http یا https است).
- **prefix** – تعریف یک پیشوند دامنه (به طور پیش فرض برای http است [www](http://www)).
- **domain** – نام دامنه اینترنت را تعریف می کند (مانند [pvlearn.com](http://www.pvlearn.com)).
- **port** – شماره پورت را در میزبان تعریف می کند (به طور پیش فرض برای http 80 است).
- **path** – مسیری را در سرور تعریف می کند (اگر حذف شده باشد: دایرکتوری ریشه ای از سایت).
- **filename** – نام سند یا منبع را تعریف می کند.

## طرح های URL

جدول زیر شامل لیستی از طرح های عمومی URL است:

کاربرد	مخفف شده	طرح
Common web pages. Not encrypted	HyperText Transfer Protocol	http
Secure web pages. Encrypted	Secure HyperText Transfer Protocol	https
Downloading or uploading files	File Transfer Protocol	ftp
A file on your computer		file

## رمزگذاری URL

رمزگذاری URL، کاراکترهای غیر ASCII را به یک فرمت تبدیل می کند که می تواند از طریق اینترنت منتقل شود.

رمزگذاری URL جایگزین نویسه های غیر ASCII با "%" و بعد از رقم های هگزادسیمال می شود.

نشانی های اینترنتی نمی توانند فضایی داشته باشند رمزگذاری URL معمولا یک فضای با علامت + (+) یا %۲۰ جایگزین می کند.

## نمونه هایی از رمزنگاری ASCII

مرورگر شما ورودی را طبق کاراکتر مورد استفاده در صفحه شما رمزگذاری می کند.

پیش فرض character-set در HTML5 استاندارد UTF-8 است.

From UTF-8	From Windows-1252	Character
E2%82%AC%	۸۰%	€

C2%A3%	A3%	£
C2%A9%	A9%	©
C2%AE%	AE%	®
C3%80%	C0%	À
C3%81%	C1%	Á
C3%82%	C2%	Â
C3%83%	C3%	Ã
C3%84%	C4%	Ä
C3%85%	C5%	Å

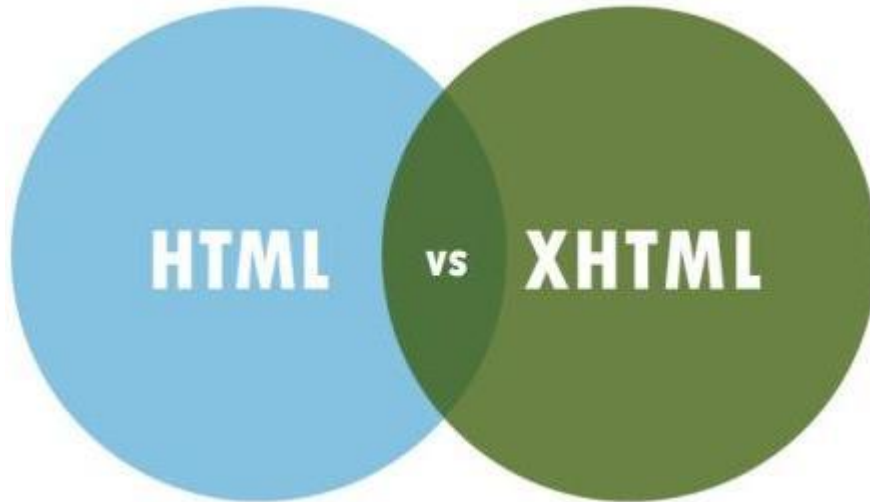
## کلام آخر

دسترسی به وب سایت های مختلف با روش های گوناگونی از جمله تایپ URL انجام می شود. از این رو ما سعی کردیم در این مبحث شما را با منابع مکانی در **html** و ساختار URL آشنا کنیم.

### مقدمه

با عرض سلام خدمت کاربران سایت **پی وی لرن** . در موضوعات آموزشی اخیر ، در مورد بحث **نمایش کاراکترها** و **مبحث Encode** ، توضیحاتی ارائه شد. در این بخش از آموزشهای html ، در این مطلب قصد داریم به **مبحث بررسی ویژگی های HTML و XHTML** ، **مزایا و معایب و تفاوت های** هر یک بپردازیم. HTML و XHTML هر دو با XML نوشته شده اند. و در اکثر ویژگی ها و ساختار شبیه هم هستند.

در ادامه ی این بخش ، به موضوع **بررسی ویژگی های HTML و XHTML** ، می پردازیم.



XHTML

### بررسی ویژگی های HTML و XHTML

ابتدا به معرفی ویژگی های XHTML ، می پردازیم

### XHTML چیست

- XHTML مخفف عبارت **EXtensible HyperText Markup Language** می باشد.
- XHTML ، تقریبا مشابه HTML است.
- XHTML از HTML ، سخت تر است.
- HTML و XHTML به عنوان برنامه XML معرفی شده اند.
- XHTML در تمام مرورگر های بزرگ پشتیبانی می شود.

### دلایل استفاده از XHTML

بسیاری از صفحات اینترنتی حاوی محتوای بد HTML هستند.

این کد HTML در اغلب مرورگرها خوب کار می‌کند:

مثال :

```
1. <html>
2. <head>
3.   <title>This is bad HTML</title>
4.
5. <body>
6.   <h1>Bad HTML
7.   <p>This is a paragraph
8. </body>
```

XML یک زبان نشانه گذاری است که در آن اسناد باید به درستی مشخص شوند.

XHTML با ترکیب نقاط قوت HTML و XML توسعه داده شد.

XHTML در واقع HTML است، که به صورت XML طراحی شده است.

## مهمترین تفاوت های XHTML با HTML

### ساختار اسناد

- تگ DOCTYPE در XHTML اجباری است.
- صفت xmlns ، در HTML ضروری است.
- تگ های <title>، <head>، <html> و <body>، در HTML ضروری است.

### عناصر XHTML

- عناصر XHTML، باید به خوبی بصورت تو در تو تعریف شوند.
- عناصر XHTML باید همیشه بسته شوند.
- عناصر XHTML، باید به در قالب حروف کوچک تعریف شوند.
- عناصر XHTML، باید یک عنصر والد داشته باشند.

### صفات XHTML

- نام صفات باید با حروف کوچک تعریف شود.
- مقادیر صفات باید تعیین شود.
- به حداقل رساندن صفات، ممنوع است.

### تعریف تگ DOCTYPE ، ضروری است

هر سند XHTML ، باید حتما دارای تگ DOCTYPE باشد.

مثال :

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2. "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3.
4. <html xmlns="http://www.w3.org/1999/xhtml">
5.
6. <head>
7.   <title>Title of document</title>
8. </head>
9.
10. <body>
11.   some content
12. </body>
13.
14. </html>
```

### عناصر XHTML ، باید به خوبی تو در تو تعریف شده باشد.

در HTML ، برخی از عناصر به طور نامناسبی تو در تو تعریف می شود، مانند این مثال:

مثال :

```
1. <b><i>This text is bold and italic</b></i>
```

در XHTML، تمام عناصر باید به خوبی تو در تو ، تعریف شوند ، یکی در داخل دیگری، مانند این مثال:

مثال :

```
1. <b><i>This text is bold and italic</i></b>
```

### عناصر XHTML ، باید همیشه بسته شوند

حالت اشتباه :

مثال :

```
1. <p>This is a paragraph
2. <p>This is another paragraph
```

حالت صحیح:

مثال :

1. `<p>This is a paragraph</p>`
2. `<p>This is another paragraph</p>`

## عناصر خالی نیز باید همیشه بسته شوند

حالت اشتباه :

مثال :

1. A break: `<br>`
2. A horizontal rule: `<hr>`
3. An image: ``

حالت صحیح :

مثال :

1. A break: `<br />`
2. A horizontal rule: `<hr />`
3. An image: ``

## صفات هر تگ در XHTML ، باید با حروف کوچک تعریف شود

حالت اشتباه :

مثال :

1. `<BODY>`
2. `<P>This is a paragraph</P>`
3. `</BODY>`

حالت صحیح :

مثال :

1. `<table width="100%">`

## صفات نیز باید با حروف کوچک نوشته شوند

حالت اشتباه :



مثال :

1. `<table WIDTH="100%">`

حالت صحیح :

مثال :

1. `<table width="100%">`

**مقادیر صفات حتما باید در داخل نقل قول ("") ، تعریف شوند**

حالت اشتباه :

مثال :

1. `<table width=100%>`

حالت صحیح :

مثال :

1. `<table width="100%">`

**در XHTML ، صفات هر تگ باید کامل باشد**

و به حداقل رساندن صفات ناردست است مانند این مثال:

مثال :

1. `<input type="checkbox" name="vehicle" value="car" checked="checked" />`

این مثال در XHTML اشتباه است:

مثال :

1. `<input type="checkbox" name="vehicle" value="car" checked />`

حالت اشتباه :

مثال :

```
1. <input type="text" name="lastname" disabled />
```

حالت صحیح :

مثال :

```
1. <input type="text" name="lastname" disabled="disabled" />
```

## کلام آخر

در آخرین بخش فصل جاری شما را با تفاوت های شاخص **ویژگی های HTML و XHTML** و حالت های صحیح و استاندارد آشنا کردیم، در فصل آتی به فرم ها و ساختار عناصر فرم ها در HTML خواهیم پرداخت.

## فصل دوم : فرم در HTML

### جلسه ۳۳ : فرم در کد HTML

#### مقدمه

با عرض سلام خدمت کاربران سایت [پی وی لرن](#) . در این پست آموزشی قصد داریم به آموزش **تعریف و نحوه کار با فرم ها در html** ، بپردازیم. فرم ها در حقیقت رابط بین داده های ورودی کاربران و سرور است. و از اصلی ترین بخش های هر وب سائتی می باشد. و در حقیقت تصور یک وب سایت بدون هیچگونه فرمی غیر ممکن است. در آموزش های قبلی ما چگونگی کار با تگ های [Table](#) , [Div](#) و تگ [img](#) و همچنین **تعریف رنگ ها در html** را بررسی کردیم. در ادامه این پست آموزشی نیز به چگونگی تعریف و استفاده از فرم ها در [html](#) و فیلدهای مهم آن ، می پردازیم.



#### فرم ها در html

مثال : فرم ها

```
1. <!DOCTYPE html>
2. <html>
3. <body>
4.
5. <h2>HTML Forms</h2>
6.
7. <form action="/action_page.php">
8.   First name:<br>
9.   <input type="text" name="firstname" value="Mickey">
10.  <br>
11.  Last name:<br>
12.  <input type="text" name="lastname" value="Mouse">
13.  <br><br>
14.  <input type="submit" value="Submit">
15. </form>
16.
17. <p>If you click the "Submit" button, the form-
18.   data will be sent to a page called "/action_page.php".</p>
19. </body>
```

## [خودتان امتحان کنید<<](#)

### عنصر <form>

تگ <form> در HTML یک فرم را تعریف می کند که برای دریافت ورودی های کاربر استفاده می شود.

سینتکس تعریف فرم :

مثال :

```

1. <form>
2. .
3. form elements
4. .
5. </form>

```

عناصر فرم انواع مختلفی از عناصر ورودی مانند فیلد های متن، جعبه چک، دکمه های رادیویی، دکمه ارسال و غیره است. که در ادامه به بررسی هر کدام از آنها می پردازیم:

### تگ <input>

عنصر <input> مهمترین عنصر فرم است. عنصر <input> بسته به نوع ویژگی ، می تواند به روش های مختلف نمایش داده شود.

چند مثال از انواع تگ : input

- <input type = "text"> : یک فیلد ورودی یک متن را تعریف می کند.
- <input type = "radio">: یک دکمه رادیویی (برای انتخاب یکی از گزینه های بسیاری)
- <input type = "submit">: یک دکمه ارسال را تعریف می کند (برای ارسال فرم)

### ورودی Text:

تگ <input type="text"> فیلد ی برای دریافت متن ا زکاربر، بکار می رود.

مثال : ورودی Text

```

1. <form>
2. First name:<br>
3. <input type="text" name="firstname"><br>
4. Last name:<br>
5. <input type="text" name="lastname">
6. </form>

```

## [خودتان امتحان کنید<<](#)

نکته: فرم خود قابل مشاهده نیست همچنین توجه داشته باشید که عرض پیش فرض یک فیلد متنی ۲۰ کاراکتر است.

## ورودی radio button:

تگ `<input type = "radio">` وقتی بین چند گزینه باید فقط یکی انتخاب شود، کاربرد دارد.

مثال : ورودی radio

```
1. <form>
2.   <input type="radio" name="gender" value="male" checked> Male<br>
3.   <input type="radio" name="gender" value="female"> Female<br>
4.   input type="radio" name="gender" value="other"> Other>
5. </form>
```

## [خودتان امتحان کنید<<](#)

### : Submit Button

تگ `<input type = "submit">` یک دکمه برای ارسال اطلاعات به **form-handler** تعریف می کند. که معمولاً یک صفحه سرور با یک اسکریپت برای پردازش داده های ورودی است. آدرس `form-handler` در ویژگی **action** فرم ها مشخص می شود.

مثال : عنصر submit

```
1. <"form action="/action_page.php>
2. < First name:<br>
3.   <input type="text" name="firstname" value="Mickey"><br>
4.   <Last name:<br>
5.   <input type="text" name="lastname" value="Mouse"><br><br>
6.   <input type="submit" value="Submit">
7. </form>
```

## [خودتان امتحان کنید<<](#)

### ویژگی action

این ویژگی برای ارسال اطلاعات فرم ها کاربرد دارد. به طور معمول، اطلاعات فرم به یک صفحه وب در سرور ارسال می شود زمانی که کاربر بر روی دکمه **submit** کلیک کند. در مثال بالا، در حقیقت داده های فرم به یک صفحه در سرور به نام `action_page.php/""` ارسال می شود. این صفحه شامل یک اسکریپت سمت سرور است که داده های فرم را مدیریت می کند:

`<form action="/action_page.php">`

### ویژگی target

ویژگی **target** مشخص می کند که آیا نتایج ارسال شده در یک تب جدید مرورگر، یک قاب یا در پنجره فعلی باز می شود. مقدار پیش فرض آن `"self"` است. که به این معنی است که فرم در پنجره فعلی نمایش داده می شود. برای نمایش نتایج در یک `tab` جدید از مرورگر باید از مقدار `"blank_"` استفاده کرد.

1. `<"form action="/action_page.php" target="_blank">`

[خودتان امتحان کنید<<](#)

## ویژگی method

ویژگی method مشخص می کند که اطلاعات فرم با روش GET یا POST ارسال شود.

مثال : ویژگی method

1. `<"form action="/action_page.php" method="get">`
- 2.
3. `<"form action="/action_page.php" method="post">`

[خودتان امتحان کنید<<](#)

و یا :

مثال : مقدار post

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

[خودتان امتحان کنید<<](#)

چه زمانی از GET و یا POST استفاده کنیم؟

بطور پس فرض اطلاعات در حالت GET ارسال می شود. هنگامی که GET استفاده می شود، اطلاعات با سرعت بالاتری ارسال می شود. اما داده های فرم ارسال شده در قسمت آدرس صفحه قابل مشاهده خواهد بود:

[/lastname=Mouse&action\\_page.php?firstname=Mickey](#)

روش POST اطلاعات را با سرعت کند تری نسبت به GET ارسال می کند. اما داده های فرم ارسال شده را در فیلد آدرس صفحه نشان نمی دهد. پس اگر اطلاعات فرم حاوی اطلاعات حساس یا شخصی باشد. از POST استفاده کنید.

## ویژگی Name

هر کدام از تگ های input باید ویژگی name را داشته باشد تا اطلاعات آن ارسال شود. به عنوان مثال در فیلدهای فرم پایین ، با کلیک روی submit فقط داده های فیلد "Last name" ارسال می شود:

مثال : ویژگی Name

1. `<"form action="/action_page.php">`

```

2. <First name:<br>
3. <input type="text" value="Mickey"><br>
4. < Last name:<br>
5. <input type="text" name="lastname" value="Mouse"><br><br>
6. <input type="submit" value="Submit">
7. </form>

```

[خودتان امتحان کنید<<](#)

گروه بندی تمام عناصر یک فرم، با تگ <fieldset>

fieldset

تگ <fieldset> برای دسته بندی یا گروه بندی عناصر و تگ های موجود در یک فرم بکار می رود. از تگ <legend> برای تعیین عنوان دسته استفاده می شود.

مثال : عنصر fieldset

```

1. <"form action="/action_page.php>
2. <fieldset>
3. <legend>Personal information:</legend>
4. < First name:<br>
5. <input type="text" name="firstname" value="Mickey"><br>
6. < Last name:<br>
7. <input type="text" name="lastname" value="Mouse"><br><br>
8. <input type="submit" value="Submit">
9. </fieldset>
10. </form>

```

[خودتان امتحان کنید<<](#)

لیست تمام ویژگی های <form> :

توضیحات	صفت

accept-charset	مشخصه charset مورد استفاده در فرم ارسال شده (پیش فرض: charset صفحه).
action	آدرس (URL) که در آن برای ارسال فرم (یک پیشفرض: صفحه ارسالی) را مشخص می کند.
autocomplete	مشخص می کند که آیا مرورگر باید فرم را به صورت خودکار تکمیل کند
enctype	رمزگذاری داده های ارسال شده را مشخص می کند (پیش فرض: کد گذاری شده توسط URL است).
method	روش HTTP مورد استفاده برای ارسال فرم را تعیین می کند (به طور پیش فرض: GET).
name	نامی که برای تشخیص فرم استفاده می شود را مشخص می کند (برای استفاده: DOM: document.forms.name).
novalidate	مشخص می کند که مرورگر نباید فرم را تأیید کند.
target	مقصد هدف آدرس را در عمل مشخص می کند (به طور پیش فرض: self_).

## کلام آخر

**فرم در کد HTML** از بخش های اصلی تشکیل دهنده ی اغلب وب سایت ها می باشد، چراکه دریافت انواع ورودی ها از کاربران ، از طریق فرم ها انجام می شود. از این رو فراگیری نحوه ی پیاده سازی فرم در کد HTML از اهمیت ویژه ای برخوردار است.



### مقدمه

با عرض سلام و وقت بخیر خدمت کاربران سایت [پی وی لرن](#) . و کاربرانی که [دوره کامل آموزش html](#) را دنبال می کنند. فرم ها یکی از اصلی ترین بخش ها در ساختار صفحات وب اغلب وب سایت ها می باشند چرا که شما برای دریافت ورودی های مختلف از کاربران ناچار به استفاده از فرم ها هستید، فرم ها خود از عناصر مختلفی برای دریافت ورودی های مختلف از کاربران تشکیل می شوند که نوع **عناصر فرم در HTML** بستگی به نوع ورودی کاربران دارد. در ادامه ی این مباحث ما شما را با مهمترین عناصر فرم در HTML آشنا خواهیم کرد.

### عناصر فرم در HTML

در ادامه ی این مبحث به معرفی تعدادی از پرکاربردترین عناصر فرم در HTML پرداخته ایم.

#### عنصر ورودی <input>

مهمترین عنصر در هر فرم ، عنصر <input> است. عنصر <input> را می توان در قالب ها و فرمت های مختلفی نمایش داد که بستگی به صفت **type** دارد.

مثال : عنصر input

```
1. <input name="firstname" type="text">
```

#### [خودتان امتحان کنید<>](#)

اگر ویژگی type را مشخص نکنید، فیلد input نوع پیش فرض را که "text" است، در نظر می گیرد.

#### عنصر <select>

عنصر <select> یک لیست کشویی را ایجاد می کند.

مثال : عنصر select

```
1. <select name="cars">
2.   <option value="volvo">Volvo</option>
3.   <option value="saab">Saab</option>
4.   <option value="fiat">Fiat</option>
5.   <option value="audi">Audi</option>
6. </select>
```

#### [خودتان امتحان کنید<>](#)

عنصر <option> آیتم های <select> را که قابل انتخاب هستند را تعریف می کند. در حالت پیش فرض اولین آیتم لیست <select> در حالت انتخاب شده است.

برای مشخص کردن یک عنصر به عنوان انتخاب شده، باید صفت **selected** را در عنصر <option> مربوطه تعریف کرد:

مثال : عنصر انتخاب شده

```
1. <option value="fiat" selected>Fiat</option>
```

### [خودتان امتحان کنید >>](#)

### ارزش های قابل مشاهده

از صفت `size` می توانید برای مشخص کردن تعداد ارزش های قابل مشاهده استفاده کنید.

مثال : ارزش ها

```
1. <select name="cars" size="3">
2.   <option value="volvo">Volvo</option>
3.   <option value="saab">Saab</option>
4.   <option value="fiat">Fiat</option>
5.   <option value="audi">Audi</option>
6. </select>
```

### [خودتان امتحان کنید >>](#)

### انتخاب های چندگانه

استفاده از صفت `multiple` در لیست به کاربر اجازه می دهد تا بیش از یک انتخاب بتواند انجام دهد:

مثال : انتخاب چندگانه

```
1. <select name="cars" size="4" multiple>
2.   <option value="volvo">Volvo</option>
3.   <option value="saab">Saab</option>
4.   <option value="fiat">Fiat</option>
5.   <option value="audi">Audi</option>
6. </select>
```

### [خودتان امتحان کنید >>](#)

### عنصر <textarea>

عنصر `<textarea>` یک فیلد ورودی چند خطی را تعریف می کند.

مثال : عنصر textarea

```
1. <textarea name="message" rows="10" cols="30">
2. The cat was playing in the garden.
```

### [خودتان امتحان کنید >>](#)

صفت `rows` تعداد قابل مشاهده خطوط را در یک منطقه متن مشخص می کند.  
خصیصه `cols` عرض یک ناحیه از متن را مشخص می کند.

فیلد ورودی زیر، چگونگی نمایش کد HTML بالا را در مرورگر نشان می دهد:

```
</ The cat was playing in the garden.<br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br />
```

شما همچنین می توانید با استفاده از CSS اندازه ی عنصر text area را مشخص کنید.

مثال : تعیین اندازه ی textarea

1. `<textarea name="message" style="width:200px; height:600px">`
2. The cat was playing in the garden.

## خودتان امتحان کنید

### عنصر <button>

عنصر `<button>` یک button قابل کلیک را تعریف می کند.

مثال : عنصر button

1. `<button type="button" onclick="alert('Hello World!')">Click Me!</button>`

## خودتان امتحان کنید

### عناصر فرم در HTML 5

HTML 5 عناصر فرم زیر را ارائه کرده است:

- `<datalist>`
- `<output>`

نکته : مرورگرها عناصر ناشناخته را نمایش نمی دهند، عناصر جدید نیز در برخی مرورگرهای قدیمی پشتیبانی نشده و در نتیجه نمایش داده نمی شود.

### عنصر <datalist> در HTML 5

عنصر `<datalist>` یک لیست از گزینه های از پیش تعریف شده برای یک عنصر `<input>` را مشخص می کند. کاربران یک لیست کشویی از گزینه های از پیش تعیین شده را مشاهده می کنند.

صفت `list` از عنصر `<input>` باید به صفت `id` از عنصر `<datalist>` اشاره کند.

```

1. <form action="/action_page.php">
2.   <input list="browsers">
3.   <datalist id="browsers">
4.     <option value="Internet Explorer">
5.     <option value="Firefox">
6.     <option value="Chrome">
7.     <option value="Opera">
8.     <option value="Safari">
9.   </datalist>
10. </form>

```

## [خودتان امتحان کنید<<](#)

### عنصر <output> در HTML 5

عنصر <output> نتیجه یک محاسبه است (مثل یک محاسبه انجام شده توسط یک اسکریپت):

مثال : خروجی

```

1. <form action="/action_page.php"
2.   oninput="x.value=parseInt(a.value)+parseInt(b.value)">
3.   0
4.   <input type="range" id="a" name="a" value="50">
5.   100 +
6.   <input type="number" id="b" name="b" value="50">
7.   =
8.   <output name="x" for="a b"></output>
9.   <br><br>
10.  <input type="submit">
11. </form>

```

## [خودتان امتحان کنید<<](#)

### عناصر فرم در HTML

توضیحات	تگ
یک فرم HTML را برای ورودی کاربر تعریف می کند.	<form>
یک کنترل ورودی را تعریف می کند.	<input>

کنترل ورودی چند خطی را تعریف می کند (text area)	<textarea>
یک برچسب برای عنصر <input> تعریف می کند.	<label>
عناصر گروه در یک فرم	<fieldset>
یک عنوان را برای عنصر <fieldset> تعریف می کند.	<legend>
یک لیست کشویی را تعریف می کند.	<select>
یک گروه از گزینه های مرتبط را در لیست کشویی تعریف می کند.	<optgroup>
یک گزینه را در لیست کشویی تعریف می کند.	<option>
یک دکمه قابل کلیک را تعریف می کند.	<button>
لیستی از گزینه های از پیش تعیین شده برای کنترل های ورودی را مشخص می کند.	<datalist>
نتیجه محاسبات را تعریف می کند.	<output>

## کلام آخر

یکی از اصلی ترین بخش های تشکیل دهنده ی اغلب وب سایت ها کنترل ها و عناصر فرم می باشند که کار دریافت انواع ورودی از کاربران را انجام می دهند، از این رو برای طراحی یک وب سایت آشنایی با **عناصر فرم در HTML** از اهمیت ویژه ای برخوردار می باشد.

در بخش بعدی به انواع ورودی در HTML می پردازیم.

### مقدمه

با عرض سلام و وقت بخیر خدمت کاربران سایت [پی وی لرن](#) . و کاربرانی که [دوره کامل آموزش html](#) را دنبال می کنند. در بخش قبلی ما انواع عناصر فرم پر کاربرد در HTML را معرفی کردیم. همچنین ساختار عنصر input که بهترین عنصر برای دریافت ورودی در فرم هاست را تا حدودی مورد بررسی قرار داده و با طرز بکارگیری آن نیز آشنا شدیم. در ادامه در این بخش قصد داریم تا به جزئیات بیشتری از عنصر فوق پرداخته و انواع ورودی در HTML را شرح دهیم، شما نیاز به دریافت ورودی ها مختلف از کاربران دارید ، از این رو باید با **انواع ورودی در HTML** آشنا بوده و از آن ها برحسب نیاز استفاده نمائید. برای آشنایی بیشتر با این مباحث و کاربرد انواع ورودی در HTML در ادامه آموزش با ما همراه باشید.

### انواع ورودی در HTML

عنصر input بسته به نوع ورودی در فرمت های مختلفی تعریف می شود که در این بخش به آن پرداخته ایم.

#### نوع text در Input

کد `<input type="text">` یک فیلد ورودی تکی را تعریف می کند:

مثال : نوع ورودی text

```
1. <form>
2.   First name:<br>
3.   <input type="text" name="firstname"><br>
4.   Last name:<br>
5.   <input type="text" name="lastname">
6. </form>
```

#### [خودتان امتحان کنید<>](#)

کد html فوق عناصر زیر را در صفحه مرورگر نشان می دهد:

:First name

:Last name

#### نوع ورودی password در Input

تگ `<input type="password">` یک فیلد password را تعریف می کند.

مثال : نوع ورودی password

```
1. <form>
2.   User name:<br>
3.   <input type="text" name="username"><br>
4.   User password:<br>
5.   <input type="password" name="psw">
6. </form>
```

## خودتان امتحان کنید<<

کد HTML فوق نتیجه زیر را نشان می دهد:

:User name

:User password

نکته : کاراکترهای یک فیلد رمز عبور (به صورت ستاره یا حلقه نشان داده میشوند) پنهان می شوند.

## نوع ورودی submit

تگ `<input type="submit">` یک button را برای ارسال داده های فرم به **form-handler** ایجاد می کند. **form-handler** فرم معمولاً یک صفحه سرور با یک اسکریپت برای پردازش داده های ورودی است.

ویژگی **form-handler** در صفت **action** فرم تعریف می شود.

مثال : نوع ورودی submit

```
1. <form action="/action_page.php">
2.   First name:<br>
3.   <input type="text" name="firstname" value="Mickey"><br>
4.   Last name:<br>
5.   <input type="text" name="lastname" value="Mouse"><br><br>
6.   <input type="submit" value="Submit">
7. </form>
```

## خودتان امتحان کنید<<

کد html فوق نتیجه زیر را نشان می دهد:

:First name

:Last name

اگر مقدار value را از submit را حذف کنید، دکمه یک متن پیش فرض را دریافت می کند:

مثال : ویژگی value

```
1. <form action="/action_page.php">
2.   First name:<br>
3.   <input type="text" name="firstname" value="Mickey"><br>
4.   Last name:<br>
5.   <input type="text" name="lastname" value="Mouse"><br><br>
6.   <input type="submit">
7. </form>
```

## خودتان امتحان کنید<<



## نوع reset در عنصر input

تگ `<input type="reset">` یک دکمه reset را تعریف می کند تا با کلیک روی آن تمام مقادیر فرم به حالت پیش فرض تنظیم شود.

مثال : عنصر reset

```
1. <form action="/action_page.php">
2.   First name:<br>
3.   <input type="text" name="firstname" value="Mickey"><br>
4.   Last name:<br>
5.   <input type="text" name="lastname" value="Mouse"><br><br>
6.   <input type="submit" value="Submit">
7.   <input type="reset">
8. </form>
```

### خودتان امتحان کنید<<

کد html فوق نتیجه ی زیر را نشان می دهد:

:First name

:Last name

اگر مقادیر ورودی را تغییر دهید و سپس بر روی دکمه "Reset" کلیک کنید، داده های فرم به مقادیر پیش فرض بازنشانی می شوند.

## نوع ورودی radio

عنصر `<input type="radio">` یک **radio button** تعریف می کند. **radio button** به شما اجازه می دهد تا فقط یک گزینه از تعداد محدودی از گزینه ها را انتخاب کنید:

مثال : ورودی radio

```
1. <form>
2.   <input type="radio" name="gender" value="male" checked> Male<br>
3.   <input type="radio" name="gender" value="female"> Female<br>
4.   <input type="radio" name="gender" value="other"> Other
5. </form>
```

### خودتان امتحان کنید<<

کد html فوق نتیجه ی زیر را در مرورگر نمایش می دهد:

Male

Female

Other

## نوع ورودی checkbox

تگ `<input type="checkbox">` یک عنصر checkbox را ایجاد می کند. Checkbox گزینه هایی را ایجاد می کند، که کاربر می تواند یک یا چند مورد از آن گزینه ها را انتخاب کرده و یا هیچیک را انتخاب نکند:

مثال : ورودی checkbox

```
1. <form>
2.   <input type="checkbox" name="vehicle1" value="Bike"> I have a bike<br>
3.   <input type="checkbox" name="vehicle2" value="Car"> I have a car
4. </form>
```

### [خودتان امتحان کنید<<](#)

کد html فوق نتیجه زیر را در مرورگر ایجاد می کندم:

I have a bike

I have a car

## نوع ورودی button

تگ `<input type="button">` یک عنصر button تعریف می کند:

مثال : ورودی button

```
1. <input type="button" onclick="alert('Hello World!')" value="Click Me!">
```

### [خودتان امتحان کنید<<](#)

## انواع عناصر input در HTML 5

در HTML 5 انواع ورودی های input زیر نیز اضافه شده است:

- color
- date
- datetime-local
- email
- month
- number
- range
- search
- tel
- time
- url

انواع ورودی جدید که توسط مرورگرهای وب قدیمی پشتیبانی نمی شوند، به عنوان `<input type = "text">` رفتار خواهند کرد.

## نوع ورودی color

تگ `<input type="color">` برای فیلدهای ورودی که شامل رنگ هستند استفاده می شود. بسته به پشتیبانی مرورگر، یک انتخاب رنگ می تواند در زمینه ورودی نشان داده شود.

مثال : ورودی color

```
1. <form>
2.   Select your favorite color:
3.   <input type="color" name="favcolor">
4. </form>
```

## خودتان امتحان کنید

### نوع ورودی Date

تگ `<input type="date">` برای فیلدهایی که شامل تاریخ می شوند استفاده می شود. بسته به پشتیبانی مرورگر، یک انتخاب تاریخ در قسمت ورودی نشان داده می شود:

مثال : ورودی date

```
1. <form>
2.   Birthday:
3.   <input type="date" name="bday">
4. </form>
```

## خودتان امتحان کنید

شما می توانید از صفات `min` و `max` برای اعمال محدودیت در تاریخ استفاده کنید.

مثال : صفات min و max

```
1. <form>
2.   Enter a date before 1980-01-01:
3.   <input type="date" name="bday" max="1979-12-31"><br>
4.   Enter a date after 2000-01-01:
5.   <input type="date" name="bday" min="2000-01-02"><br>
6. </form>
```

## خودتان امتحان کنید

### نوع ورودی Datetime local

تگ `<input type="datetime-local">` یک فیلد ورودی تاریخ و زمان را بدون time zone مشخص می کند. بسته به پشتیبانی مرورگر، یک انتخاب تاریخ در قسمت ورودی نشان داده می شود.

مثال : نوع datetime-local

```
1. <form>
2.   Birthday (date and time):
3.   <input type="datetime-local" name="bdaytime">
4. </form>
```

[خودتان امتحان کنید](#)

## نوع ورودی Email

تگ `<input type="email">` برای فیلدهای input که نیاز به وارد کردن آدرس Email دارند استفاده می شود. بسته به پشتیبانی مرورگر، آدرس ایمیل می تواند به صورت خودکار هنگام تأیید تأیید شود.

بعضی از گوشی های هوشمند نوع ایمیل را تشخیص می دهند و "com." را به صفحه کلید اضافه می کنند تا ورودی های ایمیل را مطابقت دهند.

مثال : نوع ورودی Email

```
1. <form>
2.   E-mail:
3.   <input type="email" name="email">
4. </form>
```

[خودتان امتحان کنید](#)

## نوع ورودی File

تگ `<input type="file">` برای مواردی که نیاز به وارد کردن فایل توسط کاربر دارد استفاده می شود.

مثال : نوع ورودی file

```
1. <form>
2.   Select a file: <input type="file" name="myFile">
3. </form>
```

[خودتان امتحان کنید](#)

## نوع ورودی month

تگ `<input type="month">` به کاربر اجازه می دهد تا ماه و سال را انتخاب می کند. بسته به پشتیبانی مرورگر، یک انتخاب تاریخ در قسمت ورودی نشان داده می شود.

مثال : نوع ورودی ماه

```
1. <form>
```

```

2. Birthday (month and year):
3. <input type="month" name="bdaymonth">
4. </form>

```

## [خودتان امتحان کنید<<](#)

### نوع ورودی عدد

تگ `<input type="number">` یک فیلد ورودی عددی تعریف می کند. شما همچنین می توانید محدودیت هایی را برای تعداد اعداد پذیرفته شده تنظیم کنید. مثال زیر یک فیلد ورودی عددی را نشان می دهد، جایی که می توانید مقدار ۱ تا ۵ را وارد کنید:

مثال : نوع ورودی عدد

```

1. <form>
2. Quantity (between 1 and 5):
3. <input type="number" name="quantity" min="1" max="5">
4. </form>

```

## [خودتان امتحان کنید<<](#)

### صفات عنصر Input

در اینجا یک لیست از محدودیت های ورودی مشترک وجود دارد (برخی از آنها در HTML5 جدید هستند):

توضیحات	صفت
تعیین می کند که یک فیلد ورودی باید غیرفعال شود.	disabled
حداکثر مقدار یک فیلد ورودی را مشخص می کند.	max
حداکثر تعداد کاراکتر برای یک فیلد ورودی را مشخص می کند.	maxlength
مقدار حداقل برای یک فیلد ورودی را مشخص می کند.	min

الگوی یک عبارت منظم را برای بررسی مقدار ورودی در مقابل مشخص می کند.	pattern
مشخص می کند که یک فیلد ورودی فقط خواندنی (قابل تغییر نیست).	readonly
مشخص می کند که یک فیلد ورودی مورد نیاز است (باید پر شود).	required
تعیین عرض (در حروف) یک فیلد ورودی.	size
تعداد قانونی را برای فیلد ورودی مشخص می کند.	step
مقدار پیش فرض برای یک فیلد ورودی را مشخص می کند.	value

مثال زیر یک فیلد ورودی عددی را نشان می دهد، که در آن می توانید یک مقدار از ۰ تا ۱۰۰ را در مراحل ۱۰ وارد کنید. مقدار پیش فرض ۳۰ است:

### مثال : رنج عددی

```

1. <form>
2.   Quantity:
3.   <input type="number" name="points" min="0" max="100" step="10" value="30">
4. </form>

```

## خودتان امتحان کنید<>

### نوع ورودی range

تگ `<input type="range">` یک نوار لغزان را با یک دکمه روی آن، برای انتخاب یک رنج عددی توسط کاربر ایجاد می کند. حال پیش فرض مقادیر آن ۰ تا ۱۰۰ است. با این حال شما می توانید با صفات `min` و `max` و `step` حداقل و حداکثر مقادیر و فواصل بین مقادیر را مشخص کنید:

### مثال : نوع ورودی Range

```

1. <form>
2.   <input type="range" name="points" min="0" max="10">
3. </form>

```

## [خودتان امتحان کنید](#)

### نوع ورودی search

تگ `<input type="search">` برای ایجاد فیلدهای search استفاده می شود.

مثال : نوع search

```
1. <form>
2.   Search Google:
3.   <input type="search" name="googlesearch">
4. </form>
```

## [خودتان امتحان کنید](#)

### نوع ورودی tel

تگ `<input type="tel">` فیلدی را برای دریافت شماره تلفن از کاربر ایجاد می کند.  
نکته : این ویژگی در مرورگر Safari 8 پشتیبانی نمی شود.

مثال : ورودی شماره تلفن

```
1. <form>
2.   Telephone:
3.   <input type="tel" name="usrtel">
4. </form>
```

## [خودتان امتحان کنید](#)

### نوع ورودی Time

تگ `<input type="time">` به کاربر اجازه انتخاب یک Time را می دهد.  
بسته به پشتیبانی مرورگر شما یک تگ فوق time picker را در فیلد input ظاهر می کند.

مثال : ورودی time

```
1. <form>
2.   Select a time:
3.   <input type="time" name="usr_time">
4. </form>
```

## [خودتان امتحان کنید](#)

### نوع ورودی URL

تگ `<input type="url">` برای فیلدهایی استفاده می شود که نیاز به تایپ آدرس URL دارند.  
بسته به پشتیبانی مرورگر، فیلد URL زمانی که ارسال می شود، می تواند به صورت خودکار تایید شود.

```

1. <form>
2.   Add your homepage:
3.   <input type="url" name="homepage">
4. </form>

```

### [خودتان امتحان کنید<<](#)

## نوع ورودی week

تگ `<input type="week">` به کاربر اجازه می دهد تا هفته و سال را انتخاب کند. بسته به پشتیبانی مرورگر شما یک تگ فوق Date picker را در فیلد input ظاهر می کند.

### مثال : ورودی week

```

1. <form>
2.   Select a week:
3.   <input type="week" name="week_year">
4. </form>

```

### [خودتان امتحان کنید<<](#)

## کلام آخر

مهمترین و پرکاربردترین عنصر فرم در html تگ های ورودی (Input) هستند که طیف وسیعی از داده های مختلف را از کاربر دریافت می کنند، از این رو مباحث این بخش را به آشنایی شما با **انواع ورودی در HTML** اختصاص دادیم.



### مقدمه

با عرض سلام و وقت بخیر خدمت کاربران سایت [پی وی لرن](#) . و کاربرانی که [دوره کامل آموزش html](#) را دنبال می کنند. در بخش قبلی انواع مختلف عناصر input فرم ها را ارائه کردیم و کاربرد هر یک را نیز توضیح دادیم. عناصر input دارای صفات و ویژگی ها گسترده ای است که به شما اجازه می دهد تا انواع مختلف قوانین و محدودیت ها را به عناصر input خود اعمال کرده و یا ویژگی های خاصی را به آن اضافه کنید. این صفات در واقع در ایجاد عناصر input سفارشی کاربرد وسیعی دارند. برای آشنایی با انواع **صفات input در html** و طرز استفاده از آن در ادامه ی مباحث این بخش با ما همراه باشید.

### خصوصیات input در html

در ادامه ی آموزش ما انواع صفات input در html به همراه ساختار و مثال هایی از آن ها را ارائه کرده ایم.

#### صفت value

صفت `value` مقدار داخل یک فیلد input را تعریف می کند:

مثال : صفت value

```
1. <form action="">
2.   First name:<br>
3.   <input type="text" name="firstname" value="John">
4. </form>
```

### [خودتان امتحان کنید<>](#)

#### صفت readonly

صفت `readonly` محتوای یک فیلد input را غیر قابل ویرایش و فقط خواندنی می کند.

مثال : صفت readonly

```
1. <form action="">
2.   First name:<br>
3.   <input type="text" name="firstname" value="John" readonly>
4. </form>
```

### [خودتان امتحان کنید<>](#)

#### صفت disabled

صفت `disabled` یک فیلد input را غیر فعال می کند. یک فیلد ورودی `disabled` (غیرفعال) غیر قابل استفاده و غیر قابل کلیک است و در هنگام ارسال فرم آن ارزش ارسال نمی شود.

مثال : صفت disabled

```
1. <form action="">
2.   First name:<br>
3.   <input type="text" name="firstname" value="John" disabled>
4. </form>
```

[خودتان امتحان کنید](#)

### صفت size

صفت `size` اندازه (در کاراکترها) را برای فیلد `input` تعریف می کند.

مثال : صفت size

```
1. <form action="">
2.   First name:<br>
3.   <input type="text" name="firstname" value="John" size="40">
4. </form>
```

[خودتان امتحان کنید](#)

### صفت maxlength

صفت `maxlength` حداکثر طول ورودی را برای فیلد `input` تعریف می کند.

مثال : صفت maxlenght

```
1. <form action="">
2.   First name:<br>
3.   <input type="text" name="firstname" maxlength="10">
4. </form>
```

[خودتان امتحان کنید](#)

با استفاده از ویژگی `maxlength` ، فیلد `text` ، ورودی بیشتر از حد تعیین شده را قبول نمی کند.

ویژگی `maxlength` فقط جلوی ورودی اضافه تر را می گیرد و به کاربر هیچ اعلامی نمی کند.

برای هرگونه اعلام به کاربر باید از کدهای جاوااسکریپت استفاده کنید.

### صفات HTML 5

در HTML 5 صفات جدیدی به فیلد `input` اضافه شده که به شرح زیر است:

- `autocomplete`
- `autofocus`
- `form`
- `formaction`
- `formenctype`

- formmethod
- formnovalidate
- formtarget
- height and width
- list
- min and max
- multiple
- pattern (regexp)
- placeholder
- required
- step

همچنین صفات زیر برای `<form>` است:

- autocomplete
- novalidate

## صفت autocomplete

خاصیت `autocomplete` مشخص می کند که آیا یک فرم یا فیلد ورودی باید به صورت خودکار تکمیل شود یا خیر. زمانیکه صفت `autocomplete` فعال باشد، مرورگر بصورت خودکار مقادیر ورودی را در هنگام تایپ کردن کاربر بر اساس مقادیری که کاربر قبلاً وارد کرده تکمیل می کند.

**نکته:** ممکن است برای تکمیل فرم به صورت خودکار صفت `autocomplete` برای فرم "on" و برای فیلدهای ورودی خاص "off" یا برعکس باشد. صفت `autocomplete` با `<form>` و انواع `<input>` از جمله: text, search, url, tel, email, password ... کار می کند.

مثال : تکمیل خودکار کد

```
1. <form action="/action_page.php" autocomplete="on">
2.   First name:<input type="text" name="fname"><br>
3.   Last name: <input type="text" name="lname"><br>
4.   E-mail: <input type="email" name="email" autocomplete="off"><br>
5.   <input type="submit">
6. </form>
```

## [خودتان امتحان کنید<>](#)

### صفت novalidate

ویژگی `novalidate` یک صفت برای یک `<form>` است. `Novalidate` مشخص می کند که داده های فرم هنگام ارسال نمایش داده نمی شود.

مثال : عدم اعتبارسنجی

```
1. <form action="/action_page.php" novalidate>
2.   E-mail: <input type="email" name="user_email">
```

- `<input type="submit">`
- `</form>`

## [خودتان امتحان کنید<<](#)

### صفت autofocus

صفت `focus` مشخص می کند که یک فیلد `input` هنگام بارگذاری صفحه وب باید به حالت `focus` (انتخاب) برود.

مثال : انتخاب خودکار

- First name: `<input type="text" name="fname" autofocus>`

## [خودتان امتحان کنید<<](#)

### صفت form

خصوصیت `form` یک یا چند فرم را مشخص می کند که عنصر `<input>` متعلق به آن است. نکته: برای مراجعه به بیش از یک فرم، از یک لیست از شناسه های فرم `space-separated` استفاده کنید.

مثال : عنصر form

- `<form action="/action_page.php" id="form1">`
- First name: `<input type="text" name="fname"><br>`
- `<input type="submit" value="Submit">`
- `</form>`
- 
- Last name: `<input type="text" name="lname" form="form1">`

## [خودتان امتحان کنید<<](#)

### صفت formaction

صفت `formaction` آدرس URL یک فایل در کنترل `input` که در هنگام ارسال فرم پردازش می شود را مشخص می کند. صفت `formaction` روی ویژگی `action` از عنصر `<form>` اعمال می شود. صفت `formaction` با کدهای `type="image"` و `type="submit"` استفاده می شود.

مثال : صفت formaction

- `<form action="/action_page.php">`
- First name: `<input type="text" name="fname"><br>`
- Last name: `<input type="text" name="lname"><br>`
- `<input type="submit" value="Submit"><br>`
- `<input type="submit" formaction="/action_page2.php"`
- `value="Submit as admin">`
- `</form>`

## [خودتان امتحان کنید <<](#)

### صفت formenctype

صفت `formenctype` مشخص می کند که چگونه داده های یک فرم در هنگام ارسال encode شود. صفت `formenctype` روی ویژگی `enctype` از عنصر `<form>` اعمال می شود. صفت `formenctype` با `type="submit"` و `type="image"` استفاده می شود.

مثال : صفت formenctype

```
1. <form action="/action_page_binary.asp" method="post">
2.   First name: <input type="text" name="fname"><br>
3.   <input type="submit" value="Submit">
4.   <input type="submit" formenctype="multipart/form-data"
5.     value="Submit as Multipart/form-data">
6. </form>
```

## [خودتان امتحان کنید <<](#)

### صفت formmethod

صفت `formmethod` متد HTTP را برای ارسال داده های فرم به آدرس `action` تعریف می کند. صفت `formmethod` روی صفت `method` از عنصر `<form>` اعمال می شود. صفت `formmethod` می تواند با `type="submit"` و `type="image"` استفاده شود.

مثال : ویژگی method

```
1. <form action="/action_page.php" method="get">
2.   First name: <input type="text" name="fname"><br>
3.   Last name: <input type="text" name="lname"><br>
4.   <input type="submit" value="Submit">
5.   <input type="submit" formmethod="post" value="Submit using POST">
6. </form>
```

## [خودتان امتحان کنید <<](#)

### صفت formnovalidate

صفت `formnovalidate` روی صفت `novalidate` از عنصر `<form>` اعمال می شود. صفت `formnovalidate` می تواند با `type="submit"` استفاده شود.

مثال : اعتبارسنجی

```
1. <form action="/action_page.php">
2.   E-mail: <input type="email" name="userid"><br>
3.   <input type="submit" value="Submit"><br>
4.   <input type="submit" formnovalidate value="Submit without validation">
5. </form>
```

## خودتان امتحان کنید

### صفت formtarget

ویژگی `formtarget` یک نام یا یک کلمه کلیدی را مشخص می کند که نشان می دهد کجا پاسخی را که پس از ارسال فرم دریافت شده است نمایش دهد.

صفت `formtarget` روی صفت `target` از عنصر `<form>` اعمال می شود. صفت `formtarget` می تواند با `type="submit"` و `type="image"` استفاده شود.

مثال : نمایش پاسخ دریافتی از سرور

```
1. <form action="/action_page.php">
2.   First name: <input type="text" name="fname"><br>
3.   Last name: <input type="text" name="lname"><br>
4.   <input type="submit" value="Submit as normal">
5.   <input type="submit" formtarget="_blank"
6.     value="Submit to a new window">
7. </form>
```

## خودتان امتحان کنید

### صفات width و height

صفات `width` و `height` ارتفاع و عرض یک عنصر `<input type="image">` را مشخص می کند. نکته : همیشه اندازه تصاویر را مشخص کنید. اگر مرورگر اندازه را نداشته باشد، صفحه هنگام بارگیری تصاویر، سوسو می شود.

مثال : صفت width و height

```
1. <input type="image" src="img_submit.gif" alt="Submit" width="48" height="48">
```

## خودتان امتحان کنید

### صفت list

صفت `list` به یک عنصر `<datalist>` که شامل آپشن هایی برای تگ `<input>` است اشاره می کند.

مثال : عنصر datalist

```
1. <input list="browsers">
2.
3. <datalist id="browsers">
4.   <option value="Internet Explorer">
5.   <option value="Firefox">
6.   <option value="Chrome">
7.   <option value="Opera">
8.   <option value="Safari">
9. </datalist>
```

## خودتان امتحان کنید

### صفات max و min

صفات `min` و `max` حداقل و حداکثر مقادیر را برای عنصر `<input>` مشخص می کند. صفات `min` و `max` با انواع `input` که شامل : `number`, `range`, `date`, `datetime-local`, `month`, `time` and `week` می شود کار می کند.

مثال : صفت max و min

1. Enter a date before 1980-01-01:
2. `<input type="date" name="bday" max="1979-12-31">`
- 3.
4. Enter a date after 2000-01-01:
5. `<input type="date" name="bday" min="2000-01-02">`
- 6.
7. Quantity (between 1 and 5):
8. `<input type="number" name="quantity" min="1" max="5">`

## خودتان امتحان کنید

### صفت multiple

صفت `multiple` مشخص می کند که کاربر می تواند بیشتر از یک مقدار را در ورودی `input` وارد کند. صفت `multiple` برای دو نوع از عنصر `input` که شامل : `email` و `file` می شود کاربرد دارد.

مثال : ورودی چندگانه

1. Select images: `<input type="file" name="img" multiple>`

## خودتان امتحان کنید

### صفت pattern

صفت `pattern` در واقع یک `regular expression` (عبارت منظم) که ورودی تگ `<input>` را چک می کند مشخص می کند. با انواع عناصر `input` از جمله : `text`, `search`, `url`, `tel`, `email` و `password` کار می کند.

مثال : صفت pattern

1. Country code: `<input type="text" name="country_code" pattern="[A-Za-z]{3}" title="Three letter country code">`

## [خودتان امتحان کنید <<](#)

### صفت placeholder

صفت `placeholder` در واقع یک متن پیش فرض یا راهنما داخل ورودی `input` قرار می دهد که به محض کلیک کاربر حذف می شود. ویژگی `placeholder` را می توان در انواع عناصر `input` از جمله : `text`, `search`, `url`, `tel`, `email` و `password` بکار برد.

مثال : صفت placeholder

1. `<input type="text" name="fname" placeholder="First name">`

## [خودتان امتحان کنید <<](#)

### صفت required

صفت `required` مشخص می کند که یک فیلد ورودی `input` قبل از ارسال فرم باید مقدار داشته باشد. صفت `required` با عناصر `input` از جمله: `text`, `search`, `url`, `tel`, `email`, `password`, `date pickers`, `number`, `checkbox`, `radio` و `file` کار می کند.

مثال : صفت required

1. Username: `<input type="text" name="username" required>`

## [خودتان امتحان کنید <<](#)

### صفت step

صفت `step` فواصل عددی را در عنصر `<input>` مشخص می کند. به عنوان مثال اگر `step="3"` باشد فواصل عددی برابر با : `۰`, `۳`, `۶` خواهد بود.

نکته : ویژگی `step` را می توان با ویژگی های `min` و `max` برای ایجاد طیف وسیعی از مقادیر قانونی مورد استفاده قرار داد.

ویژگی `step` را می توان با انواع ورودی های `input` از جمله : `number`, `range`, `date`, `datetime-local`, `month`, `time` و `week` مورد استفاده قرار داد.

مثال : صفت step

1. `<input type="number" name="points" step="3">`

## [خودتان امتحان کنید <<](#)

### کلام آخر

در بخش قبلی انواع فیلدهای `input` و کاربردهای هر یک را ارائه دادیم ، همانطور که عناصر `input` گستره ی وسیعی را شامل می شوند، صفات `input` در `html` نیز از تنوع و کاربرد گسترده ای برخوردار است که ما در این بخش به آن پرداختیم.





## فصل سوم : HTML5

### جلسه ۳۷ : مقدمه ای بر HTML5

#### مقدمه

با عرض سلام خدمت کاربران سایت [پی وی لرن](#) . در مباحث قبلی که در این سایت در دوره آموزشهای HTML ، ارائه شد، ما به مبحث [بررسی ویژگی های HTML و XHTML](#) ، بررسی تفاوت ، مزایا و معایب آن پرداختیم. در ادامه در این بخش نیز ، در ادامه آن مبحث ، قصد داریم به بحث [آشنایی با ویژگی های HTML5](#) ، و قابلیت های این نسخه ی HTML ، بپردازیم.

در مطالب قبلی بیشتر در مورد ویژگی ها و مزایای XHTML ، و شکل ساختاری کدهای آن مطالبی را مورد بررسی قرار دادیم. در ادامه ی این بخش نیز به موضوع آشنایی با ویژگی های HTML5 ، می پردازیم.



html5

#### آشنایی با ویژگی های HTML5

اکنون به بررسی ویژگی های و ساختارهای جدید در HTML5 ، می پردازیم ، که در نسخه های قبلی وجود نداشت:

#### ساختار تگ DOCTYPE

ساختار تگ DOCTYPE در HTML5 ، بسیار ساده است.

مثال :

1. `<!DOCTYPE html>`

## ساختار تعریف encoding

تعریف کاراکتر encoding نیز بسیار ساده شده است:

مثال :

1. `<meta charset="UTF-8">`

مثال : مثال-1

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="UTF-8">
5. <title>Title of the document</title>
6. </head>
7.
8. <body>
9. Content of the document.....
10. </body>
11.
12. </html>
```

## [خودتان امتحان کنید](#)

## تگ های جدید در HTML5

تگ های جدیدی که در HTML5 تعریف شده است ، عبارتند از:

تگ های معنایی جدید: <header> ، <footer> ، <article> و <section>.

صفات جدید: range, number, date, time, calendar.

عناصر گرافیکی جدید: <svg> و <canvas>.

عناصر چند رسانه ای جدید: <audio> و <video>.

## API (رابط برنامه ها) جدید

API های جدیدی که در HTML5 ، ارائه شده است عبارتند از :

- HTML Geolocation
- HTML Drag and Drop
- HTML Local Storage
- HTML Application Cache
- HTML Web Workers
- HTML SSE

## تاریخچه ی HTML

از روزهای ابتدایی توسعه اینترنت، نسخه های بسیاری از HTML ارائه شده است:

سال	نسخه
۱۹۸۹	Tim Berners-Lee invented www
۱۹۹۱	Tim Berners-Lee invented HTML
۱۹۹۳	Dave Raggett drafted HTML+
۱۹۹۵	HTML Working Group defined HTML 2.0
۱۹۹۷	W3C Recommendation: HTML 3.2
۱۹۹۹	W3C Recommendation: HTML 4.01
۲۰۰۰	W3C Recommendation: XHTML 1.0

WHATWG HTML5 First Public Draft	۲۰۰۸
WHATWG HTML5 Living Standard	۲۰۱۲
W3C Recommendation: HTML5	۲۰۱۴
W3C Candidate Recommendation: HTML 5.1	۲۰۱۶
W3C Recommendation: HTML5.1 2nd Edition	۲۰۱۷
W3C Recommendation: HTML5.2	۲۰۱۷

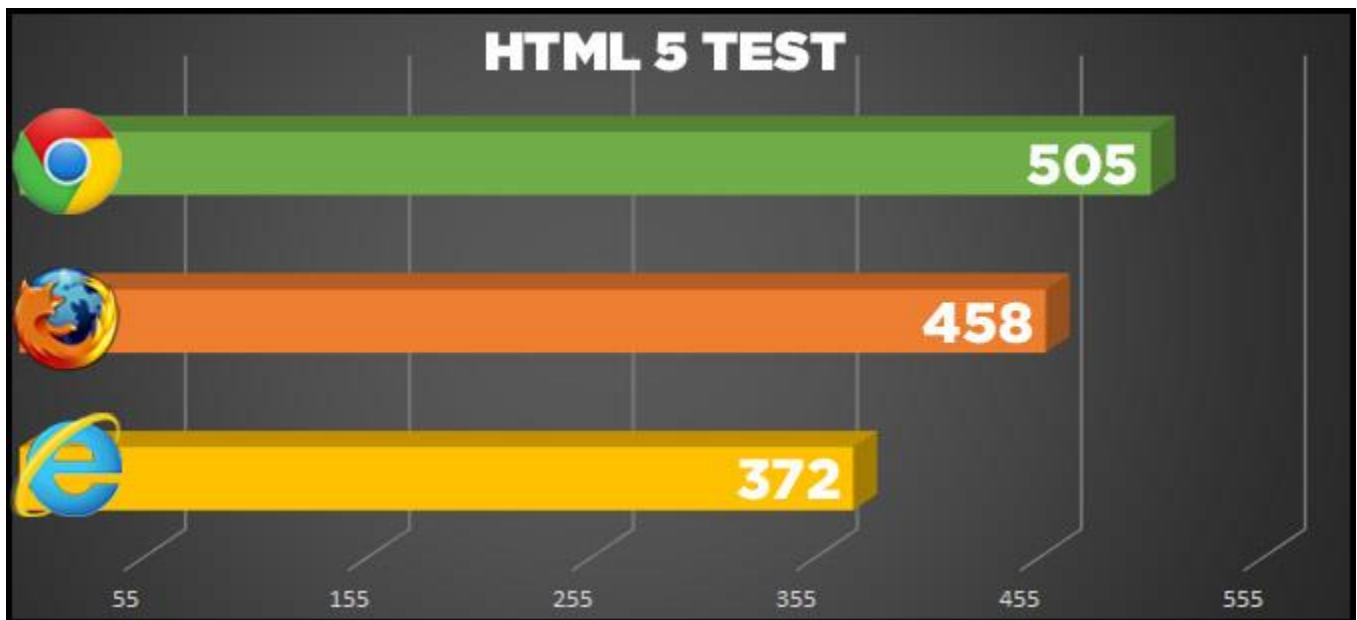
## کلام آخر

در اولین مبحث این فصل به **آشنایی با ویژگی های HTML5** پرداختیم، در بخش های بعدی این فصل شما را با هر یک از عناصر جدید در HTML5 و ساختار های این نسخه آشنا خواهیم کرد.

## جلسه ۳۸ : پشتیبانی HTML5 از مرورگرها

### مقدمه

با عرض سلام خدمت کاربران سایت [پی وی لرن](#) . در مباحثی که در موضوعات قبلی، در دوره آموزشهای HTML ، ارائه شد، به [آشنایی با HTML5](#) ، ویژگی ها و قابلیت های جدید این نسخه ، پرداختیم. همچنین [مزایا و معایب و تفاوت های بین HTML و XHTML](#) را مورد بررسی و تحلیل قرار دادیم. اکنون در این بخش قصد داریم به بحث [آشنایی با پشتیبانی HTML5 از مرورگرها و HTML5 در مرورگرهای قدیمی](#)، بپردازیم. تمام مرورگرهای جدید ، از HTML5 ، پشتیبانی می کنند. و مرورگرهای قدیمی را نیز می توان با HTML5 ، سازگار کرد. در ادامه این بخش آموزشی ، به بررسی موضوع آشنایی با پشتیبانی HTML5 از مرورگرها ، می پردازیم.



html

### آشنایی با پشتیبانی HTML5 از مرورگرها

همانطور که اشاره شده HTML5 ، از تمام مرورگرها ی مدرن پشتیبانی می کند. علاوه بر این، تمام مرورگرهای قدیمی و جدید، به طور خودکار عناصر HTML نامعلوم را به عنوان عناصر HTML درون خطی (inline) اداره می کنند.

به همین دلیل شما می توانید، به مرورگرهای قدیمی تر عناصر HTML ناشناخته را ، بشناسانید. شما حتی می توانید برای مرورگر IE6 ، عناصر HTML جدید و ناشناخته را ، قابل اجرا در آورید.

### تعریف عناصر semantic به عنوان عناصر block

HTML5 هشت عنصر semantic جدید را تعریف کرده است. همه ی آن ها از نوع عناصر block هستند، برای اطمینان از رفتار درست، این عناصر، در مرورگرهای قدیمی تر ، شما می توانید خصوصیت **display** ، تمام این عناصر را بصورت **block** تنظیم کنید:

مثال :

```
1. header, section, footer, aside, nav, main, article {  
2.     display: block;  
3. }
```

## افزودن عناصر جدید به html

شما می توانید عناصر html جدیدی را به هر کدام از صفحات html ، با استفاده از browser trick ، اضافه کنید.

مثال: در این مثال یک عنصر جدید به نام <myHero> به یک صفحه html اضافه شده و برای آن style هم تعریف شده است:

مثال : افزودن عنصر جدید

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <script>document.createElement("myHero")</script>
5. <style>
6. myHero {
7.     display: block;
8.     background-color: #dddddd;
9.     padding: 50px;
10.    font-size: 30px;
11. }
12. </style>
13. </head>
14. <body>
15.
16. <h1>A Heading</h1>
17. <myHero>My Hero Element</myHero>
18.
19. </body>
20. </html>
```

### [خودتان امتحان کنید<>](#)

در مثال بالا ، تابع جاوااسکریپت (document.createElement("myHero")) ، برای افزودن عناصر جدید در مرورگرهای IE 9 ، و قدیمی تر از آن ، نیاز است.

## مشکل مرورگر ۸ Internet Explorer

شما می توانید از راه حل توصیف شده در بالا برای تمام عناصر HTML5 جدید استفاده کنید.

با این وجود مرورگر IE8 (و قدیمی تر) اجازه تعریف style ، برای عناصر جدید را نمی دهد.

خوشبختانه ابزار HTML5Shiv ، برای حل این مشکل ایجاد شد. HTML5Shiv ، یک راه حل جاوااسکریپت است که style تعریف شده برای عناصر html را ، در مرورگرهای IE9 و قدیمی تر فعال می کند.

## ساختار HTML5Shiv

HTML5Shiv در html در تگ <head> ، قرار می گیرد.

HTML5Shiv ، یک فایل جاوااسکریپت است، که در داخل تگ <script> ، تعریف می شود.

بهتر است، شما زمانیکه از تگ های <nav> ، <aside> ، <section> ، <article> و <footer> ، استفاده می کنید، از HTML5Shiv ، استفاده کنید.

## مثال :

```

1. <head>
2.   <!--[if lt IE 9]>
3.     <script src="/js/html5shiv.js"></script>
4.   <![endif]-->
5. </head>

```

مثال- برای استفاده از HTML5Shiv، می توانید نسخه های جدید تر این ابزار را از سایت CDN دریافت کنید. در مثال زیر با چگونگی استفاده از این ابزار در کد html بیشتر آشنا می شوید:

## مثال : مثال-2

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="UTF-8">
5. <!--[if lt IE 9]>
6.   <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
7. <![endif]-->
8. </head>
9. <body>
10.
11. <section>
12.
13. <h1>Famous Cities</h1>
14.
15. <article>
16. <h2>London</h2>
17. <p>London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.</p>
18. </article>
19.
20. <article>
21. <h2>Paris</h2>
22. <p>Paris is the capital and most populous city of France.</p>
23. </article>
24.
25. <article>
26. <h2>Tokyo</h2>
27. <p>Tokyo is the capital of Japan, the center of the Greater Tokyo Area, and the most populous metropolitan area in the world.</p>
28. </article>
29.
30. </section>
31.
32. </body>
33. </html>

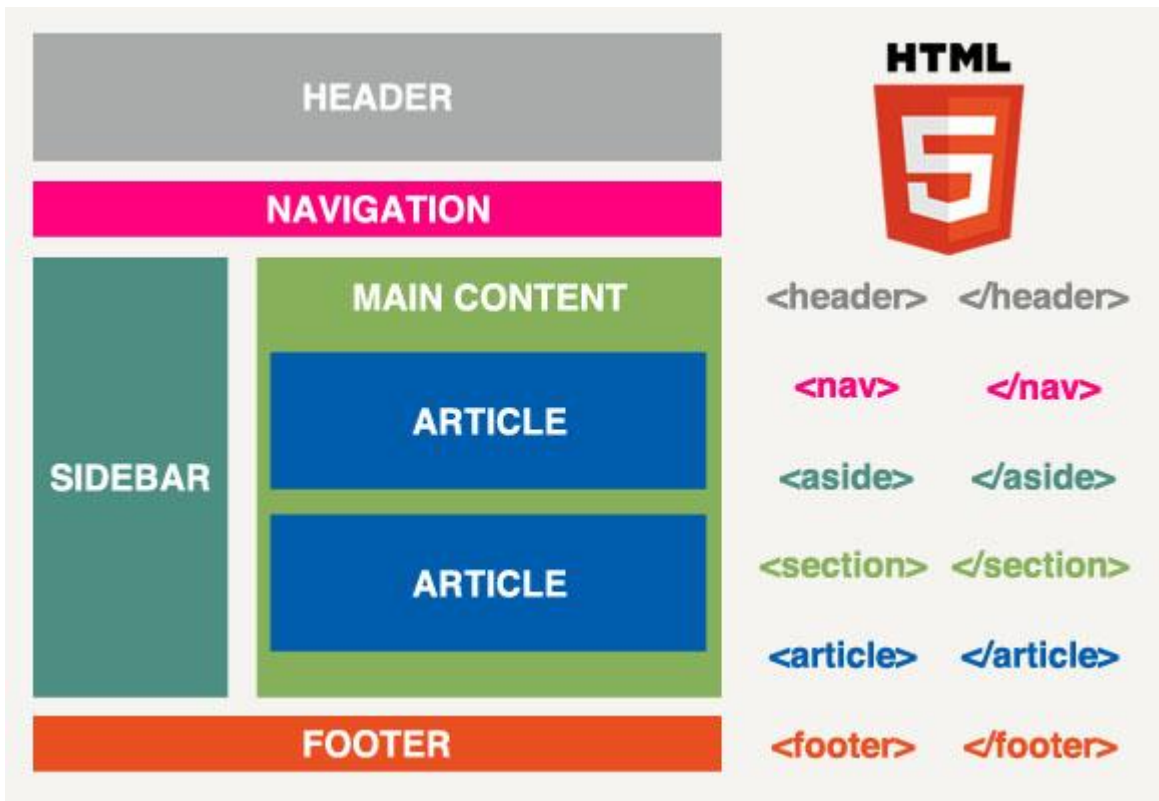
```

[خودتان امتحان کنید<<](#)



### مقدمه

با عرض سلام خدمت کاربران سایت [پی وی لرن](#) . در مباحث قبلی در مورد HTML5، به موضوعات [آشنایی با HTML5](#) و همچنین در ادامه آموزشها ی html ، به موضوع [بشتریانی HTML5](#) ، [در مرورگرهای قدیمی](#) ، پرداختیم. اکنون در ادامه در این بخش آموزشی ، قصد داریم به آشنایی با عناصر جدید که در html5 اضافه شده اند ، و کاربرد های هر یک آشنا شویم. در این بخش عناصر جدید را در دسته های جداگانه ، با کاربرد های هر یک قرار داده ایم. در ادامه به بررسی کامل ویژگی ها و **کاربرد عناصر جدید در html5** یا **قابلیت های جدید در html5** ، می پردازیم.



HTML5-elements

### عناصر جدید در html5

در این پست آموزشی ما ، لیستی از عناصر جدید را به تفکیک دسته ، و کاربردهای هر یک قرار داده ایم.

کاربرد	تگ
یک مقاله را در یک سند تعریف می کند.	<article>
مطالب جدا از محتوای صفحه را، تعریف می کند.	<aside>
بخشی از متن را که ممکن است خارج از متن دیگری، فرمت شده باشد، جدا می کند.	<bdi>
مشخصات اضافی را که کاربر می تواند مشاهده یا پنهان کند تعریف می کند.	<details>
برای تعریف dialog box ، مورد استفاده قرار می گیرد.	<dialog>
یک عنوان برای تگ <figure> ، تعریف می کند.	<figcaption>
برای تعریف محتوی مستقل مورد استفاده قرار می گیرد.	<figure>
برای تعریف قسمت پاصفحه در صفحه وب مرود استفاده قرار می گیرد.	<footer>
برای تعریف سر صفحه، در صفحه وب مورد استفاده قرار می گیرد.	<header>

برای تعریف محتوی اصلی صفحه استفاده می شود.	<main>
بریا علامتدار کردن تمام یا قسمتی از یک متن استفاده می شود.	<mark>
واحد اندازه گیری را در محدوده ی خاصی تعریف می کند.	<meter>
برای تعریف لینک های navigation استفاده می شود.	<nav>
برای نمایش پیشرفت یک فرآیند، استفاده می شود.	<progress>
در مرورگرهایی که حاشیه نویسی را پشتیبانی نمی کنند، استفاده می شود.	<rp>
توضیح / تلفظ شخصیت ها را تعریف می کند.	<rt>
حاشیه نویسی ( ruby برای تایپوگرافی شرق آسیا) را تعریف می کند.	<ruby>
بخشی از یک سند را تعریف می کند.	<section>
یک عنوان را برای تگ <details> تعریف می کند.	<summary>
برای تعریف تاریخ و زمان استفاده می شود.	<time>

وضعیت line-break را فعال می کند.

<wbr>

## عناصر جدید فرم

کاربرد	تگ
لیستی از گزینه های پیش تعیین شده برای کنترل های ورودی را مشخص می کند	<datalist>
برای نمایش ، نتیجه محاسبات استفاده می شود.	<output>

## تگ های ورودی جدید

صفات جدید در تگ ها ورودی	تگ های ورودی جدید
<ul style="list-style-type: none"><li>autocomplete</li><li>autofocus</li><li>form</li><li>formaction</li><li>formenctype</li><li>formmethod</li><li>formnovalidate</li><li>formtarget</li><li>height and width</li><li>list</li><li>min and max</li><li>multiple</li></ul>	<ul style="list-style-type: none"><li>color</li><li>date</li><li>datetime</li><li>datetime-local</li><li>email</li><li>month</li><li>number</li><li>range</li><li>search</li><li>tel</li><li>time</li><li>url</li></ul>

- pattern (regexp)
- placeholder
- required
- step

- week

## ساختارهای جدید برای تعریف صفات

HTML5 ، به شما اجازه می دهد ۴ ساختار مختلف را برای تعریف صفات استفاده کنید.

مثال:

- `<input type="text" value="John" disabled>`
- `<input type="text" value=John>`
- `<input type="text" value="John Doe">`
- `<input type="text" value='John Doe'>`

## تگ های گرافیکی جدید

کاربرد	تگ
برای رسم اشکال هندسی با استفاده از جاوااسکریپت مورد استفاده قرار می گیرد.	<code>&lt;canvas&gt;</code>
برای رسم اشکال مقیاس پذیر استفاده می شود.	<code>&lt;svg&gt;</code>

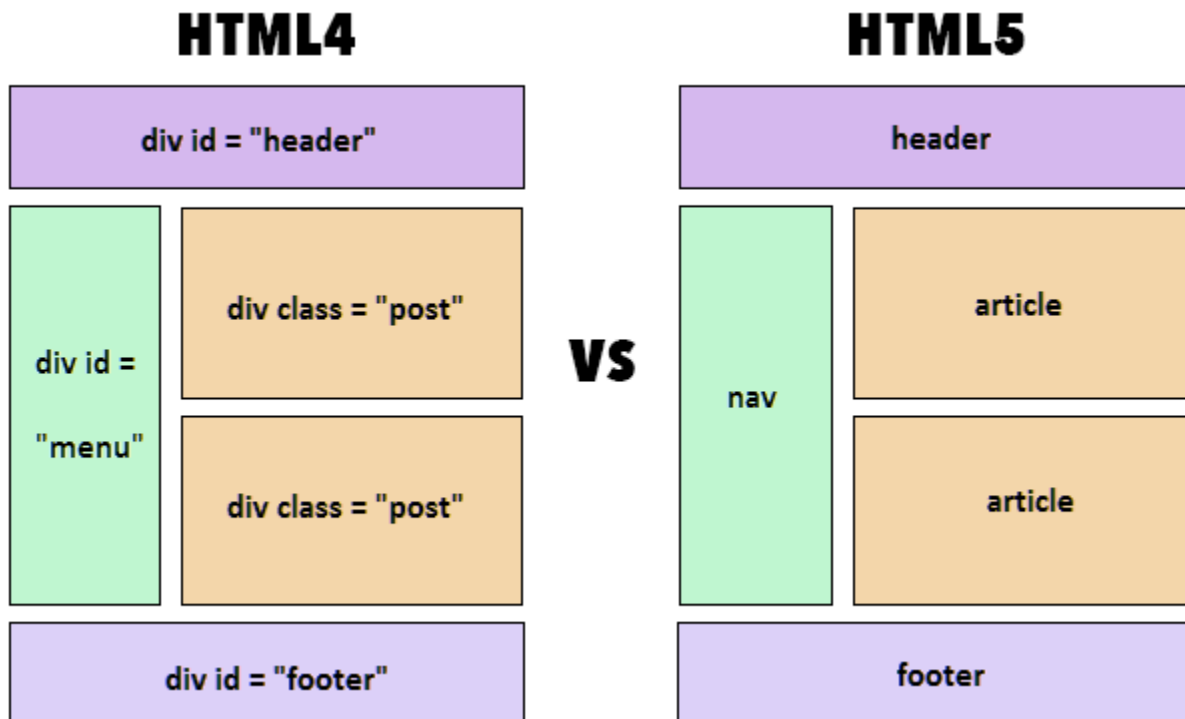
## عناصر چند رسانه ای جدید

کاربرد	تگ
برای تعریف محتوی صوتی استفاده می شود.	<audio>
به عنوان یک ظرف برای برنامه های خارج از html استفاده می شود.	<embed>
منابع چند رسانه ای را تعریف می کند	<source>
برای تعریف track های متنی در عناصر چند رسانه ای استفاده می شود.	<track>
برای تعریف محتوی ویدئویی استفاده می شود.	<video>



### مقدمه

با عرض سلام خدمت کاربران سایت [پی وی لرن](#) . در ادامه ی آموزش های html ، وارد مباحث **html5** شدیم ، و مباحث مهم و کاربردی از جمله [آشنایی با html5](#) و همچنین [عناصر جدیدی در html5](#) ، را مورد بررسی قرار دادیم. و تقریباً با نسخه html5 و تمام عناصر جدید، با قابلیت های جدیدی که در این نسخه ارائه شده است ، آشنا شدید. در ادامه بحث html5 ، در این بخش ، قصد داریم به موضوع **مهاجرت از HTML4 به HTML5** یا **تبدیل HTML4 به HTML5** ، بپردازیم. این بخش، نشان می دهد. که چگونه یک صفحه HTML4 را به یک صفحه HTML5 تبدیل کنید، بدون اینکه هیچ چیزی را از محتوا یا ساختار اولیه حذف کنید. همچنین در این بخش ما به بررسی ویژگی ها و تفاوت های بین نسخه ی HTML4 و HTML5 ، می پردازیم.



تفاوت های بین نسخه ی HTML4 و HTML5

### مهاجرت از HTML4 به HTML5

برخی از بخش ها به سادگی از نسخه HTML4 قابل تبدیل به HTML5 است:

HTML4	HTML5
<code>&lt;div id="header"&gt;</code>	<code>&lt;header&gt;</code>



<div id="menu">	<nav>
<div id="content">	<section>
<div class="article">	<article>
<div id="footer">	<footer>

## صفحه وب نوع HTML4

مثال : نمونه صفحه HTML4

```

1. <!DOCTYPE HTML PUBLIC "-
//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
2. <html lang="en">
3. <head>
4. <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
5. <title>HTML4</title>
6. <style>
7. body {
8.     font-family: Verdana,sans-serif;
9.     font-size: 0.9em;
10. }
11.
12. div#header, div#footer {
13.     padding: 10px;
14.     color: white;
15.     background-color: black;
16. }
17.
18. div#content {
19.     margin: 5px;
20.     padding: 10px;
21.     background-color: lightgrey;
22. }
23.
24. div.article {
25.     margin: 5px;
26.     padding: 10px;
27.     background-color: white;
28. }
29.
30. div#menu ul {
31.     padding: 0;
32. }

```

```

33.
34. div#menu ul li {
35.     display: inline;
36.     margin: 5px;
37. }
38. </style>
39. </head>
40. <body>
41.
42. <div id="header">
43.   <h1>Monday Times</h1>
44. </div>
45.
46. <div id="menu">
47.   <ul>
48.     <li>News</li>
49.     <li>Sports</li>
50.     <li>Weather</li>
51.   </ul>
52. </div>
53.
54. <div id="content">
55.   <h2>News Section</h2>
56.   <div class="article">
57.     <h2>News Article</h2>
58.     <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in porta lorem. Morbi c
ondimentum est nibh, et consectetur tortor feugiat at.</p>
59.   </div>
60.   <div class="article">
61.     <h2>News Article</h2>
62.     <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in porta lorem. Morbi c
ondimentum est nibh, et consectetur tortor feugiat at.</p>
63.   </div>
64. </div>
65.
66. <div id="footer">
67.   <p>&copy; 2016 Monday Times. All rights reserved.</p>
68. </div>
69.
70. </body>
71. </html>

```

[خودتان امتحان کنید <<](#)

## تغییر دادن تگ DOCTYPE به html5

ساختار تگ DOCTYPE در: html4:

مثال :

```

1. <!DOCTYPE HTML PUBLIC "-//
//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

```

در: html5

مثال : افزودن doctype به مثال قبل

1. <!DOCTYPE html>

## خودتان امتحان کنید<<

### تغییر encoding به html5

ساختار encoding در html4:

مثال :

1. <meta http-equiv="Content-Type" content="text/html; charset=utf-8">

ساختار encoding در html5:

مثال : افزودن encoding

1. [http://pvlearn.com/codeLab/?fileName=tryhtml5\\_migrate\\_3](http://pvlearn.com/codeLab/?fileName=tryhtml5_migrate_3)

## خودتان امتحان کنید<<

### تبدیل عناصر semantic به html5

کدهای CSS برای تعریف عناصر semantic در html4:

مثال :

```
1. body {
2.     font-family: Verdana,sans-serif;
3.     font-size: 0.9em;
4. }
5.
6. div#header, div#footer {
7.     padding: 10px;
8.     color: white;
9.     background-color: black;
10. }
11.
12. div#content {
13.     margin: 5px;
14.     padding: 10px;
15.     background-color: lightgrey;
16. }
17.
18. div.article {
19.     margin: 5px;
20.     padding: 10px;
21.     background-color: white;
22. }
23.
24. div#menu ul {
25.     padding: 0;
26. }
```

```
27.  
28. div#menu ul li {  
29.     display: inline;  
30.     margin: 5px;  
31. }
```

کدهای CSS عناصر معنایی در html5:

مثال :

```
1. body {  
2.     font-family: Verdana,sans-serif;  
3.     font-size: 0.9em;  
4. }  
5.  
6. header, footer {  
7.     padding: 10px;  
8.     color: white;  
9.     background-color: black;  
10. }  
11.  
12. section {  
13.     margin: 5px;  
14.     padding: 10px;  
15.     background-color: lightgrey;  
16. }  
17.  
18. article {  
19.     margin: 5px;  
20.     padding: 10px;  
21.     background-color: white;  
22. }  
23.  
24. nav ul {  
25.     padding: 0;  
26. }  
27.  
28. nav ul li {  
29.     display: inline;  
30.     margin: 5px;  
31. }
```

در نهایت عناصر semantic نسخه html5 را جایگزین می کنیم:

مثال : نمونه ی کامل HTML5

```
1. <body>  
2.  
3. <header>  
4. <h1>Monday Times</h1>  
5. </header>  
6.  
7. <nav>  
8. <ul>  
9. <li>News</li>
```

```

10. <li>Sports</li>
11. <li>Weather</li>
12. </ul>
13. </nav>
14.
15. <section>
16. <h2>News Section</h2>
17. <article>
18. <h2>News Article</h2>
19. <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in porta lorem. Morbi condimentum est nibh, et consectetur tortor feugiat at.</p>
20. </article>
21. <article>
22. <h2>News Article</h2>
23. <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in porta lorem. Morbi condimentum est nibh, et consectetur tortor feugiat at.</p>
24. </article>
25. </section>
26.
27. <footer>
28. <p>© 2014 Monday Times. All rights reserved.</p>
29. </footer>
30.
31. </body>

```

## [خودتان امتحان کنید](#)

### تفاوت بین تگ های <article> ، <section> و <div>

در استاندارد HTML5، تگ <section> برای عناصر html با ویژگی block تعریف شده است، تگ <article>، نیز برای تعریف محتوی مستقل در عناصر html با ویژگی block تعریف شده است.

عناصر <div>، برای عناصر زیر مجموعه با ویژگی block، تعریف شده است، در مثال بالا، ما <section> را به عنوان یک ظرف برای مرتبط <articles> استفاده کردیم.

اما، ما همچنین می توانیم <article> را به عنوان یک ظرف برای مقالات نیز استفاده کنیم:

### مثال : استفاده از article

```

1. <article>
2.
3. <h2>Famous Cities</h2>
4.
5. <article>
6. <h2>London</h2>
7. <p>London is the capital city of England. It is the most populous city in the United Kingdom,
8. with a metropolitan area of over 13 million inhabitants.</p>
9. </article>
10.
11. <article>
12. <h2>Paris</h2>
13. <p>Paris is the capital and most populous city of France.</p>
14. </article>
15.
16. <article>
17. <h2>Tokyo</h2>
18. <p>Tokyo is the capital of Japan, the center of the Greater Tokyo Area,
19. and the most populous metropolitan area in the world.</p>
20. </article>

```

- 21.
22. `</article>`

[خودتان امتحان کنید<<](#)

مثال : استفاده از article

1. `<article>`
- 2.
3. `<h2>Famous Cities</h2>`
- 4.
5. `<div class="city">`
6. `<h2>London</h2>`
7. `<p>London is the capital city of England. It is the most populous city in the United Kingdom,`
8. `with a metropolitan area of over 13 million inhabitants.</p>`
9. `</div>`
- 10.
11. `<div class="city">`
12. `<h2>Paris</h2>`
13. `<p>Paris is the capital and most populous city of France.</p>`
14. `</div>`
- 15.
16. `<div class="city">`
17. `<h2>Tokyo</h2>`
18. `<p>Tokyo is the capital of Japan, the center of the Greater Tokyo Area,`
19. `and the most populous metropolitan area in the world.</p>`
20. `</div>`
- 21.
22. `</article>`

[خودتان امتحان کنید<<](#)

مثال : استفاده از article

1. `<article>`
- 2.
3. `<section>`
4. `<h2>Famous Cities</h2>`
- 5.
6. `<div class="city">`
7. `<h2>London</h2>`
8. `<p>London is the capital city of England. It is the most populous city in the United Kingdom,`
9. `with a metropolitan area of over 13 million inhabitants.</p>`
10. `</div>`
- 11.
12. `<div class="city">`
13. `<h2>Paris</h2>`
14. `<p>Paris is the capital and most populous city of France.</p>`
15. `</div>`
- 16.

```
17. <div class="city">
18. <h2>Tokyo</h2>
19. <p>Tokyo is the capital of Japan, the center of the Greater Tokyo Area,
20. and the most populous metropolitan area in the world.</p>
21. </div>
22. </section>
23.
24. <section>
25. <h2>Famous Countries</h2>
26.
27. <div class="country">
28. <h2>England</h2>
29. <p>London is the capital city of England. It is the most populous city in the United Kingdom,
30. with a metropolitan area of over 13 million inhabitants.</p>
31. </div>
32.
33. <div class="country">
34. <h2>France</h2>
35. <p>Paris is the capital and most populous city of France.</p>
36. </div>
37.
38. <div class="country">
39. <h2>Japan</h2>
40. <p>Tokyo is the capital of Japan, the center of the Greater Tokyo Area,
41. and the most populous metropolitan area in the world.</p>
42. </div>
43. </section>
44.
45. </article>
```

[خودتان امتحان کنید<<](#)

## کلام آخر

در مباحث قبلی فصل به مرور ویژگی ها و عناصر معنایی در HTML5، مزایای استفاده از عناصر HTML5 پرداختیم، در این بخش به طور کامل روال تبدیل اسناد HTML4 به HTML5 و یا به اصطلاح مهاجرت از HTML4 به HTML5 را ارائه کردیم.

### مقدمه

با عرض سلام خدمت کاربران سایت [پی وی لرن](#) . در مطالبی که تاکنون در بحث [آشنایی با html5](#) ، ارائه شده است. با ویژگی ها و عناصر جدید html5 ، آشنا شدید، همچنین تا مباحثی نظیر مهاجرت از HTML4 به HTML5 نیز پیش رفتیم. تاکنون با [ساختار کلی html5](#) و [عناصر جدید](#) این نسخه آشنایی لازم ارائه گردید. و چگونگی تبدیل یک صفحه وب نوع [HTML4 به HTML5](#) ، را نیز فرا گرفتید. اکنون در ادامه بحث html5 ، در این بخش ، قصد داریم برای آشنایی کاملتر شما با html5 ، موضوع [راهنمای کد نویسی صحیح در html5](#) یا [کد نویسی استاندارد در html5](#) ، را در این بخش ارائه دهیم.

```
<div class="container">
  <div class="row">
    <div class="col-md-6 col-lg-8"> <!-- _____ BEGIN NAVIGATION
      <nav id="nav" role="navigation">
        <ul>
          <li><a href="index.html">Home</a></li>
          <li><a href="home-events.html">Home Events</a></li>
          <li><a href="multi-col-menu.html">Multiple Column Men
          <li class="has-children"> <a href="#" class="current">
            <ul>
              <li><a href="tall-button-header.html">Tall But
              <li><a href="image-logo.html">Image Logo</a></li>
              <li class="active"><a href="tall-logo.html">Ta
            </ul>
          </li>
          <li class="has-children"> <a href="#">Carousels</a>
            <ul>
              <li><a href="variable-width-slider.html">Variab
              <li><a href="variable-width-slider.html">Testimoni
```

راهنمای کد نویسی صحیح در html5

### راهنمای کد نویسی صحیح در html5

توسعه دهندگان وب اغلب در مورد style (سبک) و نحو کد گذاری برای استفاده در HTML روش مشخصی ندارند.

بین سالهای ۲۰۰۰ تا ۲۰۱۰ تعداد زیادی از توسعه دهندگان وب ، از HTML به XHTML ، مهاجرت کردند.

تا اینکه HTML5 ، ارائه شد.

### هوشمند بودن و اثبات آینده

خواندن و استفاده از style و طراحی صفحه وب شما را برای دیگران آسانتر می کند. در آینده ممکن است برنامه های XML، سعی کنند صفحه html شما را بخوانند.



## نوع سند

همیشه نوع سند را به عنوان خط اول در سند خود تایپ کنید:

مثال :

```
1. <!DOCTYPE html>
```

نام عناصر html را همیشه بصورت کوچک تایپ کنید:

نادرست:

مثال :

```
1. <SECTION>
2. <p>This is a paragraph.</p>
3. </SECTION>
```

بسیار بد :

مثال :

```
1. <Section>
2. <p>This is a paragraph.</p>
3. </SECTION>
```

قابل قبول:

مثال :

```
1. <section>
2. <p>This is a paragraph.</p>
3. </section>
```

تمام تگ های html خود را ببندید

نادرست:

مثال :

```
1. <section>
2. <p>This is a paragraph.
3. <p>This is a paragraph.
```

4. `</section>`

صحیح:

مثال :

```
1. <section>
2.   <p>This is a paragraph.</p>
3.   <p>This is a paragraph.</p>
4. </section>
```

## بستن عناصر خالی HTML

در HTML5 بستن عناصر خالی اختیاری است :

مثال :

```
1. <meta charset="utf-8">
```

در HTML5 بستن عناصر خالی اختیاری است :

مثال :

```
1. <meta charset="utf-8" />
```

## در استفاده از صفات از حروف کوچک استفاده کنید

در HTML5، می توانید هم از حروف کوچک و هم از حروف بزرگ در تعریف صفات استفاده کنید.

اما توصیه می شود که از حروف کوچک استفاده کنید، زیرا:

- مخلوط کردن نام های بزرگ و کوچک بد است.
- توسعه دهندگان معمولا از نام های کوچک (از جمله XHTML استفاده می کنند).
- حروف کوچکتر بهتر به چشم می آیند.
- نوشتن حروف کوچکتر ساده تر است.

حالت ناردست:

مثال :

```
1. <div CLASS="menu">
```

صحیح:

مثال :

1. `<div class="menu">`

## مقادیر صفات را در داخل نقل قول ("") تعریف کنید

HTML5 ف به شما اجازه می دهد که صفات را بدون نقل قول ("") نیز تعریف کنید.

اما توصیه می شود که از نقل قول استفاده کنید. زیرا:

- مخلوط کردن مقادیر بزرگ و کوچک، بد است.
- مقادیر دارای نقل قول آسان تر است.
- شما باید از نقل قول ها استفاده کنید اگر مقدار صفت، حاوی فضاهای خالی باشد.

حالت بسیار بد:

مثال :

1. `<table class=table striped>`

حالت بد و نادرست:

مثال :

1. `<table class=striped>`

حالت صحیح:

مثال :

1. `<table class="striped">`

## صفات عکس ها

همیشه از صفت **alt** در عکس ها استفاده کنید. چراکه اگر به هر دلیلی عکس های وب سایت شما بارگزاری نشود و در مرورگر برای کاربران نمایش داده نشود. یک متن جایگزین به جای آن تصویر نمایش داده شود. همیشه ویژگی های **width** و **height** را نیز در عکس ها استفاده کنید. این باعث می شود که تصویر وب سایت شما هنگام بارگزاری سو سو نزنند.

حالت نادرست:

مثال :

```
1. 
```

صحیح:

مثال :

```
1. 
```

## از space کم و برابر بین صفات در عناصر استفاده کنید

HTML5 اجازه می دهد بین صفات در عناصر از space استفاده کنید. اما توجه داشته باشید که space کمتر، باعث خوانایی بهتر می شود.

مثال- توصیه نمی شود:

مثال :

```
1. <link rel = "stylesheet" href = "styles.css">
```

مثال ۲- خوب است و توصیه می شود:

مثال :

```
1. <link rel="stylesheet" href="styles.css">
```

## از خط کدهای طولانی اجتناب کنید

زمانیکه از ادیتور های html استفاده می کنید، اگر خطوط کدها طولانی شود مجبورید که از scroll راست و چپ ، استفاده کنید.

سعی کنید که طول خط کدها را از ۸۰ کاراکتر کمتر کنید.

## خط های خالی

سعی کنید بدون دلیل ، خط های خالی را به کد اضافه نکنید.

سعی کنید از خطوط خالی برای جداسازی بلوک های کد بزرگ استفاده کنید.

مثالی از استفاده از خطوط خالی غیر ضروری:

مثال :

```
1. <body>
2.
3. <h1>Famous Cities</h1>
4.
5. <h2>Tokyo</h2>
6.
7. <p>
8. Tokyo is the capital of Japan, the center of the Greater Tokyo Area,
9. and the most populous metropolitan area in the world.
10. It is the seat of the Japanese government and the Imperial Palace,
11. and the home of the Japanese Imperial Family.
12. </p>
13.
14. </body>
```

مثال ۲: اصلاح شده ی کد مثال قبل:

مثال :

```
1. <body>
2.
3. <h1>Famous Cities</h1>
4.
5. <h2>Tokyo</h2>
6. <p>Tokyo is the capital of Japan, the center of the Greater Tokyo Area,
7. and the most populous metropolitan area in the world.
8. It is the seat of the Japanese government and the Imperial Palace,
9. and the home of the Japanese Imperial Family.</p>
10.
11. </body>
```

مثال جدول:

مثال :

```
1. <table>
2.   <tr>
3.     <th>Name</th>
4.     <th>Description</th>
5.   </tr>
6.   <tr>
7.     <td>A</td>
8.     <td>Description of A</td>
9.   </tr>
10.  <tr>
11.   <td>B</td>
12.   <td>Description of B</td>
13. </tr>
14. </table>
```

مثال لیست:

مثال :

```
1. <ol>
2.   <li>London</li>
3.   <li>Paris</li>
4.   <li>Tokyo</li>
5. </ol>
```

## حذف تگ های <html> و <body>؟

در HTML5 ، شما می توانید تگ های <html> و <body> را حذف کنید.

مثال : حذف بخش های html و body

```
1. <!DOCTYPE html>
2. <head>
3.   <title>Page Title</title>
4. </head>
5.
6. <h1>This is a heading</h1>
7. <p>This is a paragraph.</p>
```

## [خودتان امتحان کنید<<](#)

با این وجود توصیه می شود که تگ های <html> و <body> را از کد html حذف نکنید.

## حذف تگ <head>

در حالت پیش فرض مرورگر ، تمام عناصر قبل از تگ <body> ، را به عنصر <head> اضافه می کند.

HTML5 به شما اجازه می دهد که عنصر <head> ، را نیز حذف کنید.

```

1. <!DOCTYPE html>
2. <html>
3. <title>Page Title</title>
4.
5. <body>
6.   <h1>This is a heading</h1>
7.   <p>This is a paragraph.</p>
8. </body>
9.
10. </html>

```

## [خودتان امتحان کنید<<](#)

با این وجود ، حذف عنصر <head> ، نیز توصیه نمی شود.

### Meta Data

درج عنصر <title> در HTML5 ضروری است:

#### مثال :

```

1. <title>HTML5 Syntax and Coding Style</title>

```

برای اطمینان از تفسیر مناسب و درست کردن فهرست موتور جستجو، هر دو زبان و رمزگذاری کاراکتر باید در اسناد در اسرع وقت تعریف شوند:

#### مثال :

```

1. <!DOCTYPE html>
2. <html lang="en-US">
3. <head>
4.   <meta charset="UTF-8">
5.   <title>HTML5 Syntax and Coding Style</title>
6. </head>

```

### تنظیمات viewport

HTML5 به طراح وب اجازه می دهد که کنترل کاملی روی viewport ها داشته باشد. این قابلیت ، از طریق تگ <meta> امکانپذیر است. در مطالب آموزشی قبلی اشاره ای داشتیم به [ویژگی responsive](#) ، و توضیحاتی در مورد اعمال این ویژگی بوسیله تگ <meta> ، دادیم.

#### مثال :

1. `<meta name="viewport" content="width=device-width, initial-scale=1.0">`

## توضیحات در HTML

توضیحات کوتاه در یک خط و به صورت زیر نوشته می شود :

مثال :

1. `<!-- This is a comment -->`

توضیحات بیشتر از یک خط نیز به صورت زیر خواهد بود :

مثال :

```
1. <!--  
2.   This is a long comment example. This is a long comment example.  
3.   This is a long comment example. This is a long comment example.  
4. -->
```

## صفحات Style

از سینتکس ساده ی زیر برای لینک شدن به صفحات style استفاده کنید :

مثال :

1. `<link rel="stylesheet" href="styles.css">`

قوانین کوتاه می توانند فشرده شوند مانند این:

مثال :

1. `p.intro {font-family: Verdana; font-size: 16em;}`

همچنین می توان کدهای چندگانه را در چند خط نوشت:

مثال :



```
1. body {
2.   background-color: lightgrey;
3.   font-family: "Arial Black", Helvetica, sans-serif;
4.   font-size: 16em;
5.   color: black;
6. }
```

براکت باز را در همان خط اول انتخابگر قرار دهید. قبل از باز کردن براکت از یک فضای خالی استفاده کنید. از کد نویسی دندانان ای استفاده کنید. بعد از هر مقدار و انتهای کدها، از semicolon استفاده کنید. اگر مقدار شامل فضاهای خالی باشد، فقط نقل قول ها (کوته‌نویس) در اطراف مقادیر استفاده می شود. بستن براکت ها را بر روی یک خط جدید قرار دهید. اجتناب از خطوط بیش از ۸۰ کاراکتر.

## بارگذاری جاوا اسکریپت در HTML

از سینتکس ساده ی زیر برای بارگذاری اسکریپت ها در HTML استفاده کنید :

مثال :

```
1. <script src="myscript.js">
```

دسترسی به عناصر HTML با جاوا اسکریپت ، یک نتیجه از استفاده از استایل های HTML غیرمتعارف می تواند خطاهای جاوا اسکریپت باشد.

این دو عبارت جاوا اسکریپت نتایج مختلفی تولید خواهند کرد:

مثال : مثال کد JS

```
1. var obj = getElementById("Demo")
2.
3. var obj = getElementById("demo")
```

[خودتان امتحان کنید <<](#)

## استفاده از حروف کوچک در نام فایل ها

برخی از وب سرورها (Apache، Unix) حساس به حروف در مورد نام فایل هستند: "london.jpg" قابل دسترسی نیست به عنوان "London.jpg" ، سایر سرورهای وب (مایکروسافت، IIS) حساس به حروف حساس نیستند: "london.jpg" می تواند به عنوان "London.jpg" یا "london.jpg" قابل دسترسی باشد.

اگر از مخلوطی از حروف کوچک و بزرگ استفاده می کنید، باید بسیار سازگار باشید، اگر از یک حساس به حساس به یک سرور حساس حرکت می کنید، حتی خطاهای کوچک وب شما را از بین می برد!

برای جلوگیری از این مشکلات، همیشه از نام پرونده های پرونده استفاده کنید.

## پسوند فایل ها

فایل های HTML باید یک پسوند .html یا .htm داشته باشند ، فایل های CSS باید یک پسوند CSS داشته باشند.

فایل های جاوا اسکریپت باید دارای پسوند .js باشد.

## تفاوت بین .html و .htm

هیچ تفاوتی میان افزونه های .html و .htm وجود ندارد. هر دو به عنوان HTML توسط هر مرورگر وب یا سرور وب مورد استفاده قرار می گیرند.

تفاوت ها فرهنگی هستند:

.htm از سیستم های DOS اولیه است که در آن سیستم برنامه های افزودنی را به ۳ کاراکتر محدود می کند.

.html از سیستم عامل های یونیکس است که این محدودیت را نداشتند.

## تفاوت های تکنیکی

هنگامی که یک URL نام فایل را مشخص نمی کند (مثل <https://www.pvlearn.com/css>)، سرور یک نام پرونده پیش فرض را باز می کند. نام پرونده های معمول پیش فرض default.html، index.htm، index.html و default.htm است.

اگر سرور شما فقط با نام index.html به عنوان پیش فرض نام فایل پیکربندی شود، فایل شما باید به نام index.html باشد، نه «index.htm»

با این حال، سرورها را می توان با بیش از یک نام پرونده پیش فرض پیکربندی کرد، و به طور معمول می توانید به عنوان نام پرونده های پیش فرض به همان صورت که نیاز دارید، تنظیم کنید.

به هر حال، فرمت کامل فایل های .html، HTML است و هیچ دلیلی برای استفاده از آن وجود ندارد.

## کلام آخر

HTML5 به عنوان جدیدتری نسخه ی معرفی شده از HTML قوانین و سینتکس خاصی را برای کدنویسی دنبال می کند که باید آن ها را رعایت کنیم، از این رو در این بخش به بررسی موارد کامل **کد نویسی صحیح در html5** پرداختیم.

## فصل چهارم : گرافیک در HTML

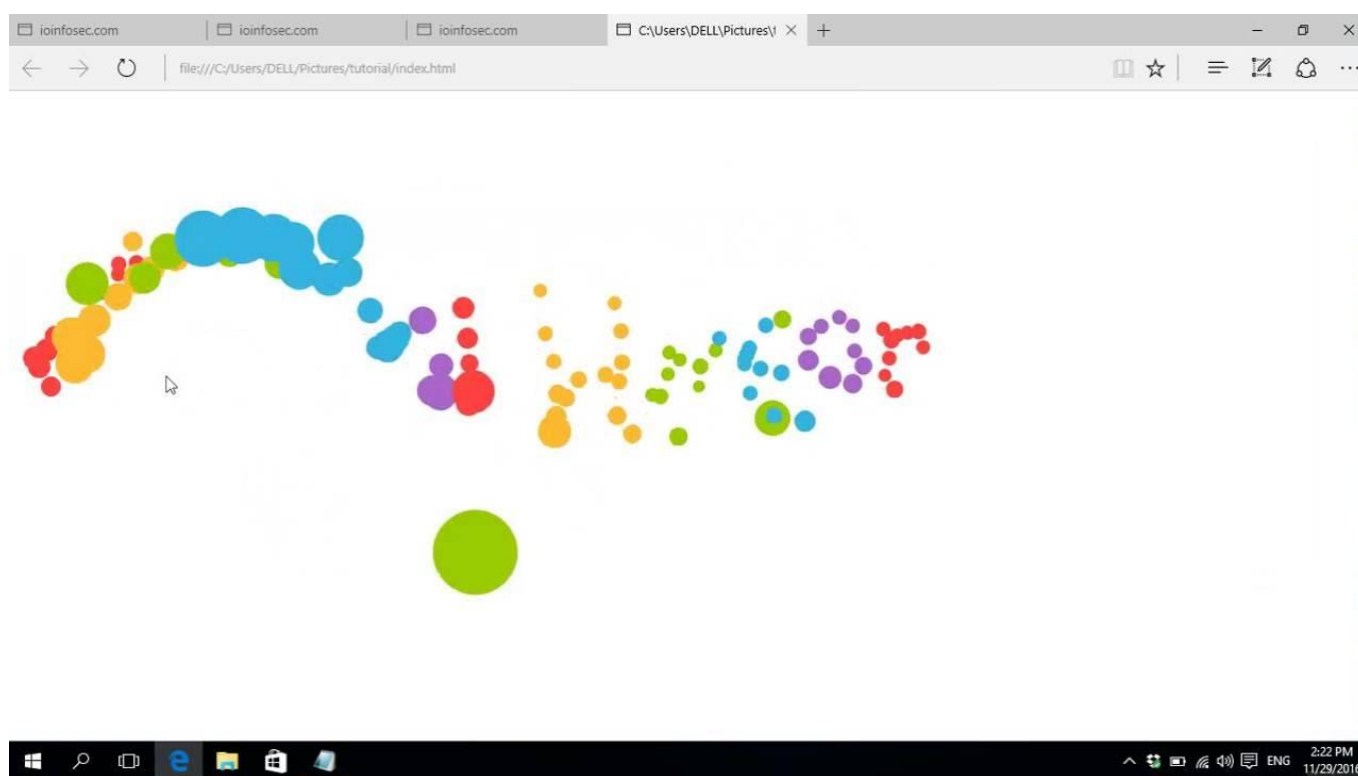
### جلسه ۴۳ : آموزش گرافیک در HTML

#### مقدمه

با عرض سلام خدمت کاربران سایت [پی وی لرن](#) . در این پست آموزشی قصد داریم به آموزش **چگونگی کار با گرافیک در html** بپردازیم. در HTML 5 شما می توانید اشکال هندسی مختلف را رسم کرده و آنها را ویرایش کنید. که اینکار در HTML با کمک جاوا اسکریپت انجام می شود. این اشکال هندسی ساده در HTML در موارد مختلفی از جمله سایت های آموزشی و تدریس، کاربرد دارد.

در مطالب آموزشی قبلی در حوزه HTML ما به آموزش چگونگی کار با تگ های [Table](#) , [Div](#) , [img](#) , [تعریف رنگ ها در HTML](#) , [تعریف فرم ها در HTML](#) , [چگونگی افزودن ویدئو در HTML](#) و ... پرداختیم.

در این پست آموزشی نیز به آموزش کار با گرافیک در HTML می پردازیم.



Graphic کار با گرافیک-

#### کار با گرافیک در HTML

در HTML برای رسم اشکال گرافیکی از تگ **<canvas>** استفاده می شود. از این تگ برای رسم اشکال هندسی با استفاده از طریق جاوا اسکریپت استفاده می شود. در واقع تگ **<canvas>** فقط یک ظرف برای ایجاد گرافیک در HTML است. و شما باید از جاوا اسکریپت برای رسم و ویرایش گرافیک استفاده کنید.

تگ **<canvas>** به تنهایی فقط یک محدوده ی مستطیلی برای رسم اشکال هندسی ایجاد می کند. و تعریف آن در HTML به این صورت است:

مثال :

```
1. <canvas id="myCanvas" width="200" height="100"></canvas>
```

صفت id به عنوان یک شناسه برای دسترسی به آن از طریق جاوا اسکریپت یا CSS و صفات width و height نیز طول و عرض مستطیل را مشخص می کند.

این یک مثال از ساختار اصلی canvas که یک مستطیل خالی را رسم می کند:

مثال : مثال canvas

```
1. <canvas id="myCanvas" width="200" height="100" style="border:1px solid #000000;">
2. </canvas>
```

[خودتان امتحان کنید <<](#)

**رسم اشکال هندسی**

مثال : کد اسکریپت زیر یک دایره را در محیط مستطیلی <canvas> ایجاد می کند:

مثال : رسم دایره

```
1. var c = document.getElementById("myCanvas");
2. var ctx = c.getContext("2d");
3. ctx.beginPath();
4. ctx.arc(95, 50, 40, 0, 2 * Math.PI);
5. ctx.stroke();
```

[خودتان امتحان کنید <<](#)

مثال ۲ - کد اسکریپت رسم خط یکی از قطرهای مستطیل:

مثال : رسم قطر

```
1. var c = document.getElementById("myCanvas");
2. var ctx = c.getContext("2d");
3. ctx.moveTo(0, 0);
4. ctx.lineTo(200, 100);
5. ctx.stroke();
```

## خودتان امتحان کنید<<

### نوشتن متن

اینکار را به راحتی می توان با کد جاوا اسکریپت انجام داد.

مثال- کد زیر متن "Hellow World" را در محیط گرافیکی می نویسد:

مثال : نوشتن متن

```
1. var c = document.getElementById("myCanvas");
2. var ctx = c.getContext("2d");
3. ctx.font = "30px Arial";
4. ctx.fillText("Hello World", 10, 50);
```

## خودتان امتحان کنید<<

متن تو خالی :

مثال : متن تو خالی

```
1. var c = document.getElementById("myCanvas");
2. var ctx = c.getContext("2d");
3. ctx.font = "30px Arial";
4. ctx.strokeText("Hello World", 10, 50);
```

## خودتان امتحان کنید<<

رسم : Gradient



مثال : رسم gradient

```
1. var c = document.getElementById("myCanvas");
2. var ctx = c.getContext("2d");
3.
4. // Create gradient
5. var grd = ctx.createLinearGradient(0, 0, 200, 0);
6. grd.addColorStop(0, "red");
7. grd.addColorStop(1, "white");
8.
9. // Fill with gradient
10. ctx.fillStyle = grd;
```

```
11. ctx.fillRect(10, 10, 150, 80);
```

[خودتان امتحان کنید](#)

رسم: Circular Gradient



مثال : رسم دایره ای gradient

```
1. var c = document.getElementById("myCanvas");
2. var ctx = c.getContext("2d");
3.
4. // Create gradient
5. var grd = ctx.createRadialGradient(75, 50, 5, 90, 60, 100);
6. grd.addColorStop(0, "red");
7. grd.addColorStop(1, "white");
8.
9. // Fill with gradient
10. ctx.fillStyle = grd;
11. ctx.fillRect(10, 10, 150, 80);
```

[خودتان امتحان کنید](#)

رسم: image

مثال : رسم image

```
1. var c = document.getElementById("myCanvas");
2. var ctx = c.getContext("2d");
3. var img = document.getElementById("scream");
4. ctx.drawImage(img, 10, 10);
```

[خودتان امتحان کنید](#)

## کلام آخر

در html5 شما می توانید مستقل از کدهای CSS به رسم اشکال هندسی ساده و خاص بپردازید، این اشکال بیشتر در موارد علمی و آموزشی ممکن است کاربرد داشته باشند. که برای اینگونه مواقع مباحث **چگونگی کار با گرافیک در html** ارائه کردیم.

## جلسه ۴۴ : کار با تگ گرافیکی SVG در html5

### مقدمه

با عرض سلام خدمت کاربران سایت [پی وی لرن](#) . در آخرین مطالبی که در ادامه بحث **html5** ارائه شد ، پس از [آشنایی با html5](#) و ساختار کلی عناصر در **html5** و همچنین آشنایی با [عناصر جدید در html5](#)، در این بخش از آموزش ها، قصد داریم در مورد **کار با تگ گرافیکی SVG در html5** و تفاوت بین **تگ SVG و تگ Canvas**، مطالبی آموزشی و کاربردی را ارائه دهیم. البته در مطالب آموزشی اولیه **html**، در پی وی لرن، در مورد گرافیک در **html**، اشاره ای به تگ کاربردی **<canvas>** داشتیم. در این بخش نیز ، در ادامه مبحث گرافیک در **html**، به بررسی تگ گرافیکی و پرکاربردی در **html** در قالب موضوع آموزشی کار با تگ گرافیکی **SVG**، می پردازیم.

### کار با تگ گرافیکی SVG در html5

تگ **SVG**، برای رسم اشکال هندسی با قابلیت تغییر اندازه، بکار می رود. این تگ در حالت کلی برای تعریف گرافیک در وب بکار می رود.

### تگ <svg>

تگ **<svg>**، به عنوان ظرفی برای گرافیک های رسم شده عمل می کند.

تگ **<svg>**، دارای چندین روش برای رسم مسیرها، مربع یا مستطیل، دایره، متن و عکس های گرافیکی است.

### رسم دایره

مثال- رسم دایره با تگ **svg**



مثال : رسم دایره

```
1. <!DOCTYPE html>
2. <html>
3. <body>
4.
5. <svg width="100" height="100">
6.   <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />
7. </svg>
8.
9. </body>
```

[خودتان امتحان کنید <<](#)

رسم مستطیل

مثال- رسم مستطیل با تگ <svg>:



مثال : رسم مستطیل

```
1. <svg width="400" height="100">
2.   <rect width="400" height="100" style="fill:rgb(0,0,255);stroke-width:10;stroke:rgb(0,0,0)" />
3. </svg>
```

[خودتان امتحان کنید](#)

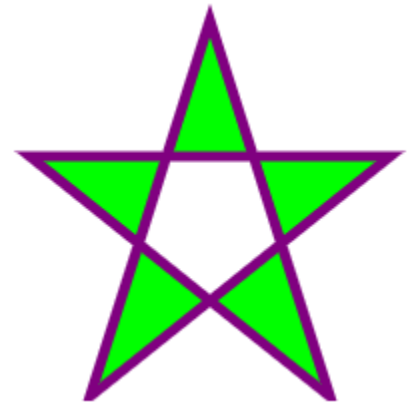
رسم مستطیل به گوشه های گرد



مثال : مستطیل با گوشه های گرد

```
1. <svg width="400" height="180">
2.   <rect x="50" y="20" rx="20" ry="20" width="150" height="150"
3.     style="fill:red;stroke:black;stroke-width:5;opacity:0.5" />
4. </svg>
```





مثال : رسم ستاره

```
1. <svg width="300" height="200">
2.   <polygon points="100,10 40,198 190,78 10,78 160,198"
3.     style="fill:lime;stroke:purple;stroke-width:5;fill-rule:evenodd;" />
4. </svg>
```



مثال : لوگوی svg

```
1. <svg height="130" width="500">
2.   <defs>
3.     <linearGradient id="grad1" x1="0%" y1="0%" x2="100%" y2="0%">
4.       <stop offset="0%" style="stop-color:rgb(255,255,0);stop-opacity:1" />
5.       <stop offset="100%" style="stop-color:rgb(255,0,0);stop-opacity:1" />
6.     </linearGradient>
7.   </defs>
8.   <ellipse cx="100" cy="70" rx="85" ry="55" fill="url(#grad1)" />
9.   <text fill="ffffff" font-size="45" font-family="Verdana" x="50" y="86">SVG</text>
10. Sorry, your browser does not support inline SVG.
11. </svg>
```

## تفاوت بین تگ SVG و تگ Canvas

SVG زبانی برای رسم اشکل دوبعدی گرافیکی در XML است.

Canvas اشکل هندسی دو بعدی را در حال حرکت رسم می کند. (با استفاده از جاوااسکریپت).

SVG مبتنی بر XML است، به این معنی که هر عنصر در SVG DOM موجود است. می توانید یک عنصر را برای رویداد جاوااسکریپت اضافه کنید.

در SVG هر شکل کشیده به عنوان یک شیء در نظر گرفته شده است. اگر ویژگی یک شیء SVG تغییر کند، مرورگر می تواند به صورت خودکار شکل را بروز کند.

## مقایسه ی SVG و Canvas

جدول زیر برخی از تفاوت های مهم بین Canvas و SVG را نشان می دهد:

SVG	Canvas
رزولوشن مستقل پشتیبانی از رویداد پردازنده برای برنامه های کاربردی با مناطق رندر بزرگ مناسب (نقشه های گوگل) مناسب است رندر کند اگر پیچیده باشد (هر چیزی که از DOM استفاده می کند بسیار کند خواهد بود) برای برنامه های بازی مناسب نیست	رزولوشن وابسته است پشتیبانی از رویداد پردازنده پشتیبانی نمی کند قابلیت رندر متن ضعیف شما می توانید تصویر حاصل را به عنوان png یا jpg ذخیره کنید مناسب برای بازی های گرافیکی شدید

## کلام آخر

گاهها در طراحی وب صفحات وب و یا درج محتوای آموزشی، علمی و ... ممکن است که نیاز به رسم اشکال گرافیکی ساده در بخش هایی داشته باشید، که در این خصوص مباحث فوق که در رابطه با کار با **تگ گرافیکی SVG در HTML5** ارائه شد تا حدی می تواند به شما کمک کند.

## فصل پنجم : چند رسانه ای در HTML

### جلسه ۴۵ : چند رسانه ای در html

#### مقدمه

با عرض سلام و وقت بخیر خدمت کاربران سایت **پی وی لرن** . و کاربرانی که **دوره کامل آموزش html** را دنبال می کنند. اغلب اوقات شما نیاز دارید در بخش یا بخش هایی از صفحه وب خود یک یا چند فایل چند رسانه ای نظیر ویدئو ، صوت و... را قرار دهید، این فایل های چند رسانه ای در موارد مختلفی کاربرد دارد به عنوان مثال زمانیکه شما قصد دارید یک فیلم یا فایل صوتی آموزشی را برای کاربران وب سایت نمایش دهید و یا فیلمی را برای دانلود در پیش نمایش صفحه خود قرار دهید، در اینگونه موارد باید با طرز استفاده از **چند رسانه ای در html** آشنایی داشته باشید. از این رو ما مباحث آموزش این بخش را به چگونگی کار با **Multimedia در html** اختصاص داده ایم.

#### چند رسانه ای در html

در ادامه ی این بخش شما را با مفاهیم Multimedia در html ، پشتیبانی در مرورگرها و فرمت های قابل استفاده ی چند رسانه ای در html آشنا می کنیم.

#### html در Multimedia

چند رسانه ای در html در قالب های مختلفی می آید. این می تواند تقریباً هر چیزی را که می توانید بشنوید یا ببینید شامل شود.

مثال: تصاویر، موسیقی، صدا، فیلم ها، سوابق، فیلم ها، انیمیشن ها و... صفحات وب اغلب حاوی عناصر چند رسانه ای از انواع و فرمت های مختلف هستند.

#### پشتیبانی در مرورگرها

اولین مرورگرهای وب فقط از متن پشتیبانی می کردند.

بعدها مرورگرها با پشتیبانی از رنگ ها و فونت ها و تصاویر وارد شدند!

سپس با پیشرفت مرورگرها و html پشتیبانی از فایل های ویدئویی و صوتی نیز اضافه شد.

#### فرمت های چند رسانه ای در html

عناصر چند رسانه ای (مانند صوتی و تصویری) در فایل های رسانه ذخیره می شوند.

رایج ترین روش برای کشف نوع فایل، این است که به دنبال فرمت فایل بروید.

فایل های چندرسانه ای دارای فرمت های مختلف مانند: .wav ، .mp3 ، .mp4 ، .mpg ، .wmv ، و .avi می باشند.

## فرمت های عمومی ویدئوی



MP4 فرمت جدید و آینده برای ویدئو اینترنتی است.

MP4 توسط YouTube توصیه می شود.

MP4 توسط Flash Player ها پشتیبانی می شود.

MP4 توسط HTML5 پشتیبانی می شود.

توضیحات	File	Format
اولین قالب ویدئوی معروف در وب که برای همه مرورگرها پشتیبانی می شود، اما در HTML5 پشتیبانی نمی شود.	mpg. mpeg.	MPEG
معمولا در دوربین های ویدئویی و سخت افزار تلویزیون استفاده می شود و به خوبی روی کامپیوترهای ویندوز پخش می شود، اما در مرورگرهای وب پشتیبانی نمی شود.	avi.	AVI
معمولا در دوربین های ویدئویی و سخت افزار تلویزیون استفاده می شود و به خوبی روی کامپیوترهای ویندوز پخش می شود، اما در مرورگرهای وب پشتیبانی نمی شود.	wmv.	WMV
معمولا در دوربین های ویدئویی و سخت افزار تلویزیون استفاده می شود و به خوبی روی کامپیوترهای اپل پخش می شود، اما در مرورگرهای وب پشتیبانی نمی شود.	mov.	QuickTime
برای ویدئوی های آنلاین و تلویزیون های متصل به اینترنت کاربرد دارد و در مرورگرها پخش نمی شود.	rm. ram.	RealVideo

این فرمت با استفاده از یک پلاگین flash در مرورگرها اجرا می شود.	swf. flv.	Flash
در HTML5 پشتیبانی می شود.	ogg.	Ogg
توسط Mozilla, Opera, Adobe و Google طراحی شده و در HTML5 پشتیبانی می شود.	webm.	WebM
پشتیبانی شده توسط همه مرورگرهای HTML5 و توصیه شده توسط یوتیوب.	mp4.	MPEG-4 or MP4

**نکته:** از فرمت های فوق، فقط ویدئویی MP4, WebM و Ogg توسط HTML5 پشتیبانی می شود.

## فرمت های صوتی

MP3 جدیدترین فرمت برای موسیقی ضبط شده فشرده است. اصطلاح MP3 مترادف با موسیقی دیجیتال شده است.

انواع فرمت های صوتی:

توضیحات	File	Format
فرمت اصلی برای تمام دستگاه های موسیقی الکترونیک مانند سینت سایزر ها و کارت های صوتی PC. فایل های MIDI شامل صدا، اکه بخوبی در دستگاههای رایانه پخش می شود، اما در مرورگرها پشتیبانی نمی شود.	mid. midi.	MIDI
RealAudio توسعه یافته توسط Real Media برای اجازه پخش صوت با پهنای باند کم، که در مرورگرهای وب پخش نمی شود.	rm. ram.	RealAudio
به خوبی در ویندوزها پخش می شود اما در مرورگرهای وب پشتیبانی نمی شود.	wma.	WMA

طراحی شده توسط اپل به عنوان قالب پیش فرض برای iTunes به خوبی در رایانه های اپل پخش می شود اما در مرورگرهای وب پشتیبانی نمی شود.	aac.	AAC
WAV توسعه یافته توسط IBM و میکروسافت، که در سیستم عامل های ویندوز، مکینتاش و لینوکس به خوبی پخش می شود. و در HTML5 پشتیبانی می شود.	wav.	WAV
اوگ توسعه یافته توسط بنیاد Xiph.Org و در HTML5 پشتیبانی می شود.	ogg.	Ogg
MP3 محبوب ترین فرمت برای پخش کننده های موسیقی است. ترکیبی از فشرده سازی خوب (فایل های کوچک) با کیفیت بالا. و در تمام مرورگرهای وب پشتیبانی می شود.	mp3.	MP3
MP4 فرمت ویدئویی آینده در اینترنت است. که همچنین منجر به پشتیبانی خودکار از MP4 صوتی در تمام مرورگرها می شود.	mp4.	MP4

**نکته :** از فرمت های صوتی فوق تنها فرمت های WAV, MP3, Ogg توسط HTML 5 پشتیبانی می شود.

## کلام آخر

با پیشرفت مرورگرها و تکنولوژی HTML پشتیبانی از فرمت های مختلف **Multimedia در html** از جمله فایل های مختلف ویدئوی و صوتی ارائه شد و قابلیت ها و کاربردهای مهمی را به وب سایت ها اضافه کرد از این رو مباحث این بخش را به آشنایی با **چند رسانه ای در html** اختصاص دادیم.

### مقدمه

با عرض سلام خدمت کاربران سایت [پی وی لرن](#) . در این پست آموزشی قصد داریم به آموزش چگونگی **اضافه کردن یک فایل ویدئو در html** بپردازیم. در [html 5](#) این قابلیت وجود دارد که شما می توانید یک فایل ویدئو، با فرمت هایی که توسط html 5 پشتیبانی می شود را در صفحه وب خود قرار دهید. html 5 از فایل های چند رسانه ای با فرمت های : **swf, .wav, .mp3, .mp4, .mpg, .wmv, and .avi**. پشتیبانی می کند. فرمت ویدئویی که عموماً استفاده می شود و توصیه شده است فرمت **MP4** است.



html-VideoHTML1فایل ویدئو در html

### play کردن ویدئو در html

مثال کامل از ویدئو در html:

مثال : مثال کامل از ویدئو در html

```
1. <!DOCTYPE html>
2. <html>
3. <body>
4.
5. <video width="400" controls>
6.   <source src="mov_bbb.mp4" type="video/mp4">
7.   <source src="mov_bbb.ogv" type="video/ogg">
8.   Your browser does not support HTML5 video.
9. </video>
10.
11. <p>
12. Video courtesy of
13. <a href="https://www.bigbuckbunny.org/" target="_blank">Big Buck Bunny</a>.
14. </p>
15.
16. </body>
17. </html>
```

[خودتان امتحان کنید <<](#)

قبل از HTML5، یک فایل ویدئو در html ، فقط می توانست در یک مرورگر با یک افزونه (مانند فلش) پخش شود. تگ <video> یک روش استاندارد برای جاسازی ویدئو در یک صفحه وب را مشخص می کند. برای نمایش ویدئو در html از تگ <video> استفاده کنید.

## مثال : مثال-2

1. <video width="320" height="240" controls>
2. <source src="movie.mp4" type="video/mp4">
3. <source src="movie.ogg" type="video/ogg">
4. Your browser does not support the video tag.
5. </video>

## خودتان امتحان کنید<>

تشریح عملکرد مثال فوق:

ویژگی **control** در تگ <video>، دکمه های کنترلی ویدئو را نظیر : **play, pause, and volume** را اضافه می کند. ایده خوبی است که همیشه ویدئو شامل ویژگی های عرض (**width**) و ارتفاع (**height**) باشد.چراکه اگر ارتفاع و عرض تنظیم نشود، ممکن است صفحه هنگام بارگیری ویدئو، سوسو زده شود. تگ <source> به شما این امکان را می دهد که نام ویدئو و فرمت آن را برای مرورگر مشخص کنید.

## Autoplay کردن فایل ویدئو در html



ipad

در HTML5 شما می توانید. فایل ویدئو را با استفاده از ویژگی **autoplay**، به حالت پخش خودکار تبدیل کنید.



1. `<video width="320" height="240" autoplay>`
2. `<"source src="movie.mp4" type="video/mp4>`
3. `<"source src="movie.ogg" type="video/ogg>`
4. Your browser does not support the video tag.
5. `</video>`

### [خودتان امتحان کنید](#)

نکته : ویژگی Autoplay در دستگاه های تلفن همراه مانند iPad و iPhone کار نمی کند.

### پشتیبانی مرورگرها از ویدئوهای html

در HTML5، ۳ فرمت ویدئو هایی که پشتیبانی می شود شامل: MP4، WebM، و Ogg می باشد.

پشتیبانی مرورگر برای فرمت های مختلف:

Ogg	WebM	MP4	مرورگر
NO	NO	YES	Internet Explorer
YES	YES	YES	Chrome
YES	YES	YES	Firefox
NO	NO	YES	Safari
YES	YES	YES (from Opera 25)	Opera

## انواع فرمت چند رسانه ای

فرمت فایل	نوع چندرسانه ای
MP4	video/mp4
WebM	video/webm
Ogg	video/ogg

### متدها، خصوصیات و رویدادها در ویدئوهای html

HTML5 متدهای DOM، خواص ها و وقایع را برای عنصر <video> تعریف می کند. این اجازه می دهد تا فیلم ها را بارگیری، پخش و مکث کنید، همچنین مدت زمان و حجم را نیز تنظیم کنید.

همچنین رویدادهای DOM وجود دارد که می توانند به شما اطلاع دهند هنگامیکه یک ویدیو شروع به پخش میکند، متوقف شده است ...و

مثال : کنترل پخش،توقف و ... در ویدئوها

```
1. <!DOCTYPE html>
2. <html>
3. <body>
4.
5. <div style="text-align:center">
6.   <button onclick="playPause()">Play/Pause</button>
7.   <button onclick="makeBig()">Big</button>
8.   <button onclick="makeSmall()">Small</button>
9.   <button onclick="makeNormal()">Normal</button>
10.  <br><br>
11. <video id="video1" width="420">
12.   <source src="mov_bbb.mp4" type="video/mp4">
13.   <source src="mov_bbb.ogv" type="video/ogg">
14.   Your browser does not support HTML5 video.
15. </video>
16. </div>
17.
18. <script>
19. var myVideo = document.getElementById("video1");
20.
21. function playPause() {
22.   if (myVideo.paused)
```

```

23.     myVideo.play();
24.     else
25.         myVideo.pause();
26. }
27.
28. function makeBig() {
29.     myVideo.width = 560;
30. }
31.
32. function makeSmall() {
33.     myVideo.width = 320;
34. }
35.
36. function makeNormal() {
37.     myVideo.width = 420;
38. }
39. </script>
40.
41. <p>Video courtesy of <a href="https://www.bigbuckbunny.org/" target="_blank">Big Buck Bunny</a>.</p>
42. </body>
43. </html>

```

## خودتان امتحان کنید

### تگ های ویدئو در HTML5

توضیحات	تگ
یک ویدئو یا فیلم را تعریف می کند.	<video>
منابع رسانه چندگانه را برای عناصر رسانه ای تعریف می کند، مانند <video> و <audio>	<source>
آهنگ های متن را در پخش کننده های رسانه تعریف می کند.	<track>

### کلام آخر

امروزه در میان محتوای چندرسانه ای در وب سایت ها، ویدئو ها و فیلم ها به عنوان یکی از پرکاربردترین محتواها در وب سایت های جهان استفاده می شوند. که کاربردهای مختلفی از جمله آموزشی، علمی، سرگرمی و... دارند. که برای این منظور باید با چگونگی اضافه کردن ویدئو در HTML مطابق مباحث فوق آشنایی داشته باشید.

### مقدمه

با عرض سلام خدمت کاربران سایت پی وی لرن . در این پست آموزشی قصد داریم به آموزش چگونگی اضافه کردن یک فایل صوتی در html بپردازیم. در 5 html این امکان نیز وجود دارد که شما می توانید یک فایل صوتی با فرمت هایی که توسط 5 html پشتیبانی می شود را در صفحه وب خود اضافه کنید. 5 html از فایل های چند رسانه ای با فرمت های : **swf, .wav, .mp3, .mp4, .mpg, .wmv, and .avi**. پشتیبانی می کند. و فایل های صوتی نیز با فرمت های **MP3, Wav, and Ogg** پشتیبانی می شود.

در پست های آموزشی قبلی html، ما به آموزش مباحث چگونگی کار با تگ های **Div, Table, img**، تعریف رنگ ها در html، تعریف فرم ها در html همچنین آموزش چگونگی افزودن ویدئو در html پرداختیم. در ادامه این پست نیز به آموزش افزودن فایل صوتی در html خواهیم پرداخت.

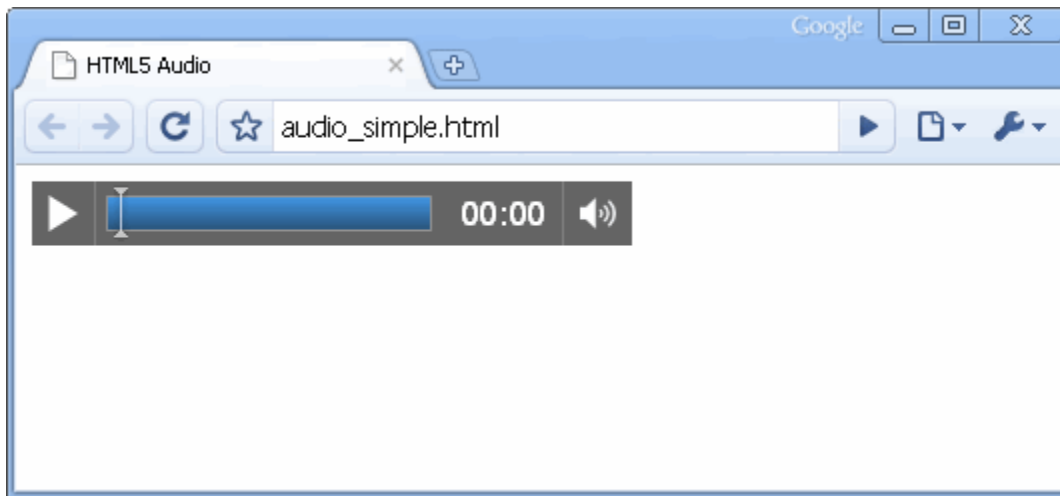


fig4-2-فایل های صوتی در html

### فایل صوتی در html

قبل از HTML5، یک فایل صوتی در html، فقط می توانست در یک مرورگر با یک افزونه (مانند فلش) پخش شود. در html برای اضافه کردن فایل های صوتی به وب، از تگ **<audio>** استفاده می شود.

### استفاده از تگ **<audio>**

در html تگ **<audio>** فایل های صوتی را در صفحه وب پخش می کند.

مثال : فایل صوتی

1. `<audio controls>`
2. `<source src="horse.ogg" type="audio/ogg">`
3. `<source src="horse.mp3" type="audio/mpeg">`
4. Your browser does not support the audio element.
5. `</audio>`

[خودتان امتحان کنید <<](#)

## ویژگی های تگ <audio>

ویژگی **control** در تگ <audio>، دکمه های کنترلی پخش فایل های صوتی نظیر : **play, pause, and volume** را اضافه می کند. تگ <source> به شما این امکان را می دهد که نام فایل صوتی و فرمت آن را برای مرورگر خود، مشخص کنید. در مرورگر هایی که از تگ <audio> پشتیبانی نمی کنند، مرورگر متن بین <audio/>..و<audio> را نمایش می دهد.

## پشتیبانی مرورگرها از فایل های صوتی در HTML

در HTML5، ۳ فرمت صوتی فرمت های صوتی که پشتیبانی می شود شامل: MP3، WAV، و OGG می باشد.

پشتیبانی مرورگر برای فرمت های مختلف:

OGG	WAV	MP3	مرورگر
NO	NO	YES	Internet Explorer
YES	YES	YES	Chrome
YES	YES	YES	Firefox
NO	YES	YES	Safari
YES	YES	YES	Opera

## انواع فرمت های چندرسانه ای صوتی در HTML

فرمت فایل	نوع رسانه
MP3	audio/mpeg
OGG	audio/ogg
WAV	audio/wav

### متد، خصوصیات و رویدادهای صوت در HTML

متدهای DOM در HTML5، خواص و حوادث را برای عنصر `<audio>` تعریف می کند. این به شما اجازه می دهد بارهای، پخش و مکث صوت، و همچنین مدت زمان و حجم را تنظیم کنید.

همچنین رویدادهای DOM وجود دارد که می تواند هنگامی که یک صدا شروع به پخش می کند، متوقف شود و غیره به شما اطلاع دهد.

### تگ های صوتی در HTML5

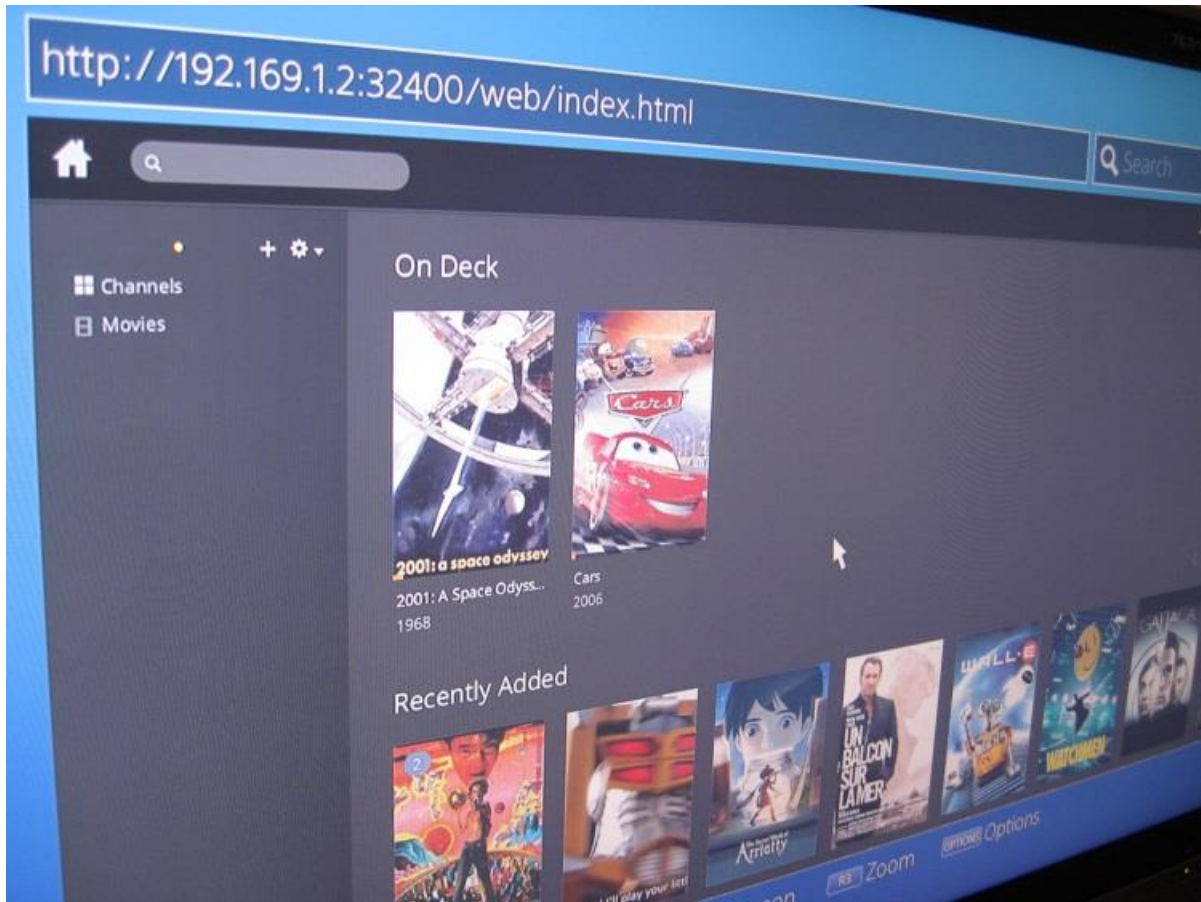
توضیحات	تگ
تعریف محتوای صوتی	<code>&lt;audio&gt;</code>
منابع رسانه چندگانه را برای عناصر رسانه ای تعریف می کند، مانند <code>&lt;video&gt;</code> و <code>&lt;audio&gt;</code>	<code>&lt;source&gt;</code>

### کلام آخر

در جلسه قبل به بررسی سینتکس و نحوه ی افزودن فایل های ویدئویی در HTML پرداختیم، در کنار ویدئو در برخی موارد ممکن است به درج فایل های صوتی نیز نیاز پیدا کنید، که برای اینگونه موارد مباحث **فایل صوتی در HTML** را ارائه کردیم.

### مقدمه

با عرض سلام خدمت کاربران سایت [پی وی لرن](#) . در مباحث آموزشی قبلی که در سایت پی وی لرن، در دوره آموزشهای کامل html ارائه شد. مطالب آموزشی با هدف [آشنایی بیشتر با html5](#)، و [عناصر جدید html5](#) و... ارائه شد. همچنین در ادامه وارد مباحث گرافیک در html، شدیم و به آموزش موضوعات کاربردی [SVG در html5](#) و چگونگی نمایش [نقشه گوگل در html](#) پرداختیم. در این بخش از آموزش ها در زمینه چند رسانه ای ها در html، قصد داریم به [آموزش کار با Plug-ins در html](#) و [کاربرد Plug-ins در html](#)، بپردازیم.



آموزش کار با Plug-ins در html

### آموزش کار با Plug-ins در html

هدف از یک Plug-ins (افزونه)، این است که قابلیت استفاده از یک مرورگر وب را گسترش دهد.

### انواع Plug-ins در html

Plug-ins، در واقع برنامه های کامپیوتری است، که برای گسترش دادن قابلیت های استاندارد یک مرورگر وب بکار می رود.

Plug-ins، می تواند با استفاده از تگ های `<object>` و `<embed>`، به صفحه وب اضافه شود.

Plug-ins ها برای اهداف زیادی از جمله: نمایش نقشه ها، scan و ویروسها، چک کردن بانک id و ... استفاده می شود.

## عنصر <object>

تگ <object> ، در تمام مرورگر ها پشتیبانی می شود. تگ <object> در واقع اشیای جاسازی شده را در HTML تعریف می کند.

مثال : عنصر آبجکت

1. `<object width="400" height="50" data="bookmark.swf"></object>`

### خودتان امتحان کنید<>

همچنین تگ <object> ، می تواند از سند HTML دیگری در داخل HTML دیگری تعریف شود.

مثال : شامل یک صفحه وب

1. `<object width="100%" height="500px" data="snippet.html"></object>`

### خودتان امتحان کنید<>

و یا می تواند شامل یک عکس باشد:

مثال : استفاده از عکس در آبجکت

1. `<object data="audi.jpeg"></object>`

### خودتان امتحان کنید<>

## عنصر <embed>

عنصر <embed> ، در تمام مرورگر های بزرگ ، پشتیبانی می شود. مرورگرهای وب مدت زمان زیادی از عنصر <embed> پشتیبانی می کنند. با این حال تا قبل از HTML5 ، این عنصر به عنوان بخشی از مشخصات HTML ، محسوب نمی شود.

مثال : فایل های فلش

1. `<embed width="400" height="50" src="bookmark.swf">`

### خودتان امتحان کنید<>

همچنین تگ <embed> ، می تواند از سند HTML دیگری در داخل HTML دیگری تعریف شود.

مثال : شامل یک صفحه وب

1. `<embed width="100%" height="500px" src="snippet.html">`

### خودتان امتحان کنید<>



و همچنین می تواند شامل یک عکس باشد:

مثال : عنصر embed شامل یک عکس

1. `<embed src="audi.jpeg">`

[خودتان امتحان کنید<>](#)

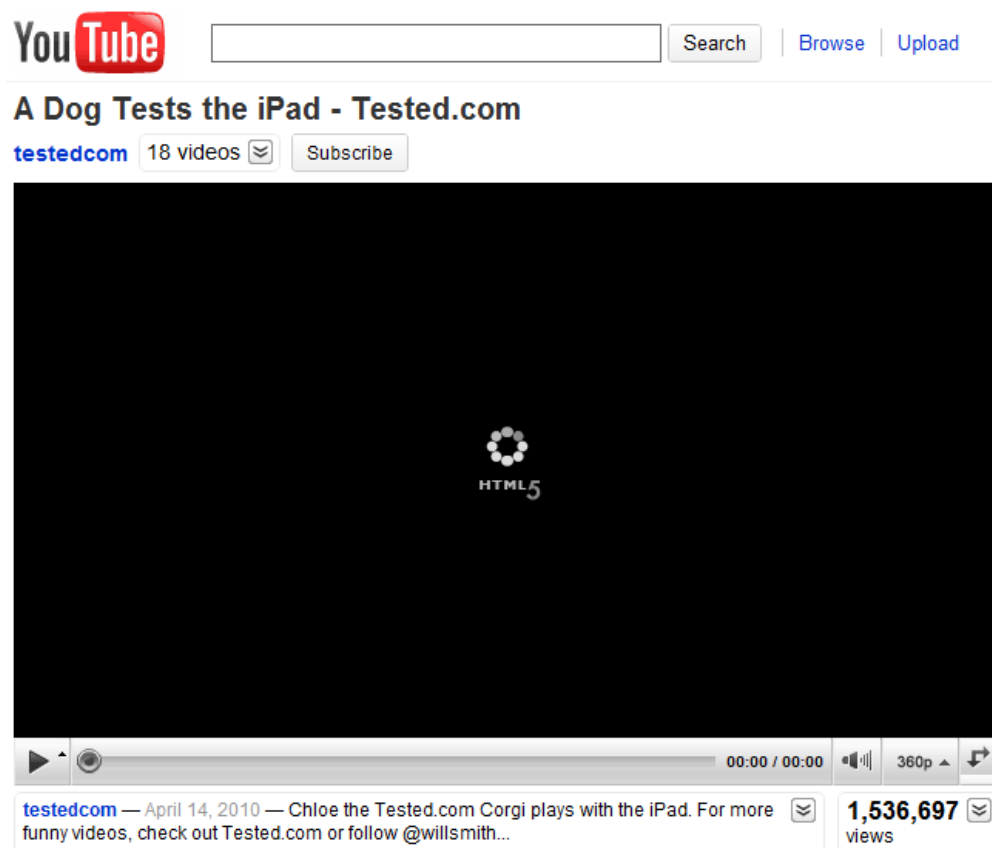
## کلام آخر

در مباحث قبلی فصل چگونگی درج و کار با فایل های ویدئویی و صوتی در html5 را بررسی کردیم، ممکن است گاهی به درج فایل های دیگری نظیر فرمت های فلش نیاز داشته باشید، که برای اینگونه موارد مباحث این بخش را به **آموزش کار با Plug-ins در html** اختصاص دادیم.

## جلسه ۴۹ : ویدئوهای YouTube در html

### مقدمه

با عرض سلام خدمت کاربران سایت [پی وی لرن](#) . در آخرین مطالب آموزشی که در ادامه آموزش های html در سایت پی وی لرن ، ارائه شد. ما وارد مباحث گرافیک در html شدیم. و به ارائه موضوعات پرکاربردی نظیر [SVG در html5](#) ، و چگونگی نمایش و استفاده از [نقشه گوگل در html](#) ، پرداختیم. سپس وارد مبحث [چند رسانه ای در html](#) ، شدیم . و به بررسی موضوع [کار با Plug-ins در html](#) ، پرداختیم. اکنون در این بخش از ادامه آموزشها، در ادامه مبحث چند رسانه ای ، قصد داریم به موضوع جالب و پرکاربرد آموزش [استفاده از ویدئوهای YouTube در html](#) یا [پخش ویدئوهای یوتیوب در html](#) ، بپردازیم.



آموزش استفاده از ویدئوهای YouTube در html

### استفاده از ویدئوهای YouTube در html

آسان ترین راه ، برای نمایش ویدئوها در وب سایت ، استفاده از YouTube است.

### فرمت های ویدئویی

شاید تا قبل از این آموزش ، شما برای اینکه ویدئوهای مختلف را در مرورگر خود نمایش دهید، مجبور به تغییر دادن فرمت ویدئو ها می شدید.

تبدیل ویدئو ها به فرمت های دیگر ، مشکل و وقت گیر است. آسانترین راه این است ، که اجازه بدهید، YouTube ویدئو ها را اجرا کند.

## id ویدئو در YouTube

YouTube با یک id (مانند tgbNymZ7vqY) نمایش می دهد. (زمانیکه ویدئو را ذخیره یا اجر امی کنید، این اتفاق می افتد). شما می توانید از این id ، در کد html خود برای نمایش ویدئو ها استفاده کنید.

## اجرای ویدئوهای YouTube در html

برای اجرای ویدئو با استفاده از YouTube در کد html خود مراحل زیر را دنبال کنید:

- ویدیو را به YouTube آپلود کنید.
- توجه داشته باشید از شناسه ویدیو، کپی بگیرید.
- یک عنصر `<iframe>` را در صفحه وب خود تعریف کنید.
- صفت `src` را با URL ویدئو مقدار دهی کنید.
- از ویژگی های عرض و ارتفاع برای مشخص کردن ابعاد پخش استفاده کنید.
- پارامتر های دیگر را نیز به URL اضافه کنید (زیر را ببینید).

### مثال : اجرای ویدئوهای youtube در html

```
1. <iframe width="420" height="315"
2. src="https://www.youtube.com/embed/tgbNymZ7vqY">
3. </iframe>
```

## خودتان امتحان کنید<>

### پخش خودکار YouTube

شما می توانید ویدیوی خود را به طور خودکار شروع کنید زمانی که یک کاربر به بازدید از این صفحه با اضافه کردن یک پارامتر ساده به آدرس YouTube شما دسترسی پیدا کند.

کافیست ، مقادیر زیر را در صفت `autoplay` تنظیم کنید:

- Value 0 (پیش فرض) : ویدئو بصورت خودکار پخش نمی شود.
- Value 1 : ویدئو زمانیکه ، player بارگزاری شود ، بصورت خودکار ، پخش می شود.

### مثال : پخش خودکار ویدئوهای یوتیوب

```
1. <iframe width="420" height="315"
2. src="https://www.youtube.com/embed/tgbNymZ7vqY?autoplay=1">
3. </iframe>
```

## لیست پخش (YouTube Playlis)

این قابلیت لیست کاملی از ویدیوهایی که اجرا می شود، را اضافه می کند (علاوه بر URL اولیه).

## پخش مجدد ویدئو پس از اتمام (Loop)

- Value 0 (پیش فرض): ویدئو به خودی خود، یکبار اجرا می شود.
- Value 1: زمانیکه ویدئو به انتهای پخش برسد، مجدداً اجرا می شود.

مثال : لیست پخش

```
1. <iframe width="420" height="315"
2. src="https://www.youtube.com/embed/tgbNymZ7vqY?playlist=tgbNymZ7vqY&loop=1">
3. </iframe>
```

[خودتان امتحان کنید <<](#)

## کنترل های YouTube

- Value 0: کنترل های player نمایش داده نمی شود.
- Value 1 (پیش فرض): کنترل های player نمایش داده می شود.

مثال : کنترل پخش ویدئوها

```
1. <iframe width="420" height="315"
2. src="https://www.youtube.com/embed/tgbNymZ7vqY?controls=0">
3. </iframe>
```

[خودتان امتحان کنید <<](#)

## استفاده از عناصر <object> یا <embed>

نکته: تگ های <object> و <embed> در youtube از ژانویه ۲۰۱۵ منسوخ شده اند. شما باید به جای استفاده از <object> و <embed> از تگ iframe استفاده کنید.

مثال : استفاده از تگ object (منسوخ شده)

```
1. <object width="420" height="315"
2. data="https://www.youtube.com/embed/tgbNymZ7vqY">
3. </object>
```

[خودتان امتحان کنید <<](#)

مثال : استفاده از تگ embed (منسوخ شده)

```
1. <embed width="420" height="315"
2. src="https://www.youtube.com/embed/tgbNymZ7vqY">
```

## کلام آخر

youtube به عنوان یکی از بزرگترین منابع آنلاین فیلم در جهان ، یکی از کامل ترین مجموعه های ویدئوهای مختلف آموزشی،علمی،سرگرمی و... می باشد، از آنجایی که **ویدئوهای YouTube در html** پخش می شود. شما می توانید به راحتی از ویدئوهای youtube را در وب سایت خود پخش کنید.

#### مقدمه

با عرض سلام خدمت کاربران سایت [پی وی لرن](#) . ما تاکنون در دوره آموزش کامل html در پی وی لرن، به بحث آشنایی کامل با html5 با ویژگی ها و عناصر آن ، پرداختیم. سپس به مبحث [گرافیک در html5](#) ، پرداختیم. و در ادامه در آخرین مطالب آموزشی در مباحث چند رسانه ای در html، به مباحث آموزشی نظیر [کار با Plug-ins در html](#) و [استفاده از ویدئوهای YouTube در html](#) ، پرداختیم. اکنون در این فصل از آموزشهای html، قصد داریم وارد مباحث کار با انواع API ، در HTML شویم. و در اولین بخش آموزشی به موضوع کاربردی [موقعیت جغرافیای در html5](#) و [Google Maps در html](#) از بخش آموزش API در html5، می پردازیم.



موقعیت جغرافیای در html5

#### موقعیت جغرافیای در html5

موقعیت جغرافیایی (HTML Geolocation) ، یک API است، که برای مکانیابی موقعیت جغرافیایی کاربران، استفاده می شود.

#### محل جغرافیایی کاربران

HTML Geolocation برای گرفتن موقعیت جغرافیایی یک کاربر استفاده می شود. از آنجا که این ویژگی به حریم خصوصی کاربران مربوط می شود ، موقعیت آنها در دسترس نیست، مگر اینکه کاربر آن را تایید کند.

#### استفاده از قابلیت HTML Geolocation

از تابع `getCurrentPosition()` در جاوااسکریپت برای برگرداندن موقعیت جغرافیایی کاربران استفاده می شود.

مثال زیر، طول و عرض جغرافیایی موقعیت کاربر را نشان می دهد:

مثال : کد موقعیت مکانی

```
1. <script>
2. var x = document.getElementById("demo");
3. function getLocation() {
```

```

4.     if (navigator.geolocation) {
5.         navigator.geolocation.getCurrentPosition(showPosition);
6.     } else {
7.         x.innerHTML = "Geolocation is not supported by this browser.";
8.     }
9. }
10. function showPosition(position) {
11.     x.innerHTML = "Latitude: " + position.coords.latitude +
12.     "<br>Longitude: " + position.coords.longitude;
13. }
14. </script>

```

## خودتان امتحان کنید<>

بررسی کد مثال بالا:

- ابتدا بررسی می شود، که آیا موقعیت جغرافیایی پشتیبانی می شود.
- اگر پشتیبانی شود، روش `getCurrentPosition()` را اجرا می کند. اگر نه، یک پیام به کاربر نمایش می دهد.
- اگر روش `getCurrentPosition()` موفق باشد، یک شی مختصات را به تابع مشخص شده در پارامتر (`showPosition`) ارسال می کند.
- تابع `showPosition()` گزینه های `Latitude` و `Longitude` را به عنوان خروجی می دهد.

## رفع خطاهای احتمالی

دومین پارامتر از روش `getCurrentPosition()` برای رسیدگی به خطاها استفاده می شود. این تابع را برای اجرا، در صورت عدم دسترسی به مکان کاربر مشخص شده است:

### مثال : خطا زدایی

```

1. function showError(error) {
2.     switch(error.code) {
3.         case error.PERMISSION_DENIED:
4.             x.innerHTML = "User denied the request for Geolocation."
5.             break;
6.         case error.POSITION_UNAVAILABLE:
7.             x.innerHTML = "Location information is unavailable."
8.             break;
9.         case error.TIMEOUT:
10.            x.innerHTML = "The request to get user location timed out."
11.            break;
12.         case error.UNKNOWN_ERROR:
13.            x.innerHTML = "An unknown error occurred."
14.            break;
15.     }
16. }

```

## خودتان امتحان کنید<>

### نمایش نتایج در نقشه

برای نمایش نتیجه در یک نقشه، شما نیاز به دسترسی به یک سرویس نقشه، مانند Google Maps را دارید.

در مثال زیر، طول و عرض جغرافیایی بازگشتی برای نشان دادن موقعیت مکانی در نقشه Google (با استفاده از یک تصویر استاتیک) بکار می رود:

```

1. function showPosition(position) {
2.     var latlon = position.coords.latitude + "," + position.coords.longitude;
3.
4.     var img_url = "https://maps.googleapis.com/maps/api/staticmap?center="
5.     "+latlon+"&zoom=14&size=400x300&sensor=false&key=YOUR_KEY";
6.
7.     document.getElementById("mapholder").innerHTML = "<img src='"+img_url+"'>";
8. }

```

## اطلاعات خاص موقعیت

این صفحه نشان داده است که چگونه موقعیت کاربر را بر روی یک نقشه می توان نشان داد.

مکان جغرافیایی نیز برای اطلاعات مکان خاص مفید است، مانند:

اطلاعات محلی به روز  
نمایش امتیازات مورد علاقه در نزدیکی کاربر  
ناوبری به نوبه خود (GPS)

## متد `getCurrentPosition()`

متد `getCurrentPosition()` یک شی را در صورت موفقیت برمی گرداند. پارامترهای ابعاد عرض، طول جغرافیایی و دقت همیشه بازگشت می یابد. و خواص دیگر نیز اگر موجود باشند برمی گرداند:

Returns	خصوصیت
طول عرضی به عنوان یک عدد دهدی (همیشه برگشت)	coords.latitude
طول جغرافیایی به عنوان یک عدد دهدی (همیشه بازگشتی)	coords.longitude
دقت موقعیت (همیشه بازگشتی)	coords.accuracy



ارتفاع در متر بالاتر از سطح دریا (در صورت وجود)	coords.altitude
صحت ارتفاع موقعیت (در صورت موجود بودن)	coords.altitudeAccuracy
عنوان درجه در جهت عقربه های ساعت از شمال (اگر در دسترس است)	coords.heading
سرعت در متر در ثانیه (در صورت موجود بودن)	coords.speed
تاریخ / زمان پاسخ (اگر در دسترس است)	timestamp

## سایر متدهای جالب

جغرافیایی شی نیز روش های جالب دیگری دارد:

`watchPosition` () - موقعیت فعلی کاربر را برمی گرداند و همچنان به موقعیت به روز شده به عنوان حرکت کاربر (مانند GPS در یک خودرو) ادامه می دهد.  
`clearWatch` () - روش `watchPosition` () را متوقف می کند.  
مثال زیر روش `watchPosition` () را نشان می دهد. برای آزمایش این (به عنوان مثال گوشی هوشمند) نیاز به دستگاه GPS دقیق دارید:

### مثال : تست متد watchPosition

```

1. <script>
2. var x = document.getElementById("demo");
3. function getLocation() {
4.     if (navigator.geolocation) {
5.         navigator.geolocation.watchPosition(showPosition);
6.     } else {
7.         x.innerHTML = "Geolocation is not supported by this browser.";
8.     }
9. }
10. function showPosition(position) {
11.     x.innerHTML = "Latitude: " + position.coords.latitude +
12.     "<br>Longitude: " + position.coords.longitude;
13. }
14. </script>

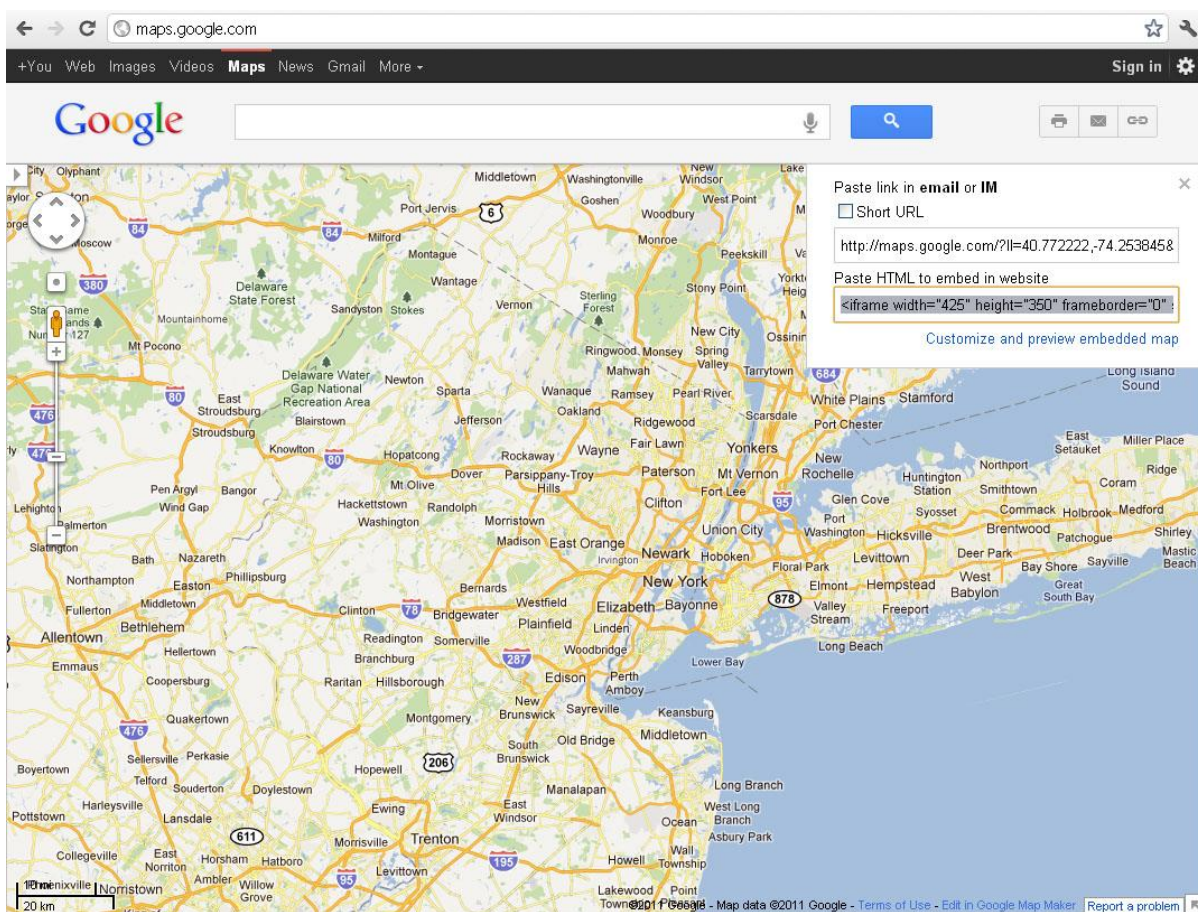
```

## کلام آخر

یکی از قابلیت های جالب در html که به کمک کدهای جاوا اسکریپت قابل پیاده سازی می باشد، **موقعیت جغرافیای در html5** است که این ویژگی به سادگی و با استفاده از کدهای html و js مباحث فوق در دسترس قرار می گیرد.

با عرض سلام خدمت کاربران سایت [پی وی لرن](#) . در آخرین مطالب ارائه شده در ادامه آموزشهای کامل html ، پس از [آشنایی با مبحث html5](#) و ساختار کد نویسی این نسخه ، و همچنین ارائه مطالبی در مورد [عناصر جدید در html5](#) ، و در مبحث گرافیک در html5 شدیم. و اشاره ای داشتیم به موضوع کار با [تگ SVG در html](#) ، اکنون در ادامه بحث گرافیک در html ، در این بخش قصد داریم به موضوع [کار با Google Maps در html](#) یا [کار با گوگل مپ در html](#) ، بپردازیم.

دسترسی به نقشه Google Maps در html ، با استفاده از تگ های html و کدهای جاوااسکریپت امکان پذیر است.



google-map

## کار با Google Maps در html

Google Maps ، به شما اجازه می دهد تا نقشه ها را در وب سایت خود نمایش دهید.

## صفحه وب پایه برای نمایش Google Maps

برای نشان دادن نحوه اضافه کردن یک نقشه گوگل به یک صفحه وب، از یک صفحه ساده HTML استفاده خواهیم کرد:

مثال :

```
1. <!DOCTYPE html>
2. <html>
3. <body>
4.
5. <h1>My First Google Map</h1>
6.
7. <div id="map">My map will go here</div>
8.
9. </body>
10.</html>
```

## تنظیم اندازه نقشه ها

مثال :

```
1. <div id="map" style="width:400px;height:400px">
```

## ایجاد تابع برای تنظیم خصوصیات نقشه

مثال :

```
1. function myMap() {
2.     var mapOptions = {
3.         center: new google.maps.LatLng(51.5, -0.12),
4.         zoom: 10,
5.         mapTypeId: google.maps.MapTypeId.HYBRID
6.     }
7.     var map = new google.maps.Map(document.getElementById("map"), mapOptions);
8. }
```

## بررسی کد مثال بالا

متغیر `mapOptions` ، خصوصیات نقشه را تعریف می کند.

ویژگی `center` ، تعیین می کند که مرکز نقشه کجا قرار بگیرد.

ویژگی `zoom` ، سطح زوم را برای نقشه تعیین می کند.

ویژگی `mapTypeId` نیز نوع نمایش نقشه را تعیین می کند.

## ایجاد نقشه های چندگانه

ایجاد ۴ نقشه با انواع مختلف:

مثال :

```
1. var map1 = new google.maps.Map(document.getElementById("googleMap1"), mapOptions1);
```

```
2. var map2 = new google.maps.Map(document.getElementById("googleMap2"), mapOptions2);
3. var map3 = new google.maps.Map(document.getElementById("googleMap3"), mapOptions3);
4. var map4 = new google.maps.Map(document.getElementById("googleMap4"), mapOptions4);
```

## تنظیم API برای نمایش نقشه گوگل

سرانجام نقشه در صفحه شما نمایش داده می شود.

قابلیت نمایش نقشه در وب ، توسط یک کتابخانه جاوااسکریپت در گوگل ارائه شده است. اکنون باید API نمایش نقشه را نیز اضافه کنید:

مثال :

```
1. <script src="https://maps.googleapis.com/maps/api/js?callback=myMap"></script>
```

با عرض سلام خدمت کاربران سایت **پی وی لرن** . در آخرین مباحث ارائه شده در **دوره آموزش کامل HTML** ، **در پی وی لرن**، موضوعات مهم و کاربردی **گرافیک در HTML5** و مبحث **چند رسانه ای در HTML5** ، پرداختیم. اکنون به تازگی وارد مباحث آموزشی **کار با API در HTML** شده ایم. و همچنین موضوع **موقعیت جغرافیای در html5**، را بررسی کردیم. در ادامه این فصل ، در این بخش از آموزش به موضوع **کار با قابلیت درگ کردن در HTML** یا **کد درگ در صفحه وب** ، می پردازیم.

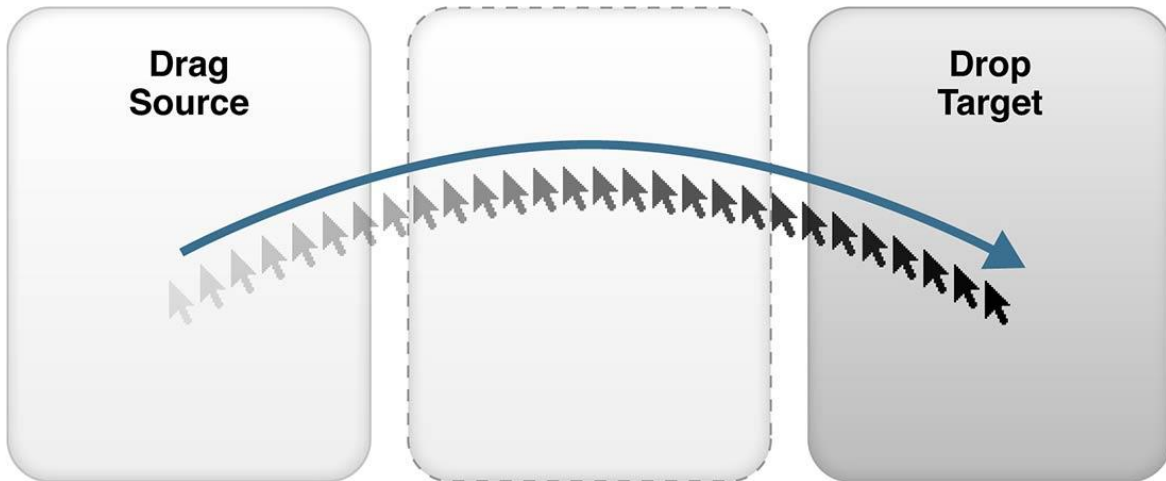


fig9-1

### کار با قابلیت درگ کردن در HTML

کشیدن و رها کردن یک ویژگی بسیار رایج است. این زمانی است که شما "شیء" را جابجا کنید و آن را به مکان دیگری بکشید.

در HTML5، کشیدن و رها بخشی از این استاندارد است: هر عنصر می تواند داری ویژگی **draggable** (قابلیت درگ) باشد.

مثال-این یک مثال ساده از قابلیت درگ است:

#### مثال : تست درگ کردن

```
1. <!DOCTYPE HTML>
2. <html>
3. <head>
4. <script>
5. function allowDrop(ev) {
6.     ev.preventDefault();
7. }
8.
9. function drag(ev) {
10.    ev.dataTransfer.setData("text", ev.target.id);
11. }
12.
13. function drop(ev) {
14.    ev.preventDefault();
15.    var data = ev.dataTransfer.getData("text");
16.    ev.target.appendChild(document.getElementById(data));
```

```

17. }
18. </script>
19. </head>
20. <body>
21.
22. <div id="div1" ondrop="drop(event)" ondragover="allowDrop(event)"></div>
23.
24. 
26.
27. </body>
28. </html>

```

## خودتان امتحان کنید

این کد ممکن است به نظر کمی پیچیده باشد، اما اجازه می دهد تا از طریق تمام قسمت های مختلف یک رویداد کشیدن و رها کردن اتفاق بیفتد.

## اعمال ویژگی draggable روی عناصر مختلف

ابتدا برای تمام عناصر، قابلیت draggable، را با ارزش TRUE برابر کنید:

مثال :

```
1. <img draggable="true">
```

## مشخص کردن عمل درگ

سپس در این قسمت مشخص می کنید که وقتی عناصر با موس کشیده شود چه اتفاقی بیفتد.

تابع `dataTransfer.setData()` نوع داده و مقدار داده هایی که درگ می شود را تعیین می کند.

مثال :

```

1. function drag(ev) {
2.     ev.dataTransfer.setData("text", ev.target.id);
3. }

```

## عنصر به کجا درگ شود

رویداد `ondragover`، جایی که عنصر قرار است به آنجا درگ شود را مشخص می کند.

در حالت عادی عنصر نمی تواند به عنصر دیگری درگ شود. برای اینکار باید ابتدا از ویژگی پیش فرض `prevent`، استفاده کنید:

مثال :

```
1. event.preventDefault()
```

## دستور کامل درگ

هنگامی عنصری کشیده و رها شود. تابع `drop`، فراخوانی می شود.

## مثال :

```
1. function drop(ev) {
2.     ev.preventDefault();
3.     var data = ev.dataTransfer.getData("text");
4.     ev.target.appendChild(document.getElementById(data));
5. }
```

تشریح کد فوق :

call preventDefault() برای جلوگیری از بارگیری پیش فرض مرورگر داده ها (به طور پیش فرض به عنوان لینک در حالت افتتاح باز است).

داده های درگ شده را با متد dataTransfer.getData() دریافت کنید. این متد هر نوع اطلاعاتی را که در متد setData() برای همان نوع تنظیم شده است را بازمی گرداند. داده درگ شده، شناسه عنصر کشیده شده است (drag1) عنصر درگ شده را به عنصر رها اضافه کنید.

## مثال : مثال دوم درگ کردن

```
1. <!DOCTYPE HTML>
2. <html>
3. <head>
4. <style>
5. #div1, #div2 {
6.     float: left;
7.     width: 100px;
8.     height: 35px;
9.     margin: 10px;
10.    padding: 10px;
11.    border: 1px solid black;
12. }
13. </style>
14. <script>
15. function allowDrop(ev) {
16.     ev.preventDefault();
17. }
18.
19. function drag(ev) {
20.     ev.dataTransfer.setData("text", ev.target.id);
21. }
22.
23. function drop(ev) {
24.     ev.preventDefault();
25.     var data = ev.dataTransfer.getData("text");
26.     ev.target.appendChild(document.getElementById(data));
27. }
28. </script>
29. </head>
30. <body>
31.
32. <h2>Drag and Drop</h2>
```



```
33. <p>Drag the image back and forth between the two div elements.</p>
34.
35. <div id="div1" ondrop="drop(event)" ondragover="allowDrop(event)">
36.   
37. </div>
38.
39. <div id="div2" ondrop="drop(event)" ondragover="allowDrop(event)"></div>
40.
41. </body>
42. </html>
```

## خودتان امتحان کنید<<

### کلام آخر

یکی دیگر از ویژگی‌ها و قابلیت‌های جالبی که در HTML5 ارائه شده و به کمک کدهای جاوا اسکریپت قابل پیاده‌سازی است، **قابلیت درگ کردن در HTML** می‌باشد که برای مواقعی که نیاز دارید آبجکتی مانند یک تصویر در HTML قابل درگ توسط کاربر باشد، کاربرد دارد.

### مقدمه

با عرض سلام خدمت کاربران سایت **پی وی لرن** . در ادامه **دوره آموزش کامل html** ، مبحث **کار با چند رسانه ای در html** را بطور کامل ارائه کردیم. و با نحوه کار کردن با انواع چند رسانه ای و کار با پلاگین در html ، آشنا شدید. در حال حاضر ادامه مبحث **API** را دنبال می کنیم. در جلسات قبلی نیز مطالبی در مورد موقعیت جغرافیایی و **درگ در html** ارائه کردیم. در ادامه فصل ، قصد داریم به موضوع **آشنایی با ذخیره سازی وب در html** یا **آموزش web storage** و **نحوه استفاده از web storage** ، بپردازیم.



ذخیره سازی وب در html

### آشنایی با ذخیره سازی وب در html

تکنیک **web storage** (ذخیره سازی وب) ، از **cookie** ، بهتر است ، مرورگرهایی که از **web storage** پشتیبانی می کنند:

Opera	Safari	Firefox	Internet Explorer / Edge	Chrome	API
۱۱/۵	۴/۰	۳/۵	۸/۰	۴/۰	Web Storage

## web storage (ذخیره سازی وب) چیست

با web storage (ذخیره سازی وب)، برنامه های کاربردی وب می توانند داده ها را به صورت محلی در مرورگر کاربر ذخیره کنند.

تا قبل از ارائه HTML5 ، داده ها و برنامه های کاربردی ، داخل cookies ها ، ذخیره می شدند.

اکنون web storage ، امن تر است. و مقدار زیادی از داده ها را بدون تاثیر بد بر عملکرد وب سایت، می توان به صورت محلی ذخیره کرد.

بر خلاف کوکی ها، حد ذخیره سازی بسیار بزرگتر است (حداقل ۵ مگابایت) و اطلاعات هرگز به سرور منتقل نمی شود.

### بخش های web storage

web storage ، دو شی را بای ذخیره سازی داده ها، فراهم می کند.

- **window.localStorage**: اطلاعات ذخیره بدون تاریخ انقضا ، ذخیره می شود.
- **window.sessionStorage**: ذخیره داده ها برای یک جلسه (داده ها از بین می روند زمانی که برگه مرورگر بسته شود).

قبل از اینکه از web storage ، استفاده کنید ، ابتدا بررسی کنید که آیا مرورگر شما از localStorage و sessionStorage ، پشتیبانی می کند:

مثال :

```
1. if (typeof(Storage) !== "undefined") {  
2.     // Code for localStorage/sessionStorage.  
3. } else {  
4.     // Sorry! No Web Storage support..  
5. }
```

### شی localStorage

شیء localStorage داده ها را بدون تاریخ انقضا ذخیره می کند. این به این معنی است که ، وقتی مرورگر بسته می شود داده ها حذف نخواهد شد و روز، هفته یا سال بعد در دسترس خواهند بود.

مثال : تعریف localStorage

```
1. // Store  
2. localStorage.setItem("lastname", "Smith");  
3. // Retrieve  
4. document.getElementById("result").innerHTML = localStorage.getItem("lastname");
```

### [خودتان امتحان کنید<<](#)

توضیح کد بالا:

- ابتدا یک localStorage با نام "lastname" و مقدار "value="Smith" ایجاد می شود.
- مقدار "lastname" را برمی گرداند و آن را در عنصری با ID با مقدار "result" قرار می دهد.

همچنین می توان کد فوق را به صورت زیر نوشت :

مثال :

```
1. // Store
2. localStorage.lastname = "Smith";
3. // Retrieve
4. document.getElementById("result").innerHTML = localStorage.lastname;
```

نحو حذف موارد نام خانوادگی localStorage به شرح زیر است:

مثال :

```
1. localStorage.removeItem("lastname");
```

**نکته:** ارزش های Name/value همیشه به عنوان رشته ذخیره می شود. به یاد داشته باشید که در صورت نیاز، آنها را به قالب دیگری تبدیل کنید!

مثال زیر تعداد دفعاتی که یک کاربر یک دکمه را کلیک کرده است را شمارش می کند. در این کد رشته ارزش به یک عدد تبدیل می شود تا بتواند شمارنده را افزایش دهد:

مثال : تعداد دفعات کلیک کاربران

```
1. if (localStorage.clickcount) {
2.     localStorage.clickcount = Number(localStorage.clickcount) + 1;
3. } else {
4.     localStorage.clickcount = 1;
5. }
6. document.getElementById("result").innerHTML = "You have clicked the button " +
7.     localStorage.clickcount + " time(s).";
```

[خودتان امتحان کنید <<](#)

**شیء sessionStorage**

تفاوت sessionStorage با localStorage ، این است که داده ها در این شیء با بسته شدن برگ مرورگر ، حذف خواهند شد، مثال زیر شمارش تعداد دفعاتی که کاربر یک دکمه را در جلسه جاری کلیک کرده است را نشان می دهد:

مثال : شیء sessionStorage

```
1. if (sessionStorage.clickcount) {
2.     sessionStorage.clickcount = Number(sessionStorage.clickcount) + 1;
3. } else {
```

```
4.     sessionStorage.clickcount = 1;
5. }
6. document.getElementById("result").innerHTML = "You have clicked the button " +
7. sessionStorage.clickcount + " time(s) in this session.";
```

[خودتان امتحان کنید<<](#)

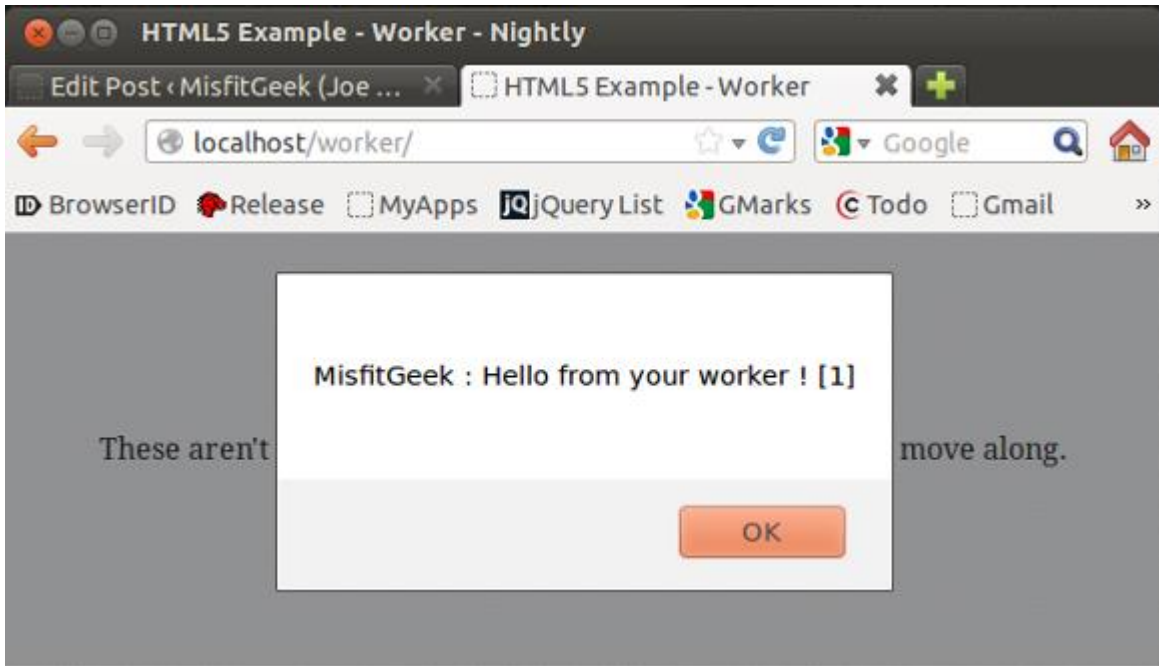
## کلام آخر

در این بخش از فصل جاری یکی دیگر از ویژگی های HTML5 که با کمک جاوا اسکریپت قابل پیاده سازی است را که ویژگی **ذخیره سازی وب در html** می باشد را نیز بررسی کردیم، در بخش بعدی به مباحث web worker ها خواهیم پرداخت.

## جلسه ۵۴ : آشنایی با Web Workers در html5

### مقدمه

با عرض سلام خدمت کاربران سایت [پی وی لرن](#) . در آخرین مباحثی که در فصل آموزش **API** ، مرور کردیم ، به مباحث آموزشی و کاربردی [موقعیت جغرافیایی در HTML](#) و [درگ کردن در HTML](#) ، نیز آشنا شدید. همچنین در مطالب اخیر ، به موضوع مهم و جذاب [ذخیره سازی وب در html](#) ، نیز پرداختیم . و کدهای جاوااسکریپت برای ذخیره سازی وب را نیز ارائه کردیم. در ادامه مبحث **API** ، در این بخش از آموزش ما قصد داریم. به موضوع [آشنایی با Web Workers در html5](#) و [استفاده از Web Worker در HTML5](#) ، بپردازیم.



worker

### آشنایی با Web Workers در html5

کارگر وب (**Web Workers**) ، در واقع یک JavaScript در حال اجرا در پس زمینه است، بدون اینکه تاثیر منفی بر عملکرد صفحه وب داشته باشد.

مرورگرهایی که از **web storage** پشتیبانی می کنند:

Opera	Safari	Firefox	Internet Explorer / Edge	Chrome	API
۱۱/۵	۴/۰	۳/۵	۱۰/۰	۴/۰	Web Storage

## آشنایی بیشتر با Web Workers

هنگام اجرای اسکریپت ها در یک صفحه HTML، صفحه تا زمانیکه اسکریپت به اتمام برسد، پاسخگو نخواهد بود. اما Web Workers ، یک جاوا اسکریپت است که در پس زمینه بصورت مستقل از سایر اسکریپت ها، اجرا می شود. بدون اینکه تاثیر منفی بر عملکرد صفحه داشته باشد.

شما نیز ، می توانید، هر کاری که می خواهید انجام دهید: کلیک کردن، انتخاب چیزها و ... در حالی که کارگر وب در پس زمینه اجرا می شود.

مثال : مثالی از webworker

```
1. <!DOCTYPE html>
2. <html>
3. <body>
4.
5. <p>Count numbers: <output id="result"></output></p>
6. <button onClick="startWorker()">Start Worker</button>
7. <button onClick="stopWorker()">Stop Worker</button>
8.
9. <p><strong>Note:</strong> Internet Explorer 9 and earlier versions do not support Web Workers.</p>
10.
11. <script>
12. var w;
13.
14. function startWorker() {
15.     if(typeof(Worker) !== "undefined") {
16.         if(typeof(w) == "undefined") {
17.             w = new Worker("demo_workers.js");
18.         }
19.         w.onmessage = function(event) {
20.             document.getElementById("result").innerHTML = event.data;
21.         };
22.     } else {
23.         document.getElementById("result").innerHTML = "Sorry, your browser does not support Web Work
24.         ers...";
25.     }
26.
27. function stopWorker() {
28.     w.terminate();
29.     w = undefined;
30. }
31. </script>
32.
33. </body>
34. </html>
```

[خودتان امتحان کنید<<](#)

### بررسی پشتیبانی Web Workers در مرورگر وب

قبل از اینکه یک web worker را ایجاد کنید ، ابتدا بررسی کنید که آیا مرورگر شما از web worker پشتیبانی می کند:

مثال :

```
1. if (typeof(Worker) !== "undefined") {
2.     // Yes! Web worker support!
3.     // Some code.....
4. } else {
5.     // Sorry! No Web Worker support..
6. }
```

## ایجاد فایل web worker

اکنون کد جاوااسکریپت web worker خود را ایجاد کنید. در این مثال یک web worker ایجاد کردیم، که از زمان اجرا شروع به شمارش می کند:

مثال :

```
1. var i = 0;
2.
3. function timedCount() {
4.     i = i + 1;
5.     postMessage(i);
6.     setTimeout("timedCount()",500);
7. }
8.
9. timedCount();
```

مهمترین قسمت کد بالا ، تابع **postMessage()** است. که برای ارسال یک پیام به صفحه HTML استفاده می شود.

معمولا از web worker ، برای پنین کارهای ساده ای استفاده نمی شود. اما در کارهای فشرده تر که CPU را زیاد درگیر می کند ، کاربرد دارد.

## ایجاد شی web worker

حالا که ما فایل web worker را داریم، باید آن را از یک صفحه HTML فراخوانی کنیم.

خطوط زیر بررسی می کند که آیا web worker ، در حال حاضر وجود دارد یا نه -سپس یک شی جدید web worker ، ایجاد می کند. و کد جاوااسکریپت را در "demo\_workers.js" اجرا می کند:

مثال :

```
1. if (typeof(w) == "undefined") {
2.     w = new Worker("demo_workers.js");
3. }
```

سپس ما می توانیم پیام ها را از طریق web worker ، ارسال و دریافت کنیم.



رویداد "onmessage" را نیز به web worker ، اضافه می کنیم:

مثال :

```
1. w.onmessage = function(event){
2.     document.getElementById("result").innerHTML = event.data;
3. };
```

وقتی web worker ، یک پیام ارسال می کند، کد در رویداد listener ، اجرا می شود. داده های web worker نیز در event.data ذخیره می شود.

## ایجاد پایان web worker

زمانیکه یک web worker ، را ایجاد می کنید. و اجرا می شود. تا زمانیکه یک پایان برای web worker ، تعریف نکرده باشید. همچنان اجرا می شود.

برای پایان دادن به یک web worker ، از تابع **terminate()** استفاده کنید:

مثال :

```
1. w.terminate();
```

## استفاده ی مجدد از Web Worker در HTML 5

پس از پایان web worker ، شما مجددا می توانید از کد web worker ، استفاده کنید:

مثال :

```
1. w = undefined;
```

## یک مثال کامل از web worker

اکنون به یک مثال کاملتر از web worker ، می پردازیم:

مثال : مثالی کامل از web worker

```
1. <!DOCTYPE html>
2. <html>
3. <body>
4.
5. <p>Count numbers: <output id="result"></output></p>
6. <button onclick="startWorker()">Start Worker</button>
7. <button onclick="stopWorker()">Stop Worker</button>
8.
9. <script>
10. var w;
```

```

11.
12. function startWorker() {
13.     if(typeof(Worker) !== "undefined") {
14.         if(typeof(w) == "undefined") {
15.             w = new Worker("demo_workers.js");
16.         }
17.         w.onmessage = function(event) {
18.             document.getElementById("result").innerHTML = event.data;
19.         };
20.     } else {
21.         document.getElementById("result").innerHTML = "Sorry! No Web Worker support.";
22.     }
23. }
24.
25. function stopWorker() {
26.     w.terminate();
27.     w = undefined;
28. }
29. </script>
30.
31. </body>
32. </html>

```

## [خودتان امتحان کنید<<](#)

### DOM و web worker

از آنجا که web worker در فایل های خارجی هستند، آنها دسترسی به اشیاء جاوا اسکریپت زیر را ندارند:

- The window object
- The document object
- The parent object

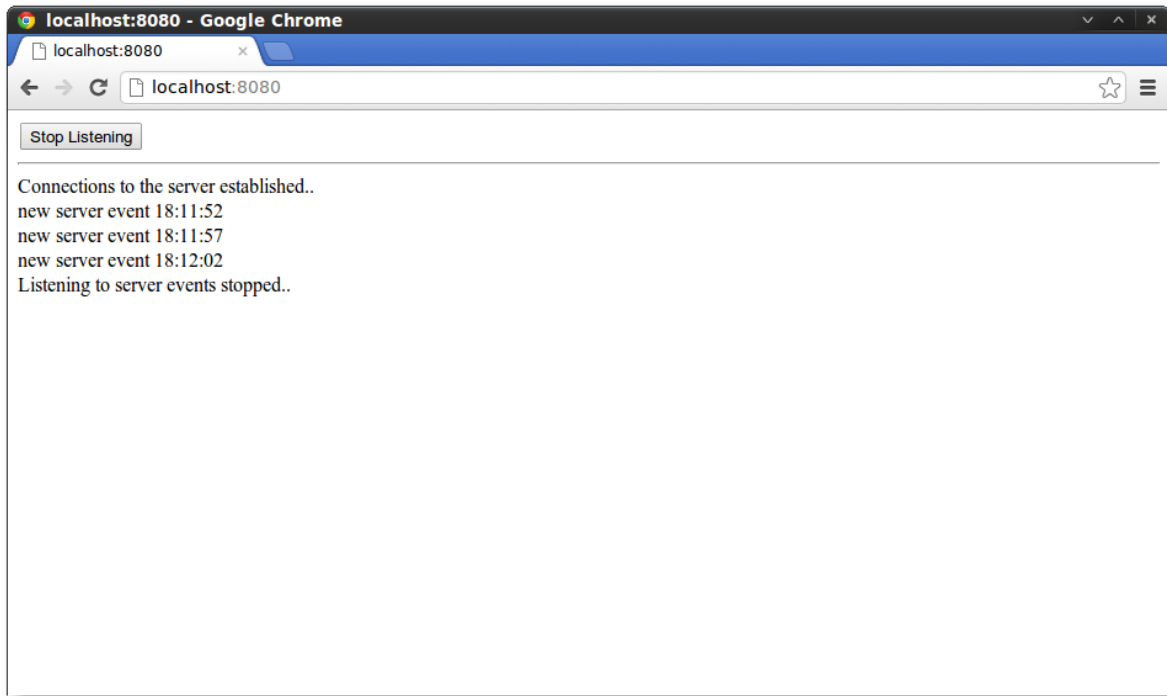
## کلام آخر

در بخش های قبلی با ویژگی ها و قابلیت های جالب و کاربردی در HTML5 با استفاده از کدهای جاوا اسکریپت آشنا شدیم. در این بخش نیز به مباحث **آشنایی با Web Workers در HTML5** پرداختیم که امکان اجرای پردازشی در پس زمینه را بدون اینکه کاربران متوجه شوند را فراهم می سازد.

## جلسه ۵۵ : چگونگی ارتباط با سرور در HTML

### مقدمه

با عرض سلام خدمت کاربران سایت [پی وی لرن](#) . در ادامه [دوره آموزش کامل HTML](#)، و در مباحث آموزش API در HTML ، با موضوعات آموزشی [موقعیت جغرافیایی در HTML](#) و [مبحث درگ کردن در HTML](#) ، آشنا شدید. همچنین در آخرین مطالب آموزشی که ارائه شد، با تکنیک [Web Workers در HTML5](#) ، و چگونگی استفاده از کدهای جاوااسکریپت در این تکنیک نیز ، آشنا شدید. در ادامه فصل آموزشی [API در HTML](#) ، در این بخش ، قصد داریم به موضوع آموزشی آشنایی با چگونگی [ارتباط با سرور در HTML](#) و [کد ارتباط با سرور در HTML](#) ، بپردازیم.



event

### ارتباط با سرور در HTML

رویداد **Server-Sent** در HTML ، اجازه می دهد که یک صفحه وب بروزرسانی جدید را از سرور دریافت کند. پشتیبانی این ویژگی در مرورگرهای مختلف به صورت زیر است :

Opera	Safari	Firefox	Internet Explorer / Edge	Chrome	ویژگی API
-------	--------	---------	--------------------------	--------	--------------

۱۱/۵	۵/۰	۶/۰	پشتیبانی نمی شود!	۶/۰	Web Storage
------	-----	-----	-------------------	-----	-------------

## پیام یکطرفه

server-sent زمانی رخ می دهد که ، یک صفحه وب بصورت خودکار بروزرسانی ها را از سرور دریافت می کند. البته باید ابتدا چک کنید. که آیا بروزرسانی جدید در دسترس است. با رویدادهای فرستاده شده توسط سرور، به روز رسانی ها به طور خودکار می آیند.

## دریافت اطلاعاتی ها از طریق server-sent

مثال: شی `EventSource` ، از server-sent برای دریافت اطلاعاتی ها استفاده می کند:

مثال :

```
1. var source = new EventSource("demo_sse.php");
2. source.onmessage = function(event) {
3.     document.getElementById("result").innerHTML += event.data + "<br>";
4. };
```

توضیح مثال بالا:

- ابتدا یک شی `EventSource` ، جدید ایجاد شده است. سپس آدرس صفحه وبی که بروزرسانی ها را ارسال می کند، مشخص شده است.
- هر وقت یک بروزرسانی دریافت شود ، رویداد `onmessage` ، فراخوانی می شود.
- وقتی یک رویداد در حال ارسال رخ می دهد، داده های دریافت شده ، به یک عنصر HTML با ID با نام "result" ، ارسال می شود.

## چک کردن پشتیبانی server-sent

قبل از استفاده از server-sent ، ابتدا باید بررسی کنید که آیا مرورگر شما ، از server-sent ، پشتیبانی می کند یا خیر:

مثال :

```
1. if(typeof(EventSource) !== "undefined") {
2.     // Yes! Server-sent events support!
3.     // Some code.....
4. } else {
5.     // Sorry! No server-sent events support..
6. }
```

## کد سمت سرور

اکنون باید در کد سمت سرور نیز ، فرآیند ارسال بروزرسانی ها را بنویسید:

مثال- در این مثال، کدهای لازم در سمت سرور، در زبان PHP ارائه شده است:

مثال :

```

1. <?php
2. header('Content-Type: text/event-stream');
3. header('Cache-Control: no-cache');
4.
5. $time = date('r');
6. echo "data: The server time is: {$time}\n\n";
7. flush();
8. ?>

```

مثال - کد بالا در زبان ASP:

مثال :

```

1. <%
2. Response.ContentType = "text/event-stream"
3. Response.Expires = -1
4. Response.Write("data: The server time is: " & now())
5. Response.Flush()
6. %>

```

تشریح کدهای فوق:

- هدر "Content-Type" را به "text / event-stream" تنظیم می کند.
- مشخص می کند که صفحه وب، نباید ذخیره شود.
- خروجی داده ها را ارسال می کند (همیشه با "داده ها" شروع می شود).
- اطلاعات خروجی را به صفحه وب ارسال می کند.

در مثال های بالا ما از رویداد onmessage برای دریافت پیام ها استفاده می کنیم. اما رویدادهای دیگر نیز موجود است:

Description	Events
هنگامی که یک اتصال به سرور باز می شود.	onopen
هنگامی که پیام دریافت می شود.	onmessage
وقتی یک خطا رخ می دهد.	onerror

## کلام آخر

در آخرین بخش از آموزش های فصل جاری و دوره HTML و HTML5 به مبحث پیاده سازی **چگونگی ارتباط با سرور در HTML** پرداختیم، که به شما اجازه می دهد اطلاعاتی را از سرور دریافت کرده و در کدهای HTML برای کاربران نمایش دهید.

امیدواریم که از آموزش های این دوره به خوبی استفاده کرده باشید، موفق و پیروز باشید.