

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:

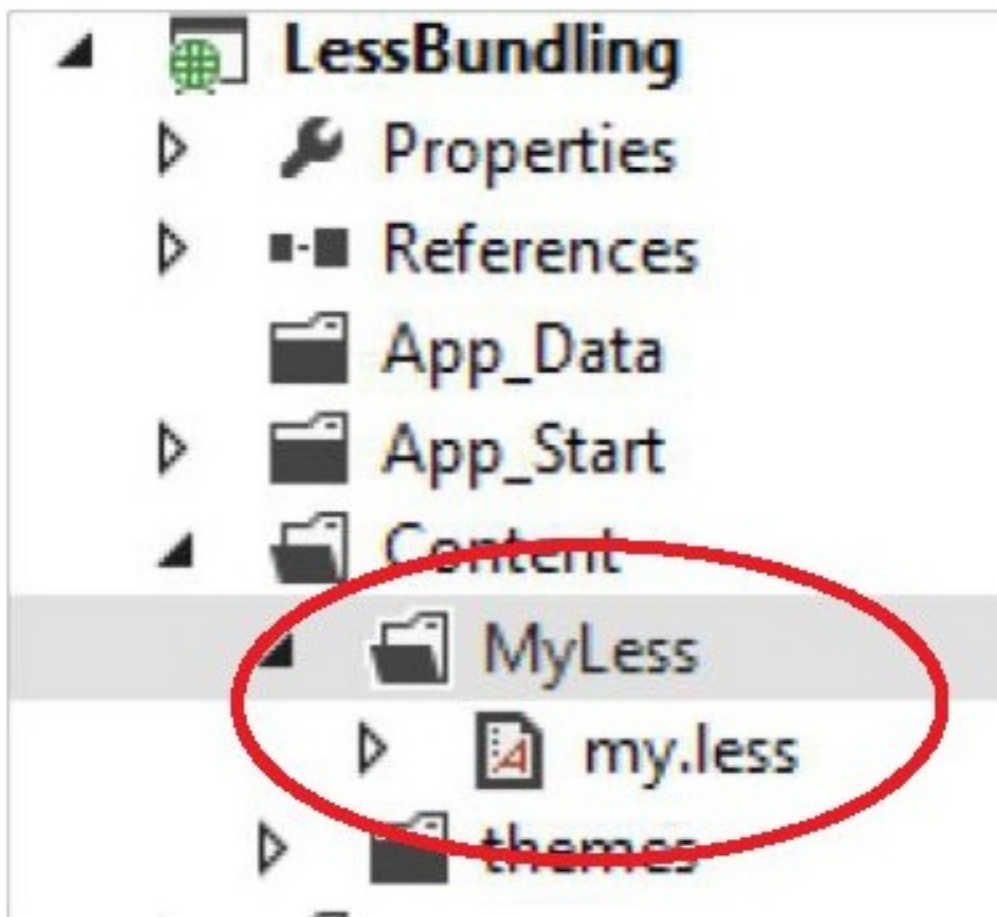
1- به منوی project و بخش Manage NuGet Packages ... رفته و dotless را جستجو و نصب نمایید.

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](#)

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 نمایید 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:

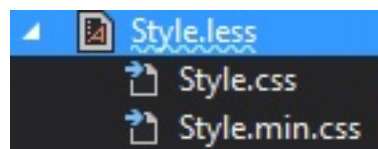
```
// Core variables and mixins
@import "variables.less";
@import "mixins.less";
.wrapper {
  .make-row();
}
.content-main {
```

```
.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: 8.333333333333332%;
  }
}
```

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";
```

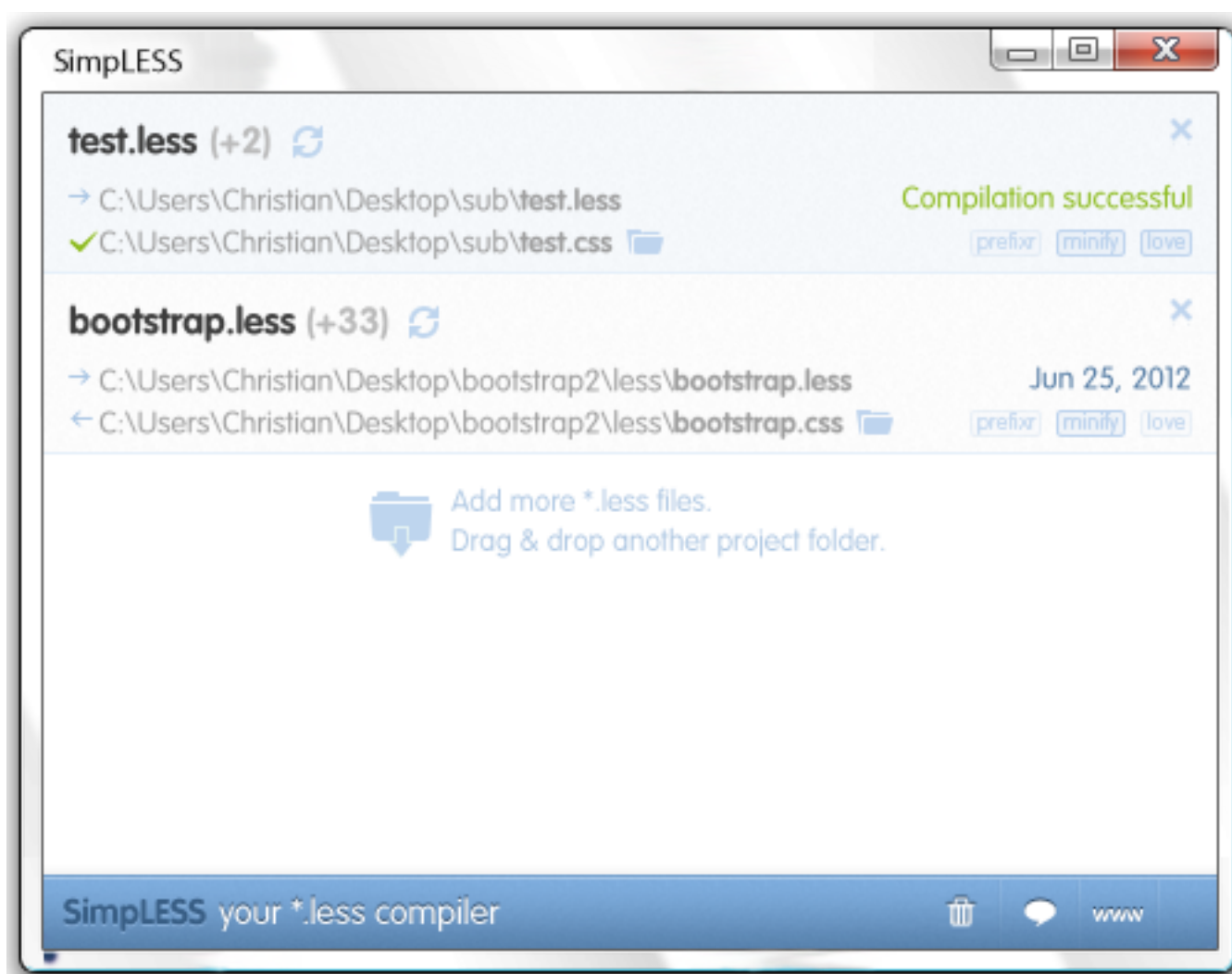
```
// Core CSS
@import "forms.less";

// Components
@import "navbar.less";

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


در این مطلب، نحوه‌ی استفاده از افزونه‌ی Web Essentials جهت پردازش فایل‌های LESS را بررسی می‌کنیم. پیش‌تر مطالبی را در رابطه با [CSS pre-processor](#) ها مطالعه کرده‌اید، [LESS](#) نیز یک CSS pre-processor است، یا در واقع بهتر است بگوئیم یک زبان جهت پویا کردن CSS می‌باشد که در سال 2009 توسط Alexis Sellier به صورت سورس باز [ایجاد شد](#). یکی از خصوصیات جالب LESS نسبت به دیگر CSS preprocessors، قابلیت کامپایل فایل‌های CSS به صورت real-time از طریق مرورگر توسط LESS.js می‌باشد.

روش‌های استفاده از LESS در دات نت 1- استفاده از SimpleLESS تمام تغییرات فایل‌های CSS را مشاهده می‌کند و در نهایت به صورت خودکار آنها را به CSS کامپایل و همچنین minify می‌کند. این روش مستقل از دات نت است و برای هر فایل LESSی جوابگو خواهد بود.



2- استفاده از .less{}

Dotless یک پیاده سازی از کتابخانه جاوا اسکریپتی LESS برای دات نت می‌باشد. پکیج نیوگت Dotless را نیز می‌توانید از [اینجا](#) دریافت کنید. بعد از اضافه شدن فایل‌های آن، یک ارجاع به dotless.core به پروژه تان اضافه خواهد شد. همچنین در فایل Web.Config در قسمت HttpHandler خط زیر اضافه خواهد شد:

```
<add type="dotless.Core.LessCssHttpHandler,dotless.Core" validate="false" path="*.LESS" verb="*" />
```

خط فوق یعنی به محض مواجه شدن با فایل LESS، پردازشگر فایل‌های LESS وارد عمل می‌شود. همچنین خط زیر نیز جهت پیکربندی به قسمت configSections در فایل Web.Config اضافه می‌شود:

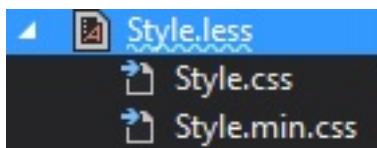
```
<section name="dotless"
type="dotless.Core.configuration.DotlessConfigurationSectionHandler,dotless.Core" />
```

همچنین اگر مایل بودید می‌توانید تنظیمات مربوط به فشرده سازی و caching را نیز فعال کنید:

```
<dotless minifyCss="false" cache="true" />
```

3- استفاده از افزونه‌ی Web Essentials

[Web Essentials](#) برای کامپایل فایل‌های LESS از کامپایلر node استفاده می‌کند. کار با این افزونه خیلی ساده است. کافی است پسوند فایل CSS موجود در پروژه تان را درون ویژوال استودیو، به less تغییر دهید. با دوبار کلیک بر روی فایل، ویرایشگر فایل‌های LESS برای شما نمایش داده می‌شود، همزمان نیز فایل یک فایل CSS و یک نسخه از فایل CSS را به صورت فشرده، برایتان تولید می‌کند. خب، هر بار که فایل LESS را تغییر دهید، Web Essentials به صورت خودکار فایل‌های css و min.css را برایتان روز رسانی می‌کند.



خوب با کلیک بر روی فایل less، ویرایشگر فایل‌های less نمایش داده می‌شود که با تغییر فایل css می‌توانید پیش نمایش آنرا در سمت راست مشاهده کنید:



تعریف متغیر

با استفاده از syntax زیر می‌توانید متغیرهای خود را تعریف کنید:

```
@variable-name: variableValue;
```

یکی از قابلیت‌های جالب در حین مقداردهی متغیرها به خصوص زمانی که مقدار یک کد رنگی باشد، نمایش کادر انتخاب رنگ است، این کادر بلافاصله بعد از نوشتن علامت # در ابتدای مقدار متغیر نمایش داده می‌شود:



به طور مثال با تعریف متغیر فوق هر جایی می‌توانیم برای تعیین رنگ از آن استفاده کنیم:

```
@primary-color: #ff6a00;
body
{
  background-color: @primary-color;
}
```

استفاده از توابع

LESS شامل تعداد زیادی [توابع](#) از پیش نوشته شده است که می‌توانید به راحتی از آنها استفاده کنید، توابعی از جمله کار با رنگ ها، اعمال ریاضی و غیره. استفاده از آنها خیلی ساده است. به طور مثال در کد زیر از تابع `percentage` جهت تبدیل 0.5 به 50% استفاده کرده ایم:

```
.myClass
{
  width: percentage(0.5);
}
```

استخراج یک فایل

یکی دیگر از قابلیت‌های Web Essentials استخراج (Extract) یک فایل می‌باشد به طور مثال فایل LESS شما شامل متغیرهای زیر است:

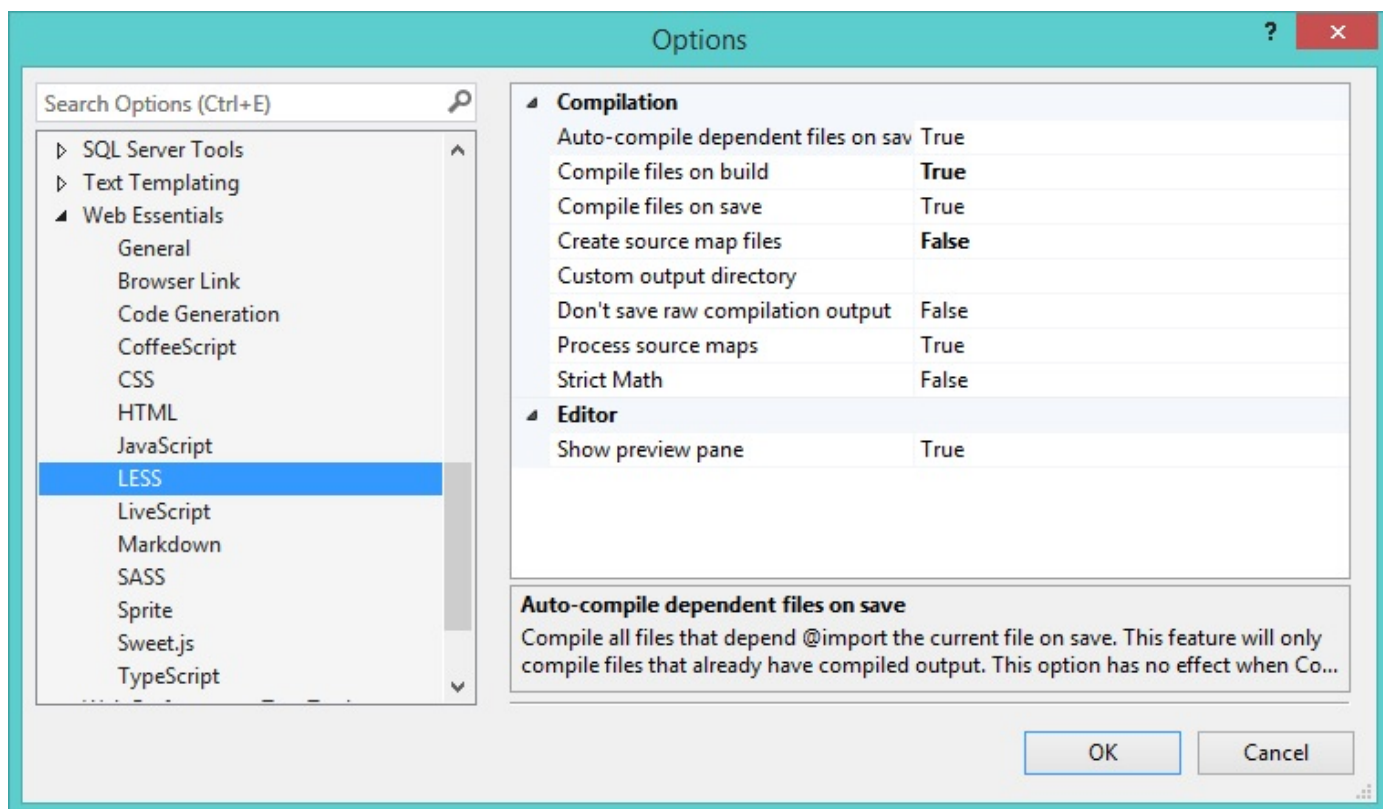
```
@primary-color: #7BA857;
@primary-color-light: #B6DE8F;
@primary-color-lighter: #D3EFC3;
@primary-color-lightest: #EFFAE6;
@secondary-color: #AE855C;
@text-color-light: #666666;
@text-color-dark: #0444;
```

به راحتی می‌توانید تعاریف فوق را درون یک فایل LESS دیگر با نام `colors.less` قرار دهید:



تغییر تنظیمات پیش فرض Web Essentials

افزونه Web Essentials دارای یک قسمت جهت تغییر تنظیمات پیش فرض برای کار با LESS می‌باشد که با مراجعه به منوی Tools در ویژوال استودیو و سپس Options می‌توانید آنها را تغییر دهید:



Auto-compile dependent files on save: توسط این گزینه می‌توانیم تعیین کنیم که فایل‌های که import کرده ایم تنها در صورتی که تغییر کرده و ذخیره شده باشند، در فایل CSS جاری کامپایل شوند.

Compile files on build: توسط این گزینه می‌توانیم تعیین کنیم که فایل‌های less در زمان Build پروژه کامپایل شوند.

Compile files on save: توسط این گزینه می‌توانیم تعیین کنیم که فایل‌های less در زمان ذخیره کردن پروژه کامپایل شوند.

Create source map files: اگر این گزینه True باشد [فایل map](#) نیز تولید خواهد شد.

Custom output directory: اگر می‌خواهید خروجی در پوشه‌ی موردنظر شما نمایش داده شود می‌توانید آدرس آن را تعیین کنید.

Don't save raw compilation output: با فعال بودن این گزینه فایل CSS عادی ایجاد نخواهد شد.

Process source maps: توسط این گزینه می‌توانید قابلیت‌های ویرایشگر فایل‌های source map را فعال یا غیرفعال کنید.

Strict Math: با فعال بودن این گزینه LESS تمام اعمال ریاضی درون فایل CSS را پردازش خواهد کرد.

Show preview pane: از این گزینه نیز جهت نمایش یا عدم نمایش preview window استفاده می‌شود.

نظرات خوانندگان

نویسنده: محمدرضا برنتی
تاریخ: ۴:۳۷ ۱۳۹۳/۰۴/۱۲

در قسمت روش‌های استفاده از less جای کامپایل با دستور زیر در CMD :

```
lessc style.less > style.css
```

و کامپایل توسط جاوا اسکریپت که به نظر من بهترین راه برای مشاهده تغییرات جزئی در فایل less پروژه قبل از کامپایل نهایی است خالی مانده .
برای کامپایل با جاوا اسکریپت :

```
<link rel="stylesheet/less" type="text/css" href="content/style.less" />  
<script src="Script/less.min.js"></script>
```

در خط اول مقدار خاصیت href برابر آدرس فایل less و src خط دوم شامل آدرس فایل کامپایلر جاوا اسکریپت بهینه شده است .
لینک دانلود کامپایلر جاوا اسکریپت: less.min.js

نویسنده: سیروان عقیفی
تاریخ: ۹:۰۲ ۱۳۹۳/۰۴/۱۲

هنگام کار با LESS در Visual Studio توسط افزونه‌ی Web Essentials نیازی به این کار نیست، شما کافی است فایل کامپایل شده CSS را داخل Layout تان قرار دهید بعد از اینکار در هر بار تغییر فایل less و کامپایل مجدد آن (چه در حالت build و چه در حالت save) فایل اضافه شده به Layout به صورت خودکار به روز رسانی می‌گردد.

نویسنده: محمدرضا برنتی
تاریخ: ۱۲:۳۹ ۱۳۹۳/۰۴/۱۲

این افزونه با هیچ کدام از روش‌های نصبی که تو سایت Web Essentials پیشنهاد شده بود روی vs 2013 من نصب نشد و با پیغام زیر متوقف شد .

This extension is not installable on any currently installed products.

نویسنده: وحید نصیری
تاریخ: ۱۲:۵۸ ۱۳۹۳/۰۴/۱۲

[اینجا](#) بحث شده. برای نصب آخرین نگارش آن، ابتدا [به روز رسانی دوم VS 2013](#) را نصب کنید. سپس از طریق منوی نصب افزونه‌های داخل خود VS.NET و نه دریافت مستقیم فایل vsix آن اقدام به نصب کنید.