

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

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

## قابلیت‌های موجود در SASS :

### Variables 1-

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

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

### Nesting 2- یا selector های تو در تو:

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

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

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

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

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

### 3- Partials :

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

برای استفاده از myPartial.scss\_ در فایل sass دیگر ، از دایرکتیو @import استفاده کنید:

```
@import "myPartial"
```

همچنین می توانید نام چندین partial را پشت هم بنویسید:

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

نمونه کد:

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

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

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

کدها بعد از تولید در مرورگر:

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

### 4- Mixins :

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

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

نمونه کد:

```
mixin ایجاد:

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

mixin استفاده از:

.box { @include border-radius(10px); }
```

## 5- Extend/Inheritance :

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

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

.success {
  @extend .message;
  border-color: green;
}
```

کدها بعد از تولید شدن به صورت زیر دیده می‌شوند:

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

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

## 6- Operators :

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

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

## نصب SASS :

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

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

روش دیگر استفاده از command line می‌باشد:

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

```
gem install sass
```

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

```
sudo gem install sass
```

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

```
sass -v
```

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

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

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

### : SassScript

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

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

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

[منبع](#)

## نظرات خوانندگان

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

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

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

از توضیح مفید و تکمیل کننده جنابعالی متشکرم.

نویسنده: ارش م  
تاریخ: ۸:۴۳ ۱۳۹۳/۰۸/۲۶

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

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

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

## SASS چیست؟

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

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

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

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

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

با استفاده از روش SCSS. (روش جدید)

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

با استفاده از روش SASS. (روش قدیم)

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

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

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

**کامپایل کردن SASS** روش‌های مختلفی برای کامپایل فایل‌های SASS وجود دارند:

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

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

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

```
npm install node-sass
```

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

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