* **Sass (Systematically Awesome Style Sheets)**

Sass stands for Systematically Awesome Style Sheets. It is a CSS pre-processor. It is an extension of CSS that is used to add power and elegance to the basic language. It facilitates you to add variables, nested rules, mixins, inline imports, inheritance and more, all with fully CSS-compatible syntax. Or 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 a document cleanly and structurally.

Sass (syntactically awesome stylesheets) is a [style sheet language](https://en.wikipedia.org/wiki/Style_sheet_language) initially designed by [Hampton Catlin](https://en.wikipedia.org/wiki/Hampton_Catlin) and developed by [Natalie Weizenbaum](https://en.wikipedia.org/w/index.php?title=Natalie_Weizenbaum&action=edit&redlink=1). After its initial versions, Weizenbaum and [Chris Eppstein](https://en.wikipedia.org/w/index.php?title=Chris_Eppstein&action=edit&redlink=1) continued to extend Sass with SassScript, a simple scripting language used in Sass files.

Sass is a [scripting language](https://en.wikipedia.org/wiki/Scripting_language) that is [interpreted](https://en.wikipedia.org/wiki/Interpreted_language) or [compiled](https://en.wikipedia.org/wiki/Compiled_language) into [Cascading Style Sheets](https://en.wikipedia.org/wiki/Cascading_Style_Sheets) (CSS). SassScript is the scripting language itself. Sass consists of two [syntaxes](https://en.wikipedia.org/wiki/Syntax_(programming_languages)). The original syntax, called "the indented syntax", uses a syntax similar to [Haml](https://en.wikipedia.org/wiki/Haml).[[4]](https://en.wikipedia.org/wiki/Sass_(stylesheet_language)#cite_note-main-4) It uses [indentation](https://en.wikipedia.org/wiki/Indent_style) to separate [code blocks](https://en.wikipedia.org/wiki/Block_(programming)) and [newline](https://en.wikipedia.org/wiki/Newline) characters to separate rules. The newer syntax, "SCSS", uses block formatting like that of CSS. It uses braces to denote code blocks and semicolons to separate lines within a block. The indented syntax and SCSS files are traditionally given the [extensions](https://en.wikipedia.org/wiki/Filename_extension) .sass and .scss, respectively.

[CSS3](https://en.wikipedia.org/wiki/CSS#CSS_3) consists of a series of selectors and pseudo-selectors that group rules that apply to them. Sass[[5]](https://en.wikipedia.org/wiki/Sass_(stylesheet_language)#cite_note-5) (in the larger context of both syntaxes) extends CSS by providing several mechanisms available in more traditional [programming languages](https://en.wikipedia.org/wiki/Programming_language), particularly [object-oriented languages](https://en.wikipedia.org/wiki/Object-oriented_programming), but that are not available to CSS3 itself. When SassScript is interpreted, it creates blocks of CSS rules for various selectors as defined by the Sass file. The Sass interpreter translates SassScript into CSS. Alternatively, Sass can monitor the .sass or .scss file and translate it to an output .css file whenever the .sass or .scss file is saved. Sass is simply [syntactic sugar](https://en.wikipedia.org/wiki/Syntactic_sugar) for CSS.

The official implementation of Sass is [open-source](https://en.wikipedia.org/wiki/Open-source_software) and coded in [Ruby](https://en.wikipedia.org/wiki/Ruby_(programming_language)); however, other implementations exist, including [PHP](https://en.wikipedia.org/wiki/PHP), and a high-performance implementation in [C](https://en.wikipedia.org/wiki/C_(programming_language)) called libSass. There's also a [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) implementation called JSass.[[9]](https://en.wikipedia.org/wiki/Sass_(stylesheet_language)#cite_note-9)Additionally, [Vaadin](https://en.wikipedia.org/wiki/Vaadin) has a Java implementation of Sass. The indented syntax is a metalanguage. SCSS is a [nested metalanguage](https://en.wikipedia.org/wiki/Metalanguage#Nested_metalanguage), as valid CSS is valid SCSS with the same [semantics](https://en.wikipedia.org/wiki/Semantics_of_programming_languages). Sass supports integration with the [Firefox](https://en.wikipedia.org/wiki/Firefox) [extension](https://en.wikipedia.org/wiki/Add-on_(Mozilla))[Firebug](https://en.wikipedia.org/wiki/Firebug_(web_development)). Sass Script provides the following mechanisms: [variables](https://en.wikipedia.org/wiki/Variable_(programming)), [nesting](https://en.wikipedia.org/wiki/Nesting_(computing)#In_programming), [mixins](https://en.wikipedia.org/wiki/Mixin), and selector  [inheritance](https://en.wikipedia.org/wiki/Inheritance_(computer_science)).

Sass is more stable and powerful CSS extension language that describes style of document cleanly and structurally. It is very useful to handle large style sheets by keeping them well organized and running quickly small style sheets.

## Features of Sass

* Sass is fully CSS-compatible.
* It is more stable, powerful and elegant than CSS.
* It is based on JavaScript and is superset of CSS.
* It has its own syntax and compiles to readable CSS.
* It is an open-source pre processor that is interpreted into CSS.
* It supports language extensions such as variables, nesting, and mixins.
* It provides many useful functions for manipulating colors and other values.
* It provides many advanced features like control directives for libraries.
* It provides well-formatted, customizable output.

# Why use Sass

Following are some reasons behind using Sass:

* Sass is a pre-processing language and it has its own syntax for CSS.
* It is easy, short and clean in a programming construct.
* It has some features that are used for creating awesome style sheets and facilitates writing code more efficiently and easy to maintain.
* It contains all the features of CSS along with some advance features.
* It provides document style presentation better than flat CSS.
* It facilitates reusability methods, logic statements and some of the built in functions like color manipulation, mathematics and parameter lists.
* It facilitates you to keep your responsive design project more organized.
* You don't need to repeat similar CSS again and again in your project.
* It is a super set of CSS, which means it contains all the features of CSS and is an open source pre-processor, coded in **Ruby**.

# Advantages and Disadvantages of Sass

## Advantages

* Sass facilitates you to write clean, easy and less CSS in a programming construct.
* It contains fewer codes so you can write CSS quicker.
* It is more stable, powerful, and elegant because it is an extension of CSS. So, it is easy for designers and developers to work more efficiently and quickly.
* It is compatible with all versions of CSS. So, you can use any available CSS libraries.
* It provides nesting so you can use nested syntax and useful functions like color manipulation, math functions and other values.

## Disadvantages

* The developer must have enough time to learn new features present in this preprocessor before using it.
* Using Sass may cause of losing benefits of browser's built-in element inspector.

# Sass vs SCSS

**Sass provides two distinct syntax:**

* Sass
* SCSS

Both are similar and do same thing, but written in different style. SCSS is latest one and considered better than Sass.

**SCSS:** SCSS provides the CSS friendly syntax to closing the gap between Sass and CSS. SCSS is called Sassy CSS.

**Advantage of Sass syntax:**

* The syntax of Sass is totally different from CSS but it is shorter and easier to type. You don't need to type semicolon or braces, even no need to use @mixin or @include, when a single character is enough: = and +.
* It also provides clean coding standards because it follows indented syntax.

**Advantage of SCSS syntax:**

* It is fully CSS compatible. You can rename a CSS file as .scss extension and it will also work. Due to the reason that it follows the syntax of CSS, makes it very easy to learn and work.
* 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.

**Sass** has two syntaxes. The new main syntax (as of **Sass** 3) is known as “**SCSS**” (for “Sassy **CSS**”), and is a superset of CSS3's syntax. This means that every valid CSS3 stylesheet is valid **SCSS** as well. **SCSS** files use the extension .**scss**. The second, older syntax is known as the indented syntax (or just “**Sass**”).

* **LESS**

LESS is a CSS pre-processor that enables customizable, manageable and reusable style sheet for website. LESS is a dynamic style sheet language that extends the capability of CSS. LESS is also cross browser friendly.

CSS Preprocessor is a scripting language that extends CSS and gets compiled into regular CSS syntax, so that it can be read by your web browser. It provides functionalities like *variables*, *functions*, *mixins* and *operations* that allow you to build dynamic CSS.

**Less** (Learner style sheet sometimes stylized as **LESS**) is a dynamic [style sheet language](https://en.wikipedia.org/wiki/Style_sheet_language) that can be compiled into [Cascading Style Sheets](https://en.wikipedia.org/wiki/Cascading_Style_Sheets)(CSS) and run on the client side or server side.[[2]](https://en.wikipedia.org/wiki/Less_(stylesheet_language)#cite_note-main-2) Designed by Alexis Sellier, Less is influenced by [Sass](https://en.wikipedia.org/wiki/Sass_(stylesheet_language)) and has influenced the newer "SCSS" syntax of Sass, which adapted its CSS-like block formatting syntax.[[3]](https://en.wikipedia.org/wiki/Less_(stylesheet_language)#cite_note-compare-3) Less is [open source](https://en.wikipedia.org/wiki/Open-source_software). Its first version was written in [Ruby](https://en.wikipedia.org/wiki/Ruby_(programming_language)); however, in the later versions, use of Ruby has been [deprecated](https://en.wikipedia.org/wiki/Deprecation) and replaced by [JavaScript](https://en.wikipedia.org/wiki/JavaScript). The indented syntax of Less is a [nested metalanguage](https://en.wikipedia.org/wiki/Metalanguage#Nested), as valid CSS is valid Less code with the same [semantics](https://en.wikipedia.org/wiki/Semantics_of_programming_languages). Less provides the following mechanisms: [variables](https://en.wikipedia.org/wiki/Variable_(programming)), [nesting](https://en.wikipedia.org/wiki/Nesting_(computing)#In_programming), [mixins](https://en.wikipedia.org/wiki/Mixin), [operators](https://en.wikipedia.org/wiki/Operator_(programming)) and [functions](https://en.wikipedia.org/wiki/Function_(computer_science)); the main difference between Less and other CSS precompilers being that Less allows real-time compilation via less.js by the browser.[[](https://en.wikipedia.org/wiki/Less_(stylesheet_language)#cite_note-main-2)

Less is a CSS pre-processor that facilitates you to customize, manage and reuse the style sheets for the webpage. Less is an extension of CSS and a dynamic style sheet language which can be run on client side or server side.

Less is an open source language. It is also cross browser compatible.

## Why LESS?

* Let us now understand why do we use LESS.
* LESS supports creating cleaner, cross-browser friendly CSS faster and easier.
* LESS is designed in JavaScript and also created to be used in *live*, which compiles faster than other CSS pre-processors.
* LESS keeps your code in modular way which is really important by making it readable and easily changeable.
* Faster maintenance can be achieved by the use of LESS *variables*.

## History

* LESS was designed by **Alexis Sellier** in 2009. LESS is an open-source. The first version of LESS was written in Ruby; in the later versions, the use of Ruby was replaced by JavaScript.

## Features

* Less is clean, compact, more readable code and written in a well organized way.
* Less supports cross-browser compatibility.
* Less is faster and easier.
* Less is written in JavaScript. It compiles faster than other preprocessor of CSS.
* Less provides variables which makes its maintenance faster.
* Less provides nesting which makes the code short, clean and organized in a specific way
* Less facilitates you to define styles which can be reused throughout the code.
* Less is an extension of CSS. So it is also called super set of CSS.
* Less is capable enough to sort out the problem of code redundancy.
* Less provides @import rule so you can easily handle external files. Importing is required because many people split their stylesheet in multiple files, rather than putting it in one file.
* Less provides extend method to groups selectors that share the same style-rules. It makes Less clean and more organized.
* Less provides merging property. It is the most exciting feature of Less that accepts multiple values like transform, transition, and box-shadow.
* LESS is an agile tool that sorts out the problem of code redundancy.

## Advantages

* Less is a CSS preprocessor. After compilation, it generates simple CSS which works across the browsers.
* Less supports cross-browser compatibility.
* Less codes are simple, clean and well organized because the use of nesting.
* Maintenance in Less can be achieved faster because the use of variables.
* Less provides a list of operators which makes coding faster and time saving.
* Less facilitates you to reuse the whole classes easily by referencing them in your rule-set.
* Less is new and preferred over the conventional CSS because may ease the lengthy styling.
* LESS provides the use of *operations* that makes coding faster and saves time.

## Disadvantages

* You must have to spend some time to learn Less if you are new to CSS preprocessing.
* Less provides fewer frameworks as compared to older preprocessor like SASS which contains frameworks like Compass, Gravity and Susy.
* In Less, there is a tight coupling between modules so it takes more effort to reuse and/or test dependent modules.

Both **Sass and LESS** are backward compatible so you can easily convert your existing **CSS** files just by renaming the .**css** file extension to .**less** or .**scss** , respectively. **LESS** is JavaScript based and **Sass** is Ruby based.

**Sass makes CSS fun again**. Sass is an extension of CSS, adding nested rules, variables, mixins, selector inheritance, and more. It's translated to well-formatted, standard CSS using the command line tool or a web-framework plugin.

**Sass Vs LESS**. **Sass** and **LESS** are both very powerful CSS extensions. ... Both **Sass** and **LESS**are backward compatible so you can easily convert your existing CSS files just by renaming the .css file extension to .**less** or .scss , respectively.

**Less Framework** is a CSS grid system for designing adaptive web sites. It contains 4 layouts and 3 sets of typography presets, all based on a single grid.

* **What is CSS Preprocessor**

CSS preprocessor is a scripting language that is an extension of CSS. It is compiled into regular CSS syntax and then the CSS is read by the web browser. Less looks very similar to CSS but it provides functionalities like variables, functions, mixins and operations etc that facilitates you to build dynamic CSS.

Less and SASS both are the example of CSS preprocessor because both are compiled and produce CSS syntaxes that are read by the web browser.

Less is written in JavaScript and requires either Node.js or web browser to run. You can include Less.js in your web site and it can compile all the linked .Less style sheets in real time, but this is slow and is not recommended.

* **Bootstrap:**

Twitter Bootstrap is the most popular front end framework in the recent time. It is sleek, intuitive, and powerful mobile first front-end framework for faster and easier web development. It uses HTML, CSS and Javascript.

**What is Twitter Bootstrap?**

Bootstrap is a sleek, intuitive, and powerful, mobile first front-end framework for faster and easier web development. It uses HTML, CSS and Javascript.

* Bootstrap is the most popular HTML, CSS and JavaScript framework for developing a responsive and mobile friendly website.
* It is absolutely free to download and use.
* It is a front-end framework used for easier and faster web development.
* It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many others.
* It can also use JavaScript plug-ins.
* It facilitates you to create responsive designs.

**History**

Bootstrap was developed by *Mark Otto* and *Jacob Thornton* at *Twitter*. It was released as an open source product in August 2011 on GitHub.

**Why Use Bootstrap?**

* **Mobile first approach** − Bootstrap 3, framework consists of Mobile first styles throughout the entire library instead them of in separate files.
* **Browser Support** − It is supported by all popular browsers.
* **Easy to get started** − With just the knowledge of HTML and CSS anyone can get started with Bootstrap. Also the Bootstrap official site has a good documentation.
* **Responsive design** − Bootstrap's responsive CSS adjusts to Desktops, Tablets and Mobiles. A website is called responsive website which can automatically adjust itself to look good on all devices, from smart phones to desktops etc.
* Provides a clean and uniform solution for building an interface for developers.
* It contains beautiful and functional built-in components which are easy to customize.
* It also provides web based customization.
* And best of all it is an open source.
* It facilitates users to develop a responsive website.
* It is compatible on most of browsers like Chrome, Firefox, Internet Explorer, Safari and Opera etc.

**What Bootstrap Package Includes?**

* **Scaffolding** − Bootstrap provides a basic structure with Grid System, link styles, and background.
* **CSS** − Bootstrap comes with the feature of global CSS settings, fundamental HTML elements styled and enhanced with extensible classes, and an advanced grid system.
* **Components** − Bootstrap contains over a dozen reusable components built to provide iconography, dropdowns, navigation, alerts, pop-overs, and much more.
* **JavaScript Plugins** − Bootstrap contains over a dozen custom jQuery plugins. You can easily include them all, or one by one.
* **Customize** − You can customize Bootstrap's components, LESS variables, and jQuery plugins to get your very own version.

**HTML5 doctype**

Bootstrap makes use of certain HTML elements and CSS properties that require the use of the HTML5 doctype. Hence include the below piece of code for HTML5 doctype at the beginning of all your projects using Bootstrap.

<!DOCTYPE html>

<html>

…

</html>

</html>

**Mobile First**

Since Bootstrap 3 has been launched, Bootstrap has become mobile first. It means 'mobile first' styles can be found throughout the entire library instead of them in separate files. You need to add the **viewport meta tag** to the **<head>** element, to ensure proper rendering and touch zooming on mobile devices

.

<meta name = "viewport" content = "width = device-width, initial-scale = 1.0">

*width* property controls the width of the device. Setting it to *device-width* will make sure that it is rendered across various devices (mobiles, desktops, tablets...) properly.

*initial-scale = 1.0* ensures that when loaded, your web page will be rendered at a 1:1 scale, and no zooming will be applied out of the box.

Add **user-scalable = no** to the **content** attribute to disable zooming capabilities on mobile devices as shown below. Users are only able to scroll and not zoom with this change, and results in your site feeling a bit more like a native application.

<meta name = "viewport" content = "width = device-width, initial-scale = 1.0, maximum-scale = 1.0, user-scalable = no">

Normally *maximum-scale = 1.0* is used along with *user-scalable = no*. As mentioned above **user-scalable = no** may give users an experience more like a native app, hence Bootstrap doesn't recommend using this attribute.

**Responsive Images**

Bootstrap 3 allows you to make the images responsive by adding a class **.img-responsive** to the **<img>** tag. This class applies **max-width: 100%;**and **height: auto;** to the image so that it scales nicely to the parent element

.

<img src = "..." class = "img-responsive" alt = "Responsive image">

**Typography and Links**

Bootstrap sets a basic global display (background), typography, and link styles −

**Basic Global display** − Sets *background-color: #fff;* on the *<body>*element.

**Typography** − Uses the *@font-family-base*, *@font-size-base*, and *@line-height-base* attributes as the typographic base.

**Link styles** − Sets the global link color via attribute *@link-color* and apply link underlines only on *:hover*.

If you intend to use LESS code, you may find all these within *scaffolding.less*.

**Normalize**

Bootstrap uses [Normalize](http://necolas.github.io/normalize.css/) to establish cross browser consistency.

Normalize.css is a modern, HTML5-ready alternative to CSS resets. It is a small CSS file that provides better cross-browser consistency in the default styling of HTML elements.

**Containers**

Use class **.container** to wrap a page's content and easily center the content's as shown below.

<div class = "container">

...

</div>

Take a look at the **.container** class in *bootstrap.css* file −

.container {

padding-right: 15px;

padding-left: 15px;

margin-right: auto;

margin-left: auto;

}

Note that, due to padding and fixed widths, containers are not nestable by default.

Take a look at *bootstrap.css* file −

@media (min-width: 768px) {

.container {

width: 750px;

}

}

Here you can see that CSS has media-queries for containers with **width**. This helps for applying responsiveness and within those the container class is modified accordingly to render the grid system properly.

Bootstrap is the popular HTML, CSS and JavaScript framework for developing a responsive and mobile friendly website.