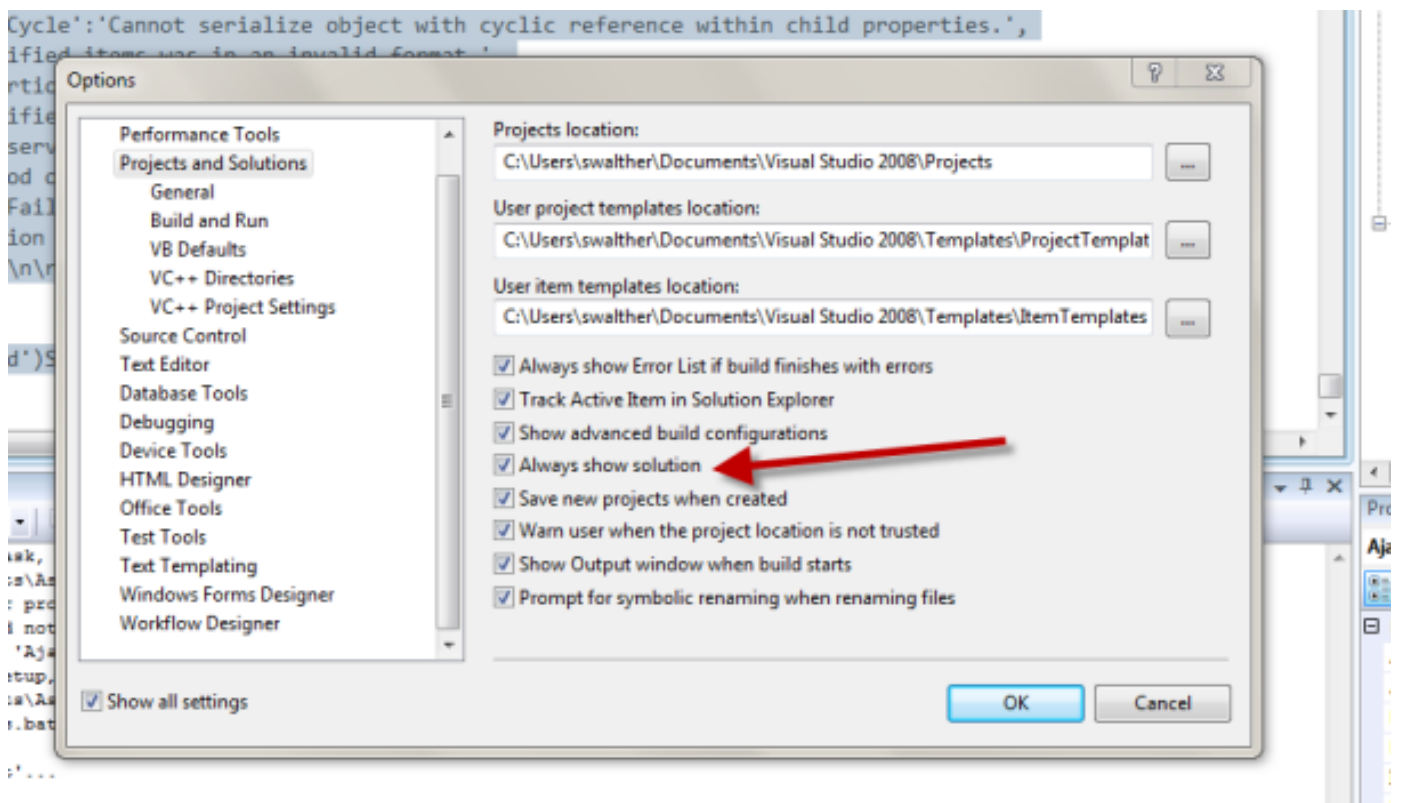


با توجه به افزایش کاربرد jQuery و دیگر کتابخانه‌های جاوا اسکریپت در برنامه‌های تحت وب، یکی از چالش‌های همیشگی برنامه نویسان، فشرده سازی فایل‌های دربرگیرنده کدهای جاوا اسکریپت و شیوه نامه‌ها می باشد. برای این منظور راه‌های مختلفی مانند استفاده از ابزارهای آنلاین مانند [این](#) + و [این](#) + وجود دارند. اما یک روش خودکار هم وجود دارد که در زمان Build پروژه‌های دات نت می‌توان از آن بهره گرفت.

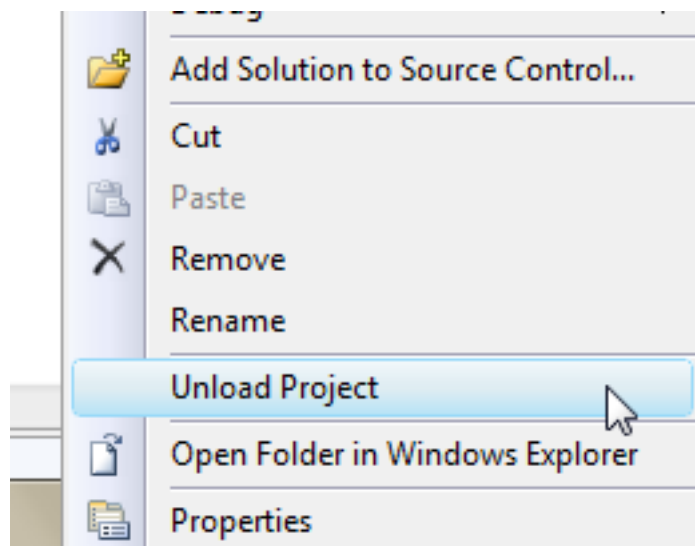
Microsoft Ajax Minifier

یک ابزار رایگان جهت فشرده سازی فایل‌های جاوا اسکریپت و شیوه نامه‌ها است. شما می‌توانید این ابزار را از [صفحه خانگی آن](#) در [سایت asp.net](#) دریافت کنید. جهت استفاده از این ابزار می‌توان از طریق خط فرمان عمل کرد. اما روش ساده‌تر که هدف اصلی این مطلب است به شرح زیر است:

1. در VisualStudio.NET از طریق منو به مسیر **Tools, Options, Projects and Solutions** بروید و گزینه **Always show solution** را تیک بزنید.



2. از Solution Explorer بر روی عنوان پروژه کلیک راست کرده و گزینه **Unload Project** را انتخاب نمایید.



3. مجدداً روی عنوان پروژه کلیک راست کرده و گزینه **Edit** را انتخاب کنید و دستورات زیر را قبل از بسته شدن تگ Project اضافه کنید:

```
<Import Project="$(MSBuildExtensionsPath)\Microsoft\MicrosoftAjax\ajaxmin.tasks" />
<Target Name="AfterBuild">
  <ItemGroup>
    <JS Include="**\*.js" Exclude="**\*.min.js;Scripts\*.js" />
  </ItemGroup>
  <ItemGroup>
    <CSS Include="**\*.css" Exclude="**\*.min.css" />
  </ItemGroup>
  <AjaxMin
    JsSourceFiles="@{JS}" JsSourceExtensionPattern="\.js$" JsTargetExtension=".min.js"
    CssSourceFiles="@{CSS}" CssSourceExtensionPattern="\.css$" CssTargetExtension=".min.css" />
</Target>
```

4. دوباره بر روی عنوان پروژه کلیک راست کرده و گزینه **Reload Project** را انتخاب کنید. توجه کنید که با این کار ما یک MSBuild task با عنوان ajaxmini به پروژه اضافه کردیم. این وظیفه که در زمان Build پروژه اجرا خواهد شد فایل های جاوا اسکریپت را فشرده و با پسوند min.js و همچنین فایل های CSS را پس از فشرده سازی با پسوند min.css در همان مسیر فایل مادر بطور خودکار ذخیره می کند.

نکته:

اگر به دستورات تنظیمات فوق نگاه دقیقتری بیندازیم، متوجه عبارات **Include** و **Exclude** می شویم. توسط این دو صفت شما می توانید الگوهایی را جهت فشرده سازی و یا عدم فشرده سازی تعریف کنید. بدین معنا که توسط الگوی های ذکر شده در تنظیمات فوق از فشرده سازی فایل های با پسوند min.js و min.css خودداری می شود. در این شرایط در حین توسعه برنامه، شما می توانید از فایل های با کد خوانا استفاده نمایید و زمان انتشار و Build پروژه بصورت خودکار آنها را با فایل های فشرده جایگزین کنید. این ابزار تمامی فضاهای خالی، ';' و '{ }' های اضافی و توضیحات را از کدهای شما حذف می کند. متغیرها و توابع شما را به اسامی کوچک تر تغییر نام می دهد. [و ...](#) همچنین شما از کتابخانه این پروژه می توانید در زمان اجرا و سورس برنامه خود استفاده کنید. جهت اطلاعات بیشتر می توانید به [سایت مربوطه](#) مراجعه نمایید.

نظرات خوانندگان

نویسنده:

وحید نصیری

تاریخ:

۲۲:۴۲ ۱۳۹۱/۰۴/۲۹

ممنون. من از این کتابخانه استفاده می‌کنم:

[Yahoo! UI Library: YUI Compressor for .Net](#)

```

using System.Globalization;
using System.IO;
using System.Text;
using Yahoo.Yui.Compressor;

namespace Deploy.Core
{
    public static class CompressCssJs
    {
        public static void Compress(string file)
        {
            var ext = Path.GetExtension(file).ToLower();
            switch (ext)
            {
                case ".css":
                    compressCss(file);
                    break;
                case ".js":
                    if (!file.ToLower().EndsWith(".min.js") && !file.ToLower().EndsWith(".pack.js"))
                        compressJs(file);
                    break;
            }
        }

        static void compressCss(string file)
        {
            var css = File.ReadAllText(file);
            var compressedCss = new CssCompressor().Compress(css);
            File.WriteAllText(file, compressedCss, Encoding.UTF8);
        }

        static void compressJs(string file)
        {
            var js = File.ReadAllText(file);

            var compressedJavaScript = new JavaScriptCompressor
            {
                CompressionType = CompressionType.Standard,
                DisableOptimizations = false,
                Encoding = Encoding.UTF8,
                LineBreakPosition = -1,
                ObfuscateJavaScript = true,
                PreserveAllSemicolons = false,
                ThreadCulture = CultureInfo.CurrentCulture,
                IgnoreEval = false,
                LoggingType = LoggingType.None
            }.Compress(js);
            File.WriteAllText(file, compressedJavaScript, Encoding.UTF8);
        }
    }
}

```

نحوه استفاده از اون رو با کدنویسی در بالا ملاحظه می‌کنید (ملاحظات utf8 و زبان فارسی هم در آن لحاظ شده). کاری که هنگام ارائه نهایی انجام می‌دم، اسکن فایل‌های نهایی و بررسی پسوندها و سپس استفاده از متد Compress فوق روی فایل‌های اسکریپت و css یافت شده است.

نویسنده:

علیرضا اسم‌رام

تاریخ:

۹:۰۶ ۱۳۹۱/۰۴/۳۰

ضمن تشکر از آقای نصیری بخاطر معرفی روش دیگر، توجه فرمایید من تنها یک معرفی اجمالی درباره این ابزار داشتم. این ابزار

امکانات دیگری نیز دارد که بطور خاص برخی از آنها به شرح زیر است:
آنالیز فایل های جاوا اسکریپت و اعلام هشدارهای مناسب به برنامه نویس. [+](#)
اگر از فایل های RESX جهت چند زبانه کردن سایت استفاده می کنید این ابزار بسیار کارآمد خواهد بود. [+](#)
همچنین این ابزار نیز از UTF-8 بخوبی پشتیبانی می کند.

نویسنده: وحید نصیری
تاریخ: ۹:۱۸ ۱۳۹۱/۰۴/۳۰

ممنون. ابزار یاهو هم امکان آنالیز ([LoggingType](#)) را دارد. فقط در اینجا چون مد نظر من نبوده به none تنظیم شده.

نویسنده: علیرضا اسمرام
تاریخ: ۹:۳۰ ۱۳۹۱/۰۴/۳۰

به نظر میاد نسبت به هم در یک سطح باشند. سپاس از مطالب به موقع و بجای شما.

نویسنده: رضا.ب
تاریخ: ۱۹:۰۶ ۱۳۹۱/۰۴/۳۰

فکر میکنم از این رهیافت در سلوشن های مرجع میکروسافت نشه یاد کرد. در asp.net mvc 4 دو تکنیک Bundle کردن (دسته کردن فایل های مشابه در یک دسته) و Minify کردن (از بین بردن فواصل اضافه و کوچک کردن نام متغیرها و حذف کامنت ها) اضافه شده. که این کار رو به صورت توکار خود فریم ورک انجام میده فقط لازمه با دستورات Syte.Render و Script.Render اونارو آدرس دهی کنیم.
همچنین این قابلیت وجود دارد که در زمان debug فایل ها رو خانا و قابل فهم مشاهده کنید. در یکی از سری پست های آقای شهروز جعفری [^](#) این مهم اشاره کردن. هرچند پست ایشان اندکی قدیمی است (موقع نگارش beta مطلب منتشر شد که حالا با آمدن نگارش RC اندکی فرق کرده است)

نویسنده: علیرضا اسمرام
تاریخ: ۱۹:۳۵ ۱۳۹۱/۰۴/۳۰

با تشکر از آقای رضا.ب....
مطلب آقای جعفری را مطالعه کردم. آیا امکان استفاده از فایل های Resource جهت Localization در این تکنیک وجود دارد؟
همچنین آیا فایل ها اسکریپت آنالیز می شوند و ما از هشدارهای مناسب مطلع می شویم؟

نویسنده: ایمان اسلامی
تاریخ: ۲۰:۳۹ ۱۳۹۱/۰۴/۳۰

با تشکر از دوستان بابت تمامی روش های بحث شده.
میخواستم بدونم برای فشرده سازی یک web site قبل از upload با gzip هم میشه اطلاعات مختصری بدهید.
با تشکر

نویسنده: علیرضا اسمرام
تاریخ: ۲۳:۰۳ ۱۳۹۱/۰۴/۳۰

تا جایی که من مطلع هستم از طریق تنظیمات IIS می توان پاسخ به درخواست ها را (شامل فایل های استاتیک و داینامیک) به کمک gzip فشرده کرد. اما اگر اطلاعات صحیح باشد این موضوع کمی بار CPU را افزایش می دهد، هر چند گاهی تا 75% حجم اطلاعات رد و بدل شده را کاهش می دهد.
برای اطلاعات بیشتر در مورد تنظیمات IIS6 می توانید به [+](#) و [+](#) مراجعه کنید.
همچنین اگر روی IIS7 به بعد میزبانی می شوید می توانید درون فایل Web.Config و درون تگ system.webServer تنظیمات زیر را اضافه کنید:

```
<httpCompression directory="%SystemDrive%\inetpub\temp\IIS Temporary Compressed Files">
  <scheme name="gzip" dll="%Windir%\system32\inetsrv\gzip.dll" staticCompressionLevel="9"
dynamicCompressionLevel="4" />
  <scheme name="deflate" dll="%Windir%\system32\inetsrv\gzip.dll" staticCompressionLevel="9"
dynamicCompressionLevel="4" />
  <dynamicTypes>
    <add mimeType="text/*" enabled="true" />
    <add mimeType="message/*" enabled="true" />
    <add mimeType="application/x-javascript" enabled="true" />
    <add mimeType="application/atom+xml; charset=utf-8" enabled="true" />
    <add mimeType="*/*" enabled="false" />
  </dynamicTypes>
  <staticTypes>
    <add mimeType="text/*" enabled="true" />
    <add mimeType="message/*" enabled="true" />
    <add mimeType="application/javascript" enabled="true" />
    <add mimeType="*/*" enabled="false" />
  </staticTypes>
</httpCompression>
```

(برای مطالعه بیشتر این [+](#) را ببینید.)

نویسنده: ایمان اسلامی
تاریخ: ۸:۱۴ ۱۳۹۱/۰۴/۳۱

ممنون از جواب خوبتون
پس بطور کلی بهتر است که اینکارو انجام ندهیم؟

نویسنده: علیرضا اسم‌رام
تاریخ: ۹:۵۱ ۱۳۹۱/۰۴/۳۱

من پیشنهاد می‌کنم اگر شرایطش رو دارید، امتحانش کنید. حتی برای تجربه شده! و خوشحال میشم نتیجه تجربه تان را با ما به اشتراک بگذارید.

نویسنده: رضا.ب
تاریخ: ۱۹:۱ ۱۳۹۱/۰۴/۳۱

تا جایی که اطلاع دارم، تکنیک B/M فقط باعث کاهش میزان درخواست‌های HTTP بر روی وب سرور میشود. (آموزش نسخه 4 MVC RC [^](#))

نویسنده: صابر فتح اللهی
تاریخ: ۱۱:۱۹ ۱۳۹۱/۰۵/۱۶

سلام
مهندس نصیری این روشی که شما فرمودین
در زمان اجرا خروجی فشرده میکنه یا اینکه نه فشرده میکنه و همونجا هم ذخیره میکنه؟

نویسنده: وحید نصیری
تاریخ: ۱۳:۵۴ ۱۳۹۱/۰۵/۱۶

هنگام ارائه نهایی (در پوشه‌ای جداگانه از فایل‌های اصلی) یکبار روی فایل‌ها اجرا می‌شود.

نویسنده: صابر فتح اللهی
تاریخ: ۸:۳۹ ۱۳۹۱/۰۹/۰۶

سلام

به نظر شما بهتر نیست این کدها در یک هندلر استفاده بشه که در زمان اجرا خروجی فشرده شده برای کاربر ارسال بشه با توجه به کش شدن این نوع فایل ها ایا سربار سیستم اضافه میشه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۹/۰۶ ۹:۵۶

سیستم bundling جدید MVC4 در زمان اجرا این کارها را انجام می دهد. کش شدن هم سربار سیستم رو کم می کنه چون درخواست جدیدی به سرور ارسال نخواهد شد.

نویسنده: سیاه
تاریخ: ۱۳۹۱/۱۱/۱۸ ۵:۲۷

سلام؛ راهی وجود داره که لینک فایل های css و js رو در source page مرورگر غیرفعال کرد ؟
و امکان اینکه فرد با کلیک روی لینک محتوای فایل های css و js رو نبینه .
با تشکر .

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۱۸ ۹:۰۹

خیر. مرورگر باید به این لینک ها دسترسی داشته باشه برای رندر سایت. به همین نحو این دسترسی برای تمام دنیا وجود دارد.

نویسنده: مرتضی مختاری
تاریخ: ۱۳۹۲/۰۳/۲۳ ۲۳:۵۲

سلام؛ اگه از کش کردن اطلاعات استاتیک استفاده کنیم دیگه نیازی به minify کردن فایل ها داریم یا خیر؟

احتمالا با عباراتی مانند طراحی table less و مزیت‌های طراحی با CSS، همانند سرعت بالاتر بارگذاری سایت در مقایسه با نمایش یک جدول که نیازمند دریافت تمام جزئیات آن و سپس رندر نهایی اطلاعات آن توسط مرورگر است، بارها برخورد داشته‌اید. اما ... آیا یکبار سعی کرده‌اید که به صورت دستی همان کارهایی را که پیشتر با HTML table انجام می‌دادید، اینبار توسط CSS پیاده سازی کنید؟

در اکثر اوقات نتیجه کار مایوس کننده، بسیار سخت و نگهداری آن در طول زمان بسیار مشکل خواهد بود؛ به علاوه سازگاری با مرورگرهای مختلف و نکات ریز هر کدام را نیز لحاظ کنید. به همین جهت تعدادی فریم ورک CSS برای شبیه سازی گرید و جدول تهیه شده‌اند که کار طراحی table less را بسیار ساده و لذت بخش کرده‌اند. یکی از این موارد، فریم ورک [Blueprint CSS](http://www.dotnettips.info) نام دارد و در ادامه نحوه استفاده از آن را مرور خواهیم کرد. این مرور هم مستقل است از فناوری سمت سرور مورد استفاده و صرفا مباحث HTML و CSS آن بررسی خواهند شد.

دریافت Blueprint CSS

این فریم ورک سورس باز را از مخزن کدهای آن در GitHub می‌توانید دریافت کنید: ([^](#)) البته نگران حجم نزدیک به 4 مگابایتی بسته دریافتی آن نباشید؛ زیرا نهایتا با سه فایل CSS از آن بیشتر کاری نداریم و مابقی مثال‌های آن هستند.

پس از دریافت آن، یک پوشه را به نام blueprint ایجاد کرده و سه فایل ie.css, print.css و screen.css را در آن قرار دهید. به علاوه داخل این پوشه، یک پوشه جدید دیگر را به نام src ایجاد کرده و فایل grid.png موجود در این بسته را نیز در آن کپی کنید.

ساختار ابتدایی یک صفحه مبتنی بر Blueprint CSS

پس از ایجاد پوشه blueprint و src به نحوی که توضیح داده شد، ابتدایی‌ترین ساختار یک صفحه تشکیل شده با blueprint css به نحو زیر است:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Blueprint test page</title>

<!-- Framework CSS -->
<link rel="stylesheet" href="blueprint/screen.css" type="text/css" media="screen, projection">
<link rel="stylesheet" href="blueprint/print.css" type="text/css" media="print">
<!--[if lt IE 8]><link rel="stylesheet" href="blueprint/ie.css" type="text/css" media="screen,
projection"><![endif]>-->
</head>
<body>
<div class="container showgrid">
test
<hr class="space" />
<hr class="space" />
<hr class="space" />
<hr class="space" />
<hr class="space" />
test
</div>
</body>
</html>
```

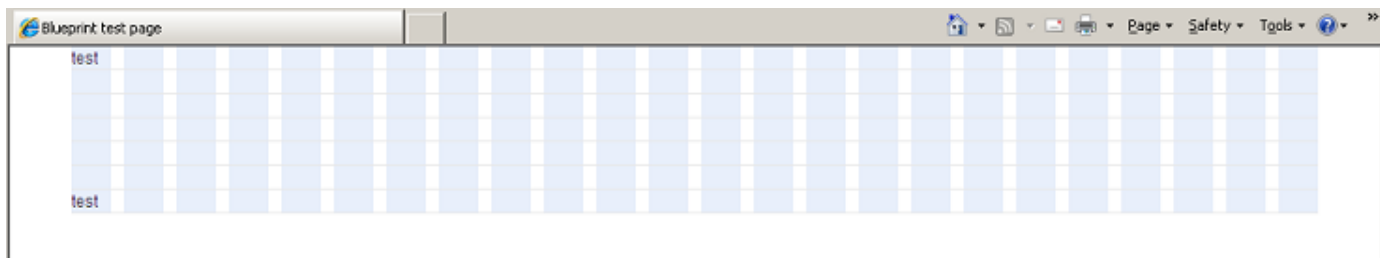
توضیحات:

پس از مشخص سازی DocType (مهم)، سه فایل CSS یاد شده به header صفحه اضافه خواهند شد. همانطور که ملاحظه می‌کنید، سازگاری با IE نیز مدنظر آن بوده است.

کار با blueprint css همواره داخل div زیر انجام می‌شود:

```
<div class="container">
  page
</div>
```

توسط کلاس container یک گرید به عرض 950px در میانه صفحه برای شما تشکیل خواهد شد. اگر علاقمند باشید که این گرید را مشاهده نمائید و همچنین بتوانید ستون‌های آن را نیز شمارش کنید، تنها کافی است showgrid را به این class تعریف شده اضافه نمائید (همانند ساختار صفحه فوق). به این ترتیب شکل زیر نمایان خواهد شد:



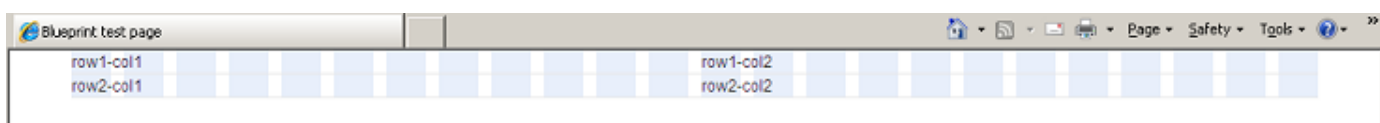
مطابق شکل فوق، در این عرض مشخص، 24 ستون آن در اختیار ما خواهند بود. به علاوه ذکر hr با class=space سبب خواهد شد تا مطابق تنظیمات و فاصله بندی منظم این فریم ورک، یک سطر خالی برای ما ایجاد شود.

طراحی بدون جدول با Blueprint CSS

در ادامه قصد داریم در این صفحه ابتدایی، یک جدول با دو ستون و دو ردیف را ایجاد کنیم:

```
<body>
  <div class="container showgrid">
    <div class="span-12">
      row1-col1
    </div>
    <div class="span-12 last">
      row1-col2
    </div>
    <div class="span-12">
      row2-col1
    </div>
    <div class="span-12 last">
      row2-col2
    </div>
  </div>
</body>
```

که شکل زیر را برای ما ایجاد خواهد کرد:



توضیحات:

ستون‌های گرید نهایی با رنگ آبی مشخص هستند (class=container showgrid). اگر نیاز به 12 ستون داریم، می‌نویسیم span-12 و ... همین! به این ترتیب یک سلول جدول، با 12 ستون در اختیار ما خواهد بود. سلول بعدی هم در اینجا 12 ستونه است. اما یک last را اضافه‌تر دارد. در span-12 last این last به معنای انتهای ردیف جاری است و ذکر آن الزامی است. تا اینجا یک ردیف تمام شد. اکنون در ادامه ردیف دوم را نیز به همین ترتیب با دو div و classهایی که ملاحظه می‌کنید، مشخص خواهیم کرد.

نحوه کار کلی با Blueprint css به همین سادگی است که ملاحظه می‌کنید. تعداد ستون‌های مورد نیاز را با ذکر container showgrid به سادگی می‌توان شمارش کرد. سپس این اعداد شمارش شده و مد نظر را پس از span ذکر کنید. مثلاً اگر یک طرح سه ستونه نیاز دارید به صورت زیر خواهد بود:

```
<body>
  <div class="container showgrid">
    <div class="span-8">
      row1-col1
    </div>
    <div class="span-8">
      row1-col2
    </div>
    <div class="span-8 last">
      row1-col3
    </div>
  </div>
</body>
```

طراحی سلول‌های تو در تو

سؤال: ما پیشتر در یک html table به سادگی می‌توانستیم داخل یک سلول آن حتی یک جدول جدید نیز قرار دهیم، اینجا چطور؟ پاسخ: در اینجا هم بجای td و tr و table، از divهای تو در تو استفاده کنید. بستن ستون آخر را با last یاد شده فراموش نکنید. مثلاً:

```
<body>
  <div class="container showgrid">
    <div class="span-8">
      <div class="span-4">
        row-1, col1 : cell-1
      </div>
      <div class="span-4 last">
        row-1, col1 : cell-2
      </div>
    </div>
    <div class="span-8">
      row1-col2
    </div>
    <div class="span-8 last">
      row1-col3
    </div>
  </div>
</body>
```

در اینجا در اولین div تعریف شده دو div تو در تو اضافه شده‌اند. البته با توجه به اینکه div والد 8 ستونی است، جمع عرض divهای فرزند باید 8 باشد که در اینجا به دو div چهارستونی تقسیم شده است.



تا اینجا با کلیات نحوه طراحی یک جدول به کمک CSS و فریم ورک Blueprint CSS آشنا شدیم (به کمک container و span-n آن). در ادامه مرور سریعی خواهیم داشت بر سایر امکانات این فریم ورک CSS و منظور از این امکانات، کلمات و عبارات مجازی است که می‌توانید داخل classهای divهای تعریف شده اضافه نمایید (CSS selectors تعریف شده در آن):

prepend-n و border:

فرض کنید در divهای تو در تو قسمت قبل، قصد داریم عرض ستون اول را بجای 4 ستون به 3 ستون تبدیل کنیم، اما این div را یک ستون به سمت راست حرکت دهیم:

```
<body>
  <div class="container showgrid">
    <div class="span-8">
      <div class="prepend-1 span-3 border">
        row-1, col1 : cell-1
      </div>
      <div class="span-4 last">
        row-1, col1 : cell-2
      </div>
    </div>
    <div class="span-8">
      row1-col2
    </div>
    <div class="span-8 last">
      row1-col3
    </div>
  </div>
</body>
```

برای این منظور همانطور که ملاحظه می‌کنید از prepend-1 استفاده شده است. border در اینجا سبب خواهد شد تا در سمت راست div یک خط عمودی رسم شود. در مقابل آن colborder هم وجود دارد که سبب ترسیم حاشیه با فاصله بیشتری نسبت به border می‌شود.

شبهه به همین قابلیت، با append-x (افزودن تعدادی ستون به سمت راست)، prepend-top (فاصله‌ای به اندازه 1.5em را به بالای div اضافه می‌کند) و append-bottom (فاصله‌ای به اندازه 1.5em را به پایین div اضافه می‌کند) نیز وجود دارد. در مقابل این‌ها، push-n و pull-n هم وجود دارند. کار append و prepend اضافه کردن چند ستون به بعد و قبل از یک div است. push یک div را به تعداد واحدی که مشخص می‌کنیم به سمت راست حرکت می‌دهد. pull یک div را n ستون به سمت چپ حرکت خواهد داد (بدون تغییری در تعداد ستون‌ها).

[دریافت مرجع سریع Blueprint CSS](#)

نظرات خوانندگان

نویسنده: مجتبی حسینی
تاریخ: ۱۷:۳۰ ۱۳۹۱/۰۶/۰۱

از سرعت پاسخ‌گویی‌تان در شگفتم!
بسیار ممنونم.

نویسنده: na3er-faraji
تاریخ: ۲۰:۳۶ ۱۳۹۱/۰۶/۰۱

خسته نباشید. بسیار عالی و کاربردی بود. sample هاشو دارم می‌بینم چیزهای کاربردی دیگه ای هم توش پیدا میشه.

نویسنده: هوشنگ
تاریخ: ۲۲:۳۹ ۱۳۹۱/۰۶/۰۱

طرز کار همه css framework ها شبیه به هم هست . من از 960gs استفاده میکنم . تنها تفاوت این اضافه داشتن typography و style فرم هست که من خودم اضافه میکردم

نویسنده: میثم
تاریخ: ۱۷:۱۷ ۱۳۹۱/۰۶/۲۳

سلام و تشکر فراوان
به نظر بنده این CSS Framework یک مشکل کوچک دارد و اون محدودیت عرض هستش و رویکرد جدیدی که در طراحی وب سایت‌ها مطرح هستش رو در نظر نگرفتند. سایتهایی که محدودیت عرض ندارند و بنا به رزولوشن صفحه کاربر عرض اونا تغییر میکنه و در حقیقت فقط یک minwidth دارند. یه چیزی شبیه به فریم ورک [YAML](#)
آیا همچون طرح‌هایی شبیه به همین [سایت](#) اشکالی دارند؟ یا این مساله به سلیقه بستگی دارد
ممنون

نویسنده: وحید نصیری
تاریخ: ۱۷:۴۰ ۱۳۹۱/۰۶/۲۳

من چون در سایت جاری از blueprint css استفاده می‌کنم، خواستم نحوه کار با آن را به اشتراک بگذارم.

نویسنده: احسان بوربور
تاریخ: ۱۹:۲۱ ۱۳۹۱/۰۷/۱۳

با سپاس فراوان خیلی کاربردی بود

من هم خواستم برای به اشتراک گذاشتن این مطلب ازتون تشکر کنم.

نویسنده: علی مهدوی
تاریخ: ۱۱:۵۴ ۱۳۹۱/۰۸/۰۹

با bootstrap چه فرقی داره؟

نویسنده: وحید نصیری
تاریخ: ۱۲:۵ ۱۳۹۱/۰۸/۰۹

بسته به نیاز و کاربرد باید ابزار مناسبی رو انتخاب کرد. آیا شما نیاز به تمام عناصر موجود در فریم ورک توئیتر bootstrap دارید؟ من نیازی نداشتم به همین جهت انتخاب انجام شده هم بسیار سبک، کم حجم و متناسب بوده. از توئیتر bootstrap هم فقط دکمه‌های css ایی آن برای من جالب بود. نیازی به مابقی آن نداشتم.

نویسنده: علی

تاریخ: ۱۳۹۲/۰۵/۰۹ ۱۵:۸

سلام و خسته نباشید

ممنونم از بابت معرفی این فریم ورک، ولی آیا از rtl هم پشتیبانی میکنه ؟
و اینکه من موفق به دانلود نشدم، شما چطور دانلود کردین ؟

نویسنده: وحید نصیری

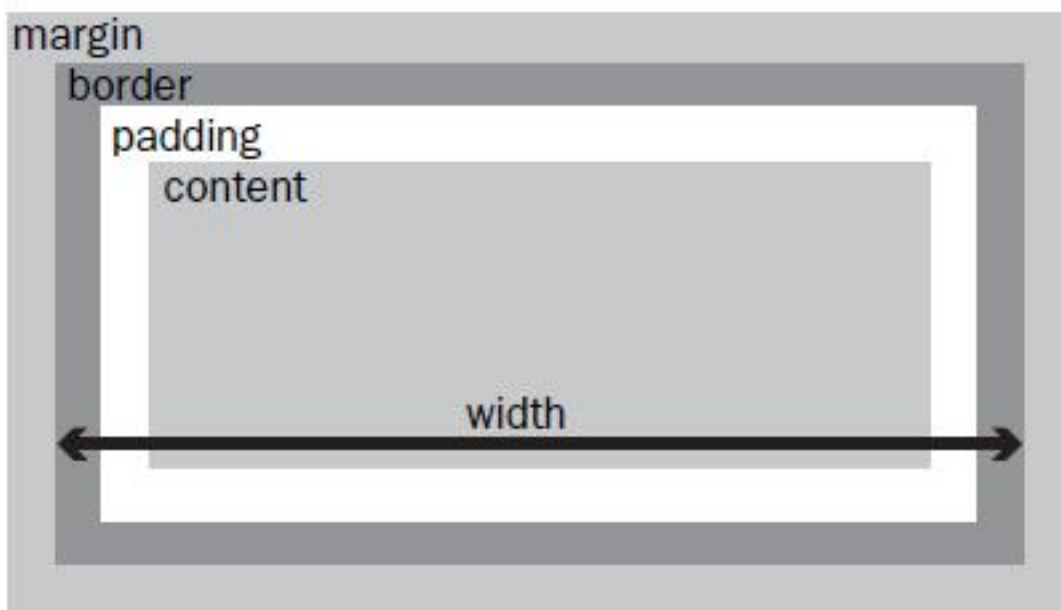
تاریخ: ۱۳۹۲/۰۵/۰۹ ۱۵:۲۰

- به آدرس مخزن کد آن در متن [اشاره شده](#) .

- نیازی به نسخه RTL نداره و من از نسخه اصلش استفاده کردم. ولی بله، نسخه RTL [هم داره](#) .

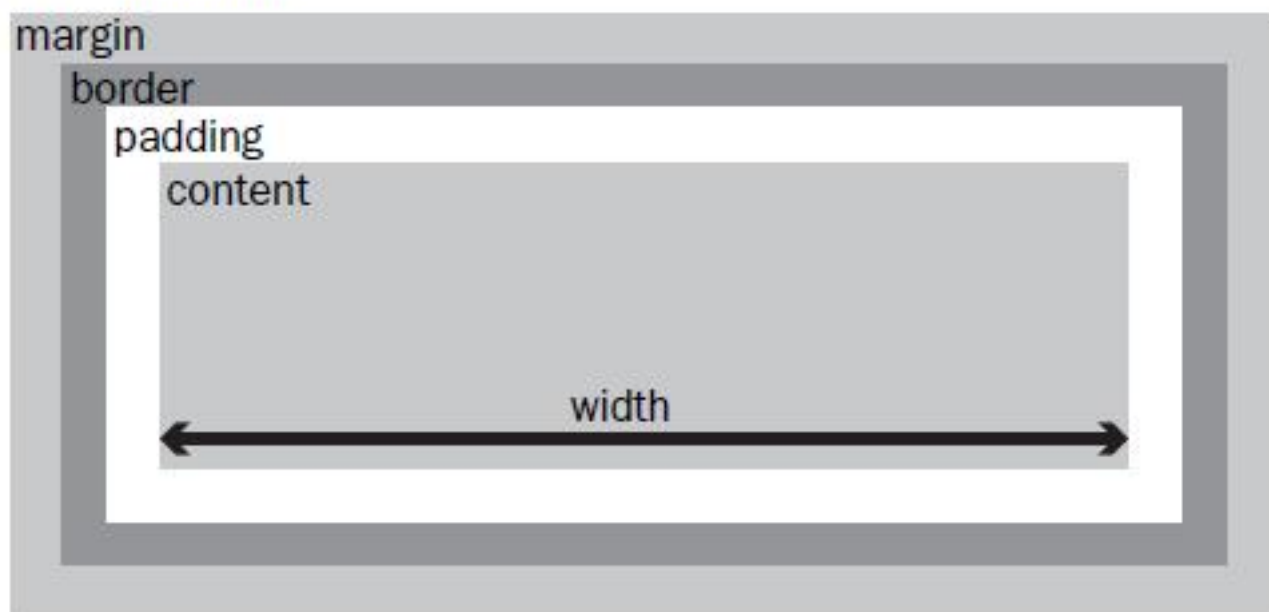
وقتی که در حال ساخت یک وب سایت می‌باشید با مشکلات و باگ‌های اینترنت اکسپلورر به احتمال زیاد برخورد داشته‌اید. یکی از این مشکلات باگی است که اینترنت اکسپلورر در محاسبه عرض (width) یک box دارد. محاسبه عرض یک box در اینترنت اکسپلورر 6 به صورت زیر است:

IE Box Model



این در حالی است که روش استاندارد محاسبه عرض یک box به صورت زیر است:

W3C Box Model



برای رفع این مشکل چندین روش وجود دارد. اما به نظر من بهترین روش قرار دادن تگ زیر در ابتدای صفحه است:

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

نظرات خوانندگان

نویسنده: سعید
تاریخ: ۱۳۹۱/۰۶/۰۶ ۱۲:۷

لطفا پشتیبانی از ie6 رو متوقف کنید. روش خوب برای اینکار :

[ie6-upgrade-warning](#)

اجازه بدید کاربران ieهای قدیمی احساس خوبی از کاربری آن نداشته باشند و مجبور شوند حداقل یک سرویس پک 3 نصب کنند با ie8 .

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۰۶ ۱۲:۱۹

ممنون. برای دوستانی که به گوگل کد دسترسی ندارند، اینجا آپلود شد:

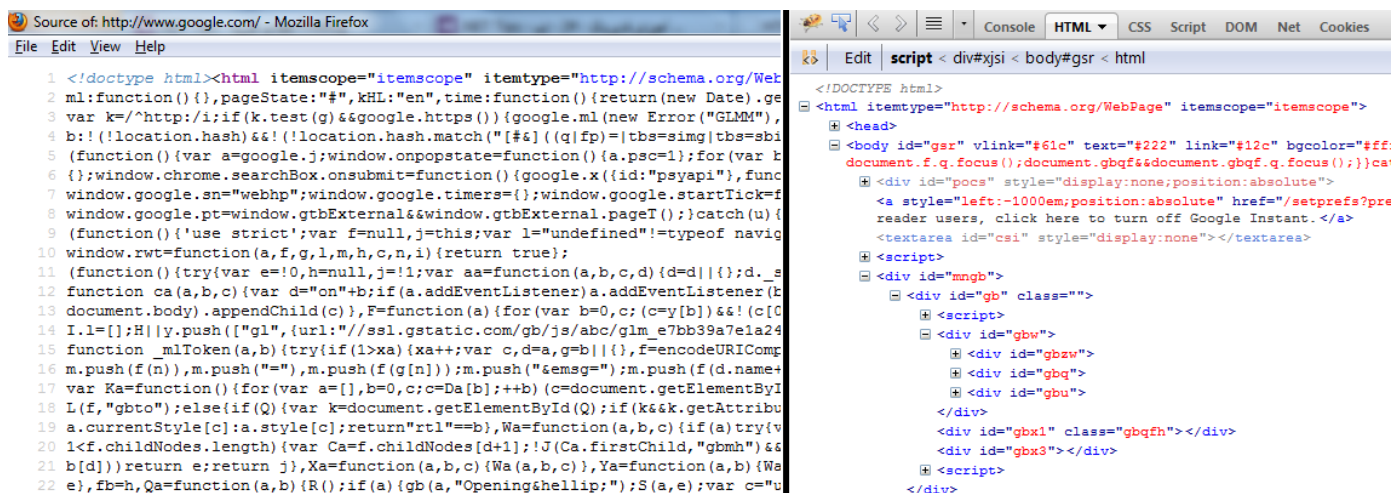
[ie6-upgrade-warning_2.zip](#)

در این سری از [مقالات آموزش FireBug](#) ، به صورت ترتیبی پیش رفتیم و ابتدا توضیحات تقریباً مفصلی در مورد [پنل Console](#) دادیم.

اکنون به پرکاربردترین بخش آن ، یعنی پنل HTML می‌رسیم.

محتویاتی که در این پنل نمایش داده می‌شود ، کدهای صفحه‌ی جاری ، بصورت زنده است و با چیزی که از سمت سرور به مرورگر ارسال می‌شود متفاوت است ، تگ‌های HTML بصورت درختی مرتب شده اند و رنگی نمایش داده می‌شوند و امکان ویرایش محتوا ، ویرایش استایل ، مشاهده‌ی آرایش تگ‌ها بصورت بصری و ... وجود دارد.

نگاهی به کدهای ارسال شده به مرورگر در آدرس [Google.com](http://www.google.com) و نحوه‌ی نمایش آن در فایرباگ را ملاحظه بفرمایید.
(کدهای ارسال شده سمت چپ - نمایش در فایرباگ سمت راست)



این پنل به سه بخش اصلی تقسیم می‌شود :

بخش اصلی یا NodeView که محتوای صفحه را بصورت درختی و مرتب و رنگی نمایش می‌دهد.

Panel Toolbar که در بالای پنل قرار دارد.

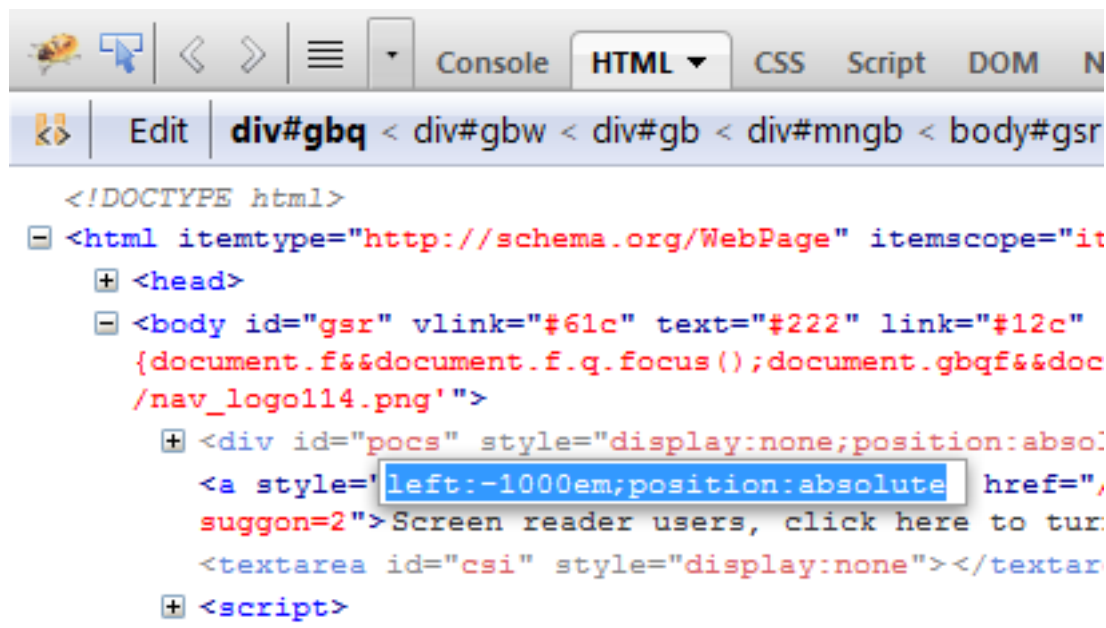
Side Panels که شامل پنل‌های DOM , Layout , Computed , Style می‌شود.

که به ترتیب برای نمایش و ویرایش استایل‌ها ، مشاهده استایل‌های محاسبه شده ، مشاهده Layout یا آرایش و نمایش اطلاعات DOM تگ انتخاب شده در NodeView است.

در این مقاله با دو بخش NodeView و Panel Toolbar ، و در مقاله‌ی بعد با پنل‌های سمت راست یعنی Side Panels آشنا می‌شویم.

ویرایش HTML

هر تگ HTML از یک سری Attribute تشکیل می‌شود که در فایرباگ نام ویژگی بصورت آبی تیره و مقدار آن با رنگ قرمز مشخص شده است. برای ویرایش هریک از آن‌ها کافیت برویش کلیک کنید تا آن مقدار در یک باکس ویرایش به نمایش دربیاید. با ویرایش مقدار ، این تغییر در لحظه بروی صفحه اعمال می‌شود.

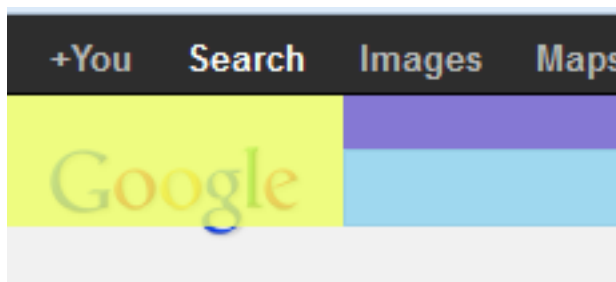


برای اضافه کردن یک Attribute جدید به تگ هم بروی تگ مورد نظر راست کلیک می‌کنید و سپس گزینه‌ی New Attribute را انتخاب می‌کنید. ابتدا نام ویژگی ، یک Enter ، سپس مقدار را وارد می‌کنید. با زدن کلیدهای Enter متوالی ، می‌توانید به وارد کردن ویژگی‌ها ادامه دهید.

برای ویرایش کردن یک تگ و تگ‌های فرزندش ، بروی تگ مورد نظر کلیک کنید تا به حالت انتخاب دربیاید ، سپس بروی دکمه‌ی Edit در بالای پنل کلیک کنید. در نهایت بعد از انجام ویرایش ، مجدداً بروی دکمه‌ی Edit کلیک کنید.

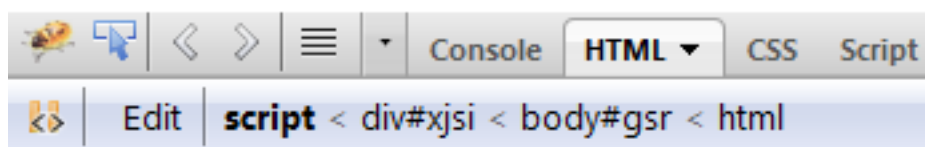
Node View

NodeView نام بخش اصلی پنل HTML است که محتویات صفحه را بصورت مرتب شده و درختی نمایش می‌دهد. تگ (Node) هایی که دارای فرزند می‌باشند ، یک علامت + یا - در کنارشان وجود دارد که با کلیک بروی آن می‌توانید آن تگ را باز/بسته کنید. همچنین این کار با کلیدهای + و - یا Left Arrow و Right Arrow از روی کیبورد هم قابل انجام است. برای باز کردن یک لینک در این قسمت هم از ترکیب Ctrl و کلیک موس کمک بگیرید. در نهایت زمانی که موس را بروی یک تگ قرار می‌دهید ، محیطی که توسط آن تگ در صفحه اشغال شده است بصورت رنگی مشخص می‌شود. که رنگ زرد به معنی محیط margin ، رنگ آبی تیره به معنی محیط padding و رنگ آبی روشن هم به معنی محیط محتوا است.



Panel Toolbar

این قسمت در بالای پنل HTML قرار دارد که گزینه‌های زیر را دارا می‌باشد:



Break On Mutate

این دکمه امکان توقف کد JavaScript ای که سعی در ویرایش محتوای صفحه را دارد ، می‌دهد. زمانی که FireBug تشخیص دهد که کدی سعی در ویرایش محتوا دارد ، شما را به خط مورد نظر از کد ، در پنل Script منتقل می‌کند.

Edit

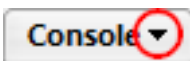
این دکمه برای ویرایش مستقیم محتوای یک تگ بکار می‌رود نکته‌ی جالب در ویرایش محتوا در فایرباگ این است که تغییرات در لحظه اعمال می‌شوند و نیاز به عمل بروزرسانی جداگانه نیست. برای مثال در همین قسمت Edit ، با هر ویرایش محتوا ، تغییرات در لحظه اعمال می‌شوند.

Element Path

زمانی که یک تگ را در FireBug انتخاب می‌کنید ، لیستی از تگ‌ها که از تگ جاری شروع و به تگ ریشه ختم می‌شود ، نمایش داده می‌شود که با کلیک بروی هر کدام ، همان به عنوان تگ فعلی یا انتخاب شده تعیین می‌شود. نتیجه‌ی عملیاتی که بروی این تگ‌های انجام می‌دهید (حرکت موس و راست کلیک کردن) معادل همان عملیات بروی تگ‌های نمایش داده شده در قسمت اصلی (NodeView) است.

Options Menu

هر تب یا پنل در فایرباگ دارای یک سری تنظیمات است که Options Menu نام دارد. تب HTML هم دارای یک سری تنظیمات است که دانش‌تان آنها بسیار به شما کمک خواهد کرد. این منو با کلیک کردن بروی فلش تب (



(یا راست کلیک کردن بروی تب ظاهر می‌شود.

Show Full Text

در صورت فعال بودن ، متون بصورت کامل نمایش داده می‌شوند ، در غیراینصورت بصورت خلاصه شده نمایش داده می‌شوند.

Show White Space

در صورت فعال بودن ، فضاهاى خالى ، کرکترهاى خط جدید و ... را نمایش می‌دهد.

```
<p>
  This menu is reachable via the little
  arrow in the panel tab (
  <a class="image" href="/wiki/index.php
  /File:OptionsMenuArrow.png">
  ) or by right-
  clicking on on the panel tab (since
  <a title="Firebug Release
  Notes" href="/wiki/index.php
  /Firebug_Release_Notes#Firebug_1.9">Fire
  bug 1.9</a>
  ) .
</p>
```

Show Comments

در صورت فعال بودن ، کامنت‌ها را هم نمایش می‌دهد

```
<div id="jump-to-nav">
  <!-- start content -->
  <div class="thumb tright">
    <p>The HTML panel displays the generated
    HTML/XML of the currently opened page.
```

سه گزینه ی Show Entities As Unicode و Show Entities As Symbols ، Show Entities As Names ، نوع نمایش کرکترهای ویژه را تعیین می‌کنند.

Highlight Changes

در صورت فعال بودن ، تگ تغییر یافته توسط JavaScript (یا تگ والدی که قابل مشاهده باشد) Highlight می‌شود

Expand Changes

در صورت فعال بودن ، زمان تغییر دادن یک تگ توسط JavaScript ، والدهای آن تگ باز (Expand) می‌شوند

Scroll Changes Into View

در صورت فعال بودن این گزینه ، هنگام تغییر یک تگ در صفحه توسط JavaScript ، قسمت NodeView به قسمت تغییر بافته حرکت می‌کند.

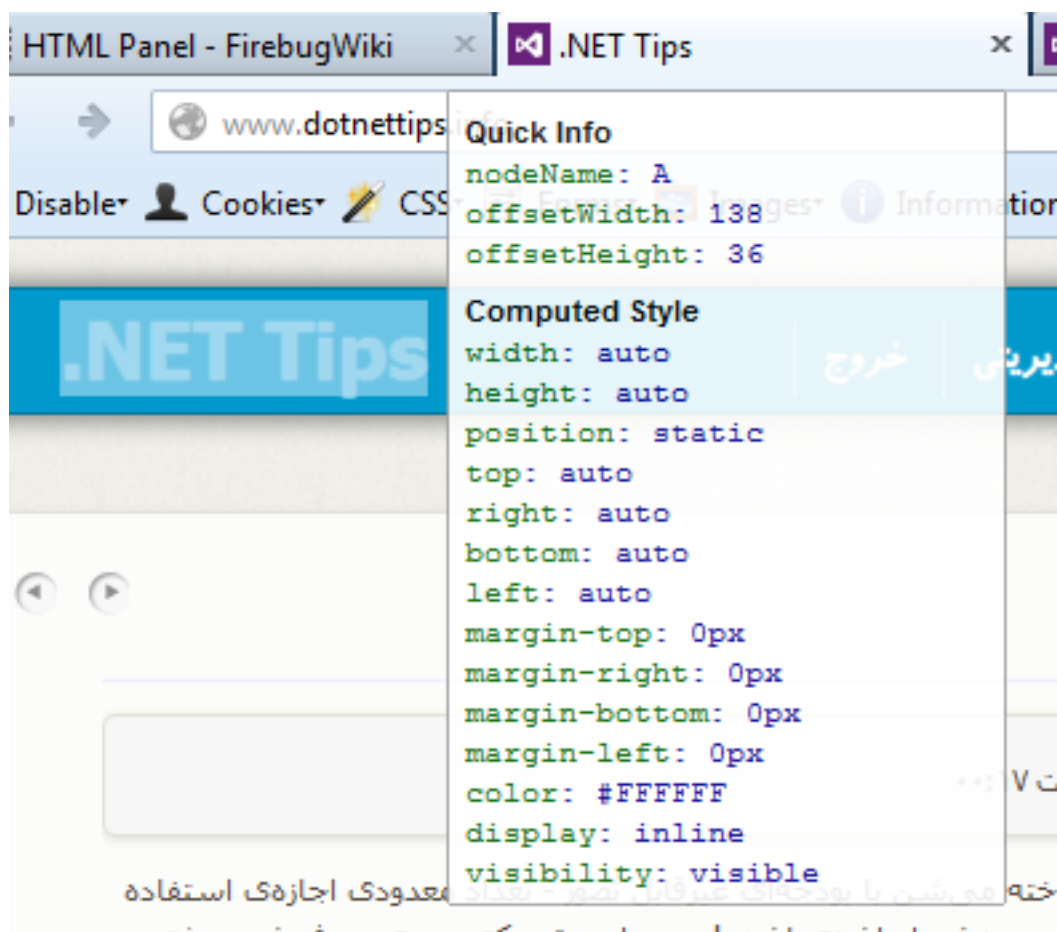
(فعال بودن این گزینه هنگام بررسی سایت هایی که اسلایدر یا سیستم هایی مشابه دارند ، باعث میشه که نتوانید بروی قسمت‌های سایت تمرکز کنید و مدام اسکرول به قسمت تغییرات منتقل می‌شود.)

Shade Box Model

در صورت فعال بودن ، فضای padding , margin و content را به شکلی که در بالا گفته شد نمایش می‌دهد ، در غیر اینصورت فقط یک خط دور تگ نشان می‌دهد.

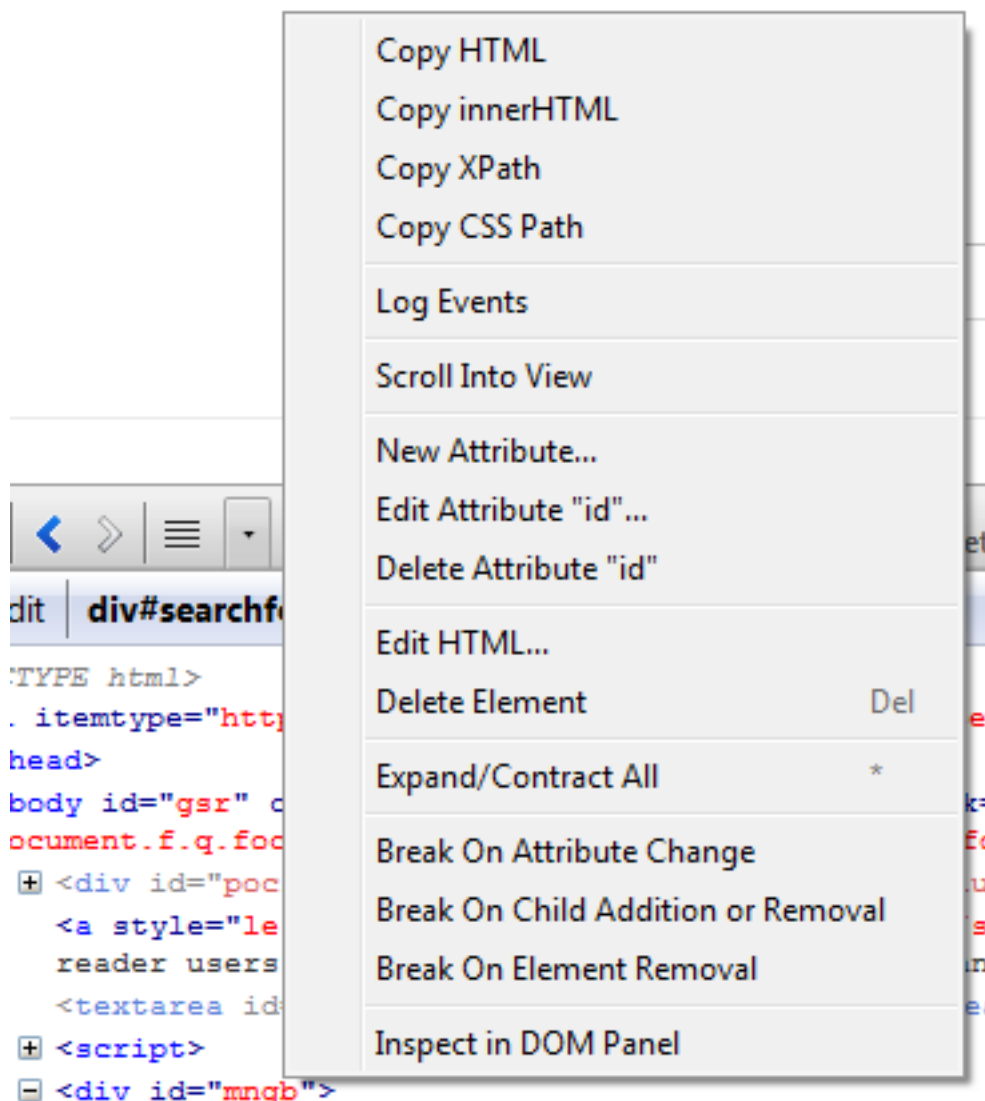
Show Quick Info Box

در صورت فعال بودن ، یک پاپ‌آپ به همراه اطلاعات مختصری از تگ در صفحه نمایش می‌دهد.



Context Menu

اگر بروی یک تگ راست کلیک کنید ، یک منو نمایش داده می‌شود ، در این قسمت با گزینه‌های این منو آشنا می‌شویم.

**Copy HTML**

خود تگ و محتوایش را بصورت HTML در حافظه کپی می‌کند.

Copy innerHTML

محتوای تگ را در حافظه کپی می‌کند.

Copy XPath

آدرس [XPath](#) تگ را در حافظه کپی می‌کند.

Copy CSS Path

[CSS Selector](#) تگ را در حافظه کپی می‌کند.

Log Events

رویدادهای تگ را در پنل Console ثبت می‌کند. (کلیک موس ، فشردن کلید ، ...) برای لغو لاگ کردن ، مجدداً بروی تگ راست کلیک کرده و این گزینه را از حالت انتخاب خارج کنید.

Scroll Into View

صفحه را به جایی که تگ قابل نمایش است منتقل می‌کند.

...New Attribute

یک attribute جدید به تگ اضافه می‌کند.
برای لغو عملیات ، دکمه‌ی Esc را بزنید.

..."<Edit Attribute "<attribute name

اگر بروی یک attribute راست کلیک کرده باشید ، این گزینه و گزینه‌ی بعدی را مشاهده خواهید کرد.
معادل کلیک بروی نام attribute است ، نام را به حالت ویرایش درمی آورد.

"<Delete Attribute "<attribute name

attribute را حذف می‌کند.

...Edit HTML

تگ را به حالت ویرایش می‌برد.
معادل انتخاب تگ ، زدن کلید Edit است.

Delete Element

تگ را حذف می‌کند.
راه دیگر حذف یک تگ ، انتخاب تگ و فشردن کلید Del از کیبورد است.

Expand/Contract All

تگ و Childهایش را باز/بسته می‌کند. (بجز تگ های <script> , <style> , <link>)
می‌توان با ترکیب کلید **+ Shift *** هم این کار را انجام داد که در این حالت تگ‌های فوق هم شامل باز/بسته شدن می‌شوند.

Break On Attribute Change

مانع اجرای کد JavaScript ای که attribute تگ را تغییر می‌دهد می‌شود و فایرباگ به خطی که باعث این عمل شده است در پنل Script منتقل می‌شود.
به عبارتی دیگر یک Break Point در خط JavaScript ای که باعث ویرایش attribute می‌شود قرار می‌دهد.

Break On Child Addition or Removal

مشابه توضیح قبل ، Break Point را در خطی که باعث اضافه/حذف شدن تگ Child شده است قرار می‌دهد.

Break On Element Removal

مشابه توضیح قبل ، Break Point را در خطی که باعث حذف شدن تگ شده است قرار می‌دهد.

Inspect in DOM Tab

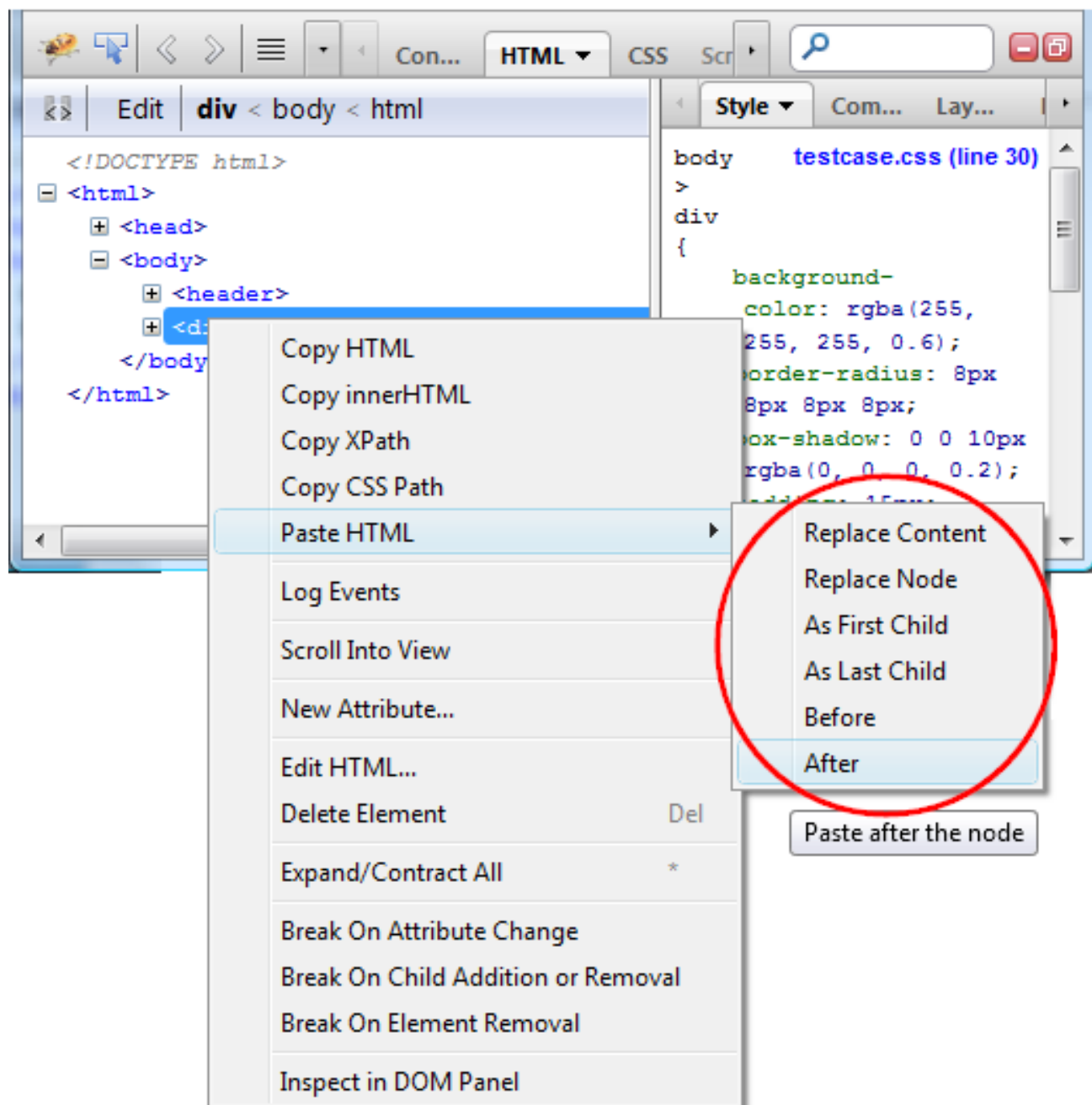
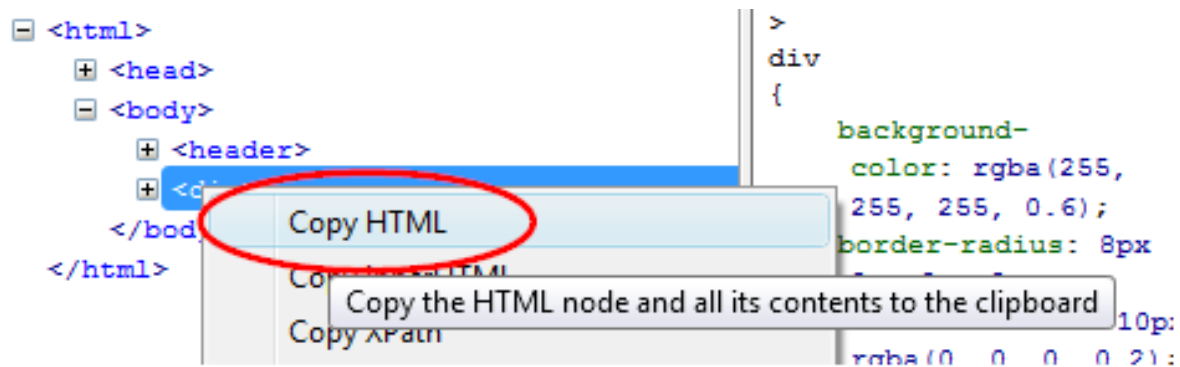
تگ فعلی را در پنل DOM ، برای بررسی باز می‌کند.

در [قسمت بعدی](#) با پنل‌های سمت راست (Side Panels) آشنا می‌شویم.

نظرات خوانندگان

نویسنده: احمد احمدی
تاریخ: ۱۳۹۱/۱۲/۲۵ ۱۵:۰

در [ورژن 1.11](#) از این افزونه ، امکانی برای paste کردن Html به صفحه ، به Context منوی پنل HTML اضافه شده که کار ویرایش محتویات Html را ساده‌تر کرده.
در حالی که قبلا برای این کار می‌بایست تگ را به حالت ویرایش درآورد و محتویات را اضافه کرد.



در [قسمت قبل](#) توضیحاتی در مورد تب HTML ارائه کردیم. در این قسمت توضیحات کاملی در مورد پنل‌های جانبی داخل پنل HTML می‌دهیم.

Side Panels

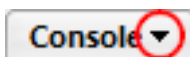
در پنل HTML در کنار ارائه امکاناتی برای مشاهده و کار با تگ‌های صفحه، اطلاعات و امکانات دیگری هم برای تگ انتخاب شده در قسمت NodeView وجود دارد. این امکانات در پنل‌هایی که سمت راست پنل اصلی قرار دارند گنجانده شده است که به ترتیب برای نمایش و ویرایش استایل‌ها، مشاهده استایل‌های محاسبه شده، مشاهده Layout یا آرایش و نمایش اطلاعات DOM تگ انتخاب شده در NodeView هستند.

Style - 1

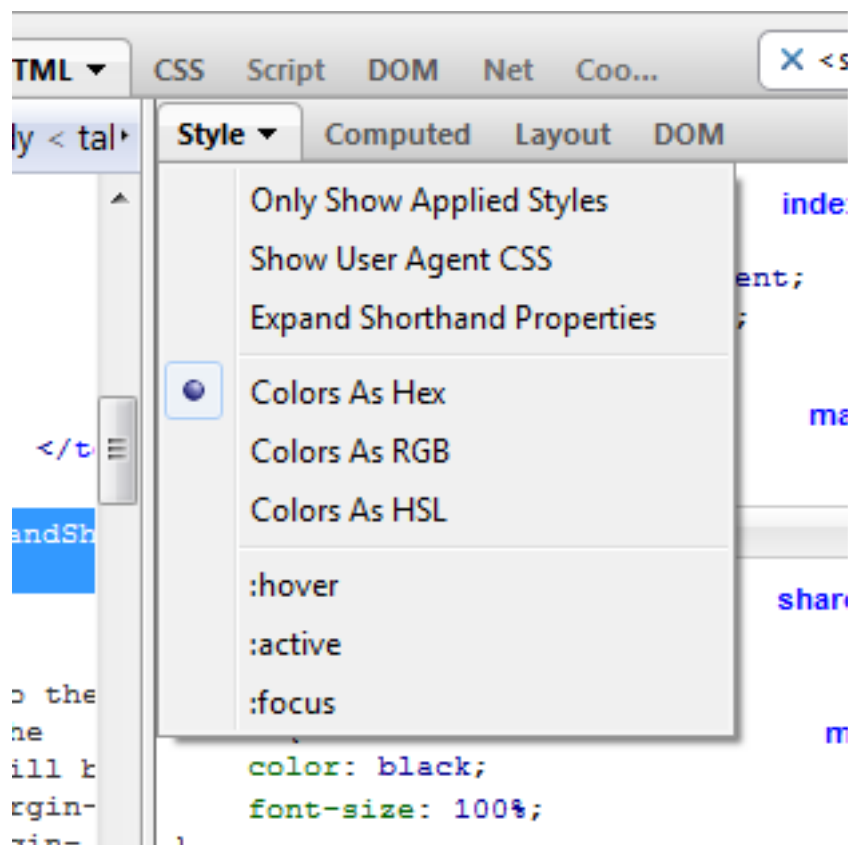
در این تب استایل‌هایی که در حال حاضر بروی تگ انتخاب شده اعمال شده اند، نمایش داده می‌شود. در صورتی که موس را بروی مقادیر استایل‌هایی که جلوه‌ی بصری دارند بگیرید، یک پاپ‌آپ کوچک نمایان می‌شود که مقدار را نمایش می‌دهد.

Options Menu

هر تب یا پنل در فایرباگ دارای یک سری تنظیمات است که Options Menu نام دارد. تب Style هم دارای یک سری تنظیمات است که دانستن آنها بسیار به شما کمک خواهد کرد. این منو با کلیک کردن بروی فلش تب (



) یا راست کلیک کردن بروی تب ظاهر می‌شود.



Only Show Applied Styles

در صورت انتخاب ، فقط استایل هایی که اعمال شده اند نمایش داده می شوند. (استایل های Overwrite شده نمایش داده نمی شوند.)

(این گزینه قابلیت خوبی است ، اما چندبار برای بنده پیش آمده که این مورد به اشتباه استایلی که اعمال شده بود را هم Overwrite شده در نظر گرفته بود. پس در هین طراحی استایل و کار با CSS اگر احیانا یکی از استایل هایتان وجود نداشت و از وجود آن اطمینان داشتید ، غیرفعال کردن این گزینه را امتحان کنید.)

Show User Agent CSS

با فعال کردن این گزینه ، استایل هایی که توسط مرورگر اعمال شده اند هم نمایش داده می شوند.

Expand Shorthand Properties

با فعال کردن این گزینه ، استایل هایی که بصورت کوتاه شده تعریف شده اند را بصورت گسترده و باز شده نمایش می دهد. برای مثال ، دستور margin را بصورت margin-top , margin-right , margin-bottom , margin-left نمایش می دهد.

سه گزینه ی Colors As RGB ، Colors As Hex و Colors As HSL تعیین کننده ی فرمت نمایش رنگ ها هستند.

سه گزینه ی :hover ، :active و :focus هم برای تعیین کلاس کاذب برای تگ جاری کاربرد دارند. برای مثال شما می خواهید استایلی که یک لینک زمان موس برویش قرار دارد را بررسی کنید ، لینک را در NodeView انتخاب می کنید و سپس از گزینه ی :hover را فعال می کنید.

Panel

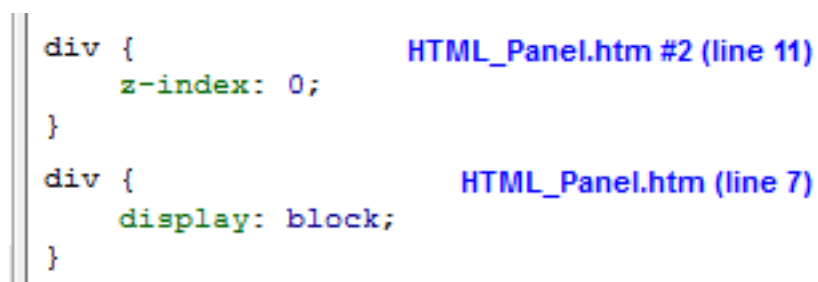
Element styles

استایل هایی که بصورت inline (در خود تگ) تعریف شده اند هم در این قسمت نمایش داده می شود و نام rule آن element.style است.



Source Links

در بالا-راست هر بخش ، یک لینک قرار دارد که لینک فایل استایلی است که در همان قسمت وجود دارد و عددی که در پرانتز قرار دارد ، شماره خط استایل در همان فایل است.
اگر نام فایل با نام صفحه‌ی جاری برابر باشد ، به معنی وجود استایل در تگ <style> در صفحه‌ی جاری است و شماره‌ی بعد از # هم ایندکس تگ <style> است.
(با کلیک بروی لینک فایل ، فایل در خط مورد نظر در پنل CSS نمایش داده می شود.)



Inherited rules

rule های به ارث رسیده هم در قسمت های جداگانه به همراه استایل های به ارث رسیده نمایش داده می شود و تگی والد که استایل ها از آن به ارث رسیده اند هم در قسمت عنوان همان استایل ها نمایش قرار داده شده است. (با کلیک بروی آن ، در قسمت Nodeview انتخاب می شود.)

Inline editing

استایل‌های نمایش داده شده در این پنل را براحتی و با کلیک کردن بروی نام یا مقدار هر یک از دستورات می‌توانید تغییر دهید. برای نوشتن دستورات و مقادیر آن‌ها می‌توانید از پیشنهادهای فایرباگ هم کمک بگیرید و با دکمه‌های Arrow Up و Arrow Down هم بین مقادیر مجاز حرکت کنید.

دستورات یا مقادیر نا صحیح در هین تایپ ، با رنگ قرمز و مقادیر صحیح با رنگ سبز مشخص می‌شوند. (این امکان خیلی مفید است ، برای مثال می‌خواهید فونت‌های مختلف را برای یک استایل امتحان کنید ، دستور font-family را می‌نویسید و بعد از زدن Enter ، با دکمه های Arrow Up و Arrow Down در لحظه نتیجه‌ی اعمال فونت‌های مختلف و در دسترس را مشاهده می‌کنید و بهترین را بر می‌گزینید. یا برای یافتن بهترین مقدار margin ، بعد از دستور margin ، زدن کلید Enter ، وارد کردن یک عدد برای شروع ، می‌توان باز هم با دکمه های Arrow Up و Arrow Down به سرعت تغییر را در صفحه مشاهده کرد.)

Rendered font highlighted

برای دستور font ، فایرباگ هوشمندانه عمل کرده و فونتی که در حال استفاده است را پررنگ می‌کند. این امکان برای یافتن خطاهای متداول هنگام تعریف فونت‌های غیر سیستمی ، بسیار مفید است.

```
#headermenu li {                                css?v=...HIE5j41 (line
border-left: 1px solid rgba(255, 255, 255, 0.5
display: inline-block;
font: bold 16px "b koodak",arial,sans-serif;
```

Context Menu

این منو زمانی که در پنل راست کلیک کنید ظاهر می‌شود و نسبت به منطقه (Context) ای که در آن راست کلیک کرده اید ، گزینه‌های متفاوتی را مشاهده خواهید کرد. در جدول زیر ، گزینه‌ها ، Contextشان و توضیح هر گزینه آمده است.

گزینه	Context	توضیحات
Copy Rule Declaration	CSS selector	CSS Rule فعلی را به همراه استایل هایش در clipboard کپی می‌کند
Copy Style Declaration	CSS selector	استایل‌های CSS Rule فعلی را در clipboard کپی می‌کند
Copy Location	source link	آدرس فایل تعریف CSS Rule را در clipboard کپی می‌کند
Open in New Tab	source link	آدرس فایل تعریف CSS Rule را در تب جدید باز می‌کند
Edit Element Style...	everywhere	امکان تعریف استایل‌های درون تگ (inline) را محیا می‌کند
Add Rule...	everywhere	یک Rule جدید ایجاد می‌کند CSS Rule هایی که در حال حاضر وجود دارد را هم پیشنهاد می‌دهد
Delete "<rule name>"	CSS selector	CSS Rule فعلی را حذف می‌کند

گزینه	Context	توضیحات
New Property...	CSS rule	یک استایل جدید به CSS Rule فعلی اضافه می‌کند
Edit "<property name>"...	CSS property	Property فعلی را به حالت ویرایش می‌برد راه دیگر ویرایش Property ، کلیک بروی آن است
Delete "<property name>"	CSS property	Property فعلی را حذف می‌کند
Disable "<property name>"	CSS property	Property فعلی را غیر فعال می‌کند را سریع تر ، کلیک کردن در ناحیه‌ی پشت Property ، بروی علامت قرمز رنگ است
Refresh	everywhere	محتویات پنل را بروز می‌کند
Inspect in DOM Panel	CSS rule	CSS Rule فعلی را در پنل DOM برای بررسی باز می‌کند
Inspect in CSS Panel	CSS rule	CSS Rule فعلی را در پنل CSS برای بررسی باز می‌کند
<Default Editor Name>	CSS rule	فایل تعریف استایل‌ها را در ادیتور تعریف شده باز می‌کند (این گزینه در صورت تعریف ادیتور در تنظیمات FireBug نمایش داده خواهد شد)

Computed - 2

در این تب نتیجه‌ی پردازش استایل‌های ارائه شده توسط کاربر ، برای تگ مشخص شده در قسمت NodeView نمایش داده می‌شود. (مقادیر استایل‌هایی که در نهایت بروی تگ اعمال شده اند.)

Style	Computed ▼	Layout	DOM
[-] Text			
[-] font-family			
	"Trebuchet MS",helvetica,sans-serif		
.editsection	"Trebuchet MS",helvetica,sans-serif		index.....e=18000 (line 62)
body	sans-serif		main.css?270 (line 42)
body	"Trebuchet MS",helvetica,sans-serif		index.....e=18000 (line 14)
[-] font-size			
	10px		
@element.style	10px		
h3.editsection	76%		main.css?270 (line 133)
#globalWrapper	127%		main.css?270 (line 51)
h3	132%		main.css?270 (line 132)
body	x-small		main.css?270 (line 42)
h3	138%		index.....e=18000 (line 44)

Style Tracing

برای ردیابی استایل‌ها ، استایل‌ها به ترتیب اعمال شدنشان مرتب شده اند و اولین مقدار ، مقداری است که اعمال شده است. مقادیر Overwrite بصورت خط کشیده شده و استایل‌های Overwrite شده بصورت خاکستری-کمرنگ نمایش داده می‌شوند. هر استایل هم مانند تب Style ، یک لینک به منبع خود دارد.

Options Menu

Show User Agent CSS

در صورت انتخاب ، فقط استایل‌هایی که اعمال شده اند نمایش داده می‌شوند.

Sort alphabetically

در صورت انتخاب ، استایل‌ها به ترتیب الفبا ، و در صورت عدم انتخاب بصورت گروه بندی نمایش داده می‌شوند.

Show Mozilla Specific Styles

در صورت انتخاب ، استایل‌های مخصوص Mozilla را نمایش می‌دهد. (استایل‌هایی با پیشوند -moz-)

سه گزینه‌ی Colors As RGB ، Colors As Hex و Colors As HSL تعیین کننده‌ی فرمت نمایش رنگ‌ها هستند.

Context Menu

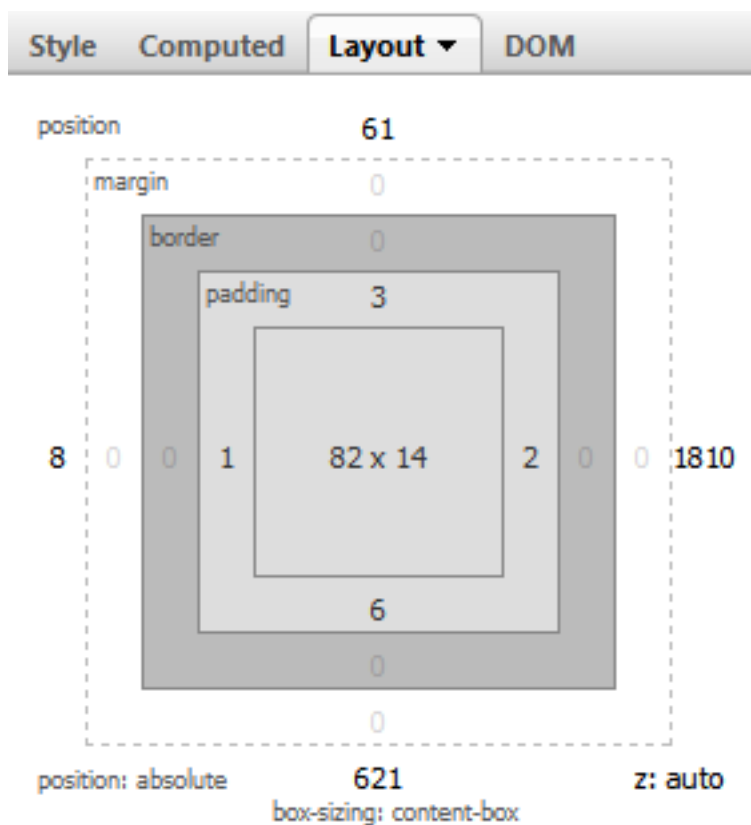
این منو زمانی که در پنل راست کلیک کنید ظاهر می‌شود و نسبت به منطقه (Context) ای که در آن راست کلیک کرده اید ، گزینه‌های متفاوتی را مشاهده خواهید کرد. در جدول زیر ، گزینه‌ها ، Context‌شان و توضیح هر گزینه آمده است.

گزینه	Context	توضیحات
Expand All Styles	everywhere	CSS Rule فعلی را به همراه استایل‌هایش در clipboard کپی می‌کند
Collapse All Styles	everywhere	استایل‌های CSS Rule فعلی را در clipboard کپی می‌کند
Inspect in DOM panel	styles	آدرس فایل تعریف CSS Rule را در تب جدید باز می‌کند
Copy Location	style source link	امکان تعریف استایل‌های درون تگ (inline) را محیا می‌کند
Open in New Tab	style source link	یک Rule جدید ایجاد می‌کند CSS Rule‌هایی که در حال حاضر وجود دارد را هم پیشنهاد می‌دهد
Inspect in CSS panel	style source link	CSS Rule فعلی را حذف می‌کند
<Default Editor Name>	style source link	فایل تعریف استایل‌ها را در ادیتور تعریف شده باز می‌کند (این گزینه در صورت تعریف ادیتور در تنظیمات FireBug نمایش داده خواهد شد)

Layout - 3

در این تب ، مقادیر Box Model بصورت بصری نمایش می‌دهد. می‌توان با کلیک کردن بروی هریک از مقادیر ، آن را ویرایش کرد. (این تغییر بصورت inline در تگ اعمال می‌شود).

با حرکت موس بروی قسمت‌های مختلف ، می‌توان همان قسمت‌ها را در صفحه بصورت خط کشی شده مشاهده کرد. (البته ظاهراً در ورژن 1.10.4 که بنده استفاده می‌کنم ، عملیات ویرایش مقادیر به درستی انجام نمی‌شود).



Options Menu

Show Rulers and Guides

در صورت انتخاب ، خط‌های راهنما را هنگام حرکت موس بروی اجزای Box Model در صفحه نمایش می‌دهد.



این پنل اطلاعات DOM تگ جاری را نمایش می‌دهد.
این پنل تمام قابلیت‌های [پنل DOM](#) اصلی را دارا می‌باشد.
(در مقالات آینده با تب DOM آشنا خواهیم شد.)

Style	Computed	Layout	DOM ▼
accessKey			""
accessKeyLabel			"Alt+Shift+"
contextMenu			null
dataset			DOMStringMap { }
isContentEditable			false
itemId			""
+ itemProp			{ length=0, value more... }
+ itemRef			{ length=0, value more... }
itemScope			false
+ itemType			{ length=0, value more... }

تو این مطلب میخوام روش ساخت arrow رو با css آموزش بدم. حتما تا بحال modal box و خیلی دیگه از افزونه‌های jQuery رو دیدید، حتی bootstrap هم از این ویژگی برای ساخت arrow در tooltip و popover هاش استفاده میکنه. [اینجا](#)

Users

Fields

[Reserved Credentials](#)

[Default User Associations](#)

☒ Terms of Use Required

☐ Autogenerate User Screen Names

به تصویر زیر نگاه کنید:



```
height: 20px;
width: 20px;
border: 10px solid #000;
border-top-color: #ccc;
```



```
height: 0;
width: 0;
```



```
border-color: transparent;
border-top-color: #ccc;
```

به طور خلاصه کل کار برای ساخت arrow کلاسی به صورت زیر هستش:
ابتدا خاصیت border برای یک مقدار اولیه با رنگ **transparent** مقداردهی شده و سپس مقدار border-top-color با #ccc مقداردهی شده و حتما عرض و طول باید با 0 مقداردهی بشن تا کادر اضافی حذف بشه، [به این صورت](#) :

```
.arrow{
  width:0;
  height:0;
  border:10px solid transparent;
  border-top-color:#ccc;
}
```

قرار دادن Arrow در کادرها !

مقدمات:

-حالا که دونستیم Arrowها چطور ایجاد میشن، این سوال پیش میاد که چطور از این Arrowها در ساخت tooltipها میشه استفاده کرد؟

CSS به ما این اجازه رو میده که با استفاده از خاصیت content محتوا ایجاد کنیم، البته نه هر محتوایی!، و چیزی که جالبه میتونیم دقیقا مثل یه عنصر باهاش رفتار کنیم و بهش استایل مورد نظر بدیم. خاصیت content رو فقط میشه با انتخابگرهای after و before به کار برد.

*****نکته‌ی بسیار بسیار مهم و حیاتی در طراحی : *****

اگه به خاصیت position عنصری مقدار relative داده بشه و در داخل این عنصر، عنصری (فرزند) با خاصیت position برابر absolute ایجاد کنیم، عنصر پدر به عنوان **مبدأ مختصات** برای عنصر فرزند عمل میکنه. (این نکته خیلی از جاها حلال مشکلات برای من بوده!)

خب حالا بریم سر موضوع اصلی، ابتدا استایل زیر رو به container مورد نظر میدیم:

```
#demo {
background-color: #333;
height: 100px;
position: relative;
width: 100px;
}
```

کاملا واضحه و نیاز به توضیح اضافی نداره، فقط به relative بودن عنصر دقت کنید . حالا استایل پایه رو برای ایجاد Arrow برای این container ایجاد می‌کنیم:

```
#demo:after {
content: ' ';
height: 0;
position: absolute;
width: 0;
}
```

در اینجا توجه به دو نکته ضروریه:

1-ما فقط یک space به عنوان محتوا به این استایل دادیم، در واقع همینم کافیه چون ما فقط می‌خوایم یه بهانه داشته باشیم که با استایل دادن بهش به هدف برسیم، اگه content رو اصلا قید نکنیم، هر استایلی که ایجا بشه نادیده گرفته میشه چون نمیشه هوا رو رنگ آمیزی کرد!

2-ما خاصیت position این عنصر رو به absolute ست کردیم، پس میتونیم این عنصر رو نسبت به container جابجا کنیم.

حالا به عنصر استایل Arrow رو میدیم:

```
#demo:after {
content: ' ';
height: 0;
position: absolute;
width: 0;

border: 10px solid transparent;
border-top-color: #333;
}
```

نکته‌ی دیگه ای که تو اینجا باید بهش توجه داشت اینه که پهنای Arrow توسط border مشخص میشه، که در اینجا 20px هستش! بله 20px، در واقع 10px مربوط یک گوشه‌ی شیب‌دار Arrow و 10px مربوط به گوشه‌ی دیگه میشه.

حالا که Arrow رو ایجاد کردیم و بعد از عنصر اصلی قرار دادیم، می‌خوایم یکم جابجاش کنیم تا به نتیجه مورد نظر برسیم:

```
#demo:after {
content: ' ';
height: 0;
position: absolute;
width: 0;
}
```

```
border: 10px solid transparent;
border-top-color: #333;

top: 100%;
left: 10px;
}
```

همونطور که میبینید خاصیت top به 100% که باعث میشه در پایینترین نقطه‌ی عنصر اصلی (مبدأ!) قرار بگیره، و خاصیت left برابر 10px ست شده که از عنصر اصلی 10px به سمت راست حرکت کردیم (فاصله از چپ 10px) و Arrow رو در این نقطه نقاشی کردیم!

[نتیجه نهایی تا اینجا کار](#)

نظرات خوانندگان

نویسنده:

محسن کریمی

تاریخ:

۱۸:۵۶ ۱۳۹۱/۰۹/۱۰

تشکر

مفید بود

نویسنده:

امیر هاشم زاده

تاریخ:

۱۷:۳۲ ۱۳۹۲/۱۰/۲۹

قصد ادامه سایر قسمت ها رو ندارید؟

همان طور که می‌دانید در css امکان استفاده از فونت‌های فارسی مهیاست. برای این کار کافیت با دستور زیر فونت را در فایل css خود تعریف کنیم و در صورتیکه فونت روی سیستم کاربر موجود نباشد ابتدا فونت روی سیستم دانلود شده و سپس نمایش داده می‌شود. استفاده از سه پسوند مختلف نیز برای مرورگرهای مختلف در نظر گرفته شده است تا خروجی در تمامی مرورگرها به درستی نمایش داده شود.

```
@font-face {
    src: url('Font/BYekan.eot?#') format('eot'), /* IE6-8 */
         url('Font/BYekan.woff') format('woff'), /* FF3.6+, IE9, Chrome6+, Saf5.1+ */
         url('Font/BYekan.ttf') format('truetype'); /* Saf3-5, Chrome4+, FF3.5, Opera 10+ */
}
```

در یک پروژه با مشکل عجیبی روبرو شدم. پروژه روی لوکال به خوبی کار می‌کرد اما بعد از آپلود روی سرور فونت‌ها بسیار دیر لود می‌شد. یعنی ابتدا تمامی اجزای صفحه لود شده و با تاخیری چند ثانیه ای فونت نمایش داده می‌شد. مرورگری که در حال کار روی آن برای تست پروژه بودم باید از پسوند woff مربوط به آن فونت استفاده می‌کرد.

بعد از تست پروژه با فایر باگ متوجه شدم که متاسفانه اصلا فونت روی سیستم دانلود نشده و خروجی 404 که مربوط به File Not Found است برگشت داده می‌شود. سعی کردم با دادن آدرس فونت در برنامه download manager فونت را دانلود کنم. اما باز هم فونت دانلود نمی‌شد! (یعنی فایل روی سرور موجود بود ولی هیچ دسترسی به آن ممکن نبود) برای دو پسوند دیگر یعنی eot و ttf هم این کار را تست کردم و متوجه شدم روی این دو پسوند مشکل خاصی وجود ندارد. با تماس با پشتیبانی هاست متوجه شدم اصلا این پسوند در Mime Type های سرور تعریف نشده است. بنابراین به صورت دستی Mime Type مورد نظر را روی سرور تعریف کردم و مشکل حل شد:

Home Folder	Virtual Dirs	Secured Folders	Extensions	Custom Errors	Headers	Web Publishing	MIME Types
Add MIME							
Extension		MIME-Type					
.woff		application/x-font-woff					
Update		Cancel		Delete			

همچنین لیست کامل Mime Types ها در آدرس زیر موجود است:

[لیست کامل Mime Types](#)

نظرات خوانندگان

نویسنده: آموزش مجازی
تاریخ: ۱۵:۲۷ ۱۳۹۱/۱۲/۲۹

شما اگر این تنظیمات رو توی پروژه خودت انجام بدی ، احتیاجی نیست برای IIS حتما MimeType تعریف کنی. البته فقط قسمتهایی از این تنظیمات در مورد مساله ای که گفتی لازم هست اما من برای ایده گرفتن در مورد تنظیمات IIS 7 و به بالا در web.config که کلا مهجور مونده ، تنظیماتی که خودم معمولا می‌ذارم رو گذاشتم اینجا.

```
<system.webServer>
  <urlCompression doStaticCompression="true" doDynamicCompression="true" />
  <staticContent>
    <remove fileExtension=".js"/>
    <mimeMap fileExtension=".js" mimeType="text/javascript" />
    <mimeMap fileExtension=".mp4" mimeType="video/mp4" />
    <mimeMap fileExtension=".m4v" mimeType="video/m4v" />
    <mimeMap fileExtension=".ogg" mimeType="video/ogg" />
    <mimeMap fileExtension=".ogv" mimeType="video/ogg" />
    <mimeMap fileExtension=".webm" mimeType="video/webm" />
    <mimeMap fileExtension=".oga" mimeType="audio/ogg" />
    <mimeMap fileExtension=".spx" mimeType="audio/ogg" />
    <mimeMap fileExtension=".svg" mimeType="images/svg+xml" />
    <mimeMap fileExtension=".svgz" mimeType="images/svg+xml" />
    <mimeMap fileExtension=".eot" mimeType="application/vnd.ms-fontobject" />
    <mimeMap fileExtension=".otf" mimeType="font/otf" />
    <mimeMap fileExtension=".woff" mimeType="font/x-woff" />
  </staticContent>
```

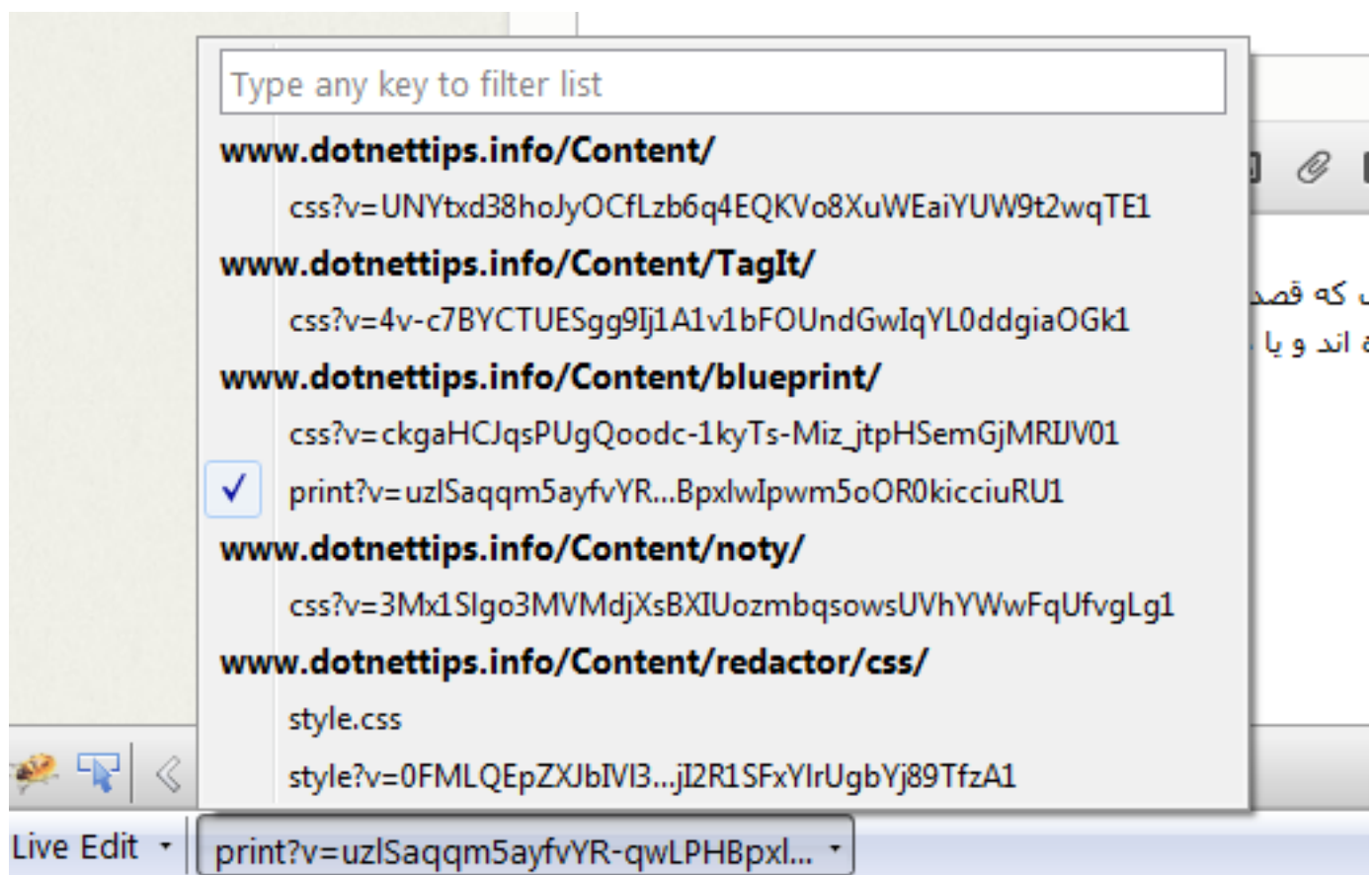
نویسنده: آرمان فرقانی
تاریخ: ۱۶:۳۹ ۱۳۹۱/۱۲/۲۹

البته با توجه به پردازش سلسه مراتبی فایل‌های config در وب سرور احتمال دارد در برخی میزبان‌های اشتراکی برخی از قابلیت‌ها توسط مدیر سرور Lock شود.

نویسنده: ناصر فرجی
تاریخ: ۱۵:۴۱ ۱۳۹۲/۰۱/۰۱

بله. تمامی تنظیمات iis رو به نوعی میشه از طریق web.config هم انجام داد ولی برای اعمال سراسری تنظیمات بهتره از iis استفاده بشه.

پنل CSS مانند پنل جانبی Style که در [مقاله‌ی قبل](#) بررسی کردیم است با این تفاوت که امکانات بیشتری برای افرادی که قصد تصریف استایل دارند محیا کرده است. در این پنل می‌توان به اضافه، ویرایش و حذف استایل‌هایی که به صفحه‌ی جاری توسط فایل‌های مختلف اضافه شده‌اند و یا داخل خود صفحه تعریف شده‌اند پرداخت.



```
body {
    background: none repeat scroll 0 center transparent;
    color: #000000;
    font-family: Tahoma;
    font-size: 10pt;
    line-height: 1.5;
}

.container {
    background: none repeat scroll 0 center transparent;
}
```

همچنین در این پنل امکان ویرایش یک فایل CSS بصورت کامل وجود دارد، به این صورت که می‌توانید تمام محتویات فایل مورد نظر را در یک Text area ویرایش کنید.

مطابق با روالی که در قسمت قبل پیش گرفتیم عمل می‌کنیم و به ترتیب به تشریح ابزار پنل ، خود پنل ، منوی راست کلیک و پنل جانبی Elements می‌پردازیم.

Options Menu

با راست کلیک کردن بروی تب CSS و یا کلیک کردن بروی فلش کوچک روی تب CSS قابل دسترس است و امکانات زیر را محیا می‌کند:

Expand Shorthand Properties : نمایش دستورات css بصورت کامل ، به این معنی که دستوراتی مانند margin که هم بصورت خلاصه و هم بصورت کامل تعریف می‌شوند را بصورت کامل نمایش می‌دهد. مثلا margin را چهار مقدار margin-top , margin-right , margin-bottom , margin-left نمایش می‌دهد.

Color As Hex , Color As RGB , Color As HSL : با انتخاب یکی از سه مقدار ذکر شده ، فرمت نمایش رنگ‌ها به حالت انتخاب شده تغییر می‌کند.

مثلا دستور color : #000000 به این صورت نمایش داده می‌شود : color : rgb(0, 0, 0)

Refresh : این گزینه محتویات پنل را بروز می‌کند.

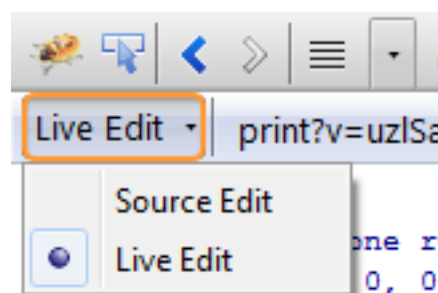
Panel Toolbar

ابزاری موجود در این پنل مشابه پنل‌های دیگر در بالای پنل و در زیر تب پنل‌ها قرار دارد و شامل ابزارهای زیر می‌شود:

Edit Button : این ابزار به دو صورت Live Edit و Source Edit در دسترس هست که با کلیک بروی فلش کوچکی که در سمت راست دکمه قرار دارد می‌توان نوع آن را تغییر داد.

انتخاب حالت Source Edit ، باعث می‌شود سورس فایل به همان صورتی که به مرورگر ارسال شده نمایش داده شود. (کامنت‌ها ، فرمت فایل و ... حفظ می‌شود.)

حالت Live Edit هم باعث نمایش محتویات فایل بصورت مرتب شده می‌شود. (کامنت‌ها و دستوراتی که توسط مرورگر تفسیر نشده اند نمایش داده نمی‌شوند.)

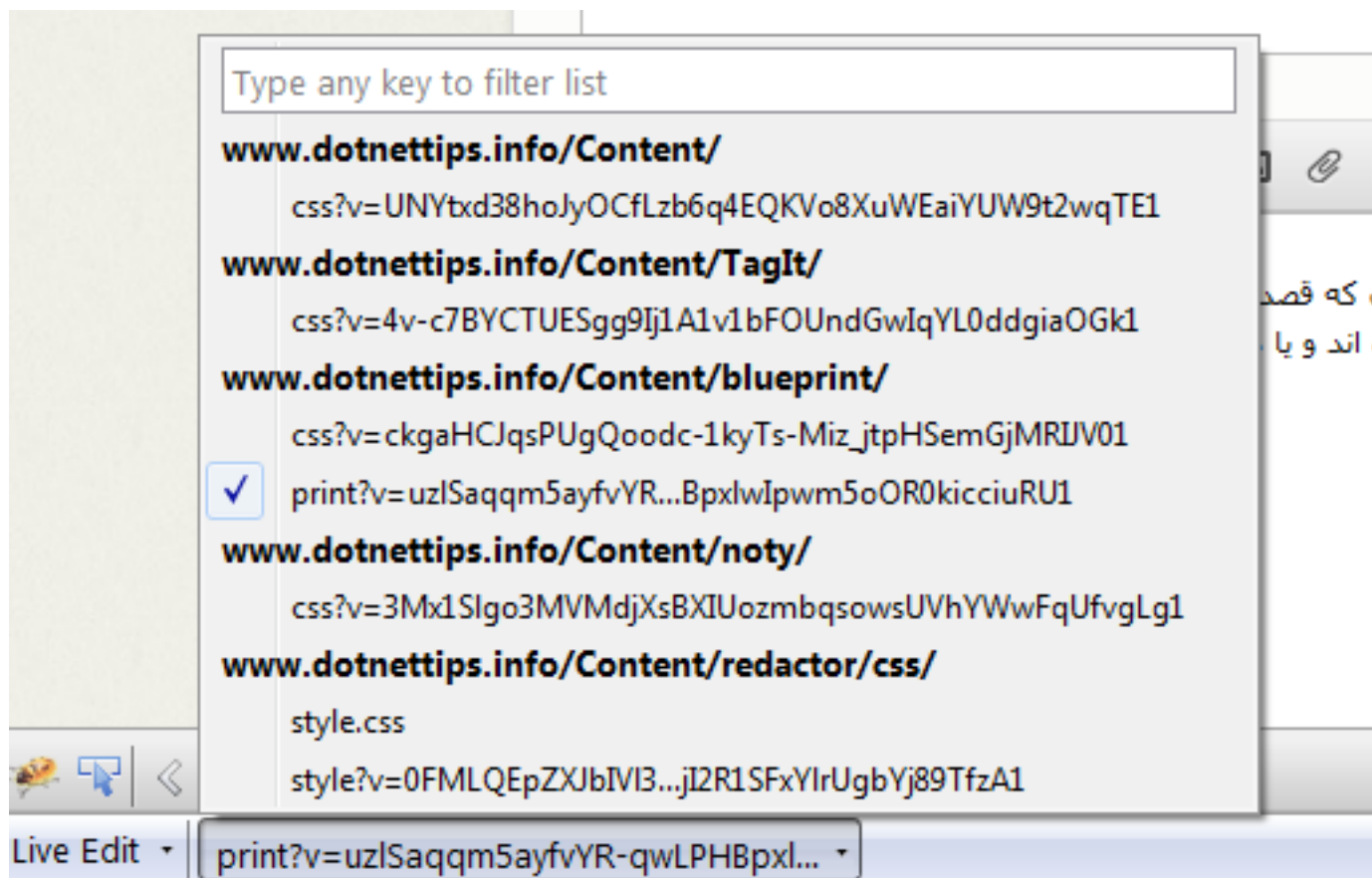


CSS Location Menu : نام فایل جاری که در پنل در حال نمایش است را نمایش می‌دهد و همچنین با کلیک بروی آن ، فایل‌های استایل دیگری که در صفحه بارگزاری شده اند را نمایش می‌دهد. با کلیک بروی هرکدام از فایل‌های نمایش داده شده ، همان فایل در پنل باز می‌شود.

استایل‌هایی که در خود صفحه تعریف شده اند با نام خود صفحه در این قسمت نمایش داده می‌شوند و اگر استایل‌های در چندین تگ style تعریف شده باشند ، دومین تگ با #2 ، سومین با #3 و ... مشخص می‌شوند.

هنگام باز بودن :

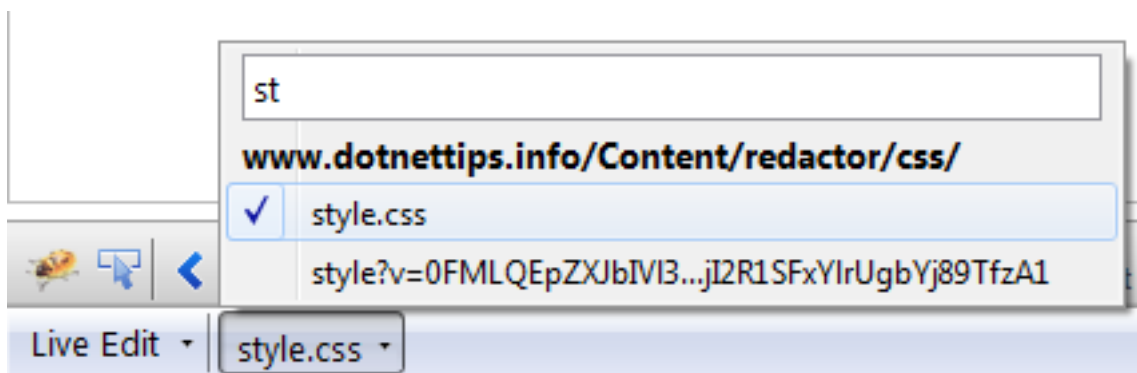
1- فایل هایی که از پوشه های مختلف در صفحه بارگذاری شده اند ، با نام پوشه از هم تفکیک شده اند:



```
body {
    background: none repeat scroll 0 center transparent;
    color: #000000;
    font-family: Tahoma;
    font-size: 10pt;
    line-height: 1.5;
}

.container {
    background: none repeat scroll 0 center transparent;
}
```

2- با تایپ کردن عبارت دلخواه می توان نتایج را محدود کرد:



Panel

منظور از پنل ، قسمتی هست که استایل بصورت فرمت شده و نمایش داده شده است.

Infotips

توسط این قابلیت ، زمانی که موس را بروی آدرس تصاویر ، نوع فونت و ... ببرید ، پاپ آپ کوچکی باز می شود و اطلاعاتی در مورد مقادیر می دهد. مثلا اینکه آیا تصویر مورد نظر به درستی بارگذاری شده یا نه ، نمای کوچکی از تصویر ، شکل فونت و ...

Editing rules

برای ویرایش تعاریف CSS شامل Selector ها ، دستورات و مقادیر ، این پنل ابزارهای مفیدی ارائه می کند. برای ویرایش یک selector ، دستور یا مقدار آن بروی آن عبارت کلیک کنید ، در همین حال یک text box ظاهر می شود و می توانید مقدار جدید را وارد کنید. پس از انجام ویرایش مورد نظر بروی قسمتی از صفحه کلیک کنید یا کلید Tab سپس Esc را بزنید. برای ویرایش دستورات بعدی ، کلید Tab را بزنید و برای لغو تغییر جاری ، کلید Esc را بزنید.

برای ایجاد یک rule جدید ، بروی قسمتی از پنل راست کلیک کرده و سپس گزینه ی " New Rule ... " را برگزینید. برای ایجاد یک property هم چند راه وجود دارد: بروی قسمت از فضای خالی تعریف یک استایل دوبار کلیک کنید. در قسمتی از تعریف یک استایل راست کلیک کرده و گزینه ی " New Property ... " را انتخاب کنید.

بروی مقدار آخرین property کلیک کرده و کلید Tab را بزنید.

هنگام ویرایش یا ایجاد یک Property ، Rule یا مقدار بصورت inline ، حاشیه ی text box متناسب با مقدار وارد شده رنگی را که نمایانگر حالت ذخیره ی مقدار وارد شده است نمایش می دهد. برای مثال اگر مقداری که برای یک selector وارد شده است نامعتبر باشد ، رنگ حاشیه ی text box **قرمز** می شود. جدول زیر حالت های مختلف را شرح می دهد:

رنگ حاشیه	Selector ها	نام Property ها	مقادیر Property ها و Rule ها
خاکستری	تغییری ایجاد نشده است	تغییری ایجاد نشده است	تغییری ایجاد نشده است
قرمز	selector نامعتبر است	نام نامعتبر است	مقدار نامعتبر است
زرد	selector صحیح است اما بروی Element فعلی تاثیر ندارد	نام صحیح است اما مقدار Property غیر مجاز است یا وارد نشده است	
سبز	selector صحیح است	نام و مقدار صحیح هستند	مقدار صحیح است

Auto-completion

زمانی که در حال ایجاد/ویرایش کردن یک Rule, Property یا مقدار آنها هستید ، میتوانید از این قابلیت استفاده کنید. مثلا با وارد کردن # تمام Selector هایی که می‌توانند با # شروع شوند در دسترس شما هستند و با دکمه‌های Up/Down می‌توانید مقادیر ممکن را مرور کنید.

اگر در حال ویرایش یک مقدار عددی هستید ، می‌توانید با دکمه‌های Up/Down مقادیر را یک واحد یک واحد افزایش دهید. با نگه داشتن کلید Shift و فشردن Up/Down می‌توانید مقادیر را 10 تا 10 تا تغییر دهید و با نگه داشتن Ctrl بجای Shift ، یک دهم یک دهم.

Toggling styles

با این امکان می‌توانید یک Property را بطور موقت فعال/غیرفعال کنید. با حرکت موس از یک Property یک آیکن قرمز رنگ در کنار آن نمایش داده می‌شود که با کلیک بروی آن ، Property و مقدار آن کمرنگ شده و آیکن قرمز رنگ کنار آن ثابت می‌شود. با کلیک مجدد بروی آن ، Property فعال می‌شود.

```
table.wikitable {
  border-collapse: collapse;
}

table {
  color: black;
  font-size: 100%;
}
```

Context Menu

این منو زمانی که در پنل راست کلیک کنید ظاهر می‌شود و نسبت به منطقه (Context) ای که در آن راست کلیک کرده اید ، گزینه‌های متفاوتی را مشاهده خواهید کرد. در جدول زیر ، گزینه‌ها ، Context شان و توضیح هر گزینه آمده است.

گزینه	Context	توضیحات
Copy Location	CSS Location Menu	آدرس فایل استایل را در حافظه کپی می‌کند.
Open in New Tab	CSS Location Menu	فایل استایل را در یک تب جدید باز می‌کند.
Copy Image Location	image values	آدرس تصویر را در حافظه کپی می‌کند.
Open Image in New Tab	image values	تصویر را در یک تب جدید باز می‌کند.
Copy Color	color values	مقدار رنگ را در حافظه کپی می‌کند.
Copy Rule Declaration	CSS selector	Property و Selector ها را در حافظه کپی می‌کند.
Copy Style Declaration	CSS selector	فقط Property ها را در حافظه کپی می‌کند.
New Rule...	همه جای پنل	یک Rule جدید بالای قسمتی که راست کلیک شده ایجاد می‌کند.
Delete "<selector>"	CSS selector	Rule را حذف می‌کند.

گزینه	Context	توضیحات
New Property...	CSS rule	یک Property جدید در Rule ی که در آن راست کلیک شده ایجاد می کند.
Edit "<property name>"...	CSS property	Property فعلی به حالت ویرایش در می آید. (راه ساده تر ، کلیک بروی Property است.)
Delete "<property name>"...	CSS property	Property فعلی را حذف می کند.
Disable "<property name>"...	CSS property	Property فعلی را غیرفعال می کند.
Refresh	همه جای پنل	محتویات پنل را بروز رسانی می کند.
Inspect in DOM panel	CSS Location Menu, CSS rule	فایل استایل یا Rule را در پنل DOM باز می کند.

Elements Side Panel

در این پنل که سمت راست پنل CSS قرار دارد ، با وارد کردن یک CSS Selector می توانید Element هایی که در صفحه با آن مطابقت دارند را مشاهده کنید.

برای وارد کردن CSS Selector هم می توان مقدار مورد نظر را در قسمت Try a selector ... وارد کرد هم می توان بروی یکی از Selector های پنل راست کلیک کرد و گزینه ی Get Matching Elements را برگزید.

Context Menu این قسمت هم مشابه Context Menu پنل HTML هست و فقط در ورژن 1.11 گزینه ی Paste HTML اضافه شده که در [این کامنت](#) از مقاله ی [آموزش فایرباگ - #5 - HTML Panel](#) توضیح داده شده است.

نظرات خوانندگان

نویسنده:

راضیه

تاریخ:

۲۳:۴۶ ۱۳۹۲/۰۱/۰۸

عالی بود

استفاده از CSS علاوه بر جذابیت و قابلیت‌های مفید آن، پیچیدگی‌هایی دارد و کدهای شما معمولاً طولانی می‌شود و هرچه کدها طولانی‌تر شوند، مدیریت آن نیز سخت‌تر می‌گردد. اما با استفاده از SASS، قابلیت‌هایی به CSS اضافه می‌شود که قبلاً وجود نداشت، از جمله استفاده از variable ها، نوشتن کدهای تو در تو (nesting) و ... با استفاده از SASS کدهای CSS کوتاه‌تر شده و در نتیجه سریع‌تر اجرا شوند. SASS با CSS3 سازگار است. همچنین امکان مشاهده فایل‌های آن (با پسوند .scss) توسط افزونه [Firesass For Firebug](#) وجود دارد.

دو syntax برای SASS وجود دارد: یکی Sassy CSS (SCSS) که شکل توسعه یافته CSS3 می‌باشد و دیگری که قدیمی‌تر است، Indented syntax می‌باشد که در آن به جای استفاده از براکت، از تورفتگی خط‌های کد استفاده می‌شود و همچنین از به جای استفاده از سمی کولن، باید به خط جدید بروید.

قابلیت‌های موجود در SASS :

Variables 1-

متغیرها امکان ایجاد تغییرات در کدهای CSS را بسیار راحت‌تر می‌سازند. به عنوان مثال یک متغیر برای یک کد رنگ دلخواه تعریف می‌کنید، از این به بعد به جای استفاده از کد رنگ در کدهای CSS، از متغیر تعریف شده برای آن بهره می‌گیرید، به این ترتیب، چنانچه در آینده نیاز به تغییر این کد رنگ داشته باشید، تنها با تغییر آن در متغیر، در کل فایل CSS تغییر ایجاد خواهد شد. برای تعریف متغیر، در ابتدای اسم دلخواه خود از علامت \$ استفاده کنید:

```
$myColor: #ff0000;
body {
  color: $myColor;
}
.box{
  Border-color:$myColor;
}
```

Nesting 2- یا selector های تو در تو:

می‌توانید selector ها را مانند کدهای html به صورت hierarchy تعریف کنید:

```
nav {
  ul {
    list-style: none;
  }
  li { display: inline-block; }
  a {
    text-decoration: none;
  }
}
```

کدهای بالا بعد از تولید شدن در مرورگر به شکل زیر دیده می‌شود:

```
nav ul {
  list-style: none;
}
nav li {
  display: inline-block;
}
```



```
nav a {
  text-decoration: none;
}
```

3- Partials :

می توانید قطعاتی از کدهای CSS را به صورت Partial SASS تعریف کنید و سپس آن را در فایل های SASS دیگر استفاده نمایید. همانند Partialview در MVC ، هنگام نام گذاری آن از _ در ابتدای نام استفاده نمایید. فایل partial SASS دارای پسوند SCSS می باشد : "myPartial.scss_"

برای استفاده از myPartial.scss_ در فایل sass دیگر ، از دایرکتیو @import استفاده کنید:

```
@import "myPartial"
```

همچنین می توانید نام چندین partial را پشت هم بنویسید:

```
@import "myPartial1","myPartial2"
```

نمونه کد:

```
/*_myPartial1.scss codes...*/
html,body,ul,ol {
  margin: 0;
  padding: 0;
}

/*_myPartial2.scss codes...*/
@import "myPartial1"

body, {
  background-color: #efefef;
}
```

کدها بعد از تولید در مرورگر:

```
html, body, ul, ol {
  margin: 0;
  padding: 0;
}
body {
  background-color: #efefef;
}
```

4- Mixins :

از آنجایی که استفاده و نوشتن بعضی property های CSS سخت می باشد، می توانید از روش mixin استفاده کرده و قطعه کدهایی را ایجاد کنید که بتوانید در کدهایتان از آنها بارها و بارها استفاده کنید. به عنوان مثال قطع کدی برای border-radius ایجاد کنید ، (همانطور که میدانید border-radius برای مرورگرهای مختلف ، حالت های مختلفی تعریف می شود). برای ایجاد mixin ، در ابتدای قطع کد از @mixin استفاده نمایید و برای استفاده از آن ، از @include استفاده نمایید:

```
@mixin cssProperty $yourCustomName{
...
Your css properties...
}
```

نمونه کد:

```
ایجاد mixin:
@mixin border-radius($radius) {
  -webkit-border-radius: $radius;
  -moz-border-radius: $radius;
  -ms-border-radius: $radius;
  -o-border-radius: $radius;
  border-radius: $radius;
}

mixin استفاده از:
.box { @include border-radius(10px); }
```

5- Extend/Inheritance :

@XETEND به شما این امکان را می‌دهد تا بخشی از Property های یک selector را برای استفاده در selector های دیگر به اشتراک بگذارید:

```
.message {
  border: 1px solid #ccc;
  padding: 10px;
  color: #333;
}
.success {
  @extend .message;
  border-color: green;
}
```

کدها بعد از تولید شدن به صورت زیر دیده می‌شوند:

```
.message, .success {
  border: 1px solid #cccccc;
  padding: 10px;
  color: #333;
}

.success {
  border-color: green;
}
```

6- Operators :

می‌توانید از عملگرهای ضرب و تقسیم و جمع و تفریق در کدهای CSS خود استفاده نمایید:

```
article[role="main"] {
  float: left;
  width: 600px / 960px * 100%;
}
```

نصب SASS :

حال که با SASS آشنا شدید ، انگیزه کافی برای دانستن روش نصب و استفاده آن خواهید داشت. برای استفاده از SASS می‌توانید از نرم افزارهایی که برای ویندوز ، مک و لینوکس وجود دارند، استفاده کنید از جمله این نرم افزارها :

[CodeKit](#) , [Compass.app](#) , [Hammer](#) , [Koala](#) , [LiveReload](#) , [Mixture](#) , [Prepros](#) , [Prepros](#)

روش دیگر استفاده از command line می‌باشد:

چنانچه سیستم عامل شما ویندوز می‌باشد، برای استفاده از sass ابتدا باید [ruby](#) را نصب نمایید. سپس در Cmd خط زیر را اجرا کنید:

```
gem install sass
```

چنانچه به خطایی برخوردید، ابتدا gem توسط sudo را نصب کنید:

```
sudo gem install sass
```

سپس توسط خط زیر چک کنید که SASS نصب شده است یا خیر:

```
sass -v
```

خط فوق، ورژن SASS نصب شده را برای شما می‌گرداند که نشان می‌دهد نصب SASS با موفقیت صورت گرفته است:

```
Sass 3.2.12 (Media Mark)
```

برای کسب اطلاعات بیشتر و روش نصب در سایر سیستم عاملها به این [لینک](#) مراجعه نمایید.

: SassScript

فایل SASS اسکریپتی برای اجرای یک سری از فانکشنها دارد، از جمله :

```
- rgb($red, $green, $blue) /* برای ایجاد کد رنگ */
```

برای مشاهده لیست کامل این فانکشنها به این [لینک](#) مراجعه کنید.

[منبع](#)

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۲۳:۵۲ ۱۳۹۲/۰۹/۱۰

- نگارش‌های اخیر VS.NET به صورت [توکار](#) LESS را پشتیبانی می‌کنند (از آپدیت دوم VS 2012 به بعد).
- برای SASS فعلا [Web Workbench](#) هست. افزونه معروف [Web essentials](#) هم [به زودی ...](#)

نویسنده: شهره مرتضوی
تاریخ: ۱۰:۵ ۱۳۹۲/۰۹/۱۲

از توضیح مفید و تکمیل کننده جنابعالی متشکرم.

نویسنده: ارش م
تاریخ: ۸:۴۳ ۱۳۹۳/۰۸/۲۶

آیا در نسخه 2013 ، SASS به صورت توکار وجود دارد؟

نویسنده: وحید نصیری
تاریخ: ۹:۱۰ ۱۳۹۳/۰۸/۲۶

بله. از [به روز رسانی دوم](#) آن به بعد. به روز رسانی [چهارم](#) را نصب کنید.

چنانچه قبلاً با فایل‌های Less کار کرده باشید، متوجه خواهید شد که به صورت پیش فرض و همانند فایل‌های css و js قابلیت افزوده شدن به Bundle.config را دارا نمی‌باشند. برای انجام این کار باید مراحل کوتاهی را طی نمایید:

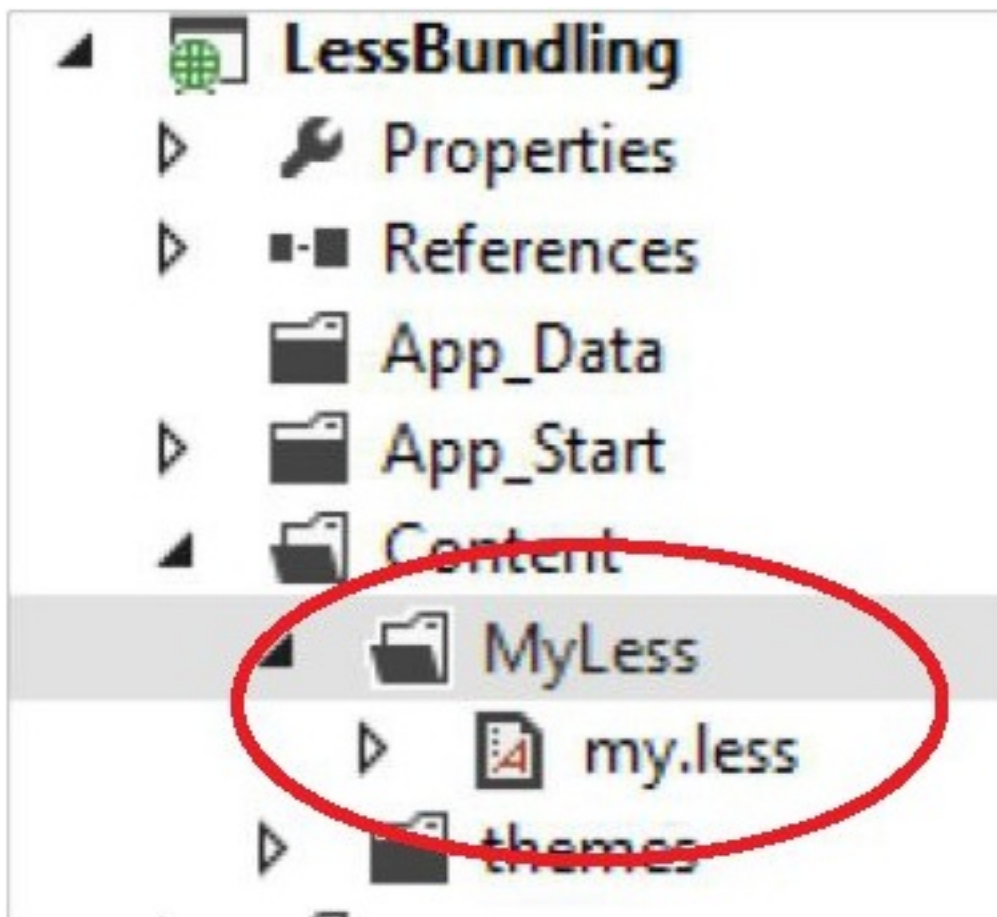
1- به منوی project و بخش Manage NuGet Packages ... رفته و dotless را جستجو و نصب نمایید.

2- کلاسی به نام "LessTransForm" ایجاد کنید که از "IBundleTransform" ارث بری کند، کدهای آن را به صورت ذیل تغییر دهید:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Optimization;

namespace LessBundling
{
    public class LessTransform:IBundleTransform
    {
        public void Process(BundleContext context, BundleResponse response)
        {
            response.Content = dotless.Core.Less.Parse(response.Content);
            response.ContentType = "text/css";
        }
    }
}
```

3- فولدري برای فایل‌های Less خود ایجاد کنید:



4- به App_Start/BundleConfig.cs رفته و کدهای ذیل در آن اضافه کنید:

```
var lessBundle = new Bundle("~/Content/Less").IncludeDirectory("~/Content/MyLess", "*.less");  
lessBundle.Transforms.Add(new LessTransform());  
lessBundle.Transforms.Add(new CssMinify());  
  
bundles.Add(lessBundle);
```

Bundling به اتمام رسید! حال می‌توانید از کد ذیل در view ها و یا در Layout خود، همانند فایل‌های css و js استفاده کنید:

```
@Styles.Render("~/Content/Less")
```

[منبع](#)

در صورتیکه بخواهید برای نسخه‌های مختلف اینترنت اکسپلورر style‌های مختلفی را بنویسید، کافی‌است از 4 کاراکتر منحصر به فرد (_ *) استفاده کنید.

IE8 and Below

برای اینکار در IE8 و پایین‌تر از آن، به انتهای هر استایل "9" را اضافه کنید. فقط "9" را می‌توان برای این کار استفاده کرد و تغییر دادن آن به عبارتی دیگر مثلاً "IE" اشتباه است. حتی "8" هم برای انجام اینکار درست نیست و فقط "9" کار می‌کند.

```
body {
  color: red; /* all browsers, of course */
  color : green\9; /* IE8 and below */
}
```

IE7 and Below برای اینکه در IE7 به طور مشخص، تغییری ایجاد کنید، کافی‌است ابتدای هر property، یک کاراکتر "*" اضافه کنید.

```
body {
  color: red; /* all browsers, of course */
  color : green\9; /* IE8 and below */
  *color : yellow; /* IE7 and below */
}
```

IE6 و برای اینکه در IE6 به طور مشخص، تغییری ایجاد کنید، کافی‌است ابتدای هر property یک کاراکتر "_ " استفاده کنید.





```
body {
  color: red; /* all browsers, of course */
  color : green\9; /* IE8 and below */
  *color : yellow; /* IE7 and below */
  _color : orange; /* IE6 */
}
```

class. 1-

این Selector تگ‌هایی را انتخاب می‌نماید که عضو یک کلاس خاص باشند.

```
<style>
  .first{ color: red}
  .content{color:blue}
</style>
<div class="first">Text 1</div>
<div>Text 2</div>
<p class="first">Text 3</p>
<div class="content">Text 4</div>
```

در مثال فوق Text 1 و Text 3 به رنگ قرمز و Text 4 به رنگ آبی نمایش می‌یابند. پشتیبانی در مرورگرها:






نسخه CSS	Selector					
1	.class	Yes	Yes	Yes	Yes	Yes

id# 2-

این Selector تگی را انتخاب می‌کند که دارای یک id خاص می‌باشد.

```
<style>
  #pass {
    color: red;
  }
</style>
<input type="password" id="pass"/>
```

در مثال فوق محتوای کادر رمز عبور به رنگ قرمز نمایش می‌یابد. پشتیبانی در مرورگرها:

نسخه CSS	Selector					
1	#id	Yes	Yes	Yes	Yes	Yes

E 3-

جهت انتخاب تگ‌ها براساس عنوان آنها استفاده می‌شود.

```
<style>
  div {
```








```

        color: red;
    }
</style>
<div>Text 1</div>
<p>Text 2</p>
<div>Text 3</div>

```

در مثال فوق Text 1 و Text 3 به رنگ قرمز نمایش می‌یابند. پشتیبانی در مرورگرها:

نسخه CSS	Selector					
1	E	Yes	Yes	Yes	Yes	Yes

* 4-


این Selector تمامی تگ‌ها را انتخاب می‌نماید

```

<style>
* {
    color: red;
}
</style>
<div>Text 1</div>
<input type="text" value="Text 2"/>
<select>
    <option>Option 1</option>
    <option>Option 2</option>
    <option>Option 3</option>
    <option>Option 4</option>
</select>

```

در مثال فوق رنگ متن تمامی تگ‌های فوق قرمز خواهد شد. پشتیبانی در مرورگرها:

نسخه CSS	Selector					
2	*	4.0	2.0	7.0	9.6	3.1

S1 S2 5-

تمامی المنت‌های S2 که فرزند S1 می‌باشند، انتخاب خواهند شد.






```

<style>
div span {
    color:red
}
</style>
<div>
    <h1>Text 1</h1>
    <span>Text 2</span>
</div>

```

```
<span>Text 3</span>
<div>Text 4</div>
</p>
<span>Text 5</span>
</div>
```

در مثال فوق Text 3، Text 2، و Text 5 به رنگ قرمز نمایش می‌یابند. پشتیبانی در مرورگرها:

نسخه CSS	Selector					
1	S1 S2	Yes	Yes	Yes	Yes	Yes

6- S1>S2

تمامی المنت‌های S2 که فرزند مستقیم S1 می‌باشند، انتخاب خواهند شد.

```
<style>
  div>span {
    color:red
  }
</style>
<div>
  <h1>Text 1</h1>
  <span>Text 2</span>
  <p>
    <span>Text 3</span>
    <div>Text 4</div>
  </p>
  <span>Text 5</span>
</div>
```

در مثال فوق Text 2 و Text 5 به رنگ قرمز نمایش می‌یابند. پشتیبانی در مرورگرها:

نسخه CSS	Selector					
2	S1>S2	Yes	Yes	7.0	Yes	Yes

7- S1+S2

تمامی المنت‌های S2 که هم تراز S1 و دقیقاً بعد از S1 قرار گرفته اند را انتخاب می‌نماید.

```
<style>
  h1+p {
    color: red;
  }
</style>
<div>
  <h1>Text 1</h1>
  <p>Text 2</p>
  <div>Text 3</div>
```

```
<p>Text 4</p>
<h1>
  <p>Text 5</p>
</h1>
<p>Text 6</p>
</div>
```

در مثال فوق Text 2 و Text 6 به رنگ قرمز نمایش می‌یابند. پشتیبانی در مرورگرها:

نسخه CSS	Selector					
2	S1+S2	Yes	Yes	7.0	Yes	Yes

8- S1~S2

تمامی المنت‌های S2 که هم تراز S1 می‌باشند و بعد از S1 گرفته اند را انتخاب می‌نماید.

```
<style>
  div ~ p {
    color: red;
  }
</style>
<div>
  <h1>Text 1</h1>
  <p>Text 2</p>
  <div>Text 3</div>
  <p>Text 4</p>
  <h1>
    <p>Text 5</p>
  </h1>
  <p>Text 6</p>
</div>
```

در مثال فوق Text 4 و Text 6 به رنگ قرمز نمایش می‌یابند. پشتیبانی در مرورگرها:

نسخه CSS	Selector					
3	S1~S2	4.0	3.5	7.0	9.6	3.2

9- S1!>S2

این Selector المنت‌های S1 را انتخاب می‌کند که والد مستقیم S2 می‌باشند.






```
<style>
  div!>span {
    border: 1px solid red;
  }
</style>
<div>
  <h1>Text 1</h1>
```

```

<div>
  <span>Text 2</span>
  <h1>Text 3</h1>
  <p>Text 4</p>
</div>
<div>Text 5</div>
</div>

```

در مثال فوق کادری قرمز رنگ را دور آن div که شامل محتوای Text 3، Text 2، و Text 4 می‌باشد، ترسیم خواهد نمود. **پشتیبانی در مرورگرها:**

نسخه CSS	Selector					
4	S1!>S2	No	No	No	No	No

10- S1 /attribute/ S2



تمامی المنت‌های S2 انتخاب می‌شوند که توسط یک attribute از S1 به id المنت S2 ارجاع داده است.

```

<style>
  label /for/ input {
    color: red;
  }
</style>
<label for="user">User Name:</label>
<input type="text" id="user"/>
<label>Password:</label>
<input type="password" id="pass"/>

```

در مثال فوق، اولین تگ input که id آن user می‌باشد و توسط تگ label و با استفاده از ویژگی for (به id تگ input) ارجاع داده شده است، انتخاب و محتوای آن قرمز می‌شود. **پشتیبانی در مرورگرها:**



نسخه CSS	Selector					
4	S1 /attribute/ S2	No	No	No	No	No

11-S1,S2

اگر بخواهیم قالب بندی را برای چند Selector به صورت یکجا انجام دهیم، این Selector ها را با کاما (,) از هم جدا می‌نماییم.

```
<style>
  div,.content,table.main {
    color: red;
  }
</style>
<div>Text 1</div>
<p>Text 2</p>
<table class="main" border="1">
  <tr>
    <td>Cell 1</td>
    <td>Cell 2</td>
    <td>Cell 3</td>
  </tr>
  <tr>
    <td>Cell 4</td>
    <td>Cell 5</td>
    <td>Cell 6</td>
  </tr>
</table>
<span class="content">Text 3</span>
<h1>Text 4</h1>
```

در مثال فوق Text 1 و Text 3 و همچنین محتوای تگ table به رنگ قرمز نمایش می‌یابند. پشتیبانی در مرورگرها:

	Selector	نسخه CSS					
1	S1,S2	Yes	Yes	Yes	Yes	Yes	Yes

توجه:

Selector ها می‌توانند به صورت ترکیبی نیز استفاده شوند. به مثال زیر توجه کنید:

```
<style>
  .content .tag {
    color: red;
  }

  h1#index {
    color: blue;
  }

  ul.list li.even {
    color: green;
  }
</style>
<h1 id="index">Index</h1>
<h1>Header 1</h1>
<div class="content">
  Lorem ipsum dolor sit amet, <span class="tag">consectetuer</span> adipiscing elit.
  Maecenas porttitor congue massa. Fusce posuere, magna sed pulvinar ultricies,
  purus lectus malesuada libero, <span class="tag">sit</span> amet commodo magna
  eros quis urna. Nunc viverra <span class="tag">imperdiet</span> enim. Fusce est.
</div>
<h1>Header 2</h1>
<div class="content">
```

```
<ul class="list">
  <li>Item 1</li>
  <li class="even">Item 2</li>
  <li>Item 3</li>
  <li class="even">Item 4</li>
  <li>Item 5</li>
  <li class="even">Item 6</li>
</ul>
</div>
```

در مثال فوق، تگ h1 که ویژگی id آن برابر index می‌باشد، با توجه به Selector ی که بصورت h1#index تعریف شده است، به رنگ آبی نمایش می‌یابد. تمامی تگ‌های span که عضو کلاس tag می‌باشند و در داخل تگ div ی قرار دارند که عضو کلاس content است، با توجه به Selector ی که بصورت .tag تعریف شده است، به رنگ قرمز نمایش می‌یابند. تمامی تگ‌های li که ویژگی class آنها برابر even می‌باشد، و فرزند تگ ul ی هستند که عضو کلاس list است، با توجه به Selector ی که بصورت ul.list li.even تعریف شده است، به رنگ سبز نمایش می‌یابند.

12- [attribute]

تگ‌هایی را انتخاب می‌نماید که دارای یک attribute یا ویژگی خاص باشند.

```
<style>
  [readonly] {
    background: green;
  }
</style>
<input type="text" value="Value 1" readonly="readonly"/>
<input type="text"/>
```

در مثال فوق، رنگ پس زمینه تگ input اول که دارای ویژگی readonly می‌باشد، به رنگ سبز نمایش می‌یابد. **پشتیبانی در مرورگرها:**






نسخه CSS	Selector					
2	[attribute]	4.0	2.0	7.0	9.6	3.1

13- [attribute=value]

تگ‌هایی را انتخاب می‌نماید که دارای یک attribute یا ویژگی خاص هستند و مقدار آن ویژگی دقیقاً برابر با value می‌باشد.

```
<style>
  [lang=fa] {
    direction: rtl;
  }
</style>
<div lang="fa">1 متن</div>
```

در مثال فوق، جهت نوشتاری محتوای تگ div که دارای ویژگی lang با مقدار fa می‌باشد، از راست به چپ می‌شود و در سمت راست صفحه نمایش می‌یابد. **پشتیبانی در مرورگرها:**

نسخه CSS	Selector					
2	[attribute=value]	4.0	2.0	7.0	9.6	3.1






[attribute=value i] 14-

تگ‌هایی را انتخاب می‌نماید که دارای یک attribute یا ویژگی خاص هستند و مقدار آن ویژگی دقیقاً برابر با value می‌باشد. همچنین value، نسبت به حروف کوچک و بزرگ حساس نمی‌باشد.

```
<style>
  [lang=fa] {
    direction: rtl;
  }
</style>
<div lang="FA">1 متن</div>
```

در مثال فوق، جهت نوشتاری محتوای تگ div که دارای ویژگی lang با مقدار FA می‌باشد، از راست به چپ می‌شود و در سمت راست صفحه نمایش می‌یابد. با اینکه در Selector مقدار fa با حروف کوچک ذکر شده است.

پشتیبانی در مرورگرها:

نسخه CSS	Selector					
4	[attribute=value i]	No	No	No	No	No






[attribute|=value] 15-

تگ‌هایی را انتخاب می‌نماید که دارای یک attribute یا ویژگی خاص هستند و مقدار آن ویژگی با یک value خاص آغاز می‌شود. کلمه آغازین مقدار حتماً باید با value برابر باشد یا با dash (-) از کلمه‌ی بعدی جدا شده باشد.

```
<style>
  [class|=info] {
    color: red;
  }
</style>
<div class="info">Text 1</div>
<div class="infobar">Text 2</div>
<div class="info bar">Text 3</div>
<div class="info-bar">Text 4</div>
<div class="btninfo">Text 5</div>
<div class="btn info">Text 6</div>
<div class="btn-info">Text 7</div>
<div class="toolinfoicon">Text 8</div>
```

```
<div class="tool info icon">Text 9</div>
<div class="tool-info-icon">Text 10</div>
```

در مثال فوق Text 1 و Text 4 به رنگ قرمز نمایش می‌یابند. پشتیبانی در مرورگرها:

نسخه CSS	Selector					
2	[attribute =value]	4.0	2.0	7.0	9.6	3.1






16- [attribute^=value]

تگ‌هایی را انتخاب می‌نماید که دارای یک attribute یا ویژگی خاص هستند و مقدار آن ویژگی با یک value خاص آغاز می‌شود.

```
<style>
  [class^=info] {
    color: red;
  }
</style>
<div class="info">Text 1</div>
<div class="infobar">Text 2</div>
<div class="info bar">Text 3</div>
<div class="info-bar">Text 4</div>
<div class="btninfo">Text 5</div>
<div class="btn info">Text 6</div>
<div class="btn-info">Text 7</div>
<div class="toolinfoicon">Text 8</div>
<div class="tool info icon">Text 9</div>
<div class="tool-info-icon">Text 10</div>
```

در مثال فوق Text 1 و Text 2 و Text 3 و Text 4 به رنگ قرمز نمایش می‌یابند.

پشتیبانی در مرورگرها:

نسخه CSS	Selector					
3	[attribute^=value]	4.0	3.5	7.0	9.6	3.2

17- [attribute~=value]

تگ‌هایی را انتخاب می‌نماید که دارای یک attribute یا ویژگی خاص هستند و مقدار آن ویژگی شامل یک value خاص می‌باشد که با Space یا فاصله‌ی خالی از سایر مقادیر جدا شده است.

```
<style>
```








```

    [class~=info] {
        color: red;
    }
</style>
<div class="info">Text 1</div>
<div class="infobar">Text 2</div>
<div class="info bar">Text 3</div>
<div class="info-bar">Text 4</div>
<div class="btninfo">Text 5</div>
<div class="btn info">Text 6</div>
<div class="btn-info">Text 7</div>
<div class="toolinfoicon">Text 8</div>
<div class="tool info icon">Text 9</div>
<div class="tool-info-icon">Text 10</div>

```

در مثال فوق Text 1 و Text 3 و Text 6 و Text 9 به رنگ قرمز نمایش می‌یابند. پشتیبانی در مرورگرها:

نسخه CSS	Selector					
2	[attribute~=value]	4.0	2.0	7.0	9.6	3.1

18- [attribute*=value]






تگ‌هایی را انتخاب می‌نماید که دارای یک attribute یا ویژگی خاص هستند و مقدار آن ویژگی شامل یک value خاص می‌باشد.

```

<style>
    [class*=info] {
        color: red;
    }
</style>
<div class="info">Text 1</div>
<div class="infobar">Text 2</div>
<div class="info bar">Text 3</div>
<div class="info-bar">Text 4</div>
<div class="btninfo">Text 5</div>
<div class="btn info">Text 6</div>
<div class="btn-info">Text 7</div>
<div class="toolinfoicon">Text 8</div>
<div class="tool info icon">Text 9</div>
<div class="tool-info-icon">Text 10</div>

```

در مثال فوق تمامی متون به رنگ قرمز نمایش می‌یابند. پشتیبانی در مرورگرها:






نسخه CSS	Selector					
3	[attribute*=value]	4.0	3.5	7.0	9.6	3.2

19- [attribute\$=value]

تگ‌هایی را انتخاب می‌نماید که دارای یک attribute یا ویژگی خاص هستند و مقدار آن ویژگی به یک value خاص ختم می‌شود.

```
<style>
  [class$=info] {
    color: red;
  }
</style>
<div class="info">Text 1</div>
<div class="infobar">Text 2</div>
<div class="info bar">Text 3</div>
<div class="info-bar">Text 4</div>
<div class="btninfo">Text 5</div>
<div class="btn info">Text 6</div>
<div class="btn-info">Text 7</div>
<div class="toolinfoicon">Text 8</div>
<div class="tool info icon">Text 9</div>
<div class="tool-info-icon">Text 10</div>
```

در مقال فوق Text 1 و Text 5 و Text 6 و Text 7 به رنگ قرمز نمایش می‌یابند. پشتیبانی در مرورگرها:

نسخه CSS	Selector					
3	[attribute\$=value]	4.0	3.5	7.0	9.6	3.2

Pseudo Class

به Selector هایی که با : آغاز می‌شوند Pseudo Class یا کلاس‌های کاذب گفته می‌شود.

link: 20-

تمامی تگ‌های a را انتخاب می‌کند که لینک می‌باشند یا به عبارتی دارای ویژگی href هستند.

```
<style>
:link {
    color: red;
}
</style>
<a href="page1.html">Link 1</a>
<a>Link 2</a>
<a href="page2.html">Link 3</a>
```

در مثال فوق Link 1 و Link 3 به رنگ قرمز نمایش می‌یابند. پشتیبانی در مرورگرها:

	Selector					
1	:link	4.0	2.0	7.0	9.6	3.1

visited: 21-

تمامی تگ‌های a را انتخاب می‌کند که قبلاً مشاهده یا بازدید شده اند.

```
<style>
:visited {
    color: green;
}
</style>
<a href="page1.html">Link 1</a>
<a>Link 2</a>
<a href="page2.html">Link 3</a>
```

در مثال فوق یکبار بر روی Link 1 و Link 3 کلیک نمایید تا صفحه‌ی مورد نظر باز شود. سپس مجدداً به همین صفحه بازگردید و مشاهده نمایید که Link 1 یا Link 3 یا هر دو به رنگ سبز نمایش می‌یابند. پشتیبانی در مرورگرها:





نسخه CSS	Selector					
1	:visited	4.0	2.0	7.0	9.6	3.1

focus: 22-

المتنی را انتخاب می‌کند که در حال حاضر فعال می‌باشند یا اصطلاحاً فوکوس (Focus) بر روی آن قرار دارد.

```
<style>
  :focus {
    background: yellow;
  }
</style>
<input type="text"/>
<input type="password"/>
<input type="text" />
```

در مثال فوق با فشردن Tab بر روی inputها حرکت کنید و مشاهده نمایید که رنگ پس زمینه‌ی آنها به رنگ زرد تغییر می‌کنند.
پشتیبانی در مرورگرها:

نسخه CSS	Selector					
2	:focus	4.0	2.0	8.0	9.6	3.1

hover: 23-

المتنی را انتخاب می‌کند که در حال حاضر ماوس (Mouse) بر روی آن قرار دارد.

```
<style>
  input:hover {
    background: yellow;
  }
</style>
<input type="text" />
<input type="password" />
<input type="text" />
```

در مثال فوق ماوس را بر روی المنتها قرار دهید و مشاهده نمایید که رنگ پس زمینه‌ی آنها به رنگ زرد تغییر می‌کنند.
پشتیبانی در مرورگرها:

نسخه CSS	Selector					
1	:hover	4.0	2.0	7.0	9.6	3.1

active: 24-

المتنی را انتخاب می‌کند که با ماوس بر روی آن کلیک شده باشد.

```
<style>
  button:active {
    background: yellow;
  }
</style>
```

```
</style>
<button>Button 1</button>
<button>Button 2</button>
```

در مثال فوق بر روی Button 1 یا Button 2 کلیک نمایید و دکمه‌ی ماوس را پایین نگه دارید و مشاهده نمایید تا زمانیکه کلید ماوس در حالت فشرده قرار دارد رنگ پس زمینه‌ی آنها به رنگ زرد نمایش می‌یابند. **پشتیبانی در مرورگرها:**

نسخه CSS	Selector					
1	:active	4.0	2.0	7.0	9.6	3.1

target: 25-

توسط تگ a و با استفاده از ویژگی name می‌توان بخشی از صفحه را نامگذاری نمود. سپس می‌توان توسط تگهای a به این نقطه از صفحه ارجاع داد. به این صورت که نام آن بخش از صفحه را با پیشوند # در ویژگی href تگ a ذکر نمود، تا لینک ما را به بخشی از یک صفحه منتقل کند. این Selector در زمان کلیک شدن بر روی تگ a که دارای href می‌باشد، آن تگ a که دارای ویژگی name می‌باشد را انتخاب می‌نماید.

```
<style>
:target {
    color: green;
}
</style>
<h2><a href="#part1">Link 1</a></h2>
<h2><a href="#part2">Link 2</a></h2>
<p>This is a paragraph</p>
<h1><a name="part1">Part 1</a></h1>
<p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit.
    Maecenas porttitor congue massa.
    Fusce posuere, magna sed pulvinar ultricies,
    purus lectus malesuada libero, sit amet commodo
    magna eros quis urna.
</p>
<h1><a name="part2">Part 2</a></h1>
<p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit.
    Maecenas porttitor congue massa.
    Fusce posuere, magna sed pulvinar ultricies,
    purus lectus malesuada libero, sit amet commodo
    magna eros quis urna.
</p>
```

در مثال فوق با کلیک بر روی Link 1 و Link 2 تگهای a مقصد با عنوان Part 1 و Part 2 به رنگ سبز نمایش می‌یابند. **پشتیبانی در مرورگرها:**

نسخه CSS	Selector					
3	:target	4.0	3.5	9.0	9.6	3.2

first-letter: 26-

اولین کاراکتر موجود در محتوای یک تگ را انتخاب می‌نماید.

```
<style>
  :first-letter {
    font-size: xx-large;
  }
</style>
<p>
  Lorem ipsum dolor sit amet, consectetur adipiscing elit.
  Maecenas porttitor congue massa.
  Fusce posuere, magna sed pulvinar ultricies,
  purus lectus malesuada libero, sit amet commodo
  magna eros quis urna.
</p>
```

در مثال فوق اولین کاراکتر تگ p یعنی حرف L به اندازه xx-large نمایش می‌یابد. پشتیبانی در مرورگرها:

نسخه CSS	Selector					
1	:first-letter	1.0	1.0	9.0	7.0	1.0

first-line: 27-

اولین سطر موجود در محتوای یک تگ را انتخاب می‌نماید.

```
<style>
  :first-line {
    color: red;
  }
</style>
<p>
  Lorem ipsum dolor sit amet, consectetur adipiscing elit.
  Maecenas porttitor congue massa. Fusce posuere,
  magna sed pulvinar ultricies,
  purus lectus malesuada libero, sit amet commodo magna eros
  quis urna. Nunc viverra imperdiet enim. Fusce est.
  Vivamus a tellus. Pellentesque habitant morbi tristique senectus
  et netus et malesuada fames ac turpis egestas. Proin pharetra
  nonummy pede. Mauris et orci. Aenean nec lorem.
</p>
```

در مثال فوق اولین سطر تگ p به رنگ قرمز نمایش می‌یابد. پشتیبانی در مرورگرها:






نسخه CSS	Selector					
1	:first-letter	1.0	1.0	9.0	7.0	1.0

28- empty:

تگ هایی را انتخاب می‌کند که هیچ محتوایی ندارند و خالی می‌باشند.

```
<style>
  :empty {
    background: gray;
  }
</style>
<table border="1" cellpadding="10" cellspacing="10">
  <tr>
    <td>A</td>
    <td>B</td>
    <td></td>
  </tr>
  <tr>
    <td>C</td>
    <td></td>
    <td>D</td>
  </tr>
  <tr>
    <td></td>
    <td>E</td>
    <td>F</td>
  </tr>
</table>
```

در مثال فوق رنگ پس زمینه سلول‌های خالی به رنگ خاکستری تیره نمایش می‌یابد. پشتیبانی در مرورگرها:



نسخه CSS	Selector					
3	:target	4.0	3.5	9.0	9.6	3.2

29- dir(direction):

تگ هایی را انتخاب می‌نماید که دارای ویژگی dir با یک مقدار خاص می‌باشند.

```
<style>
  :dir rtl {
    color: red;
  }
</style>
<div dir="rtl">1 متن</div>
<div>2 متن</div>
```

در مثال فوق "متن 1" به رنگ قرمز نمایش می‌یابد. پشتیبانی در مرورگرها:






نسخه CSS	Selector					
4	:dir(direction)	No	No	No	No	No


30- (...lang(language1, language2: 30-

تگ هایی را انتخاب می کند که دارای ویژگی lang با یک مقدار خاص می باشند. می توان 1 یا چند زبان را در این Selector مشخص نمود.

```
<style>
:lang(en) {
    color: red;
}
</style>
<div lang="en">Text 1</div>
<div>Text 2</div>
<div lang="en">Text 3</div>
```

در مثال فوق Text 1 و Text 3 به رنگ قرمز نمایش می یابند. پشتیبانی در مرورگرها:

نسخه CSS	Selector					
2	:lang(language 1)	4.0	2.0	8.0	9.6	3.1






نسخه CSS	Selector					
4	:lang(language 1, language2,...)	No	No	No	No	No

disabled: 31-

تگ‌هایی را انتخاب می‌نماید که دارای ویژگی disabled می‌باشند.

```
<style>
  :disabled {
    background: yellow;
  }
</style>
<input disabled="disabled"/>
<input />
<input disabled="disabled"/>
<input />
```

در مثال فوق input اول و سوم به رنگ زرد نمایش می‌یابند. پشتیبانی در مرورگرها:

نسخه CSS	Selector					
3	:disabled	4.0	3.5	9.0	9.6	3.2






enabled: 32-

تگ‌هایی را انتخاب می‌نماید که فاقد ویژگی disabled می‌باشند. در حال حاضر در اکثر مرورگرها فقط برای تگ input و textarea عمل می‌کند.

```
<style>
  :enabled {
    background: yellow;
  }
</style>
<input disabled="disabled" />
<input />
<input disabled="disabled" />
<input />
```

در مثال فوق input دوم و چهارم به رنگ زرد نمایش می‌یابند.

پشتیبانی در مرورگرها:

نسخه CSS	Selector					
3	:enabled	4.0	3.5	9.0	9.6	3.2



33- read-only:

تگ‌هایی را انتخاب می‌نماید که دارای ویژگی readonly می‌باشند. در برخی مرورگرها ممکن است تگ body و سایر تگ‌هایی را که محتوای آنها غیر قابل ویرایش می‌باشد نیز انتخاب نماید.

```
<style>
  :read-only {
    background: yellow;
  }
</style>
<input readonly="readonly" />
<input />
<input readonly="readonly" />
<input />
```

در مثال فوق input اول و سوم به رنگ زرد نمایش می‌یابند. در برخی مرورگرها نیز ممکن است پس زمینه‌ی صفحه نیز به رنگ زرد نمایش یابد.

پشتیبانی در مرورگرها:

نسخه CSS	Selector					
3	:read-only	Yes	-moz-	No	Yes	Yes






34- read-write:

تگ‌هایی را انتخاب می‌نماید که فاقد ویژگی readonly می‌باشند و یا محتوای آنها قابل ویرایش است.

```
<style>
  :read-write {
    background: yellow;
  }
</style>
<input readonly="readonly" />
<input />
<input readonly="readonly" />
<input />
```

در مثال فوق input دوم و چهارم به رنگ زرد نمایش می‌یابند.

پشتیبانی در مرورگرها:

نسخه CSS	Selector					
3	:read-write	Yes	-moz-	No	Yes	Yes

35- valid:

تگ‌های input ایی را انتخاب می‌نماید که مقدار ورودی آنها با توجه به ویژگی‌های type و pattern معتبر باشند. مثال مربوط به این

بخش در Selector بعدی، یعنی **invalid:** ذکر شده است. پشتیبانی در مرورگرها:

نسخه CSS	Selector					
3	:valid	10.0	4.0	10.0	10.0	5.0

invalid: 36-

تگ های input ی را انتخاب می نماید که مقدار ورودی آنها با توجه به ویژگی های type و pattern نامعتبر باشند.

```
<style>
  :valid {
    border: 3px solid green;
  }
  :invalid {
    border: 3px solid red;
  }
</style>
<input type="number" />
<input type="text" pattern="^[a-z]{2,10}$"/>
<input type="email" />
```

در مثال فوق، در ابتدا، کادر دور تگ های input با توجه به Selector ی با عنوان **valid:**، به رنگ سبز نمایش می یابند. با ورود اطلاعات، در صورتی که مقدار ورودی نامعتبر باشد، کادر به رنگ قرمز تغییر می یابد. به عنوان مثال با ورود حرف در input اول، با ورود عدد در input دوم و با ورود یک متن تا قبل از ورود @ و دامین، کادر به رنگ قرمز نمایش می یابد. **پشتیبانی در مرورگرها:**



نسخه CSS	Selector					
3	:invalid	10.0	4.0	10.0	10.0	5.0

required: 37-

تگ هایی را انتخاب می نماید که دارای ویژگی **required** می باشند.

```
<style>
  :required {
    background: yellow;
  }
</style>
<input required="required" />
<input />
<input required="required" />
<input />
```

در مثال فوق input اول و سوم به رنگ زرد نمایش می یابند.
پشتیبانی در مرورگرها:

نسخه CSS	Selector					
3	:required	10.0	4.0	10.0	10.0	5.0





optional: 38-

تگ‌هایی را انتخاب می‌نماید که فاقد ویژگی required می‌باشند.

```
<style>
:optional {
  background: yellow;
}
</style>
<input required="required" />
<input />
<input required="required" />
<input />
```


در مثال فوق input دوم و چهارم به رنگ زرد نمایش می‌یابند.

پشتیبانی در مرورگرها:

نسخه CSS	Selector					
3	:optional	10.0	4.0	10.0	10.0	5.0

in-range: 39-

برای تگ‌های input ی استفاده می‌شود که ویژگی min و max دارند. تگ‌هایی را انتخاب می‌نماید که مقدار ورودی آنها در بازه‌ی min و max باشند. مثال مربوط به این بخش در Selector بعدی، یعنی out-of-range: ذکر شده است. پشتیبانی در مرورگرها:

نسخه CSS	Selector					
3	:in-range	10.0	28.0	No	11.0	5.2

out-of-range: 40-

برای تگ‌های input ی استفاده می‌شود که ویژگی min و max دارند. تگ‌هایی را انتخاب می‌نماید که مقدار ورودی آنها خارج از بازه‌ی min و max باشد.

```
<style>
  :in-range {
    border: 3px solid green;
  }
  :out-of-range {
    border: 3px solid red;
  }
</style>
<input type="number" min="10" max="50" />
```

در مثال فوق با ورود عددی خارج از محدوده‌ی 10 تا 50، کادر دور تگ به رنگ قرمز نمایش می‌یابد، در غیر اینصورت به رنگ سبز نمایش می‌یابد. پشتیبانی در مرورگرها:






نسخه CSS	Selector					
3	:out-of-range	10.0	28.0	No	11.0	5.2

checked: 41-

برای تگ‌های radio و checkbox استفاده می‌شود و آنهایی را انتخاب می‌کند که گزینه‌ی آن‌ها انتخاب شده است یا شامل ویژگی checked می‌باشند.

```
<style>
    :checked {
        width: 50px;
        height: 50px;
    }
</style>
<input type="checkbox" checked="checked"/>
<input type="checkbox"/>
<input type="radio" name="test"/>
<input type="radio" name="test" checked="checked"/>
<input type="radio" name="test"/>
```

در مثال فوق تمامی checkbox هایی که گزینه‌ی آن‌ها انتخاب شده‌اند با ابعاد 50 در 50 نمایش می‌یابند. پشتیبانی در مرورگرها:

نسخه CSS	Selector					
3	:checked	4.0	3.5	9.0	9.6	3.2

indeterminate: 42-

در HTML نمی‌توان وضعیت indeterminate را برای checkbox ایجاد نمود. اما با جاوا اسکریپت می‌توان این وضعیت را ایجاد کرد. هر checkbox می‌تواند دارای سه وضعیت انتخاب شده، انتخاب نشده و نامعلوم (indeterminate) باشد. این Selector تگ‌های checkbox ی را انتخاب می‌نماید که وضعیت نامعلوم دارند.

```
<style>
    :indeterminate {
        width: 50px;
        height: 50px;
    }
</style>
<input type="checkbox" id="chk1"/>
<input type="checkbox" id="chk2" checked="checked"/>
<input type="checkbox" id="chk3"/>
<script>
    document.getElementById("chk1").indeterminate=true;
</script>
```

در مثال فوق chk1 به صورت indeterminate انتخاب شده است که با ابعاد 50 در 50 نمایش می‌یابد. پشتیبانی در مرورگرها:

نسخه CSS	Selector					
4	:indeterminate	No	No	No	No	No

default: 43-

برای تگ‌های radio و checkbox، آنهایی را انتخاب می‌کند که در زمان بارگزاری فرم و بصورت پیش فرض انتخاب شده‌اند. برای دکمه‌ی submit ایی نیز استفاده می‌شود که در حال حاضر در فرم فعال می‌باشد. پشتیبانی از این Selector در مرورگرها متفاوت می‌باشد؛ بنابراین در چند مرورگر کارآیی آن را بررسی کنید.

```
<style>
  :default {
    width: 50px;
    height: 50px;
    background: lime;
  }
</style>
<form>
  <input type="checkbox" checked="checked" />
  <input type="checkbox" />
  <input type="radio" name="radio1" />
  <input type="radio" name="radio1" checked="checked" />
  <input type="radio" name="radio1" />
  <input type="submit" value="Default" />
  <input type="submit" value="Submit 2" />
</form>
```

در مثال فوق checkbox و radio هایی که در لود اولیه فرم دارای ویژگی checked می‌باشند به همراه دکمه‌ی submit با عنوان Default با ابعاد 50 در 50 و رنگ پس زمینه‌ی لیمویی نمایش می‌یابند. **پشتیبانی در مرورگرها:**






نسخه CSS	Selector					
4	:default	No	No	No	No	No

root: 44-

المنت ریشه‌ی یک صفحه را انتخاب می‌نماید. معمولاً تگ html به عنوان المنت ریشه انتخاب می‌شود.

```
<style>
  :root {
    background: khaki;
  }
</style>
```

در مثال فوق رنگ پس زمینه به رنگ خاکی نمایش می‌یابد. **پشتیبانی در مرورگرها:**






نسخه CSS	Selector					
3	:root	4.0	3.5	9.0	9.6	3.2

after: 45-

جهت درج محتوا بعد از تگ استفاده می‌شود.

```
<style>
  form:after {
    content: "[ * : Required]";
    color: red;
  }
  input+span:after {
    content: ' * ';
    color: red;
  }
</style>
<form>
  <div><input type="text" /><span></span></div>
  <div><input type="password" /><span></span></div>
  <div><input type="email" /><span></span></div>
</form>
```

در مثال فوق بعد از تمامی span‌های قرار گرفته در کنار تگ‌های input، یک کاراکتر * را به رنگ قرمز نمایش می‌دهد. بعد از فرم نیز رشته‌ی [* : Required] را به رنگ قرمز نمایش می‌دهد. **پشتیبانی در مرورگرها:**

نسخه CSS	Selector					
2	:after	4.0	3.5	9.0	7.0	3.1






before: 46-

جهت درج محتوا قبل از تگ استفاده می‌شود.

```
<style>
  form:before {
    content: "[ * : Required]";
    color: red;
  }
  input + span:before {
    content: ' * ';
    color: red;
  }
</style>
<form>
  <div><input type="text" /><span></span></div>
  <div><input type="password" /><span></span></div>
  <div><input type="email" /><span></span></div>
</form>
```


در مثال فوق قبل از تمامی span های قرار گرفته در کنار تگ های input، یک کاراکتر * را به رنگ قرمز نمایش می دهد. قبل از فرم نیز رشته ی [* : Required] را به رنگ قرمز نمایش می دهد.

پشتیبانی در مرورگرها:

نسخه CSS	Selector					
2	:before	4.0	3.5	9.0	7.0	3.1

47- selection::

محتوایی را انتخاب می کند که در حال حاضر در صفحه انتخاب یا Select شده است.

```
<style>
  ::selection {
    background: navy;
    color: orange;
  }
</style>
<p>
  Lorem ipsum dolor sit amet, consectetur adipiscing elit.
  Maecenas porttitor congue massa. Fusce posuere,
  magna sed pulvinar ultricies, purus lectus malesuada libero,
  sit amet commodo magna eros quis urna. Nunc viverra imperdiet enim. Fusce est.
</p>
```

در مثال فوق با انتخاب متن توسط ماوس، رنگ زمینه ی متن انتخاب شده به صورت سرمه ای و رنگ متن آن نارنجی نمایش می یابد. **پشتیبانی در مرورگرها:**

نسخه CSS	Selector					
در CSS 3 معرفی و سپس حذف شد ولی امکان بازگشت مجدد وجود دارد	::selection	4.0	2.0 -moz-	9.0	9.6	3.1






48- (not(S1,S2:

تگ هایی را انتخاب می کند که شامل Selector های S1 یا S2 یا ... یا Sn نباشند.

```
<style>
  :not([readonly]) {
    background: yellow;
  }
</style>
```

```
<input type="text"/>
<input type="text" readonly="readonly"/>
<input type="text"/>
<input type="text" readonly="readonly" />
```

در مثال فوق پس زمینه‌ی تگ‌های input که ویژگی readonly ندارند به رنگ زرد نمایش می‌یابند. پشتیبانی در مرورگرها:

نسخه CSS	Selector					
3	:not(S1,S2)	4.0	3.5	9.0	9.6	3.2

(matches(S1,S2: 49-

تگ‌هایی را انتخاب می‌کند که شامل Selectorهای S1 و/یا S2 ... و/یا Sn باشند.

```
<style>
  :matches([readonly]) {
    background: yellow;
  }
</style>
<input type="text" />
<input type="text" readonly="readonly" />
<input type="text" />
<input type="text" readonly="readonly" />
```

در مثال فوق پس زمینه‌ی تگ‌های input که ویژگی readonly دارند به رنگ زرد نمایش می‌یابند.

پشتیبانی در مرورگرها:

نسخه CSS	Selector					
4	:matches(S1,S2)	No	No	No	No	No

(has(S1,S2: 50-

تگ‌هایی را انتخاب می‌کند که در ارتباط خاصی با Selectorهای S1 و/یا S2 ... و/یا Sn می‌باشند.

```
<style>
  :has(>span) {
    color: red;
  }
  :has(+div) {
    color: blue;
  }
</style>
<div>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit.
    Maecenas <span>porttitor</span> congue massa. Fusce posuere,
```


```

    magna sed pulvinar ultricies, purus lectus malesuada libero,
    sit amet commodo magna eros quis urna. Nunc viverra imperdiet enim. Fusce est.
</p>
<h1>Header 1</h1>
<div>

    Lorem ipsum dolor sit amet, consectetur adipiscing elit.
    Maecenas porttitor congue massa. Fusce posuere,
    magna sed pulvinar ultricies, purus lectus malesuada libero,
    sit amet commodo magna eros quis urna. Nunc viverra imperdiet enim. Fusce est.
</div>

```

در مثال فوق متن موجود در تگ p که دارای فرزند مستقیم span می‌باشد، با توجه به Selector ی که بصورت (has(>span: تعریف شده است، به رنگ قرمز نمایش می‌یابد. همچنین Header 1 که دارای تگ div هم تراز و دقیقاً بعد از h1 است، با توجه به Selector ی که بصورت (has(+div: تعریف شده است، به رنگ آبی نمایش می‌یابد. **پشتیبانی در مرورگرها:**

نسخه CSS	Selector					
4	:has(S1,S2)	No	No	No	No	No

first-child: 51-

تگی را انتخاب می‌کند که اولین فرزند والد خود باشد.

```
<style>
  div.container :first-child {
    color: red;
  }
</style>
<div class="container">
  <h1>Text 1</h1>
  <span>Text 2</span>
  <p>Text 3</p>
  <div>Text 4</div>
  <div>Text 5</div>
  <div>
    <h1>Text 6</h1>
    <span>Text 7</span>
    <p>Text 8</p>
    <p>
      <span>Text 9</span>
    </p>
    <div>Text 10</div>
    <p>Text 11</p>
  </div>
  <h1>Text 12</h1>
  <div>Text 13</div>
  <span>Text 14</span>
  <div>Text 15</div>
</div>
```

در مثال فوق Text 1، Text 6 و Text 9 به رنگ قرمز نمایش می‌یابند.

پشتیبانی در مرورگرها:

نسخه CSS	Selector					
2	:first-child	4.0	3.0	7.0	9.6	3.1

last-child: 52-

تگی را انتخاب می‌کند که آخرین فرزند والد خود باشد.






```
<style>
  div.container :last-child {
    color: red;
  }
</style>
<div class="container">
  <h1>Text 1</h1>
  <span>Text 2</span>
  <p>Text 3</p>
  <div>Text 4</div>
  <div>Text 5</div>
  <div>
    <h1>Text 6</h1>
    <span>Text 7</span>
  </div>
```

```

    <p>Text 8</p>
    <p>
      <span>Text 9</span>
    </p>
    <div>Text 10</div>
    <p>Text 11</p>
  </div>
  <h1>Text 12</h1>
  <div>Text 13</div>
  <span>Text 14</span>
  <div>Text 15</div>
</div>

```

در مثال فوق Text 11، Text 15 و Text 9 به رنگ قرمز نمایش می‌یابند. پشتیبانی در مرورگرها:

نسخه CSS	Selector					
3	:last-child	4.0	3.5	9.0	9.6	3.2

53- only-child:

تگی را انتخاب می‌کند که تنها فرزند والد خود باشد.






```

<style>
  div.container :only-child {
    color: red;
  }
</style>
<div class="container">
  <h1>Text 1</h1>
  <span>Text 2</span>
  <p>Text 3</p>
  <div>Text 4</div>
  <div>Text 5</div>
  <div>
    <h1>Text 6</h1>
    <span>Text 7</span>
    <p>Text 8</p>
    <p>
      <span>Text 9</span>
    </p>
    <div>Text 10</div>
    <p>Text 11</p>
  </div>
  <h1>Text 12</h1>
  <div>Text 13</div>
  <span>Text 14</span>
  <div>Text 15</div>
</div>

```

در مثال فوق Text 9 به رنگ قرمز نمایش می‌یابد.

پشتیبانی در مرورگرها:






نسخه CSS	Selector					
3	:only-child	4.0	3.5	9.0	9.6	3.2

(nth-child(n: 54-

تگی را انتخاب می‌کند که n امین فرزند والد خود باشد. به جای n می‌توان از مقادیر odd (فرزندان فرد)، even (فرزندان زوج) و an+b استفاده نمود.

```
<style>
  div.container :nth-child(2) {
    color: red;
  }
</style>
<div class="container">
  <h1>Text 1</h1>
  <span>Text 2</span>
  <p>Text 3</p>
  <div>Text 4</div>
  <div>Text 5</div>
  <div>
    <h1>Text 6</h1>
    <span>Text 7</span>
    <p>Text 8</p>
    <p>
      <span>Text 9</span>
    </p>
    <div>Text 10</div>
    <p>Text 11</p>
  </div>
  <h1>Text 12</h1>
  <div>Text 13</div>
  <span>Text 14</span>
  <div>Text 15</div>
</div>
```

در مثال فوق Text 2 و Text 7 به رنگ قرمز نمایش می‌یابند. پشتیبانی در مرورگرها:

نسخه CSS	Selector					
3	:nth-child(n)	4.0	3.5	9.0	9.6	3.2

(nth-last-child(n: 55-

تگی را انتخاب می‌کند که n امین فرزند والد خود از آخر باشد. به جای n می‌توان از مقادیر odd (فرزندان فرد)، even (فرزندان زوج) و an+b استفاده نمود.






```
<style>
  div.container :nth-last-child(2) {
    color: red;
  }
</style>
```

```

</style>
<div class="container">
  <h1>Text 1</h1>
  <span>Text 2</span>
  <p>Text 3</p>
  <div>Text 4</div>
  <div>Text 5</div>
  <div>
    <h1>Text 6</h1>
    <span>Text 7</span>
    <p>Text 8</p>
    <p>
      <span>Text 9</span>
    </p>
    <div>Text 10</div>
    <p>Text 11</p>
  </div>
  <h1>Text 12</h1>
  <div>Text 13</div>
  <span>Text 14</span>
  <div>Text 15</div>
</div>

```

در مثال فوق Text 14 و Text 10 به رنگ قرمز نمایش می‌یابند.
پشتیبانی در مرورگرها:

نسخه CSS	Selector					
3	:nth-last-child(n)	4.0	3.5	9.0	9.6	3.2

56- first-of-type:






تگی را انتخاب می‌کند که اولین تگ در بین هم نوعان خودش و در یک والد باشد.

```

<style>
  div.container :first-of-type {
    color: red;
  }
</style>
<div class="container">
  <h1>Text 1</h1>
  <span>Text 2</span>
  <p>Text 3</p>
  <div>Text 4</div>
  <div>Text 5</div>
  <div>
    <h1>Text 6</h1>
    <span>Text 7</span>
    <p>Text 8</p>
    <p>
      <span>Text 9</span>
    </p>
    <div>Text 10</div>
    <p>Text 11</p>
  </div>
  <h1>Text 12</h1>
  <div>Text 13</div>
  <span>Text 14</span>
  <div>Text 15</div>
</div>

```

در مثال فوق Text 1, Text 2, Text 3, Text 4, Text 6, Text 7, Text 8, Text 9 و Text 10 به رنگ قرمز نمایش می‌یابند.
پشتیبانی در مرورگرها:






نسخه CSS	Selector					
3	:first-of-type	4.0	3.5	9.0	9.6	3.2

last-of-type: 57-

تگی را انتخاب می‌کند که آخرین تگ در بین هم نوعان خودش و در یک والد باشد.

```
<style>
  div.container :last-of-type {
    color: red;
  }
</style>
<div class="container">
  <h1>Text 1</h1>
  <span>Text 2</span>
  <p>Text 3</p>
  <div>Text 4</div>
  <div>Text 5</div>
  <div>
    <h1>Text 6</h1>
    <span>Text 7</span>
    <p>Text 8</p>
    <p>
      <span>Text 9</span>
    </p>
    <div>Text 10</div>
    <p>Text 11</p>
  </div>
  <h1>Text 12</h1>
  <div>Text 13</div>
  <span>Text 14</span>
  <div>Text 15</div>
</div>
```

در مثال فوق Text 3 و Text 15, Text 14, Text 12, Text 11, Text 10, Text 9, Text 7, Text 6 به رنگ قرمز نمایش می‌یابند.
پشتیبانی در مرورگرها:

نسخه CSS	Selector					
3	:last-of-type	4.0	3.5	9.0	9.6	3.2

only-of-type: 58-

تگی را انتخاب می‌کند که تنها تگ در بین هم نوعان خودش و در یک والد باشد.

```
<style>
  div.container :only-of-type {
```








```

        color: red;
    }
</style>
<div class="container">
    <h1>Text 1</h1>
    <span>Text 2</span>
    <p>Text 3</p>
    <div>Text 4</div>
    <div>Text 5</div>
    <div>
        <h1>Text 6</h1>
        <span>Text 7</span>
        <p>Text 8</p>
        <p>
            <span>Text 9</span>
        </p>
        <div>Text 10</div>
        <p>Text 11</p>
    </div>
    <h1>Text 12</h1>
    <div>Text 13</div>
    <span>Text 14</span>
    <div>Text 15</div>
</div>

```

در مثال فوق Text 9، Text 7، Text 6، Text 3 و Text 10 به رنگ قرمز نمایش می‌یابند. پشتیبانی در مرورگرها:

نسخه CSS	Selector					
3	:only-of-type	4.0	3.5	9.0	9.6	3.2

(nth-of-type(n: 59-






تگی را انتخاب می‌کند که n امین تگ در بین هم نوعان خودش و در یک والد باشد. به جای n می‌توان از مقادیر odd (فرزندان فرد)، even (فرزندان زوج) و $an+b$ استفاده نمود.

```

<style>
    div.container :nth-of-type(2) {
        color: red;
    }
</style>
<div class="container">
    <h1>Text 1</h1>
    <span>Text 2</span>
    <p>Text 3</p>
    <div>Text 4</div>
    <div>Text 5</div>
    <div>
        <h1>Text 6</h1>
        <span>Text 7</span>
        <p>Text 8</p>
        <p>
            <span>Text 9</span>
        </p>
        <div>Text 10</div>
        <p>Text 11</p>
    </div>
    <h1>Text 12</h1>
    <div>Text 13</div>
    <span>Text 14</span>
    <div>Text 15</div>
</div>

```

در مثال فوق Text 12، Text 9، Text 5، و Text 14 به رنگ قرمز نمایش می‌یابند. پشتیبانی در مرورگرها:






نسخه CSS	Selector					
3	:nth-of-type(n)	4.0	3.5	9.0	9.6	3.2

(nth-last-of-type(n 60-

تگی را انتخاب می‌کند که nامین تگ از آخر در بین هم نوعان خودش و در یک والد باشد. به جای n می‌توان از مقادیر odd (فرزندان فرد)، even (فرزندان زوج) و $an+b$ استفاده نمود.

```
<style>
  div.container :nth-last-of-type(2) {
    color: red;
  }
</style>
<div class="container">
  <h1>Text 1</h1>
  <span>Text 2</span>
  <p>Text 3</p>
  <div>Text 4</div>
  <div>Text 5</div>
  <div>
    <h1>Text 6</h1>
    <span>Text 7</span>
    <p>Text 8</p>
    <p>
      <span>Text 9</span>
    </p>
    <div>Text 10</div>
    <p>Text 11</p>
  </div>
  <h1>Text 12</h1>
  <div>Text 13</div>
  <span>Text 14</span>
  <div>Text 15</div>
</div>
```

در مثال فوق Text 9، Text 2، Text 1، و Text 13 به رنگ قرمز نمایش می‌یابند. پشتیبانی در مرورگرها:

نسخه CSS	Selector					
3	:nth-last-of-type(n)	4.0	3.5	9.0	9.6	3.2

یکی از مشکلات سینتکس Razor سمت سرور، این است که در فایل‌های JavaScript و CSS سمت کاربر نمی‌توانیم از آن استفاده کنیم. به عنوان مثال فرض کنید در یک فایل JavaScript نیاز به مشخص سازی آدرس یک اکشن متد دارید؛ مثلاً انجام یک عملیات ای‌جکسی. در این حالت به عنوان یک Best Practice بهتر است از [Url.Action](#) استفاده کنید. اما همانطور که عنوان شد، این امکان یعنی استفاده از سینتکس Razor در فایل‌های JS و CSS مهیا نیست.

ساده‌ترین راه‌حل، تولید ویوهای سمت سرور JavaScript ایی است. برای اینکار تنها کاری که باید انجام دهیم، تغییر مقدار Content-Type صفحه به مقدار موردنظر می‌باشد؛ مثلاً text/javascript برای فایل‌های JS و text/css برای فایل‌های CSS. به عنوان مثال برای فایل‌های CSS به این صورت عمل خواهیم کرد:

```
public ActionResult Style()
{
    Response.ContentType = "text/css";
    var model = new Style
    {
        Color = "red",
        Background = "blue"
    };
    return View(model);
}
```

برای ویوی آن نیز خواهیم داشت:

```
@model ExternalJavaScript.Models.Style
@{
    Layout = null;
}
body {
    color : @Model.Color;
    background-color : @Model.Background;
}
```

در نهایت ویوی فوق را به عنوان فایل CSS در فایل Layout استفاده خواهیم کرد:

```
<link rel="stylesheet" href="@Url.Action("Style","Home")" />
```

برای حالت فوق می‌توانیم یک اکشن فیلتر به صورت زیر تهیه کنیم:

```
public class ContentType : ActionFilterAttribute
{
    private string _contentType;
    public ContentType(string ct)
    {
        this._contentType = ct;
    }

    public override void OnActionExecuted(ActionExecutedContext context) { /* nada */ }
    public override void OnActionExecuting(ActionExecutingContext context)
    {
        context.HttpContext.Response.ContentType = this._contentType;
    }
}
```

و برای استفاده از آن خواهیم داشت:

```
[ContentType("text/css")]
public ActionResult Style()
{
    var model = new Style
    {
        Color = "red",
        Background = "blue"
    };
    return View(model);
}
```

برای فایل‌های JS نیز می‌توانیم از یک View به عنوان محل قرارگیری کدهای جاوا اسکریپت استفاده کنیم:

```
public class JavaScriptSettingsController : Controller
{
    public ActionResult Index()
    {
        return PartialView();
    }
}
```

در این حالت در داخل فایل Index.cshtml کدهای جاوا اسکریپت را همراه با سینتکس Razor می‌توانیم بنویسیم:

```
$(function(){
    $.post('@Url.Action("GetData", "Home")', function (data) {
        $('#notificationList').html(data);
        if ($(data).filter("li").length != 0) {
            $('#notificationCounter').html($(data).filter("li").length);
        }
    });
});
```

سپس در داخل فایل _Layout.cshtml می‌توانیم به ویوی فوق ارجاعی داشته باشیم:

```
<script src="/JavaScriptSettings"></script>
```

این روش به خوبی برای ویوهای JS و CSS کار خواهد کرد؛ اما از آنجائیکه ویوی ما توسط ویژوال استودیو به عنوان یک فایل JS یا CSS معتبر شناخته نمی‌شود، IntelliSense برای آن مهیا نیست. برای فعال سازی IntelliSense و همچنین معتبر شناخته شدن ویوی فوق، بهترین راه حل قرار دادن کدهای JS درون بلاک script است (برای فایل‌های CSS نیز همینطور):

```
<script>
$(function () {
    $.post('@Url.Action("Index", "Home")', function (data) {
        $('#notificationList').html(data);
        if ($(data).filter("li").length != 0) {
            $('#notificationCounter').html($(data).filter("li").length);
        }
    });
});
</script>
```

اما با اجرای برنامه، در کنسول مرورگر بلافاصله خطای Uncaught SyntaxError: Unexpected token > را دریافت خواهید کرد. در این حالت به روشی نیاز داریم که در زمان اجرا بلاک script را حذف نمائید. بنابراین از یک اکشن فیلتر سفارشی برای اینکار استفاده خواهیم کرد. کار این اکشن فیلتر، تغییر مقدار Content-Type و همچنین حذف بلاک مورد نظر می‌باشد:

```
public class ExternalFileAttribute : ActionFilterAttribute
{
    private readonly string _contentType;
    private readonly string _tag;
    public ExternalFileAttribute(string ct, string tag)
    {
        this._contentType = ct;
        _tag = tag;
    }
}
```

```

    }

    public override void OnResultExecuted(ResultExecutedContext filterContext)
    {
        var response = filterContext.HttpContext.Response;
        response.Filter = new StripEnclosingTagsFilter(response.Filter, _tag);
        response.ContentType = _contentType;
    }

    private class StripEnclosingTagsFilter : MemoryStream
    {
        private static Regex _leadingOpeningScriptTag;
        private static Regex _trailingClosingScriptTag;

        //private static string Tag;

        private readonly StringBuilder _output;
        private readonly Stream _responseStream;

        /*static StripEnclosingTagsFilter()
        {
            LeadingOpeningScriptTag = new Regex(string.Format(@"^<\s*<{0}>[^\>]*>", Tag),
            RegexOptions.Compiled);
            TrailingClosingScriptTag = new Regex(string.Format(@"</{0}>\s*$", Tag),
            RegexOptions.Compiled);
        }*/

        public StripEnclosingTagsFilter(Stream responseStream, string tag)
        {
            _leadingOpeningScriptTag = new Regex(string.Format(@"^<\s*<{0}>[^\>]*>", tag),
            RegexOptions.Compiled);
            _trailingClosingScriptTag = new Regex(string.Format(@"</{0}>\s*$", tag),
            RegexOptions.Compiled);

            _responseStream = responseStream;
            _output = new StringBuilder();
        }

        public override void Write(byte[] buffer, int offset, int count)
        {
            string response = GetStringResponse(buffer, offset, count);
            _output.Append(response);
        }

        public override void Flush()
        {
            string response = _output.ToString();

            if (_leadingOpeningScriptTag.IsMatch(response) &&
            _trailingClosingScriptTag.IsMatch(response))
            {
                response = _leadingOpeningScriptTag.Replace(response, string.Empty);
                response = _trailingClosingScriptTag.Replace(response, string.Empty);
            }

            WriteStringResponse(response);
            _output.Clear();
        }

        private static string GetStringResponse(byte[] buffer, int offset, int count)
        {
            byte[] responseData = new byte[count];
            Buffer.BlockCopy(buffer, offset, responseData, 0, count);

            return Encoding.Default.GetString(responseData);
        }

        private void WriteStringResponse(string response)
        {
            byte[] outdata = Encoding.Default.GetBytes(response);
            _responseStream.Write(outdata, 0, outdata.GetLength(0));
        }
    }
}

```

در نهایت می‌توانیم اکشن‌متد موردنظرمان را با فیلتر سفارشی مزین کنیم:

```
[ExternalFile("text/javascript", "script")]
```

```
public ActionResult Index()
{
    return PartialView();
}
```

برای تولید ویوهای CSS نیز کافی است مقادیر فیلتر را تغییر دهیم:

```
[ExternalFile("text/css", "style")]
public ActionResult Style()
{
    var model = new Style
    {
        Color = "red",
        Background = "blue"
    };
    return View(model);
}
```

نظرات خوانندگان

نویسنده: حسین شجاعی
تاریخ: ۲۳:۴۱۳۹۴/۰۲/۳۰

سپاس از مطلبتون.
فقط یک سوال : این [پاسخ](#) صحیحه؟
موفق باشید

نویسنده: سیروان عقیفی
تاریخ: ۰:۴۸۱۳۹۴/۰۲/۳۱

سوال مربوط به استفاده از razor درون ویو (در واقع درون تگ اسکریپت) است، مطلب جاری درباره استفاده از razor درون یک فایل جاوا اسکریپت و یا CSS اکسترنال می‌باشد.

نویسنده: مهدی سعیدی فر
تاریخ: ۱۹:۲۸۱۳۹۴/۰۲/۳۱

برای پیاده سازی minification و bundling روشی هست؟

نویسنده: سیروان عقیفی
تاریخ: ۱۳:۲۲۱۳۹۴/۰۳/۰۱

برای bundle کردن من روشی به ذهنم نمی‌رسه ولی برای minification می‌تونید از روش [حذف فضاهای خالی در خروجی صفحات ASP.NET MVC](#) استفاده کنید.

نویسنده: وحید نصیری
تاریخ: ۲۲:۵۳۱۳۹۴/۰۳/۰۲

یک روش دیگر هم استفاده از ویژگی‌های data-* مربوط به HTML 5 است. برای مثال اگر صرفاً هدف مشخص سازی Url و یا اطلاعاتی از این دست است، بهتر است این موارد را داخل فایل اسکریپت قرار نداد. در همان View معمولی یک ویژگی data سفارشی را ایجاد کنید:

```
<div data-url="@Url.Action(...)">  
</div>
```

و بعد در فایل اسکریپت خارجی [به این نحو](#) قابل خواندن خواهد بود:

```
var url = $("div").data("url") ;
```

SASS چیست؟

SASS مخفف Syntactically Awesome Style Sheets است که توسط آقای [Hampton Catlin](#) طراحی و ایجاد شده است و همانند CoffeeScript که پس از کامپایل به جاوااسکریپت تبدیل می‌شد، SASS نیز پس از کامپایل به CSS تبدیل می‌شود. SASS با استفاده از متغیرها، mixins، ارث بری و قوانین تودرتو، CSS را با مهارت زیادی در بهترین حالت تولید می‌کند.

SASS باعث کمتر نوشتن کد CSS، سبب افزایش خوانایی و دستکاری کردن راحت‌تر و پویای آن می‌شود. این مساله راهی عالی برای نوشتن کدهای CSS کاربردی‌تر است و می‌تواند سرعت گردش کار هر توسعه دهنده و یا طراح وب را افزایش دهد.

وقتی اولین بار SASS عرضه شد، syntax آن تفاوت قابل توجهی با CSS داشت (پسوند فایل‌های آن **SASS** است) که به جای نوشتن براکت‌ها، از تورفتگی استفاده می‌شد و دیگر نیازی به نوشتن " " نبود. البته با عدم استقبال از این syntax مواجه شد و با عرضه نسخه 3 SASS، (که پسوند فایل‌های آن **SCSS** است) syntax آن بسیار شبیه به CSS شد؛ البته با همه‌ی ویژگی‌های SASS.

برای مثال کد CSS زیر را می‌خواهیم به دو روش بنویسیم:

```
header {
  margin: 0;
  padding: 0;
  color: #fff;
}
```

با استفاده از روش SCSS. (روش جدید)

```
$color: #fff;
header {
  margin: 0;
  padding: 0;
  color: $color;
}
```

با استفاده از روش SASS. (روش قدیم)

```
$color: #fff
header
  margin: 0
  padding: 0
  color: $color
```

همانطور که مشاهده می‌کنید برای نوشتن مقدار color از متغیر \$color استفاده کردیم. در ادامه به قابلیت‌های SASS خواهیم پرداخت.

توجه: syntax ایی که در این سری آموزشی با آن کار می‌کنیم **SCSS** است.

کامپایل کردن SASS روش‌های مختلفی برای کامپایل فایل‌های SASS وجود دارند:
روش اصلی استفاده از SASS در Ruby است که پس از نصب [Ruby](#) و اجرای فرمان **gem install sass**، SASS نصب می‌شود و برای کامپایل، اجرای فرمان زیر:

```
sass myfile.scss myfile.css
```

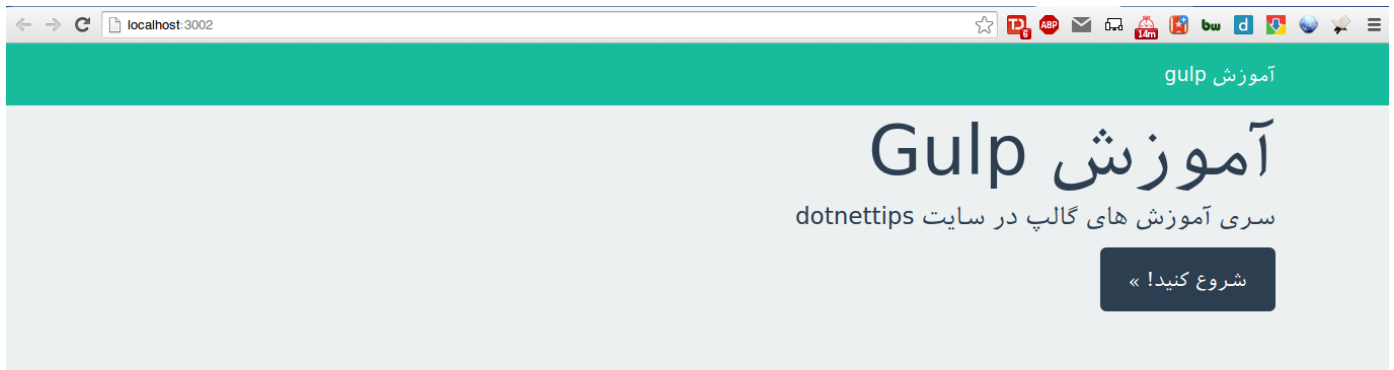

استفاده از برنامه‌های گرافیکی مانند [CodeKit](#) , [Hammer](#) و [Compass](#) .
استفاده از برنامه‌های رایگان مانند [libsass](#) که با یک کامپایلر سریع نوشته شده با C/C++ است و همچنین می‌توانید [libsass](#) را از طریق NPM با [node-sass](#) نصب کنید.

```
npm install node-sass
```

استفاده از افزونه [Web Essentials](#) در Visual Studio
نکته: در صورتیکه می‌خواهید با استفاده از Ruby کار کامپایل را انجام دهید در هنگام نصب Ruby گزینه‌ی "Add Ruby executables to your PATH" را تیک بزنید.

خب سوالی که ممکن است برای شما پیش آمده باشد این است که باید از کدام یک از این روش‌ها را استفاده کنیم؟
بستگی به این دارد که شما چه کاری را می‌خواهید انجام دهید.
در صورتیکه بر روی یک پروژه‌ی بزرگ با میزان کد زیاد کار می‌کنید، استفاده از Ruby SASS، کمی کند کار کامپایل را انجام می‌دهد.
اگر بخواهید از [libsass](#) استفاده کنید، این مسئله وجود دارد که به طور 100% با قابلیت‌های Ruby SASS برابری ندارد.
در صورتیکه نمی‌خواهید از command line استفاده کنید، برنامه‌های گرافیکی گزینه‌ای عالی هستند. شما می‌توانید طوری تنظیم کنید که تمامی تغییراتی که در فایل SASS انجام می‌شود، به صورت خودکار کار کامپایل انجام شود.
اگر هم فقط می‌خواهید کدی را که نوشته‌اید تست کنید، می‌توانید از ابزارهای آنلاین مانند [SassMeister](#) استفاده کنید.

همانطور که در [مقاله ی قبلی](#) پایه ی ورک فلو ی خود را راه اندازی کردیم، در این مقاله می‌خواهیم با طراحی یک صفحه، با بوت استرپ شخصی سازی شده، در عمل با کارایی گالپ آشنا شویم.
دمو پایانی:



قسمت اول	قسمت دوم	قسمت سوم
Gulp ابزاری ست که شما را در انجام دادن کارهای مختلف توسعه ی وب، در سمت Front-end کمک می‌کند و اغلب برای کارهایی همچون موارد ذیل بکار می‌رود:	در این مقاله و مقالات بعد می‌خواهیم کار خود را با راه اندازی یک workflow برای بوت استرپ، روند شخصی سازی آن را بسیار آسان و لذت بخش‌تر کنیم. ...	در قسمت اول گالپ را معرفی کردیم و در مقاله قبلی به نوشتن اولین تسک با گالپ پرداختیم. در این قسمت می‌خواهیم با نصب bower، پروژه ی workflow بوت استرپ راستچین شده را انجام دهیم.
شروع کنید! »	شروع کنید! »	شروع کنید! »

به هنگام سازی مرورگر و بارگذاری مجدد به صورت خودکار

یکی از موارد فوق العاده تکراری در هنگام توسعه ی وب، برای یک توسعه دهنده سمت کاربر (Front end Developer) ریلود کردن مرورگر است. همچنین تست وب سایت یا آپلود در موبایل و سایر دستگاه‌ها، متداول است. با پلاگین گالپ می‌توان این مشکل را به صورت بهینه‌ای حل کرد.

نصب

برای نصب دستور زیر را در مسیر پروژه، در ترمینال سیستم عامل خود وارد کنید.

```
npm install browser-sync gulp --save-dev
```

می‌دانیم برای استفاده از یک پلاگین باید توسط متد require آن را به یک متغیر انتساب دهیم؛ به صورت زیر:

```
var gulp = require('gulp'),
    sass = require('gulp-ruby-sass'),
    notify = require('gulp-notify'),
    browserSync = require('browser-sync'), // Add browser syncs plugin
    bower = require('gulp-bower');
```

توجه کنید هر پلاگینی که اضافه می‌کنید، باید تسک مربوط به آن را بنویسیم تا بتوانیم از آن استفاده کنیم. برای این پلاگین فقط مشخص کردن مسیر root سرور کافی است.

```
gulp.task('browserSync', function() {
  browserSync({
    server: {
      baseDir: './' //our server root
    }
  });
});
```

حال می‌خواهیم با زدن `gulp watch`، تمام کارهای ما به صورت خودکار انجام شوند. اما این دستور که در جلسه‌ی قبل آن‌را تعریف کردیم، فقط منتظر انجام یک تغییر است. تسک `watch` را به گونه‌ای تغییر می‌دهیم که ابتدا تسک‌های `browser sync`، `css`، انجام شوند (به دلیل اینکه باید ابتدا، سرور راه اندازی شود) سپس گالپ منتظر تغییرات باشد و آنها را اعمال کند.

```
gulp.task('watch', [ 'css','browserSync'], function() {
})
```

تسک‌های `html`، `css`، `browserSync` قبل از تسک `watch` اجرا می‌شوند. طبق مستندات، این پلاگین یکی از توابع `API` [متد watch](#) است و کار آن همانند متد مشابهی در گالپ است. آن را برای ریلود خودکار مرورگر استفاده می‌کنیم.

```
// Rerun the task when a file changes
gulp.task('watch', ['html', 'css', 'browserSync'], function() {
  gulp.watch(config.sassPath + '/*.scss', ['css']);
  gulp.watch(config.htmlPath, ['html'])
  browserSync.watch("./*.html").on("change", browserSync.reload); // browserSync watch task
});
```

می‌خواهیم بعد از کامپایل، فایل‌های `sass` هم مرورگر دوباره بارگذاری شوند. کد زیر را به انتهای تسک `css` اضافه می‌کنیم:

```
.pipe(browserSync.reload({
  stream: true
}));
```

بسیار خوب با انجام این کارها پلاگین باید به درستی کار کند.

شخصی سازی بوت استرپ

برای شخصی سازی بوت استرپ کافی است ابتدا فایل‌های `sass` بوت استرپ و `FontAwesome` را در `style.scss` ایمپورت کنیم؛ به این صورت:

```
@import "bootstrap";
@import "font-awesome";
```

حال دستور `gulp` را می‌زنیم. با اینکار فایل `style.scss` کامپایل می‌شود. می‌خواهیم یک فونت فارسی و یک قالب فلت را به پروژه‌امان اعمال کنیم. من فایل‌ها را اضافه کرده‌ام و شما با یک نگاه می‌توانید، چیزی را که گفتم درک کنید.

```
@import "fonts-fa";
@import "variable";
@import "bootstrap";
@import "font-awesome";
@import "rtl.scss";
@import "typography";
```

نکته : سعی کنید برای استایل هر قسمت، یک فایل مجزا درست کنید؛ مانند مثال بالا که در پروژه لحاظ شده. برای توسعه‌ی پروژه، ابتدا مخزن گیت هاب را فورک کرده و با زدن دستورات زیر کار خود را آغاز کنید:

```
sudo npm install
      gulp
      gulp watch
```

```
mmdsharifi@mmdsharifi ~/Work/Gulp-RTLbootstrap-fontawesome $ gulp watch
[12:58:37] Using gulpfile ~/Work/Gulp-RTLbootstrap-fontawesome/gulpfile.js
[12:58:37] Starting 'css'...
[12:58:37] Starting 'browserSync'...
[12:58:37] Finished 'browserSync' after 43 ms
[BS] Access URLs:
-----
      Local: http://localhost:3002
      External: http://192.168.1.7:3002
-----
      UI: http://localhost:3003
      UI External: http://192.168.1.7:3003
-----
[BS] Serving files from: ./
[12:58:39] Finished 'css' after 1.9 s
[12:58:39] Starting 'watch'...
[12:58:39] Finished 'watch' after 37 ms
[13:00:41] Starting 'css'...
[BS] 1 file changed (style.css)
[13:00:43] Finished 'css' after 1.62 s
[BS] File changed: index.html
[BS] File changed: index.html
[13:05:52] Starting 'css'...
[BS] 1 file changed (style.css)
[13:05:54] Finished 'css' after 1.83 s
[13:06:43] Starting 'css'...
[BS] 1 file changed (style.css)
[13:06:44] Finished 'css' after 1.69 s
[13:09:03] Starting 'css'...
```

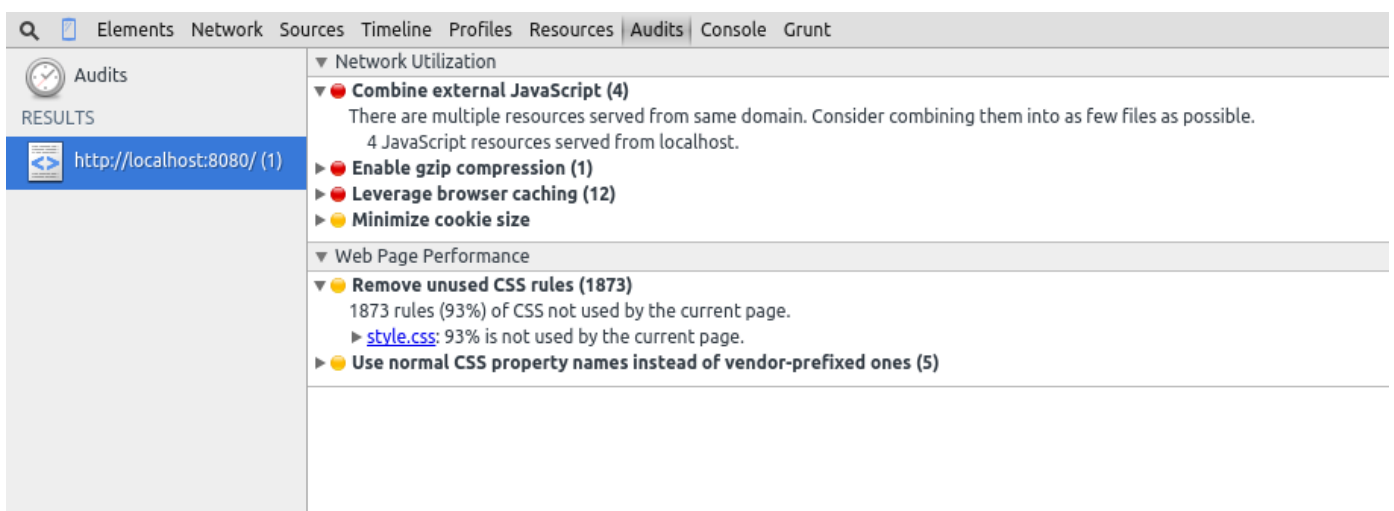
مخزن گیت هاب : کامیت :

Add : browserSync plugin and index.html

در [مقالات قبلی](#) به طور کامل با گالپ آشنا شدیم و گفتیم که می‌تواند ما را در بهینه سازی ورک فلویمان کمک کند. در این قسمت یاد خواهیم گرفت که چگونه تجربه‌ی کاربری بهتری را از سرعت بارگذاری سایتمان ایجاد کنیم.

افزایش کارآیی Performance وب با گالپ

برای اینکه بفهمیم چه کارهایی می‌تواند سایت یا اپلیکیشن ما را کاراتر کند، از Developer tools با زدن Ctrl+Shift+I درون گوگل کروم، کار خود را شروع می‌کنیم. به برگه‌ی Audits می‌رویم و دکمه‌ی Run را با تنظیمات پیش فرض می‌زنیم. نتایج آن بعد از اندکی صبر، برای من به صورت شکل زیر است:



ما قصد داریم بدانیم گالپ چه ابزاهایی را برای راه حل‌های داده شده توسط مرورگر دارد؟

۱- کنار هم قرار دادن و فشرده کردن فایل‌های جاوا اسکریپت

پلاگین‌های گالپ برای اینکار، [gulp-concat](#) و [gulp-uglify](#) هستند. آنها را در مسیر ریشه‌ی پروژه نصب می‌کنیم.

```
npm install --save-dev gulp-uglify gulp-concat
```

بعد از نصب، فایل gulpfile.js را به صورت زیر ویرایش و تسک js را به آن اضافه می‌کنیم.

```
// first load all required js files
// concat them in to script.min.js
// and minify it.
gulp.task('js', function() {
  return gulp.src([
    config.bowerDir + '/jquery/dist/jquery.min.js', // این فایل وابستگی فایل‌های زیر است
    config.bowerDir + '/materialize/dist/js/materialize.min.js',
    './resources/js/app.js'
  ])
  .pipe(concat('script.min.js'))
  .pipe(uglify())
  .pipe(size())
  .pipe(gulp.dest('./public/js'));
});
```

خروجی:

```
mmdsharifi@mmdsharifi ~/Work/Lenus $ gulp js
[13:27:20] Using gulpfile ~/Work/Lenus/gulpfile.js
[13:27:20] Starting 'js'...
[13:27:23] all files 206.05 kB
[13:27:23] Finished 'js' after 3.18 s
```

فشرده کردن جاوا اسکریپت، حجم فایل‌ها را ۳۰ تا ۹۰ درصد [کاهش](#) می‌دهد.

۲- حذف سکتورهای بدون استفاده css

همانگونه که در عکس اول آمده، ۹۳ درصد سکتورها در این صفحه بلا استفاده هستند! و این یعنی کاهش فوق العاده زیاد حجم فایل فشرده شده css. به طور معمول، توسعه دهندگان ۸۵ درصد حجم فایل css خود را می‌توانند با این کار [کاهش](#) دهند (البته بیشتر این اتفاق هنگام استفاده از فریک ورک‌هایی مانند bootstrap, ... می‌افتد) برای این کار از پلاگین [gulp-uncss](#) استفاده می‌کنیم. نصب:

```
npm install gulp-uncss --save-dev
```

سپس تسک مربوطه را می‌نویسیم:

```
gulp.task('css', function() {
  return sass(config.sassPath + '/style.scss', {
    style: 'compressed',
    loadPath: [
      './resources/sass',
      config.bowerDir + '/materialize/sass'
    ]
  })
  .on('error', util.log)
  .pipe(size())
  .pipe(uncss({
    html: ['./index.html', './posts.html']
  }))
  .pipe(gulp.dest('./public/css'))
  .pipe(size())
  .pipe(connect.reload());
});
```

نتیجه:

```
mmdsharifi@mmdsharifi ~/Work/Lenus $ gulp css
[14:24:58] Using gulpfile ~/Work/Lenus/gulpfile.js
[14:24:58] Starting 'css'...
[14:25:01] all files 136.68 kB
[14:25:03] all files 17.34 kB
[14:25:03] Finished 'css' after 4.46 s
```

نتیجه فوق العاده است! **۸۷ درصد کاهش حجم css!** اما ممکن است بعضی از استایل‌های شما توسط javascript به صفحه تزریق شوند. در این صورت نباید سکتورهای لازم را حذف کرد و آنها را داخل آرایه‌ی ignore قرار می‌دهیم. برای این منظور، تسک بالا را به صورت زیر به روز رسانی می‌کنیم.

```

gulp.task('css', function() {
  return sass(config.sassPath + '/style.scss', {
    style: 'compressed',
    loadPath: [
      './resources/sass',
      config.bowerDir + '/materialize/sass'
    ]
  })
  .on('error', util.log)
  .pipe(size())
  .pipe(uncss({
    html: ['./index.html', './posts.html'],
    timeout: 2000, // wait for load js files
    ignore: [
      ".waves-ripple ",
      ".drag-target",
      "#sidenav-overlay",
      ".waves-effect",
      ".waves-effect .waves-ripple",
      ".waves-effect.waves-pinck .waves-ripple",
      ".waves-block.waves-light"
    ]
  }))
  .pipe(minifyCss())
  .pipe(size())
  .pipe(gulp.dest('./public/css'))
  .pipe(connect.reload());
});

```

بعد از اینکار حجم فایل css من کمی افزایش پیدا کرد ولی بعد از فشرده کردن، نهایتاً به حدود ۱۴KB رسید و این یعنی ۸۷ درصد کاهش حجم فایل css؛ تنها بعد از حذف سکلتورهای اضافی و فشرده کردن آنها. می‌توانید [پلاگین‌های بیشتری](#) را در اینجا ببینید و استفاده کنید.

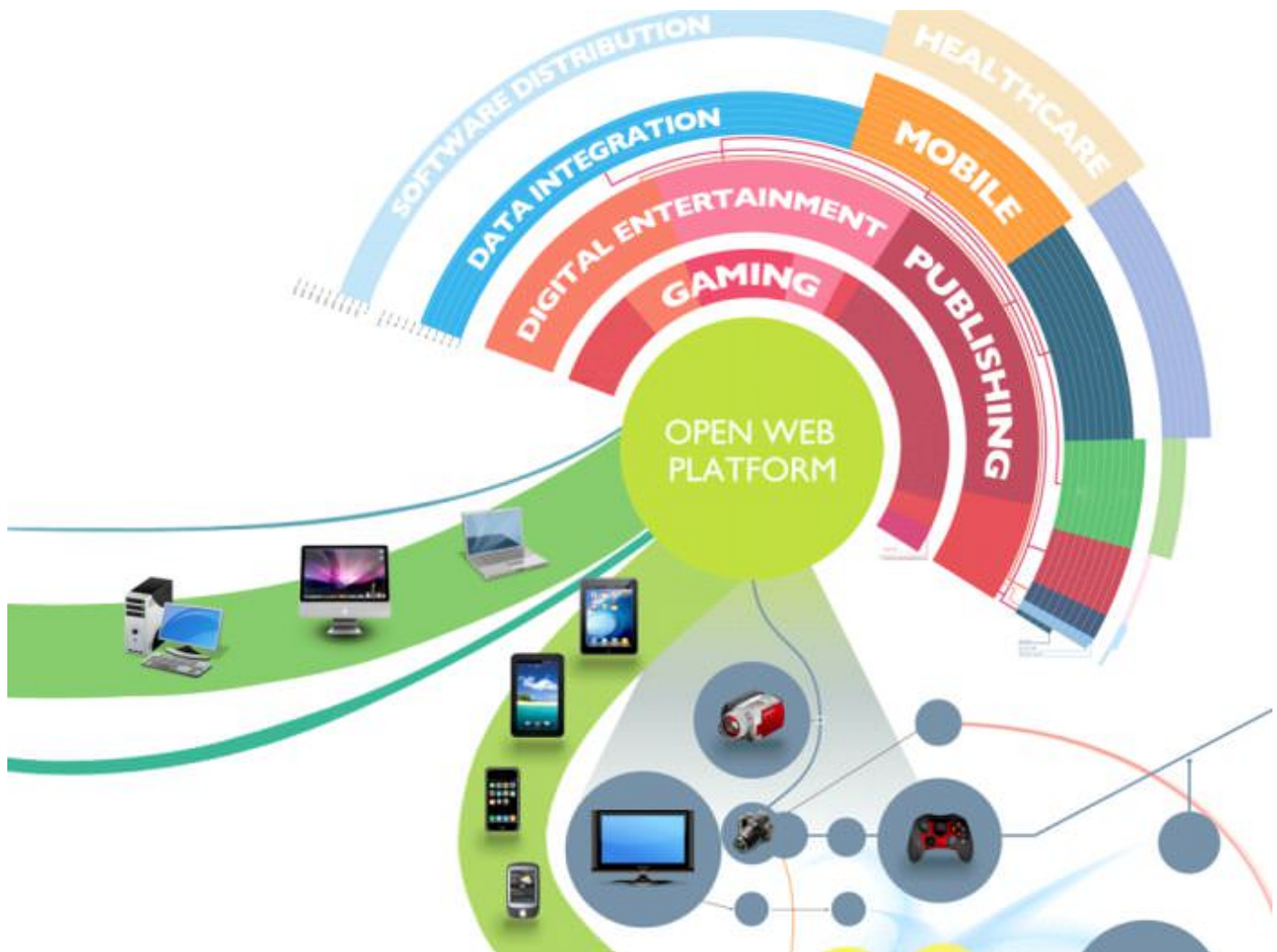
[Gist](#)

چند روز پیش [مطلبی](#) به عنوان اشتراک در سایت جاری معرفی شده که به ما یادآوری می‌کرد، ما تنها استفاده کنندگان سیستم‌های کامپیوتری، به خصوص اینترنت نیستیم و معلولین هم نیازمند استفاده از این فناوری‌ها هستند.

[WAI-ARIA](#) که برگرفته از *Web Accessibility Initiative - Accessible Rich internet Application* است به معنی برنامه‌ی اینترنتی تعامل گرا با خاصیت دسترسی پذیری بالا می‌باشد و یک راهنماست که توسط [کنسرسیوم وب \(+\)](#) معرفی گشته است تا وب سایت‌ها با رعایت این قوانین، دسترسی سایت خود را بالاتر ببرند. این قوانین به خصوص برای سایت‌هایی با محتوای پویا هستند که از فناوری‌هایی چون Ajax, Javascript, HTML و دیگر فناوری‌های مرتبط استفاده می‌کنند.

امروزه طراحان وب بیش از هر وقتی از فناوری‌های سمت کلاینت چون جاوااسکریپت برای ساخت رابط‌های کاربری استفاده می‌کنند که html به تنها قادر به ایجاد آن‌ها نیست. یکی از تکنیک‌های جاوااسکریپت، دریافت محتوای جدید و به روزآوری قسمتی از صفحات وب است بدون اینکه مجدداً کل صفحه از وب سرور درخواست گردد که به این تکنیک **Rich Internet Application** هم می‌گویند. تا به اینجای کار هیچ مشکلی نیست و خوب هم هست؛ ولی مشکلی که در این بین وجود دارد این است که این نوع تکنیک‌ها باعث از بین رفتن خاصیت دسترسی پذیری معلولین می‌گردند. معلولینی که از [صفحه خوان‌ها](#) استفاده می‌کنند یا به دلیل معلولیت‌های خود قادر به حرکت دادن ماوس نیستند.

ARIA با استفاده از خصوصیت‌ها Properties، نقش‌ها Roles و وضعیت‌ها States به طراحان برنامه‌های وب و سازندگان فناوری‌های یاری رسان، اجازه می‌دهد که با ابزارهای کمکی معلولان ارتباط برقرار کنیم و یک صفحه‌ی وب ساده را به یک صفحه‌ی پویا تبدیل کنیم. ARIA تنها یک استاندارد برای وب نیست، بلکه یک فناوری چند پلتفرم است که برای بازی‌های رایانه‌ای، موبایل‌ها، دستگاه‌های سرگرمی و سلامتی و دیگر انواع برنامه‌ها نیز تعریف شده است.



فریمورک ARIA

خود HTML به تنهایی نمی‌تواند نقش‌های هر المان و ارتباط بین آن‌ها را به درستی بیان کند و به این منظور ARIA به کمک می‌آید. با استفاده از نقش‌ها میتوان هدف هر المان را مشخص کرد و با استفاده از خصوصیات ARIA نحوه‌ی عملکرد آن‌ها را تعریف کرد.

نقش‌ها

نقش‌ها طبق مستندات کنسرسیوم وب بر 4 نوع هستند:

Abstract Roles

Widget Roles

Document Roles

Landmark Roles

فریمورک ARIA با استفاده از Landmark Roles یا نقش‌های راهنما، به معرفی بخش‌های ویژوالی می‌پردازد تا فناوری‌های کمکی به آن دسترسی سریعی داشته باشند. هشت نقش راهنما وجود دارد که در زیر آن‌ها را بررسی می‌کنیم.

این قسمت که عموماً برای اجزای مهمی مثل هدر سایت قرار می‌گیرد و شامل معرفی وب سایت هست و در تمامی صفحات وجود دارد که شامل لوگو، اطلاعات عمومی سایت و اسپانسرها و ... می‌گردد و بسیار مهم است که تنها یکبار در صفحه‌ی وب به کار برود و تکرار آن پرهیز شود.	Banner
این نقش به محتوای اصلی وب سایت اشاره می‌کند و نباید بیشتر از یکبار در هر صفحه‌ی وب به کار برود و عموماً بهتر است این خصوصیت در تگ div قرار گیرد: <code><div Role="main"></div></code> یا در HTML5 به طور مفهومی ساخته می‌شود: <code><main role="main">.....</code>	Main
اشاره به یک ناحیه پر از المان‌های لینک برای ارتباط با صفحات دیگر	Navigation
مشخص سازی ناحیه‌ای که اطلاعات اضافی درباره‌ی محتوای اصلی سایت دارد؛ مانند بخش مقالات مرتبط، آخرین کامنت‌ها و ...	Complementary
این نقش که بیشتر برای فوتر مناسب است برای محتوایی به کار می‌رود که در آن به قوانین کپی رایت و ... اشاره می‌شود.	ContentInfo
برای اشاره به فرم‌ها که دارای قسمت‌های ورودی کاربر هستند.	form
در صورتیکه فرمی دارید و از آن برای گزینه‌ی جست و جو استفاده می‌کنید، از این نقش استفاده کنید.	search
برای اینکه وب سایت خود را به صورت یک وب اپلیکیشن معرفی کنید؛ تا یک صفحه وب معمولی، استفاده می‌شود و برای وب سایت‌های قدیمی یا با حالت سنتی توصیه نمی‌شود و به برنامه‌های کمکیار معلولین می‌گویند که از حالت عادی به حالت application سوئیچ کنند؛ پس با دقت بیشتری باید از این گزینه استفاده کرد.	application



خصوصیت‌ها و وضعیت‌ها

در حالیکه از نقش‌ها برای معرفی هر المان یا تگ استفاده می‌کنید؛ خصوصیت‌ها و وضعیت‌ها به کاربر اطلاعات اضافی می‌دهند که چگونه از آن استفاده کنند. برای معرفی خصوصیت‌ها و وضعیت‌ها از یک خصوصیت که با `aria-` شروع می‌شود استفاده کنید. از معروفترین آن‌ها خصوصیت `aria-required` و وضعیت `aria-checked` می‌باشند، تا به ترتیب به کاربر اعلام کنید این تگ نیاز به پر شدن دارد، یا المانی نیاز به تغییر وضعیت انتخابی دارد. نحوه‌ی استفاده از آن‌ها به شکل زیر است:

```
<div id="some-id" class="some-class" aria-live="assertive"><div>
```

`aria-live` سه مقدار می‌پذیرد که عبارتند از `Off`, `Polite`, `Assertive` و مشخص می‌کنند که المان مورد نظر آپدیت پذیر هست یا خیر و اگر آری، نحوه‌ی به روز شوندگی آن به چه نحوی است.

ساخت ارتباطات میان المان‌ها با خصوصیت‌های ارتباطی

مثال شماره یک

```
<div role="main" aria-labelledby="some-id">
  <h1 id="some-id">This Is A Heading</h1>
  Main content...
</div>
```

خصوصیت `aria-labelledby` به تعریف المان‌هایی با نام جاری می‌پردازد. این خصوصیت معرفی کننده‌ی برچسب هاست. به

عنوان مثال بین المان‌ها ورودی و برچسب آنان ارتباط ایجاد می‌کند تا ابزارهای معلولین، مانند صفحه خوان‌ها، در خواندن به مشکل برنخورند.

مثال شماره دو

در این مثال هر گروهی از المان‌ها یک برچسب و بعضی المان‌ها یک برچسب اختصاصی دارند که توسط خصوصیت معرفی شده aria-labelledby کامل شده‌اند:

```
<div id="billing">Billing Address</div>

<div>
  <div id="name">Name</div>
  <input type="text" aria-labelledby="name billing"/>
</div>
<div>
  <div id="address">Address</div>
  <input type="text" aria-labelledby="address billing"/>
</div>
```

مثال شماره سه

در این مثال گروهی از radio button با برچسبشان ارتباط برقرار می‌کنند.

```
<div id="radio_label">My radio label</div>
<ul role="radiogroup" aria-labelledby="radio_label">
  <li role="radio">Item #1</li>
  <li role="radio">Item #2</li>
  <li role="radio">Item #3</li>
</ul>
```

خصوصیت‌ها و وضعیت‌های aria را با چرخه‌ی فعالیت‌های صفحه به روز کنید

در صورتیکه چرخه‌ی فعالیت صفحه‌ی شما تغییر می‌کند و تگ‌ها نیاز به مقادیر جدیدی از aria دارند، حتما این مقادیر را هم به نسبت تغییراتی که در صفحه رخ می‌دهد، تغییر دهید تا وضعیت بحرانی برای کاربر به خصوص در حین کار با فرم‌ها و ... پیش نیاید.

هر aria را دوباره استفاده نکنید

امروزه به خصوص با آمدن html5 و ویژگی‌هایی چون تگ‌های مفهومی، کار بسیار راحت‌تر شده‌است و مرورگر به طور خودکار می‌تواند aria را بر روی بعضی از المان‌ها پیاده کند. به عنوان نمونه:

```
<form></form>
<form role="form"></form>
```

همچنین به عنوان مثال با استفاده از خصوصیات HTML مثل hidden کردن یک شیء نیازی به استفاده از وضعیت aria-hidden نمی‌باشد. مرورگر به طور پیش فرض آن را لحاظ می‌کند.

امروزه شاهد پیشرفت فناوری در همه‌ی عرصه‌ها هستیم و همیشه این پیشرفت‌ها ما را ذوق زده کرده‌اند، ولی یکی از بی نظیرترین استفاده‌های فناوری روز، استفاده در صنایع سلامتی است که نه تنها ما را ذوق زده می‌کند، بلکه از لحاظ احساسی هم ما را به وجد می‌آورد و جزء زیباترین نتایج فناوری می‌باشند. بسیاری از شرکت‌ها چون گوگل در این راستا فعالیت‌های زیادی کرده‌اند تا بتوانند سلامت جامعه را کنترل کنند، از ساخت [لنز چشمی برای کنترل دیابت](#) گرفته تا [ساخت قاشق خوری برای بیماران پارکینسون](#)، ولی استفاده‌های ساده از مسائلی مانند بالا به افراد معلول این مژده را می‌دهد که ما آن‌ها را فراموش نکرده‌ایم.