

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>
```

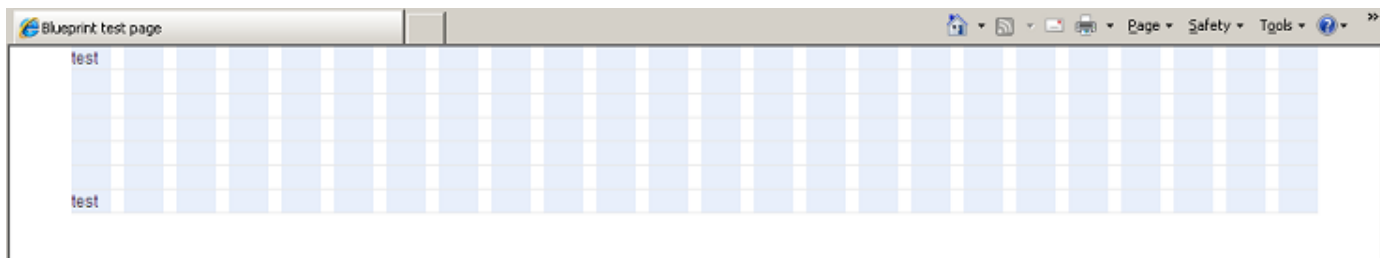
توضیحات:

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):

:border و prepend-n

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.

شبیه 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](#) .

Twitter Bootstrap یک فریم ورک CSS بسیار محبوب سورس باز تولید برنامه‌های وب به کمک CSS، HTML و جاوا اسکریپت است. این فریم ورک حاوی بسیاری از المان‌های مورد نیاز جهت تولید وب سایت‌هایی زیبا، مانند دکمه‌ها، عناصر فرم‌ها، منوها، ویجت‌ها و غیره است. تمام این‌ها نیز همانطور که عنوان شد بر مبنای CSS، HTML و جاوا اسکریپت تهیه شده‌اند؛ بنابراین در هر نوع فناوری سمت سروری مانند PHP، ASP.NET، روبی و امثال آن قابل استفاده است.

دریافت Twitter Bootstrap

محل اصلی دریافت Twitter Bootstrap، آدرس ذیل است:

<http://twitter.github.com/bootstrap>

البته ما از آن در اینجا به شکل خام فوق استفاده نخواهیم کرد؛ زیرا نیاز است قابلیت‌های استفاده در محیط‌های راست به چپ فارسی نیز به آن اضافه شود. برای این منظور می‌توانید از یکی از دو بسته نیوگت ذیل استفاده نمایید:

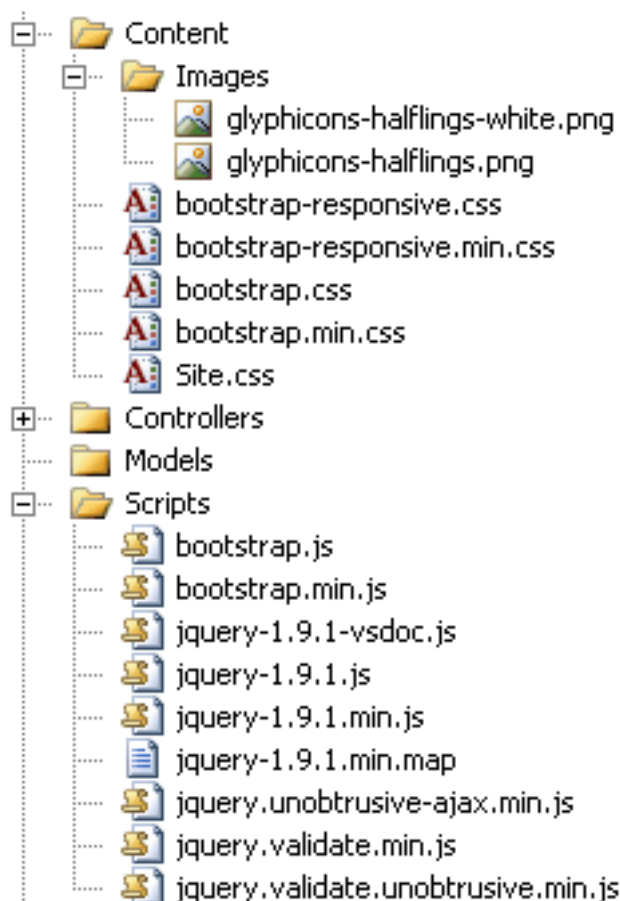
Twitter.BootstrapRTL, <https://nuget.org/packages/Twitter.BootstrapRTL>

RTL Twitter Bootstrap, <https://nuget.org/packages/Twitter.Bootstrap.RTL>

و یا حتی از منابع سایت <http://rbootstrap.ir> نیز می‌توان استفاده کرد. برای نمونه دستور زیر را در کنسول پاورشل ویژوال استودیو وارد نمایید تا اسکریپت‌ها و فایل‌های CSS مورد نیاز به پروژه جاری اضافه شوند:

```
PM> Install-Package Twitter.BootstrapRTL
```

پس از نصب، شاهد افزوده شدن چنین فایل‌هایی به یک پروژه ASP.NET MVC خواهیم بود:



در اینجا فایل‌های min، نگارش‌های فشرده شده فایل‌های js یا css هستند که با توجه به امکانات اضافه شده به ASP.NET MVC4، از آن‌ها استفاده نخواهیم کرد و برای افزودن و تعریف آن‌ها از امکانات Bundling and minification توکار فریم ورک ASP.NET MVC به نحوی که در ادامه توضیح داده خواهد شد، استفاده می‌کنیم.

فایل‌های png اضافه شده، آیکون‌های مخصوص Twitter Bootstrap هستند که اصطلاحاً به آن‌ها sprite images نیز گفته می‌شود. در این نوع تصاویر، تعداد زیادی آیکون در کنار هم، برای بهینه سازی تعداد بار رفت و برگشت به سرور جهت دریافت تصاویر، طراحی شده و قرار گرفته‌اند.

فایل‌های js این مجموعه اختیاری بوده و برای استفاده از ویجت‌های Twitter Bootstrap مانند آکاردئون کاربرد دارند. این فایل‌ها برای اجرا، نیاز به jQuery خواهند داشت.

افزودن تعاریف اولیه Twitter Bootstrap به یک پروژه ASP.NET MVC

امکانات Bundling and minification در نوع پروژه‌های نسبتاً جامع‌تر ASP.NET MVC به صورت پیش فرض لحاظ شده است. اما اگر یک پروژه خالی را شروع کرده‌اید، نیاز است بسته نیوگت آن‌را نیز نصب کنید:

```
PM> Install-Package Microsoft.AspNet.Web.Optimization
```

پس از آن، کلاس کمکی BundleConfig ذیل را به پروژه جاری اضافه نمایید. به کمک آن قصد داریم امکانات Bundling and minification را فعال کرده و همچنین به آن بگوییم اسکریپت‌ها و فایل‌های CSS تعریف شده را به همان نحوی که معرفی می‌کنیم، پردازش کن (توسط کلاس سفارشی AsIsBundleOrderer).

```
using System.Collections.Generic;
using System.IO;
```



```

using System.Web;
using System.Web.Optimization;

namespace Mvc4TwitterBootstrapTest.Helper
{
    /// <summary>
    /// A custom bundle orderer (IBundleOrderer) that will ensure bundles are
    /// included in the order you register them.
    /// </summary>
    public class AsIsBundleOrderer : IBundleOrderer
    {
        public IEnumerable<FileInfo> OrderFiles(BundleContext context, IEnumerable<FileInfo> files)
        {
            return files;
        }
    }

    public static class BundleConfig
    {
        private static void addBundle(string virtualPath, bool isCss, params string[] files)
        {
            BundleTable.EnableOptimizations = true;

            var existing = BundleTable.Bundles.GetBundleFor(virtualPath);
            if (existing != null)
                return;

            Bundle newBundle;
            if (HttpContext.Current.IsDebuggingEnabled)
            {
                newBundle = new Bundle(virtualPath);
            }
            else
            {
                newBundle = isCss ? new Bundle(virtualPath, new CssMinify()) : new Bundle(virtualPath,
                new JsMinify());
            }
            newBundle.Orderer = new AsIsBundleOrderer();

            foreach (var file in files)
                newBundle.Include(file);

            BundleTable.Bundles.Add(newBundle);
        }

        public static IHtmlString AddScripts(string virtualPath, params string[] files)
        {
            addBundle(virtualPath, false, files);
            return Scripts.Render(virtualPath);
        }

        public static IHtmlString AddStyles(string virtualPath, params string[] files)
        {
            addBundle(virtualPath, true, files);
            return Styles.Render(virtualPath);
        }

        public static IHtmlString AddScriptUrl(string virtualPath, params string[] files)
        {
            addBundle(virtualPath, false, files);
            return Scripts.Url(virtualPath);
        }

        public static IHtmlString AddStyleUrl(string virtualPath, params string[] files)
        {
            addBundle(virtualPath, true, files);
            return Styles.Url(virtualPath);
        }
    }
}

```

نکته دیگری که در این کلاس سفارشی در نظر گرفته شده، عدم فعال سازی مباحث Bundling and minification در حالت Debug است. اگر در وب کانفیگ، تنظیمات پروژه را بر روی Release قرار دهید فعال خواهد شد (مناسب برای توزیع) و در حالت توسعه برنامه (حالت دیباگ)، این اسکریپت‌ها و فایل‌های CSS، قابلیت دیباگ و بررسی را نیز خواهند داشت.

پس از افزودن کلاس‌های کمکی فوق، به فایل layout پروژه مراجعه کرده و تعاریف ذیل را به ابتدای فایل اضافه نمائید:

```
@using Mvc4TwitterBootstrapTest.Helper
<!DOCTYPE html>
<html>
<head>
    <title>@ViewBag.Title</title>
    @BundleConfig.AddStyles("~/Content/css",
        "~/Content/bootstrap.css",
        "~/Content/bootstrap-responsive.css",
        "~/Content/Site.css"
    )
    @BundleConfig.AddScripts("~/Scripts/js",
        "~/Scripts/jquery-1.9.1.min.js",
        "~/Scripts/jquery.validate.min.js",
        "~/Scripts/jquery.unobtrusive-ajax.min.js",
        "~/Scripts/jquery.validate.unobtrusive.min.js",
        "~/Scripts/bootstrap.min.js"
    )
    @RenderSection("JavaScript", required: false)
</head>
```

و اگر برنامه را اجرا کنید، بجای حداقل 8 مدخل اشاره کننده به فایل‌های اسکریپت و CSS مورد نیاز یک برنامه ASP.NET MVC، فقط دو مدخل ذیل را مشاهده خواهید نمود:

```
<!DOCTYPE html>
<html>
<head>
    <title>Index</title>
    <link href="/Content/css?v=vsUQD00Jg4AJ-RZH8jSRRCu_rj12U1nZrmSsaUyxAc1" rel="stylesheet"/>
    <script src="/Scripts/js?v=GezdoTDiWY3acc3mI2Ujm_7nKKzh6Lu1Wr8TGyyLpW41"></script>
</head>
```

به این ترتیب تعداد رفت و برگشت‌های مرورگر به سرور، برای دریافت فایل‌های مورد نیاز جهت رندر صحیح یک صفحه، به شدت کاهش خواهد یافت. همچنین این فایل‌ها در حالت release فشرده نیز خواهند شد؛ به همراه تنظیم خودکار هدر کش شدن آن‌ها برای مدتی مشخص، جهت کاهش بار سرور و کاهش پهنای باند مصرفی آن.

مفاهیم پایه‌ای Twitter Bootstrap

الف) Semantic class names

به عبارتی کلاس‌های Twitter Bootstrap دارای نام‌هایی معنا دار و مفهومی می‌باشند؛ مانند کلاس‌های CSS ای، به نام‌های Success، Error، Info و امثال آن. این نام‌ها مفهومی را می‌رسانند؛ اما در مورد نحوه پیاده سازی آن‌ها جزئیاتی را بیان نمی‌کنند. برای نمونه می‌توان کلاسی را به نام redText ایجاد کرد. هر چند این نام، توضیحاتی را در مورد علت وجودی‌اش بیان می‌کند، اما بسیار ویژه بوده و در مورد جزئیات پیاده سازی آن نیز اطلاعاتی را ارائه می‌دهد. در این حالت redText معنایی ندارد. چرا یک Text باید قرمز باشد؟ برای مثال این متن قرمز است چون مثلاً شخصی، به آن رنگ ویژه علاقه دارد، یا اینکه قرمز است بخاطر نمایش خطایی در صفحه؟ به همین جهت در Twitter Bootstrap از نام‌های مفهومی یاده شده، مانند Error استفاده می‌شود. نام‌هایی معنا دار اما بدون دقیق شدن در مورد ریز جزئیات پیاده سازی آن‌ها. در این حالت می‌توان [قالب جدیدی را تدارک دید](#) و با ارائه تعاریف جدیدی برای کلاس Error و نحوه نمایش دلخواهی را به آن اعمال نمود. یا برای نمونه نام rightside را برای نمایش ستونی در صفحه، در نظر بگیرید. این نام بسیار ویژه است؛ اما Sematic name آن می‌تواند sidebar باشد تا بدون دقیق شدن در جزئیات پیاده سازی آن، در چپ یا راست صفحه قابل اعمال باشد. Semantic class names کلیدهای هستند جهت استفاده مجدد از قابلیت‌های یک فریم ورک CSS.

ب) Compositional classes

اکثر کلاس‌های Twitter Bootstrap دارای محدوده کاری کوچکی هستند و به سادگی قابل ترکیب با یکدیگر جهت رسیدن به نمایی خاص می‌باشند. برای مثال به سادگی می‌توان به یک table سه ویژگی color، hover و width برگرفته شده از Twitter Bootstrap را انتساب داد و نهایتاً به نتیجه دلخواه رسید؛ بدون اینکه نگران باشیم افزودن کلاس جدیدی در اینجا بر روی سایر کلاس‌های انتساب داده شده، تاثیر منفی دارد.

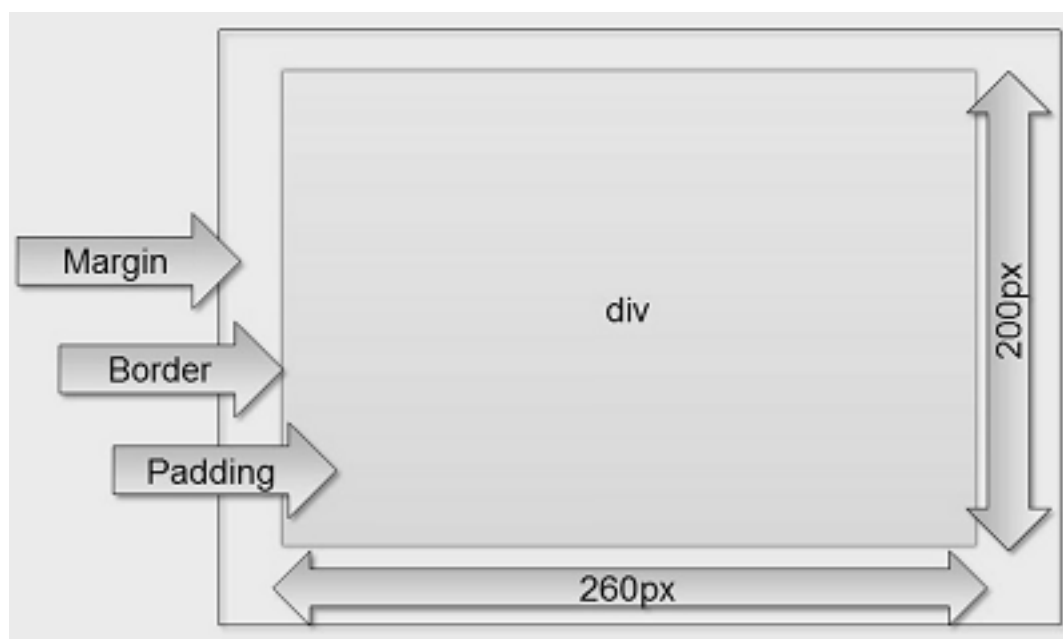
Conventions (ج)

برای استفاده از اکثر قابلیت‌های این فریم ورک CSS یک سری قراردادهای پیش فرضی وجود دارند. برای مثال اگر از کلاس توکار pagination به همراه یک سری ul و li استفاده کنید، به صورت خودکار یک pager شکل ظاهر خواهد شد. یا برای مثال اگر به یک html table کلاس‌های table-hover table-striped table را انتساب دهیم، به صورت خودکار قراردادهای پیش فرض table مجموعه Twitter Bootstrap به آن اعمال شده، به همراه رنگی ساختن یک درمیان زمینه ردیف‌ها و همچنین فعال سازی تغییر رنگ ردیف‌ها با حرکت ماوس از روی آن‌ها.

طرح‌بندی صفحات یک سایت به کمک Twitter Bootstrap

بررسی Grid layouts

Layout به معنای طرح‌بندی و چیدمان محتوا در یک صفحه است. یکی از متداولترین روش‌های طرح‌بندی صفحات چه در حالت چاپی و چه در صفحات وب، چیدمان مبتنی بر جداول و گریدها است. از این جهت که نحوه سیلان و نمایش محتوا از چپ به راست و یا راست به چپ را به سادگی میسر می‌سازند؛ به همراه اعمال حاشیه‌های مناسب جهت قسمت‌های متفاوت محتوای ارائه شده. Grid در طرح‌بندی، نمایش بصری نخواهد داشت اما در ساختار صفحه وجود داشته و مباحثی مانند جهت، موقعیت و یکپارچگی و یکدستی طراحی را سبب می‌شود. به علاوه مرورگرها و مفهوم Grid نیز به خوبی با یکدیگر سازگار هستند. در دنیای HTML و CSS، طراحی‌ها بر اساس مفهوم ساختار مستطیلی اشیاء صورت می‌گیرد:



برای نمونه در اینجا تصویر [CSS Box Model](#) را مشاهده می‌کنید. به این ترتیب، هر المان دارای محدوده‌ای مستطیلی با طول و عرض مشخص، به همراه ویژگی‌هایی مانند Margin، Border و Padding است. در سال‌های اولیه طراحی وب، عموماً کارهای طراحی صفحات به کمک HTML Tables انجام می‌شد. اما با پخته‌تر شدن CSS، استفاده از Tables برای طراحی صفحات کمتر و کمتر گشت تا اینکه نهایتاً فریم ورک‌های CSS ایی پدید آمدند تا طراحی‌های مبتنی بر CSS را با ارائه گریدها، ساده‌تر کنند. مانند 960 GS، Blue print، و ... Twitter Bootstrap که طراحی مبتنی بر گریدهای CSS ایی را به مجموعه قابلیت‌های دیگر خود افزوده است.

بررسی Fixed Grids

در اینجا در صفحه layout برنامه، یک Div دربرگیرنده دو Div دیگر را مشاهده می‌کنید:

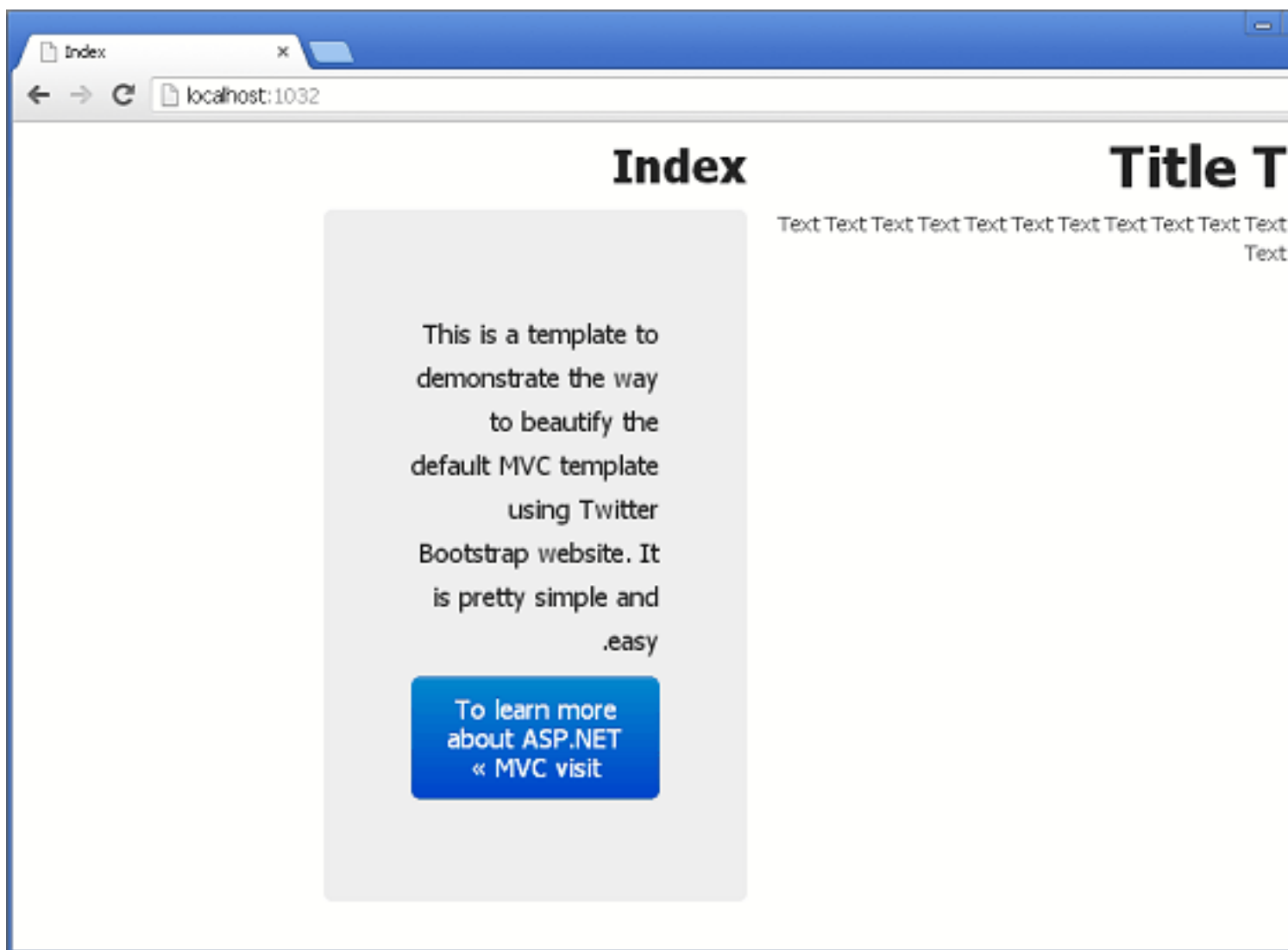
```
<body>
  <div>
    <div>
      <h1>
        Title Title
      </h1>
      Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text
    </div>
    <div>
      @RenderBody()
    </div>
  </div>
</body>
```

اگر در این حالت صفحه را در مرورگر بررسی کنیم، تمام قسمت‌های نمایش داده شده به همین نحو از بالا به پایین صفحه قرار خواهند گرفت. اما قصد داریم این محتوا را، دو ستونی نمایش دهیم. Div به همراه Title در یک طرف صفحه و Div دربرگیرنده محتوای صفحات، در قسمتی دیگر.

برای اینکار در Twitter Bootstrap از کلاسی به نام row استفاده می‌شود که بیانگر یک ردیف است. این کلاس را به خارجی‌ترین Div موجود اعمال خواهیم کرد. در یک صفحه، هر تعداد row ایی را که نیاز باشد، می‌توان تعریف کرد. داخل این ردیف‌ها، امکان تعریف ستون‌های مختلف و حتی تعریف ردیف‌های تو در تو نیز وجود دارد. هر ردیف Twitter Bootstrap از 12 ستون تشکیل می‌شود و برای تعریف آن‌ها از کلاس span استفاده می‌گردد. در این حالت جمع اعداد ذکر شده پس از 12 باید 12 را تشکیل دهند.

```
<body>
  <div class="row">
    <div class="span7">
      <h1>
        Title Title
      </h1>
      Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text
    </div>
    <div class="span5">
      @RenderBody()
    </div>
  </div>
</body>
```

برای نمونه در اینجا کلاس row به خارجی‌ترین Div موجود اعمال شده است. دو Div داخلی هر کدام دارای کلاس‌های span ایی جهت تشکیل ستون‌های داخل این ردیف هستند. جمع اعداد بکارگرفته شده پس از آن‌ها، عدد 12 را تشکیل می‌دهد. به این ترتیب به یک طراحی دو ستونی خواهیم رسید.



در این تصویر، قسمت `RenderBody` کار رندر اکشن متد `Index` کنترلر `Home` برنامه را با `View`ی معادل کدهای ذیل، انجام داده است:

```
@{
    ViewBag.Title = "Index";
}
<h2>
    Index</h2>
<div class="hero-unit">
    <h2>@ViewBag.Message</h2>
    <p>
        This is a template to demonstrate the way to beautify the default MVC template
        using Twitter Bootstrap website. It is pretty simple and easy.</p>
    <p>
        <a href="http://asp.net/mvc" class="btn btn-primary btn-large" style="color: White;">
            To learn more about ASP.NET MVC visit &raquo;</a></p>
    </div>
```

درک نحوه عملکرد `Grid` در `Twitter Bootstrap`

در مثال ذیل 5 ردیف را مشاهده می‌کنید:

```
<div class="row">
```

```

<div class="span1">1</div>
<div class="span1">1</div>
<div class="span1">1</div>
<div class="span1">1</div>
<div class="span1">1</div>
<div class="span1">1</div>
<div class="span1">1</div>
<div class="span1">1</div>
<div class="span1">1</div>
<div class="span1">1</div>
<div class="span1">1</div>
</div>
<div class="row">
  <div class="span3">3</div>
  <div class="span4">4</div>
  <div class="span5">5</div>
</div>
<div class="row">
  <div class="span5">5</div>
  <div class="span7">7</div>
</div>
<div class="row">
  <div class="span3">3</div>
  <div class="span7 offset2">7 offset 2</div>
</div>
<div class="row">
  <div class="span12">12</div>
</div>

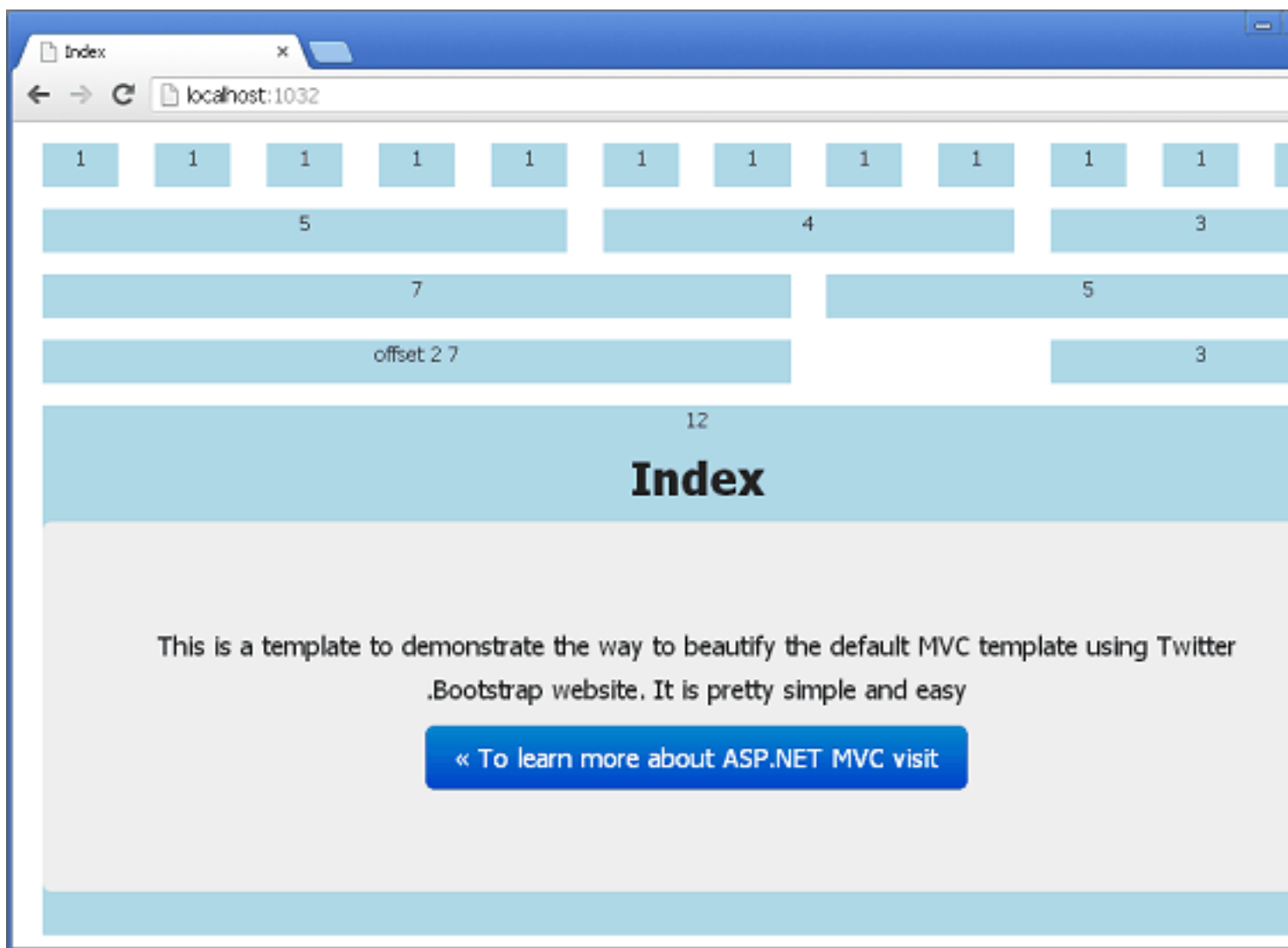
```

در هر ردیف، امکان استفاده از حداکثر 12 ستون وجود دارد که یا می‌توان مانند ردیف اول، 12 ستون را ایجاد کرد و یا مانند ردیف دوم، سه ستون با عرض‌های متفاوت (و یا هر ترکیب دیگری که به جمع 12 برسد). در ستون چهارم، از کلاس offset نیز استفاده شده است. این مورد سبب می‌شود ستون جاری به تعدادی که مشخص شده است به سمت چپ (با توجه به استفاده از حالت RTL در اینجا) رانده شود و سپس ترسیم گردد. یا اینکه می‌توان مانند ردیف آخر، یک ستون را به عرض 12 که در حقیقت 940 پیکسل است، ترسیم نمود. برای اینکه بتوانیم این گرید تشکیل شده و همچنین ستون‌ها را بهتر مشاهده کنیم، به فایل style.css سایت، تنظیم زیر را اضافه کنید:

```

[class*="span"]
{
background-color: lightblue;
text-align: center;
margin-top: 15px;
}

```



نکته جالب این گرید، Responsive یا واکنشگرا بودن آن است. در این حالت، عرض مرورگر را کم و زیاد کنید. خواهید دید که ستون‌ها در صورتیکه در عرض نمایشی جاری، قابل ارائه نباشند، به ردیف‌های بعدی منتقل خواهند شد. البته باید دقت داشت که این گرید هیچگاه یک ستون را نخواهد شکست. برای نمونه ردیف آخر، همواره با همان عرض ثابتش نمایش داده می‌شود و با کوچکتر کردن اندازه مرورگر، یک اسکرول افقی برای نمایش محتوای آن ظاهر خواهد شد.

یک نکته

اگر نمی‌خواهید که چنین رفتار واکنش‌گرایی بروز کند، نیاز است کلیه ردیف‌ها را در div ایی با کلاسی به نام container محصور کنید.

به این ترتیب ابتدا گرید نمایش داده شده، در میانه صفحه ظاهر خواهد شد (پیشتر از سمت راست شروع شده بود). همچنین دیگر با کوچک و بزرگ شدن اندازه مرورگر، ستون‌ها به شکل یک پشته بر روی هم قرار نخواهند گرفت. (اگر پس از این تنظیم، چنین قابلیتی را مشاهده نکردید و هنوز هم طراحی، واکنشگرا بود، تعریف bootstrap-responsive.css را نیاز است برای آزمایش، از هدر صفحه حذف کنید)

بررسی Fluid Grids

به گرید قسمت قبل از این جهت Fixed Grid گفته می‌شود که عرض هر span آن با یک عدد مشخص تعیین گشته است. اما در

حالت Fluid Grid، عرض هر span برحسب درصد تعیین می‌شود. بکارگیری درصد در اینجا به معنای امکان تغییر عرض یک ستون بر اساس عرض جاری Container آن است. در اینجا span12 دارای عرض 100 درصد خواهد بود. در مثال قبل، برای استفاده از Fluid grids، تنها کافی است هرجایی کلاسی مساوی row وجود دارد، به row-fluid تغییر کند. همچنین کلاس container را به container-fluid تغییر دهید. برای آزمایش آن، اندازه و عرض نمایشی مرورگر خود را تغییر دهید. اینبار مشاهده خواهید کرد که برخلاف حالت Fixed Grid، عرض ستون‌ها به صورت خودکار کم و زیاد می‌شوند. این مورد بر روی محتوای قرار گرفته در این ستون‌ها نیز تاثیر گذار است. برای مثال اگر یک تصویر را در حالت Fluid grid در ستونی قرار دهید، با تغییر عرض مرورگر، اندازه این تصویر نیز تغییر خواهند کرد؛ اما در حالت Fixed Grid خیر. حالت Fluid، شیوه متداول استفاده از bootstrap در اکثر سایت‌های مهمی است که تابحال از این فریم ورک CSS استفاده کرده‌اند.



مروری بر طراحی واکنشگرا یا Responsive

این روزها تعدادی از کاربران، با استفاده از ابزارهای موبایل و تبلت‌ها از وب سایت‌ها بازدید می‌کنند. هر کدام از این‌ها نیز دارای اندازه نمایشی متفاوتی می‌باشند. بنابراین نیاز خواهد بود تا حالت بهینه‌ای را جهت اینگونه وسایل نیز طراحی نمود. حالت بهینه در اینجا به معنای قابل خواندن بودن متون، امکان استفاده از لینک‌های ورود به صفحات مختلف و همچنین حذف اسکروول و مباحث زوم جهت مشاهده صفحات است.

یکی از ویژگی‌های CSS به نام media چنین قابلیتی را فراهم می‌سازد. برای نمونه قسمتی از فایل bootstrap-responsive.css دارای چنین تعاریفی است:

```
@media (min-width: 768px) and (max-width: 979px) {
    .hidden-desktop {
        display: inherit !important;
    }
}
```

توسط این تنظیمات، شیوه نامه تعیین شده، تنها به صفحاتی با اندازه‌هایی بین 768 تا 979 پیکسل اعمال می‌شود (تعدادی از تبلت‌ها برای نمونه).

Bootstrap برای مدیریت اندازه‌های مختلف وسایل موبایل، شیوه‌نامه‌های خاصی را تدارک دیده است که از اندازه 480px و یا کمتر، تا 1200px و یا بیشتر را پوشش می‌دهد.

به این ترتیب با اندازه 940px که پیشتر در مورد آن بحث شد، اندازه span12 به صورت خودکار به اندازه‌های متناسب با صفحات نمایشی کوچکتر تنظیم می‌گردد. همچنین برای اندازه‌های صفحات کوچکتر از 768px به صورت خودکار از Fluid columns استفاده می‌گردد.

تنها کاری که برای اعمال این قابلیت باید صورت گیرد، افزودن تعاریف فایل bootstrap-responsive.css به هدر صفحه است که در قسمت قبل انجام گردید. این فایل باید پس از فایل اصلی bootstrap.css اضافه شود.

کلاس‌های کمکی طراحی واکنشگرا

Bootstrap شامل تعدادی کلاس کمکی در فایل bootstrap-responsive.css خود می‌باشد شامل visible-phone، visible-tablet و visible-desktop به علاوه hidden-phone، hidden-tablet و hidden-desktop. به این ترتیب می‌توان محتوای خاصی را جهت وسایل ویژه‌ای نمایان یا مخفی ساخت.

برای مثال محتوای مشخص شده با کلاس hidden-desktop، در اندازه وسایل دسکتاپ نمایش داده نخواهد شد.

برای آزمایش آن، شش div را با کلاس‌های یاد شده و محتوایی دلخواه ایجاد کرده و سپس اندازه عرض مرورگر را تغییر دهید تا بهتر بتوان مخفی یا نمایان ساختن محتوا را بر اساس اندازه صفحه جاری درک کرد.

یکی از کاربردهای این قابلیت، قرار دادن تبلیغاتی با اندازه‌های تصاویری مشخص برای وسایل مختلف است؛ بجای اینکه منتظر شویم تا Fluid layout اندازه تصاویر را به صورت خودکار کوچک یا بزرگ کند، که الزاماً بهترین کیفیت را حاصل نخواهد ساخت.

```
<div class="container-fluid">
  <div class="row-fluid">
    <div class="span4">
      <div class="visible-phone">
        visible-phone</div>
      </div>
    <div class="span4">
      <div class="visible-tablet">
        visible-tablet</div>
      </div>
    <div class="span4">
      <div class="visible-desktop">
        visible-desktop</div>
      </div>
    </div>
  </div>
</div>
<div class="container-fluid">
  <div class="row-fluid">
    <div class="span4">
      <div class="hidden-phone">
        hidden-phone</div>
      </div>
    </div>
  </div>
</div>
```

```
<div class="span4">
  <div class="hidden-tablet">
    hidden-tablet</div>
  </div>
  <div class="span4">
    <div class="hidden-desktop">
      hidden-desktop</div>
    </div>
  </div>
</div>
```



نظرات خوانندگان

نویسنده: احمدعلی شفیعی
تاریخ: ۱۰:۸ ۱۳۹۲/۰۳/۱۹

محدودیت ۱۲ تایی ستون‌ها به اندازه‌ی صفحه بستگی دارد. برای من ۱۳ تاست.

نویسنده: وحید نصیری
تاریخ: ۱۷:۳۵ ۱۳۹۲/۰۳/۱۹

شعار این کتابخانه چنین چیزی هست: « [Bootstrap is built on responsive 12-column grids](#) ». یعنی حالت پیش فرض طراحی آن 12 ستونی است. اگر به فایل css آن مراجعه کنید، یک چنین عرض‌هایی قابل مشاهده است:

```
.span12 {
  width: 940px;
}
.span1 {
  width: 60px;
}
```

البته این‌ها برای حالت fixed است و نه fluid که بر اساس درصد تعریف شده

```
.row-fluid .span12 {
  width: 100%;
  *width: 99.94680851063829%;
}
```

container هم با عرض 940 پیکسل تعریف شده. همچنین یک GutterWidth را هم باید در بین ستون‌ها در نظر داشت. حالت پیش فرض:

```
@gridColumns: 12;
@gridColumnWidth: 60px;
@gridGutterWidth: 20px;
```

البته میشه مثلا عرض رو بیشتر از 940 تعریف کرد. برای اینکار باید css رو ویرایش و از ابزارهایی مانند [less استفاده کرد](#) و یا امکان [سفارشی سازی اون](#) هم هست.

نویسنده: وحید نصیری
تاریخ: ۱۲:۴۸ ۱۳۹۲/۰۳/۲۰

یک نکته تکمیلی

در مطلب فوق یک چنین تعریفی را مشاهده نمودید:

```
@BundleConfig.AddStyles("~/Content/css",
    "~/Content/bootstrap.css",
    "~/Content/bootstrap-responsive.css",
    "~/Content/Site.css"
)
```

در پوشه content، فایل‌های bootstrap.min.css و bootstrap.css وجود دارند (اگر بسته نیوگت آن‌را نصب کنید). در این بین، اگر فایل bootstrap.css را اندکی ویرایش کنید (مثلا نام فونت‌ها یا اندازه آن‌ها را تغییر دهید)، متوجه خواهید شد که تغییرات شما اصلا اعمال نمی‌شوند. علت چیست؟! سیستم Bundling به صورت هوشمند در صورت وجود فایل‌های css.min، از آن‌ها استفاده خواهد کرد (حتی در حالت دیباگ) و نه

از فایل bootstrap.css ذکر شده در تنظیمات. به همین جهت ممکن است مدتی سر درگم شوید که چرا هر تغییری را که اعمال می‌کنم، تاثیری ندارد!
بنابراین یا فایل‌های css.min را حذف کنید و یا مکان آن‌ها را تغییر دهید تا در کنار فایل‌های غیرفشرده شده پیش فرض نباشند.

نویسنده: یزدان
تاریخ: ۱۳۹۲/۰۳/۳۱ ۱:۳۷

آیا برای web form هم کارایی داره و میشه در web form از اون استفاده کرد ؟

```
@BundleConfig.AddStyles("~/Content/css",
    "~/Content/bootstrap.css",
    "~/Content/bootstrap-responsive.css",
    "~/Content/Site.css"
)
```

در وب فرم به چه صورت این رو اضافه کنیم ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۳۱ ۹:۰۰

در ASP.NET 4.5 مباحث Bundling and Minification [جزو استاندارد آن](#) است. در ASP.NET 4.0 می‌شود از امکانات ASP.NET MVC 4 هم استفاده کرد (همان Bundling and Minification کامپایل شده برای دات نت 4 است).

نویسنده: ایلینا اکبری فرد
تاریخ: ۱۳۹۲/۰۴/۰۱ ۱۱:۵۰

با سلام.

در قسمت Bundling وقتی از کد شما استفاده میکنم تنها فقط Site.css را باندل میکند و سایر css ها باندل نمی‌شوند. وقتی کد زیر را کامنت کردم مشکل حل شد.

```
@BundleConfig.AddStyles("~/Content/css",
    "~/Content/bootstrap.css",
    "~/Content/Site.css")
...
var existing = BundleTable.Bundles.GetBundleFor(virtualPath);
//if (existing != null)
//    return;
```

کامنت کردن این کد مشکلی ایجاد میکند؟

نویسنده: میثم
تاریخ: ۱۳۹۲/۰۴/۰۹ ۱۱:۴۰

فایل bootstrap.css بصورت ناقص فشرده می‌شود.
بطور دقیق‌تر تا رسیدن به کاراکترهای :: بدون مشکل کار پیش می‌رود.
در حالت اجرا وقتی فایل را css را باز میکنیم فقط کلاس‌ها و شناسه‌های قبل از :: وجود دارند.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۰۹ ۱۱:۵۵

در پروژه MVC خود، خط فرمان پاورشل نیوگت را باز کنید و دستور Update-Package را صادر کنید. تقریباً تمام اجزای MVC4 مرتباً به روز می‌شوند؛ منجمله همین بحث Web.Optimization آن:

PM> Update-Package

نویسنده: سیروان عفیفی
تاریخ: ۱۹:۲۲ ۱۳۹۲/۰۵/۱۸

دو مدخل برای من اضافه میشن

```
<link href="/Content/myCSS?v=LBa4VrARIJBaPHOXLQoCWVPKu_c6QBb4D7npv-dq5TA1" rel="stylesheet"/>
<script src="/Scripts/js?v=rk-13czPS8KW6usE0pK8aIglVqPOVEKWoSXV4N-5N2Q1"></script>
```

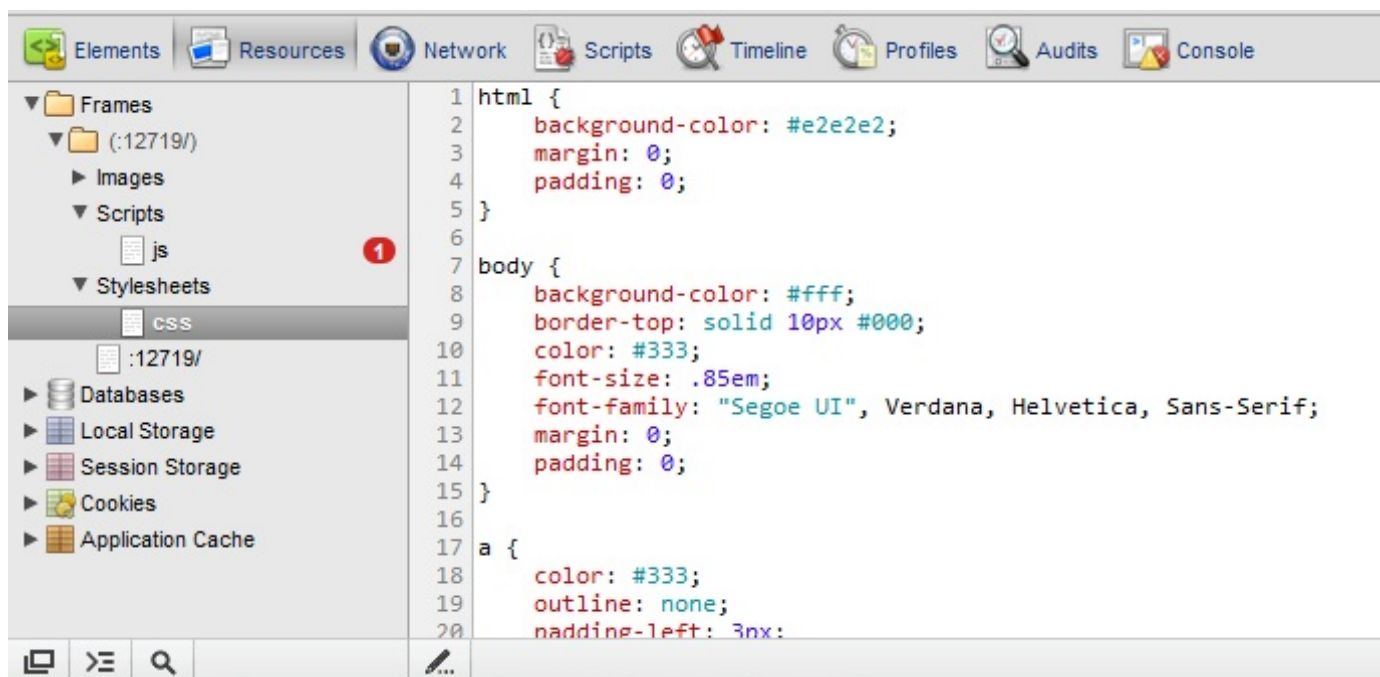
منتها برای CSS فقط فایل Style.css و برای جاوااسکریپت فقط jQuery رو اضافه میکنه، به نظرتون مشکل چی میتونه باشه؟

نویسنده: وحید نصیری
تاریخ: ۲۱:۷ ۱۳۹۲/۰۵/۱۸

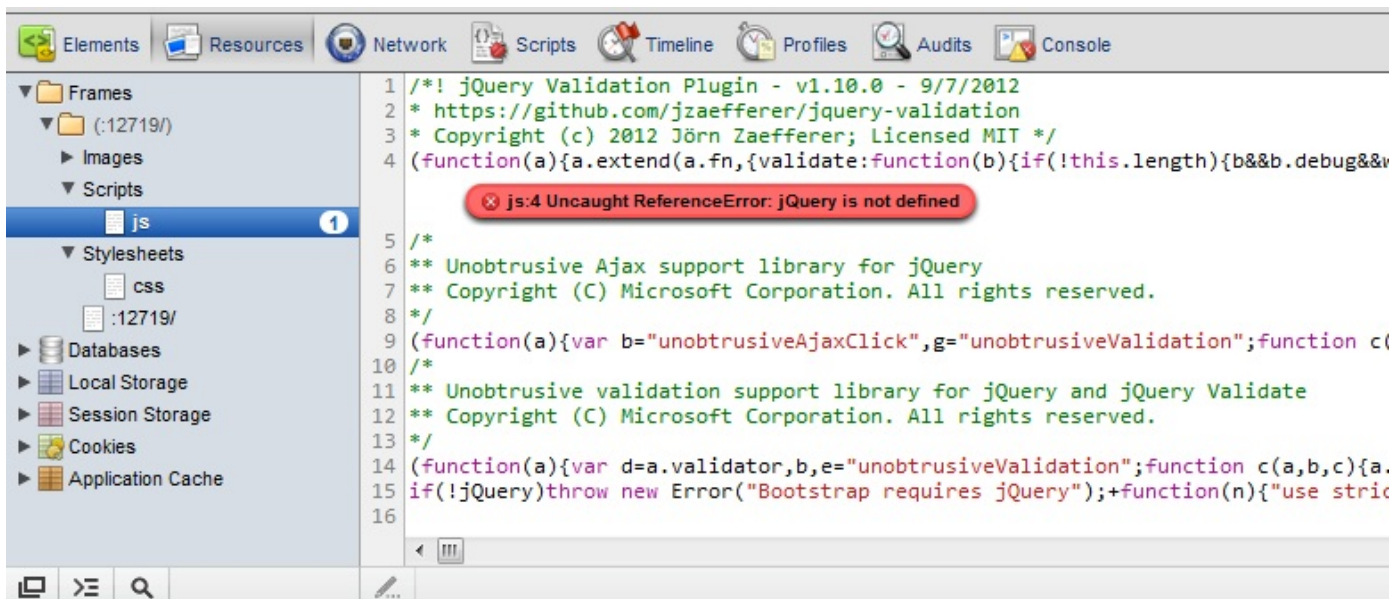
اگر واقعا مشکلی وجود داشته باشد، در همین لینک‌های پویایی که ارائه دادید، با کلیک بر روی آن‌ها عبارت «Minification failed. Returning unminified contents» ظاهر خواهد شد ([یک نمونه مثال در اینجا](#)) و سپس در سطرهای بعدی علت خطای کامپایل ذکر می‌شود که باید برطرف شود.

نویسنده: سیروان عفیفی
تاریخ: ۰:۴۱ ۱۳۹۲/۰۵/۱۹

حاصل کلیک بر روی لینک‌های پویا تولید شده به صورت زیر است :
مثلا با کلیک بر روی css تولید شده فایل Style.css را مشاهده میکنم :



و با کلیک بر روی فایل js تولید شده نیز به صورت زیر می‌باشد :



نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۵/۱۹ ۰۵:۵۸

چند سؤال اینجا هست که از راه دور قابل بررسی نیست:

- چطور تعاریف فایل‌ها رو در کدها اضافه کردید؟ (اگر خطای نبود jQuery رو بدون خطای کامپایل که ذکر شد گرفتید، یعنی اصلا jQuery تعریف نشده در تعریف bundle نهایی)
- ساختار پوشه‌های شما چیست؟ محل قرارگیری این فایل‌ها به چه ترتیبی است؟
- از چه نگارشی از اسکریپت‌ها و فایل‌ها استفاده می‌کنید؟ از کجا تهیه شده‌اند؟

ضمناً هیچ الزامی برای استفاده از minification در این مبحث نیست. فقط برای تکمیل بحث عنوان شده. به علاوه در قسمت‌های بعدی این بحث، سورس هر قسمت [پیوست شده](#) که شامل فایل‌های اسکریپت و CSS متناظر با هر بحث می‌شود. این فایل‌ها در این سری ویژه، مورد تأیید هستند و آزمایش شده‌اند.

نویسنده: فرزاد
تاریخ: ۱۳۹۲/۰۶/۱۸ ۱۱:۱۲

سلام؛ بعد از اضافه کردن کلاس AsIsBundleOrderer به Bundleconfig با خطای:

```
'MvcApplication7.Helper.AsIsBundleOrderer' does not implement interface member
'System.Web.Optimization.IBundleOrderer.OrderFiles(System.Web.Optimization.BundleContext,
System.Collections.Generic.IEnumerable<System.Web.Optimization.BundleFile>')
```

مواجهه میشم. در واقع از کلاس AsIsBundleOrderer ارور میگیره.
ممنون میشم راهنمایی کنید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۶/۱۸ ۱۱:۲۷

- در نگارش جدید System.Web.Optimization، امضای FileInfo متد OrderFiles تبدیل شده به BundleFile. (همین تغییر نام کافی است)
- در زمان نگارش این مطلب، FileInfo بوده.

نویسنده: احسان شیروان
تاریخ: ۱۷:۳۲ ۱۳۹۲/۰۷/۰۴

با سلام ضمن تشکر از شما بابت آموزش‌های بسیار خوب
یه سوال داشتم من از C#.net , VS2010 دارم استفاده می‌کنم برای استفاده کردن از Bootstrap تمامی مراحل یاد شده توسط
شما رو مو به مو اجرا کردم اما متأسفانه به هیچ وجه کار نداد البته خطا هم نمیده اما مباحث مربوط به ریسپانسیو بودن صفحه و
حتی نمایش spanها جوابی نگرفتم آیا این بدلیل ورژن VS2010 هست یا دات نت؟ یا یه چیز دیگه ؟
ظاهراً سایتی که ایجاد کردم هیچ درکی از span نداره در حالی که تمام فایل‌های bootstrap رو طبق آموزش روی سایت نصب
کردم
ممنون میشم راهنماییم نمایید

نویسنده: وحید نصیری
تاریخ: ۱۷:۵۷ ۱۳۹۲/۰۷/۰۴

- مطالبی که من در مورد بوت استرپ منتشر کردم مربوط به نگارش 2 آن است. نگارش 3 آن (که مدتی است منتشر شده و از
طریق لینک‌های ابتدای بحث به صورت پیش فرض دریافت می‌شود) دگرگون شده و خیلی از مباحث آن تغییر کرده‌اند. فایل‌های
بوت استرپ 2 در قسمت‌های بعدی این بحث، [پیوست شده‌اند](#) که شامل فایل‌های اسکریپت و CSS متناظر با هر بحث می‌شود .
+ [راهنمای مهاجرت از بوت استرپ 2 به 3](#)

نویسنده: صابر فتح الهی
تاریخ: ۱۶:۴۴ ۱۳۹۲/۰۹/۱۰

سلام؛ من توی ASP.NET Webform از قابلیت Bundle استفاده کردم (قبلاً درست کار می‌کرد) اما توی نسخه 4.5 کار نمی‌کنه، بعد از
دیباگ فهمیدم اصلاً فایلها با BundleCollection اضافه نمیشه و زمانی که می‌خواد خروجی بگیره با خطای نبود مدخل مورد نظر
مواجه میشه. Update-Package را هم اجرا کردم اما همون خطا رو دریافت می‌کنم.

نویسنده: وحید نصیری
تاریخ: ۱۸:۴۳ ۱۳۹۲/۰۹/۱۰

جزئیات روش خودتان را با روش یکی از اعضای تیم دات نت مطابقت بدید (مربوط به ASP.NET 4.5 است):
[Adding Bundling and Minification to Web Forms](#)

نویسنده: افشار محبی
تاریخ: ۱۷:۳۷ ۱۳۹۲/۰۹/۲۰

باید span را به col-md عوض کنید. همان طور که در راهنمای مهاجرت از ۲ به ۳ که لینکش را آقای نصیری مشخص کرده اند
توضیح داده شده:

replace .span* with .col-md-*

بنده هم دقایقی را معطل کشف این تغییر بوت‌استرپ شدم.

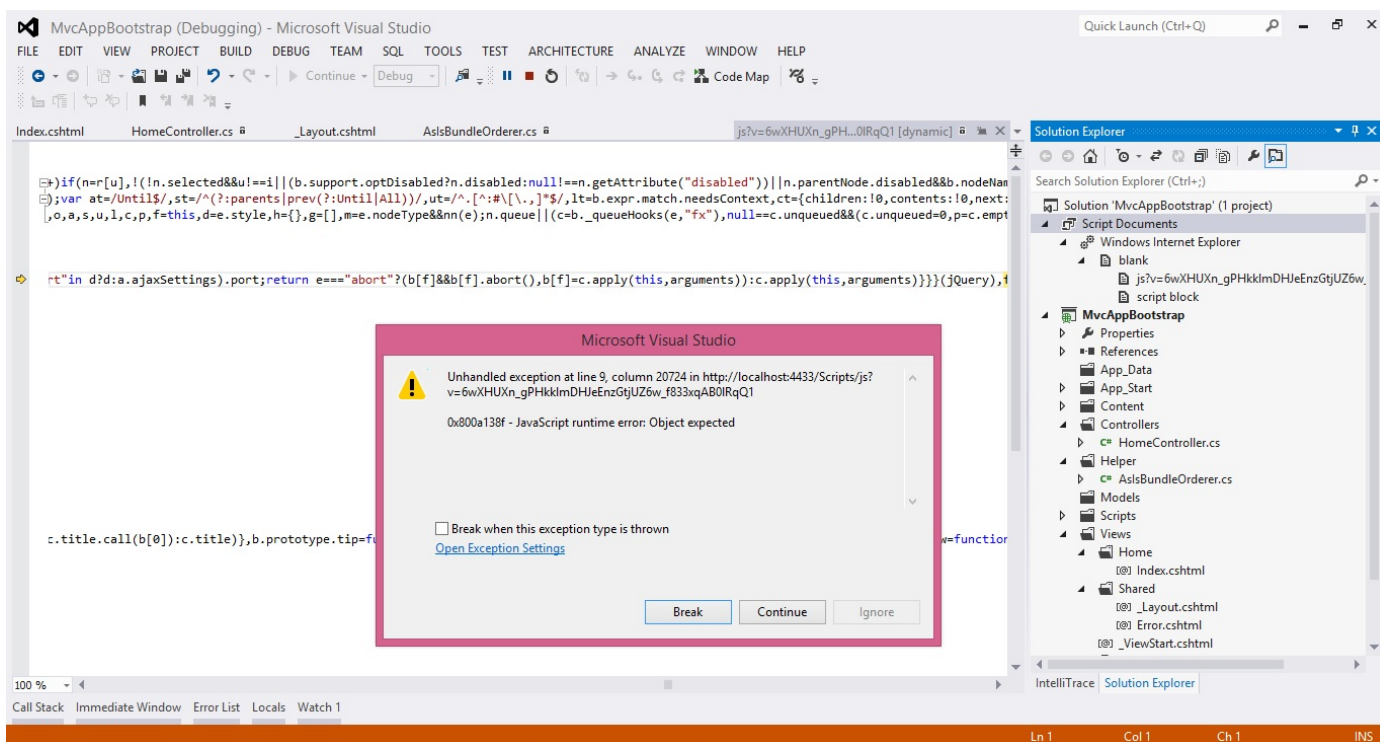
نویسنده: وحید نصیری
تاریخ: ۱۷:۴۵ ۱۳۹۲/۰۹/۲۰

توضیحات تکمیلی در اینجا: « [بررسی سیستم جدید گرید بوت استرپ 3](#) »

نویسنده: مجتبی فخاری
تاریخ: ۲۳:۰۵ ۱۳۹۲/۱۲/۰۳

بعد از نصب از طریق دستور Install-Package Twitter.BootstrapRTL و بعد هم دستور Install-Package PM>

Microsoft.AspNet.Web.Optimization بسته‌ها رو نصب کردم و یک پوشه به اسم Helper نیز اضافه نمودم و کلاسهای AsIsBundleOrderer را اضافه نمودم و در layout هم کدهای شما را آوردم ولی حالا در زمان اجرا خطای زیر را میدهد؟



میشه راهنمایی بفرمائید؟ با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۰۳ ۲۳:۱۳

مطلب جاری و سری مرتبط با آن بوت استرپ 2 نوشته شد. امروز اگر دستور `install-Package Twitter.BootstrapRTL` را صادر کنید، بوت استرپ 3.1 را نصب می‌کند که سیستم آن در بسیاری از موارد 180 درجه دگرگون شده نسبت به نگارش 2. بنابراین برای کار با نگارش 2 بهتر است دستور زیر را صادر کنید:

```
PM> Install-Package Twitter.BootstrapRTL -Version 2.3.2.1
```

و یا از مطالب [ارتقاء به بوت استرپ 3](#) کمک بگیرید.
ضمناً فایل‌های مرتبط با نگارش 2، در قسمت‌های بعدی همین سری، در نظرات هر قسمت [پیوست شده‌اند](#).

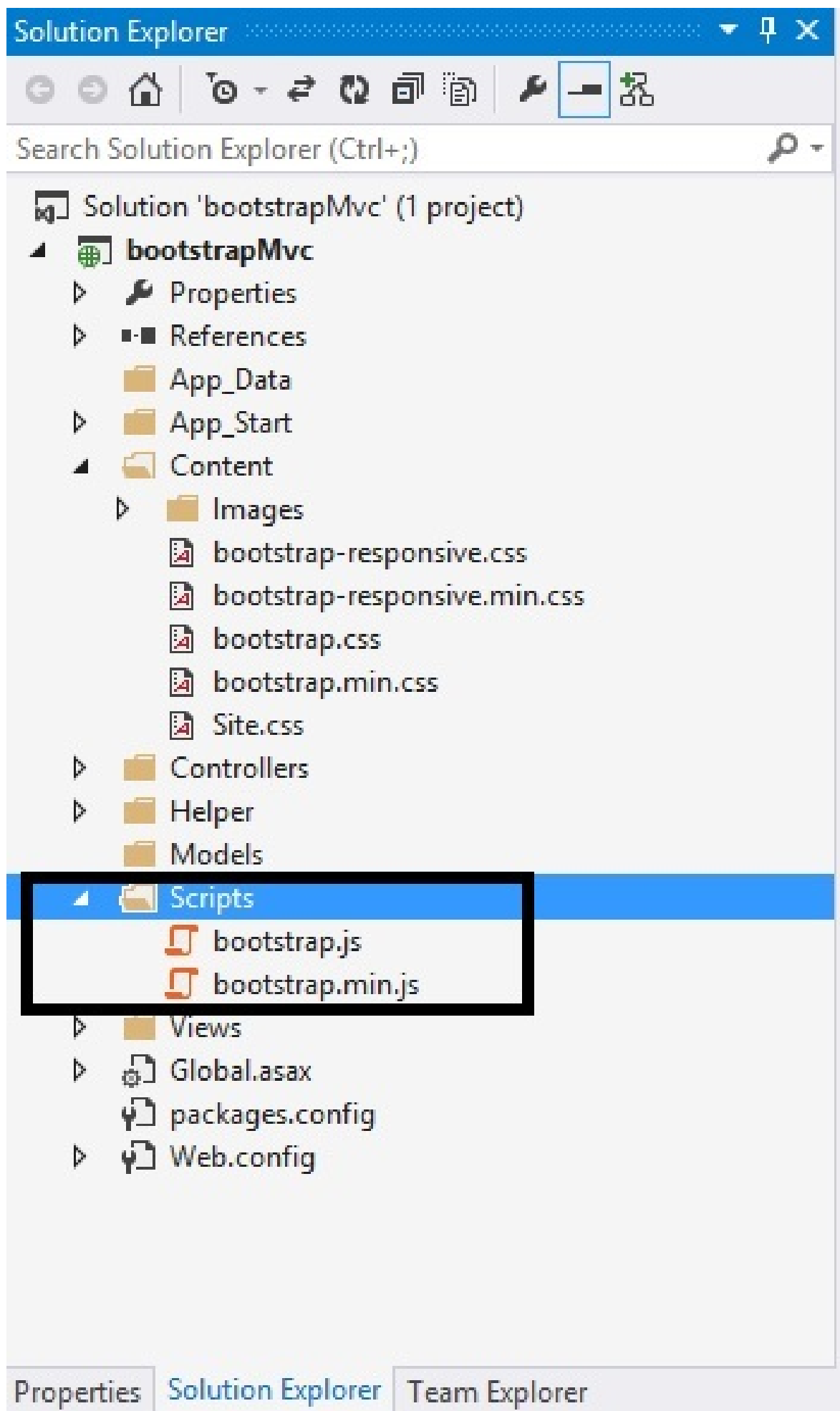
نویسنده: ایمان باقری
تاریخ: ۱۳۹۳/۰۱/۱۹ ۱۳:۲۵

سلام. من توی پروژم تو قسمت اسکریپت‌ها فقط `bootstrap.js` و `bootstrap.min.js` رو دارم و مابقی جاوا اسکریپت‌ها و جی کوئری هارو ندارم. با چه دستوری میتونم اونارو اضافه کنم. چون الان منوهای پایین افتادنی و یکسری چیزها دیگه کار نمیکنه. در ضمن پروژه خالی mvc شروع کردم. (تو حالت basic خودش پیش فرض داره)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۱/۱۹ ۱۳:۲۸

[کمی بالاتر](#) پاسخ دادم «مطلب جاری و سری مرتبط با آن بوت استرپ 2 نوشته شد ...». (چندین بار)

من هم بوت استرپ 2 رو گرفتم فقط فایل‌های جی کوئری و validation اون رو ندارم
فایل‌های jquery-1.9.1.min.js jquery.unobtrusive-ajax.min.js jquery.validate.min.js و
jquery.validate.unobtrusive.min.js



نویسنده:

وحید نصیری

تاریخ:

۱۳:۴۱ ۱۳۹۳/۰۱/۱۹

عنوان کردید حالت Basic پروژه‌های MVC 5 را شروع کردید. پیش فرض استاندارد MVC 5 [بوت استرپ 3](#) است و نه دو. فایل‌های کامل [بوت استرپ 2](#) را از طریق دستور ذیل که پیشتر نیز ذکر شده بود دریافت کنید (تمام اجزای js آن با هم یکی شده‌اند در قالب یک فایل نهایی):

```
PM> Install-Package Twitter.BootstrapRTL -Version 2.3.2.1
```

نویسنده:

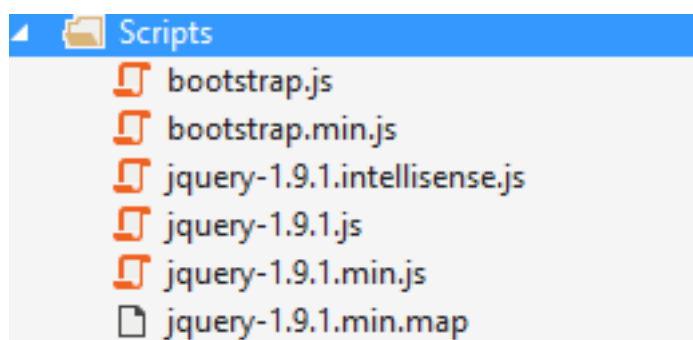
ایمان باقری

تاریخ:

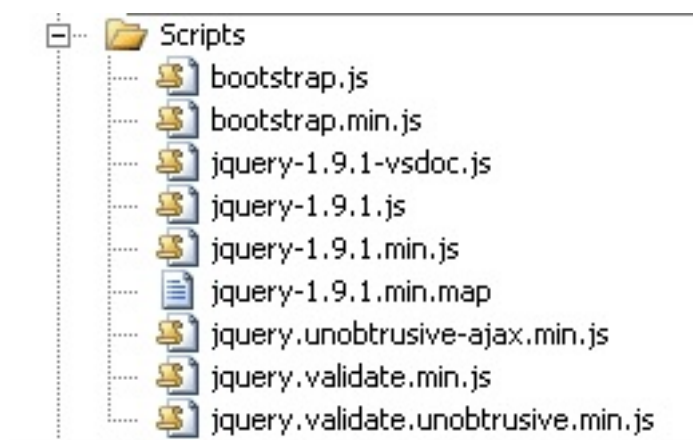
۱۴:۳ ۱۳۹۳/۰۱/۱۹

من حالت empty پروژه 4 mvc رو شروع کردم (بالا هم عرض کردم) و بوت استرپ هم از سایت rbootstrap.ir نسخه 2.3.2 v رو گرفتم.

البته الان که از دستور `PM> Install-Package Twitter.BootstrapRTL -Version 2.3.2.1` استفاده کردم jquery-1.9.1.min.js رو هم نصب کرد



و مشکل منوها حل شد. فقط فایل‌های validation و ajax رو همچنان ندارم



نویسنده:

وحید نصیری

تاریخ:

۱۴:۱۲ ۱۳۹۳/۰۱/۱۹

فایل‌های دیگری را که یاد کردید، جزو بسته‌ی بوت استرپ نیستند. آن‌ها را دستی از یک پروژه‌ی MVC دیگر کپی کنید. از نیوگت هم قابل دریافت هستند: ([^](#) و [^](#))

نویسنده: M.Shakeri

تاریخ: ۱۴:۲۱ ۱۳۹۳/۰۶/۲۱

شما اشاره کردید به این که اگر پروژه رو در حالت release قرار بدیم فشرده سازی اعمال میشه یعنی اینکه در فایل web.config در قسمت compilation خصوصیت debug رو حذف کنیم:

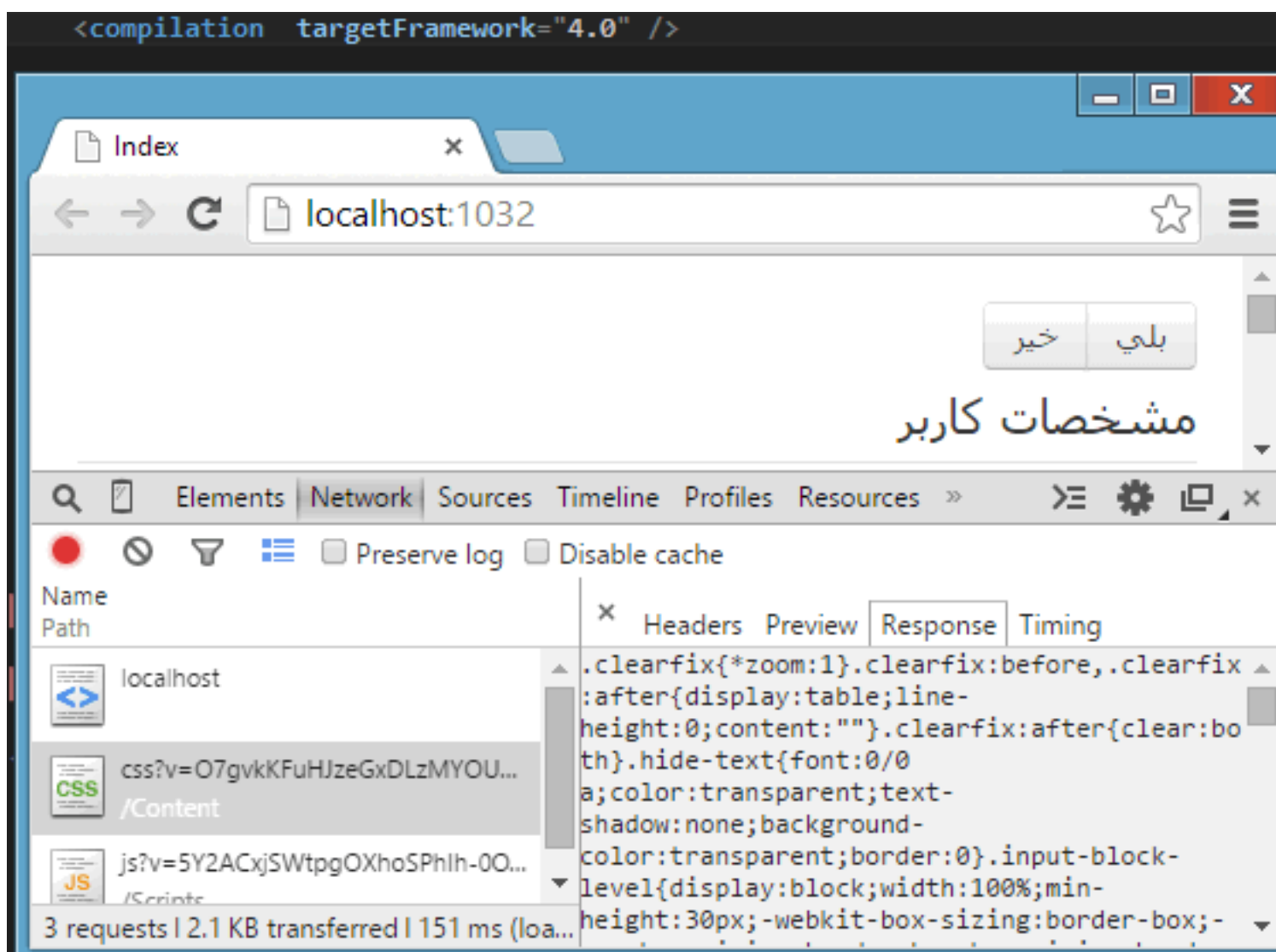
```
<compilation debug="true" targetFramework="4.5" />
```

این کارو انجام دادم اما وقتی سورس صفحه رو نگاه می‌کنم فشرده سازی اعمال نشده به فایل‌های css و js !

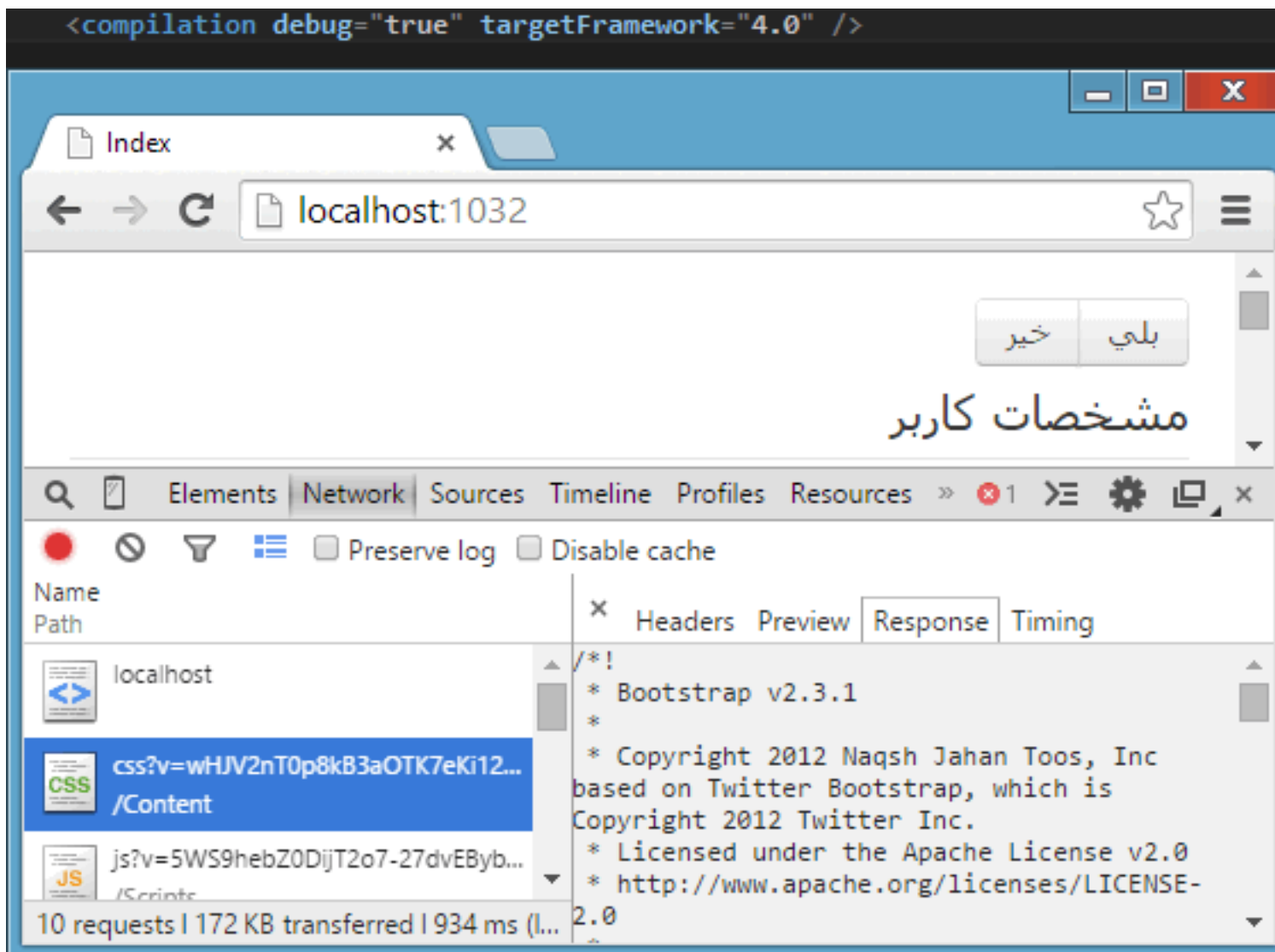
نویسنده: وحید نصیری

تاریخ: ۱۴:۴۲ ۱۳۹۳/۰۶/۲۱

پس از بررسی مجدد، مشکلی مشاهده نشد:
- رندر حالت فشرده شده در حالت غیر دیباگ:



- رندر حالت معمولی و اصل آن در حالت دیباگ:



در کل اگر این روش برای شما کار نمی‌کند، شرط `HttpContext.Current.IsDebuggingEnabled` را از متد `addBundle` حذف کنید و `else` آنرا نگه دارید.

نویسنده: امیرتنها
تاریخ: ۱۳۹۳/۱۰/۰۵ ۲۳:۱۷

با سلام
هنگام اجرا این خطا رو می‌ده .
'does not contain a definition for 'AddStyles

Line 19: @BundleConfig.AddStyles("~/Content/css",

Line 20: "~/Content/bootstrap.css",

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۰۵ ۲۳:۴۵

یکبار نظرات این بحث را کامل مطالعه کنید. این مطلب برای بوت استرپ 2 نوشته شد، در آن زمان. نگارش 3 آن [در اینجا](#) بحث شده. همچنین این کلاس‌ها [تغییراتی داشته‌اند](#) که پیشتر در نظرات بحث شده.

پس از آشنایی مقدماتی با Twitter Bootstrap، در این قسمت قصد داریم تا با ویژگی‌هایی از آن آشنا شویم که در کارهای روزمره طراحی وب بسیار مورد استفاده هستند؛ مانند تایپوگرافی، جداول، فرم‌ها، دکمه‌ها، تصاویر و آیکون‌ها.

تایپوگرافی

هدف از تایپوگرافی، چیدمان متن به نحوی است که واضح، خوانا و مشخص باشد؛ همچنین مباحث زیبایی ارائه را نیز به آن اضافه کنید. برای مثال تنظیم فاصله بین حروف و کلمات، فاصله بین خطوط و یا رعایت یک سری نسبت‌های ویژه مانند [نسبت طلایی](#) جهت دعوت خواننده به مطالعه مطالب، بجای فراری دادن او، در مباحث تایپوگرافی رعایت می‌شوند. خوشبختانه Twitter Bootstrap به همراه یک سری تنظیمات تایپوگرافی پیش فرض است که در ادامه آن‌ها را مرور خواهیم کرد. پیش فرض‌های ابتدایی آن‌را مانند قلم با اندازه 14px، قلم پیش فرض Helvetica، فاصله بین خطوط و رنگ متن را در فایل bootstrap.css می‌توانید مشاهده کنید:

```
body {
margin: 0;"Helvetica Neue", Helvetica, Arial, sans-serif;
color: #333333;
background-color: #ffffff;
}
```

در اینجا اثر تنظیمات bootstrap را بر روی تگ‌های h1 تا h6 ملاحظه می‌کنید:

سرتیتر 1
سرتیتر 2
سرتیتر 3
سرتیتر 4
سرتیتر 5
سرتیتر 6

```
<div class="row-fluid">
  <div class="span12">
    <h1>
      سرتیتر 1
    </h1>
    <h2>
      سرتیتر 2
    </h2>
    <h3>
      سرتیتر 3
    </h3>
    <h4>
      سرتیتر 4
    </h4>
    <h5>
      سرتیتر 5
    </h5>
    <h6>
      سرتیتر 6
    </h6>
  </div>
</div>
```

همچنین اگر نیاز باشد تا نسبت به متنی جلب توجه خواننده را جلب کرد می‌توان از کلاس lead استفاده نمود. به علاوه bootstrap پیش فرض‌هایی را به المان‌های small، strong و em نیز اعمال می‌کند:

```
<div class="row-fluid">
  <div class="span12">
    <p class="lead">
      تیتر
    </p>
```

```
<p>
متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن
متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن
متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن
متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن
</p>
<small>اندازه کوچک</small> <strong>متن ضخیم</strong> <em>متن ایتالیک</em>
</div>
</div>
```

تیترا

متن
متن
متن
متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن

اندازه کوچک متن ضخیم متن ایتالیک

روش دیگر جلب توجه به یک متن در bootstrap، استفاده از کلاس‌های text مانند text-error و امثال آن می‌باشد:

<pre><div class="row-fluid"> <div class="span12"> <p class="muted"> متن غیرفعال <p class="text-warning"> نمایش اخطار به کاربر <p class="text-error"> نمایش خطا به کاربر <p class="text-info"> نمایش اطلاعات به کاربر <p class="text-success"> نمایش موفقیت آمیز بودن عملیات </div> </div></pre>	<p>متن بی صدا</p> <p>نمایش اخطار به کاربر</p> <p>نمایش خطا به کاربر</p> <p>نمایش اطلاعات به کاربر</p> <p>نمایش موفقیت آمیز بودن عملیات</p>
---	--

bootstrap تنظیماتی را به المان abbreviation موجود در HTML 5 نیز اعمال می‌کند. در این حالت کاربر با نزدیک ساختن اشاره‌گر ماوس به متن مشخص شده، یک tooltip توضیح دهنده نیز ظاهر خواهد شد:

<pre><div class="row-fluid"> <div class="span12"> <p> My <abbr title="منظور واحد پردازش مرکزی است"> CPU</abbr> has N Cores. </p> </div> </div></pre>	<p>My CPU has N Cores</p> <p>منظور واحد پردازش مرکزی است</p>
--	--

در bootstrap المان آدرس نیز شیوه نامه خاص خودش را داشته و به صورت یک قطعه خاص ظاهر می‌شود:

```
<div class="row-fluid">
  <div class="span12">
    <address>
      وحید نصیری
      <br />
      ایران، تهران
      <br />
      <abbr title="Phone">
        P:</abbr>
        12345678
      <br />
      <abbr title="Cell">
        C:</abbr>
        12345678
      </address>
    </div>
  </div>
```

وحید نصیری
ایران، تهران
P: 12345678
C: 12345678
Cell

در اینجا تاثیر bootstrap را بر روی المان blockquote ملاحظه می‌کنید؛ همچنین به همراه المان cite برای ذکر مآخذ نقل قول ذکر شده:

```
<div class="row-fluid">
  <div class="span12">
    <blockquote>
      <p>جهت نمایش نقل قول</p>
      <small><cite title="کاربر">
        شماره 2</cite> از شخصی
      </small>
    </blockquote>
  </div>
</div>
```

جهت نمایش نقل قول
— از شخصی
کاربر شماره ۲

در bootstrap برای حذف بولت‌های کنار یک لیست مرتب، فقط کافی است از کلاس unstyled استفاده شود:

```
<div class="row-fluid">
  <div class="span6">
    <ul class="unstyled">
      <li>یک</li>
      <li>دو</li>
      <li>سه</li>
    </ul>
  </div>
  <div class="span6">
    <ol>
      <li>یک</li>
      <li>دو</li>
      <li>سه</li>
    </ol>
  </div>
</div>
```

1. یک
2. دو
3. سه

به علاوه امکان تعریف دو نوع خاص از definition list (المان dl در اینجا). در حالت عادی، ابتدا عنوان مشخص شده با dt یا definition term به صورت ضخیم ظاهر می‌شود؛ به همراه توضیحات ارائه شده در المان dd آن در سطر بعدی. اگر از کلاس dl-horizontal استفاده شود، همانند تصویر ذیل، عنوان در کنار توضیحات در یک سطر قرار خواهد گرفت:

```
<div class="row-fluid">
  <div class="span12">
    <dl class="dl-horizontal">
      <dt>عنوان</dt>
```

عنوان توضیحات بلند در اینجا

```
<dd>
    توضیحات بلند در اینجا
</dd>
</div>
</div>
```

در bootstrap امکان اعمال شیوه نامه ابتدایی به کدهای برنامه‌ها نیز وجود دارد. اگر از تگ code استفاده شود، فرض بر این است که قرار است اطلاعاتی را در یک سطر نمایش دهید. اگر اطلاعات کدها، بیشتر از یک سطر است، می‌توان از تگ pre استفاده کرد. در اینجا با اعمال کلاس pre-scrollable به تگ pre، به صورت خودکار یک اسکرول عمودی به قطعه کدها اعمال خواهد شد:

```
<div class="row-fluid">
  <div class="span6">
    <code>inline code</code>
  </div>
  <div class="span6">
    <pre class="pre-scrollable">
      code
      code
      code
    </pre>
  </div>
</div>
```



استفاده از جداول و تاثیر bootstrap بر آنها

در ادامه کدهای یک جدول متداول را که مزین شده‌است به کلاس‌های bootstrap ملاحظه می‌کنید:

```
<div class="container-fluid">
  <div class="row-fluid">
    <div class="span12">
      <table
class="table table-striped table-hover table-bordered table-condensed">
        <caption>
          عنوانی خاص در اینجا
        </caption>
        <thead>
          <tr>
            <th>
              ستون یک
            </th>
            <th>
              ستون دو
            </th>
            <th>
              ستون سه
            </th>
          </tr>
        </thead>
        <tbody>
          <tr class="error">
            <td>
              متن یک 1
            </td>
            <td>
              متن دو 1
            </td>
            <td>
              متن سه 1
            </td>
          </tr>
          <tr>
            <td>
              متن یک 2
            </td>
            <td>
              متن دو 2
            </td>
            <td>
              متن سه 2
            </td>
          </tr>
        </tbody>
      </table>
    </div>
  </div>
</div>
```

```

        </td>
        <td>
            متن سه 2
        </td>
    </tr>
    <tr>
        <td>
            متن یک 3
        </td>
        <td>
            متن دو 3
        </td>
        <td>
            متن سه 3
        </td>
    </tr>
</tbody>
</table>
</div>
</div>
</div>

```

عنوانی خاص در اینجا

ستون يك	ستون دو	ستون سه
1 متن يك	1 متن دو	1 متن سه
2 متن يك	2 متن دو	2 متن سه
3 متن يك	3 متن دو	3 متن سه

در اینجا ذکر caption اختیاری است. وجود thead و tbody به تشخیص هدر و ردیف‌ها جهت اعمال شیوه‌نامه‌های متناظر کمک می‌کنند. همچنین کلاس‌های ذیل نیز به جدول اعمال شده‌اند:

table: سبب می‌شود تا تنظیمات ابتدایی bootstrap به جدول طراحی شده، اعمال شوند.

table-striped: رنگ زمینه سطرها را یک در میان تغییر می‌دهد.

table-hover: سبب می‌شود تا با عبور اشاره‌گر ماوس از روی سطرها، رنگ زمینه آن‌ها تغییر کنند.

table-bordered: حاشیه‌ای را به جدول و ردیف‌ها اعمال می‌کنند. همچنین سبب نمایش گوشه‌های گرد نیز می‌شود.

table-condensed: اندکی padding اعمال شده به سلول‌های جداول را کاهش می‌دهد و جدول را فشرده‌تر می‌کند.

در جدول فوق، کلاس نمونه error به یک tr نیز اعمال شده‌است تا اثر آن را بر روی یک ردیف بهتر بتوان ملاحظه کرد.

طراحی فرم‌ها و تاثیر bootstrap بر آن‌ها

قرار دادن برجسب‌ها و عناصر صفحه، به نحوی کاربرپسند و دلپذیر، نیاز به رعایت یک سری اصول تایپوگرافی و طراحی ویژه دارد که این موارد نیز در bootstrap گنجانده شده‌اند.

(1) فرم‌های عمودی

```

<div class="container-fluid">
    <div class="row-fluid">
        <div class="span12">
            <form action="/signup"
method="post">
                <fieldset>
                    <legend>ثبت نام</legend>
                    <label>
                        ایمیل:</label>
                    <input name="email"
type="text" />
                </fieldset>
            </div>
        </div>
    </div>
</div>

```

```

        <label>
            نام:</label>
        <input name="name" type="text"
    />
    </fieldset>
</form>
</div>
</div>
</div>

```

در اینجا یک فرم ساده را مشاهده می‌کنید. بدون bootstrap، تمام عناصر این فرم در یک سطر نمایش داده می‌شوند. اما با صرفاً فعال سازی css مرتبط با آن، به شکل مدرن فوق خواهیم رسید. همانطور که ملاحظه می‌کنید، کلیه عناصر را به صورت یک پشته عمودی در صفحه قرار داده‌است و نکته مهم اینجا است که هیچگونه شیوه نامه خاص و اضافه‌تری به فرم فوق از طرف ما اعمال نشده است.

نکته: اگر می‌خواهید همان حالت پیش فرض مرورگر، یعنی قرار دادن تمام عناصر در پشت سر هم را فعال کنید، تنها کافی است کلاس form-inline را به form تعریف شده فوق اضافه نمایید.

ثبت نام

نام:

ایمیل:

2) نکاتی در مورد checkboxes و radio buttons

در حالت پیش فرض، با تعریف یک label و checkbox، برچسب متناظر با آن اندکی بالاتر از checkbox قرار گرفته شده در صفحه ظاهر می‌شود. برای رفع این مشکل تنها کافی است کلاس checkbox به label اعمال شود (و برای radio button از کلاس radio استفاده خواهد شد):

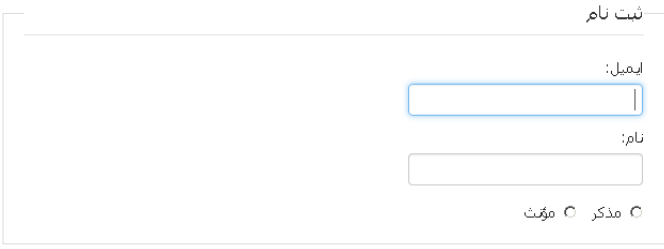
```

<label class="checkbox">
    <input type="checkbox" name="isMale" />
    مذکر
</label>

```

این مثال را یکبار با کلاس checkbox و بار دیگر بدون آن آزمایش کنید تا تفاوت را بهتر بتوان درک کرد.


نکته: برای قرار دادن چندین checkbox یا radio button در یک سطر (با توجه به حالت چیدمان عمودی پیش فرض فرم‌ها)، ابتدا آن‌ها را داخل یک div قرار دهید. سپس به تمام checkboxها یا radio buttonها کلاس inline را نیز اضافه نمایید. برای مثال:

<pre> <div> <label class="radio" inline"> <input type="radio" name="isMale" /> مذکر</label> <label class="radio" inline"> <input type="radio" name="isFemale" /> مؤنث</label> </div> </pre>	
---	--

(4) تعیین اندازه فیلدها

با استفاده از کلاس‌هایی مانند `input-mini`, `input-small`, `input-medium`, `input-large`, `input-xlarge` و `input-xxlarge` می‌توان اندازه فیلدها را کوچک‌تر از اندازه معمول یا بزرگتر کرد. یا حتی می‌توان از همان کلاس‌های `span` مانند `span12` (اختصاص کل عرض به یک فیلد) و امثال آن برای تعیین اندازه یک فیلد استفاده کرد. اگر علاقمند هستید که یک `textarea` کل عرض صفحه را به خود اختصاص دهد، از کلاس `input-block-level` استفاده کنید.

(5) فرم‌های جستجو

<pre> <form class="search-form"> <fieldset> <legend>جستجو</legend> <input type="search" class="search-query" /> <button type="submit" class="btn" >بیاب</button> </fieldset> </form> </pre>	
---	--

چند نکته در فرم‌های جستجوی طراحی شده با bootstrap نسبت به بقیه فرم‌ها حائز اهمیت است:

- کلاس فرم بهتر است `search-form` تعیین شود
- نوع `input` بهتر است `search` وارد شود
- کلاس `input` جستجو نیز `search-query` انتخاب گردد

همانطور که ملاحظه می‌کنید، در این حالت گوشه‌های جعبه متنی جستجو، نسبت به حالت‌های معمولی آن‌ها گرد شده است. کلاس دکمه نیز `btn` در نظر گرفته شده است تا حالت ویژه دکمه‌های bootstrap را پیدا کند.

(6) فرم‌های افقی

تا اینجا، با فرم‌های حالت پیش فرض یا فرم‌هایی که عناصر را به صورت پشت‌پشتی عمودی بر روی یکدیگر قرار می‌دهند، آشنا شدیم. حالت متداول دیگر طراحی فرم‌ها، حالت افقی است. به این معنا که در هر سطر، یک برچسب و یک المان قرار گیرند، بجای اینکه ابتدا برچسب نمایش داده شود و در سطر بعدی، المان مرتبط با آن. Bootstrap برای طراحی بدون استفاده از جداول این نوع فرم‌ها نیز تنظیمات خاصی را تدارک دیده است.

<pre> <form class="form-horizontal" action="/signup" method="post"> <fieldset> </pre>	
---	--

<pre> <legend>ثبت نام</legend> <div class="control-group"> <label class="control-label">ایمیل:</label> <div class="controls"> <input name="email" type="text" /> </div> </div> <div class="control-group"> <label class="control-label">نام:</label> <div class="controls"> <input name="name" type="text" /> </div> <div class="control-group"> <div class="controls"> <input type="radio" name="isMale" /> <label class="radio" type="radio" name="isMale">مذکر</label> <input type="radio" name="isFemale" /> <label class="radio" type="radio" name="isFemale">مؤنث</label> </div> </div> </div> </form> </pre>	
---	--

- مطابق مثال فوق، خلاصه طراحی فرم‌های افقی با Bootstrap به این نحو است:
- کلاس form-horizontal را به فرم جاری اضافه کنید.
 - هر سطر مورد نظر را در div ایی با کلاس control-group محصور نمایید.
 - به برچسب‌ها، کلاس control-label را انتساب دهید.
 - کنترل‌های مدنظر را در div ایی با کلاس controls محصور کنید.

هر چند این روش نیاز به اندکی HTML نویسی دارد، اما بسیاری به این نوع فرم‌ها بیشتر علاقمند هستند تا فرم‌های عمودی ابتدای بحث.

7) جلب توجه کاربران به فیلدها برای نمایش خطاهای اعتبارسنجی

<pre> <div class="control-group error"> <label class="control-label">ایمیل:</label> <div class="controls"> <input name="email" type="text" /> لطفاً ایمیل را با فرمت صحیحی وارد نمایید </div> </div> </pre>	
---	--

	<div><div></div><div>را با فرمت صحیحی وارد نمائید</div><div></div><div>مؤت</div></div>
--	--

برای تزریق خطاهای اعتبارسنجی ویژه، در اینجا می‌توان از کلاس‌های `help-block` و یا `help-inline` به همراه کلاس‌هایی مانند `success` و `error`، `info`، `warning` استفاده کرد.

8) توسعه فیلدهای استاندارد

<pre><div class="input-prepend input-append"> .00 <input dir="ltr" type="text" /> ریال </div></pre>	<div><div>ریال</div><div>123</div><div>.00</div></div>
---	--

توسط ترکیب کلاس‌های `input-prepend`، `input-append` و `span` با کلاس `add-on`، می‌توان عناصری بصری خاصی را به عنوان جزئی از فیلد مورد نظر اضافه کرد.

9) HTML Helpers مخصوص ASP.NET MVC برای کار با bootstrap

نکاتی را که در اینجا مطرح شدند، اگر علاقمند بودید که به شکلی `strongly typed` در ASP.NET MVC اعمال کنید، می‌توان به پروژه‌هایی مانند `TwitterBootstrapMvc` مراجعه کرد. تعداد این نوع پروژه‌ها هم روز به روز بیشتر می‌شوند:

<https://twitterbootstrapmvc.codeplex.com>

<https://mvc4bootstraphelper.codeplex.com>

<https://github.com/erichexter/twitter.bootstrap.mvc>

<http://bootstraphelpers.codeplex.com>

تنظیمات خاص دکمه‌ها در حین استفاده از Twitter Bootstrap

در مثال ذیل، کلاس‌های مرتبط با تزئین دکمه‌ها را توسط bootstrap، ملاحظه می‌کنید:

```
<div class="container-fluid">
  <div class="row-fluid">
    <div class="span12">
      <table
class="table table-striped table-hover table-bordered table-condensed">
        <thead>
          <tr>
            <th>
              دکمه
            </th>
            <th>
              لینک
            </th>
            <th>
              کلاس بکار گرفته شده
            </th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <td>
              <button class="btn">
                default</button>
            </td>
            <td>
              <a href="#" class="btn">default</a>
            </td>
            <td>
              <code>btn</code>
            </td>
          </tr>
          <tr>
            <td>
              <button class="btn btn-primary">
                primary</button>
            </td>
            <td>
              <a href="#" class="btn btn-primary">primary</a>
            </td>
            <td>
              <code>btn btn-primary</code>
            </td>
          </tr>
          <tr>
            <td>
              <button class="btn btn-info">
                info</button>
            </td>
            <td>
              <a href="#" class="btn btn-info">info</a>
            </td>
            <td>
              <code>btn btn-info</code>
            </td>
          </tr>
          <tr>
            <td>
              <button class="btn btn-success">
                success</button>
            </td>
            <td>
              <a href="#" class="btn btn-success">success</a>
            </td>
            <td>
              <code>btn btn-success</code>
            </td>
          </tr>
          <tr>
            <td>
              <button class="btn btn-warning">
                warning</button>
            </td>
            <td>
              <a href="#" class="btn btn-warning">warning</a>
            </td>
            <td>
              <code>btn btn-warning</code>
            </td>
          </tr>
        </tbody>
      </table>
    </div>
  </div>
</div>
```



```

        </td>
    </tr>
    <tr>
        <td>
            <button class="btn btn-danger">
                danger</button>
        </td>
        <td>
            <a href="#" class="btn btn-danger">danger</a>
        </td>
        <td>
            <code>btn btn-danger</code>
        </td>
    </tr>
    <tr>
        <td>
            <button class="btn btn-inverse">
                inverse</button>
        </td>
        <td>
            <a href="#" class="btn btn-inverse">inverse</a>
        </td>
        <td>
            <code>btn btn-inverse</code>
        </td>
    </tr>
    <tr>
        <td>
            <button class="btn btn-link">
                link</button>
        </td>
        <td>
            <a href="#" class="btn btn-link">link</a>
        </td>
        <td>
            <code>btn btn-link</code>
        </td>
    </tr>
    <tr>
        <td>
            <button class="btn btn-primary btn-large">
                large</button>
        </td>
        <td>
            <a href="#" class="btn btn-primary btn-large">large</a>
        </td>
        <td>
            <code>btn btn-primary btn-large</code>
        </td>
    </tr>
    <tr>
        <td>
            <button class="btn btn-primary btn-small">
                small</button>
        </td>
        <td>
            <a href="#" class="btn btn-primary btn-small">small</a>
        </td>
        <td>
            <code>btn btn-primary btn-small</code>
        </td>
    </tr>
    <tr>
        <td>
            <button class="btn btn-primary btn-mini">
                mini</button>
        </td>
        <td>
            <a href="#" class="btn btn-primary btn-mini">mini</a>
        </td>
        <td>
            <code>btn btn-primary btn-mini</code>
        </td>
    </tr>
    <tr>
        <td>
            <button class="btn btn-primary btn-block">
                block</button>
        </td>
        <td>
            <a href="#" class="btn btn-primary btn-block">block</a>

```

```

        </td>
        <td>
            <code>btn btn-primary btn-block</code>
        </td>
    </tr>
    <tr>
        <td>
            <code>button class="btn btn-primary disabled">
                disabled</code>
        </td>
        <td>
            <a href="#" class="btn btn-primary disabled">disabled</a>
        </td>
        <td>
            <code>btn btn-primary disabled</code>
        </td>
    </tr>
</tbody>
</table>
</div>
</div>
</div>

```

دکمه	لینک	کلاس بکار گرفته شده
		<code>btn</code>
		<code>btn btn-primary</code>
		<code>btn btn-info</code>
		<code>btn btn-success</code>
		<code>btn btn-warning</code>
		<code>btn btn-danger</code>
		<code>btn btn-inverse</code>
		<code>btn btn-link</code>
		<code>btn btn-primary btn-large</code>
		<code>btn btn-primary btn-small</code>
		<code>btn btn-primary btn-mini</code>
		<code>btn btn-primary btn-block</code>
		<code>btn btn-primary disabled</code>

همانطور که ملاحظه کردید، الزامی ندارد که این کلاس‌ها را حتماً به دکمه‌ها اعمال کرد. برای نمونه می‌توان از یک span یا لینک نیز برای تعریف دکمه‌ها بهره جست. برای یکپارچه سازی چنین دکمه‌هایی (که در اصل دکمه نیستند) با ASP.NET MVC می‌توان به مطلب تکمیلی « [استفاده از دکمه‌های CSS توئیت در ASP.NET MVC](#) » مراجعه نمود.

یک نکته: اگر علاقمند هستید که تعدادی دکمه را به شکل یک toolbar نمایش دهید، آن‌ها را در یک div محصور کرده و کلاس btn-group را به آن div اعمال نمائید.

کار با تصاویر و آیکون‌ها در Twitter Bootstrap

کلاس‌هایی مانند img-rounded, img-circle, img-polaroid با اعمال به یک تصویر، سبب گردن شدن گوشه‌های آن، نمایش دایره‌ای و یا نمایش به همراه حاشیه یک تصویر خواهند شد. Twitter Bootstrap به همراه صدها آیکون ارائه شده است. این آیکون‌ها توسط glyphicons.com ایجاد شده‌اند. روشی که برای استفاده از آن‌ها توصیه شده است، استفاده از تگ i می‌باشد. برای مثال:

```
<i class="icon-music"></i>icon-music
```

البته بدیهی است که محدودیتی در اینجا وجود نداشته و می‌توان این کلاس‌ها را به یک span نیز اعمال کرد. برای مشاهده لیست کلاس‌های قابل استفاده، کلمه icon-glass را در فایل bootstrap.css جستجو نمائید؛ تا شروع مدخل مرتبط با آیکون‌ها را بتوانید مشاهده نمائید. رنگ پیش فرض این آیکون‌ها مشکی است. اگر علاقمند بودید که آن‌ها را برای مثال با رنگ سفید نمایش دهید فقط کافی است کلاس icon-white را پس از کلاس آیکون مدنظر، ذکر کرد:

```
<i class="icon-music icon-white"></i>icon-music
```

از این نمونه قلم‌های سازگار با Twitter Bootstrap در آدرس fontawesome.github.com نیز قابل دریافت هستند. امکان اعمال این آیکون‌ها به دکمه‌ها نیز وجود دارد. برای مثال:

```
<button class="btn">
  <i class="icon-music"></i>دکمه</button>
```

کاربرد دیگر این آیکون‌ها در بحث توسعه فیلدهای استاندارد است که پیشتر بحث شد. برای مثال نمایش آیکون نامه در کنار فیلد دریافت ایمیل:

<pre><div class="input-prepend"> <i class="icon-envelope"></i> <input type="email" dir="ltr" /> </div></pre>	
--	--

نظرات خوانندگان

نویسنده: pezhman
تاریخ: ۱۵:۲ ۱۳۹۲/۰۳/۱۸

سلام
ممنون بابت آموزش
می خواستم بدونم چطور از راست به چپ رو درست کردین؟ مخصوصا فرم ها رو
آیا از pull-right استفاده کردین یا css دیگه؟
(ممنون :)

نویسنده: وحید نصیری
تاریخ: ۱۷:۵۱ ۱۳۹۲/۰۳/۱۸

من هیچ تغییر خاصی ندادم. فقط از بسته RTL [معرفی شده در قسمت طرحبندی](#) استفاده کردم.

نویسنده: _mehdi
تاریخ: ۶:۴۹ ۱۳۹۲/۰۳/۲۰

سلام
تشکر بابت آموزشهاون
شما برای تست صفحات وب در دستگاههای مختلف از جمله تبلت و موبایل اونهم با رزولیشنهای مختلف از چه ابزار یا نرم افزار استفاده میکنید.

نویسنده: وحید نصیری
تاریخ: ۸:۴۵ ۱۳۹۲/۰۳/۲۰

- جواب شخصی: چنین کاری رو نمیکنم. چون مطابق آمار حداقل سایت جاری حدود یک درصد یا کمتر از بازدیدکنندگان این سایت کاربر غیر PC هستند که عملا در محاسبات به حساب نمیان. (واقع گرایی)
- جواب کلی: یک سری شبیه ساز رو [از اینجا](#) میتونید دریافت کنید. [امکان یکپارچه سازی](#) با VS.NET هم وجود دارد.

پس از آشنایی با [طرح‌بندی صفحات](#) و [امکانات متداول](#) Twitter Bootstrap، در این قسمت به کامپوننت‌ها و اجزای تعاملی آن مانند منوها، برگه‌ها و امثال آن خواهیم پرداخت.

منوهای پایین افتادنی (Dropdown menus) در Twitter Bootstrap

برای کار با منوها، حداقل نیازهایی که باید به صفحه اضافه شوند، فایل css مرتبط با Twitter Bootstrap، فایل اسکریپت‌های جی‌کوئری و فایل bootstrap.min.js است. در ادامه، ساختار متداول یک منوی نمونه ایجاد شده را ملاحظه می‌کنید:

```
<div class="container-fluid">
  <div class="row-fluid">
    <div class="span12">
      <div class="dropdown">
        <a class="dropdown-toggle" data-toggle="dropdown" href="#">منوی پایین افتادنی</a>
        <span class="caret"></span></a>
        <ul class="dropdown-menu">
          <li><a href="http://www.site1.com">site1</a></li>
          <li><a href="http://www.site2.com">site2</a></li>
          <li><a href="http://www.site3.com">site3</a></li>
          <li class="divider"></li>
          <li class="dropdown-submenu"><a href="#">سایر موارد</a>
            <ul class="dropdown-menu">
              <li><a href="http://www.site4.com">site4</a></li>
              <li><a href="http://www.site5.com">site5</a></li>
            </ul>
          </li>
        </ul>
      </div>
    </div>
  </div>
</div>
```



در اینجا به یک div، کلاس dropdown نسبت داده شده است؛ از این جهت که می‌توان این منو را به انواع و اقسام المان‌های صفحه

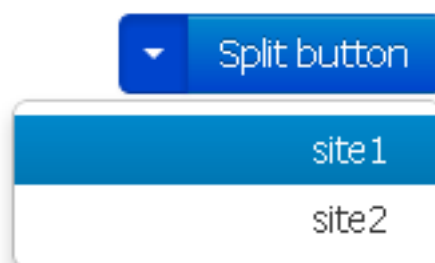
نیز نسبت داد. به این ترتیب، منو از محل قرارگیری این المان باز خواهد شد. در ادامه روال متداول تعریف منو را در Bootstrap، ملاحظه می‌کنید. کلاس dropdown-toggle سبب خواهد شد تا با کلیک ماوس بر روی لینک، منو ظاهر شود. کلاس caret نیز سبب نمایش مثلث رو به پایین، جهت مشخص سازی وجود منو در این محل خاص، تعریف شده است. المان‌های منو، توسط لیست‌ها تعریف می‌شوند. این لیست باید با کلاس dropdown-menu شروع شود. سپس هر عضو این لیست، یک آیتم منو را تشکیل خواهد داد. اگر نیاز به خط جدا کننده‌ای در این میان باشد، باید از کلاس divider استفاده کرد. در اینجا همچنین امکان تعریف منوهای تو در تو نیز وجود دارد. برای اینکار تنها کافی است یک لیست دیگر را در این میان اضافه کرد. کلاس dropdown-submenu به عضو لیست دربرگیرنده زیر منو، اضافه می‌شود. زیر منو نیز دارای کلاس dropdown-menu خواهد بود.

چند نکته تکمیلی در مورد منوها:

- همانطور که در قسمت بررسی دکمه‌های Bootstrap نیز عنوان شد، برای تبدیل یک المان به دکمه فقط کافی است کلاس btn را به آن انتساب دهیم. در اینجا نیز می‌توان کلاس لینک منوی پایین افتادنی را به صورت btn btn-primary dropdown-toggle کرد تا ظاهر آن تبدیل به یک دکمه Bootstrap شود.
- در اولین div مثال فوق، کلاس dropdown ذکر شده است. می‌توان این کلاس را به dropup تغییر داد تا نحوه باز شدن منوی تعریف شده رو به بالا باشد؛ بجای رو به پایین.
- برای ساخت split button که تشکیل شده است از دو دکمه اصلی و دکمه کوچکی در کنار آن که یک زیرمنو را نمایش می‌دهد، باید به نحو زیر عمل کرد:

```
<div class="container-fluid">
  <div class="row-fluid">
    <div class="span12">
      <div class="btn-group">
        <a class="btn btn-primary" href="#">Split button</a> <a class="btn btn-primary
dropdown-toggle"
        data-toggle="dropdown" href="#"><span class="caret"></span></a>
        <ul class="dropdown-menu">
          <li><a href="http://www.site1.com">site1</a></li>
          <li><a href="http://www.site2.com">site2</a></li>
        </ul>
      </div>
    </div>
  </div>
</div>
```

تنها نکته‌ای که در اینجا نسبت به حالت قبل اضافه شده است، استفاده از کلاس btn-group است. این کلاس سبب می‌شود تا دکمه‌های تعریف شده در کنار هم به شکل یک toolbar مرتب شوند.



تعریف برگه‌ها در Twitter Bootstrap

برای تعریف منوهای راهنمای سایت به شکل برگه، کار با استفاده از کلاس nav و تعریف یک لیست ساده شروع می‌شود:

```

<div class="container">
  <div class="row">
    <h2>
      Navigational menus
    </h2>
    <div class="span3">
      <ul class="nav nav-tabs">
        <li class="nav-header">nav nav-tabs</li>
        <li class="active"><a href="#">Home</a></li>
        <li><a href="#">Contact</a></li>
        <li><a href="#">Login</a></li>
      </ul>
    </div>
    <div class="span3">
      <ul class="nav nav-tabs nav-stacked">
        <li class="nav-header">nav nav-tabs nav-stacked</li>
        <li class="active"><a href="#">Home</a></li>
        <li><a href="#">Contact</a></li>
        <li><a href="#">Login</a></li>
      </ul>
    </div>
    <div class="span3">
      <ul class="nav nav-pills">
        <li class="nav-header">nav nav-pills</li>
        <li class="active"><a href="#">Home</a></li>
        <li><a href="#">Contact</a></li>
        <li><a href="#">Login</a></li>
      </ul>
    </div>
    <div class="span3">
      <ul class="nav nav-list">
        <li class="nav-header">nav nav-list</li>
        <li class="active"><a href="#">Home</a></li>
        <li><a href="#">Contact</a></li>
        <li><a href="#">Login</a></li>
        <li class="dropdown"><a class="dropdown-toggle" data-toggle="dropdown"
href="#">Menu
          ...<span class="caret"></span></a>
          <ul class="dropdown-menu">
            <li><a href="http://www.site1.com">site1</a></li>
            <li><a href="http://www.site2.com">site2</a></li>
          </ul>
        </li>
      </ul>
    </div>
  </div>
</div>

```

Navigational menus

Login

Contact

Home

NAV NAV-TABS

NAV NAV-TABS NAV-STACKED

Home

Contact

Login

Login

Contact

Home

NAV NAV-PILLS

NAV NAV-LIST

Home

Contact

Login

... Menu

site1

site2

- اضافه شدن کلاس nav، بولت‌های لیست را حذف و ویژگی تغییر رنگ زمینه هر یک از عناصر لیست را با حرکت اشاره‌گر ماوس بر روی آن فعال می‌کند.
- برای اعمال اشکال مختلف به کلاس nav، می‌توان از سایر کلاس‌های مرتبط استفاده کرد. اگر از nav-tabs استفاده شود، این لیست عمودی تبدیل به یک لیست افقی خواهد شد. اگر کلاس active به یکی از آیتم‌های آن نسبت داده شود (بیانگر صفحه جاری)، این گزینه حالت انتخاب شده را پیدا می‌کند.
- در هر حالت، برای تبدیل چیدمان افقی به عمودی، از کلاس nav-stacked استفاده نمایید.
- کلاس nav-pills، برای نمایش عنصر فعال، از یک مستطیل با گوشه‌های گرد استفاده می‌کند.
- اگر می‌خواهید این گوشه‌ها گرد نباشند، از nav-list استفاده کنید.
- در هر کدام از حالت‌ها می‌توان از یک گزینه اختیاری با کلاس nav-header برای نمایش متنی خاکستری بالای منوی راهبری استفاده کرد.
- همانطور که در آخرین لیست اضافه شده در مثال فوق ملاحظه می‌کنید، امکان تعریف منوهای پایین افتادنی در اینجا نیز میسر است. در این حالت کلاس li تعریف شده باید dropdown تعریف شود. مابقی نکات آن مانند قبل است.

تعریف یک Tab به همراه محتوای برگه‌های آن در داخل یک صفحه

در حالت تعریف منوهای راهبری فوق، هر عنصر برگه‌های تعریف شده به یک صفحه جدید لینک شده بود. اگر نیاز باشد تا هر برگه، محتوایی داخل همان صفحه داشته باشد باید به نحو ذیل عمل کرد:

```
<div class="container">
  <div class="row">
    <div class="span12">
      <div class="tabbable">
        <ul class="nav nav-tabs">
          <li class="active"><a data-toggle="tab" href="#tab1">Home</a></li>
          <li><a data-toggle="tab" href="#tab2">About</a></li>
        </ul>
        <div class="tab-content">
          <div class="tab-pane active" id="tab1">
            data ... data ...
          </div>
          <div class="tab-pane" id="tab2">
            data2 ... data2 ...
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

About

Home

... data2 ... data2

در اینجا کار با کلاس tabbable شروع می‌شود. سپس از یک لیست با کلاس nav nav-tabs برای تعریف سرتیترهای برگه‌ها استفاده می‌کنیم.

اگر به مثال فوق دقت کنید، اینبار hrefهای تعریف شده به tab1 و tab2 اشاره می‌کنند. ایندو در حقیقت id محتوای متناظر هستند که در قسمت div ایی با کلاس tab-content به div هایی با id مساوی tab1 و tab2 انتساب داده شده‌اند.

شیوه دوم تعریف منوی راهبری سایت به کمک Twitter Bootstrap

اگر علاقمند باشید که منوی راهبری بالای سایت دارای یک حاشیه به همراه مسایل طراحی واکنش‌گرا باشد، می‌توان از کامپوننت navbar مجموعه Bootstrap استفاده کرد:

```
<div class="container">
  <div class="row">
    <div class="span12">
      <nav class="navbar navbar-inverse">
        <div class="navbar-inner">
          <a href="/" class="brand">نام سایت در اینجا</a>
          <a class="btn btn-navbar" data-toggle="collapse" data-target=".nav-collapse">
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
          </a>
        </div>
      </nav>
    </div>
  </div>
</div>
```

```

        </a>
        <div class="nav-collapse collapse">
            <ul class="nav">
                <li><a href="/about">درباره</a></li>
                <li><a href="/contact">تماس</a></li>
                <li class="dropdown"><a class="dropdown-toggle" data-toggle="dropdown"
href="#">Menu
                    ...<span class="caret"></span></a>
                    <ul class="dropdown-menu">
                        <li><a href="http://www.site1.com">site1</a></li>
                        <li><a href="http://www.site2.com">site2</a></li>
                    </ul>
                </li>
            </ul>
            <form class="navbar-search pull-right input-append" action="/search">
                <input class="span2" type="search" />
                <button class="btn" type="button">
جستجو
            </form>
        </div>
    </div>
</nav>
</div>
</div>
</div>

```



navbar نیز همانند بسیاری از کامپوننت‌های Bootstrap، بر اساس ul و liها کار می‌کند. در ابتدای کار، کلاس nav را به تگ ul انتساب می‌دهیم. این مورد با مباحث قسمت قبلی در مورد تدارک برگه‌ها، تفاوتی نمی‌کند. به این ترتیب بولت‌های لیست حذف شده و همچنین تغییر رنگ پس زمینه، با حرکت اشاره‌گر ماوس بر روی آیتم‌ها را سبب می‌شود. سپس کل این ul و liها، داخل تگ nav قرار می‌گیرند با کلاسی مساوی navbar. تا اینجا لیست عمودی تبدیل به یک لیست افقی می‌شود. در ادامه برای اضافه کردن حاشیه‌ای به اطراف این منوی راهنبری و تمایز آن از سایر قسمت‌های صفحه، یک div با کلاس navbar-inner را اضافه خواهیم کرد.

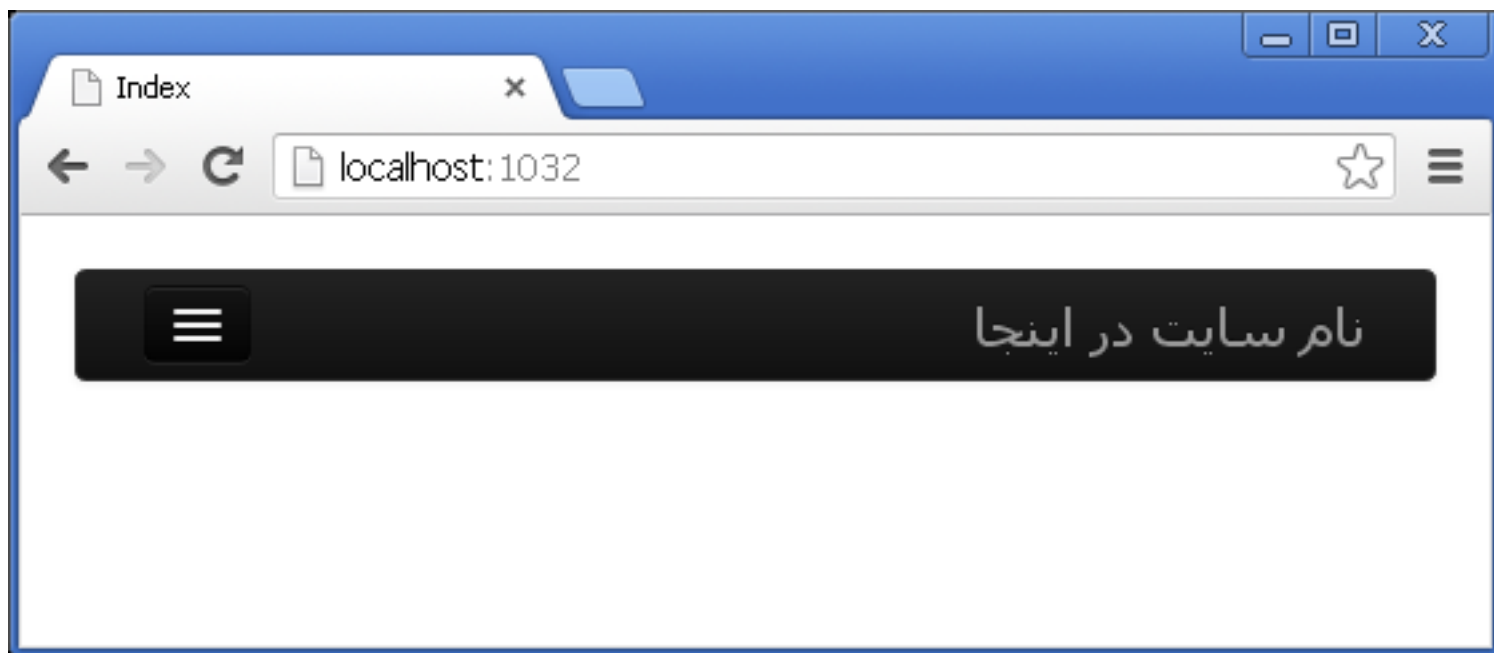
عموماً مرسوم است که در منوی راهنبری اصلی سایت، نام سایت یا متنی مشخص نیز در قسمتی از این منو ظاهر شود. برای انجام اینکار، کافی است یک لینک را با کلاس brand اضافه کنیم. این موارد، اصول کلی تهیه یک منوی راهنبری اصلی سایت را توسط Bootstrap بیان می‌کنند. در ادامه به چند نکته تکمیلی خواهیم پرداخت:

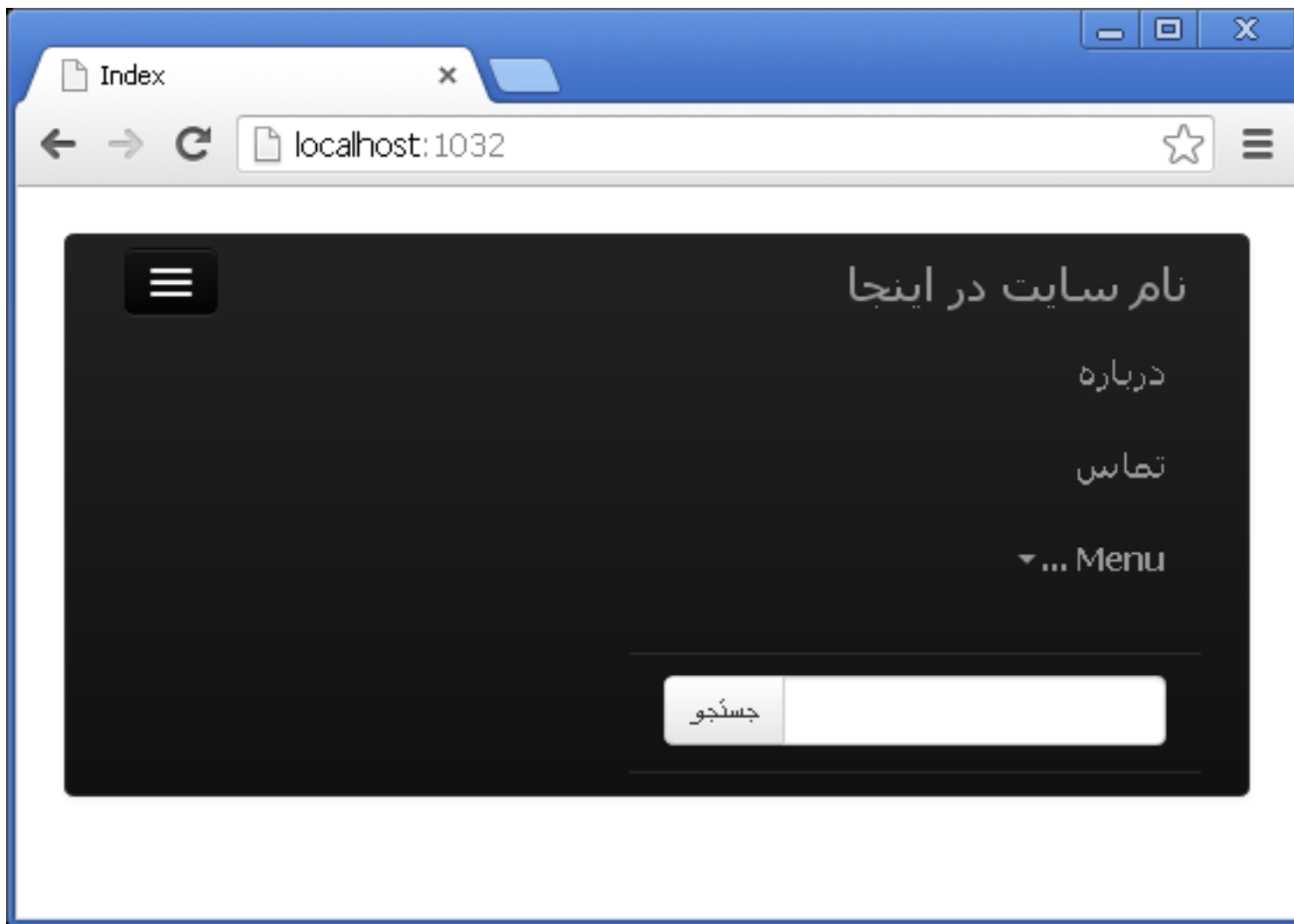
- اگر به تگ nav، کلاس navbar-static-top را اضافه کنیم، فاصله بین منو و اطراف صفحه را حذف می‌کند.
- اگر علاقمند هستیم که منوی بالای صفحه حتی با اسکرول به پایین محتوای نمایش داده شده، همواره نمایان باشد و بر روی

محتوا قرار گیرد، می‌توان به تگ `nav`، کلاس `navbar-fixed-top` را اضافه کرد. یا اگر نیاز است این منو در پایین صفحه به صورت ثابت نمایش داده شود از `navbar-fixed-bottom` می‌شود استفاده کرد.

در این حالت خاص ممکن است منو، قسمتی از ابتدای محتوای صفحه را مخفی کند و این محتوا زیر منو واقع شود. برای بهبود آن باید `margin-top` مرتبط با `body` را مقدار دهی کنید (حدود 40px کافی است).

- استفاده از کلاس `navbar-inverse` (همانند مثال فوق)، سبب معکوس شدن رنگ زمینه منوی راهبری می‌شود.
- همچنین در اینجا نیز می‌توان یک `dropdown-menu` را به نحوی که ملاحظه می‌کنید، به منوی اصلی سایت افزود.
- امکان قرار دادن فرم‌های خاصی مانند فرم جستجو نیز در منوی راهبری وجود دارد. در این حالت از کلاس `navbar-search` مانند مثال فوق استفاده خواهد شد. کلاس `pull-right` این المان را به سمت راست صفحه (در اینجا به سمت چپ با توجه به RTL بودن قالب)، هدایت می‌کند و کلاس `input-append` سبب می‌شود تا دکمه و تکست باکس تعریف شده در یک سطر و کنار هم قرارگیرند تا ظاهر بهتری حاصل شود. در حالت کلی کلاس `navbar-form` نیز برای این نوع فرم‌ها در نظر گرفته شده است.
- اگر دقت کرده باشید، کل محتوای منو، به همراه فرم تعریف شده، در یک `div` با کلاس `nav-collapse` قرار گرفته است. هدف از اینکار ناپدید شدن خودکار قسمت‌های طولانی یا نچندان مهم، در حین مرور سایت با اندازه‌های مرورگر کوچکتر مانند مرورگرهای موبایل است. همچنین اگر بخواهیم در این حالت که قسمت‌هایی ناپدید شده‌اند، یک دکمه با سه سطر که بیانگر منویی مخفی است را نمایش دهیم می‌توان از کلاس `btn-navbar` استفاده کرد؛ که نمونه‌ای از آن را در مثال فوق ملاحظه می‌کنید. این دکمه بر اساس `data-target` مشخص شده، سبب مخفی یا ظاهر شدن قسمتی از صفحه می‌گردد.





تعیین موقعیت کاربر در صفحه به کمک breadcrumbs در Bootstrap

```
<div class="container">
  <div class="row">
    <div class="span12">
      <ul class="breadcrumb">
        <li><a href="/level1">سطح یک</a> <span class="divider"></span></li>
        <li><a href="/level2">سطح دو</a> <span class="divider"></span></li>
        <li class="active">سطح سه</li>
      </ul>
    </div>
  </div>
</div>
```

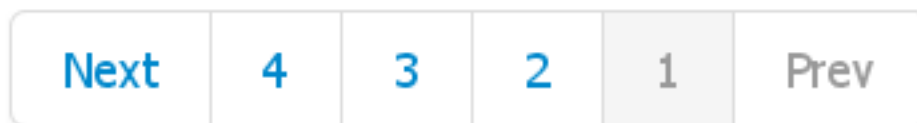
سطح يك / سطح دو / سطح سه

عموما مرسوم است در یک سایت چند سطحی، موقعیت جاری کاربر، به نحوی به او اعلام شود. یکی از این روش‌ها استفاده از breadcrumbs است که نمونه‌ای از آن‌را در مثال فوق ملاحظه می‌کنید.

- در اینجا نیز برای تعریف breadcrumbs از ul و li استفاده شده است به همراه کلاس breadcrumb.
- برای ایجاد فاصله و نمایش یک خط جداکننده، از کلاس divider کمک گرفته شده است.
- صفحه جاری با کلاس active مشخص گردیده؛ بدون تعریف لینک.

تعریف قسمت صفحه بندی یک لیست یا گرید به کمک Bootstrap

```
<div class="container">
  <div class="row">
    <div class="span12">
      <div class="pagination pagination-centered pagination-large">
        <ul>
          <li class="disabled"><a href="#">Prev</a></li>
          <li class="active"><a href="/page/1">1</a></li>
          <li><a href="/page/2">2</a></li>
          <li><a href="/page/3">3</a></li>
          <li><a href="/page/4">4</a></li>
          <li><a href="#">Next</a></li>
        </ul>
      </div>
    </div>
  </div>
</div>
```



برای نمایش زیباتر قسمت صفحه بندی یک لیست یا گرید می‌توان از یک سری ul و li مزین شده با کلاس pagination استفاده کرد. کلاس pagination-centered، آن را در وسط صفحه قرار خواهد داد (حالت‌های چپ و راست نیز وجود دارند). کلاس pagination-large، این قسمت را اندکی بزرگتر از معمول نمایش می‌دهد. در اینجا کلاس disabled، سبب غیرفعال نمایش داده شدن لینک به یک صفحه و کلاس active، بیانگر نمایش صفحه جاری است.

نمایش برچسب‌ها و اعلانات با زمینه‌های مختلف به کمک Bootstrap

برای نمایش برچسب‌های کوتاه با زمینه‌ای متفاوت از کلاس label استفاده می‌شود. افزودن کلاس‌هایی مانند label-success و label-info، رنگ‌های متفاوتی را سبب خواهند شد. برای گرد کردن بیش از حد گوشه این برچسب‌ها از کلاس badge استفاده می‌شود.

```
<div class="container">
  <div class="row">
    <div class="span12">
      <section>
        <ul class="unstyled">
          <li><span class="label">پیش فرض</span></li>
          <li><span class="label label-success">موفقیت آمیز</span></li>
          <li><span class="label label-warning">خطا</span></li>
          <li><span class="label label-important">مهم است</span></li>
          <li><span class="label label-info">اطلاعات</span></li>
          <li><span class="label label-inverse">رنگ معکوس</span></li>
        </ul>
      </section>
    </div>
  </div>
```

```

        <ul class="unstyled">
          <li><span class="badge">پیش فرض</span></li>
          <li><span class="badge badge-success">موفقیت آمیز</span></li>
          <li><span class="badge badge-warning">اخطار</span></li>
          <li><span class="badge badge-important">مهم است</span></li>
          <li><span class="badge badge-info">اطلاعات</span></li>
          <li><span class="badge badge-inverse">رنگ معکوس</span></li>
        </ul>
      </section>
      <div>
        <div class="alert">
          <button class="close" data-dismiss="alert">&times;</button>
          نمایش اعلانات
        </div>
        <div class="alert alert-danger">
          <button class="close" data-dismiss="alert">&times;</button>
          نمایش اعلانات
        </div>
        <div class="alert alert-info">
          <button class="close" data-dismiss="alert">&times;</button>
          نمایش اعلانات
        </div>
        <div class="alert alert-success">
          <button class="close" data-dismiss="alert">&times;</button>
          نمایش اعلانات
        </div>
        <div class="alert alert-error">
          <button class="close" data-dismiss="alert">&times;</button>
          نمایش اعلانات
        </div>
      </div>
    </div>
  </div>
</div>

```

پیش فرض

موفقیت آمیز

خطا

مهم است

اطلاعات

رنگ معکوس

پیش فرض

موفقیت آمیز

خطا

مهم است

اطلاعات

رنگ معکوس

×

نمایش اعلانات

×

نمایش اعلانات

×

نمایش اعلانات

×

نمایش اعلانات

×

نمایش اعلانات

همانطور که ملاحظه می‌کنید برای نمایش اعلانی به کاربر می‌توان از کلاس alert استفاده کرد. افزودن کلاس‌هایی مانند alert-danger، رنگ‌های متفاوتی را ارائه خواهند داد و اگر علاقمندید که کاربر بتواند این اخطار را با کلیک بر روی دکمه ضربدر کنار آن حذف کند، از یک دکمه با کلاس close و data-dismiss مساوی alert استفاده کنید. این اطلاعات به صورت خودکار توسط اسکریپت‌های bootstrap پردازش می‌شوند.

نظرات خوانندگان

نویسنده: راد

تاریخ: ۱۶:۳۹ ۱۳۹۲/۰۴/۲۹

با سلام و تشکر از مطالب مفید سایتتون
من یک مشکل دارم تمام مراحل رو طبق دستورالعمل شما انجام دادم اما dropdown منوی ایجاد شده وقتی روش کلیک می‌کنم
هیچ اتفاقی نمی‌افته و باز نمی‌شه.
CSS ها JS ها رو هم در باندل و هم مستقیماً در layout اضافه کردم باز جواب نگرفتم
update-package هم هیچ کمکی نکرد لطفا راهنمایی کنید

نویسنده: وحید نصیری

تاریخ: ۱۹:۳۹ ۱۳۹۲/۰۴/۲۹

دریافت یک نمونه مثال: [BootstrapMenu.zip](#)

+ [نحوه بازسازی وابستگی‌ها](#)

DOM در حالت عادی بسیار نامرتب است. همچنین با افزودن کلاس‌های CSS، کد HTML به مراتب نامرتب‌تر از قبل می‌شود. بوت استرپ نیز شامل تعداد زیادی از کلاس‌های CSS می‌باشد که برای انجام وظایف خاصی به HTML اضافه می‌شوند. روش متداول استفاده از بوت استرپ

Embedd کردن کلاس‌های CSS بوت استرپ به صورت مستقیم درون HTML

اغلب فریم‌ورک‌ها، از لحاظ معنایی یا semantic، دارای مشکل هستند. اگر به سورس HTML صفحاتی که با این نوع از فریم‌ورک‌ها ساخته شده باشند نگاهی بیندازید با حجم زیادی از کلاس‌هایی مانند `<div class="row">` و یا `<div class="col-sm">` مواجه خواهید شد. نوشتن کدهای HTML به این صورت از لحاظ معنایی اشتباه است. مثلاً اگر بنا به دلایلی سازندگان بوت استرپ تصمیم بگیرند نام کلاس‌های را در نسخه بعدی این فریم‌ورک تغییر دهند (مانند تغییر نام کلاس‌ها در نسخه‌ی 3 بوت استرپ)، و یا اگر در آینده بخواهید از یک فریم‌ورک دیگر در سایت‌تان استفاده کنید. باید این تغییرات را در تمام صفحات سایت‌تان اعمال کنید؛ در نتیجه اینکار زمان زیادی را از شما صرف می‌کند.

راه‌حل؟

استفاده از CSS preprocessors

بوت استرپ، از [Less](#) برای اینکار استفاده می‌کند. Less در واقع یک CSS preprocessor نوشته شده با جاوا اسکریپت است که قابلیت اجرا در مرورگر را دارد. Less امکانات زیادی، از قبیل استفاده از توابع، متغیرها، Mixins و ... را در اختیار شما قرار می‌دهد. در واقع هدف از Less، نگهداری آسان و قابلیت توسعه فایل‌های CSS می‌باشد. در این حالت شما کدهای CSS خود را درون فایل‌هایی با پسوند Less می‌نویسید. در این حالت بجای پیوست کردن کلاس‌های بوت استرپ در کد HTML، آن را درون استایل‌شیت پیوست خواهید کرد. همانطور که عنوان شد، بوت استرپ با Less نوشته شده است. فایل‌های Less بوت استرپ را می‌توانید از مخزن کد [گیت‌هاب](#) آن دانلود نمائید یا اینکه از طریق [نیوگت](#) می‌توانید آن را نصب کنید:

```
PM> Install-Package Twitter.Bootstrap.Less
```

کار با Less خیلی ساده است. به عنوان مثال در کد زیر یک کلاس با نام loud داریم که استایل‌هایی را به آن اعمال کرده‌ایم. اکنون جهت استفاده مجدد از این استایل‌ها برای کلاسی دیگر، نیاز به نوشتن مجدد آن نیست. کافی همانند یک تابع در هر کلاسی آن را فراخوانی کنیم:

```
.loud {
  color: red;
}

// Make all H1 elements loud
h1 {
  .loud;
}
```

نکته: در Visual Studio 2012 Update 2 به بعد به صورت توکار از فایل‌های Less [پشتیبانی](#) می‌شود. (توسط پلاگین [Web Essentials](#))

استفاده از Mixins

با استفاده از Mixins می‌توانیم عناصر داخل صفحات مان را به صورت [Semantic](#) تعریف نمائیم. به عنوان مثال می‌خواهیم با استفاده از سیستم گرید بوت استرپ، ساختاری مانند تصویر زیر را داشته باشیم:

Content Main
8 columns wide

Content
Secondary
4 columns wide

در حالت معمول با استفاده از کلاس‌های CSS بوت استرپ می‌توانیم اینکار را انجام دهیم:

```
<div class="container">
  <div class="row">
    <div class="col-md-8">
      Content - Main
    </div>
    <div class="col-md-4">
      Content - Secondary
    </div>
  </div>
</div>
```

کد فوق را بهتر است به این صورت بنویسیم:

```
<div class="wrapper">
  <div class="content-main">
    Content - Main
  </div>
  <div class="content-secondary">
    Content - Secondary
  </div>
</div>
```

در بوت استرپ از [Less Mixins](#) جهت اعمال استایل‌هایی مانند row و column می‌توانیم استفاده کنیم. به طور مثال برای اعمال استایل به کلاس‌های فوق می‌توانیم به این صورت عمل کنیم:

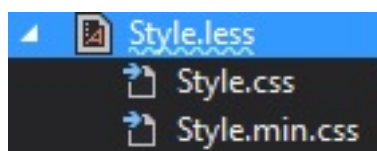
```
// Core variables and mixins
@import "variables.less";
@import "mixins.less";
.wrapper {
  .make-row();
}
.content-main {
```

```
.make-lg-column(8);
}
.content-secondary {
  .make-lg-column(3);
  .make-lg-column-offset(1);
}
```

کد فوق بعد از کامپایل به کد زیر تبدیل خواهد شد:

```
.wrapper {
  margin-left: -15px;
  margin-right: -15px;
}
.content-main {
  position: relative;
  min-height: 1px;
  padding-left: 15px;
  padding-right: 15px;
}
@media (min-width: 1200px) {
  .content-main {
    float: left;
    width: 66.66666666666666%;
  }
}
.content-secondary {
  position: relative;
  min-height: 1px;
  padding-left: 15px;
  padding-right: 15px;
}
@media (min-width: 1200px) {
  .content-secondary {
    float: left;
    width: 25%;
  }
}
@media (min-width: 1200px) {
  .content-secondary {
    margin-left: 8.333333333333333%;
  }
}
```

همانطور که قبلاً عنوان شد ویژوال استودیو به راحتی توسط افزونه Web Essentials از فایل‌های Less پشتیبانی می‌کند. در نتیجه کامپایل فایل‌های Less داخل ویژوال استودیو توسط این افزونه به راحتی قابل انجام می‌باشد. یک قابلیت جالب دیگر در رابطه با فایل‌هایی Less، تولید نسخه‌های CSS عادی و فشرده نهایی توسط افزونه Web Essentials می‌باشد. به طور مثال شما می‌توانید نسخه minified شده را به Layout تان اضافه کنید. بعد از هربار تغییر در فایل Less این فایل نیز به روز خواهد شد:



همچنین برای دیگر اجزای بوت‌استرپ نیز می‌توانید به این صورت عمل کنید:

```
<!-- Before -->
<a href="#" class="btn danger large">Click me!</a>

<!-- After -->
<a href="#" class="annoying">Click me!</a>

a.annoying {
  .btn;
  .btn-danger;
```

```
.btn-large;  
}
```

خب، با استفاده از این حالت، کدهای HTML به صورت مرتب‌تر، قابل انعطاف‌تر و همچنین از لحاظ معنایی (Semantic) استاندارد خواهند بود. بنابراین با آمدن یک فریم‌ورک جدید، به راحتی امکان سوئیچ کردن برای ما میسر و آسان‌تر از قبل خواهد شد.

نظرات خوانندگان

نویسنده: سعید شیرزادیان

تاریخ: ۱۳۹۳/۰۴/۰۳ ۰:۰

سلام. مطلب جالبی بود. آیا قابلیت فوق برای حالت بدون Less امکان دارد. من در یک پروژه از بوت استرپ استفاده کردم و یکسری تغییرات فارسی نیز در آن اعمال شده است. راهکاری دارید؟

نویسنده: وحید نصیری

تاریخ: ۱۳۹۳/۰۴/۰۳ ۰:۵۷

نسخه‌ی LESS راست به چپ بوت استرپ هم موجود است.
ضمناً نگارش 3.2 بوت استرپ قرار است که به صورت رسمی RTL پشتیبانی کند.

نویسنده: حسن محمدی

تاریخ: ۱۳۹۳/۰۴/۰۳ ۱۳:۳۵

با سلام و تشکر از سایت بی نهایت خوبتون

مقاله عالی بود اگر امکانش بود به تمپلیت کوچک سایتو بوت استرپ و less بگید بی نهایت ممنون میشم. من خودم تو استفاده از less خیلی مشکل دارم خیلی‌ها هم مثل خودم می‌شناسنم. من آپدیت 2 ویژوال استودیو 2013 و webessenial رو نصب کردم تا فایل‌های less رو برام باز کرد.

فقط bootstrap.less رو بدون مشکل باز میکنم و نسخه css رو میاره ولی مثلاً هر فایل دیگه ای مثل alerts.less رو باز میکنم پیغام خطا میده که مثلاً variable @alert-padding is undefined یا چیزهای دیگرو میگه undefined به نظرتون مشکل از چیه؟

نویسنده: وحید نصیری

تاریخ: ۱۳۹۳/۰۴/۰۳ ۱۳:۴۵

مشکلی ندارد. نسخه‌ی less بوت استرپ از چندین و چند فایل تشکیل شده که نهایتاً از کامپایل این‌ها یک فایل نهایی تولید می‌شود. برای مثال اگر فایل bootstrap.less را باز کنید، ترتیب import فایل‌ها قابل ملاحظه هستند. ابتدا فایل variables.less ذکر شده، چون مقادیر پیش فرض متغیرهای سایر فایل‌ها در آن موجود است و همینطور الی آخر. فایل نهایی که سبب کامپایل تمام تغییرات خواهد شد، فایل bootstrap.less است.

نویسنده: حسن محمدی

تاریخ: ۱۳۹۳/۰۴/۰۳ ۱۳:۵۸

ممنون که پاسخ دادید من اگر بخوام مثلاً navbar رو تغییر بدم باید bootstrap.less رو ویرایش کنم یا navbar.less ?
بعد چیکار کنم که خطا برطرف شه و این ترتیبو بفهمم و مثلاً navbar.less رو باز میکنم دیگه خطا نده؟
ممنون

نویسنده: وحید نصیری

تاریخ: ۱۳۹۳/۰۴/۰۳ ۱۴:۲۶

- هیچکدام. فایل navbar.less تشکیل شده از یک سری متغیر. این متغیرها در فایل variables.less مقدار دهی شده‌اند. بنابراین برای تغییر آن باید فایل variables.less ویرایش شوند (در این فایل، Navbar را جستجو کنید).
- یک کپی از فایل اصلی bootstrap.less را مثلاً به نام test.less ایجاد کنید (با همان محتوا). المان‌های مختلف آن را حذف کنید تا به حداقل وابستگی‌هایی که برای کامپایل navbar.less نیاز است برسید:

```
// Core variables and mixins
@import "variables.less";
@import "mixins.less";
```

```
// Core CSS
@import "forms.less";

// Components
@import "navbar.less";

// Utility classes
@import "utilities.less";
```