

به نام خداوند بخشنده مهربان

آموزش طراحی صفحات وب

با جدیدترین ورژن زبان استاندارد طراحی وب

# HTML 5



الیف و گردآوری

مهرداد فتاحی

Developer Studio Network

[www.Developer1.ir](http://www.Developer1.ir)

مهر 1391

## مقدمه آموزش HTML 5 - معرفی ویژگی های جدید

### معرفی زبان : HTML 5

HTML 5 ، ورژن جدید زبان طراحی صفحات وب یعنی HTML . این زبان به عنوان جدیدترین استاندارد طراحی صفحات وب معرفی شده است .

HTML در همین سایت ، به معرفی و آموزش ورژن قدیمی تر این زبان یعنی HTML 4 پرداختیم . HTML 4 میلادی به دنیای وب معرفی شد ، ولی از آن زمان تاکنون دنیای وب و نیازهای اینترنتی تغییر زیادی کرده است . بنابراین این نیاز وجود داشت تا یک استاندارد جدید معرفی شود تا بتواند نیازهای جدید کاربران را بهتر پوشش دهد . HTML 5 هنوز در حال بروز رسانی و ویرایش است و مرورگرهای مطرح در تلاشند تا نسخه های جدید آنها ، سازگاری بهتری با این زبان جدید داشته باشد . HTML 5 ، قابلیت ها و امکانات جدیدی قرار داده شده است که تا قبل از این برای انجام آنها نیاز به نرم افزارهای جانبی در مرورگرها داشتیم . در این بخش به معرفی این ویژگی های جدید خواهیم پرداخت .



### HTML 5 ، چگونه متولد شد:

HTML 5 ، حاصل همکاری و مشارکت دو نهاد اصلی استاندارد سازی وب یعنی W3C the Web Hypertext Application Technology Working Group (WHATWG) . این دو نهاد به صورت مستقل در حال طراحی یک استاندارد برای وب بودند ، ولی در سال توافق کردند تا با همکاری هم HTML 5 را ایجاد نمایند . در طراحی HTML 5 قوانین و راهکارهای زیر به صورت اجباری تعیین شد تا رعایت شوند :

- . تمام قابلیت های جدید باید بر پایه CSS , HTML و جاوا اسکریپت باشند.
- . باید نیاز به plugin ها و نرم افزارهای جانبی مثل فلش کاهش یابد.
- . سیستم عیب یابی و رفع نقص عیب HTML بهبود یابد.
- . سعی شود در طراحی صفحات وب نیاز کمتری به اسکریپت ها باشد.

. HTML 5 باید در اجرا مستقل از سیستم و کامپیوتر باشد.

## معرفی ویژگی های جدید در HTML 5 :

HTML 5 ، ویژگی ها و قابلیت های فراوان جدیدی قرار داده شد است که کار را برای طراحان وب بسیار ساده می کند . تا پیش از این طراحان وب برای انجام این امور نیازمند استفاده از plugin ها و نرم افزارهای جانبی مختلفی بودند .

در لیست زیر برخی از این ویژگی های جدید را به صورت مختصر معرفی کرده ایم . سپس در هر بخش جدا به آموزش نحوه کارکرد و استفاده از آنها پرداخته ایم :

- اضافه شدن تگ `< canvas >` ، برای طراحی اشیا و اشکال به صورت بعدی.
- اضافه شدن تگ `< video >` `< audio >` برای اجرای فایل های صوتی و تصویری بدون نیاز به یک نرم افزار جانبی مثل Flash Player
- کنترل های جدید فرم مثل `date` , `email` , `calendar` , `time` اضافه شده اند ، که طراحی و برنامه نویسی فرم ها را بسیار ساده می کند.
- امکان ذخیره اطلاعات بر روی مرورگر کاربر - این قابلیت تقریبا کار کوکی ها `cookie` HTML انجام می دهند.

## نکاتی راجع به نحوه آموزش HTML 5 :

در این بخش آموزشی ، ابتدا به معرفی تگ ها و قابلیت های جدید پرداخته و سپس سعی شده تا با ارایه مثال های عملی و سورس کد آنها ، کاربردشان را به صورت اکتیو توسط مرورگر نشان دهیم .  
تمامی مثال ها و مطالب این بخش بر اساس استانداردهای مرجع های آموزشی رسمی این زبان طراحی شده و با مرورگر های مطرح `Chrome` , `IE` , `Opera` , `FireFox` .  
به این بخش یک قسمت جدید نیز اضافه کرده ایم که در آن پشتیبانی یا عدم پشتیبانی مرورگرهای مطرح از خواص ذکر شده را بررسی کرده ایم .

## HTML 5 < مقدمه < تگ ها و المنت های جدید HTML 5

### معرفی المنت های جدید در Html 5 :

اینترنت و نحوه استفاده از آن از سال که ورژن **Html4.01** به دنیا عرضه شد بسیار تغییر کرده است . امروزه برخی المنت ها و تگ های قدیمی **Html** دیگر استفاده نمی شود و نحوه استفاده از آن تغییر کرده است . **Html 5** برای مدیریت بهتر اینترنت امروزی شامل المنت ها و تگ های جدیدی شده است . این المنت ها و تگ های جدید قابلیت های زیادی برای ساختار بندی بهتر صفحات ، مدیریت بهتر فرم ها و اطلاعات در صفحه و اجرای انواع رسانه های صوتی و تصویری را در اختیار طراحان وب قرار می دهد . لیست زیر مهم ترین المنت های جدید **Html** را معرفی میکند . برای دریافت اطلاعات راجع به هر کدام بر روی نام آن کلیک نمایید :

### المنت های جدید ساختار بندی صفحه در HTML 5 :

<code>&lt;article&gt;</code>	این تگ یک عنوان برای صفحه ایجاد می کند .
<code>&lt;aside&gt;</code>	این تگ برای جداسازی یک متن از سایر محتویات صفحه استفاده می شود .
<code>&lt;bdi&gt;</code>	این تگ برای جداسازی یک تکه متن از سایر متون در یک پاراگراف استفاده می شود . این کار در زمانی صورت می گیرد که امکان دارد متن جدا شده دارای ساختار و یا فرمت مختلفی نسبت به سایر متن اطرافش داشته باشد .
<code>&lt;command&gt;</code>	این تگ برای تعیین یک دکمه فرمان که کاربر می تواند بر روی آن کلیک نماید ، استفاده شود.
<code>&lt;details&gt;</code>	بوسیله این تگ می توانید توضیحات اضافه ای را برای یک عنصر تعیین کرده و در صورت لزوم آن را مخفی یا نمایان کنید .
<code>&lt;summary&gt;</code>	بوسیله این تگ می توان یک عنوان را برای تگ <code>&lt;details&gt;</code> تعیین کرد .
<code>&lt;figure&gt;</code>	این تگ یک فضای جدا را برای قرار دادن عکس ، ... در صفحه ایجاد می کند .
<code>&lt;figcaption&gt;</code>	این تگ یک عنوان را برای تگ <code>&lt;figure&gt;</code> تعیین می کند .
<code>&lt;footer&gt;</code>	بوسیله این تگ می توانید یک <b>Footer</b> (پانوشته) را برای یک صفحه یا بخش تعیین نمایید
<code>&lt;header&gt;</code>	بوسیله این تگ می توانید یک <b>Header</b> ( ) را برای یک صفحه یا بخش تعیین نمایید .
<code>&lt;hgroup&gt;</code>	این تگ برای دسته بندی تگ های عنوان <code>&lt;h1&gt;</code> <code>&lt;h6&gt;</code> استفاده می شود .
<code>&lt;mark&gt;</code>	این تگ باعث مارک دار کردن ( ایجاد پس زمینه ) رنگی برای متن درون آن می شود .

<code>&lt;meter&gt;</code>	از این تگ برای ساخت یک واحد اندازه گیری و نمایش مقداری در آن ( مثل یک ستون در ) استفاده می شود .
<code>&lt;nav&gt;</code>	این تگ برای دسته بندی و کنار هم قرار دادن چندین تگ بکار می رود .
<code>&lt;progress&gt;</code>	این تگ برای ایجاد یک نمایش دهنده میزان پیشرفت یک پروسه ، استفاده می شود .
<code>&lt;section&gt;</code>	این تگ برای ایجاد یک بخش یا فضای مجزا ، بر روی صفحه استفاده می شود .
<code>&lt;time&gt;</code>	از این تگ برای تعیین یک تاریخ یا ساعت بر روی صفحه استفاده می شود .
<code>&lt;wbr&gt;</code>	این تگ برای تعیین یک شکست احتمالی در خطوط متن به کار می رود .

### المنت های جدید رسانه در HTML 5 :

<code>&lt;audio&gt;</code>	ز این تگ برای قرار دادن محتویات صوتی در صفحه ، استفاده می شود .
<code>&lt;video&gt;</code>	از این تگ برای قرار دادن محتویات تصویری یا ویدیو در صفحه ، استفاده می شود .
<code>&lt;source&gt;</code>	از این تگ برای تعیین یک یا چندین منبع فایل ، برای تگ های <code>&lt;audio&gt;</code> یا <code>&lt;video&gt;</code> می شود .
<code>&lt;embed&gt;</code>	این تگ فضایی را برای قرار گرفتن یک نرم افزار خارجی یا یک plug-in ، بر روی صفحه ایجاد می کند .

### تگ جدید `<canvas>` HTML 5 :

<code>&lt;canvas&gt;</code>	از این تگ برای ایجاد طرح ها و اشکال هندسی یا گرافیکی ، با کمک جاوا اسکریپت استفاده می .

### تگ جدید `<canvas>` HTML 5 :

HTML 5 ، تگ های جدیدی را برای کار بهتر با فرم ها ارائه کرده است ، که عبارتند از :

<code>&lt;datalist&gt;</code>	از این تگ جهت تعیین لیستی از آیتم ها ، برای کنترل های <code>&lt;input&gt;</code> استفاده می شود .

<code>&lt;keygen&gt;</code>	از این تگ برای ایجاد یک کادر رمز و رمز گذاری محتویات آن استفاده می شود . .
<code>&lt;output&gt;</code>	این تگ می تواند نتیجه یک جمع را به وسیله جاوا اسکریپت ، در خروجی نمایش دهد .

### لیست تگ های حذف شده در HTML 5 :

تگ های زیر به علت قدیمی شدن و عدم کارایی ، از فهرست تگ های HTML 5 :

- `<acronym>` •
- `<applet>` •
- `<basefont>` •
- `<big>` •
- `<center>` •
- `<dir>` •
- `<font>` •
- `<frame>` •
- `<frameset>` •
- `<noframes>` •
- `<strike>` •
- `<tt>` •

## < article > < HTML 5 تگ های جدید HTML 5

: < article >

< article > برای تعیین یک هدر یا عنوان (متن مجزا از سایر محتویات صفحه) استفاده می شود .  
< article > ، می تواند موارد زیر باشد :

- عنوان پست ها در فروم ها.
- عنوان مطالب جدید در وبلاگ ها.
- عناوین خبری در سایت های خبری.

شکل کلی تعریف یک تگ < article > بصورت زیر است :

syntax	<article>    محتویان مورد نظر درون تگ    </article>
--------	---

پشتیبانی در مرورگر های مختلف :



این تگ در تمام مرورگرهای اصلی پشتیبانی می شود .

مثال عملی :

در مثال زیر ، نحوه استفاده از تگ < article > در عمل نمایش داده ایم :

	<pre>&lt;article&gt;   &lt;h1&gt;Internet Explorer 9&lt;/h1&gt;   &lt;p&gt;Windows Internet Explorer 9 (abbreviated as IE9) was released to   the public on March 14, 2011 at 21:00 PDT.....   &lt;/p&gt; &lt;/article&gt;</pre>
خروجی	<p><b>Internet Explorer 9</b></p> <p>Windows Internet Explorer 9 (abbreviated as IE9) was released to the public on March 14, 2011 at 21:00 PDT.....</p>

## < bdi > < HTML 5 تگ های جدید HTML 5

: < bdi >

**Bi-directional Isolation** **bdi** ، به معنای جدا سازی بی جهت یا ساختار نیافته .  
برای جدا سازی و فرمت دادن به یک متن که با سایر متون اطراف خود فرق دارد ، استفاده می شود .  
از این تگ معمولا برای جهت دادن به متن یا کاراکترهایی که خارج از زبان اصلی صفحه بوده و از لحاظ فرمت با آن فرق دارد ، استفاده می شود .

برای مثال فرض کنید که یک پاراگراف دارید به زبان انگلیسی . حال عبارت یا واژه ای فارسی در میان این متن . تفاوت زبان دو نوشته ، ممکن است باعث نمایش ناصحیح متن پاراگراف شود . با قرار دادن متن فارسی در یک تگ < bdi > ، آن را از سایر متن ایزوله کرده و مانع از بروز اشکال احتمالی در نمایش آن می شوید .

شکل کلی استفاده از یک تگ < bdi > بصورت زیر است :

syntax	<bdi>	</bdi>
--------	-------	--------

پشتیبانی در مرورگر های مختلف :



این تگ فقط در مرورگرهای فایرفاکس و کروم پشتیبانی می شود .

مثال عملی :

زیر ، ند < bdi > را در عمل نمایش داده ایم . یک نام فارسی در متن انگلیسی وجود دارد ، که در حالت عادی مرورگر آن را با اشکال نمایش می دهد . اما در خط دوم ، متن فارسی را در یک تگ < bdi > قرار داده و باعث نمایش صحیح آن شده ایم :

	<pre>&lt;ul&gt;   &lt;li&gt;User Ali : 60 points&lt;/li&gt;   &lt;li&gt;User   : ۹۰ points&lt;/li&gt;   &lt;li&gt;User &lt;bdi&gt;ایان&lt;/bdi&gt; : 90 points&lt;/li&gt; &lt;/ul&gt;</pre>
خروجی	<ul style="list-style-type: none"><li>• User Ali</li><li>• User   : ۹۰ points</li><li>• User:ایان: 90 points</li></ul>



## < details > < HTML 5 تگ های جدید HTML 5 >

### < details > :

< details > برای تعیین توضیحات اضافه درباره یک موضوع استفاده می شود ، که کاربر می تواند بر طبق نیاز و انتخاب خود ، آن را مخفی یا نمایان سازد .  
به عبارت دیگر تگ < details > ، برای ایجاد یک منو یک منوی متحرک استفاده شده ، که کاربر می تواند آن را با کلیک باز و بسته نماید . هر نوع متن یا تگ دیگر را می توان درون تگ < details > ، تعیین نمود .  
**نکته مهم :** اگر خاصیت open را برای تگ < details > تعیین نکنید ، محتویات درونی آن به صورت پیش فرض مخفی خواهد بود .  
**نکته مهم :** < summary > برای تعیین محتویات درونی تگ < details > ، استفاده می شود . هنگامی که کاربر بر روی عنوان تگ < details > کلیک نماید ، این محتویات مخفی و نمایان می شوند .  
شکل کلی تعریف یک تگ < details > بصورت زیر است :

syntax	<details>	محتویان مورد نظر	</details>
--------	-----------	------------------	------------

### پشتیبانی در مرورگر های مختلف :



این تگ فقط در مرورگر کروم و Safari پشتیبانی می شود .

### مثال عملی :

در مثال های زیر ، نحوه استفاده از تگ < details > حالت متفاوت آن ، در عمل نمایش داده ایم :

**توجه ! :** این مثال فقط در مرورگر کروم کار می کند .

عدم تعیین خاصیت open	<pre>&lt;details&gt;   &lt;summary&gt;Copyright 2008-2013.&lt;/summary&gt;   &lt;p&gt; - by Developer1.ir : All Rights Reserved.&lt;/p&gt; &lt;/details&gt;</pre>
تعیین خاصیت open	

	<pre>&lt;details open&gt;   &lt;summary&gt;Copyright 2008-2013.&lt;/summary&gt;   &lt;p&gt; - by Developer1.ir : All Rights Reserved.&lt;/p&gt; &lt;/details&gt;</pre>
خروجی	<p>Copyright 2008-2013.</p> <p>Copyright 2008-2013.</p> <p>- by Developer1.ir : All Rights Reserved.</p>

## < figure > < HTML 5 تگ های جدید HTML 5

### : < figure >

< figure > ، برای تعیین یک محتوی مجزا مثل عکس ، دیاگرام ها ، لیست ها و ... بروی صفحه استفاده می شود .

کاربردی که این تگ می تواند داشته باشد ، این است که نحوه و محل قرار گیر عنصر محتویات درون آن را به جزا از سایر محتویات صفحه تعیین می کند .

بنابراین اگر عنصر حذف شود ، در جریان و چیدمان کلی صفحه ، تغییر یا مشکل خاصی به وجود نخواهد آمد .

شکل کلی تعریف یک تگ < figure > بصورت زیر است :

syntax	<figure> محتویان مورد نظر درون تگ </figure>
--------	---

### پشتیبانی در مرورگر های مختلف :



این تگ در تمام مرورگرهای مطرح پشتیبانی می شود .

### مثال عملی :

در مثال های زیر ، نحوه استفاده از تگ < figure > عمل نمایش داده ایم :

**توجه ! :** < figcaption > ، برای تعیین یک عنوان برای تگ < figure > استفاده می شود . به مثال دقت نمایید :

<pre>&lt;p&gt;   ASP.Net is a very poewrfull language . &lt;/p&gt;  &lt;figure&gt;   &lt;img src=" ../Pic/ADS/ASP.gif" alt="ASP.Net" width="200" height="120"&gt;   &lt;figcaption&gt;     Figure 1 . ASP.Net   &lt;/figcaption&gt; &lt;/figure&gt;</pre>
---

ASP.Net is a very poewrfull language .

خروجی



Figure 1 . ASP.Net

## < figcaption > < HTML 5 تگ های جدید HTML 5 >

### > < figcaption > :

> < figcaption > ، برای تعیین یک عنوان یا توضیح برای تگ < figure > استفاده می شود .  
این عنوان هم می تواند در ابتدا و هم در انتهای عنصر مادر قرار داده شود .

شکل کلی تعریف یک تگ < figcaption > بصورت زیر است :

syntax	<pre>&lt;figure&gt; ..... figure تگ ..... &lt;figcaption&gt; عنوان یا توضیح مورد نظر &lt;/figcaption&gt; &lt;/figure&gt;</pre>
--------	--

### پشتیبانی در مرورگر های مختلف :



این تگ در تمام مرورگر های مطرح پشتیبانی می شود .

### مثال عملی :

در مثال های زیر ، نحوه استفاده از تگ < figure > و تعیین یک عذ  
را در عمل نمایش داده ایم :

	<pre>&lt;p&gt;   ASP.Net is a very poewrfull language . &lt;/p&gt;  &lt;figure&gt;   &lt;img src=" ../Pic/ADS/ASP.gif" alt="ASP.Net" width="200"   height="120"&gt;   &lt;figcaption&gt;     Figure 1 . ASP.Net   &lt;/figcaption&gt; &lt;/figure&gt;</pre>
--	---

ASP.Net is a very powerful language .

خروجی



Figure 1 . ASP.Net

## < mark > < HTML 5 تگ های جدید HTML 5

: < mark >

< mark > می توان برای برجسته و یا علامت زدن رنگی نوشته استفاده کرد .  
متنی که بین تگ باز و بسته < mark > قرار بگیرد ، معمولا رنگ زردی به عنوان پس زمینه آن نشان داده می

شکل کلی استفاده از یک تگ < mark > بصورت زیر است :

syntax	<mark>	</mark>
--------	--------	---------

پشتیبانی در مرورگر های مختلف:



این تگ در تمام مرورگرهای مطرح پشتیبانی می شود .

مثال عملی :

زیر ، نحوه استفاده از تگ < mark > را در عمل نمایش داده ایم :

	<p> You are learning <mark> HTML 5 </mark> in Developer 1 . </p>
خروجی	You are learning HTML 5 in Developer 1 .

## < meter > < HTML 5 تگ های جدید HTML 5

: < meter >

از این تگ برای نمایش یک مقدار اندازه گیری عددی یا درصدی بصورت گرافیکی استفاده می شود . به عبارت دیگر از این تگ برای نمایش یک درجه یا نمودار بر روی صفحه استفاده می شود .

برای مثال بوسیله این تگ می توان درصد کل استفاده از فضای دیسک را نمایش داد . نحوه کار این تگ به این صورت است که شما یک مقدار حداکثر و حداقل را برای آن تعیین می کنید . این دو مقدار بصورت یک نم نمایش داده می شود . سپس مقدار اصلی تگ را تعیین کرده که باید بین مقدار حداکثر و حداقل باشد .

شکل کلی استفاده از یک تگ < meter > بصورت زیر است :

syntax	<code>&lt;meter value = " value " min = " value " max = " value " &gt; &lt;/meter&gt;</code>
--------	--

توضیح خواص تگ < meter > :

توضیح	
توضیح	خاصیت
این خاصیت مقداری که تگ باید بر روی نمودار یا شکل نمایش دهد ، را تعیین می کند . این مقدار مقدار واقعی تگ بوده و حتما باید تعیین شود .	value
این خاصیت حداقل مقدار تگ را تعیین می کند .	min
این خاصیت حداکثر مقدار تگ را تعیین می کند .	max
این خاصیت کمترین مقداری که انتظارش می رود تا تگ داشته باشد را تعیین می کند . این مقدار باید از خاصیت min بیشتر بوده و از خاصیت های high max کمتر باشد . تعیین این خاصیت اختیاری است .	low
این خاصیت بیشترین مقداری که انتظار می رود تا تگ داشته باشد را تعیین می کند . این مقدار باید از خاصیت low min بیشتر و از خاصیت max کمتر باشد . تعیین این خاصیت اختیاری است .	high
این خاصیت id فرمی را تعیین می کند که تگ <meter> به آن تعلق دارد . تعیین این خاصیت اختیاری است .	form



پشتیبانی در مرورگر های مختل :



این تگ در تمام مرورگرهای مطرح به جز IE پشتیبانی می شود .

مثال عملی :

زیر ، نحوه استفاده از تگ <meter> حالت عملی نمایش داده ایم :

	<pre>&lt;meter value="2" min="0" max="10"&gt;2 out of 10&lt;/meter&gt;  &lt;meter value="0.6"&gt;60%&lt;/meter&gt;</pre>
--	--

## < HTML 5 > تگ های جدید HTML 5 < nav >

: < nav >

این تگ برای دسته بندی و کنار هم قرار دادن گروهی از لینک ها استفاده می شود . لینک هایی که درون این تگ تعیین شوند به ترتیب در کنار هم قرار می گیرند . تمام لینک های یک صفحه را نباید <nav> تعریف نمود . فقط لینک های اصلی و پایه صفحه در این بخش تعریف می شوند .

شکل کلی استفاده از یک تگ < nav > بصورت زیر است :

syntax	<pre>&lt;nav&gt;   &lt;a href="Link 1"&gt; &lt;/a&gt;   &lt;a href="Link 2"&gt; &lt;/a&gt;   &lt;a href="Link 3"&gt; &lt;/a&gt;   .... &lt;/nav&gt;</pre>
--------	---

پشتیبانی در مرورگر های مختلف :

این تگ در تمام مرورگر های مطرح پشتیبانی می شود .

مثال عملی :

زیر ، نحوه استفاده از تگ < nav > و خروجی آن را در عمل نمایش داده ایم :

	<pre>&lt;nav&gt;   &lt;a href=" ../HTML/Index.aspx"&gt;HTML&lt;/a&gt;     &lt;a href=" ../CSS/Index.aspx"&gt;CSS&lt;/a&gt;     &lt;a href=" ../JavaScript/Index.aspx"&gt;JavaScript&lt;/a&gt;     &lt;a href=" ../jQuery/Index.aspx"&gt;jQuery&lt;/a&gt; &lt;/nav&gt;</pre>
خروجی	<a href="#">HTML</a>   <a href="#">CSS</a>   <a href="#">JavaScript</a>   <a href="#">jQuery</a>

## < section > < HTML 5 تگ های جدید HTML 5

: < section >

< section > برای تعیین یک بخش یا قسمت مجزا در یک سند وب استفاده می شود . برای مثال می توان به یک بخش متن ، یک عنوان خبری و .... اشاره کرد .

شکل کلی استفاده از یک تگ < section > بصورت زیر است :

syntax	<pre>&lt;section&gt; ..... محتویات مورد نظر درون تگ ..... &lt;/section&gt;</pre>
--------	--

پشتیبانی در مرورگر های مختلف:



این تگ در تمام مرورگر های مطرح پشتیبانی می شود .

مثال عملی:

زیر ، نحوه استفاده از تگ < section > را در عمل نمایش داده ایم :

	<pre>&lt;section&gt;   &lt;h1&gt;HTML 4&lt;/h1&gt;   &lt;p&gt;The old version of web desinging language .&lt;/p&gt; &lt;/section&gt;  &lt;section&gt;   &lt;h1&gt;HTML 5&lt;/h1&gt;   &lt;p&gt;The New &amp; more intersting version of web desinging language .&lt;/p&gt; &lt;/section&gt;</pre>
--	---

خروجی	<p><b>HTML 4</b></p> <p>The old version of web desinging language .</p> <p><b>HTML 5</b></p> <p>The New &amp; more intersting version of web desinging language .</p>
-------	---

## < HTML 5 تگ های جدید HTML 5 > < summary >

### < summary > :

از این تگ برای اضافه کردن زیر مجموعه و توضیحات به یک تگ <details> استفاده می شود . کاربر می تواند با کلیک بر روی تگ <details> محتویات درونی آن را مخفی یا نمایان سازد . در حالت پیش فرض محتویات درونی <details> مخفی هستند .

شکل کلی استفاده از یک تگ < summary > بصورت زیر است :

syntax	<pre>&lt;details&gt; ..... &lt;summary &gt; محتویات &lt;/summary&gt; .... &lt;/details&gt;</pre>
--------	--

### پشتیبانی در مرورگر های مختلف:



این تگ فقط در مرورگر کروم پشتیبانی می شود .

### مثال عملی:

در مثال های زیر ، نحوه استفاده از تگ < summary > متفاوت آن ، در عمل نمایش داده ایم :

**توجه ! :** این مثال فقط در مرورگر کروم کار می کند .

**نکته :** اگر خاصیت `open` را بر روی مقدار `open` قرار دهیم ، محتویات درونی آن نمایش داده می شود . غیر اینصورت طبق حالت پیش فرض ، محتویات در ابتدا مخفی هستند .

	<pre>عدم تعیین خاصیت open &lt;details&gt;   &lt;summary&gt;Copyright 2008-2013.&lt;/summary&gt;   &lt;p&gt; - by Developer1.ir : All Rights Reserved.&lt;/p&gt; &lt;/details&gt;</pre>
--	--

	<p>open تعیین خاصیت</p> <pre>&lt;details open&gt;   &lt;summary&gt;Copyright 2008-2013.&lt;/summary&gt;   &lt;p&gt; - by Developer1.ir : All Rights Reserved.&lt;/p&gt; &lt;/details&gt;</pre>
خروجی	<p>Copyright 2008-2013.</p> <p>Copyright 2008-2013.</p> <p>- by Developer1.ir : All Rights Reserved.</p>

## < HTML 5 > تگ های جدید HTML 5 < progress >

### < progress > :

progress برای نمایش فرایند پیشرفت یک پروسه بر روی صفحه استفاده می شود . مثلا برای نمایش عملیات لود شدن یک عکس یا صفحه که از صفر درصد شروع شده و به صد درصد پایان می یابد .  
**نکته :** progress خود حالت ایستا و ثابتی داشته و باید بوسیله یک اسکریپت جلوه پویا و در حرکت به آن

**نکته :** progress برای نمایش یک نمایشگر یا درجه اندازه ، مثل مقدار استفاده شده از یک دیسک استفاده نکنید . < meter > برای این کار مناسب است .

شکل کلی استفاده از یک تگ < progress > بصورت زیر است :

syntax	<progress value = " value " max = " value " > </progress>
--------	---

### توضیح خواص تگ < progress > :

توضیح syntax	
توضیح	خاصیت
این خاصیت حداکثر مقدار عددی برای نمودار یا پروسه را تعیین می کند . مثلا کاری باید در % یا	max
این خاصیت عددی تعیین کننده میزان درصد یا واحدی است ، که از کل میزان پروسه انجام شده است . برای مثال % یا واحد از کار انجام شده است .	value

### پشتیبانی در مرورگر های مختلف:



این تگ در تمام مرورگر های مطرح پشتیبانی می شود .

**مثال عملی:** زیر ، نحوه استفاده از تگ < progress > را در عمل نمایش داده ایم . این نمودار نمایشگر یک پروسه % است که % :

<progress value="30" max="100" > </progress>
--

## HTML 5 < onafterprint > رویداد های جدید HTML 5 < رویداد onafterprint

### رویداد onafterprint :

رویداد onafterprint ، پس از اینکه کاربر صفحه را برای پرینت آماده کرده و پنجره پرینت مرورگر باز می شود ، اجرا می شود .

می توان دستور یا دستوراتی را تعیین نموده و در قالب یک اسکریپت قرار داد ، تا در هنگام اجرای این رویداد، انجام

**نکته :** این رویداد ، فقط برای تگ < body > قابل تعیین است .

شکل کلی استفاده از رویداد onafterprint بصورت زیر است :

syntax	<code>&lt;element onafterprint = " script " &gt;</code>
--------	---

\* script = نام تابعی است که در هنگام رخ دادن ، رویداد اجرا می شود .

### پشتیبانی در مرورگر های مختلف:



این رویداد مرورگرهای IE و فایرفاکس پشتیبانی می شود .

### مثال عملی:

زیر ، نحوه استفاده از رویداد onafterprint عمل نمایش داده ایم :

	<code>&lt;body onafterprint = " shoMsg ( ) " &gt;</code>
--	--



## HTML 5 < رویداد های جدید HTML 5 < onbeforeprint

### رویداد **onbeforeprint** :

رویداد **onbeforeprint** ، پس از اینکه کاربر صفحه را برای پرینت آماده کرده و قبل از باز شدن پنجره پرینت مرورگر ، اجرا می شود .  
می توان دستور یا دستوراتی را تعیین نموده و در قالب یک اسکریپت قرار داد ، تا در هنگام اجرای این رویداد، انجام

**نکته :** این رویداد ، فقط برای تگ `< body >` قابل تعیین است .

شکل کلی استفاده از رویداد **onafterprint** بصورت زیر است :

syntax	<code>&lt;element onbeforeprint = " script " &gt;</code>
--------	--

**\* script =** نام تابعی است که در هنگام رخ دادن ، رویداد اجرا می شود .

### پشتیبانی در مرورگر های مختلف:



این رویداد مرورگرهای IE و فایرفاکس پشتیبانی می شود .

### مثال عملی :

زیر ، نحوه استفاده از رویداد **onbeforeprint** عمل نمایش داده ایم :

	<code>&lt;body onbeforeprint = " shoMsg ( ) " &gt;</code>
--	---

## HTML 5 < رویداد های جدید HTML 5 < onresize

### رویداد onresize :

رویداد onresize ، زمانی که کاربر اندازه پنجره مرورگر را بزرگ یا کوچک نماید ، اجرا می شود .  
از این رویداد می توان برای تعیین دستور یا دستوراتی استفاده کرد ، که میخاهیم در صورت تغییر اندازه پنجره مرورگر توسط کاربر ، اجرا شوند .  
**نکته :** این رویداد ، فقط برای تگ < body > قابل تعیین است .

شکل کلی استفاده از رویداد resize بصورت زیر است :

syntax	<code>&lt;element onresize = " script " &gt;</code>
--------	---

\* script = نام تابعی است که در هنگام رخ دادن ، رویداد اجرا می شود .

### پشتیبانی در مرورگر های مختلف:



این رویداد در تمام مرورگر های مطرح پشتیبانی می شود .

### مثال عملی:

زیر ، نحوه استفاده از رویداد onresize عمل نمایش داده ایم :

<pre>&lt;!DOCTYPE html &gt; &lt;html &gt; &lt;head&gt;   &lt;script&gt;     function showMsg ( ) {       alert("شما اندازه پنجره را تغییر دادید");     }   &lt;/script&gt; &lt;/head&gt; &lt;body onresize="showMsg( )" &gt;   &lt;p&gt; دازه پنجره را برای اجرای رویداد تغییر دهید &lt;/p&gt; &lt;/body&gt; &lt;/html&gt;</pre>
--

## HTML 5 < کنترل های جدید HTML 5 < لیست کنترل های جدید HTML 5

### لیست کنترل های جدید HTML 5 :

HTML 5 شامل چندین کنترل <input> جدید می شود که قابلیت ها و امکاناتی را مثل اعتبار سنجی اطلاعات انواع فرم های انتخابی را به کاربران می دهد . لیست زیر شامل کنترل های <input> جدید در HTML5 . برای دریافت اطلاعات بر روی نام هر کدام کلیک نمایید :

نام کنترل	
<a href="#">color</a>	از این کنترل برای نمایش یک کنترل انتخاب رنگ به کاربر استفاده میشود .
<a href="#">date</a>	از این کنترل برای نمایش یک کنترل تاریخ و انتخاب تاریخ مورد نظر استفاده می شود .
<a href="#">datetime</a>	از این کنترل برای نمایش یک کادر تاریخ و زمان به کاربر استفاده می .
<a href="#">email</a>	از این کنترل برای نمایش یک کادر متن که کاربر بایستی حتما در آن یک آدرس ایمیل وارد نماید ، استفاده می شود .
<a href="#">month</a>	از این کنترل برای نمایش یک کادر متن جهت انتخاب ماه و سال استفاده می شود .
<a href="#">number</a>	از این کنترل برای نمایش یک کادر متن به کاربر جهت وارد نمودن مقادیر عددی استفاده می شود .
<a href="#">range</a>	از این کنترل برای قرار دادن یک کادر متن بر روی صفحه استفاده می شود ، که کاربر بایستی یک مقدار عددی را در محدوده تعیین شده برای آن ، وارد نماید .
<a href="#">search</a>	از این کنترل برای قرار دادن یک کادر انتخابی جهت جستجو در سایت استفاده می شود . این کنترل از موتور جستجو گوگل استفاده می کند .
<a href="#">time</a>	از این کنترل برای قرار دادن یک کادر انتخابی جهت انتخاب تاریخ به کاربر استفاده می شود .
<a href="#">url</a>	از این کنترل برای قرار دادن یک کادر متن بر روی صفحه استفاده می شود ، که کاربر بایستی در آن یک آدرس اینترنتی وارد نماید .
<a href="#">week</a>	از این کنترل برای انتخاب هفته یا سال توسط یک کادر متن استفاده می شود .

## <input> HTML 5 < کنترل های جدید HTML 5 <

### : <input>

<input> برای قرار دادن انواع کنترل های دریافت اطلاعات از کاربر در فرم های Html استفاده می شود . بوسیله خاصیت type <input> می توانید نوع کنترل آن را مشخص نمایید ، مثلا اینکه کادر متن باشد یا منوی کرکره ای .  
چند نکته راجع به تگ <input> :

- این تگ حتما بایستی درون یک تگ <form> قرار بگیرد.
- <input> در همان تگ ابتدایی بسته شده و دارای محتویات نیست . فقط می توان برای آن خواص تعیین کرد.

شکل کلی استفاده از تگ <input> بصورت زیر است :

syntax	<input type="type" name="name" >
--------	----------------------------------

### توضیح موارد syntax :

توضیح syntax	
توضیح	Parameter
نوع کنترل را مشخص می کند . که می تواند یکی از انواع زیر باشد :	type
button	
checkbox	
color	
date	
datetime	
email	
file	
hidden	
image	
month	
number	
password	
radio	

range reset search submit text time url week	
یک نام را برای کنترل تعیین می نماید .	name

### خواص جدید تگ <input> HTML 5 :

نام خاصیت	نوع خاصیت	
accept	audio/* video/* image/*	این خاصیت نوع فایلی که سرور برای کنترل قبول می کند را تعیین می کند .
autocomplete	on off	این خاصیت تعیین می کند که آیا قابلیت پر کردن خودکار کنترل روشن باشد یا خیر .
autofocus	autofocus	تعیین می کند که آیا کنترل بصورت اتوماتیک در هنگام لود شدن صفحه فوکوس را در اختیار بگیرد یا خیر .
form	form_id	این خاصیت تعیین کننده یک یا در id فرم است که کنترل مورد نظر به آن .
formaction	URL	این خاصیت آدرس URL ای را تعیین می کند که قرار است اطلاعات کنترل را پردازش کند .
formenctype	application/x-www-form-urlencoded multipart/form-data text/plain	مشخص می کند هنگامی که اطلاعات کنترل به سرور ارسال می شود چگونه و با چه فرمتی پردازش شود .
formmethod	get post	این خاصیت متد HTTP مورد نظر برای ارسال اطلاعات فرم را مشخص می کند .
formnovalidate	formnovalidate	تعیین می کند که اطلاعات کنترل های فرم در هنگام ارسال اعتبارسنجی نشوند .

این خاصیت نحوه باز شدن صفحه ای که فرم به آن ارسال می شود را مشخص می کند .	blank_ self_ parent_ top_ frameName	formtarget
این خاصیت تعیین کننده id <datalist> ای است که آیتم های از پیش تعیین شده ای را برای کنترل مشخص می کند .	datalist_id	list
این خاصیت حداکثر مقدار عددی قابل قبول برای یک کنترل <input> را تعیین می کند .	number date	max
این خاصیت حداقل مقدار عددی قابل قبول برای یک کنترل <input> را تعیین می کند .	number date	min
این خاصیت تعیین می کند که آیا کاربر می تواند در یک کنترل بیش از یک مقدار را وارد کند یا خیر .	multiple	multiple
این خاصیت یک قالب یا شیوه را برای سنجش و اعتبارسنجی داده وارد شده در کنترل تعیین می کند .	regexp	pattern
بوسیله این خاصیت می توان یک مقدار از پیش تعیین شده را در یک کادر متن برای راهنمایی کاربر جهت وارد نمودن صحیح اطلاعات درخواستی قرار داد .	text	placeholder
این خاصیت تعیین می کند که حتما کنترل باید پر شود ( خالی ) .	required	required
این خاصیت میزان گام افزایش یا کاهش مقدار عددی را برای یک کنترل <input> تعیین می کند .	number	step

## HTML 5 < کنترل های جدید در HTML 5 < کنترل color

### کنترل: color

از این کنترل برای قرار دادن یک دکمه فرمان بر روی صفحه استفاده می شود که کاربر با کلیک بر روی آن می تواند یک رنگ را برای ارسال به صفحه انتخاب نماید . با کلیک بر روی دکمه فرمان منوی انتخاب رنگ ویندوز باز شده و شما می توانید رنگ مورد نظر خود را انتخاب نمایید .  
شکل کلی استفاده از این کنترل به صورت زیر است :

syntax	<code>&lt;input type="color" name="name" &gt;</code>
--------	--

### پشتیبانی در مرورگر های مختلف:



این کنترل فقط در مرورگر های کروم و اپرا پشتیبانی می شود .

### مثال عملی:

: در مثال زیر یک کنترل `<input type="color">` را بر روی صفحه قرار داده ایم . به وسیله این کنترل یک رنگ دلخواه را انتخاب کرده و سپس با کلیک بر روی دکمه فرمان `set color` پس زمینه `div` مثال را تغییر دهید :

Example	
<pre>&lt;input type="color" name="favcolor" id="inputColor" &gt; : نمایید &lt;br /&gt; &lt;input type="button" id="btn1" value="set color" onclick="changeColor()" /&gt; &lt;br /&gt; &lt;div id="MyDiv"&gt; Developer Studio Network&lt;/div&gt;  &lt;script type="text/javascript"&gt; function changeColor() {     document.getElementById("MyDiv").style.backgroundColor = document.getElementById("inputColor").value; } &lt;/script&gt;</pre>	کد

## HTML 5 < کنترل های جدید در HTML 5 < کنترل date

### کنترل date :

از این کنترل برای قرار دادن یک کنترل تاریخ بر روی صفحه استفاده می شود . کاربر می تواند تاریخ مورد نظر خود را در آن انتخاب کرده و مقدارش را به صفحه ارسال کند .  
شکل کلی استفاده از این کنترل به صورت زیر است :

syntax	<code>&lt;input type="date" name="name" &gt;</code>
--------	---

### پشتیبانی در مرورگر های مختلف:



این کنترل فقط در مرورگر های کروم و اپرا پشتیبانی می شود .

### مثال عملی:

: در مثال زیر یک کنترل `<input>` `type="date"` را بر روی صفحه قرار داده ایم . به وسیله این کنترل یک تاریخ دلخواه را انتخاب کرده و سپس با کلیک بر روی دکمه فرمان `set date` ، تاریخ انتخاب شده را بر روی صفحه نمایش دهید :

Example	
<pre>&lt;input type="date" id="inputDate" &gt; تاریخ مورد نظر را انتخاب نمایید : &lt;br /&gt; &lt;input type="button" id="btn1" value="set date" onclick="changeDate()" /&gt; &lt;br /&gt; &lt;div id="MyDiv"&gt; &lt;/div&gt;  &lt;script type="text/javascript"&gt; function changeDate() {     document.getElementById("MyDiv").innerHTML = "Time is : " + document.getElementById("inputDate").value; } &lt;/script&gt;</pre>	کد



## HTML 5 < کنترل های جدید در HTML 5 < کنترل email

### کنترل email :

از این کنترل برای قرار دادن یک کادر متن بر روی صفحه استفاده می شود که کاربر حتما باید در آن یک آدرس ایمیل وارد نماید . اگر مقدار ورودی توسط کاربر فرمت صحیحی یک ایمیل را نداشته باشد ، کنترل error مانع از ارسال اطلاعات می شود .  
شکل کلی استفاده از این کنترل به صورت زیر است :

syntax	<code>&lt;input type="email" name="name" &gt;</code>
--------	--

### پشتیبانی در مرورگر های مختلف:



این کنترل فقط در مرورگرهای کروم ، فایرفاکس و اپرا پشتیبانی می شود .

### مثال عملی:

: در مثال زیر یک کنترل `<input type="email">` را بر روی صفحه قرار داده ایم . به وسیله این کنترل آدرس ایمیل خود را وارد کرده و سپس با کلیک بر روی دکمه فرمان `show email` ، آن را بر روی صفحه نمایش دهید :

Example	
<pre>&lt;input type="email" id="inputEmail" &gt; آدرس ایمیل خود را وارد نمایید : &lt;br /&gt; &lt;input type="button" id="btn1" value="show email" onclick="showEmail()" /&gt; &lt;br /&gt; &lt;div id="MyDiv"&gt; &lt;/div&gt;  &lt;script type="text/javascript"&gt; function showEmail() {     document.getElementById("MyDiv").innerHTML = "Your Email is : " + document.getElementById("inputEmail").value; } &lt;/script&gt;</pre>	کد

## HTML 5 < کنترل های جدید در HTML 5 < کنترل month

### کنترل: month

از این کنترل برای قرار دادن یک کادر متن بر روی صفحه استفاده می شود که کاربر بایستی نام یک هفته یا تاریخ یک سال را در آن وارد نماید. سپس این کنترل مقدار را به صفحه ارسال می کند .  
شکل کلی استفاده از این کنترل به صورت زیر است :

syntax	<code>&lt;input type="month" name="name" &gt;</code>
--------	--

### پشتیبانی در مرورگر های مختلف:



این کنترل فقط در مرورگر های کروم و اپرا پشتیبانی می شود .

### مثال عملی:

: در مثال زیر یک کنترل `<input>` `type="month"` را بر روی صفحه قرار داده ایم . به وسیله این کنترل ماه مورد نظر خود را وارد کرده و سپس با کلیک بر روی دکمه فرمان `show month` ، آن را بر روی صفحه نمایش دهید :

Example	
<pre>&lt;input type="month" id="inputMonth" &gt; &lt;br /&gt; &lt;input type="button" id="btn1" value="show month" onclick="showMonth()" /&gt; &lt;br /&gt; &lt;br /&gt; &lt;div id="MyDiv"&gt; &lt;/div&gt;  &lt;script type="text/javascript"&gt; function showMonth() { document.getElementById("MyDiv").innerHTML = "Month is : " + document.getElementById("inputMonth").value; } &lt;/script&gt;</pre>	کد

## HTML 5 < کنترل های جدید در HTML 5 < کنترل search

### کنترل search :

از این کنترل برای قرار دادن یک کادر متن جهت جستجو در وب سایت خودتان استفاده می شود . این کنترل معمولا از موتور جستجو گوگل برای Search استفاده می کند .  
شکل کلی استفاده از این کنترل به صورت زیر است :

syntax	<code>&lt;input type="search" name="name" &gt;</code>
--------	---

### پشتیبانی در مرورگر های مختلف:



این کنترل فقط در مرورگر کروم پشتیبانی می شود .

### مثال عملی:

: در مثال زیر یک کنترل `<input type="search">` را بر روی صفحه قرار داده ایم . به وسیله این کنترل می توانید عملیات جستجو را انجام دهید :

Example	
<code>&lt;input type="search" id="inputSearch" &gt; دنبال چه می گردید ؟ &lt;br /&gt; &lt;input type="submit" id="btn1" value="Search" /&gt;</code>	کد
<p>دنبال چه می گردید ؟</p> <input type="text" value="Search"/>	خروجی

## HTML 5 < کنترل های جدید در HTML 5 < کنترل range

### کنترل number :

از این کنترل برای قرار دادن یک کادر متن بر روی صفحه استفاده می شود که کاربر فقط مجاز است در آن مقادیر عددی را در محدوده تعیین شده ، انتخاب نماید .

کردن مقادیر غیر عددی و همچنین مقادیر بیشتر و یا کمتر از محدوده تعیین شده ، برنامه این مقادیر را نادیده گرفته و ارسال نمی کند .

شکل کلی استفاده از این کنترل به صورت زیر است :

syntax	<code>&lt;input type="range" name="name" min="value" max="value" step="value" value="value" &gt;</code>
--------	---

توضیح موارد syntax	
توضیح	
این پارامتر تعیین کننده حداکثر مقدار عددی است که کاربر می تواند در کنترل وارد نماید .	max
این پارامتر تعیین کننده حداقل مقدار عددی است که کاربر می تواند در کنترل وارد نماید .	min
گام افزایش یا کاهش مقدار عددی کنترل را تعیین می کند .	step
این خاصیت مقدار پیش فرض کنترل را تعیین می کند .	value

### پشتیبانی در مرورگر های مختلف:



این کنترل فقط در مرورگر های کروم و اپرا پشتیبانی می شود .

### مثال عملی:

: در مثال زیر یک کنترل `<input type="range">` را بر روی صفحه قرار داده ایم . به وسیله این کنترل باید عددی بین      را انتخاب نمایید . مقادیر بزرگتر و یا کوچکتر را کنترل error داده و قبول نمی کند .. سپس با کلیک بر روی دکمه فرمان show number ، می توانید عدد انتخاب شده را بر روی صفحه نمایش دهید ( ر پیش فرض کنترل ) :

### Example

```
<input type="range" id="inputRange" min="1" max="1" value="5"> :  
عدد مورد نظر خود را انتخاب نمایید  
<br />  
<input type="button" id="btn1" value="show number"  
onclick="showNumber()" />  
<br />  
<div id="MyDiv"> </div>  
  
<script type="text/javascript">  
  function showNumber() {  
    document.getElementById("MyDiv").innerHTML = "Number is : " +  
document.getElementById("inputRange").value;  
  }  
</script>
```

ک

## HTML 5 < کنترل های جدید HTML 5 < کنترل time

### کنترل time :

از این کنترل برای قرار دادن یک کنترل زمان بر روی صفحه استفاده می شود . کاربر می تواند زمان مورد نظر خود را در آن انتخاب کرده و مقدارش را به صفحه ارسال کند .  
شکل کلی استفاده از این کنترل به صورت زیر است :

syntax	<code>&lt;input type="time" name="name" &gt;</code>
--------	---

### پشتیبانی در مرورگر های مختلف:



این کنترل فقط در مرورگر های کروم و اپرا پشتیبانی می شود .

### مثال عملی:

: در مثال زیر یک کنترل `<input>` `type="time"` را بر روی صفحه قرار داده ایم . به وسیله این کنترل یک زمان دلخواه را انتخاب کرده و سپس با کلیک بر روی دکمه فرمان `set time` ، تاریخ انتخاب شده را بر روی صفحه نمایش دهید :

Example	
<pre>&lt;input type="time" id="inputTime" &gt; &lt;br /&gt; &lt;input type="button" id="btn1" value="set time" onclick="changeTime()" /&gt; &lt;br /&gt; &lt;div id="MyDiv"&gt; &lt;/div&gt;  &lt;script type="text/javascript"&gt; function changeTime() {     document.getElementById("MyDiv").innerHTML = "Time is : " + document.getElementById("inputTime").value; } &lt;/script&gt;</pre>	کد

## HTML 5 < کنترل های جدید در HTML 5 < کنترل url

### کنترل url :

از این کنترل برای قرار دادن یک کنترل کادر متن بر روی صفحه استفاده می شود، که کاربر حتما باید یک آدرس اینترنتی با فرمت صحیح را در آن وارد نماید . اگر مقدار وارد شده در کنترل دارای فرمت صحیح یک آدرس اینترنتی نباشد ، کنترل error داده و مانع ارسال اطلاعات می شود .

**نکته :** فرمت صحیح یک آدرس اینترنتی به صورت زیر است :

http://www.abc.xyz

شکل کلی استفاده از این کنترل به صورت زیر است :

syntax	<code>&lt;input type="url" name="name" &gt;</code>
--------	--

### پشتیبانی در مرورگر های مختلف:



این کنترل در مرورگرهای کروم ، فایرفاکس و اپرا پشتیبانی می شود .

**مثال عملی:** در مثال زیر یک کنترل `<input type="url">` را بر روی صفحه قرار داده ایم . بایستی در کادر متن کنترل یک آدرس اینترنتی ، با فرمت صحیح را وارد نماید . در صورت وارد کردن مقدار اشتباه کنترل error می دهد :

Example	
<pre>&lt;input type="url" id="inputUrl" &gt; آدرس مورد نظر خود را وارد نمایید : &lt;br /&gt; &lt;input type="button" id="btn1" value="show url" onclick="showUrl( )" /&gt; &lt;br /&gt; &lt;div id="MyDiv"&gt; &lt;/div&gt;  &lt;script type="text/javascript"&gt; function showUrl( ) {     document.getElementById("MyDiv").innerHTML = "Url is : " + document.getElementById("inputUrl").value; } &lt;/script&gt;</pre>	کد

## HTML 5 < خواص جدید کنترل ها در HTML 5 > خاصیت autocomplete

### خاصیت autocomplete :

خاصیت autocomplete تعیین می کند که آیا کنترل باید قابلیت پر شدن خودکار داشته باشند یا خیر . هنگامی که یک کنترل قابلیت پر شدن خودکار دارد ، مرورگر بصورت اتوماتیک مقادیر آنها را بر حسب مقادیری که قبلا کاربر وارد نموده است ، پر می کند . هنگامی که کاربر شروع به تایپ در کنترل می کند ، مرورگر مقادیر قبلی که وی وارد نموده است را در یک کادر نمایش می دهد و می تواند از آنها یکی را انتخاب کند . این خاصیت در حالت پیش فرض خاموش است و در کنترل های زیر کار می کند :

- text
- search
- url
- password
- date
- rang
- email
- color

**نکته** : می توان قابلیت پر شدن خودکار یک فرم را بر روی مقدار روشن تنظیم نمود ، ولی همین قابلیت را برای کنترل موجود در آن خاموش کرد .

**نکته** : در برخی از مرورگرها بایستی در تنظیمات قابلیت پر شدن خودکار را فعال کرد .

شکل کلی استفاده از خاصیت autocomplete ورت زیر است :

syntax	<input autocomplete = " on   off " >
--------	--------------------------------------

توضیح syntax	
توضیح	Parameter
در این حالت قابلیت پر شدن خودکار فیلد ها در فرم فعال است .	on
در این حالت قابلیت پر شدن خودکار فیلد ها در فرم غیر فعال است . مقدار پیش فرض است .	off



## پشتیبانی در مرورگر های مختلف:



این خاصیت در تمام مرورگرهای اصلی پشتیبانی می شود .

## مثال عملی:

: در مثال زیر ، دو کنترل را در فرم روی صفحه قرار داده ایم . در کنترل اول مقدار خاصیت autocomplete روی on و در کنترل دوم روی off تنظیم شده است . شروع به تایپ در کنترل ها کرده و تفاوت ها را بسنجید :

Example	
<pre>&lt;form id="form1" &gt;   First name:&lt;input type="text" name="fname" autocomplete = "on"&gt;&lt;br /&gt;   E-mail: &lt;input type="email" name="email" autocomplete = "off"&gt;&lt;br /&gt;   &lt;input type="submit" value="form1" &gt; &lt;/form&gt;</pre>	کد
<p>First name : <input type="text"/></p> <p>E-mail : <input type="text"/></p> <p>form 1</p>	خروجی

## HTML 5 < autofocus > خاصیت

### خاصیت autofocus :

خاصیت فوق برای یک کنترل تعیین می کند آیا کنترل مذکور در هنگام لود شدن صفحه فوکوس را بصورت اتوماتیک در اختیار بگیرد یا خیر .

این خاصیت یک مقدار Boolean است و در صورتیکه آن را در یک کنترل تعیین نمایید به محض لود شدن صفحه فوکوس در اختیار آن قرار می گیرد . این خاصیت در حال پیش فرض برای هیچ کنترلی در فرم فعال نیست .

شکل کلی استفاده از خاصیت autofocus بصورت زیر است :

syntax	<input autofocus >
--------	--------------------

### پشتیبانی در مرورگر های مختلف:



این خاصیت در تمام مرورگرهای پشتیبانی می شود ( IE نسخه به بعد ) .

### مثال عملی:

: در مثال زیر ، یک فرم را برای روی صفحه قرار داده ایم ، که دارای دو کنترل است . برای کنترل دوم مقدار خاصیت autofocus را تنظیم کرده و در کنترل اول این خاصیت مقدار دهی نشده است . همانطور که می بینید ، کنترل دوم در هنگام لود صفحه ، فوکوس را در اختیار می گیرد :

Example	
<pre>&lt;form id="form1"&gt;   First name:&lt;input type="text" name="fname" &gt;&lt;br /&gt;   E-mail: &lt;input type="email" name="email" autofocus&gt;&lt;br /&gt;   &lt;input type="submit" value="form1" &gt; &lt;/form&gt;</pre>	کد
<p>First name : <input type="text"/></p> <p>E-mail : <input type="text"/></p> <p><input type="submit" value="form1"/></p>	خروجی

## HTML 5 < form خاصیت > HTML 5

### خاصیت form :

این خاصیت در یک کنترل <input> ، شناسه ( id ) یک یا چند فرم را تعیین می کند ، که کنترل مذکور به آن فرم

این خاصیت در زمانی استفاده می شود که کنترل <input> در مکانی خارج از تگ <form> متعلق به آن قرار داشته باشد . شما می توانید id بیش از یک فرم را در این خاصیت تعیین نمایید . برای این منظور بایستی نام فرم ها را به ترتیب و با فاصله قرار دهید .

شکل کلی استفاده از خاصیت form بصورت زیر است :

syntax	<input form = " form_id ">
--------	----------------------------

توضیح syntax	
توضیح	Parameter
تعیین کننده id یک یا چند فرم است که کنترل به آن تعلق دارد .	form_id

### پشتیبانی در مرورگر های مختلف:



این خاصیت در تمام مرورگرهای اصلی به جز اینترنت اکسپلورر ، پشتیبانی می شود .

### عملی:

: در مثال زیر ، یک فرم را برای روی صفحه قرار داده ایم . کنترل email متعلق به فرم بوده ، ولی خارج از . با استفاده از خاصیت form ، کنترل را به فرم متصل کرده ایم :

Example	
<pre>&lt;form id="form1" &gt;   First name:&lt;input type="text" name="fname"&gt;&lt;br /&gt;   &lt;input type="submit" value="form1" &gt; &lt;/form&gt; &lt;p&gt; این کنترل خارج از فرم است &lt;/p&gt;</pre>	کد

E-mail: <input form="form1" name="email" type="email"/> 		
First name : <input type="text"/>		
<input type="button" value="form 1"/>		
این کنترل خارج از فرم است		
E-mail :		خروجی

## HTML 5 < formaction > خاصیت جدید کنترل ها در HTML 5 < formaction > خاصیت

### خاصیت formaction :

این خاصیت برای یک کنترل <input> URL فایلی را تعیین می کند که پس از ارسال فرم قرار است ، اطلاعات کنترل را پردازش نماید . این خاصیت مقدار تعیین شده برای خاصیت action فرم را نادیده گرفته و بر مبنای مقدار خود عمل می کند .  
**نکته :** خاصیت formaction فقط برای کنترل های نوع Submit image کار می کند .

شکل کلی استفاده از خاصیت formaction بصورت زیر است :

syntax	<input formaction = " URL ">
--------	------------------------------

توضیح syntax	
توضیح	Parameter
این خاصیت آدرس صفحه یا فایلی که قرار است اطلاعات کنترل را پس از ارسال فرم پردازش کند ، را تعیین می کند .	URL

### شتیانی در مرورگر های مختلف:



این خاصیت در تمام مرورگرهای اصلی پشتیبانی می شود .

: در مثال زیر ، یک فرم را برای روی صفحه قرار داده ایم . این فرم دارای دو کنترل به نام email , name است ، که قرار است email.js مقدار آن را پردازش کند :

Example	
<pre>&lt;form id="form1" action="default.aspx"&gt;   First name:&lt;input type="text" name="fname"&gt;&lt;br /&gt;   E-mail: &lt;input type="email" name="email" form="form1"&gt;&lt;br /&gt;   &lt;input type="submit" value="form1" formaction="email.js" &gt; &lt;/form&gt;</pre>	کد
<p>First name : <input type="text"/></p> <p>E-mail : <input type="text"/></p> <p><input type="submit" value="form1"/></p>	خروجی

## HTML 5 < خواص جدید کنترل ها در HTML 5 < خاصیت formenctype

### خاصیت formenctype :

این خاصیت تعیین می کند که اطلاعات ارسالی از کنترل به سرور با چه روشی کد گذاری شده و ارسال شود ( فقط برای فرم هایی که متد ارسال اطلاعات آنها post است به کار می رود ) .

**نکته :** مقدار این خاصیت در کنترل مقدار تعیین شده برای خاصیت enctype در فرم را نادیده می گیرد .

**نکته :** این خاصیت فقط برای کنترل های نوع submit image بکار می رود .

شکل کلی استفاده از خاصیت formenctype بصورت زیر است :

syntax	<input formenctype = " value ">
--------	---------------------------------

توضیح syntax	
توضیح	Parameter
این پارامتر روش و نحوه کد گذاری اطلاعات ارسالی فرم را به سرور تعیین می کند و می تواند یکی از مقدار زیر :	value
<ul style="list-style-type: none"><li>• <b>application/x-www-form-urlencoded</b> : تمام کاراکتر های ارسالی از کنترل رمز گذاری می شوند . فاصله ها نیز به + تبدیل شده و کاراکترهای خاص نیز به کد متناظر ASCII آن تبدیل می</li></ul>	
<ul style="list-style-type: none"><li>• <b>multipart/form-data</b> : هیچ کدام از کاراکتر های رمز گذاری و Encode نمی شوند .</li><li>• <b>text/plain</b> : فاصله ها به کاراکتر + یل شده ، ولی سایر کاراکتر ها بدون تغییر ارسال می شوند</li></ul>	

### شتیانی در مرورگر های مختلف:



این خاصیت در تمام مرورگرهای اصلی پشتیبانی می شود .

### مثال عملی:

: در مثال زیر ، یک فرم را برای روی صفحه قرار داده و در کنترل submit آن ، روش کدگذاری و ارسال اطلاعات فرم را تعیین

کرده ایم :

Example	
<pre>&lt;form id="form1" method ="post"&gt;   First name:&lt;input type="text" name="fname"&gt;&lt;br /&gt;   E-mail: &lt;input type="email" name="email" &gt;&lt;br /&gt;   &lt;input type="submit" value="form1" formenctype="text/palin" &gt; &lt;/form&gt;</pre>	کد
<p>First name : <input type="text"/></p> <p>E-mail : <input type="text"/></p> <p><input type="submit" value="form1"/></p>	خروجی

## HTML 5 < formmethod > خاصیت جدید کنترل ها در HTML 5 < خاصیت formmethod

### خاصیت formmethod :

خاصیت formmethod HTTP ارسال اطلاعات کنترل ربه سرور را تعیین می کند . خاصیت formmethod تعیین شده در کنترل ، مقدار تعیین شده برای آن خاصیت در فرم را نادیده می گیرد . را به دو روش می توان ارسال کرد . `get` `post` .  
**نکته :** این خاصیت را فقط می توان با کنترل های `submit` `image` به کار برد .

### ویژگی های روش `get` :

- در این روش اطلاعات فرم بصورت جفت های نام و مقدار به انتهای آدرس صفحه اضافه شده و به سرور ارسال می شود.
- این روش معمولا برای زمانی مناسب است که می خواهید اطلاعات ارسالی را به کاربر نیز نمایش دهید.
- ه ارسالی در این روش بسته به نوع مرورگر محدود بوده و ممکن است نتوان همه حجم اطلاعات مورد نظر را ارسال کرد.
- `get` برای ارسال اطلاعات حساس مثل رمز عبور و ... استفاده نکنید . زیرا اطلاعات در نوار آدرس مرور گر قابل مشاهده است.

### ویژگی های روش `post` :

- در این روش صفحه اطلاعات را به روش HTTP و بصورت پیش پرده ارسال می کند.
- اطلاعات ارسالی با روش `POST` قابل مشاهده نیستند.
- `post` `get` ، امنیت بیشتری داشته و حجم اطلاعات ارسالی محدود نخواهد بود.

شکل کلی استفاده از خاصیت formmethod بصورت زیر است :

syntax	<code>&lt;input formmethod = " get   post "&gt;</code>
--------	--

توضیح syntax	
توضیح	Parameter
در این حالت اطلاعات به روش <code>get</code> ارسال می شوند .	<code>get</code>
در این حالت اطلاعات به روش <code>post</code> ارسال می شوند .	<code>post</code>

پشتیبانی در مرورگر های مختلف:





این خاصیت در تمام مرورگرهای اصلی پشتیبانی می .

### مثال عملی:

: در مثال زیر ، یک فرم را برای روی صفحه قرار داده و دو کنترل submit را برای آن تعیین کرده ایم .  
کنترل اول ، با روش تعیین شده در تگ فرم که get اطلاعات را ارسال می کند . اما کنترل دوم این مقدار  
را نادیده گرفته و با روش post که در خاصیت formmethod آن تعیین شده ، اقدام به ارسال اطلاعات می کند :

Example	
<pre>&lt;form id="form1" method ="get"&gt;   First name:&lt;input type="text" name="fname"&gt;&lt;br /&gt;   E-mail: &lt;input type="email" name="email" &gt;&lt;br /&gt;   &lt;input type="submit" value="get" &gt;   &lt;input type="submit" value="post" formmethod="post" &gt; &lt;/form&gt;</pre>	کد
<p>First name : <input type="text"/></p> <p>E-mail : <input type="text"/></p> <p><input type="button" value="get"/> <input type="button" value="post"/></p>	خروجی

## HTML 5 < formnovalidate > خاصیت HTML 5 < formnovalidate > خاصیت

### خاصیت formnovalidate :

خاصیت formnovalidate یک مقدار Boolean است و هنگامی که در تگ یک کنترل تعیین شود ، باعث خواهد شد تا اطلاعات ارسالی آن کنترل به سرور مورد اعتبارسنجی قرار نگیرد . برای مثال فرض کنید در فرم یک کادر متن برای دریافت ایمیل دارید ، که قبل از ارسال اطلاعات به سرور بررسی می کند آیا فرمت ایمیل وارد شده صحیح است یا خیر .

اگر خاصیت formnovalidate را برای کنترل submit آن فرم تعیین کنید ، دیگر اطلاعات ارسالی اعتبارسنجی نشده و حتی اگر کاربر آدرس ایمیل با فرمت غلط نیز وارد کند پذیرفته می شود .  
**نکته :** خاصیت formnovalidate ، فقط برای کنترل از نوع submit به کار می رود .

شکل کلی استفاده از خاصیت formnovalidate بصورت زیر است :

syntax	<code>&lt;input formnovalidate = " formnovalidate "&gt;</code>
--------	--

**نکته :** بیت formnovalidate را به شکل های زیر را نیز می توان نوشت :

syntax	<code>&lt;input formnovalidate&gt;</code> <code>&lt;input formnovalidate=""&gt;</code>
--------	---

### پشتیبانی در مرورگر های مختلف:



این خاصیت در تمام مرورگر های اصلی پشتیبانی می شود .

### مثال عملی:

: در مثال زیر ، دو فرم را برای روی صفحه قرار داده و برای کنترل submit فرم دوم ، خاصیت formnovalidate آن را تنظیم کرده ایم . تعیین این خاصیت باعث شده تا عملیات اعتبارسنجی در کادر ایمیل فرم انجام نشده و کاربر می تواند ، ایمیل با فرمت غلط در آن وارد نموده و ارسال کند :

ملیات اعتبارسنجی صورت گرفته و با اطلاعات غلط ، فرم submit نمی شود .

### Example

```
<form id="form1" >  
  First name:<input type="text" name="fname"><br />  
  E-mail: <input type="email" name="email" ><br />  
  <input type="submit" value="submit" >  
</form>
```

ک

```
<form id="form2" >  
  First name:<input type="text" name="fname"><br />  
  E-mail: <input type="email" name="email" ><br />  
  <input type="submit" value="submit" formnovalidate >  
</form>
```

First name:   
E-mail:

خروجی

First name:   
E-mail:

## HTML 5 < خواص جدید کنترل ها در HTML 5 < خاصیت list

**خاصیت list :** خاصیت list برای یک کنترل input تعیین کننده id `<datalist>` ی که آیتم های از پیش تعیین شده ای را برای آن کنترل نگهداری می کند . این خاصیت برای کنترل هایی که چندین آیتم را به عنوان نگهداری می کنند بکار می رود . شکل کلی استفاده از خاصیت list بصورت زیر است :

syntax	<code>&lt;input list = " datalist_id " &gt;</code>
--------	--

توضیح syntax	
توضیح	Parameter
تعیین کننده id <code>&lt;datalist&gt;</code> ی که برای کنترل مورد نظر یا چندین آیتم را نگهداری می کند .	datalist_id

پشتیبانی در مرورگر های مختلف:




این خاصیت در تمام مرورگر های اصلی پشتیبانی می شود .

و به وسیله یک تگ `<datalist>` یتم های

: در مثال زیر ، یک کنترل input

مورد نظر را برای آن تعیین کرده ایم :

**راهنمایی:** برای مشاهده آیتم های کادر متن ، درون آن کلیک نمایید .

Example	
<pre>&lt;form &gt; &lt;input list="browsers"&gt;  &lt;datalist id="browsers" &gt;   &lt;option value="Internet Explorer"&gt;   &lt;option value="Firefox"&gt;   &lt;option value="Google Chrome"&gt; &lt;/datalist&gt; &lt;/form&gt;</pre>	کد
	خروجی

## HTML 5 < خواص جدید کنترل ها در HTML 5 < خاصیت multiple

### خاصیت multiple :

خاصیت multiple یک مقدار Boolean است و در صورتی که در تگ یک کنترل قرار داده شود کاربر می تواند بیش از یک مقدار در آن کنترل وارد نماید .  
برای مثال در یک کنترل Email می توانید بیش از یک آدرس Email وارد نمایید .  
**نکته :** خاصیت multiple ، فقط با کنترل از نوع email file به کار می رود .

شکل کلی استفاده از خاصیت multiple بصورت زیر است :

syntax	<input multiple >
--------	-------------------

### پشتیبانی در مرورگر های مختلف:



این خاصیت در تمام مرورگرهای اصلی پشتیبانی می شود .

### مثال عملی:

در مثال زیر ، یک کنترل file ، و مقدار خاصیت multiple را برای آن تنظیم نموده ایم . بنابراین شما می توانید بیش از یک فایل را برای کنترل file انتخاب نمایید :

در هنگام انتخاب فایل ها برای ارسال ، چندین فایل را از منوی باز شده انتخاب نمایید ، تا نتیجه را در خروجی مشاهده کنید .

Example	
<pre>&lt;form &gt;   Select images : &lt;input type="file" name="img" multiple &gt;   &lt;input type="submit" value="Send"&gt; &lt;/form&gt;</pre>	کد
<p>Select images: <input type="file"/> <input type="button" value="Send"/></p>	خروجی

## HTML 5 < خواص جدید کنترل ها در HTML 5 < خاصیت max

### خاصیت max :

خاصیت Max تعیین کننده بیشترین مقدار است که کاربر می تواند در فیلد های عددی یا تاریخ وارد نماید . مقادیر بیش از مقدار تعیین شده قبول نخواهند شد .  
**نکته :** با استفاده همزمان از دو خاصیت max min می توانید محدوده اعداد مجاز برای هر کنترل تعیین نمایید

**نکته :** خاصیت max با کنترل های زیر به کار می رود :

- number
- date
- time
- rang
- month

شکل کلی استفاده از خاصیت max بصورت زیر است :

syntax	<input max = " value   date " >
--------	---------------------------------

توضیح syntax	
توضیح	Parameter
تعیین کننده حداکثر مقدار عددی یا تاریخ است که کاربر می تواند در کنترل وارد نماید .	value   date

### پشتیبانی در مرورگر های مختلف:



این خاصیت در تمام مرورگر های اصلی پشتیبانی می شود .

### مثال عملی:

**number** : در مثال زیر ، یک کنترل number قرار خاصیت max را برای آن را بر روی مقدار تنظیم نموده ایم . شما می توانید حداکثر عدد را در این کنترل وارد نمایید و عددهای های بزرگتر را خطا می دهد :

### Example

```
<form >  
  <input type="number" max="10" >  
  <input type="submit" value="Send">  
</form>
```

کد

Send

خروجی

## HTML 5 < HTML 5 > خواص جدید کنترل ها در HTML 5 < خاصیت min

### خاصیت min :

خاصیت min تعیین کننده کمترین مقدار است که کاربر می تواند در فیلد های عددی یا تاریخ وارد نماید . مقادیر کمتر از مقدار تعیین شده قبول نخواهند شد .

**نکته :** با استفاده همزمان از دو خاصیت min max می توانید محدوده اعداد مجاز برای هر کنترل تعیین نمایید

**نکته :** خاصیت min با کنترل های زیر به کار می رود :

- number
- date
- time
- rang
- month

شکل کلی استفاده از خاصیت min بصورت زیر است :

syntax	<input min = " value   date " >
--------	---------------------------------

توضیح syntax	
توضیح	Parameter
تعیین کننده حداقل مقدار عددی یا تاریخ است که کاربر می تواند در کنترل وارد نماید .	value   date

### پشتیبانی در مرورگر های مختلف:



این خاصیت در تمام مرورگر های اصلی پشتیبانی می شود .

### مثال عملی:

**در مثال زیر ، یک کنترل number** و مقدار خاصیت min را برای آن را بر روی مقدار تنظیم نموده ایم . شما می توانید حداقل عدد را در این کنترل وارد نمایید و عددهای های کوچکتر را خطا می دهد :



### Example

```
<form method="get" >  
  <input type="number" min="10" >  
  <input type="submit" value="Send">  
</form>
```

کد

Send

خروجی

## HTML 5 < خواص جدید کنترل ها در HTML 5 < خاصیت placeholder

### خاصیت placeholder :

خاصیت placeholder یک متن راهنما را برای کمک به کاربر جهت پر کردن کادر متن تعیین می کند . برای مثال یک کادر متن دارید که کاربر بایستی نام خود را در آن وارد کند . می توانید یک متن مثل "Name" این خاصیت تعیین کرده و کاربر با مشاهده آن راهنمایی شود . با شروع تایپ کردن کاربر در کادر متن ، متن راهنما به صورت خودکار پاک می شود :

**نکته :** خاصیت placeholder با کنترل های زیر به کار می رود .

- text
- search
- url
- email
- password

شکل کلی استفاده از خاصیت placeholder بصورت زیر است :

syntax	<code>&lt;input placeholder = " text " &gt;</code>
--------	--

توضیح syntax	
توضیح	Parameter
این پارامتر تعیین کننده متن راهنما است .	text

### پشتیبانی در مرورگر های مختلف:



این خاصیت در تمام مرورگرهای اصلی پشتیبانی می شود .

### مثال عملی:

: در مثال زیر ، یک فرم را بر روی صفحه قرار داده ایم که دارای دو کادر متن بوده و کاربر باید نام و نام خانوادگی خود را در آن وارد نماید .

برای راهنمای :

### Example

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

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

```
  <input type="text" name ="family" placeholder="نام خانوادگی" >
```

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

```
</form>
```

کد

خروجی

## HTML 5 < HTML 5 خواص جدید کنترل ها در HTML 5 > خاصیت pattern

### خاصیت pattern :

خاصیت pattern برای یک کنترل <input> ، تعیین کننده ساختاری است که باید اطلاعات وارد شده در کنترل قبل از ارسال به سرور را بر مبنای آن اعتبارسنجی شوند .

برای مثال یک ساختار تعیین می کنید که رمز وارد شده در کنترل حداقل باید دارای کاراکتر باشد . کنترل قبل از ارسال اطلاعات به سرور ، این شرط تعیین شده را بررسی می کند .  
**نکته :** خاصیت pattern با کنترل های زیر به کار می رود :

- text
- search
- url
- email
- password

**نکته :** در هنگام استفاده از خاصیت pattern ، از خاصیت title کنترل <input> نیز ، برای نمایش یک tooltip یا راهنما جهت پر کردن کادر متن به کاربر استفاده نمایید .

**نکته :** برای دریافت اطلاعات بیشتر به بخش آموزش اعتبارسنجی داده ها در جاوا اسکریپت بروید .

شکل کلی استفاده از خاصیت pattern بصورت زیر است :

syntax	<input pattern = " regexp " >
توضیح syntax	
توضیح	Parameter
تعیین کننده ساختار یا الگویی است که قرار است اطلاعات ورودی در کادر متن ، بر مبنای آنها اعتبارسنجی شود .	regexp

پشتیبانی در مرورگر های مختلف:



این خاصیت در تمام مرورگرهای اصلی پشتیبانی می شود .

## مثال عملی :

: در مثال زیر ، یک فرم را بر روی صفحه قرار داده ایم که دارای یک کادر متن بوده و کاربر باید یک کد حرفی را در آن وارد کرده و فرم را ارسال نماید . برای راهنمایی وی نیز ، متنی در خاصیت **title** در نظر گرفته . در صورت وارد کردن مقدار صحیح ، فرم ارسال می شود :

Example	
<pre>&lt;form method="get"&gt;   یک کد : &lt;input type="text" name="country_code" pattern="[A-Za-z]{3}" title="یک کد حرفی وارد نمایید" &gt;   &lt;input type="submit" value="Send"&gt; &lt;/form&gt;</pre>	کد
<p>کد حرفی : <input type="text"/></p> <p><input type="button" value="Send"/></p>	خروجی

## HTML 5 < خواص جدید کنترل ها در HTML 5 > خاصیت required

### خاصیت required :

خاصیت required یک مقدار Boolean است و اگر آن را در تگ یک کنترل <input> قرار دهید ، تعیین می کند که کاربر حتما باید در کنترل مقداری وارد کرده و آن را خالی رها نکند . اگر کاربر مقداری در کنترل وارد نکند و بخواهد فرم را ارسال نماید برنامه error داده و مانع ارسال آن می شود .

**نکته :** خاصیت required با کنترل های زیر به کار می رود .

text , search , url , email, password, date pickers , number , checkbox , radio , file

شکل کلی استفاده از خاصیت required بصورت زیر است :

syntax	<input required >
--------	-------------------


### پشتیبانی در مرورگر های مختلف:



این خاصیت در تمام مرورگرهای اصلی پشتیبانی می شود .

### مثال عملی:

: در مثال زیر ، یک فرم را بر روی صفحه قرار داده ایم که دارای دو کادر متن بوده و کاربر باید نام و نام خانوادگی خود را در آن وارد نماید . برای کادر اول که نام است ، مقدار خاصیت required را تنظیم نکرده ایم . بنابراین کاربر می تواند آن را پر نکرده و خالی ارسال نماید . اما برای کادر دوم که نام خانوادگی بوده ، مقدار خاصیت required ده ایم . بنابراین بایستی حتما مقداری را در آن وارد نماید :

Example	
<pre>&lt;form method="get"&gt;   &lt;input type="text" name ="name" &gt;   &lt;input type="text" name ="family" required &gt;   &lt;input type="submit" value="Send"&gt; &lt;/form&gt;</pre>	کد
	خروجی

## HTML 5 < خواص جدید کنترل ها در HTML 5 < خاصیت step

### خاصیت step :

خاصیت step برای یک کنترل input گام افزایش و یا کاهش مقدار آن یا به عبارت دیگر جهت وارد کردن در کنترل را تعیین می کند. برای مثال اگر مقدار این خاصیت را برای کنترلی " step = " تعیین نمایید ، مقادیر مجاز برای وارد کردن در آن عبارتند از : ... , , - , ... یا به عبارت دیگر . هر مقداری به غیر از م Step غیر قابل قبول خواهد بود .

**نکته :** خاصیت step با کنترل های زیر به کار می رود .

- number
- range
- month
- date
- time

**نکته :** از خاصیت step می توانید همراه با خواص min max برای تعیین محدوده مجاز اعداد وارد شده در یک کنترل استفاده نمایید .

شکل کلی استفاده از خاصیت step بصورت زیر است :

syntax	<input step = " number " >
--------	----------------------------

توضیح syntax	
توضیح	Parameter
این پارامتر تعیین کننده گام افزایش یا کاهش مقدار کنترل است .	number


پشتیبانی در مرورگر های مختلف:



این خاصیت در تمام مرورگر های اصلی به جز فایرفاکس پشتیبانی می شود .

**مثال عملی:**

: در مثال زیر ، یک فرم را بر روی صفحه قرار داده ایم که دارای یک کنترل عدد number . خاصیت step آن بر روی مقدار تنظیم شده است . شما می توانید فقط مصرف های را در کنترل وارد نمایید :

Example	
<pre>&lt;form method="get"&gt;   &lt;input type="number" step="3" &gt;   &lt;input type="submit" value="Send"&gt; &lt;/form&gt;</pre>	کد
	خروجی



## HTML 5 < تگ های جدید فرم در HTML 5 < تگ های جدید فرم در

### HTML 5

#### معرفی تگ های جدید فرم در HTML 5 :

HTML 5 برای کارایی بهتر و سهولت طراحی فرم های وب المنت جدید را به تگ های فرم اضافه کرده است . این تگ ها عبارتند از :

<a href="#">&lt;datalist&gt;</a>	این تگ چندین آیتم از پیش تعیین شده را برای یک کنترل لیست <input> تعیین می کند .
<a href="#">&lt;keygen&gt;</a>	از این تگ برای ایجاد یک کادر ارسال رمز یا کلمه عبور به سرور استفاده می شود .
<a href="#">&lt;output&gt;</a>	از این تگ برای محاسبه و نمایش خروجی یک عمل ریاضی در HTML استفاده می شود .

**نکته مهم :** تمام مرورگرهای مطرح ممکن است از این تگ های جدید پشتیبانی نکنند ، اما به هر حال شما می توانید از آنها در کدهای خود استفاده کنید . اگر مرورگر از تگ جدید پشتیبانی نکند ، آن را بصورت یک کادر متن معمولی نمایش می دهد .

## <datalist> < HTML 5 تگ های جدید فرم در HTML 5

### : <datalist>

<datalist> یک لیست از چند آیتم از پیش تعیین شده را برای یک کنترل <input> تعیین می کند . هدف از تعیین این تگ برای کنترل <input> تهیه یک لیست برای تکمیل اتوماتیک گزینه هاست .  
به عبارت دیگر هنگامی که این تگ را برای یک کنترل تعیین می کنید ، در هنگام کلیک بر روی آن کنترل لیست آیتم ها بصورت یک منو به کاربر نمایش داده می شود .

از خاصیت List کنترل <input> برای اتصال آن به تگ <datalist> استفاده می شود .  
**نکته:** هر آیتم تگ <datalist> توسط یک تگ <option> تعیین می شود .

شکل کلی استفاده از تگ <datalist> بصورت زیر است :

syntax	<pre>&lt;input list="datalist_id"&gt;  &lt;datalist id="datalist_id"&gt;   &lt;option value="value 1"&gt;   &lt;option value="value 2"&gt;   &lt;option value="value 3"&gt; &lt;/datalist&gt;</pre>
--------	---

### توضیح موارد syntax :

توضیح syntax	
یچ	Parameter
یک id را برای تگ <datalist> تعیین می نماید .	datalist_id
یک آیتم را برای تگ <datalist> ایجاد می نماید .	option
مقدار هر آیتم تگ <datalist> را تعیین می نماید .	value


### پشتیبانی در مرورگر های مختلف:



این تگ در مرورگرهای کروم ، فایرفاکس و اپرا پشتیبانی می شود .

## مثال عملی:

: در مثال زیر یک کنترل `<input>` تگ را ایجاد کرده و به وسیله یک تگ `<datalist>` ، آیتم های آن را تعیین نموده ایم . برای مشاهده آیتم های از پیش تعیین شده برای کنترل `input` ، موس را داخل کادر متن آن کلیک نمایید :

Example	
<pre>&lt;form action="demo_form.asp" method="get"&gt;   &lt;input list="browsers" name="browser"&gt;   &lt;datalist id="browsers"&gt;     &lt;option value="Internet Explorer"&gt;     &lt;option value="Firefox"&gt;     &lt;option value="Chrome"&gt;     &lt;option value="Opera"&gt;   &lt;/datalist&gt;   &lt;input type="submit"&gt; &lt;/form&gt;</pre>	کد
	خروجی

## <keygen> < HTML 5 تگ های جدید فرم در HTML 5

### : <keygen>

هدف <keygen> در یک فرم ، ایجاد راهی برای شناسایی هویت کاربران است . این تگ یک فیلد کلید جفتی ( کلید رمز نگاری شده ) را در فرم های Html ایجاد می کند . هنگامی که کاربر مقدار را در فیلد وارد کرده و فرم را ارسال می کند ، دو کلید تولید می شود . یکی کلید خصوصی و دیگری کلید عمومی . کلید خصوصی بصورت محرمانه بر روی کامپیوتر کاربر ذخیره شده و کلید عمومی به سرور ارسال می شود . کلید عمومی می تواند به عنوان یک شناسه یا مدرک برای شناسایی کاربر در آینده استفاده می شود . برنامه برای شناسایی کاربر در آینده مقدار جفت کلید عمومی و خصوصی را مقایسه کرده و هویت آن را شناسایی می کند .

شکل کلی استفاده از تگ <keygen> بصورت زیر است :

syntax	<code>&lt;keygen id="id" name="name"&gt;</code>
--------	---

### پشتیبانی در مرورگر های مختلف:



این تگ در مرورگرهای کروم ، فایرفاکس و اپرا پشتیبانی می شود .

### مثال عملی:

: در مثال زیر یک کنترل <input> تگ را ایجاد کرده و به وسیله یک تگ <keygen> آن را ، قبل از ارسال به سرور ، کد گذاری کرده ایم .

برای مشاهده خروجی مثال ، یک نام را در کادر متن وارد کرده و سپس Enter را بزنید . صفحه جاری به صفحه مقصد ارسال شده و دو پارامتر به نام های security user\_name به نوار آدرس صفحه اضافه می شوند :

Example	
<pre>&lt;form action="receive_keygen.html" method="get"&gt;   Username: &lt;input type="text" name="user_name"   Encryption: &lt;keygen name="security"&gt;   &lt;input type="submit"&gt; &lt;/form&gt;</pre>	کد

## <output> < HTML 5 تگ های جدید فرم در HTML 5

### : <output>

این تگ می تواند خروجی یک عمل محاسباتی را بر روی صفحه نمایش دهد . این عمل محاسباتی معمولاً توسط یک اسکریپت اجرا می شود .

شکل کلی استفاده از تگ <output> بصورت زیر است :

syntax	<code>&lt;output name="name" for="value"&gt; &lt;/output&gt;</code>
--------	---

### توضیح موارد syntax :

توضیح syntax	
توضیح	Parameter
این خاصیت یک نام را برای کنترل تعیین می کند .	name
این خاصیت ارتباط بین نتیجه محاسبه و المنت هایی که در عمل محاسبه بکار برده شده مشخص می کند . به مثال دقت نمایید .	for

### پشتیبانی در مرورگر های مختلف :



این تگ در مرورگرهای کروم ، فایرفاکس و اپرا پشتیبانی می شود .

: در مثال زیر یک عمل محاسباتی جمع مقدار دو کنترل را انجام داده و خروجی آن را توسط یک تگ

<output> ، نمایش داده ایم :

Example	
<pre>&lt;form action="output.aspx" oninput="x.value=parseInt(a.value)+parseInt(b.value)"&gt; 0   &lt;input type="range" id="a" value="50"&gt; 100 + &lt;input type="number" id="b" value="50"&gt;   = &lt;output name="x" for="a b"&gt;&lt;/output&gt; &lt;/form&gt;</pre>	کد

## HTML 5 < کار با فرم ها در HTML 5 < خاصیت autocomplete

### خاصیت autocomplete :

خاصیت autocomplete تعیین می کند که آیا کادر های متن موجود در فرم بایستی قابلیت پر شدن خودکار داشته باشند یا خیر . هنگامی که فیلد های موجود در یک فرم قابلیت پر شدن خودکار دارند ، مرورگر بصورت اتوماتیک مقادیر آنها را بر حسب مقادیری که قبلا کاربر وارد نموده است ، پر می کند . این خاصیت در حالت پیش فرض خاموش است و در یک فرم با کنترل های زیر کار می کند :

- text
- search
- url
- password
- date
- rang
- email
- color

**نکته** : می توان قابلیت پر شدن خودکار یک فرم را بر روی مقدار روشن تنظیم نمود ، ولی همین قابلیت را برای کنترل موجود در آن خاموش کرد .  
**نکته** : در برخی از مرورگرها بایستی در تنظیمات قابلیت پر شدن خودکار را فعال کرد .

شکل کلی استفاده از خاصیت autocomplete بصورت زیر است :

syntax	<input autocomplete = " on   off " >
--------	--------------------------------------

توضیح syntax	
توضیح	Parameter
در این حالت قابلیت پر شدن خودکار فیلد ها در فرم فعال است .	on
در این حالت قابلیت پر شدن خودکار فیلد ها در فرم غیر فعال است . مقدار پیش فرض است .	off

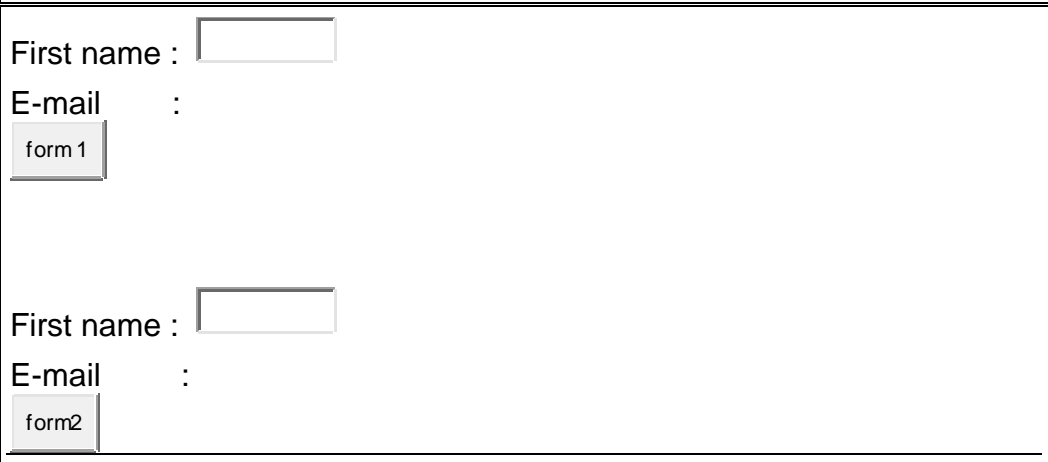
پشتیبانی در مرورگر های مختلف:



این خاصیت در تمام مرورگرهای اصلی به جز اپرا ، پشتیبانی می شود .

### مثال عملی:

: در مثال زیر ، دو فرم را برای روی صفحه قرار داده ایم . در فرم اول مقدار خاصیت autocomplete روی on و در فرم دوم روی off تنظیم شده است .

Example	
<pre>&lt;form id="form1" autocomplete="on"&gt;   First name:&lt;input type="text" name="fname"&gt;&lt;br /&gt;   E-mail: &lt;input type="email" name="email"&gt;&lt;br /&gt;   &lt;input type="submit" value="form1" &gt; &lt;/form&gt; &lt;br /&gt; &lt;br /&gt; &lt;form id="form2" autocomplete="off"&gt;   First name:&lt;input type="text" name="fname"&gt;&lt;br /&gt;   E-mail: &lt;input type="email" name="email"&gt;&lt;br /&gt;   &lt;input type="submit" value="form2"&gt; &lt;/form&gt;</pre>	کد
	خروجی

## HTML 5 < کار با فرم ها در HTML 5 < خاصیت novalidate

### خاصیت novalidate :

این خاصیت در یک فرم تعیین می کند که آیا مقادیر وارد شده در فیلد های آن در هنگام ارسال اعتبارسنجی شوند یا خیر .

این خاصیت یک مقدار Boolean است و اگر آن را در تگ ابتدایی <form> قرار دهید ، مانع اعتبارسنجی داده های ورودی در فیلد های فرم می شود . برای مثال اگر در یک کنترل Email در فرم وجود داشته باشد و این خاصیت را برای فرم تنظیم کرده باشید ، حتی اگر آدرس ایمیل با فرمت نادرست نیز وارد کنید ، فرم ارسال می شود .

شکل کلی استفاده از خاصیت novalidate بصورت زیر است :

syntax	<form novalidate >
--------	--------------------

### پشتیبانی در مرورگر های مختلف:



این خاصیت در تمام مرورگرهای اصلی پشتیبانی می شود .

### مثال عملی:

: در مثال زیر ، دو فرم را برای روی صفحه قرار داده برای فرم دوم مقدار خاصیت novalidate تنظیم کرده ایم . همانطور که می بینید ، در فرم دوم قابلیت اعتبار سنجی آن از بین رفته و می توانید ایمیل با فرمت های نا صحیح نیز وارد نمایید .:

Example	
<pre>&lt;form id="form1" &gt;   First name:&lt;input type="text" name="fname"&gt;&lt;br /&gt;   E-mail: &lt;input type="email" name="email"&gt;&lt;br /&gt;   &lt;input type="submit" value="form1" &gt; &lt;/form&gt; &lt;br /&gt; &lt;br /&gt; &lt;form id="form2" novalidate&gt;</pre>	کد



<pre>First name:&lt;input type="text" name="fname"&gt;&lt;br /&gt; E-mail: &lt;input type="email" name="email"&gt;&lt;br /&gt; &lt;input type="submit" value="form2"&gt; &lt;/form&gt;</pre>	
<p>First name : <input type="text"/></p> <p>E-mail : <input type="text"/></p> <p><input type="button" value="form 1"/></p> <p>First name : <input type="text"/></p> <p>E-mail : <input type="text"/></p> <p><input type="button" value="form2"/></p>	<p>خروجی</p>

## HTML 5 - پخش فایل های تصویری در صفحات وب

### نمایش ویدیو در صفحات HTML :

بسیاری از صفحات وب به پخش و اجرای فایل های ویدیویی و فلش می پردازند . تا قبل از ارائه HTML 5 استاندارد خاصی برای نمایش فایل های ویدیویی در صفحات وب وجود نداشت و طراحان وب برای نمایش اینگونه فایل ها از ابزارهای جانبی مثل **flash player** استفاده می کردند . این مسئله چند اشکال عمده دارد . اول اینکه باید **plugin** فلش پلیر بر روی مرورگر شما نصب شده باشد و ورژن آن به روز بوده تا فایل ویدیویی را نمایش دهد .

HTML 5 ، یک استاندارد برای نمایش فایل های ویدیویی تعیین کرده است که به وسیله آن ، دیگر نیازی به استفاده و نصب نرم افزار جانبی خاصی وجود ندارد .

تگ جدید **< video >** ، برای نمایش فایل های ویدیویی در صفحات وب به کار می رود . در جدول زیر نحوه کلی استفاده از این تگ را به شما نمایش داده و سپس به توضیح موارد مطرح شده در آن می پردازیم :





شکل کلی	<pre>&lt;video width="320" height="240" controls="controls"&gt;   &lt;source src="movie.mp4" type="video/mp4" /&gt;   &lt;source src="movie.ogv" type="video/ogg" /&gt;   Your browser does not support the video tag. &lt;/video&gt;</pre>
< video >	

در این بخش به تشریح کد بالا و مواردی که ممکن است برای شما سوال ایجاد کند می پردازیم :

- width : height** این خواص ابعاد فریم نمایش ویدیو را تعیین می کنند . تعیین این پارامترها اجباری نیست ، اما تعیین آنها از قبل ، این امکان را به مرورگر و طراح صفحه می دهد تا قبل از اجرای صفحه میزان فضای لازم برای این تگ را در نظر گرفته و طرح صفحه به هم نریزد . اگر مقدار این خواص تعیین نشوند ، فریم پخش ویدیو به اندازه سایز فایل ویدیویی در خواهد آمد .
- خاصیت : controls** مقدار دهی خاصیت **controls** ، باعث می شود تا مرورگر دکمه های پخش ، توقف و تنظیم صدا را به کاربر نمایش دهد . در حالت پیش فرض این کنترل ها نمایش داده نمی شوند .
- تعیین چند فرمت برای فایل ویدیویی توسط تگ : < source >** همانطور که متوجه شده اید ، چند فرمت مختلف از یک فایل یکسان را به وسیله تگ **< source >** **< video >** قرار داده ایم . این کار به این دلیل است که مرورگرهای مختلف از فرمت های خاص ویدیویی پشتیبانی می کنند که اسامی آنها در جدول زیر آمده است . هر مرورگر هنگامی که به این کد می رسد ، فرمتی که خود می تواند اجرا کند را پخش می کند . این کار برای سازگاری با مرورگرهای مختلف است .

نوع فرمت فایل های ویدیویی داریم که پشتیبانی یا عدم پشتیبانی آنها توسط مرورگر های مختلف را

در جدول زیر نمایش داده ایم :

Ogg	WebM	MP4	
Yes	Yes	No	
Yes	Yes	No	
No	No	Yes	
Yes	Yes	Yes	

### تگ های مربوط به پخش ویدیو در صفحات وب :

در لیست زیر تگ های اصلی و مهم مرتبط با پخش ویدیو در HTML 5 را به همراه توضیحات مختصری راجع به هر کدام ، قرار داده ایم . برای دریافت اطلاعات بیشتر بر روی نام آن تگ کلیک نمایید :

این تگ اصلی پخش ویدیو در HTML 5 است ، که به وسیله آن کد لازم برای نمایش ویدیو را قرار می دهیم .	<code>&lt; video &gt;</code>
این تگ آدرس و نوع فایل ویدیویی که می خواهیم نمایش دهیم را تعیین می کند .	<code>&lt; source &gt;</code>

### مثال عملی استفاده از تگ < video > و نمایش ویدیو در صفحه:

در مثال زیر کاربرد تگ < video > را در عمل و نمایش یک ویدیو را به شما نمایش داده ایم :  
**نکته :** < video > پشتیبانی نکند ، پیام هشدار قرار داده شده در کد مثال نمایش داده خواهد

Example	
<pre>&lt;video width="320" height="240" controls="controls"&gt;   &lt;source src="movie.mp4" type="video/mp4" /&gt;   &lt;source src="movie.ogg" type="video/ogg" /&gt;   Your browser does not support the video tag. &lt;/video&gt;</pre>	کد

: در مثال دوم ، همان مثال قبل را بازنویسی کرده ایم ، با این تفاوت در آن خاصیت `controls` را مقدار دهی نکرده ایم . می بینید که ویدیو دکمه های پخش و توقف را ندارد :

**نکته :** `< video >` پشتیبانی نکند ، پیام هشدار قرار داده شده در کد مثال نمایش داده خواهد

Example	
<pre>&lt;video width="320" height="240" &gt;   &lt;source src="movie.mp4" type="video/mp4" /&gt;   &lt;source src="movie.ogg" type="video/ogg" /&gt;   Your browser does not support the video tag. &lt;/video&gt;</pre>	کد

## HTML 5 – پخش فایل های صوتی در صفحات وب

### پخش صوت و آهنگ در صفحات HTML :

ممکن است در صفحات وب نیاز داشته باشید تا به پخش و اجرای فایل های صوتی و آهنگ بپردازید . تا قبل از ارائه HTML 5 ، استاندارد خاصی برای نمایش فایل های صوتی در صفحات وب وجود نداشت و طراحان وب برای پخش اینگونه فایل ها از ابزارهای جانبی مثل `real palyer` `falsh player` استفاده می کردند . این مسئله چند اشکال عمده دارد . اول اینکه باید `plugin` فلش پلیئر بر روی مرورگر شما نصب شده باشد و ورژن آن به روز بوده تا فایل صوتی را پخش کند .

HTML 5 ، یک استاندارد برای پخش فایل های صوتی تعیین کرده است که به وسیله آن ، دیگر نیازی به استفاده و نصب نرم افزار جانبی خاصی وجود ندارد .

تگ جدید `< audio >` ی پخش فایل های صوتی در صفحات وب به کار می رود .

در جدول زیر نحوه کلی استفاده از این تگ را به شما نمایش داده و سپس به توضیح موارد مطرح شده در آن می پردازیم :

شکل کلی	<code>&lt;audio controls="controls"&gt;</code>
<code>&lt; audio &gt;</code>	<code>&lt;source src="song.mp3" type="audio/mp3" /&gt;</code> <code>&lt;source src="song.ogg" type="audio/ogg" /&gt;</code> Your browser does not support the audio elemet. <code>&lt;/audio&gt;</code>





در این بخش به تشریح کد بالا و مواردی که ممکن است برای شما سوال ایجاد کند می پردازیم :

- خاصیت controls :** مقدار دهی خاصیت `controls` ، باعث می شود تا مرورگر دکمه های پخش ، توقف و تنظیم صدا را به کاربر نمایش دهد . در حالت پیش فرض این کنترل ها نمایش داده نمی شوند.
- تعیین چند فرمت برای فایل صوتی توسط تگ source :** `< source >` همانطور که متوجه شده اید ، چند فرمت مختلف از یک فایل یکسان را به وسیله تگ `< source >` `< audio >` قرار داده ایم . این کار به این دلیل است که مرورگرهای مختلف از فرمت های خاص صوتی پشتیبانی می کنند که اسامی آنها در جدول زیر آمده است . هر مرورگر هنگامی که به این کد می رسد ، فرمتی که خود می تواند اجرا کند را پخش می کند . این کار برای سازگاری با مرورگرهای مختلف است.

نوع فرمت فایل های ویدیویی داریم که پشتیبانی یا عدم پشتیبانی آنها توسط مرورگر های مختلف را

در جدول زیر نمایش داده ایم :

Ogg	WebM	MP3	
-----	------	-----	--

Yes	Yes	No	
Yes	Yes	No	
No	No	Yes	
Yes	Yes	Yes	

### تگ های مربوط به پخش صوت و آهنگ در صفحات وب :

در لیست زیر تگ های اصلی و مهم مرتبط با پخش صوت در HTML 5 را به همراه توضیحات مختصری راجع به هر کدام ، قرار داده ایم . برای دریافت اطلاعات بیشتر بر روی نام آن تگ کلیک نمایید :

این تگ ، تگ اصلی پخش صوت در HTML 5 است ، که به وسیله آن کد لازم برای نمایش صوت را قرار می دهیم .	<code>&lt; audio &gt;</code>
این تگ آدرس و نوع فایل صوتی که می خواهیم پخش کنیم را تعیین می کند .	<code>&lt; source &gt;</code>

### مثال عملی استفاده از تگ `< audio >` و نمایش ویدیو در صفحه:

در مثال زیر کاربرد تگ `< audio >` عمل و پخش یک آهنگ را به شما نمایش داده ایم :  
**نکته :** `< audio >` پشتیبانی نکند ، پیام هشدار قرار داده شده در کد مثال نمایش داده خواهد

Example	
<pre>&lt;audio controls="controls"&gt;   &lt;source src="song.mp3" type="audio/mp3" /&gt;   &lt;source src="song.ogg" type="audio/ogg" /&gt;   Your browser does not support the audio tag. &lt;/audio&gt;</pre>	کد

در مثال دوم ، همان مثال قبل را بازنویسی کرده ایم ، با این تفاوت در آن خاصیت `controls` را مقدار دهی نکرده ایم . می بینید که پخش صوت دکمه های پخش و توقف را :  
**نکته :** `< audio >` پشتیبانی نکند ، پیام هشدار قرار داده شده در کد مثال نمایش داده خواهد

### Example

```
<audio >  
  <source src="song.mp3" type="audio/mp3" />  
  <source src="song.ogg" type="audio/ogg" />  
  Your browser does not support the audio tag.  
</audio>
```

ک

## < HTML 5 > اجرای مدیا در HTML 5 < embed >

### < embed > :

از این تگ برای تعیین ظرف یا نگهدارنده جهت قرار دادن یک برنامه خارجی یا مولتی مدیا ( مثل فلش یا SWF ) بر روی صفحه استفاده می شود . برای مثال می توان به فایل های فلش یا انواع Plug-in برای مرورگر ها اشاره  
شکل کلی استفاده از این تگ به صورت زیر است :

syntax	<embed src="آدرس فایل">
--------	-------------------------

### انواع خاصیت های تگ < embed > :

در لیست زیر ، خواص تگ < embed > و نحوه استفاده از آنها را تشریح کرده ایم :

< embed >	
خاصیت	توضیح
height	این پارامتر ارتفاع عنصر را تعیین می کند . تعیین این پارامتر ، اختیاری است .
width	این پارامتر عرض عنصر را تعیین می کند . تعیین این پارامتر ، اختیاری است .
src	این پارامتر مکان یا آدرس فایل مورد نظر برای اجرا ، را تعیین می کند . تعیین این پارامتر ، اجباری است .
type	این خاصیت نوع و فرمت فایل مورد نظر را تعیین می کند . که از جمله آنها می توان به انواع فرمت های رایج صوتی و تصویری اشاره کند . تعیین این پارامتر ، اختیاری است .

### پشتیبانی در مرورگر های مختلف:



این تگ در تمام مرورگر های اصلی پشتیبانی می شود .



: در مثال زیر به وسیله تگ < embed > ، یک فایل فلش را بر روی صفحه قرار داده و اجرا کرده ایم :

Example	
<pre>&lt;embed src="hello.swf" type="swf" height="100px" width="100px"&gt;</pre>	کد

## < audio > < HTML 5 > اجرای مدیا در HTML 5

### : < audio >

از این تگ برای قرار دادن و اجرای فایل های صوتی و موزیک بر روی صفحات وب در HTML 5 استفاده می شود . این تگ یک تگ جدید در HTML 5 است و اکثر مرورگرهای مطرح از آن پشتیبانی می کنند .  
**نکته :** گر متنی بین تگ باز و بسته <audio> قرار دهید در صورتی که مرورگر از آن تگ پشتیبانی نکند ، متن به جای فایل صوتی نمایش داده می شود .

**نکته :** برای آشنایی بیشتر با نحوه استفاده این تگ بر روی صفحات وب ، به بخش آموزش قرار دادن فایل های صوتی در HTML 5 بروید .  
شکل کلی استفاده از این تگ به صورت زیر است :

syntax	<audio src="آدرس فایل صوتی">
--------	------------------------------

### انواع خاصیت های تگ < audio > :

در لیست زیر ، خواص تگ < audio > و نحوه استفاده از آنها را تشریح کرده ایم :

< audio >	
توضیح	خاصیت
این خاصیت یک مقدار Boolean است و در صورتی که برای یک تگ <audio> تعیین شود، باعث می شود تا فایل صوتی به محض آماده شدن و لود کامل صفحه ، شروع به اجرا نماید. تعیین این پارامتر ، اختیاری است .	autoplay
این خاصیت یک مقدار Boolean می باشد که برای یک تگ <audio> تعیین شود، باعث می شود تا کنترل های پخش ... برای فایل صوتی بر روی صفحه نمایش داده شود . تعیین این پارامتر ، اختیاری است .	controls
این پارامتر مکان یا آدرس فایل صوتی مورد نظر برای اجرا ، را تعیین می کند . تعیین این پارامتر ، اجباری است .	src
این خاصیت یک مقدار Boolean است و در صورتی که برای یک تگ <audio> تعیین شود، اجرای مجدد فایل صوتی ، پس از پایان یافتن آن می شود . تعیین این پارامتر ، اختیاری است .	loop
این خاصیت یک مقدار Boolean است و در صورتی که برای یک تگ <audio> تعیین شود، می شود تا فایل صوتی در ابتدای اجرا ، بدون صدا باشد.	muted

تعیین این پارامتر ، اختیاری است .

پشتیبانی در مرورگر های مختلف :



این تگ در تمام مرورگرهای اصلی پشتیبانی می شود .

مثال عملی:

: در مثال زیر به وسیله تگ `< audio >` ، یک فایل صوتی را بر روی صفحه قرار داده و اجرا کرده ایم :  
نکته : `< source >` ، می توان برای تعیین فایل های صوتی مختلف برای تگ `< audio >` استفاده کرد .

### Example

- به صورت مستقیم

```
<audio controls autoplay src="song.ogg">  
  Your browser does not support the audio tag.  
</audio>
```

- تعیین فایل در یک تگ `< source >`

```
<audio controls="controls">  
  <source src="song.mp3" type="audio/mp3" />  
  <source src="song.ogg" type="audio/ogg" />  
  Your browser does not support the audio tag.  
</audio>
```

کد

## < HTML 5 > اجرای مدیا در HTML 5 < video >

### : < video >

از این تگ برای قرار دادن و اجرای فایل های تصویری و متحرک بر روی صفحات وب در HTML 5 استفاده می کنند . این تگ یک تگ جدید در HTML 5 است و اکثر مرورگرهای مطرح از آن پشتیبانی می کنند .  
**نکته :** اگر متنی بین تگ باز و بسته <video> قرار دهید در صورتی که مرورگر از آن تگ پشتیبانی نکند ، متن به جای فایل تصویری نمایش داده می شود .

**نکته :** برای آشنایی بیشتر با نحوه استفاده این تگ بر روی صفحات وب ، به بخش آموزش قرار دادن فایل های تصویری در HTML 5 بروید .  
شکل کلی استفاده از این تگ به صورت زیر است :

syntax	<video src="آدرس فایل صوتی">
--------	------------------------------

### انواع خاصیت های تگ < video > :

در لیست زیر ، خواص تگ < video > و نحوه استفاده از آنها را تشریح کرده ایم :

< video >	
توضیح	خاصیت
این خاصیت یک مقدار Boolean است و در صورتی که برای یک تگ <video> تعیین شود، باعث می شود تا فایل تصویری به محض آماده شدن و لود کامل صفحه ، شروع به اجرا نماید . تعیین این پارامتر ، اختیاری است .	autoplay
این خاصیت یک مقدار Boolean است و در صورتی که برای یک تگ <video> تعیین شود، باعث می شود تا کنترل های پخش ... برای فایل تصویری بر روی صفحه نمایش داده شود . تعیین این پارامتر ، اختیاری است .	controls
این پارامتر مکان یا آدرس فایل تصویری مورد نظر برای اجرا ، را تعیین می کند . تعیین این پارامتر ، اجباری است .	src
این خاصیت یک مقدار Boolean است و در صورتی که برای یک تگ <video> تعیین شود، اجرای مجدد فایل تصویری ، پس از پایان یافتن آن می شود . تعیین این پارامتر ، اختیاری است .	loop
این خاصیت یک مقدار Boolean است و در صورتی که برای یک تگ <video> تعیین شود،	muted

می شود تا فایل تصویری در ابتدای اجرا ، بدون صدا باشد. تعیین این پارامتر ، اختیاری است .	
این خاصیت ی، آدرس URL یک عکس را تعیین می کند ، تا در زمانی که ویدیو در حال دانلود شدن بوده و یا کاربر هنوز دکمه play را نزده ، به جای ویدیو نمایش داده شود . تعیین این پارامتر ، اختیاری است .	poster

### پشتیبانی در مرورگر های مختلف:



این تگ در تمام مرورگرهای اصلی پشتیبانی می شود .

### مثال عملی:

: در مثال زیر به وسیله تگ < video > ، یک فایل تصویری را بر روی صفحه قرار داده و اجرا کرده ایم :  
**نکته :** < source > ، می توان برای تعیین فایل های تصویری مختلف برای تگ < video > استفاده کرد .

Example	
<p>- به صورت مستقیم</p> <pre>&lt;video controls autoplay src="movie.ogg"&gt;   Your browser does not support the video tag. &lt;/video&gt;</pre> <p>- تعیین فایل در یک تگ &lt; source &gt;</p> <pre>&lt;video controls="controls"&gt;   &lt;source src="movie.mp4" type="video/mp3" /&gt;   &lt;source src="movie.ogg" type="video/ogg" /&gt;   Your browser does not support the video tag. &lt;/video&gt;</pre>	کد

## < source > < HTML 5 اجرای مدیا در HTML 5

### < source > :

از این تگ می توان برای تعیین یک یا چندین فایل به عنوان منبع جهت اجرا در تگ های صوتی و تصویری <Audio> <Video> استفاده کرد . از این تگ معمولا برای تعیین فرمت های مختلف یک موزیک یا فیلم برای تگ های <Audio> <Video> استفاده می شود ، تا هر مرورگر فرمتی از فایل را که پشتیبانی می کند اجرا نماید .

برای مثال سه نوع فرمت avi , swf , wmv را برای یک ویدیو واحد تعیین می کنید. سپس هر مرورگر نسخه یا فرمتی که می تواند اجرا کند را نمایش می دهد .

شکل کلی استفاده از این تگ به صورت زیر است :

syntax	<audio   video >
	<source 1 src = " نوع فایل " type=" آدرس فایل "
	<source 2 src = " نوع فایل " type=" آدرس فایل "
	....
	< /audio   / video >

### انواع خاصیت های تگ < audio > :

در لیست زیر ، خواص تگ < audio > نحوه استفاده از آنها را تشریح کرده ایم :

< audio >	
توضیح	خاصیت
این پارامتر مکان یا آدرس فایل صوتی یا تصویری مورد نظر برای اجرا ، را تعیین می کند . تعیین می کند . تعیین این پارامتر ، اجباری است .	src
این خاصیت نوع و فرمت فایل صوتی یا تصویری را تعیین می کند . برای مثال MP3 یا SWF . تعیین این پارامتر ، اختیاری است .	MIME Type

پشتیبانی در مرورگر های مختلف:



این تگ در تمام مرورگرهای اصلی پشتیبانی می شود .

### مثال عملی:

: در مثال زیر به وسیله تگ < audio > ، یک فایل صوتی را بر روی صفحه قرار داده و اجرا کرده ایم :

**نکته :** < source > ، نیز برای تعیین منبع چند فایل صوتی مختلف برای تگ < audio >

Example	
<pre>&lt;audio controls="controls"&gt;   &lt;source src="song.mp3" type="audio/mp3" /&gt;   &lt;source src="song.ogg" type="audio/ogg" /&gt;   Your browser does not support the audio tag. &lt;/audio&gt;</pre>	کد
	خروجی

## drop & HTML 5 - drag

: drag & drop

یکی از قابلیت های هیجان انگیز و جالب HTML 5 ، این است که شما می توانید کاری کنید عنصر در صفحه وب ، قابل جابه جا شدن باشد . به عبارت دیگر کاربر بتواند آن را از نقطه ای به نقطه دیگر drag & drop کند .

برای مثال شما می توانید کدی طراحی کنید تا کاربر بتواند یک عکس را از مکان اولیه آن به یک مکان دوم که تعیین نموده اید ، به وسیله موس بلند کرده و جابه جا کند .

این کار نیازمند کد نویسی و تعریف چند تابع است . برای اینکه درک بهتری از نحوه کار داشته باشید ، ابتدا نمونه کد لازم برای انجام این عملیات را به شما نشان می دهیم . سپس مرحله به مرحله اقدام به تشریح کد می کنیم . کد لازم برای dargable ( قابل جابه جا بودن ) یک عکس و سپس انتقال آن در جدول زیر به شما نمایش داده شده . به کد دقت کنید . در ادامه هر مرحله از کد را توضیح داده ایم :

نمونه کد	<pre>&lt;head&gt; &lt;script type="text/javascript"&gt; function allowDrop(ev) {     ev.preventDefault(); } function drag(ev) {     ev.dataTransfer.setData("Text",ev.target.id); } function drop(ev) {     var data=ev.dataTransfer.getData("Text");     ev.target.appendChild(document.getElementById(data));     ev.preventDefault(); } &lt;/script&gt; &lt;/head&gt; &lt;body&gt; &lt;div id="div1" ondrop="drop(event)" ondragover="allowDrop(event)"&gt; &lt;/div&gt; &lt;img id="drag1" src="img_logo.gif" draggable="true" ondragstart="drag(event)"</pre>
----------	--



	<pre>width="336" height="69" /&gt; &lt;/body&gt;</pre>
--	--

ممکن است کد بالا در نگاه اول بسیار پیچیده باشد ، ولی در ادامه آن را جز به جز شرح داده ایم ، تا متوجه نحوه کار شوید .

### مرحله اول - عنصر مورد نظر را قابل جا به جا کردن کنید:

در مرحله اول باید کاری کنید تا عنصر مورد نظرتان برای drop & drag کردن ، قابل جا به جا شدن باشد . برای این منظور بایر مقدار خاصیت draggable آن را بر روی مقدار true تنظیم نمایید .

کد لازم	<pre>&lt; img draggable = " true " /&gt;</pre>
---------	--

### مرحله دوم - تعیین عنصری که می خواهید جا به جا شود:

در مرحله دوم باید عنصری را که می خواهید جا به جا شود و همچنین اتفاقی که پس از drag کردن آن قرار است بیفتد را تعیین می کنید . کد مربوط به این قسمت ، قطعه کد زیر است :

کد لازم	<pre>function drag ( ev ) { ev.dataTransfer.setData ( " Text " , ev.target.id ) ; }</pre>
---------	---

در کد بالا ، خاصیت ondragstart عنصر عکس تگ ( < img > ) ، رویداد drag ( event ) را فراخوانی می کند . این رویداد تعیین می کند چه عنصری می خواهد drag . این رویداد زمانی اتفاق می افتد که شما دکمه موس را بر روی عکس کلیک کرده و سعی می کنید آن را بکشید . ( ) dataTransfer.setData اطلاعات عنصری که قرار است drag شود ، یعنی id آن را دریافت کرده و در حافظه مرورگر قرار می دهد . این id " text " بوده و مقدار آن در این مثال drag1 .

### مرحله سوم - darg شده در کجا drop :


پس از اینکه عنصر مورد نظر را انتخاب کرده و آن را drag کردیم ، باید مکانی که فرود ( drop ) ی آید را تعیین کنیم . این کار توسط کد زیر انجام می شود :

کد لازم	<pre>function drop(ev) { var data = ev.dataTransfer.getData( " Text " ); ev.target.appendChild ( document.getElementById ( data ) ); ev.preventDefault( ); }</pre>
---------	--

هنگامی که عکس را در مکان تعیین شده رها می کنید ، رویداد `ondrop` `< div >` `drop` فراخوانی می کند .. `drop` `ev.dataTransfer.getData ( )` شناسه یا همان `id` شده را دریافت کرده و با متد `ev.target.appendChild` آن را به `div` مرتبط می سازد .

### مشاهده خروجی کد مثال:

خروجی اکتیو کد بالا در جدول زیر نمایش داده شده است عکس لوگوی HTML 5 `drop & drag` کرده و در `div` تعیین شده ، رها کنید :

Example	
	خروجی

# HTML 5 - ذخیره سازی اطلاعات کاربران در صفحات وب

## Cookie & Web Storage

### ندمه ذخیره سازی اطلاعات:

یکی از قابلیت ها و ویژگی های جدید که به HTML 5 ، اضافه شده است ، امکان ذخیره و بازیابی اطلاعات کاربران در صفحات وب توسط خاصیت web storage .

برای مثال شما هنگامی که وارد یک سایت شده و نام کاربری و رمز عبور خود را وارد می کنید ، مرورگر این اطلاعات را به خاطر می آورد . در بازدید از صفحات دیگر همان سایت دیگر لازم نیست ، مجددا اطلاعات کاربری خود را وارد نمایید . یا مثلا فردا که به آن سایت سر می زنید ، متوجه می شوید از قبل در سایت log in شده اید .

با این امور آشنا هستید . تکرار آنها برای تشریح بیشتر مسئله است . این کارها در HTML 4 و یا در PHP توسط کوکی ها cookies انجام می شوند . HTML 5 به جای کوکی ها ، قابلیت درون ساخته ایی اضافه شده است ، که کار آنها را انجام می دهد .

مسئله با مفهوم کوکی آشنا هستید ، اما برای یادآوری به توضیح مجدد cookie و کار آن می پردازیم .

### کوکی چیست ؟

کوکی یک قطعه اطلاعات است که سرور بر روی کامپیوتر کاربر ذخیره کرده و در مراجعات کاربر به همان سایت ، از آن اطلاعات برای شناسایی وی استفاده می کند . این اطلاعات معمولا راجع به شناسه کاربری ، رمز عبور ، تنظیمات یک کاربر بر روی سایت و ... می باشد . زمانی که کاربر به وسیله کامپیوتر خود همان سایت را باز می کند ، مرورگر اطلاعات کوکی ها را به سرور ارسال می کند . از کوکی برای بازیابی اطلاعات کاربری و یا سایر تنظیماتی که کاربر در یک سایت ایجاد کرده است ، در مراجعات بعدی به همان سایت استفاده می شود .

---

### web storage چیست ؟

web storage HTML 5 ، اطلاعات کاربران می تواند به صورت محلی در مرورگر و کامپیوتر کاربر ذخیره شود .

web storage از کوکی ها مطمئن تر و سریعتر بوده و حجم بالاتری از اطلاعات را می توان به وسیله آن ذخیره نمود . اطلاعات ذخیره web storage فقط در زمان هایی که به آنها نیاز است ، ارسال می شوند ، برخلاف کوکی ها که با هر بار درخواست ، آنها نیز ارسال می شوند . اطلاعات کاربران به وسیله این خاصیت ، توسط جفت های نام و مقدار ذخیره می شوند . این جفت شامل یک نام منحصر به فرد و یک مقدار دلخواه است . در ادامه به تشریح ، نحوه استفاده از این قابلیت می پردازیم ، اما ابتدا به بررسی سازگاری آن با مرورگرهای مطرح می پردازیم .

---

### نتیاتی مرورگرهای مختلف از خاصیت web storage :



این خاصیت در تمام نسخه های جدید مرورگرهای مطرح پشتیبانی می شود . به وسیله کد زیر نیز می توان پشتیبانی یا عدم پشتیبانی مرورگر مورد نظر خود از این خاصیت را بررسی کرد :

خروجی این کد نیز نمایش داده شده است . به وسیله خروجی می توانید پشتیبانی مرورگرگان را از این خاصیت مشاهده نمایید .

Syntax	<pre> &lt; script type="text/javascript" &gt; if ( typeof ( Storage ) !== " undefined " ) {     document.write ( " Yes! localStorage and sessionStorage support! " ) ; } else {     document.write ( " Sorry! No web storage support.. " ) ; } &lt; /script &gt; </pre>
خروجی	Yes! localStorage and sessionStorage support!

## واع حالت ها و اشیای شی : web storage

دو نوع شی و حالت برای نگهداری و ذخیره اطلاعات کاربر توسط شی web storage . در لیست زیر به معرفی این دو حالت پرداخته ایم . برای دریافت اطلاعات بیشتر راجع به هر کدام بر روی نام آن کلیک نمایید :

- **شی : localStorage** این شی برای نگهداری اطلاعات به مدت طولانی استفاده می شود . اطلاعات ذخیره شده توسط این شی بر روی کامپیوتر کاربر قرار گرفته و با بسته شدن مرورگر و یا خاموش کردن کامپیوتر از بین نمی روند و در آینده قابل بازیابی هستند.
- **شی : sessionStorage** این شی برای نگهداری اطلاعات به صورت موقت به کار رفته و در حافظه موقت کامپیوتر کاربر ذخیره می شود . با بسته شدن صفحه جاری در مرورگر و خاموش شدن کامپیوتر ، این اطلاعات نیز از بین رفته و دیگر قابل بازیابی نیستند .

## HTML 5 < ذخیره سازی اطلاعات کاربران در صفحات وب > شی localStorage

### شی localStorage :

در بخش قبل ، به معرفی خاصیت web storage HTML 5 پرداختیم . در آنجا گفتیم که این خاصیت جایگزین کوکی ها در صفحات وب شده و برای ذخیره اطلاعات کاربران استفاده می شود . این اطلاعات می تواند شامل نام کاربری ، تاریخ آخرین مراجعه به صفحه و یا اطلاعاتی از این دست باشد . خاصیت web storage شی برای ذخیره اطلاعات استفاده می کند . شی اول ، شی localStorage است ، که در ادامه به معرفی آن می پردازیم .

از شی localStorage برای نگهداری اطلاعات کاربران به مدت طولانی و بدون تاریخ انقضا استفاده می شود . این اطلاعات در مرورگر کاربر ذخیره شده و با بستن مرورگر و خاموش کردن کامپیوتر از بین نرفته و مجددا می توانند مورد بازیابی و استفاده قرار گیرند . در نتیجه از این شی برای ذخیره اطلاعاتی استفاده می شود که در آینده به دفعات نیاز به آنها خواهد بود ( و نه اطلاعات مقطعی ) .

شکل کلی استفاده از شی localStorage بصورت زیر است :

syntax	localStorage.name = " value " ;
--------	---------------------------------

توضیح موارد syntax	
توضیح	
این پارامتر تعیین کننده نام شی localStorage بوده و در سطح برنامه و برای دسترسی به آن از این نام استفاده می شود . تعیین این پارامتر اجباری است .	name
این پارامتر نیز تعیین کننده مقدار مورد نظر برای شی localStorage . این مقدار ، واقع مقدار کوکی مورد نظرمان است . تعیین این پارامتر اجباری است .	value

### آموزش نحوه کار با شی localStorage در قالب مثال های عملی :

: در مثال زیر یک شی ساده localStorage را به نام name ایجاد و مقدار دهی کرده ایم .  
در سطح صفحه استفاده نموده ایم . به کد مثال دقت نمایید . سپس به تشریح ، خط به خط آن پرداخته ایم :

Example	
<pre> &lt;div id="result"&gt; &lt;/div&gt;  &lt;script&gt;   if (typeof ( Storage ) !== " undefined " ) {     localStorage.name = " Ali ";     document.getElementById("result").innerHTML = "Name: " + localStorage.name;   }   else {     document.getElementById("result").innerHTML = "Sorry, your browser does not support web storage...";   } &lt;/script&gt; </pre>	کد
Name: Ali	خروجی

- در خط قرمز به بررسی این پرداخته ایم که آیا مرورگر از شی `localStorage` پشتیبانی می کند یا خیر و کدهای لازم برای هر حالت را تعیین نموده ایم.
- در خط سبز یک شی `localStorage` به نام `name` را ایجاد و مقدار `" Ali "` را به آن نسبت داده ایم.
- سپس در خط آبی مقدار شی `localStorage` را خوانده و آن را در یک عنصر `Html` به `result` ریخته و نمایش داده ایم.
- در نهایت در خط آخر ، که قهوه ای است ، در صورتی که مرورگر کاربر از شی `localStorage` پشتیبانی نکند ، یک پیام هشدار به وی نشان داده شده است.

## HTML 5 < ذخیره سازی اطلاعات کاربران در صفحات وب > شی sessionStorage

### شی sessionStorage :

این شی همانند شی localStorage می تواند اطلاعات کاربر را روی کامپیوتر و مرورگر وی ذخیره نماید ، با این تفاوت که شی sessionStorage اطلاعات را بصورت موقتی ذخیره کرده و با بسته شدن مرورگر تمام این اطلاعات از بین خواهد رفت .  
بنابراین از این شی بایستی برای ذخیره سازی اطلاعات مقطعی و اطلاعاتی که در آینده نیاز به آن نداریم استفاده کرد .

شکل کلی استفاده از شی sessionStorage بصورت زیر است :

syntax	sessionStorage.name =" value " ;
--------	----------------------------------

توضیح موارد syntax	
توضیح	
این پارامتر تعیین کننده نام شی sessionStorage بوده و در سطح برنامه و برای دسترسی به آن از این نام استفاده می شود . تعیین این پارامتر اجباری است .	name
این پارامتر نیز تعیین کننده مقدار مورد نظر برای شی sessionStorage . این مقدار ، واقع مقدار کوکی مورد نظرمان است . تعیین این پارامتر اجباری است .	value

### آموزش نحوه کار با شی sessionStorage در قالب مثال های عملی :

: در مثال زیر یک شی sessionStorage را به نام clickcount ایجاد کرده ایم . این شی تعداد کلیک کاربر بر روی دکمه فرمان مثال را شمرده و آن را در کوکی clickcount ذخیره کرده و در خروجی نمایش می دهد . با بسته شدن مرورگر ، اطلاعات این شی از بین می رود و ریست می شود :

Example	
<pre>&lt;script type="text/javascript"&gt; function clickCounter() {     if (typeof (Storage) !== "undefined") {         if (sessionStorage.clickcount) {</pre>	کد کد

<pre>         sessionStorage.clickcount = Number(sessionStorage.clickcount) + 1;     }     else {         sessionStorage.clickcount = 1;     }     document.getElementById("result").innerHTML = " + sessionStorage.clickcount + " دکمه فرمان را کلیک کرده اید "     }     else {         document.getElementById("result").innerHTML = " "شما از این شی پشتیبانی نمی کند     } } &lt;/script&gt;  &lt;button onclick="clickCounter()" type="button"&gt; Click me! &lt;/button&gt; &lt;div id="result"&gt; &lt;/div&gt; &lt;p&gt; بر روی دکمه فرمان کلیک نمایید ، تا شمارنده بالا رود &lt;/p&gt; &lt;p&gt; سپس صفحه را بسته و مجددا باز کنید . خواهید دید که کوکی ذخیره شده ، از بین رفته و &lt;/p&gt; ریست شده است &lt;/p&gt; </pre>	
<p>Click me!</p> <p>بر روی دکمه فرمان کلیک نمایید ، تا شمارنده بالا رود</p> <p>سپس صفحه را بسته و مجددا باز کنید . خواهید دید که کوکی ذخیره شده ، از بین رفته و ریست شده</p>	<p>خروجی</p>

- ابتدا در خط قرمز به بررسی این پرداخته ایم که آیا مرورگر از شی پشتیبانی می کند یا خیر و کدهای لازم برای هر حالت را تعیین نموده ایم.
- در خط سبز یک شی به نام `clickcount` را ایجاد کرده و با هر بار کلیک بر روی دکمه فرمان یک واحد مقدار آن اضافه می شود .
- سپس در خط آبی مقدار شی `localStorage` را خوانده و آن را در یک عنصر `Html` به نام `result` ریخته و نمایش داده ایم.
- در نهایت در خط آخر ، که نارنجی است ، در صورتی که مرورگر کاربر از شی `localStorage` پشتیبانی نکند ، یک پیام هشدار به وی نشان داده شده است.



## • HTML 5 < اجرای اسکریپت ها در پس زمینه < web

### workers

#### • web workers :

• در زمان اجرای برخی اسکریپت ها در صفحات ، ممکن است صفحه در اصطلاح هنگ کرده و به کلیک ها و درخواست های کاربر تا زمانی که اجرای آن اسکریپت تمام شود ، پاسخ ندهد . این مسئله بیشتر اوقات در دسر ساز شده و حتی باعث کرک شدن . خروج اجباری مرورگر از صفحه وبسته شدن آن می شود . HTML5 برای حل این مشکل راه حلی به نام متد Web Worker پیشبینی شده است .

• Web Worker باعث می شود تا اسکریپت ها بدون توجه به سایر اسکریپت های صفحه در پشت پرده اجرا شده و بر روی کارایی و عملکرد صفحه تاثیری نداشته باشند . با استفاده از این متد می توان در حین اجرای اسکریپت ها ، به صورت عادی به کار با صفحه پرداخته و به کلیک کردن و انجام سایر امور مورد نظر خود پردازید .

این بخش نحوه استفاده از این تکنیک جدید را در عمل نمایش خواهیم داد .

#### • مرحله اول : چک کردن اینکه مرورگر از Web Worker پشتیبانی می کند یا خیر ؟

• در مرحله اول باید چک کرد که آیا مرورگر ما از متد Web Worker پشتیبانی می کند یا خیر ، تا متناسب راه هر حالت ، کد لازم را تعیین کنیم .

• کد زیر بررسی می کند آیا مرورگر در حال استفاده از متد web worker پشتیبانی می کند یا خیر . هر حالت کد مناسب آن اجرا شود :

چک کردن پشتیبانی	<pre>if ( typeof (Worker) !== " undefined " ) {     // مرورگر پشتیبانی می کند     // کد های مورد نظر برای زمان پشتیبانی } else {     // متاسفانه مرورگر پشتیبانی نمی کند     // کد های مورد نظر برای عدم پشتیبانی }</pre>
---------------------	---

#### • مرحله دوم : یک فایل یک فایل Web Worker بسازیم .

• Web Worker کد های جاوا اسکریپت مورد نظرم را در یک فایل خارجی و در بیرون از صفحه قرار می دهیم .

بنابراین در مرحله دوم اسکریپت مورد نظرم که در این مثال بصورت زیر است ، را در فایل خارجی

Example.js قرار می دهیم . کد زیر یک کد ساده جاوا اسکریپت می باشد که عمل شمارش را

انجام می دهد و فقط به عنوان یک مثال مطرح شده است :

- مهمترین نکته در کد زیر متد ( ) Post Message است ، که مقدار خروجی تابع را بصورت یک پیام به صفحه ارسال می کند .

کد اسکریپت	<pre> var i=0;  function timedCount( ) {     i=i+1;     <b>postMessage ( i ) ;</b>     setTimeout("timedCount()",500); }  timedCount(); </pre>
------------	--

- **مرحله سوم : ایجاد یک شی Web Worker .**
- اکنون که فایل خارجی اسکریپت را در اختیار داریم باید به وسیله ای آن را از صفحه HTML فراخوانی کنیم . این کار بوسیله یک شی Web Worker انجام می شود . کد زیر ابتدا چک می کند که آیا یک شی Web Worker وجود دارد یا خیر . سپس در صورتی که شی وجود نداشته باشد ، یک نمونه جدید ساخته و فایل Example.js را اجرا می کند . برنامه کد موجود در فایل را اجرا کرده و جواب (Message) را به صفحه برمی گرداند .

چک کردن شی web worker	<pre> if ( typeof (w) == "undefined" ) {     w=new Worker("Example.js"); } </pre>
-----------------------	---

- سپس بوسیله کد زیر می توان جواب برگشتی از فایل جاوا اسکریپت را در صفحه اعمال کرد . هنگامی که Web Worker برمی گردد تابع موجود در متد Event . اطلاعات برگشتی از Web event.data نگهداری می شود .

کد	<pre> w.onmessage=function(event){     document.getElementById("result").innerHTML=event.data; }; </pre>
----	--

- **مرحله آخر : توقف اجرای یک شی Web Worker .**

- هنگامی که یک شی Web Worker ایجاد می شود این شی مرتب به پاسخ ارسالی از طرف اسکریپت ها توجه می کند ( حتی پس از اتمام فایل جاوا اسکریپت ) . به وسیله متد ( terminate ) ، اجرای آن شی را متوقف می کنیم :

متوقف کردن اجرای شی web worker	w.terminate( ) ;
--------------------------------------	------------------

- **مثال عملی :**
- در مثال زیر ، کد کامل و خروجی کار با شی web worker را نمایش داده ایم :

Example	
<pre> &lt;!DOCTYPE html&gt; &lt;html&gt; &lt;body&gt;   &lt;p&gt; Count numbers : &lt;output id="result" &gt; &lt;/output&gt; &lt;/p&gt;   &lt;button onclick="startWorker( )" &gt; Start Worker &lt;/button&gt;   &lt;button onclick="stopWorker( )" &gt; Stop Worker &lt;/button&gt;   &lt;script&gt;     var w;     function startWorker( ) {       if (typeof (Worker) !== "undefined") {         if (typeof (w) == "undefined") {           w = new Worker("Example.js");         }         w.onmessage = function (event) {           document.getElementById("result").innerHTML = event.data;         };       }       else {         document.getElementById("result").innerHTML = "Sorry, your browser does not support Web Workers...";       }     }     function stopWorker( ) { </pre>	کد

<pre>w.terminate( );      }     &lt;/script&gt; &lt;/body&gt; &lt;/html&gt;</pre>	
Count numbers :  Start Worker Stop Worker	خروجی

## SSE < HTML 5 دریافت آپدیت های ارسالی از سرور <

### SSE ( Server-Sent Events ) - رویدادهای طرف سرور :

SSE قابلیت جدیدی در HTML5 است که به مرورگر این امکان را می دهد تا آپدیت های ارسالی از طرف سرور را دریافت کند . این قابلیت در گذشته بصورت دیگری وجود داشته است . هر زمان که آپدیتی ایجاد می شد مرورگر از کاربر برای دریافت و اعمال آپدیت ارسالی سوال می کرد . SSE این کار بصورت اتوماتیک و پشت پرده صورت می گیرد .

ال هایی برای این آپدیت عبارتند از : تغییر عناوین خبرها ، نتایج ورزشی ، تغییر در شاخص ها و نمودارهای سایت های اقتصادی و قیمت ... .

### پشتیبانی در مرورگر های مختلف:



این خاصیت در تمام مرورگرهای اصلی به جزء IE پشتیبانی می شود .

### مرحله اول : مرحله اول : چک کردن اینکه مرورگر از SSE پشتیبانی می کند یا خیر ؟

بهتر است اینکه آیا مرورگر کاربر از متد SSE پشتیبانی می کند یا خیر را ، ابتدا بررسی کنیم تا در هر حالت کد خود را بر مبنای آن بنویسیم . قطعه کد زیر بررسی می کند که آیا مرورگر از متد SSE پشتیبانی می کند یا خیر کد لازم را در هر صورت اجرا می نماید .

چک کردن پشتیبانی	<pre>if ( typeof ( EventSource ) !== " undefined " ) {     // پشتیبانی مرورگر از SSE     // کدهای مورد نظر } else {     // عدم پشتیبانی مرورگر از SSE     // کدهای مورد نظر }</pre>
---------------------	---

### مرحله دوم : یافت آپدیت های ارسالی از سرور :

شی EventSource برای دریافت اطلاعات در آپدیت های ارسالی از طرف سرور به کار می رود . شکل کلی استفاده از این شی بصورت زیر است .

چک کردن پشتیبانی	<pre>var source = new EventSource ( "demo_sse.php" ) ; source.onmessage=function(event) {     document.getElementById("result").innerHTML+=event.data + "&lt;br/&gt;"; };</pre>
------------------	---

### توضیح کد بالا :

- در خط اول یک شی جدید EventSource source را ایجاد کرده و صفحه demo\_sse.php را به URL یا آدرس فایلی که آپدیت را به صفحه ارسال می کند تعیین کرده ایم .
  - هر زمان که آپدیت جدیدی دریافت شود رویداد On Message اجرا می شود .
  - هر زمان که رویداد onmessage اجرا می شود نتیجه خروجی در عنصر با id Result آپدیت و به روز رسانی می شود .
- نکته :** آپدیت ارسالی در شی event.data .

### یک فاز مهم : کد لازم برای تولید آپدیت از سرور :

برای دریافت آپدیت باید صفحه ای وجود داشته باشد که قابلیت تولید و ارسال آپدیت را داشته باشد . برای مثال یک صفحه PHP یا ASP .

کد سرور باید برای ارسال آپدیت بسیار ساده است . کافی است در صفحه PHP یا ASP ، مقدار خاصیت Content-Type صفحه را بر روی مقدار text/event-stream قرار دهید . در جدول های زیر کد لازم را در هر دو زبان نمایش داده ایم :

کد لازم PHP	<pre>&lt;?php     header('Content-Type: text/event-stream');     header('Cache-Control: no-cache');     \$time = date( ' r ' );     echo "data: The server time is: {\$time}\n\n";     flush( ); ?&gt;</pre>
-------------	--

کد لازم ASP	<pre>&lt;%     Response.ContentType="text/event-stream"     Response.Expires=-1</pre>
-------------	---

	<pre>Response.Write("data: " &amp; now()) Response.Flush() %&gt;</pre>
--	--

#### موارد مختلف کدهای بالا :

- مشخص نمایید که صفحه نباید اطلاعات را ( Cashe ذخیره ) نماید.
- داده ها را برای ارسال آماده نمایید ( همواره با کلمه ( data شروع می شود.
- خروجی و نتیجه را به صفحه درخواست کننده ارسال نمایید ( Flush ).

## < canvas > HTML 5 < طراحی اشیای گرافیکی در HTML 5 >

### < canvas > :

< canvas > HTML 5 برای طراحی اشیا و اشکال گرافیکی بصورت Run Time به کمک زبان های اسکریپتی مثل جاوا اسکریپت استفاده می شود . بوسیله تگ < canvas > شما می توانید اشیای گرافیکی مثل مربع ، دایره ، تصاویر و ... را بر روی صفحه ایجاد نمایید . < canvas > خود به تنهایی خروجی بر روی صفحه نداشته و فقط به عنوان یک نگهدارنده برای اشیای گرافیکی استفاده می شود . عمل طراحی اشکال بوسیله جاوا اسکریپت انجام می شود .

شکل کلی < canvas > بصورت زیر است :

syntax	<canvas id = " id " > </canvas>
--------	---------------------------------

توضیح syntax	
توضیح	Parameter
این پارامتر تعیین کننده شناسه یا id تگ در صفحه است . حتما این پارامتر را برای تگ < canvas > تعیین نمایید ، زیرا به وسیله آن می توان در اسکریپت ها و سایر صفحه ، به تگ مورد نظرمان دسترسی داشته باشید .	id

### پشتیبانی در مرورگر های مختلف:



این تگ در تمام مرورگرهای اصلی پشتیبانی می شود .

**نکته :** < canvas > پشتیبانی نکند ، متن ( Text ) که درون آن وجود دارد ، به جای شکل گرافیکی نمایش داده می شود .  
**نکته :** < canvas > HTML 5 جدید است .

### < canvas > :

در لیست زیر ، خواص تگ < canvas > :



<code>&lt; canvas &gt;</code>	
توضیح	خاصیت
این خاصیت ارتفاع شکل هندسی را تعیین می کند .	<code>height</code>
این خاصیت عرض شکل هندسی را تعیین می کند .	<code>width</code>

### مثال عملی:

: در مثال زیر یک تگ `< canvas id = MyCanvas` ، تعریف کرده ایم . سپس به وسیله یک اسکریپت آن تگ را فراخوانی کرده و با استفاده از متدهای خاصی ، شکلی را برای آن طراحی کرده ایم . برای آشنایی با نحوه طراحی شکل به کمک اسکریپت ها ، به نحوه کار با تگ `< canvas >` بروید :

Example	
<pre> &lt;canvas id="MyCanvas" height="100px" width="100px"&gt;     اگر مرورگر از این تگ پشتیبانی نکند ، این متن نمایش داده می شود &lt;/canvas&gt;  &lt;script type="text/javascript"&gt;     var canvas = document.getElementById('MyCanvas');     var ctx = canvas.getContext('2d');     ctx.fillStyle = '#FF0000';     ctx.fillRect(0, 0, 80, 100); &lt;/script&gt; </pre>	کد

## HTML 5 < طراحی اشیای گرافیکی HTML 5 < نحوه کار با شی

### canvas

#### نحوه کار با شی canvas :

همانطور که در بخش معرفی تگ < canvas > گفتیم ، از این تگ می توانیم برای طراحی اشکال هندسی و گرافیکی بر روی صفحات وب استفاده نماییم .

< canvas > خود به تنهایی خروجی بر روی صفحه نداشته و باید بوسیله یک اسکریپت شکل مورد نظر را طراحی نماییم . شی canvas دارای چندین متد و خاصیت برای طراحی خطوط ، اشکال ، کاراکتر ها و اضافه کردن عکس ها به صفحه است . در این بخش به معرفی و آموزش نحوه کار با خواص و متدهای این شی می پردازیم .

#### ایجاد یک شی canvas :

شی canvas بوسیله یک تگ canvas بر روی صفحه ایجاد شده و به طور پیش فرض تگ < canvas > هیچ خط حاشیه یا محتویاتی ندارد . شکل کلی تعریف یک تگ < canvas > بصورت زیر است :

syntax	<code>&lt;canvas id = " id " width="200" height="100"&gt; &lt;/canvas&gt;</code>
--------	--

**نکته :** همواره خاصیت id را برای تگ < canvas > تعیین نمایید تا بتوانید به راحتی در کد های اسکریپت به آن دسترسی داشته باشید .

**نکته :** شما می توانید چندین تگ < canvas > را بر روی یک صفحه تعیین نمایید .

**نکته :** برای اضافه کردن خطوط حاشیه به عنصر از خاصیت Style استفاده نمایید .

	<code>&lt; canvas id="myCanvas" width="200" height="100" style="border: 1px solid blue" &gt; &lt;/canvas&gt;</code>
خروجی	

#### کشیدن یک شی canvas به وسیله جاوا اسکریپت :

در مرحله قبل گفتیم که شی canvas فقط به عنوان یک نگهدارنده شکل هندسی بوده و خروجی بر روی صفحه . عملیات طراحی و تعیین شکل مورد نظر را باید بوسیله یک اسکریپت ( معمولا به زبان جاوا اسکریپت ) انجام می دهیم . در این بخش با ارایه یک مثال عملی و تشریح مرحله به مرحله کد آن قصد داریم ، شما را با نحوه طراحی شکل آشنا نماییم :

. در ابتدا باید تگ canvas مورد نظر را بر روی صفحه پیدا کنیم .

کد	<code>var c = document.getElementById ( " MyCanvas " ) ;</code>
----	---

. سپس بوسیله متد canvas شی را فراخوانی و متد "d" را به آن ارسال کنیم canvas . یک شی پیش ساخته در HTML 5 دارای خواص و متدهای بسیاری برای طراحی اشیا می باشد .  
2 d به معنای دو بعدی بودن عنصر است .

کد	<code>var ctx = c.getContext ( " 2d " ) ;</code>
----	--

. در مرحله بعدی اقدام به تعیین رنگ پس زمینه و موقعیت و اندازه شکل می پردازیم :

کد	<code>ctx.fillStyle = " #FF0000 " ; ctx.fillRect( 0 , 0 , 150 , 75 ) ;</code>
----	---


. خاصیت fillStyle رنگ پس زمینه شکل را تعیین می کند . در این مثال رنگ قرمز برای مثال در نظر گرفته شده است .  
همچنین خاصیت fillRect موقعیت قرار گیری شکل در صفحه و اندازه آن را تعیین می کند .  
شکل کلی استفاده از این مند به صورت زیر است :

کد	<code>fillRect( x , y , width , height ) ;</code>
----	---

- o X : موقعیت قرار گیری عنصر در جهت افقی یا به عبارت دیگر فاصله آن از لبه سمت راستی عنصر مادر خود را تعیین می کند . ( x )
- o y : موقعیت قرار گیری عنصر در جهت عمودی یا به عبارت دیگر فاصله آن از لبه سمت لایی عنصر مادر خود را تعیین می کند . ( y )
- o width : میزان عرض عنصر را تعیین می کند.
- o height : میزان ارتفاع عنصر را تعیین می کند.

**خروجی نهایی :** در نهایت کد ما و خروجی آن به صورت زیر خواهد بود :

	<pre>&lt;canvas id="MyCanvas" width="200" height="100" style="border: 1px solid #c3c3c3;"&gt;     مرورگر شما از این تگ پشتیبانی نمی کند &lt;/canvas&gt;  &lt;script&gt;     var c = document.getElementById("myCanvas");</pre>
--	--

	<pre>var ctx = c.getContext("2d"); ctx.fillStyle = "#FF0000"; ctx.fillRect(0, 0, 150, 75); &lt;/script&gt;</pre>
خروجی	

### : کشیدن خطوط canvas به وسیله جاوا اسکریپت :

در مرحله قبل نحوه کشیدن یک شکل بعدی را بوسیله تگ <canvas> توضیح دادیم . در این مرحله می خواهیم نحوه کشیدن یک خط ساده در صفحه را نشان دهیم . برای این کار از متد های زیر استفاده می کنیم :

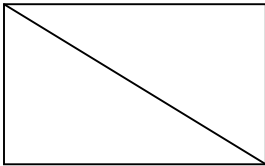
- **moveTo ( x , y )** : این متد مختصات نقطه شروع خط را تعیین می کند.
- **lineTo ( x , y )** : این متد مختصات نقطه پایان خط را تعیین می کند.

پس از تعیین مختصات نقطه شروع و پایان خط بوسیله متد ( ) **Stroke** اقدام به کشیدن خط نمایید .

**مثال کشیدن یک خط ساده :** در نهایت کد ما و خروجی آن به صورت زیر خواهد بود :

ابتدا ، شی **Canvas1** را بر روی صفحه فرا خوانده و در متغیر **c1** ذخیره می کنیم . سپس طرح دو بعدی شی **c1** را ایجاد کرده و در متغیر **ctx1** می ریزیم . در نهایت با تعیین مختصات نقطه شروع و پایان خط ، آن را در خروجی طراحی می کنیم .

	<pre>&lt;canvas id="Canvas1" width="200" height="100" style="border: 1px solid #c3c3c3;"&gt;     مرورگر شما از این تگ پشتیبانی نمی کند &lt;/canvas&gt;  &lt;script&gt;     var c1 = document.getElementById("Canvas1");     var ctx1 = c1.getContext("2d");</pre>
--	---

	<pre> ctx1.moveTo(0, 0); ctx1.lineTo(200, 100); ctx1.stroke(); &lt;/script&gt; </pre>
خروجی	

### گام چهارم – کشیدن یک دایره در صفحه وب :

برای کشیدن یک دایره بر روی صفحه ، از متد زیر استفاده می کنیم "

- **arc ( x , y , r , start ,stop )** : این متد مختصات نقطه شروع ، اندازه شعاع و نقطه پایان دایره را مشخص می کند.

پس از تعیین مختصات نقطه شروع و پایان خط بوسیله متد ( ) **Stroke** اقدام به کشیدن دایره نمایید .

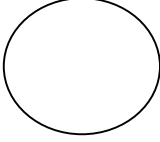
**مثال کشیدن یک دایره :** در نهایت کد ما و خروجی آن به صورت زیر خواهد بود :

ابتدا ، شی **Canvas2** را بر روی صفحه فرا خوانده و در متغیر **c2** ذخیره می کنیم .

سپس طرح دو بعدی شی **c2** را ایجاد کرده و در متغیر **ctx2** می ریزیم .

در نهایت با تعیین مختصات و اندازه ای شی دایره ، آن را در خروجی طراحی می کنیم .

	<pre> &lt;canvas id="Canvas2" width="200" height="100" style="border: 1px solid #c3c3c3;"&gt;     مرورگر شما از این تگ پشتیبانی نمی کند &lt;/canvas&gt;  &lt;script&gt;     var c2 = document.getElementById("Canvas2");     var ctx2 = c2.getContext("2d");     ctx2.beginPath( );     ctx2.arc( 95 , 50 , 40 , 0,2*Math.PI);     ctx2.stroke( ); &lt;/script&gt; </pre>
--	---

جی	
----	---


## HTML 5 < طراحی اشیای گرافیکی در HTML 5 < نحوه کار با شی canvas

### : دادن افکت به متن با شی canvas :

به وسیله شی canvas می توانید یک متن را در یک شکل گرافیکی قرار داده و یا جلوه های تصویری به آن دهید .  
برای این منظور از متدها و خواص زیر استفاده می شود :

- **خاصیت font :** این خاصیت ویژگی های مختلف متن مثل اندازه ، نوع قلم و ... را مشخص می کند .
- **fillText ( text , x , y ) :** این متد متن نوشته مورد نظر برای قرار دادن در شکل و مختصات محل قرار گیری آن را بر روی شکل تعیین می کند .  
: x این پارامتر ، فاصله متن از لبه سمت چپی شکل را تعیین می کند .  
: y این پارامتر ، فاصله متن از لبه سمت بالایی شکل را تعیین می کند .

: در مثال زیر به وسیله خاصیت فونت و متد fillText ( ) ، متن مورد نظرمان را در یک شکل مستطیل قرار داده ایم:

	<pre>&lt;canvas id="Canvas1" width="200" height="100" style="border: 1px solid #d3d3d3;"&gt;     مرورگر شما از این شی پشتیبانی نمی کند &lt;/canvas&gt;  &lt;script&gt;     var c1 = document.getElementById("Canvas1");     var ctx1 = c1.getContext("2d");     ctx1.font = "30px Arial";     ctx1.fillText("Developer1", 10, 50); &lt;/script&gt;</pre>
خروجی	

متد دیگری که می توان از آن برای قرار دادن یک متن در اشکال گرافیکی استفاده نمود ، متد strokeText ( ) . در این بخش قصد داریم تا نحوه کار با این متد را نمایش دهیم :

- **خاصیت font :** این خاصیت ویژگی های مختلف متن مثل اندازه ، نوع قلم و ... را مشخص می کند ،
- **strokeText ( text , x , y ) :** این متد متن نوشته مورد نظر برای قرار دادن در شکل و مختصات محل قرار گیری آن را بر روی شکل تعیین می کند . همچنین می توان یک جلوه تصویری را به متن داد .  
 : x این پارامتر ، فاصله متن از لبه سمت چپی شکل را تعیین می کند .  
 : y این پارامتر ، فاصله متن از لبه سمت بالایی شکل را تعیین می کند .

**:** در مثال زیر به وسیله خاصیت فونت و متد strokeText ( ) ، متن مورد نظرمان را در یک شکل مستطیل قرار داده ایم:

کد	<pre>&lt;canvas id="Canvas2" width="200" height="100" style="border: 1px solid #d3d3d3;"&gt;     مرورگر شما از این شی پشتیبانی نمی کند &lt;/canvas&gt;  &lt;script&gt;     var c2 = document.getElementById("Canvas2");     var ctx2 = c2.getContext("2d");     ctx2.font = "30px Arial";     ctx2.strokeText ("Developer1", 10, 50); &lt;/script&gt;</pre>
خروجی	

### **تعیین پس زمینه های رنگی برای شی canvas و اشکال هندسی :**

می توان به شی canvas و اشکال ایجاد شده به وسیله آن ، پس زمینه های رنگی با جلوه های خاص داد .

اشکالی که توسط شی canvas ، تولید می شوند ، محدود به استفاده از رنگ های پس زمینه ساده نیستند . به وسیله متدهای زیر ، می توانید پس زمینه های زیبا و با جلوه خاص ، برای شکل خود تعیین نمایید :

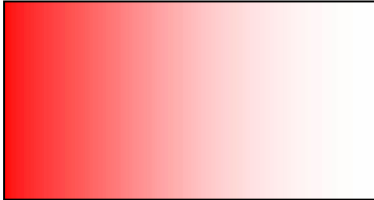
- **createLinearGradient ( x , y , x1 , y1 ) :** این متد ، یک پس زمینه با شیب خطی ایجاد می کند



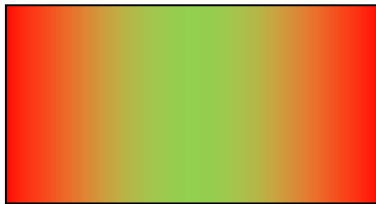
- x : این پارامتر ، فاصله متن از لبه سمت چپی شکل را تعیین می کند .
- y : این پارامتر ، فاصله متن از لبه سمت بالایی شکل را تعیین می کند.
- **createRadialGradient ( x , y , r , x1 , y1 , r1 )** : این متد ، یک پس زمینه با شیب محدب ایجاد می کند.

### توضیح بیشتر نحوه کار و مثال های عملی :

در استفاده از متدهای فوق ، می توانیم از یا چند رنگ ، برای قرار دادن در پس زمینه رنگی شکل استفاده کنیم . این رنگ ها را به وسیله تابع ( ) **addColorStop** ، اضافه می کنیم . رنگ ها به ترتیب توسط این تابع شماره گذاری شده و شماره آنها باید از ( اعشاری ) . سپس از متدهای ( ) **fillStyle** ، **fillRect** ( ) ، رنگ ها را در شی **canvas** قرار می دهیم . برای درک بهتر ، به مثال های عملی زیر دقت نمایید :

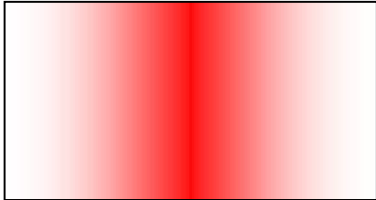
- تعیین یک پس زمینه خطی رنگی	
کد	<pre> &lt;canvas id="Canvas3" width="200" height="100" style="border: 1px solid #d3d3d3;"&gt;     مرورگر شما از این شی پشتیبانی نمی کند &lt;/canvas&gt;  &lt;script&gt;     var c3 = document.getElementById("Canvas3");     var ctx3 = c3.getContext("2d");     // Create gradient     var grd1=ctx3.createLinearGradient(0,0,200,0);     grd1.addColorStop(0,"red");      //     grd1.addColorStop(1,"white");    //      // Fill with gradient     ctx3.fillStyle=grd1;     ctx3.fillRect(10,10,150,80); &lt;/script&gt; </pre>
خروجی	

- تعیین یک پس زمینه خطی رنگی

کد	<pre>&lt;canvas id="Canvas4" width="200" height="100" style="border: 1px solid #d3d3d3;"&gt;     مرورگر شما از این شی پشتیبانی نمی کند &lt;/canvas&gt;  &lt;script&gt;     var c4 = document.getElementById("Canvas4");     var ctx4 = c4.getContext("2d");     // Create gradient     var grd2=ctx4.createLinearGradient(0,0,200,0);     grd2.addColorStop(0,"red");    //     grd2.addColorStop(0.5,"white"); //     grd2.addColorStop(1,"green");  //      // Fill with gradient     ctx4.fillStyle=grd2;     ctx4.fillRect(10,10,150,80); &lt;/script&gt;</pre>
خروجی	

- تعیین یک پس زمینه محدب رنگی

کد	<pre>&lt;canvas id="Canvas5" width="200" height="100" style="border: 1px solid #d3d3d3;"&gt;     مرورگر شما از این شی پشتیبانی نمی کند &lt;/canvas&gt;  &lt;script&gt;</pre>
----	--

	<pre>var c5 = document.getElementById("Canvas5"); var ctx5 = c5.getContext("2d"); // Create gradient var grd3=ctx5.createRadialGradient(90,50,5,90,60,100); grd3.addColorStop(0,"red"); grd3.addColorStop(1,"white");  // Fill with gradient ctx5.fillStyle=grd3; ctx5.fillRect(10,10,150,80); &lt;/script&gt;</pre>
خروجی	

## Application Cache < مرور آفلاین صفحات در HTML 5

### متد Application Cache :

Html5 قابلیت جدیدی را با نام Application Cache به کاربران وب ارائه داده که امکان ذخیره شدن صفحات مرور شده بر روی کامپیوتر کاربر و مشاهده آنها در حالت آفلاین را فراهم می کند . Application Cache این امکانات و مزیت ها را در اختیار کاربران قرار می دهد :

- **مرور آفلاین** : کاربران می توانند صفحات مورد نظر را حتی در زمان آفلاین بودند ، مشاهده نمایند.
- : صفحات ذخیره شده بر روی کامپیوتر کاربر با سرعت بیشتری لود و فراخوانی می شود.
- **کاهش حجم اطلاعات ارسالی از سرور** : باعث کاهش حجم اطلاعات ارسالی از سرور شده و مرورگر فقط صفحاتی که تغییر کرده و آپدیت شده اند ، را مجددا فراخوانی می کند.

### شتیانی در مرورگر های مختلف:



این متد در تمام مرورگرهای اصلی پشتیبانی می شود .

### مثالی از یک فایل با قابلیت ApplicationCache :

کد زیر یک صفحه وب با قابلیت ذخیره نسخه ای در کامپیوتر را نشان می دهد . در این صفحه قابلیت Application Cache توجه نمایید که این خاصیت در حالت عادی ، فعال نیست .

کد	<pre>&lt;!DOCTYPE HTML&gt; &lt;html manifest="demo.appcache"&gt;   &lt;body&gt;     محتویات صفحه   &lt;/body&gt; &lt;/html&gt;</pre>
----	--

### نظیلمات مختلف برای فایل ذخیره شده :

### ( ۱ ) فعال نمودن قابلیت Application Cache :

برای فعال شدن این قابلیت در یک صفحه ، خاصیت Manifest ، خاصیت `<html>` آن تعیین قرار دهید . با هر بار مشاهده کاربر در حافظه ذخیره می شوند . فرمت رایج برای ذخیره فایل های Manifest " .appcache " .

کد	<pre>&lt;!DOCTYPE HTML&gt; &lt;html manifest="demo.appcache"&gt; ..... &lt;/html&gt;</pre>
----	--

## ۲) فایل Manifest :

فایل Manifest که در واقع نسخه ذخیره شده صفحه است ، به مرورگر می گوید که چه چیزهایی را باید و نباید ذخیره نماید . این فایل شامل قسمت اصلی است :

- **CashManiFast** : فایل های تعیین شده در این قسمت در هر بار مشاهده کاربر دانلود و ذخیره خواهند شد.
- **Network** : فایل های تعیین شده در این قسمت نیازمند وجود ارتباط اینترنتی به سرور بوده و هیچگاه ذخیره نمی
- **Fallback** : فایل یا فایل های ذخیره شده در این قسمت ، در صورتی که به هر دلیل صفحه ذخیره شده قابل دسترسی نباشد ، فراخوانی می شوند ( به جای فایل اصلی)

## ۳) تشریح فایل Manifest :

در این قسمت به تشریح خط به خط فایل Manifest برای صفحه وب می پردازیم :

**CashManiFast** قرار گرفته و لیست فایل هایی که در زیر آن تعیین می شوند ، توسط مرورگر ذخیره می شوند . در کد زیر مرورگر فایل تعیین شده را ذخیره می کند . هنگامی که فایل Manifest اجرا می شود مرورگر فایل تعیین شده را از سرور سایت دانلود و ذخیره می کند . بنابراین هر زمان که کاربر نیز به اینترنت متصل نباشد ، سایر فایل های جانبی صفحه ( تم یا اسکریپت ها ) نیز قابل دسترسی خواهند بود .

کد	<pre>CACHE MANIFEST /theme.css /logo.gif /main.js</pre>
----	---

در قسمت بعدی عبارت Network قرار داده می شود . فایل هایی که در زیر عبارت Network تعیین شوند هیچگاه توسط مرورگر ذخیره و دانلود نخواهند شد .

کد	<pre>NETWORK: login.asp</pre>
----	-------------------------------

در قسمت آخر نیز ، **FALLBACK** قرار داده می شود . فایلی که در زیر این عبارت تعیین شود ، در صورتیکه صفحه ذخیره شده در کامپیوتر کاربر به هر دلیلی قابل دسترسی و مشاهده نباشد ، نشان داده خواهد شد .

کد	<b>FALLBACK :</b> /html/ /offline.html
----	---

#### ۴) فرم نهایی فایل MANIFEST :

در نهایت فایل MANIFEST ، به صورت زیر خواهد بود :  
**نکته :** خطوط کدی که با کاراکتر # ، شروع شوند ، توضیحات ( comments ) هستند .

کد	CACHE MANIFEST # 2012-02-21 v1.0.0 /theme.css /logo.gif /main.js  NETWORK: login.asp  FALLBACK: /html/ /offline.html
----	--

#### آپدیت و به روز رسانی فایل ذخیره شده:

فایلی که بر روی کامپیوتر کاربر ذخیره می شود ف بدون تغییر باقی می ماند . مگر اینکه یکی از حالات زیر اتفاق بیفتد :

- کاربر اطلاعات ذخیره شده مرورگر را پاک کند.
- فایل MANIFEST تغییر کند.
- Application Cache بصورت برنامه ریزی شده تغییر کند.

#### نکاتی راجع به Application Cache :

در استفاده از این قابلیت و ذخیره فایل ها بر روی کامپیوتر کاربران دقت نمایید . زیرا مرورگر پس از ذخیره صفحه بر روی کامپیوتر کاربر ، همواره صفحه ذخیره شده نمایش داده و اگر فال اصلی تغییر کند به کاربر اطلاع نمی دهد . مگر اینکه شما فایل MANIFEST آپدیت کند . همچنین مرورگر های مختلف محدودیت مشخصی در حجم فایل های ذخیره شده لحاظ می کنند ( معمولا مرورگرها حداکثر مگابایت برای ذخیره فایل های هر سایت اختصاص می دهند ) .

## HTML 5 < سیستم موقعیت یاب جهانی HTML 5 < موقعیت یابی کاربر

### سیستم موقعیت یاب جهانی HTML 5 :

بوسیله سیستم موقعیت یاب جهانی HTML 5 می توانید مختصات و موقعیت جغرافیایی یک کاربر را پیدا کرده و به وی نمایش دهید . این سیستم از اطلاعات ماهواره ها و ISP ای که به آن متصل هستید ، برای اطلاع از موقعیت شما استفاده می کند .

از آنجایی که موقعیت یابی می تواند حریم خصوصی کاربر را تحت تاثیر قرار دهد ، موقعیت وی تا زمانی که خودش تایید نکند ، در اختیار برنامه قرار نخواهد گرفت .

### پشتیبانی در مرورگر های مختلف :



این تگ در تمام مرورگرهای اصلی پشتیبانی می شود .

### ( اطلاع از موقعیت کنونی کاربر – ( ) getCurrentPosition :

از این متد می توان برای اطلاع از موقعیت دقیق کاربر استفاده نمود . کد مثال زیر یک کد ساده موقعیت یابی را نشان می دهد که بوسیله آن عرض و طول جغرافیایی مکان کاربر را بر روی صفحه نمایش داده شده است .

با کلیک بر روی دکمه فرمان ، موقعیت خود را ببینید :

Example	
<pre>&lt;p id="demo"&gt;&lt;/p&gt; &lt;button onclick="getLocation( )" &gt; &lt;/button&gt;  &lt;script&gt;   var x = document.getElementById("demo");   function getLocation() {     if (navigator.geolocation)       {         navigator.geolocation.getCurrentPosition( showPosition);       }     else       {         x.innerHTML = "مرورگر شما از سیستم موقعیت یاب پشتیبانی نمی کند";       }   } &lt;/script&gt;</pre>	کد

<pre>         }     }  function showPosition( position ) {     x.innerHTML = "Latitude: " + position.coords.latitude + "&lt;br&gt; Longitude : " + position.coords.longitude ; } &lt;/script&gt; </pre>	
---	--

- در مرحله اول چک شده که آیا مرورگر از سیستم موقعیت یاب در HTML پشتیبانی می کند یا خیر و در صورت لزوم کد های مناسب اجرا شود.
- اگر مرورگر از این سیستم پشتیبانی کند متد `getCurrentPosition` ( ) اجرا شده و در غیر این صورت یک پیام هشدار مبنی بر عدم پشتیبانی مرورگر صادر می شود.
- `getCurrentPosition` ( ) با موفقیت اجرا شود مختصات نقطه قرارگیری کاربر را بوسیله یک پارامتر به تابع `ShowPosition` ( ) برمیگرداند.
- `ShowPosition` موفقیت کاربر را بر روی خروجی نمایش می دهد.

مثال فوق کد ساده یک عملیات موقعیت یابی بدون امکان خطایابی بود و در بخش بعد نحوه خطایابی در این سیستم را آموزش می دهیم .

### ( مدیریت خطاها و عدم جوابگویی سیستم :

ممکن است در فرایند موقعیت یابی خطاهایی رخ دهد . `getCurrentPosition` ( ) می توان برای مدیریت خطاهای احتمالی استفاده کرد . این پارامتر یک تابع را تعیین می کند تا در صورت بروز خطا اجرا شده و پیام لازم را به کاربر نمایش دهد .  
کد زیر کد تابع خطایاب سیستم موقعیت یاب HTML 5 را نشان می دهد :

Example	
<pre> &lt;p id="demo"&gt;: برای مشاهده موقعیت خود کلیک نمایید &lt;/p&gt; &lt;button onclick="getLocation( )" &gt; موقعیت یابی &lt;/button&gt;  &lt;script&gt;     var x = document.getElementById("demo");     function getLocation() { </pre>	کد



```

        if (navigator.geolocation)
        {
            navigator.geolocation.getCurrentPosition( showPosition
, showError);
        }
        else
        {
            x.innerHTML = " مرورگر شما از سیستم موقعیت یاب پشتیبانی نمی
کند";
        }
    }

function showPosition( position )
{
    x.innerHTML = "Latitude: " + position.coords.latitude +
"<br> Longitude : " + position.coords.longitude ;
}

function showError(error)
{
    switch(error.code)
    {
        case error.PERMISSION_DENIED:
            x.innerHTML="User denied the request for Geolocation."
            break;
        case error.POSITION_UNAVAILABLE:
            x.innerHTML="Location information is unavailable."
            break;
        case error.TIMEOUT:
            x.innerHTML="The request to get user location timed
out."
            break;
        case error.UNKNOWN_ERROR:
            x.innerHTML="An unknown error occurred."
            break;
    }
}
}

```

</script>	
مشاهده خروجی مثال	خروجی

#### توضیح موارد کد :

- **Permission denied** : در این حالت کاربر اجازه دسترسی به موقعیت خود را نداده است.
- **Position unavailable** : در این حالت یافتن موقعیت صحیح کاربر به علت مشکلات فنی ممکن نیست.
- **Timeout** : در این حالت مدت زمان لازم برای اجرای درخواست تمام شده است.
- **Unkown error** : خطای ایجاد شده نا مفهوم است.

پایان کتاب

و انتقادات خود را به ما اطلاع دهید :

[info@developer1.ir](mailto:info@developer1.ir)

Developer Studio Network