

## 1: The Beginner's Guide to Learning Sass

*In this beginner guide we take you through the first steps of getting started with Sass and Compass. We walk you through installation, creating a test project, compiling your first lines Sass to CSS and we even "mixin" a little Sass history.*

Sass is actually written in a type of language called SCSS. There are others, like Less and Stylus. However Sass is the most powerful and has the biggest following in the web design community. To put it simply Sass is a CSS preprocessor. Sass basically brings that into normal CSS. Sass is written in a way that can be compiled or converted if you like into normal CSS. How Setup Sass Create a new folder where your project files will be stored. Next create a folder called assets. Inside assets create two more folders called css and scss. All you need to do is type cd, a space, then copy and paste the project directory from the assets folder. Next you need to tell Sass to watch your project, we can do this with: So basically, our main. Variables Before we actually write the variables we need to understand some best practices for variable naming. Instead name them by the function they do, for example: Extends and Mixins This can be very confusing for someone new to Sass. You might also like: An Extend An extend is simply a pre-written piece of code that stops you having to write the same tedious lines of repetitive code over and over again. You can write an Extend like this: So instead of having two separate Mixins we just change the color when calling the Mixin itself. Organising Project Files One of the best things about Sass is being able to structure your project however you like. This is how a simple project structure may look: The underscore makes that file a partial file, which can then be imported. So we turn ten different files into one. Conclusion and Further Reading SCSS in general is one of the most powerful tools front-end developers can have in their arsenal and when you incorporate Sass as your preprocessor of choice then your workflow will increase dramatically. Check out these great sites and tutorials to learn more about Sass and how to get the most from it: The Sass Way " A tutorial site focused completely on Sass. Probably one of the best front-end architects around. Handy Sass Mixins " Some great snippets you can grab and use on any project.

## 2: Learn LESS in 10 Minutes (or Less) - Tutorialzine

*Prerequisite. Before learning SASS, you must have the basic knowledge of HTML and CSS. Audience. Our SASS tutorial is designed to help beginners and professionals.*

It takes a lot of effort and discipline to keep your CSS maintainable. Luckily, the web development community has solved this problem. They give us a number of benefits over plain CSS: Variables, so that you can define and easily change values throughout your stylesheet. This is actually coming to CSS some day. In CSS we recently got calc , but it is only for length units. Mixins, which enable you to reuse and combine styles. They even support passing arguments. Functions, which give you a number of handy utilities for manipulating color, converting images to data-uris and more. The negative aspect is that if you use one of these pre-processors, you will need to compile your stylesheets down to regular CSS so that it works in browsers. You can include less. The recommended way is to compile your less stylesheets ahead of time and deploy a regular CSS file online. There are also a number of free graphical programs that can compile. If you have node installed, and you know what a terminal is, go ahead and open one. Then install less using NPM: With the above line, our code will be transformed to plain CSS in styles. All that is left is to link this css file our HTML. If there was a compilation mistake, it will show up in your terminal. Variables One of the main features of Less is the ability to create variables just like you would in a standard programming language. They can store any type of value you find yourself using frequently: The philosophy of less is to reuse the CSS syntax where possible. Here, we define two variables, one for background color and one for text color, both containing hexadecimal codes. Switch between the tabs to see the translated to CSS version of the code: For more about variables in Less read here. The following example will clear things up: In the following example we add an argument for the width and height of our circles, with a default value of 25 pixels. This will create a small circle 25x25 and a big one x pixels. Nesting and Scope Nesting can be used to structure your stylesheet in a way that matches the HTML structure of the page, while reducing the chance of conflicts. Here is an example of an unordered list and its children: Operations You can do basic math operations to numerical values and colors. Lets say we want to have two divs placed next to each other, the second one being twice as wide and with a different background. Functions LESS has functions too! There are a lot of other useful functions for manipulating colors, detecting the size of images and even embedding assets as data-uri in the stylesheet. See the full list of functions here. Further reading You now know enough of Less to get started! Every CSS file is a valid Less stylesheet, so you can start cleaning up that old and unwieldy. As you learn more, you will be able to make the code even better. Here is what we recommend that you read next:

## 3: Sass for Beginners – Inspirational Pixels

*Writing a lot of CSS can be overwhelming that is why it is beneficial to learn Sass and LESS. They can make any web developer and designer's life much easier. For a beginner, you might find it fine but as time goes by and your CSS skills are improving. You begin to wonder if there is a way so.*

Using tools such as Sass , Less , Stylus or PostCSS makes large and complicated stylesheets clearer to understand and easier to maintain. Thanks to features like variables, functions and mixins the code becomes more organized, allowing developers to work quicker and make less mistakes. Getting Started Sass files cannot be interpreted by the browser, so they need compiling to standard CSS before they are ready to hit the web. Here you have a couple of options: The simplest solution is a browser tool for writing and compiling Sass right on the spot - SassMeister. Use a 3rd party desktop app. Both free and paid versions are available. You can go here to find out more. If you are a CLI person like we are, you can install Sass on your computer and compile files manually. There are many other wrappers as well, but since we love Node. Here is how you can compile. They both do the same things, just are written in different ways. SCSS is the newer one and is generally considered better, so we are going to go with that. If you want more information on the difference between the two, check out this great article. Variables Variables in Sass work in a similar fashion to the those in any programming language, including principals such as data types and scope. When defining a variable we store inside it a certain value, which usually is something that will often reoccur in the CSS like a palette color, a font stack or the whole specs for a cool box-shadow. Below you can see a simple example. Mixins You can think of mixins as a simplified version of constructor classes in programming languages - you can grab a whole group of CSS declarations and re-use it wherever you want to give and element a specific set of styles. Mixins can even accept arguments with the option to set default values. In the below example we define a square mixin, and then use it to create squares of varying sizes and colors. This works similarly to the mixins system, but is preferred when we want to create a logical connection between the elements on a page. Extending should be used when we need similarly styled elements, which still differ in some detail. For a quick example, lets style a list containing a number of links: Operations With Sass you can do basic mathematical operation right in the stylesheet and it is as simple as applying the appropriate arithmetic symbol. Functions Sass offers a long list of built-in functions. They serve all kinds of purposes including string manipulation, color related operations, and some handy math methods such as random and round. Sass supports flow control as well, so if you want to, you can create quite complex behaviors. Conclusion Some of the above features are coming to standard CSS in the future, but they are not quite here yet. In the meantime, pre-processors are a great way improve the CSS writing experience and Sass is a solid option when choosing one. We only covered the surface here, but there is a lot more to Sass than this. If you want to get more familiar with everything it has to offer, follow these links:

## 4: Guides and Tutorials

*The Best collection of Sass tutorials and resources for web developers. you can Master in sass by following these tutorials and resources. SASS is a powerful javascript preprocessor for CSS; SASS extends the CSS functionality by adding variables, mixins and nested rules.*

If you want to just browse here, go ahead, but we recommend you go install Sass first. Go here if you want to learn how to get everything setup. Preprocessing CSS on its own can be fun, but stylesheets are getting larger, more complex, and harder to maintain. This is where a preprocessor can help. Once you start tinkering with Sass, it will take your preprocessed Sass file and save it as a normal CSS file that you can use in your website. The most direct way to make this happen is in your terminal. For example, running `sass input`. You can also watch individual files or directories with the `--watch` flag. The watch flag tells Sass to watch your source files for changes, and re-compile CSS each time you save your Sass. If you wanted to watch instead of manually build your input. Variables Think of variables as a way to store information that you want to reuse throughout your stylesheet. Be aware that overly nested rules will result in over-qualified CSS that could prove hard to maintain and is generally considered bad practice. This is a great way to organize your CSS and make it more readable. Partials You can create partial Sass files that contain little snippets of CSS that you can include in other Sass files. This is a great way to modularize your CSS and help keep things easier to maintain. A partial is simply a Sass file named with a leading underscore. The underscore lets Sass know that the file is only a partial file and that it should not be generated into a CSS file. Sass partials are used with the `import` directive. Sass is smart and will figure it out for you. A mixin lets you make groups of CSS declarations that you want to reuse throughout your site. You can even pass in values to make your mixin more flexible. A good use of a mixin is for vendor prefixes. After you create your mixin, you can then use it as a CSS declaration starting with `include` followed by the name of the mixin. Using `extend` lets you share a set of CSS properties from one selector to another. It helps keep your Sass very DRY. A placeholder class is a special type of class that only prints when it is extended, and can help keep your compiled CSS neat and clean. This helps you avoid having to write multiple class names on HTML elements. Operators Doing math in your CSS is very helpful. Operations in Sass let us do something like take pixel values and convert them to percentages without much hassle.

## 5: Best Sass tutorials : A comprehensive guide to Sass

*SASS (Syntactically Awesome Stylesheet) is a CSS pre-processor, which helps to reduce repetition with CSS and saves time. It is more stable and powerful CSS extension language that describes the style of document structurally. This tutorial covers the basics of SASS. This tutorial will help both.*

## 6: Sass: Sass Basics

*Here's the tutorial I wish I'd had as a beginner. This easy walkthrough is for Mac and Windows users only. I doubt that's a problem though, as the type of folks who use another OS probably wouldn't have any trouble installing Sass.*

## 7: CSS Tutorials | [www.enganchecubano.com](http://www.enganchecubano.com)

*Sass Tutorial for Beginners: Lecture 52, Web Design for Beginners Course - Duration: Flexbox Tutorial (CSS): Real Layout Examples - Duration: LearnWebCode , views.*

## 8: Gulp js Tutorial For Beginners

*Before You Start Reading: If you're a beginner in Sass, this detailed tutorial is for you. I really hope you enjoy it, I spent a*

*few days working on it to create a compelling and useful message.*

### 9: Learn Sass - Best Sass Tutorials | [www.enganchecubano.com](http://www.enganchecubano.com)

*For example, running `sass www.enganchecubano.com www.enganchecubano.com` from your terminal would take a single Sass file, `www.enganchecubano.com`, and compile that file to `www.enganchecubano.com`. You can also watch individual files or directories with the `--watch` flag.*

*Psycho-therapy to go Deathwatch the outer reach Dark places book Modern transportation and Atlantic express tracks Gardner western art northern european high renaissance 30 Days to Successful Fundraising (Psi Research Success Library) Where God and Science Meet Kenneth waltz theory of international politics chapter 5 The Animal Rescue Club (I Can Read Book 4) Breast MR Imaging, An Issue of Magnetic Resonance Imaging Clinics (The Clinics: Radiology) The Chemical Synthesis of Natural Products Kandarya Mahadeva, India Beneath the Plastic Christianizing community life Koda tadashi manga eyes DERBYSHIRE (The Hidden Places) The Ottoman Empire and its successors, 1801-1922. Extending reading power through writing Illegal entrants in Australia: Balancing control and compassion Technology in war Between the moon and the walking VI-3. Grover Cleveland and Rosa Pearl (Deal Gore 68 Inscribing community Steven M. Nolt Emil Fackenheim: theodicy, and the tikkun of protest David R. Blumenthal The Law of the Gospel Beyond the death of God, by C. Vahanian. Lpn to rn transitions 4th edition Power Up for Team Results 10 The millionaire mortgage broker To know Jesus is to know the Father, Son, and Spirit : the Trinity Careers Advisers Guide I Am Not Going to Read Any Words Today! Pmp project management professional exam study guide 5th edition Petroleum and Marine Technology Information Guide Semiotics 1982 (Semiotic Society of America Meeting//Semiotics) Total annual report 2015 Scripture tests of Christian discipleship Reflection mulan sheet music A First Course in Differential Equations (Undergraduate Texts in Mathematics) Publisher against the people*