

کتاب آموزش HTML5

نت پارادیس

www.netparadis.com

فهرست

6	مقدمه
7	HTML5
7	چرا HTML5 مهم است؟
8	چه وقت؟
8	اعلام نوع داکيومنت در HTML5
10	صدا در HTML5
12	ویژگی های <audio>
14	ویدئو در HTML5
14	چگونه؟
16	نکات
16	ویژگی های ویدئو
18	canvas چیست؟
19	آموزش Rectangle
22	چطور سایه بیندازیم
23	چگونه یک مسیر بکشیم
25	بوم نقاشی: تصویر
27	بوم نقاشی: متن
29	آموزش Web form
29	ویژگی های ورودی جدید
30	انواع ورودی های جدید
30	جستجو
30	اتریبوت Placeholder
32	آموزش Autofocus
34	آموزش Required field
35	آموزش اتریبیوت DataList
36	آموزش search
38	آموزش ایمیل، شماره تلفن و وب سایت

40	آموزش Input Type : Range
43	آموزش Input Type : Number
46	نوع ورودی: تاریخ
48	تاریخ (<input type="date" />)
48	هفته (<input type="week" />)
49	زمان (<input type="time" />)
50	تاریخ و زمان (<input type="date and time" />)
50	زمان و تاریخ محلی (<input type="datetime-local" />)
52	آموزش Color
54	به سمت وب معنایی
54	من رویایی برای وب دارم که دو بخش است
55	معنا شناسی جدید در سطح متن
55	عناصر انسداد جدید
56	آموزش Semantic mark
56	آموزش Semantic time
58	آموزش Semantic meter
60	آموزش Semantic progress
62	آموزش Semantic section
63	تقسیم بندی محتوا
64	آموزش Semantic header
66	آموزش Semantic footer
69	آموزش Semantic nav
70	آموزش Semantic article
72	آموزش Semantic aside
73	ویژگی های W3C
74	عناصری که نباید استفاده کنید
75	ویژگی هایی که نباید استفاده کنید

HTML5

من معتقدم **HTML5** دستاورد بزرگی خواهد بود (بسیاری از مردم هم چنین اعتقادی دارند). این برنامه ممکن است روش لباس پوشیدن و غذا خوردن ما را تغییر ندهد، اما مطمئناً روش توسعه ی برنامه های وب و طراحی صفحات وب را در **10** سال آینده تغییر خواهد داد. به هر حال این صفحه آموزش **HTML** نمی باشد. **HTML5** چیز کاملاً جدیدی نمی باشد. بسیاری از ویژگی های **HTML5** از **HTML4** یا **HTML1.0** گرفته شده اند. با زمان و انرژی محدود می توانم فقط آنچه را در **HTML5** جدید است و تمرینات قدیمی که دیگر مورد استفاده نیستند را پوشش دهم.

این یک سایت **HTML5** می باشد. هنگامی که در حال کشف و یادگیری **HTML5** هستم، امیدوارم بتوانم آنچه را که یاد گرفته ام با شما به اشتراک بگذارم.

" دو نوع انگیزه وجود دارد که با دونوع هدف منطبق می شود. انگیزه های سلطه گرایی هستند که هدف آنها رسیدن به متعلقات شخصی می باشد که نمی توان به اشتراک گذاشت و در مرکز انگیزه ی مادی قرار می گیرند. انگیزه های خلاقانه نیز وجود دارند که هدف آنها آوردن اموالی است که در آنها هیچ گونه تملک و یا شخصی بودن وجود ندارد.

بهترین زندگی، آن زندگی میباشد که انگیزه های خلاقانه بیشترین نقش را داشته و انگیزه های مالی کمترین را."

برتراند راسل (1872 – 1970)

با توجه به این نوشته **HTML5** یک نسخه ی در حال کار می باشد. به عبارت دیگر **HTML5** هنوز یک کار در حال پیشروی می باشد. ویژگی های آن در **W3C** منتشر شده و روز به روز تغییر می کند.

چرا HTML5 مهم است؟

احتمالاً تاکنون در مورد **HTML5** شنیده اید، فکر می کنم محبوب ترین برنامه ای که فیلم را بدون **Adobe Flash** اجرا می کند، باید از آقای استیو جابز برای تبدیل آن به یک فن آوری معروف تشکر کرد.

علاوه بر این ممکن است در مورد ایجاد انیمیشن بدون پلاگین، کنترل ورودی غنی مانند دیت پیکر، کالرپیکر، اسلایدر بدون جاوااسکریپت و در انتها ذخیره ی داده ی برون خطی (آفلاین) نیز شنیده باشید. به هرحال این هم فقط بخشی از HTML5 می باشد.

چه وقت؟

WC3 در فوریه ی 2011 اعلام کرد که جولای 2014 تاریخی است که HTML5 پیشنهاد رسمی برای استاندارد وب می شود.

زمان را فراموش کنید:

واقعیت این است که جدول زمانی پیشنهادی از طرف WHATWG واقعا اتفاق نیفتاد. توسعه ی وب در دستان دو گروه از مردم واقع شده است.

اولین گروه کمپانی های مرورگر معروفی مانند Mozilla، میکروسافت، اپل، گوگل و اپرا می باشند. گروه دوم نیز طراحان و توسعه دهندگان وب در این زمینه می باشند. ضعف HTML1.1 و HTML2 خیلی ساده است، کمپانی های مرورگر فکر می کردند که ایده ی احمقانه است، بنابراین درهای خود را به روی آن بستند.

امروزه بسیاری از مرورگرها به شدت تمایل دارند که از HTML5 پشتیبانی کنند. حتی اینترنت اکسپلورر که درآهستگی تطبیق استانداردهای جدید شهرت دارد، در IE9 خود از HTML5 پشتیبانی می کند، و این بدین معناست که HTML5 قرار است بزرگ و حقیقی باشد.

از آنجایی که میکروسافت اعلام کرده که هیچ طرحی برای آزاد کردن IE9 برای کاربرهای ویندوز XP ندارد، اگر تمایل دارید از IE9 استفاده کنید، باید از ویندوز vista sp2 یا 7 استفاده کنید.

هیچ دلیلی وجود ندارد که طراحان تحت تاثیر HTML5 قرار نگیرند، زیرا که زندگی را از همیشه برای ما آسانتر کرده است.

اعلام نوع داکيومنت در HTML5

اعلام نوع داکيومنت در HTML5 هم ساده و هم جالب می باشد. این کار به سادگی زیر میباشد:

```
<!DOCTYPE html>
```

بر خلاف **doctype** در **HTML1** و **HTML4** که با آن آشنا هستیم:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "HTTP>  
/www.w3.org/TR/html4/loose.dtd">
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "HTTP>  
/www.w3.org/TR/html4/frameset.dtd">
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "HTTP>  
/www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "HTTP>  
/www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

برخی از آنها حتی در یک خط نوشته نمی شوند. این اولین دلیلی است که نشان می دهد کار طراحان وب با **HTML5** راحت تر شده. یک صفحه ی **HTML5** می تواند به سادگی زیر باشد:

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>HTML5 is rocks!</title>  
</head>  
<body>  
  <h1>HTML5 Rocks!</h1>  
</body>  
</html>
```

اغلب اوقات در تیتتر صفحات وب خو، نوع محتوا، جاوااسکریپت و مسیر فایل **CSS** را مشخص می کنیم، و باز هم زندگی با **HTML5** ساده تر شده است.

به جای `<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">` در **HTML5** می توان فقط از دستور زیر استفاده کرد:

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

در **HTML5** اطلاعات حشو `"type=text/javascript"` دیگر لازم نیستند، در عمل فقط اسکریپت بخش کاربر روشن می باشد.

```
<script src="jquery.js"></script>
```

مشابه CSS هم مانند جاوا اسکریپت، پردازش شهودی درجه ی یک را دریافت می کند. شما از هم اکنون می توانید "type=text/css" را فراموش کنید.

```
<link rel="stylesheet" href="html-my.css">
```

اگر روت <html> شما این شکلی است:

```
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
```

می توانید آن را به شکل زیر ساده کنید:

```
<html lang="en" ">
```

صدا در HTML5

اجازه بدهید در شروع یک سرگرمی داشته باشیم. قرار است در مورد مرورگری صحبت کنیم که فایل صدا را در حالت طبیعی خود پشتیبانی می کند، درست همانطور که برچسب از 1994 پشتیبانی می شود. HTML 5 قرار است به صدای پلاگین (plug-in) مانند مدیاپلیر، سیلورلایت، اپل کویک تایم، و Adobe Flash بد نام خاتمه دهد.

اگر در بالا کنترل اجرا کننده ی صدا نمی بینید، احتمالاً مرورگر شما برچسب صدا را پشتیبانی نمی کند.

در بالا یک گالری از پخش کننده های صدا را، توسط مرورگرهای مهم می بینید. همانطور که می بینید اندازه کنترل پخش کننده ها با یکدیگر متفاوت هستند. پخش کننده ی IE9 در مقایسه با پخش کننده ی safari بسیار بزرگ می باشد. این مسئله برای طراح وب می تواند یک مشکل بزرگ باشد.

اگر برای ویندوز از safari استفاده می کنید، اما بدون Quick Time، برنامه ی Safari شما قرار نیست از برچسب مدیا (هم در حالت صوتی و هم در حالت تصویری) پشتیبانی کند. تصور می کنم که برنامه ی Safari برای اجرای فایل های مدیا، به کدک Quick Time اتکا دارد.

چگونه؟

برای اینکه صفحه ی وب شما موسیقی اجرا کند، کد html شما می تواند به سادگی زیر باشد:

```
<audio src="vincent.mp3" controls></audio>
```

متأسفانه محبوب ترین فرمت صدا **MPEG3 (mp3)** یک استاندارد باز نیست، بلکه یک اختلال عمومی است، و این بدین معنا که مرورگرها برای رمزگشایی آن باید مقداری پول پرداخت کنند و این ممکن است از لحاظ اقتصادی برای شرکت ها و سازمان های کوچکتر عملی نباشد. همانطور که در جدول زیر مشاهده می کنند تنها سران بزرگ ثروت کافی برای رمزگشایی **MP3** را دارند. **Opera** و **firefox** تنها فرمت **vorbis (ogg)** را پشتیبانی می کنند که یک استاندارد باز می باشد.

از طرف دیگر **Vorbis** توسط **Safari** و **IE9** پشتیبانی نمی شود، در حالیکه همیشه داشتن **ogg** و **MP3** در کنار هم خوب است.

Browser	mp3.	wav.	ogg.
Mozilla Firefox 3.6		✓	✓
Opera 10.63		✓	✓
Google Chrome 8.0	✓	✓	✓
(Apple Safari 5.0.3 (with QuickTime	✓	✓	
Microsoft IE 9 Beta	✓	✓	

با استفاده از

and Audio() Support Tester"

href="http://www.happyworm.com/jquery/jplayer/HTML5.Audio.Support/">HTML5

<audio> and Audio() Support Tester

امتحان کنید که مرورگر شما تا چه حد صدا را پشتیبانی می کند.

آنچه که می توانید انجام دهید در زیر ارائه شده است:


```
<audio controls>
  <source src="vincent.mp3" type="audio/mpeg" />
  <source src="vincent.ogg" type="audio/ogg" />
</audio>
```

ارائه کردن یا نکردن نوع **MIME (type="audio/mpeg")** برای مرورگر انتخابی می باشد. مدرنترین مرورگرها به اندازه ی کافی هوشمند هستند تا نوع محتوا را از طریق خود آن مشخص کنند. به هرحال کمک کننده بودن برای مرور گر همیشه خوب است، چرا که باعث می شود مرورگر شما سریعتر و بهتر کار کند.

ویژگی های <audio>

ویژگی	مقدار	Description
controls	*Boolean attribute	شما برای ظاهر شدن اجرا کننده ی داخلی به این ویژگی احتیاج دارید. در غیر اینصورت باید از DOM برای کنترل عنصر صدا برای اجرای موسیقی خود هستید.
autoplay	*Boolean attribute	اگر این ویژگی وجود داشته باشد، مرورگر فقط آواز یا صحبتان را بدون پرسیدن اجازه از ویزیتور شما پخش خواهد کرد.
loop	*Boolean attribute	موسیقی خود را تکرار کنید.
src	url	URL فایل صدای شما
preload	none metadata auto	این ویژگی قبلا با نام autobuffer معروف بود و یک ویژگی Boolean مانند کنترل ها بود. متادیتا- فقط متادیتای مربوط به فایل صدا را بافر می کند. Auto- فایل صدا را قبل از اجرا بافر می کند.

ویژگی **Boolean** ویژگی می باشد که یا در برچسب نمایش داده می شود یا نه. ویژگی **Boolean** فقط نام خود را دارد و دارای مقدار نیست. شما می توانید آن را از این طریق نیز به کار ببرید، هر مقداری که شما به ویژگی **Boolean** اختصاص دادید، فقط یک معنا دارد – **TRUE**.

```
<audio src="vincent.mp3" controls="true" loop="true" autoplay="true"></audio>
```

این ترکیب کاملاً غیرضروری است

```
<audio src="vincent.mp3" controls loop autoplay></audio>
```

و این آن چیزی است که شما لازم دارید.

HTML 5 در واقع تکاملی از وب می باشد، نه انقلابی که فقط گذشته را نابود می کند. به هر حال اجازه دهید کمی به جلوتر برویم.

برای پشتیبانی مرورگری که اصلاً نمی داند **<audio>** چیست، از قبیل **IE8** و پایین تر، با استفاده از برچسب **<object>** باید اجازه دهیم در همان روزهای قدیمی خود باقی بمانند.

```
<audio controls>
  <source src="vincent.mp3" type="audio/mpeg" />
  <source src="vincent.ogg" type="audio/ogg" />
  <object type="application/x-shockwave-flash" data="media/OriginalMusicPlayer.swf" width="225" height="86">
    <param name="movie" value="media/OriginalMusicPlayer.swf" />
    <param name="FlashVars" value="mediaPath=vincent.mp3" />
  </object>
</audio>
```

من در مثال بالا از یک اجرا کننده ی **flash mp3** از **PremiumBeat.com** استفاده می کنم.

اگر مرورگر شما از **HTML5** پشتیبانی نمی کند اما شما فلش پلاگین را دارید، احتمالاً چیزی را از دست نداده اید.

اگر **HTML5** نه مرورگر و نه فلش را پشتیبانی می کند، ممکن است به کاربر اجازه بدهید فایل صدا را دانلود کند و در گرفتن یک اجرا کننده برای اجرای آن، برای آن ها آرزوی خوش شانسی کنید.

```
<audio controls>
  <source src="vincent.mp3" type="audio/mpeg" />
  <source src="vincent.ogg" type="audio/ogg" />
  <object type="application/x-shockwave-flash" data="media/OriginalMusicPlayer.swf" width="225" height="86">
    <param name="movie" value="media/OriginalMusicPlayer.swf" />
    <param name="FlashVars" value="mediaPath=vincent.mp3" />
  </object>
  <a href="vincent.mp3">Download this lovely song and wish you all the best!</a>
</audio>
```

به جای استفاده از مرورگر اجرای صدای استاندارد، با استفاده از خلاقیت خود می توانید برنامه ی خود را بنویسید.

```
<audio id="player" src="vincent.mp3"></audio>
<div>
  <button onclick="document.getElementById('player').play()">Play</button>
  <button onclick="document.getElementById('player').pause()">Pause</button>
  <button onclick="document.getElementById('player').volume += 0.1">Vol+ </button>
  <button onclick="document.getElementById('player').volume -= 0.1">Vol- </button>
</div>
```

ویدئو در HTML5

جریان ویدئو در اینترنت چیز تازه ای نیست، تلاش برای انجام چنین کاری به نیمه ی قرن بیستم برمی گردد. به خاطر پایین بودن پهنای باند شبکه ها و قدرت پایین CPU، هیچ موفقیتی حاصل نشد، تا اینکه Youtube در سال 2005 آغاز به کار کرد.

اکنون با HTML5، می توانیم ویدئو را مانند یک عنصر داخلی، مانند ، قرار دهیم. طول، عرض و پوستر وجود دارند، طول و عرض سایز صفحه را مشخص می کنند و پوستر تنها یک فایل تصویر ثابت می باشد که قبل از اجرای فیلم روی صفحه قرار گرفته است.

چگونه؟

کد اجرای فیلم از صفحه ی وب شما، می تواند به سادگی خط زیر باشد.

```
<video src="rain.mp4" controls width="480" height="360"></video>
```

در واقع زندگی آنطوری نیست که ما فکر می کردیم باید باشد، خط ساده ی کد بالا تنها در **google** و **chrome** و **apple safari** کار می کند و حتی برای اجرای فیلم در **apple safari** باید مطمئن شوید که **quick time** در کامپیوتر شما نصب شده است.

مشابه گرفتاری صوتی، یک فرمت جهانی برای ویدئو نداریم که همه ی مرورگرهای وب از آن پشتیبانی کنند، حداقل در زمان نگارش آن (13 دسامبر 2010).

در اینجا دوباره بازی پول مطرح می شود. **H.264** یک مشکل همگانی است، فایرفاکس و اپرا خرید امتیاز آزاد یک کدک را پذیرفتند که **Theora** و **WebM** می باشد (کدک VP8).

Browser	H.264+ AAC+ MP4	WebM	Theora +Vorbis +Ogg
Mozilla Firefox 3.6		✓	✓
Opera 10.63		✓	✓
Google Chrome 8.0	✓	✓	✓
Apple Safari 5.0.3 (with QuickTime)	✓		
Microsoft IE 9 Beta	✓	✓	

گوگل در ژانویه ی 2011 اعلام کرد که در ورژن آینده ی گوگل کروم از **H.264** پشتیبانی نمی کند. به طور خلاصه کد تک خطی ساده، همانطور که در بالا توضیح دادم، عملی نمی باشد. شما حداقل باید دو منبع برای یک ویدئو تامین کنید.

```
<video controls width="480" height="288">
  <source src="media/html5iscool.mp4" type="video/mp4; codecs="avc1.42E01E, mp4a.40.2" />
  <source src="media/html5iscool.ogv" type="video/ogg; codecs="theora, vorbis" />
```

```
<source src="media/html5iscool.webm" type="video/webm; codecs="vp8, vorbis" />
</video>
```

نکات

برای اینکه فایل ویدئوی خود را در **H.264** رمزگشایی کنید، ممکن است تمایل داشته باشید **HandBrake** را امتحان کنید، که یک **GPL** مناسب و یک ابزار منبع باز می باشد. اگر می خواهید تنها یک ویدئو را رمزگشایی کنید و تمایلی به نصب برنامه ای دیگر روی **OS** تقریباً بسته ی خود ندارید، می توانید فیلم خود را روی **Youtube** آپلود کنید. سپس **Youtube** برنامه ی شما را به فرمت **H.264** آپلود کرده و شما می توانید آن را از **Youtube** دانلود کنید.

همانطور که برای فرمت های **WEBM** و **Theora** ممکن است **Firefogg** را (یک فایرفاکس پلاگین است که حداقل دانلود را لازم دارد.) امتحان کنید.

ویژگی های ویدئو

Attribute	Value	Description
controls	ویژگی Boolean*	اگر می خواهید کنترل اجرا کننده ی مرورگر داخلی در اطراف باشد، کنترل ها را مشخص کنید.
autoplay	ویژگی Boolean*	اگر این ویژگی وجود داشته باشد، مرورگر فیلم شما را بدون پرسیدن اجازه از ویزیتور، اجرا می کند.
loop	ویژگی Boolean*	فیلم شما را تکرار می کند.
src	url	url فایل تصویری شما

preload	none metadata auto	این ویژگی پیش از این با عنوان autobuffer شناخته می شد و مانند کنترل ها یک ویژگی Boolean بود. None – فایل ویدئو را به طور خودکار بافر نکن. Metadata – فقط متادیتای فیلم را بافر کن. Auto – فایل فیلم را قبل از اجرا بافر کن.
width	عرض به پیکسل	عرض پخش کننده ی فیلم.
height	طول به پیکسل	طول پخش کننده ی فیلم.
poster	url یک فایل تصویر	اگر وجود داشته باشد، تا زمان دانلود شدن اولین فریم فیلم تصویر پوستر را نشان می دهد.

مرورگرهایی که در مورد **HTML5** دارای معلومات کمتری هستند چطور؟

اگر فکر می کنید که اجرای ویدئو برای سه دفعه سخت است، به مدت کمی طولانی تر نفس خود را نگه دارید، زیرا قرار است برای تامین کاربرهای بیشتر که از مرورگرهای قدیمی تری مانند **internet explorer 8** و یا پایین تر استفاده می کنند، بیشتر اذیت شوید.

مایکروسافت **internet explorer 9 Beta** را در نوامبر 2010 آزاد کرد که **HTML5** را نیز پشتیبانی می کند. اما برچسب **<video>** به عنوان نگارش هنوز پشتیبانی می شود. به هر حال شما مجبورید برای قرار دادن یک فلش پلیئر در **<object>** به جلوتر بروید زیرا بیشتر کاربرها فلش پلاگین را نصب کرده اند. من معتقدم که مایکروسافت در نهایت **<video>** را پشتیبانی می کند، چرا که همیشه برای اکثر کاربرها رسیدن به آخرین ورژن مرورگر زمان بر است.

```
<video controls width="320" height="240">
  <source src="media/html5iscool.mp4" type="video/mp4; codecs="avc1.42E01E, mp4a.40.2" />
  <source src="media/html5iscool.ogv" type="video/ogg; codecs="theora, vorbis" />
```

```

<source src="media/html5iscool.webm" type='video/webm; codecs="vp8, vorbis"' />
<object width="320" height="240" type="application/x-shockwave-flash" data="media/flowplayer-3.2.5.swf">
  <param name="movie" value="media/flowplayer-3.2.5.swf" />
  <param name="allowfullscreen" value="false" />
  <param name="flashvars"
    value='config={"clip": {"url": "html5iscool.mp4", "autoPlay": false, "autoBuffering": true}}' />
</object>
</video>

```

canvas چیست؟

یکی از داغ ترین موضوعات **HTML5** موضوع **canvas** می باشد، اما دقیقاً **canvas** چیست؟

canvas یک پارچه ی ساده بافته شده ی سنگین می باشد که برای ساختن قایق، چادر، کوله پشتی، خیمه و دیگر ابزاری که استحکام لازم دارند، استفاده می شود. و همچنین توسط نقاشان به عنوان سطح نقاشی که روی چارچوب چوبی نصب شده باشد، استفاده می شود.

شاید این جوابی که شما به دنبال آن بودید، نباشد. به هر حال **canvas** در **HTML5** از کاربرد آن توسط هنرمندان تقلید می کند و بوم نقاشی نامیده می شود.

بوم نقاشی یک فضای خالی بدون رنگ (حتی بدون سفید) است، که قرار است با مداد و قلم موی جاوااسکریپت تبدیل به یک کار هنری شود.

البته که وجود بوم نقاشی تنها برای یک تصویر ثابت نیست. تصویر ثابت می تواند به سادگی به وسیله ی عنصر قدیمی **img** مورد بررسی قرار بگیرد. با جاوااسکریپت به عنوان قلم موی خود، می توانید یک انیمیشن شاد و یا حتی بازیهای تکرار شونده ایجاد کنید.

اگر قبلاً انیمیشن های درخشانی ایجاد کرده اید و امید دارید که یک ورژن **HTML5** ایجاد کنید، ابزاری وجود دارند که در انجام آن به شما کمک می کنند. یکی از آنها **Flash to HTML5** می باشد.

HTML5 مقوله ی خیلی بزرگی است، آنقدر بزرگ که می توانید در مورد آن یک کتاب بنویسید. هدف من در این آموزش تحت پوشش قرار دادن جزئیات مربوط به بوم نقاشی نیست، بلکه تنها معرفی مختصری از آن است. به یاد داشته باشید که این تنها شروع آن است.

برای ایجاد بوم نقاشی، کد ساده ی زیر لازم است.

```
<canvas width="200" height="200"></canvas>
```

به هر حال به یاد داشته باشید که مرز یک بوم نقاشی یک آبجکت ناپیداست. این برنامه فضایی را می گیرد اما شما آن را نمی بینید. اجازه دهید یک سبک مرزی اضافه کرده و از ID برای مرجع استفاده کنیم. هدف از ID این است که با استفاده از DOM (document object model) آن را قابل دسترس کنیم. بدون مرجع ID، جاوا اسکریپت، در این مورد قلم موی شما، هیچ ایده ای برای نقاشی روی بوم ندارد.

```
<canvas id="Canvas1" width="200" height="200" style="border:solid 1px  
#000000"></canvas>
```

اگر در قسمت بالا به جای یک تیره و مربع، یک جمجمه با دو استخوان ضربدری می بینید به این معناست که مرورگر شما بوم نقاشی را پشتیبانی نمی کند. ممکن است تمایل داشته باشید مرورگری دانلود کنید که با بوم نقاشی کار کند، همانطور که در زیر می بینید.

Browsers	Basic Canvas Support
IE 7	✓ (requires ExplorerCanvas)
IE 9 Beta	✓
Firefox 3.0	✓
Safari 3.0	✓
Chrome 3.0	✓
Opera 10	✓

آموزش Rectangle

قبل از فکر کردن در مورد ایجاد بازی های کنشی HTML5، اجازه دهید با اساس آن شروع کنیم.

ابتدا روی دکمه ی قرمز کلیک کنید. اگر دکمه ی قرمز در بوم نقاشی نمی بینید، احتمالا تمایل داشته باشید مرورگر درستی برای ادامه بگیرید.

```
<canvas id="c1" width="200" height="200" style="border:solid 1px #000000;"></canvas>
<button onclick="draw_square();return true;">Red Square</button>
<script>
function draw_square() {
var c1 = document.getElementById("c1");
var c1_context = c1.getContext("2d");
c1_context.fillStyle = "#f00";
c1_context.fillRect(50, 50, 100, 100);
}
</script>
```

برای کشیدن هر چیزی در بوم نقاشی، چیزی بین برچسب **opening** و برچسب **closing** از **canvas** قرار ندهید، مرورگرهایی که بوم نقاشی را پشتیبانی می کنند، آن را نادیده خواهند گرفت. شما می توانید فقط از جاوااسکریپت برای انجام نقاشی خود استفاده کنید.

قانون شماره یک، عنصر بوم نقاشی شما باید دارای یک **ID** باشد، بنابراین می توانیم از جاوااسکریپت برای قراردادن آن استفاده کنیم. قانون شماره دو، هر بوم نقاشی دارای یک کانتکست می باشد. در واقع کانتکست بوم نقاشی آن چیزی است که قرار است روی آن نقاشی بکشیم، نه خود بوم.

```
var c1 = document.getElementById("c1");
var c1_context = c1.getContext("2d");
```

از لحاظ نگارش، نه تنها کانتکست های دو بعدی، بلکه سه بعدی آن نیز در دسترس می باشند. به هرحال کانتکست سه بعدی که **webGL** نامیده می شود، برای شما توسط **W3G** یا **WHATWG** آورده نمی شود بلکه توسط یک شرکت غیرانتفاعی به نام **Khronos Group** ارائه می شود. علاوه بر آن گوگل هم در دسامبر **2010** برای اطمینان بیشتر آینده ی این استاندارد، تولید شگفت انگیز **Body Browser** را آزاد کرد.

اجازه دهید از رنگ قرمز (**fillStyle="#f00";**) استفاده کرده و یک مربع قرمز با اضلاع **100px** بکشیم.

```
c1_context.fillStyle = "#f00";
c1_context.fillRect(50, 50, 100, 100);
```

جدول زیر روش های کشیدن مستطیل را نشان می دهد.

Methods of "Context"	Descriptions
fillStyle	رنگ CSS، الگو یا گرادیان داخل شکل. پیش فرض fillstyle رنگ سیاه می باشد.
strokeStyle	رنگ یا روش CSS از خطوط شکل.
fillRect(x, y, width, height)	مسططیلی بکشید که از نقطه ی X و y شروع می شود و اندازه ی طول X عرض باشد.
strokeRect(x, y, width, height)	مسططیلی فقط با حاشیه های آن رسم کنید.
clearRect(x, y, width, height)	ناحیه مشخص شده در مختصات X و y و همچنین در عرض X طول را روشن کنید.

```

<div>
  <canvas id="Canvas2" width="200" height="200" style="border:solid 1px #000000;"></canvas>
  <div>
    <button onclick="blue_square_2();return true;">Blue Square</button>
    <button onclick="red_stroke_2();return true;">Red Square</button>
    <button onclick="clear_rect_2();return true;">Erase Everything</button>
  </div>
</div>
<script>
var c2 = document.getElementById("c2");
var c2_context = c2.getContext("2d");
function blue_square_2() { //Blue color square
  c2_context.fillStyle = "#00f";
  c2_context.fillRect(50, 50, 100, 100);
}
function red_stroke_2() { //Red color edges
  c2_context.strokeStyle = "#f00";
  c2_context.strokeRect(45, 45, 110, 110);
}
function clear_rect_2() { //Clear all
  c2_context.clearRect(40, 40, 120, 120);
}

```

```
}
</script>
```

چطور سایه بیندازیم

از آنجایی که اکنون می دانید چطور مستطیل بکشید، احتمالاً تمایل داشته باشید با انداختن سایه آن را زیباتر کنید. این کار خیلی راحت انجام می شود. لازم است که فقط چند خط بیشتر به کد برنامه خود اضافه کنید. اجازه دهید اول به چند ویژگی که در انداختن سایه دخیل هستند، نگاهی داشته باشیم.

Context properties	Descriptions
shadowOffsetX	فاصله ی افقی (x-axis) بین سایه و شکل در پیکسل.
shadowOffsetY	فاصله ی عمودی (y-axis) بین سایه و شکل در پیکسل.
shadowBlur	تا چه اندازه می خواهید سایه شما محو باشد.
shadowColor	واضح است که برای تنظیم رنگ سایه استفاده می شود.

و در ادامه می بینید.

```
<div>
<canvas id="c8" width="200" height="200" style="border:solid 1px #000000;">
</canvas>
</div>
<script>
var c8 = document.getElementById("c8");
var c8_context = c8.getContext("2d");
function draw_rectangle() {
c8_context.shadowOffsetX = 5;
c8_context.shadowOffsetY = 5;
c8_context.shadowBlur = 10;
c8_context.shadowColor = "DarkGoldenRod";
c8_context.fillStyle = "Gold";
c8_context.fillRect(20, 20, 100, 120);
}
window.onload = draw_rectangle();
</script>
```

همانطور که در کد بالا می بینید، برای ترسیم یک شکل با سایه، قبل از کشیدن سایه لازم است ویژگی های آن را مشخص کنید.

چگونه یک مسیر بکشیم

ابتدا باید یک خط بکشیم. در اینجا چگونگی کشیدن خط قهوه ای به شما ارائه شده است.

```
<div>
  <canvas id="Canvas2" width="600" height="200" style="border:solid 1px #000000;"></canvas>
  <div>
    <button onclick="Vertical_line();return true;">Click me to draw a brown vertical line</button>
  </div>
</div>
<script>
  var c3 = document.getElementById("c3");
  var c3_context = c3.getContext("2d");
  function Vertical_line() {
    c3_context.moveTo(300, 10);
    c3_context.lineTo(300, 190);
    c3_context.strokeStyle = "brown";
    c3_context.stroke();
  }
</script>
```

روش هایی که واقعا در اینجا اتفاق می افتند، عبارتند از **moveTo**، **lineTo**، **stroke**، و **strokeStyle**.

Context method	Descriptions
moveTo(x,y)	حرکت به نقطه ی آغاز با مختصات X و y.
lineTo(x,y)	از نقطه ی شروع به این نقطه خطی بکشید. دوباره X و y مختصات باشند.
strokeStyle	رنگ CSS خط
stroke	روشی که در واقع باعث می شود جاوا اسکریپت خط بکشد.
beginPath	قبل از اینکه شروع به کشیدن خط جدید با رنگ متفاوت کنید، باید beginPath را فرا بخوانید.

نمایش آن را در زیر چک کنید، که روش **beginPath** نقش مهمی دارد. بدون آن آخرین **stroke** خط قبلی را با آخرین رنگ **strokeStyle** فراخوانده شده، دوباره ترسیم می کند.

```
<div>
  <canvas id="c4" width="600" height="200" style="border:solid 1px #000000;"></canvas>
  <div>
    <button onclick="Vertical_2px_Red();return true;">Vertical 2px Red line</button>
    <button onclick="Vertical_1px_Blue();return true;">Vertical 1px Blue line</button>
    <button onclick="Horizontal_2px_Green();return true;">Horizontal 2px Green line</button>
    <button onclick="Clear_line();return true;">Erase Everything</button>
  </div>
</div>
<script>
  var c4 = document.getElementById("c4");
  var c4_context = c4.getContext("2d");
  function Vertical_2px_Red() {
    c4_context.beginPath();
    c4_context.moveTo(300, 10);
    c4_context.lineTo(300, 190);
    c4_context.strokeStyle = "Red";
    c4_context.stroke();
  }
  function Vertical_1px_Blue() {
    c4_context.beginPath();
    c4_context.moveTo(350.5, 10);
    c4_context.lineTo(350.5, 190);
    c4_context.strokeStyle = "Blue";
    c4_context.stroke();
  }
  function Horizontal_2px_Green() {
    c4_context.beginPath();
    c4_context.moveTo(100, 100);
    c4_context.lineTo(500, 100);
    c4_context.strokeStyle = "Green";
    c4_context.stroke();
  }
  function Clear_line() {
    c4_context.clearRect(1, 1, 600, 190);
  }
</script>
```

شاید متوجه شده باشید که در نمایش بالا دو دکمه برای کشیدن خط عمودی به شما کمک می کنند. یکی خط یک پیکسلی و دیگری خط دو پیکسلی می کشد. آنچه تفاوت ایجاد میکند مانند زیر است.

```
c4_context.moveTo(300, 10);
```

```
c4_context.lineTo(300, 190);
```

9

```
c4_context.moveTo(300.5, 10);
```

```
c4_context.lineTo(300.5, 190);
```

برای کشیدن خطی با عرض یک پیکسل، مختصات را باید به سمت راست یا چپ به 0.5 پیکسل تغییر دهید. این امر به خاطر این است که مقدار صحیح مختصات قرار است در حاشیه ی هر پیکسل در نظر گرفته شود. کشیدن خط در لبه باعث خواهد شد که خط در سراسر دو پیکسل مجاور گسترده شود. با تغییر مختصات 0.5 پیکسل، خط هنوز بین دو پیکسل گسترده شده اما نیمی از هر پیکسل را اشغال می کند.

بوم نقاشی: تصویر

کشیدن خط یا اشکال ساده خسته کننده می باشد، اجازه بدهید چند تصویر طراحی کنیم.

در نمایش بالا فقط از کانتکست روش **drawimage()** استفاده کردم. به هر حال این روش می تواند 3، 5 و یا 9 استدلال داشته باشد. من سه عملکرد دارم که عبارتند از **draw dragon** (کشیدن اژدها)، **draw smaller dragon** (کشیدن اژدهای کوچکتر) و **draw dragon head** (کشیدن سر اژدها) و هر کدام از آنها 3، 5 و 9 استدلال مربوطه را دنبال می کنند.

Overloading of drawImage()	Description
3 arguments drawImage (img,x,y)	Img اساس تصویر می باشد. X و y مختصات هستند تا آبجکت تصویر را قرار دهند.
5 arguments drawImage (img,x,y,width,height)	Img اساس تصویر می باشد. X و y مختصات هستند تا آبجکت تصویر را قرار دهند. طول و عرض اندازه ای می باشد که شما برای تصویر می خواهید، می توانید آن را طوری که می خواهید تغییر اندازه دهید.

<p>9 arguments</p> <p>drawImage (img, crop_x, crop_y, crop_width, crop_height, x, y, width, height)</p>	<p>Img اساس تصویر می باشد. Crop-x و crop-y مکانی است که شما شروع به بریدن تصویر خود می کنید.</p> <p>Crop-width و crop-height اندازه ای است که می خواهید طبق آن تصویر را برش بزنید.</p> <p>X و y همچنان مختصات هستند تا آبجکت تصویر را قرار دهند. طول و عرض اندازه ای می باشد که شما برای تصویر می خواهید، می توانید آن را طوری که می خواهید تغییر اندازه دهید.</p>
---	--

در اینجا کد را مشاهده می کنید.

```

<div>
  <canvas id="c5" width="600" height="300" style="border:solid 1px #000000;"></canvas>
  <div>
    <button onclick="draw_dragon();return true;">Draw Dragon</button>
    <button onclick="draw_smaller_dragon();return true;">Draw smaller dragon</button>
    <button onclick="draw_dragon_head();return true;">Draw Dragon Head</button>
    <button onclick="Clear_image();return true;">Erase Everything</button>
  </div>
</div>
<script>
var c5 = document.getElementById("c5");
var c5_context = c5.getContext("2d");
var dragon = new Image();
dragon.src = 'images/chinese_dragon.png';
function draw_dragon() {
c5_context.drawImage(dragon, 100, 5);
}
function draw_smaller_dragon() {
c5_context.drawImage(dragon, 10, 5, 58, 100);
}
function draw_dragon_head() {
c5_context.drawImage(dragon, 0, 19, 69, 97, 300, 100, 103, 145);
}
function Clear_image() {
c5_context.clearRect(1, 1, 600, 300);
}
</script>

```

بوم نقاشی: متن

چگونه متن را در بوم نقاشی اجرا کنیم؟

کلمات جادویی برای چاپ متن در بوم نقاشی عبارتند از **Font**، **Strokestyle**، **Fillstyle**، **TextBaseline** و در انتها **strokeText** و **fillText**.

```
<div>
  <canvas id="c6" width="600" height="200" style="border:solid 1px #000000;"></canvas>
  <div>
    <button onclick="showFillText();return true;">Fill Text</button>
    <button onclick="showStrokeText();return true;">Stroke Text</button>
    <button onclick="Clear_text();return true;">Erase Everything</button>
  </div>
</div>
<script>
var c6 = document.getElementById("c6");
var c6_context = c6.getContext("2d");
function showFillText() {
  c6_context.fillStyle = '#f00';
  c6_context.font = 'italic bold 30px sans-serif';
  c6_context.textBaseline = 'bottom';
  c6_context.fillText('HTML5 is cool!', 50, 100);
}
function showStrokeText() {
  c6_context.strokeStyle = "#003300";
  c6_context.font = '40px san-serif';
  c6_context.textBaseline = 'bottom';
  c6_context.strokeText('HTML5 is cool?', 300, 100);
}
function Clear_text() {
  c6_context.clearRect(1, 1, 600, 300);
}
</script>
```

Context method	Description
fillText(text,x,y)	متن را با رنگ داخل آن چاپ می کند. رنگ متن توسط fillstyle مشخص می شود.
strokeText(text,x,y)	متن را فقط با رنگ در OutLine متن چاپ می کند. رنگ متن توسط strokeStyle() تنظیم میشود.

strokeStyle	رنگ CSS برای متن که strokeText را فرا می خواند.
fillStyle	رنگ CSS برای متن که fillText را فرا می خواند.
font	فونت CSS مانند bold ، 10px ، san-serif
textBaseline	توضیح این مورد کمی گیج کننده است و برای آن به نمایش دیگری احتیاج خواهیم داشت. مقدار برای این ویژگی میتواند top ، hanging ، middle ، alphabetic ، ideographic ، و bottom باشد. مقدار پیش فرض alphabetic می باشد.

تصویری را که می بینید از WHATWG اقتباس کرده ام که توضیح کامل برای انواع خطوط پایه ی کتن می باشد. آنچه را که باید مشاهده کنید این است که چگونه یک متن در ارتباط با آن خطوط پایه قرار می گیرد.



در اینجا یک خط خاکستری در $y=100$ کشیده ام و قصد دارم هر لغت را در $y=100$ قرار دهم، اما با استفاده از **textbaseline** متفاوت.

در مورد نگارش، فایرفاکس هیچگونه پشتیبانی روی خط پایه ی **hanging** ندارد.

```
c7_context.textBaseline = "top";
c7_context.fillText('Top', 5, 100);
c7_context.textBaseline = "bottom";
```

```
c7_context.fillText('Bottom', 80, 100);
c7_context.textBaseline = "middle";
c7_context.fillText('Middle', 200, 100);
c7_context.textBaseline = "alphabetic";
c7_context.fillText('Alphabetic', 300, 100);
c7_context.textBaseline = "hanging";
c7_context.fillText('Hanging', 400, 100);
```

اگر قصد دارید چیزی را در بوم نقاشی گسترش دهید و تمایل به پشتیبانی کاربرهایی داشتید که از **IE8** یا پایین تر استفاده می کنند، می توانید از یک جاوا اسکریپت با منبع آزاد به نام **Explorercanvas** استفاده کنید. اما در جریان باشید که وجود مسایل غیریکنواختی ممکن است باعث عصبانیت شما بشوند.

باز هم این معرفی مختصری از بوم نقاشی **HTML5** بود. هنوز ویژگی های جالب دیگری در مورد این بوم نقاشی وجود دارد که باید در مورد آنها بیشتر بدانید.

آموزش Web form

تکامل **HTML5** برای ساختن یک دنیای بهتر از وب می باشد. نوشتن کدهای کمتر (به ویژه جاوا اسکریپت) خیلی خوب است، در انتهای روز کارهای تکراری توسعه دهندگان وب باید توسط مرورگرها گرفته شود. پس از همه ی اینها، این مهم ترین دلیل کشف ماشین توسط انسان است.

آنچه در زیر می بینید لیستی از ویژگی های ورودی های جدید و انواع ورودی ها می باشد که باید همه را تک به تک بررسی کنیم.

ویژگی های ورودی جدید

```
<input type="text" "new input attribute" />
$ Placeholder
$ Autofocus
$ Required
$ DataList
```

برای سالها فقط مشتتی از انواعی به ما ارائه شد که در **Input** مورد استفاده قرار می گیرند، مانند متن، رادیو، چک باکس، پسورد، فایل و سابمیت. در **HTML5** با انواع جدید روی بلوک سرگرمی بیشتری خواهیم داشت.

انواع ورودی های جدید

`<input type="new input type" />`

جستجو

- ایمیل، URL، تلفن
- دامنه به عنوان اسلایدر
- عدد به عنوان اسپینر
- تاریخ و زمان
- انتخاب کننده ی رنگ

Placeholder

Placeholder یک تکست باکس است که وقتی هیچ مقداری مشخص نشده و یا هیچ مرکزیتی وجود ندارد، متنی را با سایه ی روشن تر نگه می دارد.

در زیر می بینید که اگر هیچ مقداری مشخص نشده باشد یک **placeholder** چگونه به نظر می رسد (آنچه شما می بینید فقط یک تصویر است و نه یک **placeholder** واقعی، روی آن کلیک نکنید.)



وقتی که تکست باکس روی چیزی تمرکز می کند، متن خاموش می شود و شما باید متن خود را در آن قرار دهید.



در اینجا کد مربوط به **placeholder** را می بینید. دیگر لازم نیست که جاوا اسکریپت و **CSS** به ظاهر متن رسیدگی کنند.

`<label for="first_name">First Name</label> : <input id="first_name" placeholder="First name goes here">`

اجازه بدهید ببینیم کدام مرورگرها **placeholder** را پشتیبانی می کنند.

Browsers	Placeholder Support
IE 10	✓
Firefox 3.7	✓
Safari 4.0	✓
Chrome 4.0	✓
Opera 11	✓

اگر **placeholder** در صفحه ی وب شما فقط یک ویژگی خوب است، احتمالاً در مورد پشتیبانی مرورگرها به جز موارد بالا، اذیت نخواهید شد.

در غیراینصورت مجبورید راه های اضافه ای بروید. ابتدا برای اینکه چک کنید مرورگر **placeholder** را پشتیبانی می کند، از جاوا اسکریپت استفاده کنید. برای مرورگرهایی که **placeholder** را پشتیبانی نمی کنند از **javascript+CSS** استفاده کنید تا **placeholder** خود را ایجاد کنید.

```
<label for="demo">Placeholder demo</label> : <input id="demo" placeholder="Support Placeholder" />
<script>
function testAttribute(element, attribute)
{
var test = document.createElement(element);
if (attribute in test)
return true;
else
return false;
}
if (!testAttribute("input", "placeholder"))
{
window.onload = function()
{
var demo = document.getElementById("demo");
var text_content = "No Placeholder support";
demo.style.color = "gray";
demo.value = text_content;
}
```

```
demo.onfocus = function() {
  if (this.style.color == "gray")
  { this.value = ""; this.style.color = "black" }
}
demo.onblur = function() {
  if (this.value == "")
  { this.style.color = "gray"; this.value = text_content; }
}
}
}
</script>
```

عملکرد **testAttribute** در بالا برای امتحان کردن این است که آیا یک ویژگی از یک عنصر توسط یک مرورگر پشتیبانی می شود. در این مورد می خواهیم بدانیم که آیا مرورگر شما ویژگی **placeholder** مربوط به عنصر ورودی را پشتیبانی می کند. اگر **placeholder** پشتیبانی نمی شود، من مرورگر را به اجرای چند کد جاوااسکریپت مجبور می کنم تا چیزی با عنوان **placeholder** ایجاد کند. شما می توانید از جاوااسکریپت خود استفاده کنید.

Placeholder demo :

اگر از مرورگری استفاده می کنید که **placeholder** را پشتیبانی می کند، باید قار باشید عبارت **support place holder** را در تکست باکس ببینید. بالعکس اگر از مرورگری استفاده می کنید که **placeholder** را پشتیبانی نمی کند، عبارت **"No Placeholder Support"** را در تکست باکس خواهید دید.

آموزش Autofocus

Autofocus یک ویژگی **Boolean** از فیلد فرم می باشد که مرورگر را به تنظیم تمرکز روی آن، وقتی که یک صفحه بارگذاری می شود، وامی دارد. اگر هنوز متوجه نشدید که چیست، لطفاً به گوگل رفته و بدون اینکه ابتدا روی تکست باکس بزرگ کلیک کنید.

فقط رشته ی جستجوی خود را تایپ کنید. این مسئله به این خاطر است که گوگل به طور خودکار روی آن تکست باکس تمرکز دارد (منظور من این نیست که گوگل از **autofocus** مربوط به **HTML5** استفاده می کند.) و با یک کلیک ذخیره می شود.

در زیر کد انجام این کار را می بینید که خیلی ساده می باشد.

```
<label for="name">Your name goes here :</label><input id="name" type="text" autofocus /> <br />
<label for="mothername">Your dog's name goes here :</label><input id="mothername" type="text" />
```

Autofocus یک ویژگی **Boolean** می باشد و هیچ نیازی به تنظیم مقدار روی آن ندارد.

طبق معمول همه ی مرورگرها **autofocus** را پشتیبانی نمی کنند.

Browsers	Autofocus Support
IE 10	✓
Firefox 4	✓
Safari 4.0	✓
Chrome 3.0	✓
Opera 10	✓

برای اجرای یک **autofocus** با خطای کمتر باید دوباره از جاوااسکریپت استفاده کنید تا چک کنید آیا مرورگر **autofocus** را پشتیبانی می کند، اگر نه تمرکز

را فقط با جاوااسکریپت تنظیم کنید.

در زیر نمایش و همچنین یک تست میبینید. اگر از **autofocus** پشتیبانی کند، متوجه می شود که مکان نما در تکست باکس اول چشمک می زند. و اگر مرورگر شما از آن پشتیبانی نمی کند، تمرکز روی دومین تکست باکس قرار می گیرد.

Support Autofocus :

No Autofocus Support :

```
<script>
function testAttribute(element, attribute) {
    var test = document.createElement(element);
    if (attribute in test) {
        return true;
    }
}
```

```

else
    return false;
}
window.onload = function() {
    if (!testAttribute('input', 'autofocus'))
        document.getElementById('Text2').focus();
    //for browser has no autofocus support, set focus to Text2.
}
</script>
<label for="Text1">Support Autofocus :</label><input id="Text1" type="text" autofocus /><br />
<label for="Text2">No Autofocus Support :</label><input id="Text2" type="text" />

```

آموزش Required field

فیلد مورد نیاز فیلدی می باشد که باید قبل از ارائه ی یک فرم با یک مقدار پر شود. این فیلد گاهی با عنوان فیلد اجباری (**compulsory or mandatory field**) شناخته می شود.

در مورد نگارش فقط اپرا و فایرفاکس تکست باکس این فیلد را پشتیبانی می کنند. لطفا به جدول زیر توجه کنید.

Browsers	"Required" Support
IE 10	✓
Firefox 4	✓
Safari 5	
Chrome 20	✓
Opera 9	✓

تبدیل یک فیلد به فیلد مورد نیاز کاملا ساده است، شما تنها باید یک ویژگی **Boolean** به عنصر ورودی خود اضافه کنید.

```

<form>
    <label for="movie">What is your favorite movie : </label>
    <input name="movie" type="text" required />
    <input type="submit" value="Submit" />

```

</form>

تصویر زیر به ما نشان می دهد که اگر یک فیلد در اجرا خالی باشد، چگونه فایرفاکس و اپرا در پر کردن مقدار آن کاربر را به جلو می برند.

آموزش اتریبیوت Datalist

دیتا لیست به نظر می رسد شبیه پیشنهادهایی است که در هنگام تایپ در تکست باکس ارائه می شود، مانند آنچه در جستجوی گوگل مشاهده می کنید. البته که فقط به نظر شبیه می رسد، اما حتی نزدیک به آن هم نیست. دیتا لیست مانند لیست رو به پایین انتخاب می باشد، اما کاربر علاوه بر آنچه در لیست ارائه شده می تواند آزادانه هرچه می خواهد در تکست باکس قرار دهد. در مورد لیست انتخاب هر چیزی در قسمت کاربر قرار می گیرد. قبل از اینکه به کد و نمایش برسیم، اجازه بدهید ببینیم چه مرورگرهایی آن را پشتیبانی می کنند.

Browsers	Datalist Support
IE 10	✓
Firefox 4	✓
Safari 5	
Chrome 20	✓
Opera 11	✓

حالا ببینیم چطور یک دیتالیزت HTML5 را تکمیل می کنیم که کاملاً ساده است.

```
<label for="country_name">Country : </label>
<input id="country_name" name="country_name" type="text" list="country" />
<datalist id="country">
  <option value="Afghanistan">
  <option value="Albania">
  <option value="Algeria">
  <option value="Andorra">
  <option value="Angola">
</datalist>
```

خیلی راحت در دیتالیزت نام کشور خود را وارد کنید.

Country :

اگر مرورگری ندارید که آن را پشتیبانی کند، در اینجا می توانید ببینید که در فایرفاکس چگونه به نظر می رسد.



آموزش search

اطلاعات وب هر ثانیه در حال رشد است. از آنجایی که گوگل جستجو (search) را در اواخر دهه ی 90 دوباره تعریف کرد، یافتن اطلاعات از طریق مسیرهای سازماندهی شده از بین رفته است. امروزه می توانیم یک سرچ باکس (search box) تقریباً در محتوای همه ی وب سایت های غنی قرار دهیم، این کار حتی در بلاگ شخصی هم محبوب می باشد.

در HTML5 می توانیم یک تکست باکس را به عنوان سرچ باکس معرفی کنیم. در واقع آنچه می توانید برای سرچ باکس خود انجام دهید عبارت است از

```
<input id="mysearch" type="search" />
```

با انجام چنین کاری انتظار نداشته باشید که مرور گر به طور خودکار آن را با یک موتور جستجو ترکیب کند. این تقریباً مانند یک تکست باکس عادی می باشد `<input type="text">`، اما مرورگر به طور خودکار کمی آن را دستکاری می کند تا زیباتر شود.

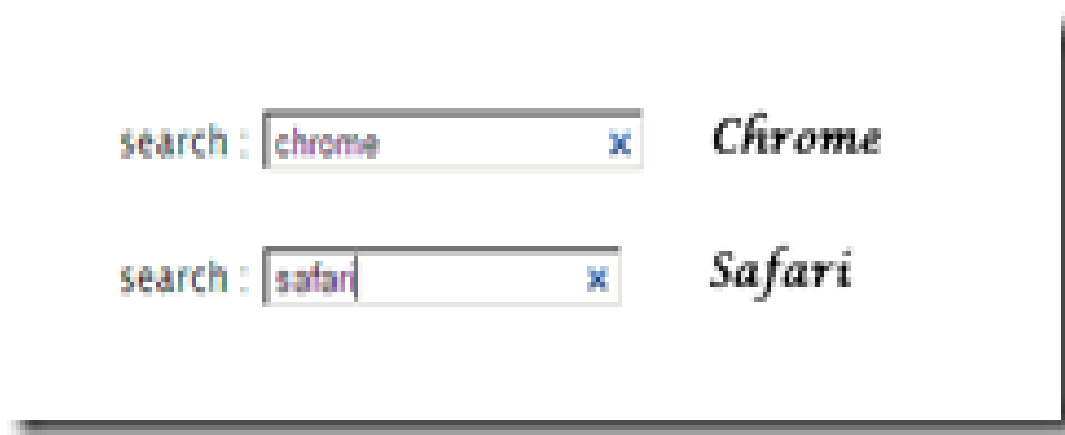
در نگارش فقط apple safari (آزمایش شده با Safari 5) و گوگل کروم (آزمایش شده با کروم 8) این کار را انجام می دهند.

Browsers	Tweak search box?
IE 10	✓
Firefox 13	

Safari 5	✓
Chrome 8	✓
Opera 9	

در اینجا یک نمونه ساده را می بینید.

search



همانطور که ممکن است متوجه شده باشید وقتی چیزی در سرچ باکس می نویسید، یک ضربدر آبی در در تکست باکس ظاهر می شود. وقتی روی این ضربدر کلیک کنید، رشته ی ورودی شما پاک شده و می توانید تایپ یک رشته ی جدید را آغاز کنید.

می توانید ویژگی های بیشتری به آن اضافه کنید، مانند **placeholder** و **autofocus**.

در اینجا کد مربوط به آن را مشاهده می کنید.

```
<label for="mysearch2">Enter your search string here : </label>
<input id="mysearch2" type="search" placeholder="search">
```

آموزش ایمیل، شماره تلفن و وب سایت

ایمیل، شماره تلفن و وب سایت **url** از متداول ترین جزئیات تماس هستند که الگوهای ورودی خاص خود را دارند. ایمیل دارای علامت **@** می باشد و **.com**، **.net** و یا چیز دیگری به پایان می رسد. وب سایت حداقل دارای یک دات در وسط است و اکثر آنها با **.com** به پایان می رسند و شماره تلفن هم فقط عدد است، بدون هیچ حروف الفبایی.

گیج نشوید زیرا این انواع حداقل هنوز مربوط به اعتبار سنجی داده نیستند. شما مجبور خواهید بود از همان عبارت های متداول برای انجام آن استفاده کنید.

در جدول زیر می توانید ببینید چگونه به نظر می رسد، ممکن است سعی کنید در هر کدام از فیلدها چیزی وارد کنید و در مقایسه با تکست باکس ساده، چیز خاصی نخواهید دید (**<input type="text" />**).

html-5.my demo	
Email Address	
Web Site	
Phone Number	

و در اینجا کد را می بینید که چگونه به نظر می رسد.

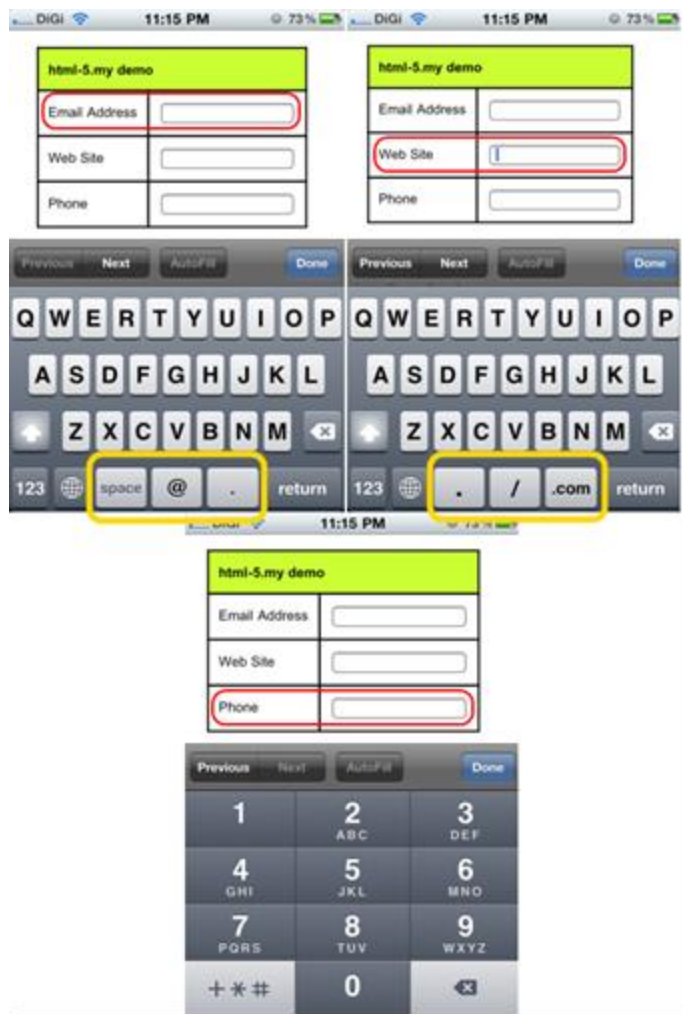
```
<input id="email" type="email" />
```

```
<input id="website" type="url" />
```

```
<input id="phone" type="tel" />
```

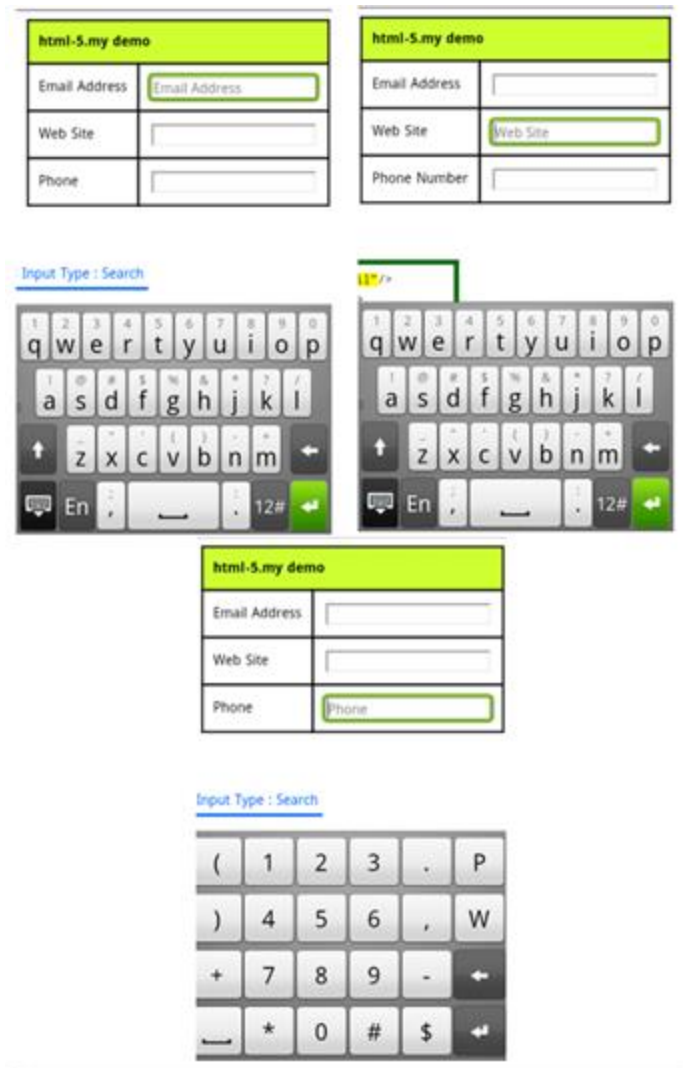
به هر حال تفاوت هایی در چگونگی رفتار صفحه لمسی مرورگر محور با این فیلدها وجود دارد.

در زیر صفحه ی مربوط به **iphone 4 (iOS 4.2)** و **HTC Desire (android 2.2)** را در زمان تمرکز هر فیلد می بینید.



همانطور که می بینید، وقتی روی فیلد شماره تلفن تمرکز می کنید، مرورگر به شما یک صفحه ی شماره گیر ارائه می دهد، **iphone** حتی به جلوتر رفته تا قرار دادن ورودی ایمیل و وب **url** را ساده تر کند، هر ایمیل دارای یک نشان **@** می باشد و اکثر وب های **url** با **.com** به پایان می رسند.

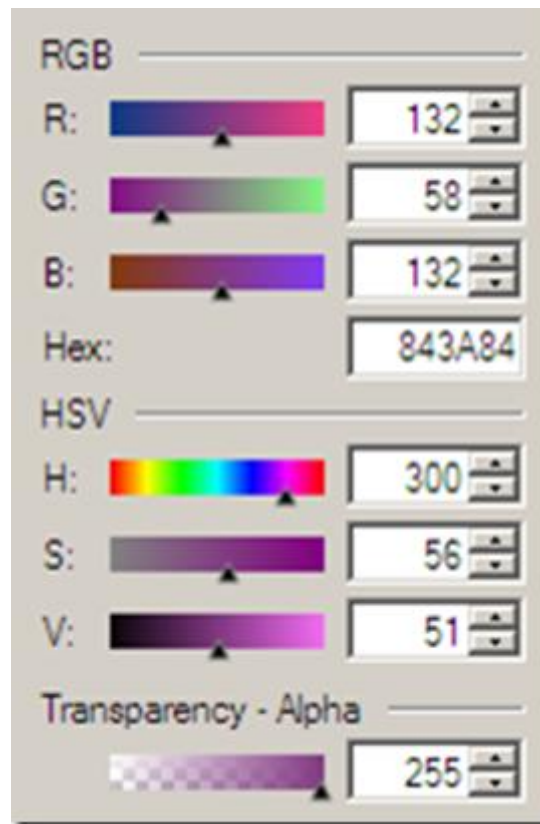
از طرف دیگر ورودی صفحه کلید اندروید من به اندازه ی آیفون کاربر پسند نیست.



در خاتمه اینکه دلیلی وجود ندارد که شما از اکنون از این انواع استفاده نکنید، چرا که مرورگرهایی که انواع جدید را نمی شناسند با آنها مانند یک متن ساده رفتار می کنند. این امر صفحه ی شما را با کاربرهای موبایل دوستانه تر می کند و برای ماشین خواناتر می کند. این ویژگی وب معنایی می باشد.

آموزش Input Type : Range

کنترل اسلایدر یک کاربر اینترفیس (user interface) برای تنظیم یک عدد در یک دامنه بسیار مشهود می باشد. یک نمونه اسلایدر معمولاً در انتخاب گر رنگ (color picker) یافت می شود، جایی که می توانیم پیکان را به سمت راست یا چپ درگ کرده و مقدار RGB درست را انتخاب کنیم.



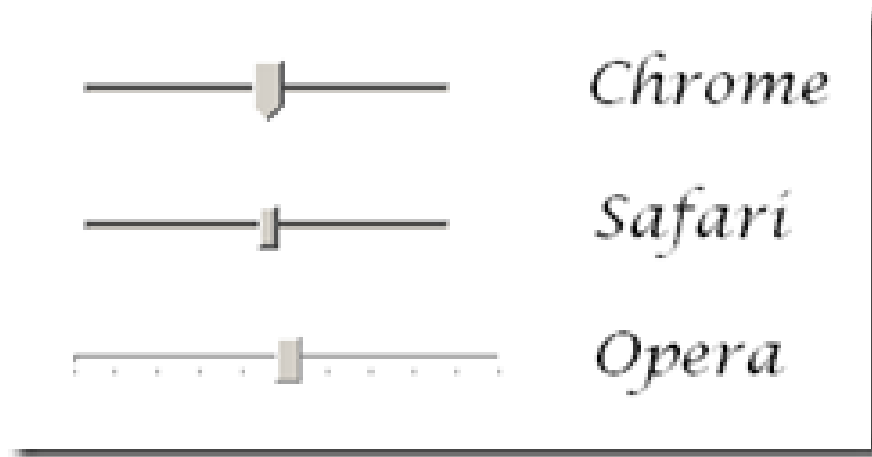
قبلا برای ایجاد اسلایدر در صفحه ی وب بین جاوا اسکریپت یا فلش یکی را به عنوان راه حل انتخاب می کردیم. **HTML5** برای ذخیره ی هزاران بایت در کد شما اینجاست. با نوع جدید ورودی، کنترل اسلایدر

(`<input type="range">`) مانند لیست **dropdown** خواهد بود.

کد آن به سادگی زیر می باشد.

```
<input id="test" type="range" />
```

طبق معمول طراحان هر مرورگر در اجرای کاربر اینترنترفیس سلیقه ی مربوط به خود را دارند. مجددا تمایل دارم به شما یک گالری از اسلایدهای متنوعی نشان دهم که مرورگرهای مختلف در ویندوزها اجرا می کنند.



مرورگرها `<input type="range">` را اجرا نمی کنند، طوری که اسلایدر به سادگی آن را مانند یک تکست باکس ساده اجرا خواهد کرد.

مرورگرهایی که `<input type="range">` را مانند اسلایدر اجرا می کنند در جدول زیر ارائه شده اند.

Browsers	Render Range input as Slider
IE 10	✓
Firefox 13	
Safari 5	✓
Chrome 8	✓
Opera 11	✓

نوع دامنه دارای ویژگی های جالبی است که شاید تمایل داشته باشید یاد بگیرید:

Attribute	Descriptions
value	مقدار یک ویژگی متداول از عنصر ورودی می باشد. مقدار می تواند هر عدد ممیزی معتبری باشد، نه اینکه لزوما عدد صحیح باشد. مقدار پیش فرض عبارت است از کمترین مقدار به اضافه ی نیمی از بیشترین مقدار.

min	کمترین مقدار دامنه: کمترین مقدار پیش فرض صفر می باشد.
max	بیشترین مقدار دامنه: بیشترین مقدار دامنه 100 می باشد.
step	این فاکتور مقیاس گام اسلایدر می باشد: مقدار پیش فرض یک است و فقط اعداد صحیح مجاز می باشند، مگر اینکه کمترین مقدار عدد غیر صحیح باشد.
list	لیست همان دیتالیست در فصل قبلی می باشد. دیتالیست می تواند با نوع دامنه ی ورودی یکی شود، به هرحال هیچ یک از مرورگرها این ویژگی را در نگارش اجرا نکرده اند.

همیشه درک این مطالب در نمایش آسانتر می باشد.

اجازه دهید بیشترین مقدار، کمترین مقدار و گام فاکتور را مشخص کنیم.

```
<input id="slider1" type="range" min="100" max="500" step="10" />
```

تنظیم مقدار پیش فرض

```
<input id="slider2" type="range" min="100" max="500" step="50" value="100" />
```

کار آن با اعداد ممیزی

```
<input id="slider3" type="range" min="-2.5" max="3.0" step="0.1" />
```

آموزش Input Type : Number

در **HTML5** می توانید عددی را به عنوان **Spinner** در فیلد ورودی داشته باشد که پیکان هایی به سمت بالا و پایی در قسمت راست تکست باکس وجود دارند که برای افزایش یا کاهش این عدد به کار می روند. لطفا گالری اسپینر را در زیر ببینید.

در امر نگارش مرورگرهایی که در ویندوز این کار را انجام می دهند اپرا و گوگل کروم می باشند.

Browsers	Render "Number" input as Spinner
IE 9 Beta	

Firefox 13	
Safari 5	✓ (in Mac OS, but not Windows)
Chrome 8	✓
Opera 11	✓

<label for="movie">How often you watch movie in week : </label>

<input id="movie" type="number" value="0" />

اگر شما از گوگل کروم 8 استفاده می کنید یک مقدار غیر عددی مانند a ، b و یا e را امتحان کنید، کروم به شما اجازه چنین کاری نخواهد داد اما اجازه گذاشتن e را می دهد، چرا چنین است؟! زیرا e می تواند بخشی از یک عدد ممیزی شناور باشد، برای مثال 5.34×10^{12} .

از طرف دیگر اپرا به شما اجازه می دهد هرچه می خواهید در فیلد وارد کنید.

ما باید برخی ویژگی هایی را که می توانیم در نوع عدد ورودی داشته باشیم کشف کنیم.

Attributes	Descriptions
value	Value مقدار پیش فرض جعبه ی ورودی است وقتی که یک صفحه بارگذاری می شود. این ویژگی متداول برای عنصر ورودی می باشد، بدون توجه به نوعی که استفاده می کنید.
min	کمترین مقدار شما از عدد: من باید کمترین مقدار را 0 تعیین کنم چرا که عدد منفی در تعداد دفعات تماشای فیلم در یک هفته بی معناست.
max	واضح است که بزرگترین عدد قسمت ورودی را مشخص می کند.
step	فاکتور مقیاس گام: اگر این ویژگی مشخص نشده باشد مقدار پیش فرض 1 می باشد.

اگر شما یک شرکت نرم افزار هستید و نرم افزار شما توسط تعدادی مجوز کاربر فروخته شده است، حداقل مجوزهایی که فروخته اید 5 می باشد و این عدد 5 تا 5 تا افزایش می یابد و بیشترین مجوزی که می توانید به هر مشتری بفروشید 30 است.

Number of user license

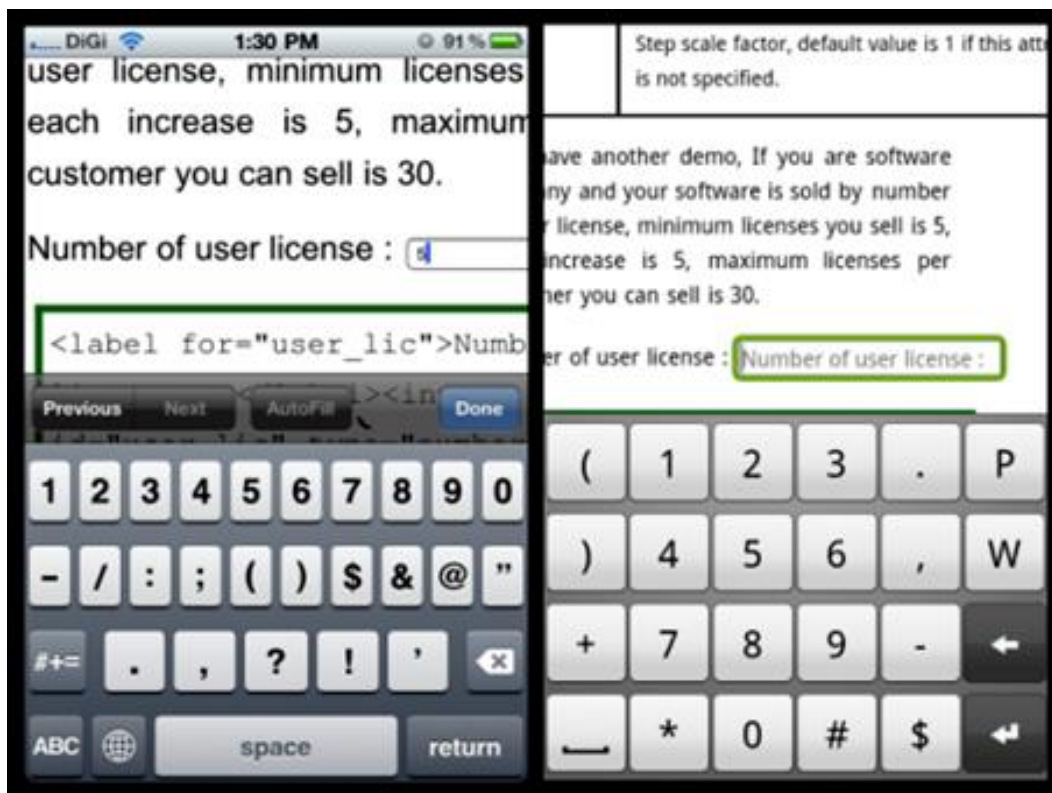
```
<label for="user_lic">Number of user license : </label>
<input id="user_lic" type="number" min="5" max="30" step="5" value="5" />
```

اگر برای افزایش و یا کاهش مقدار از جاوااسکریپت استفاده کردید، راه حل آن اینجاست.

- `spinner.stepUp(x)` - مقداری که می خواهید در فیلد افزایش دهید.
- `spinner.stepDown(x)` - مقداری که تمایل دارید در فیلد کاهش دهید.
- `spinner.valueAsNumber` - مقدار ورودی را به جای یک رشته به عنوان عدد ممیزی شناور گزارش می دهد.

یک **Spinner** با پیکان های نسبتاً کوچک رو به بالا و پایین ممکن است زیاد برای لمس توسط انگشتان برای یک عملکرد لمسی مناسب نباشند. مرورگرهای آیفون و اندروید 2.2 هر دو `<input type="number">` را به عنوان یک تکست باکس نرمال اجرا می کنند اما به کاربرها یک صفحه کلید اعداد برای وارد کردن داده ها ارائه می دهند.

صفحه ی آیفون در سمت چپ و صفحه ی **HTC Desire (android 2.2)** در سمت راست ارائه شده است.



نوع ورودی: تاریخ

فیلد تاریخ و زمان به راحتی به شکل های مختلف وب پیدا می شوند. برنامه های نمونه مانند رزرو بلیط، تعیین وقت ملاقات، سفارش پیتزا و غیره.

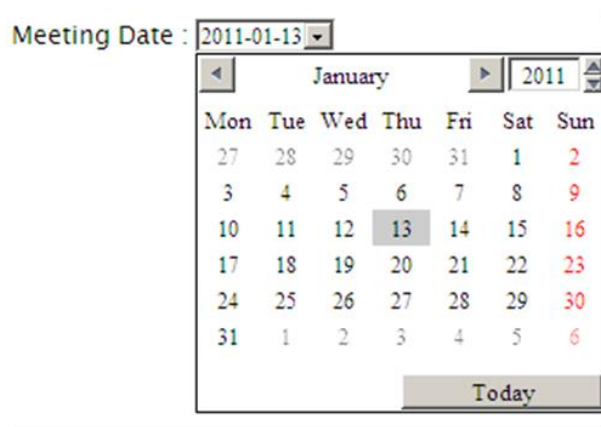
متداول ترین راه استفاده برای ورودی تاریخ استفاده از چپننده ی تاریخ جاوااسکریپت می باشد. باور می کنید؟ فقط چپننده ی تاریخ جاوااسکریپت گوگل. بیشتر چپننده های تاریخ از یک تقویم استفاده می کنند تا به کاربر اجازه انتخاب تاریخ مورد نظر را داده و آن را در تکست باکس مربوط به تاریخ قرار دهد.

در نگارش تنها مرورگری که کاملاً ورودی تاریخ و زمان را پشتیبانی می کند اپرا (v11) و گوگل کروم (v20) می باشند. در **HTML5** کار مرورگر این است که به کاربر اطمینان دهد که می تواند تنها یک رشته تاریخ معتبر در تکست باکس ورودی قرار دهد.

انتخاب یک تاریخ از تقویم تنها راه وارد کردن تاریخ نیست، گرچه محبوبترین راه اجرا می باشد. مشخصات **HTML5** چیزی در مورد نمایش یک تقویم برای ورودی تاریخ ذکر نکرده است.

Browsers	Date Time Support
IE 9 Beta	
Firefox 13	
Safari 5	
Chrome 20	✓
Opera 11	✓

اگر روی کامپیوتر خود اپرا ورژن 11 و گوگل کروم 20 (و البته ورژن های بالاتر دارید) در مورد چک کردن آن راحت خواهید بود. اگر مرورگری ندارید که آن را پشتیبانی کند، تصویر زیر آن چیزی است که به شما نشان داده خواهد شد.



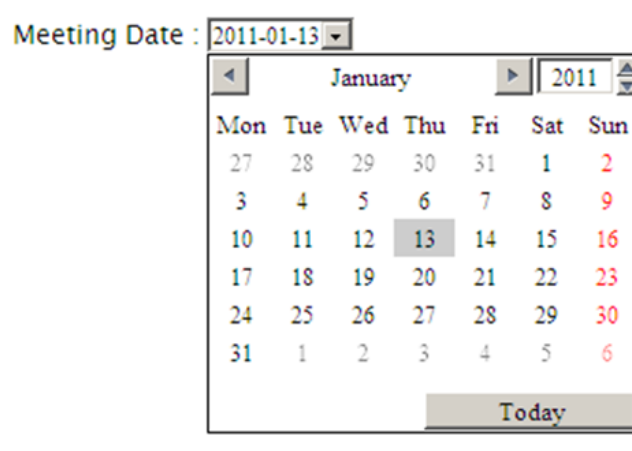
با پشتیبانی HTML5 طراح وب برای داده ی پایه ی تاریخ نیازی به دانلود کنترل جاوااسکریپت ندارد. دقیقاً آنچه لازم است انجام دهید `<input type="date" />` می باشد.

```
<label for="meeting">Meeting Date : </label>
<input id="meeting" type="date" value="2011-01-13" />
```

یک بار دیگر **HTML5** زندگی ما را ساده تر کرده است. شما فقط داده ی تاریخ وارد نمی کنید، داده های تاریخ و زمان در ارتباط با هم هستند که می توانید با هم وارد کنید. البته از زمانی که شرکت های مرورگر تصمیم گرفتند این استاندارد را اجرا کنند، توپ در زمین آنهاست.

تاریخ (<input type="date" />)

واضح است که این مرورگر داخلی چپننده ی تاریخ می باشد. شما می توانید تنها یک تاریخ خاص را از تقویم انتخاب کنید.



هفته (<input type="week" />)

به جای انتخاب تاریخ می توانید یک هفته را انتخاب کنید. لطفا به شماره هفته در سمت چپ تقویم توجه کنید.

2011-W03 ▾

◀

March

▶

2011 ▴ ▾

Week	Mon	Tue	Wed	Thu	Fri	Sat	Sun
9	28	1	2	3	4	5	6
10	7	8	9	10	11	12	13
11	14	15	16	17	18	19	20
12	21	22	23	24	25	26	27
13	28	29	30	31	1	2	3
14	4	5	6	7	8	9	10

Today

3. شما حتی می توانید چیننده ی ماه هم داشته باشد. در اینجا تقویمی را مشاهده می کند که به شما اجازه می دهد یک ماه در سال را انتخاب کنید.

2010-12 ▾

◀

December

▶

2010 ▴ ▾

Mon	Tue	Wed	Thu	Fri	Sat	Sun
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

Today

زمان (<input type="time" />)

یک چیننده ی زمان برای انتخاب ورودی زمان می باشد.

تاریخ و زمان (<input type="date and time" />)

می توانید تاریخ و زمان را با زمان محلی انتخاب کنید. مقدار در زمان UTC نمایش داده می شود.

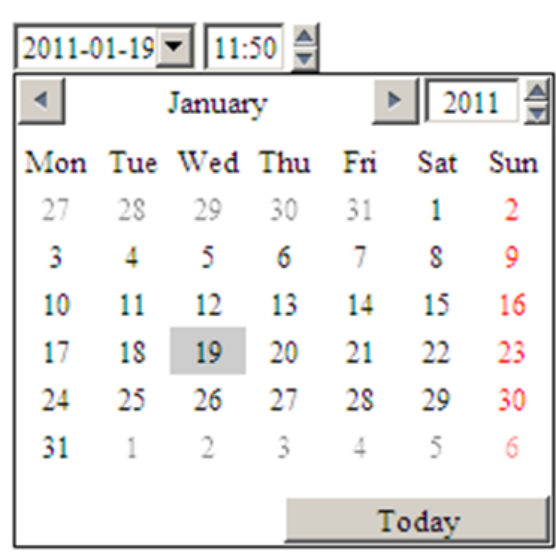
UTC

January 2011						
Mon	Tue	Wed	Thu	Fri	Sat	Sun
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

Today

زمان و تاریخ محلی (<input type="datetime-local" />)

در مقایسه با زمان UTC مقدار ورودی زمان تاریخ را با زمان محلی نشان می دهد.



اجازه بدهید به ویژگی های مرتبط که احتمالاً استفاده می کنیم، نگاهی داشته باشیم.

Attributes	Descriptions
value	Value مقدار پیش فرض باکس ورودی می باشد زمانی که در ابتدا بارگذاری می شود. این یک ویژگی متداول برای عنصر <code><input></code> ، بدون توجه به نوع مورد استفاده ی شما، می باشد.
min	کمترین تاریخ یا زمان
max	بیشترین تاریخ یا زمان
step	<p>فاکتور مقیاس گام: انواع مختلف ورودی مقدار گام پیش فرض خود را دارد.</p> <p>§ تاریخ – پیش فرض یک روز می باشد.</p> <p>§ هفته – پیش فرض یک هفته می باشد.</p> <p>§ ماه – پیش فرض یک ماه می باشد.</p> <p>§ تاریخ زمان محلی – پیش فرض همچنین یک دقیقه می باشد.</p>

آموزش Color

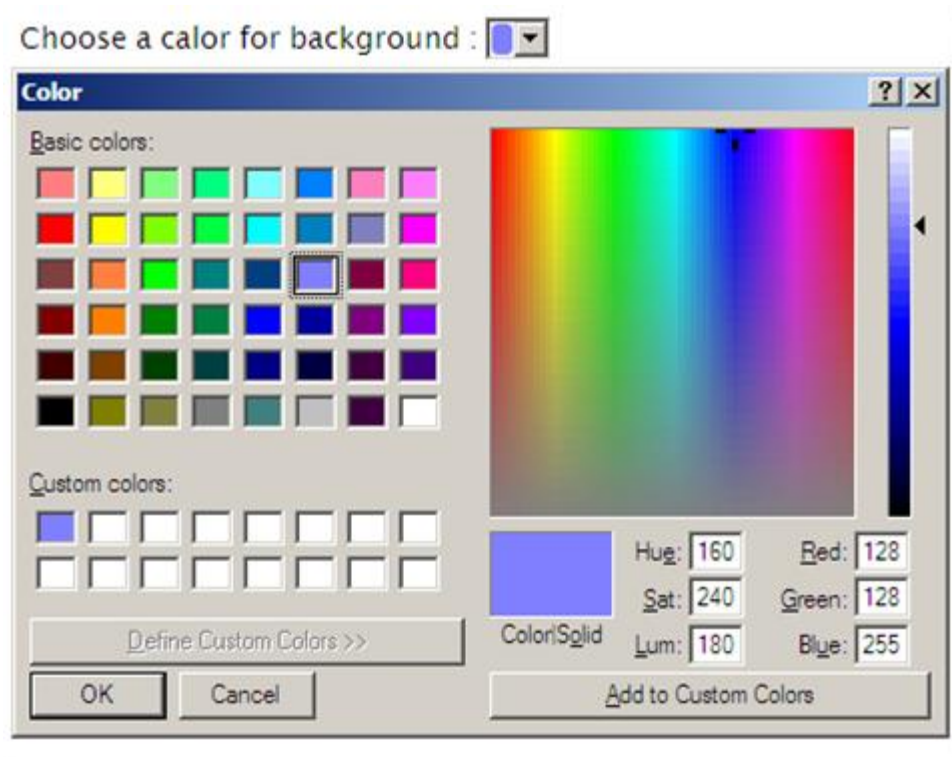
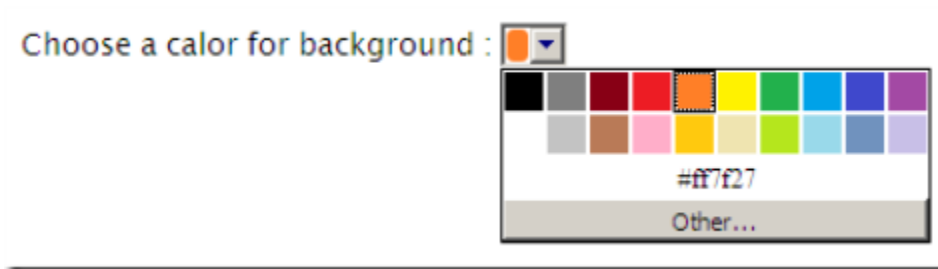
در HTML5 می توانیم به سادگی ورودی رنگ را با `<input type="color">` داشته باشیم. تکست باکس فقط باید مقدار رشته ی به اصطلاح **simple color** (رنگ ساده) را با حروف کوچک داشته باشد، به عنوان مثال **#ff0000** برای رنگ قرمز، **#000000** برای رنگ سیاه و **ffffff** برای رنگ سفید.

در نگارش تنها مرورگری که ورودی رنگ را پشتیبانی می کند، اپرا 11 می باشد.

Browsers	Color input Support
IE 9 Beta	
Firefox 13	
Safari 5	
Chrome 20	✓
Opera 11	✓

با نوع ورودی رنگ دیگر نیازی به چیننده ی رنگ پیچیده ی جاوااسکریپت ندارید. یک خط ساده از کد مانند زیر کار را انجام خواهد داد.

```
<label for="background-color">Choose a color for background :</label>
<input id="background-color" type="color" />
```



اگر به اندازه ی کافی خوش شانس باشید تا مرورگری داشته باشید که ورودی رنگ را پشتیبانی کند، لطفا در امتحان آن راحت باشید.

Choose a color for background

You have chosen :

به سمت وب معنایی

زمانی (در سال 1990) مردی به نام تیم برنرس لی دنیای پهناور وب را به وجود آورد که امروز همه ی ما آن را دوست داریم. اشتباه نکنید، او اینترنت را اختراع نکرد، اما **HTML** و پروتکل **HTTP** را چرا. سال ها بعد او کنسرسیوم **WWW** را با نام مستعار **W3C** تاسیس کرد، سازمانی که توسعه ی وب و برقراری استاندارد **HTML** و چند تکنولوژی مربوط دیگر را نظارت می کرد. او تا به امروز هنوز مدیر **W3C** می باشد.

بیشتر انسان های بزرگ با رویاهایشان حرکت می کنند، آقای تیم برنرس لی هم از این امر مستثنا نیستند. او هم رویای بزرگی دارد، که در اینجا به شما ارائه می شود.

من رویایی برای وب دارم که دو بخش است

در بخش اول وب ابزاری قوی تر برای همکاری بین انسان ها میشود. من همیشه فضای اطلاعات را جایی تصور کرده ام که هرکسی به آن دسترسی سریع و شهودی دارد، که نه فقط آنها را فهرست کند بلکه ایجاد هم بکند. نخستین برنامه ی **worldwideweb** تقریباً با یک صفحه ی خالی باز شد که برای یادداشت های کاربر آماده بود. من و رابرت کایلو زمان زیادی را صرف آن کردیم، نه به خاطر اینکه به دنبال مطالب زیادی بودیم، بلکه به این خاطر که ایده هایمان را می نوشتیم و به اشتراک می گذاشتیم. علاوه بر این رویای ارتباط بین انسان ها از طریق دانش مشترک باید برای همه ی گروه ها ممکن باشد، یک تعامل الکترونیکی آسان همانطور که اکنون با افراد به طور مستقیم دارند.

در بخش دوم رویا، همکاری ها با کامپیوترها توسعه پیدا می کند. ماشین ها قادر به تجزیه ی همه ی داده های وب می شوند، محتوا، لینک ها و داد و ستد بین مردم و کامپیوترها. یک وب معنایی (**semantic web**) که باید این امر را ممکن سازد هنوز باید ظهور کند، اما وقتی این اتفاق افتاد روز به روز مکانیزم های تجارت، ادارات و زندگی روزانه ی ما توسط ماشین ها انجام می شود، صحبت کردن با ماشین ها و رها کردن انسان ها برای ارائه ی الهام و شهود. آنچه عوامل هوشمند برای سالها تعریف کردند سرانجام ظاهر می شود. این ماشین قابل فهم وب از طریق اجرای مجموعه ای از پیشرفت های تکنیکی و توافق های اجتماعی که اکنون در حال آغاز هستند، اتفاق می افتد.

زمانی که رویا عملی شود، وب مکانی خواهد بود که در آن هوس یک انسان با استدلال یک ماشین در همنشینی با یکدیگر یک ترکیب ایده آل و قدرتمند را به وجود می آورند. برای عملی کردن یک رویا کارهای زیادی لازم است. وب یک رویای دور است و فقط یک وضعیت چند بخشی از یک ساختار می باشد که چگونگی رسیدن به رویا در آن مهم نیست و باید قطعه به قطعه مهندسی شود.

خلاصه اینکه هدف وب معنایی ساخت یک وب قابل فهم نه تنها توسط انسان بلکه توسط ماشین، میباشد (نه ماشینی با هوش مصنوعی). با آن تبادل و اشتراک اطلاعات خیلی موثرتر خواهد بود.

HTML5 مشتق از عناصر درون خطی جدید را (**abbr**، **strong**، **span**) به خانواده ی **HTML** معرفی می کند، که در واقع دیگر عنصر درون خطی نامیده نمی شود بلکه معنا شناختی در سطح متن نامیده می شود.

معنا شناسی جدید در سطح متن

§ **<mark>**
 § **<time>**
 § **<meter>**
 § **<progress>**

علاوه بر عناصر در سطح متن، مجموعه ای از عناصر برای انسداد و عناصری برای تقسیم بندی محتوا وجود دارند. با افزودن این عناصر جدید، هدف ساختن یک صفحه ی جدید پرمعنا تر می باشد، به جای غرق شدن با گروه های مختلف از قبیل **<div class="header">....<div class="footer">** یا **<div class="article">**.

عناصر انسداد جدید

§ **<section>**
 § **<header>**
 § **<footer>**
 § **<nav>**
 § **<article>**
 § **<aside>**

آموزش Semantic mark

علامت عنصر **<mark>** اجرای متنی را در یک داکيومنت نشان می دهد که به هدف مرجع و با توجه به ارتباط آن در یک کانتکست دیگر مشخص شده است.

اساسا برای معطوف کردن توجه خواننده به بخشی از متن استفاده می شود که خواننده ممکن است آن بخش را نادیده بگیرد.

برای علامت دار کردن بخشی از متن آن را بین **<mark></mark>** قرار دهید.

در نگارش هر مرورگری **<mark></mark>** را برای **highlight** متن اجرا نمی کند، برای اطمینان خاطر بیشتر بهتر است عنصر **mark** را در **CSS** نیز طراحی کنید.

```
mark { background-color:#ff9; color:#000; font-style:normal; font-weight:bold; }
```

آموزش Semantic time

Semantic **<time>**

عنصر زمان یا یک ساعت را در **24** ساعت نشان می دهد و یا یک تاریخ مشخص را در تقویم مسیحی (**Gregorian calendar**) با یک زمان و افسست منطقه ی زمانی نشان می دهد.

برای نمونه شما می توانید این کار را انجام دهید تا یک تاریخ مشخص را نشان دهید.

We were celebrating our daughter's **<time datetime="2004-07-13">birthday</time>** in that seafood restaurant.

ما تولد **<time datetime="2004-07-13">دخترمان</time>** را در آن رستوران دریایی جشن می گیریم.

از آنجایی که در مثال زیر زمان در فرمت **24** ساعتی به کاربر نمایش داده شده است، ما ویژگی **datetime** را برای بیشتر مشخص کردن مقدار زمان نیاز نداریم.

I usually go swimming at **<time>20:00</time>** every Saturday.

من معمولا در ساعت **<time>20:00</time>** هر شنبه به شنا می روم.

جمله ی بالا با استفاده از تکنولوژی طراحی شده که طراحی دوباره ی زمان ها را پشتیبانی می کند، می تواند به این شکل بیان شود.

من معمولا در ساعت 8 بعد از ظهر هر شنبه به شنا می روم.

یا

من معمولا در ساعت 20:00 هر شنبه به شنا می روم.

از یک طرف اگر مقدار زمان در فرمت 24 ساعته مشخص نشده باشد باید ساعت آن را در 24 ساعت مشخص کنید.

ما از روز شنبه تا دوشنبه از ساعت `<time datetime="08:00">8am</time>` تا ساعت `<time datetime="18:00">6pm</time>` آزاد هستیم.

برای یک زمان و تاریخ مشخص چگونگی کار در اینجا نشان داده شده است.

شما ممکن است کتاب های خود را در `<time datetime="2011-01-16T06:00Z">` 16 ژانویه ی 2011 در `<time>6pm` جمع کنید.

یه طور مشابه شاید لازم باشد یک تاریخ و زمان درست را با منطقه ی زمانی نیز مشخص کنید.

سمینار در `<time datetime="2011-02-22T09:00-4:00">` 22 فوریه ی 2011 در `<time>9amEDT` شروع خواهد شد.

مشخصه ی **HTML5** به ما می گوید که اگر تاریخ و زمان دقیق معلوم نیستند از عنصر زمان استفاده نکنیم. به عنوان مثال، من میدونم شما در تابستان گذشته چه کاری انجام دادید. یا او را از 1980 می شناختم.

ممکن است از خود بپرسید با استفاده از عنصر زمان در انتهای روز چه حاصل می شود. بسیار خوب با همه ی این تلاش ها سعی می کنیم محتوای ماشین خود را قابل خواندن کنیم. تصور کنید روزی بتوانیم **"February 22, 2011 9am EDT"** تنها با یک کلیک را در تقویم خود ذخیره کنیم.

علاوه بر ویژگی **datetime** عنصر زمان دارای ویژگی های **"pubdate"** و **"valueAsDate"** می باشد.

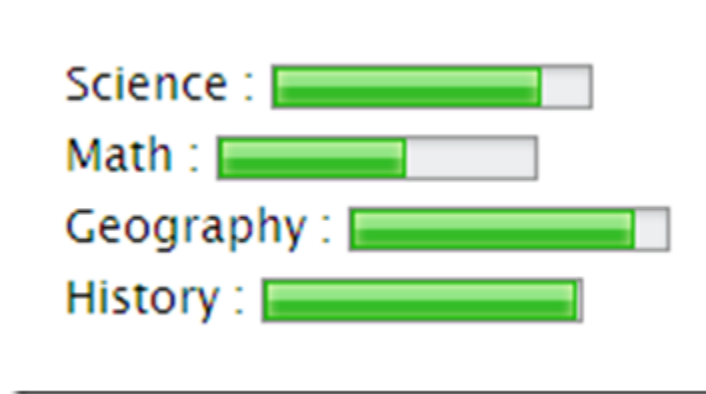
Attributes	Descriptions
dateTime	مقدار زمان یا تاریخ و یا زمان تاریخ. اگر وجود داشته باشد، تاریخ یا زمان را مشخص می کند. در غیر اینصورت تاریخ یا زمان توسط عناصر محتوا ارائه می شوند.
pubDate	تاریخ انتشار. یک ویژگی Boolean می باشد که به این معناست که لازم نیست مقداری برای آن قرار دهید. اگر مشخص باشد که یعنی تاریخ انتشار یک مقاله (<article>) ، اگر عنصر زمان در داخل بلوک مقاله نباشد به این معناست که این تاریخ، تاریخ انتشار کل پرونده می باشد. نکته: شما باید فقط یک pubdate در یک مقاله و یا یک داکيومنت داشته باشید.
valueAsDate	یک ویژگی readonly می باشد، و آبجکت تاریخ را که مشخص کننده ی تاریخ و زمان خاصی است، گزارش می دهد.

Pubdate می تواند برای ورودی وبلاگ و یا اخبار مقاله به کار رود. برای مقاله ای که در **21 ژانویه ی 2011** منتشر شده، کد آن مانند زیر می باشد.

`<time datetime="2011-01-21" pubdate>January 21 2011</time>`

آموزش Semantic meter

Meter یک عنصر جدید در **HTML5** می باشد که مقدار یک دامنه ی شناخته شده را به عنوان وسیله ی اندازه گیری نشان می دهد. کلمه ی کلیدی در اینجا دامنه ی دامنه ی شناخته شده می باشد، که بدین معناست که شما فقط زمانی اجازه ی استفاده از آن را دارید که کاملاً از مقدار حداقل و مقدار حداکثر آگاه باشید.



علاوه بر این نباید از این مقیاس اندازه گیری برای نشان دادن یک پیشروی، مثلاً پیشروی 80 درصد استفاده کنید زیرا یک عنصر پیشروی در HTML5 وجود دارد.

در نگارش فقط گوگل کروم 8 و اپرا 11 عنصر meter را در یک مقیاس اندازه گیری قابل رویت اجرا می کنند. بنابراین مشخص کردن مقدار متنی بین برچسب های meter مفید می باشد، و مرورگرها meter را به عنوان یک مقیاس اندازه گیری که بتواند به سادگی مقدار را در یک متن ساده نشان دهد، اجرا نمی کنند.

بدون تلاش شما برای مشخص کردن مقدار متنی بین برچسب های meter، کاربر تنها یک فضای خالی در مرورگرها خواهد دید که meter را پشتیبانی نمی کنند.

Browsers	Render meter element as gauge
IE 9 Beta	
Firefox 13	
Safari 5	
Chrome 8	✓
Opera 11	✓

علاوه بر ویژگی های **min** و **max** و **value** ویژگی های دیگری وجود دارند که می توانید برای اجرای وسیله ی اندازه گیری خود استفاده کنید.

Attributes	Descriptions
value	نیاز است که برای استفاده از عنصر meter مقدار را به عدد مشخص کنید. این یک امر لازم می باشد. عدد می تواند یک عدد صحیح یا ممیزی باشد.
min	حداقل مقدار یک meter، اگر مشخص نباشد، 0 خواهد بود.
max	بیشترین مقدار یک meter، اگر مشخص نباشد، 1 خواهد بود.
low	این گزینه انتخابی است، مگر اینکه شما تمایل داشته باشید مقدار پایین دامنه را نشان دهید.
high	این گزینه انتخابی است، مگر اینکه شما تمایل داشته باشید مقدار بالای دامنه را تعریف کنید.
optimum	این گزینه کاملاً انتخابی است و برای مشخص کردن یک نقطه ی مطلوب در یک دامنه به کار می رود.

آموزش Semantic progress

عنصر جدید پیشروی به نظر می رسد شبیه به عنصر **meter** می باشد و برای نشان دادن پیشروی یک عمل ایجاد شده است. پیشروی می تواند مشخص باشد یا نا مشخص. یعنی اینکه شما می توانید از عنصر پیشروی برای نشان دادن یک پیشروی استفاده کنید که حتی نمی دانید چقدر از کار مربوط به آن باقی مانده است.

گیج کننده به نظر می رسد، چطور می توانید بدون اینکه بدانید پایان کجاست، میزان پیشروی را بگویید.

اولین امکان، کار هنوز حتی شروع هم نشده، بنابراین یا 0 درصد است یا 0.0001 درصد که اصلاً مطرح نیست.

دومین امکان اینکه، مقدار حقیقی پیشروی از طرف برخی سرورها یا فرایندهای آپدیت کردن آن در انتظار است.

یکی از استفاده های عنصر پیشروی، نشان دادن پیشروی فایل دانلود می باشد.

```
<section>
<p>Progress: <progress id="p" max=100<span>0</span>%</progress></p>
<script>
```

```
var progressBar = document.getElementById('p');
function updateProgress(newValue) {
    progressBar.value = newValue;
    progressBar.getElementsByTagName('span')[0].textContent = newValue;
} </script>
</section>
```

برای مشخص کردن پیشروی، همیشه می توانید مقدار پیشروی را در قسمت **value** تنظیم کنید. یک ویژگی **max** وجود دارد که می توانید حداکثر مقدار پیشروی را نشان دهید. در مقایسه با عنصر **meter**، ویژگی **min** وجود ندارد، چرا که همیشه حداقل میزان پیشروی **0** می باشد.

Progress of Task A : `<progress value="60" max="100">60%</progress>`

در نگارش نیز فقط اپرا 11 و گوگل کروم 8 عنصر پیشروی را در قالب یک نوار پیشروی پشتیبانی می کنند.

Browsers	Progress element support
IE 9 Beta	
Firefox 13	✓
Safari 5	
Chrome 8	✓
Opera 11	✓

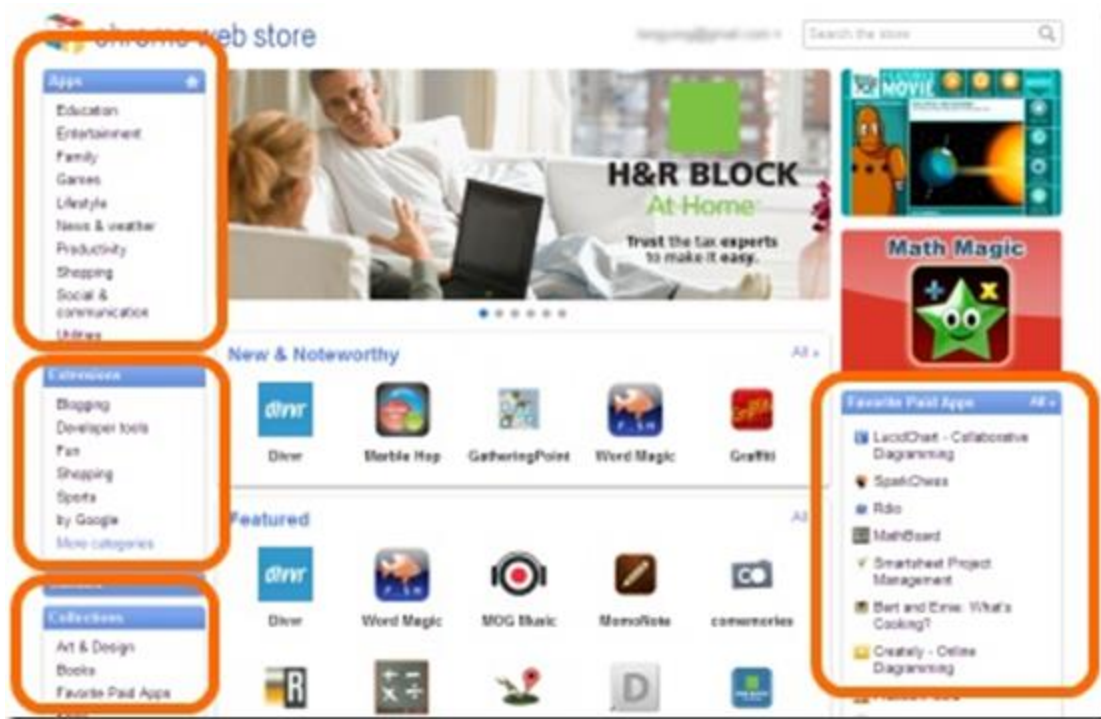
اگر از مرورگری که پیشروی را پشتیبانی کند، استفاده نمی کنید، تصویر زیر نشان می دهد که پیشروی در گوگل کروم 8 چگونه به نظر می رسد.

Progress of Task A : 

آموزش Semantic section

ویژگی های خواننده های W3C - عنصر بخش (section) یک بخش کلی از یک داکيومنت یا یک برنامه را مشخص می کند. در این متن، بخش گروه بندی معنایی محتوا با یک تیتیر می باشد.

در تصویر زیر قسمت های مشخص شده بهترین استفاده برای <section> را نشان می دهند.



توجه داشته باشید که عنصر section در اینجا برای جایگزین کردن عنصر div به کار نمی رود.

شما نباید از عنصر بخش برای موارد زیر استفاده کنید.

1. برای یک محتوای منسجم: شما نباید از بخش برای محتوا در RSS feed استفاده کنید. اگر در حال ایجاد وب سایت وبلاگ می باشید از عنصر section برای ورودی وبلاگ خود استفاده نکنید، بلکه باید از عنصر Article استفاده کنید. به هر حال یک عنصر Article می تواند دارای عناصر section نیز باشد.
2. صرفا به هدف Layout : از عنصر <div> به عنوان ظرف عمومی به هدف طراحی استفاده کنید.

```
<section>
  <h1>Application</h1>
```

```
</ul>
<li>Education</li>
<li>Entertainment</li>
<li>Family</li>
<li>Games</li>
<li>News and Weather</li>
</ul>
</section>
```

شاید متوجه شده باشید که من از **<h1>** برای عنوان در عنصر بخش استفاده میکنم. عنصر بخش در واقع نوعی تقسیم بندی محتوا در **HTML5** می باشد.

تقسیم بندی محتوا

- همیشه زیرمجموعه ای از نزدیک ترین پیش گام آن مانند **<body>**، **<article>** یا یک عنصر **<section>** دیگر می باشد.
- شما به استفاده از h1 تا h6 در داخل عنصر بخش تشویق می شوید، بدون اینکه در مورد آنچه در بیرون بخش اتفاق افتاده اذیت شوید.
- می توانند **<header>**، **<hgroup>** و **<footer>** خود را داشته باشند.

بنابراین اگر آن را مانند زیر بنویسید بسیار خوب است. درواقع شما برای انجام چنین کاری بسیار تشویق می شوید.

```
<body>
<h1>Apples</h1>
<p>Apples are fruit.</p>
<section>
<h1>Taste</h1>
<p>They taste lovely.</p>
<section>
<h1>Sweet</h1>
<p>Red apples are sweeter than green ones.</p>
</section>
</section>
<section>
<h1>Color</h1>
<p>Apples come in various colors.</p>
</section>
</body>
```

اگر شما در حال طراحی داکيومنت HTML5 می باشید، پیشنهاد می کنم HTML5 Outliner را مشاهده کنید، که اگر Outline شما درست نباشد به شما می گوید.

آموزش Semantic header

تقریباً همه ی وب سایت های مدرن یک بخش تیترا دارند که در بالای هر صفحه در داخل وب سایت نشان داده می شود. عنصر جدید header ابزاری برای آن می باشد.



در واقع آنچه گفتم یک شوخی بود، زیرا header ابزاری بیشتر از آن می باشد. ما در استفاده از header برای به اصطلاح تیترا اصلی یک وب سایت تشویق می شویم. به هر حال منظور من این نیست که New York Times از عنصر header استفاده کرد، بلکه منظورم این است که New York Times می تواند از این عنصر استفاده کند.

```
<body>
<header>
<h1>Little Green Guys With Guns</h1>
<nav>
<ul>
<li>
<a href="/games">Games</a>
<li>
<a href="/forum">Forum</a>
```

```

</li>
<a href="/download">Download</a>
</ul>
</nav>
</header>
<p>You have three active games:</p>
<!-- this is still part of the subsection entitled "Games" -->
...

```

به هرحال نباید تصور کنیم که **header** فقط برای تیترهای اصلی وب سایت ها به کار می رود. بلکه **header** می تواند به عنوان یک تیتر برای ورودی یک وبلاگ یا یک مقاله ی خبری استفاده شود، همان طور که هر مقاله ای تیتر خود را دارد و هر زمان و تاریخ منتشر می شود.

```

<article>
<header>
<h1>Military Offers Assurances to Egypt and Neighbors</h1>
Published : <time datetime="2011-02-13" pubdate>February 13 2011</time>
</header>
<p>
CAIRO - As a new era dawned in Egypt on Saturday,
the army leadership sought to reassure Egyptians and
the world that it would shepherd a transition to civilian
rule and honor international commitments like the peace treaty with Israel.
</p>
</article>

```



نه تنها این مورد، بلکه آنچه در بخش قبل در مورد **section** بیان کردیم که هر **section** احتمالا باید تیتراژ خود را داشته باشد.

```
<section>
  <header>
    <h1>Application</h1>
    <h2>Paid iphone apps</h2>
  </hgroup>
</header>
<ul>
  <li>Education</li>
  <li>Entertainment</li>
  <li>Family</li>
  <li>Games</li>
  <li>News and Weather</li>
</ul>
</section>
```

عنصر **header** معمولا شامل عناصر دیگری هم می شود (مانند **h1** و **h2** و **hgroup**) که به هر حال ضروری نیست، گرچه متداول تر تمرین شده است.

مشابه آنچه در قسمت **section** یاد گرفتیم، **header** هم دارای **Outline** مربوط به خود می باشد. شما می توانید به راحتی **h1**، **h2** و **h6** خود را تعریف کنید بدون اینکه نگران آنچه در مورد قبلی ها تعریف شده، باشید.

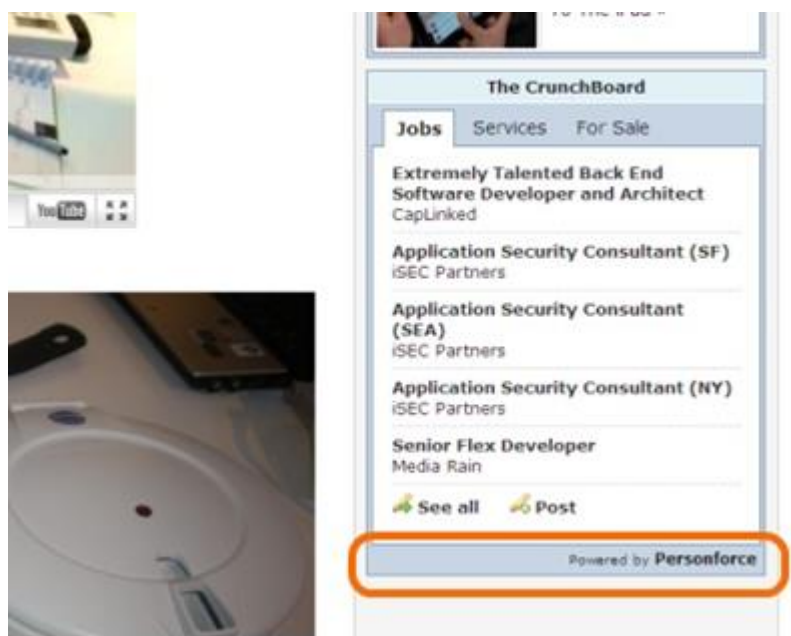
آموزش Semantic footer

مشابه **header**، عنصر **footer** نیز به پاورقی یک صفحه ی وب اشاره می کند. اکثر اوقات **footer** می تواند به عنوان آنچه ما در مورد آن فکر کردیم، استفاده شود.

به طور کل **footer** شامل اطلاعاتی در مورد یک بخش است، اطلاعاتی از قبیل نویسنده ی متن، لینک های مربوط به داکيومنت، اطلاعات کپی رایت و غیر.



لطفا این تصور را نداشته باشید که برای هر وب فقط می توانید یک **footer** داشته باشید. بلکه برای هر بخش یا هر مقاله می توانید یک **footer** داشته باشید.



لطفا به مثال زیر دقت داشته باشید که در آن دو مقاله دارای **footer** هستند و یک عنصر **site wide footer** وجود دارد که به عنوان **footer** اصلی این داکيومنت محسوب می شود.

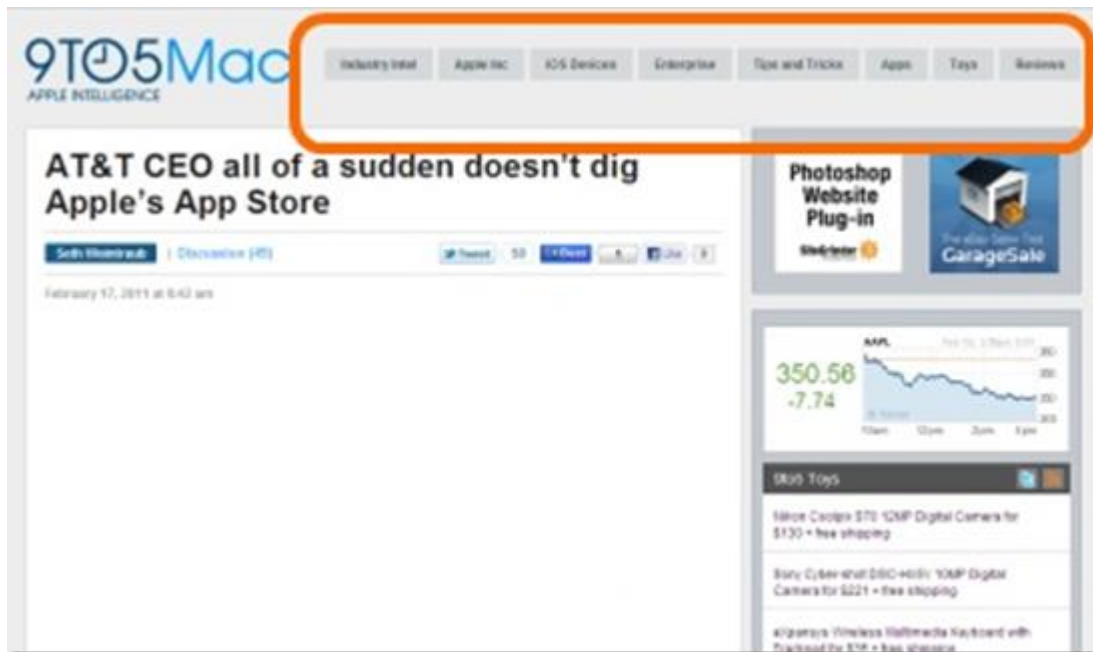
```
<!DOCTYPE html>
<html>
<head>
  <title>The Ramblings of a Scientist</title>
</head>
<body>
  <h1>The Ramblings of a Scientist</h1>
  <article>
    <h1>Episode 15</h1>
    <video src="/fm/015.ogv" controls preload>
      <p><a href="/fm/015.ogv">Download video</a>.</p>
    </video>
    <footer>
      <!-- footer for article -->
      <p>Published <time pubdate datetime="2009-10-21T18:26-07:00"></time></p>
    </footer>
  </article>
  <article>
    <h1>My Favorite Trains</h1>
    <p>I love my trains. My favorite train of all time is a Kof.</p>
    <p>
      It is fun to see them pull some coal cars because they look so
      dwarfed in comparison.
    </p>
    <footer>
      <!-- footer for article -->
      <p>Published <time pubdate datetime="2009-09-15T14:54-07:00"></time></p>
    </footer>
  </article>
  <footer>
    <!-- site wide footer -->
    <nav>
      <p>
        <a href="/credits.html">Credits</a>-
        <a href="/tos.html">Terms of Service</a> -
        <a href="/index.html">Blog Index</a>
      </p>
    </nav>
    <p>Copyright ©2015 Tahlildadeh</p>
  </footer>
</body>
</html>
```

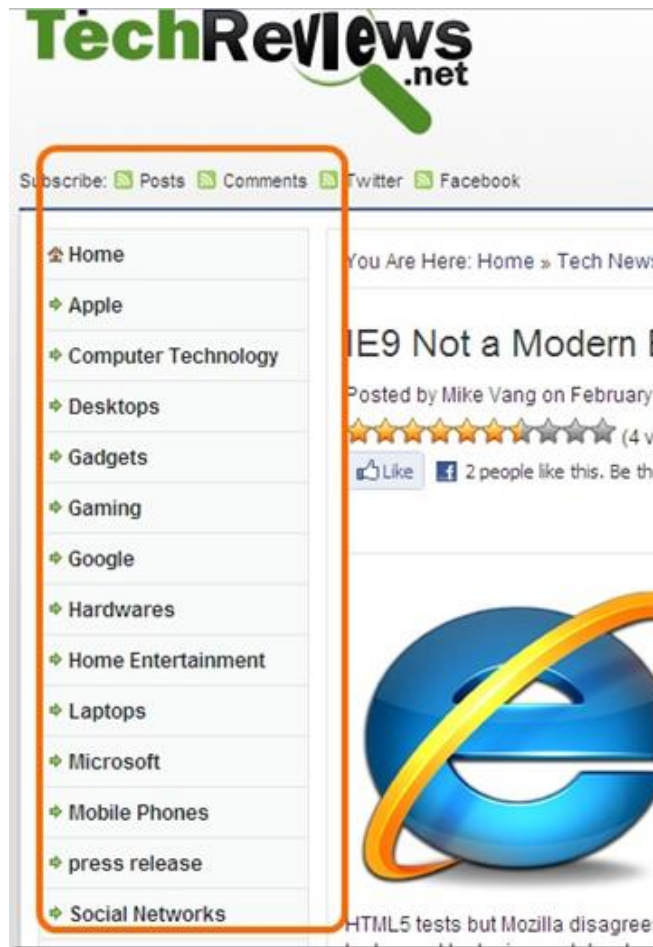
آموزش Semantic nav

همانطور که از نام آن پیداست، عنصر جدید **nav** برای مسیریابی استفاده می شود. عنصر **nav** وسیله ای است برای منطقه ای با لینک های سایت های مختلف.

لینک های مسیریابی یک وب سایت، اکثرا در بالا و یا نوارهای کناری سایت قرار گرفته اند و گاهی اوقات در قسمت **footer** قرار دارند.

به هرحال همه ی گروه های لینک ها در یک صفحه لازم نیست در عنصر **nav** قرار بگیرند – تنها قسمت هایی که دارای بلوک های مسیریابی مهمی هستند مناسب برای **عنصر nav** می باشند. به ویژه در مورد **footer** ها متداول است که لیست مختصری از لینک های صفحات مختلف یک سایت داشته باشند، از قبیل ترم های سرویس، صفحه ی اصلی و صفحه ی کپی رایت. عنصر **footer** به تنهایی و بدون **nav** کافیت.





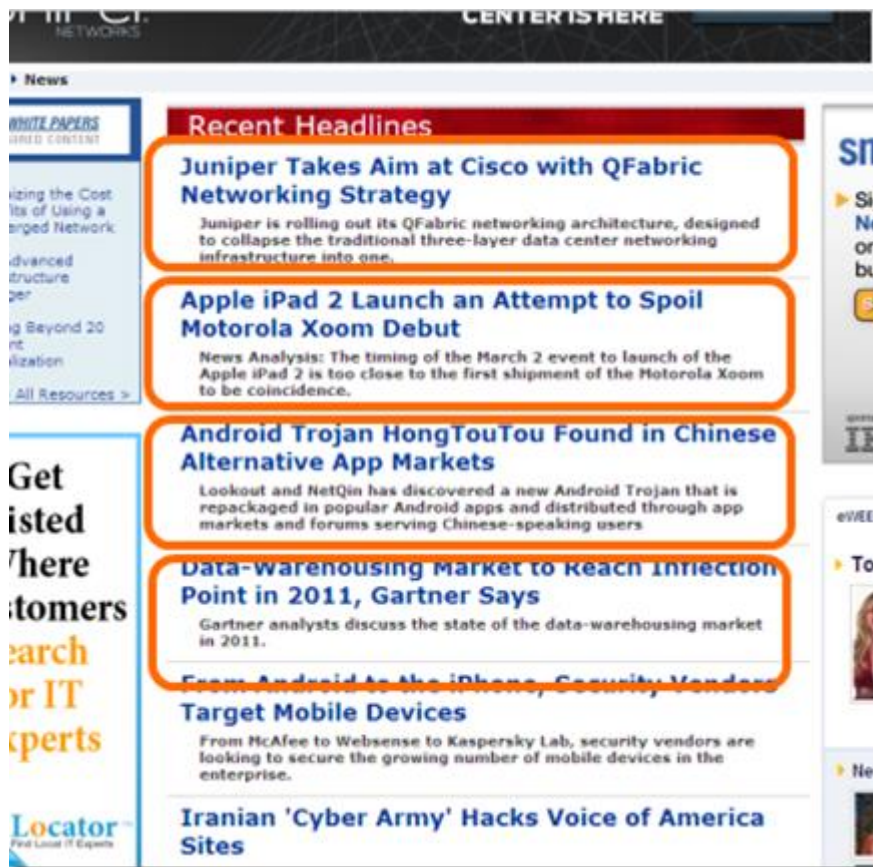
در زیر مثالی را می بینید که چگونگی استفاده از عنصر **nav** را توضیح می دهد، که معمولا با عنصر **ul** برای لینک های مسیریابی تودر تو شده است.

آموزش Semantic article

امروزه وب دارای اقیانوسی از مقالات خبری و مطالب وب می باشد. که به **W3C** دلیل خوبی برای تعریف عنصری برای مقاله به جای `<div class="article">` می دهد.

عنصر **article** وسیله ای برای مقاله ی وب می باشد. داشتن بیشتر از یک مقاله در یک صفحه ی وب بسیار متداول است.

آنچه در زیر می بینید یک صفحه ی وب می باشد که دارای بیشتر از یک مقاله می باشد. به هرحال وقتی که روی عنوان کلیک کنید صفحه ای با تنها یک مقاله باز خواهد شد.



ما باید از محتوای مقاله ای استفاده کنیم که فکر می کنیم قابل انتشار است. درست مانند اخبار و مطالب وب که می توانیم در **RSS feed** به اشتراک بگذاریم. یک **article** می تواند در یک **article** دیگر قرار بگیرد. مثالی از این نمونه کامنت های ارسال شده ی کاربر در مورد عنوان یک وبلاگ می باشد. عناصر داخلی **article** نشان می دهند که مقالاتی که در صدر هستند در ارتباط با محتوای مقالات بیرونی می باشند. به عنوان مثال تیتتر یک وبلاگ در یک سایت که پذیرنده ی کامنت های کاربرها می باشد به عنوان یک مقاله ی دیگر در داخل مقاله برای تیتتر وبلاگ قرار گرفته است.

منظور از یک مقاله فقط محتوای آن نمی باشد. شما می توانید در یک مقاله **header** و **footer** نیز داشته باشید. از آنجایی که هر مقاله باید یک تیتتر داشته باشد، داشتن عنصر **header** بسیار متداول است.

```
<body>
<h1>My blog</h1>
<article>
<header>
<h1>The Very First Rule of Life</h1>
<p><time pubdate datetime="2009-10-09T14:28-08:00"></time></p>
```

```

</header>
a<p>
  If there's a microphone anywhere near you, assume it's hot and
  sending whatever you're saying to the world. Seriously.
</p>
<p>...</p>
<footer>
  <a href="?comments=1">Show comments...</a>
</footer>
</article>
<body>

```

در HTML5 داشتن h1 تا h6 در یک article بسیار خوب است. از آنجایی که یک article تقسیم بندی محتوا می باشد، بدین معناست که یک مقاله زیرمجموعه ای از موارد قبل از خود بوده است.

در مورد ما article زیرمجموعه ای از یک بدنه می باشد و می تواند Outline خود را داشته باشد. با این توضیحات بهتر این است داکيومنت وب خود را حفظ کنیم، زیرا لازم نیست به Outline آن خارج از مقاله فکر کنید.

آموزش Semantic aside

عنصر aside بخشی است که تا حدی به محتوای اصلی مربوط می شود، اما می تواند از آن جدا شود. در این آموزش من از عنصر aside برای نکات مربوط به موضوع استفاده می کنم. آنچه در بخش زرد رنگ بیان می شود، اطلاعات اضافه ای به خواننده ارائه می دهد، اما خواننده می تواند بدون خواندن بخش زرد رنگ به خواندن مطلب ادامه دهد. به تصویر زیر توجه کنید.

A Date and time field can be easily found in many web forms. Typical applications are like ticket booking, appointment booking, ordering pizza and etc.

The most commonly used solution for date input is to use Javascript date picker. Don't believe me? Just google "Javascript date picker". Most of the date picker use a calendar to let user choose a date and fill the date into a textbox.

As of writing, the only web browser completely support date time input is Opera. In HTML5, it is the job of web browser to ensure user can only enter a valid date time string into the input textbox.

Picking a date from Calendar is not the only way to input a date value even though it's the most popular implementation. HTML5 specifications does not mention anything about displaying a calendar for date input.

<article>

<p>

As of writing, the only web browser completely support date time input is Opera.

In HTML5, it is the job of web browser to ensure user can only enter a valid date time string into the input textbox.

</p>

<aside>

Picking a date from Calendar is not the only way to input a date value even though it's the most popular implementation.

HTML5 specifications does not mention anything about displaying a calendar for date input.

</aside>

</article>

ویژگی های W3C

عنصر **aside** بخشی از یک صفحه را نشان می دهد که شامل محتوایی است که به طور سطحی مربوط به محتوای **aside** می باشد و همچنان می تواند از آن متن مجزا باشد. چنین بخش هایی معمولا در فن چاپ به شکل نوار کناری نشان داده می شوند.

این عنصر می تواند برای افکت هایی در فن چاپ استفاده شود، مثلا برای نقل قول ها، یا ستون های فرعی و برای تبلیغ و برای گروه های عناصر **nav** و برای محتوایی که از متن مجزا قرار می گیرند.

عناصری که نباید استفاده کنید

پس از معرفی عناصر و ویژگی های مختلف در بخش های گذشته، زمان آن رسیده تا در مورد آنچه باید در **HTML5** از آن پرهیز کنید، صحبت کنیم. از زمانی که **CSS** ابداع شد، توسعه دهندگان وب همیشه گفته اند که طرح باید از محتوا تفکیک شود. **CSS** از سبک ها و **HTML** از محتوا و ساختار صفحه ی شما محافظت می کند.

در **HTML5** دوران عناصر زیر به سر رسیده و نقشی در بازی **HTML5** ندارند، زیرا **CSS** همه ی آنها را گرفته است.

§ <basefont>
§ <big>
§ <center>
§
§ <strike>
§ <tt>
§ <u>

عناصر زیر در طراحی هیچ نقشی ندارند اما در برنامه ریزی **HTML5** موارد بدی می باشند زیرا به خاطر خسارت های قابلیت استفاده و در دسترس بودنشان مورد سرزنش هستند.

§ <frame>
§ <frameset>
§ <noframes>

عناصر زیر در **HTML5** وارد نمی شوند، زیرا اغلب استفاده نشده اند و ایجاد سردرگمی می کنند و همچنین ساختار آنها توسط عناصر دیگر به کار گرفته می شود.

- <acronym> رفته و به جای آن از <abbr> برای مخفف ها استفاده می شود.
- <applet> به نفع آبجکت حذف شده است.
- استفاده ی <isindex> می تواند توسط استفاده فرم کنترل ها جایگزین شود.

- **<dir>** به خاطر **** کنار گذاشته شده است.

در انتها عنصر **<noscript>** فقط با ترکیب **HTML** منطبق می شود و وارد ترکیب **xml** نمی شود، چرا که استفاده ی آن به یک تجزیه گر **HTML** وابسته است.

اگر شما هنوز به استفاده از این عناصر اصرار دارید، صفحه ی وب شما خراب خواهد شد و مرورگر از آن پشتیبانی خواهد کرد. بنابراین هزینه ی انجام آن این است که صفحه ی شما از ارزیابی **HTML5** عبور نخواهد کرد.

ویژگی هایی که نباید استفاده کنید

در **HTML5** هیچ کدام از ویژگی ها به هدف طراحی نباید استفاده شوند. طراحی باید به وسیله ی **CSS** محافظت شود. منظور من از طراحی اندازه ی فونت، رنگ، حاشیه، نوع فونت و هرآنچه به هدف نمایش استفاده می شود، می باشد.

شما باید از استفاده از ویژگی هایی که در جدول زیر ارائه شده، بپرهیزید.

Attributes	Elements
align	caption, iframe, img, input, object, legend, table, hr, div, h1, h2, h3, h4, h5, h6, p, col, colgroup, tbody, td, tfoot, th, thead and tr.
alink, link, text and vlink	body
background	body
bgcolor	table, tr, td, th and body
border	table and object
cellpadding and cellspacing	table
char and charoff	col, colgroup, tbody, td, tfoot, th, thead and tr
clear	br

compact	dl, menu, ol and ul
frame	table
frameborder	iframe
height	td and th
hspace and vspace	img and object
marginheight and marginwidth	iframe
noshade	hr
nowrap	td and th
rules	table
scrolling	iframe
size	hr
type	li, ol and ul
valign	col, colgroup, tbody, td, tfoot, th, thead and tr
width	hr, table, td, th, col, colgroup and pre

در پایان ضمن تشکر از انتخاب شما، امیدواریم مطالب این کتاب برای شما مفید بوده باشد.

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

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