



In this lesson we will focus on the Sass aspect of the project. As the CSS comprises a good portion of the development, it's important to take our time and build a good foundation that will save us time later. In addition, a clean and organised Sass file and folder structure will make website maintenance much easier in the future.

Let's begin by organizing our **Sass partial structure**. A Sass partial is, like the name says, a part that is imported into a whole. We'll have just one main SCSS file and this file will import many partials to form the final stylesheet. Sass will process all this into a single CSS file.

As with many things in web development there are various ways of proceeding with this kind of structure. The one I will use here has been used in several projects with success. Inside the sass folder located in src, let's create the following folders:

- base, for basic styles and the CSS reset.
- config, for the configuration file
- mixins, for mixins and functions
- modules, for reusable elements not tied to any particular page like the grid structure
- pages, for styles that are specific to a page or large section of the website
- structure, for structural elements like the main header and footer

The styles.scss file will remain at the sass folder. Let's now create the basic files that we will need. All partials start with the underscore character so that the preprocessor knows that it should not be converted alone to the final CSS folder.

In the base folder, create:

- *_reset.scss* for the CSS reset.
- *_typography.scss* for the general typography
- *_helper_classes.scss* for helper classes like clearfix, alignment classes, and others

In the config folder, create:

- *_config.scss* file containing configuration variables

In the mixins folder, create

- *_mixins_general.scss* for a general mixins
- *_mixins_breakpoints.scss* for mixins that will help us for the responsiveness of the page
- *_mixins_grids.scss* for mixins that will help us to generate various grid classes

In the modules folder create

- *_grids.scss*, containing the grid and responsive mixins to generate a responsive grid system

In the pages folder create

- *_blog.scss*

- *_welcome.scss*
- *_services.scss*
- *_portfolio.scss*
- *_contact.scss*

In the structure folder create

- *_footer.scss*
- *_header.scss*
- *_sections.scss*

With this structure done, we need to import all the files into the styles.scss file. Remove any styles that are present there and place the following code:

```
//
// Personal Portfolio
// Main SCSS file
//

// load config
@import 'config/config';

// load mixins
@import 'mixins/mixins_general';
@import 'mixins/mixins_breakpoints';
@import 'mixins/mixins_grids';

// load base
@import 'base/reset';
@import 'base/helper_classes';
@import 'base/typography';

// load structure
@import 'structure/header';
@import 'structure/sections';
@import 'structure/footer';

// load modules
@import 'modules/grids';

// load pages
@import 'pages/welcome';
@import 'pages/services';
@import 'pages/portfolio';
@import 'pages/blog';
@import 'pages/contact';
```

Add some variables to the config/config.scss file:

```
//
```

```
//-- Config  
//
```

```
// colors  
$color-base: #15263e;  
$color-base-light: #374362;  
$color-gradient1: #b119ec;  
$color-gradient2: #f8027e;  
$color-highlight: #a91aef;
```

```
// widths  
$site-width: 75em;
```

Variables in SaaS starts with the dollar sign and they work as you expect them to. You just use it as a value of some CSS property and SaaS substituting the value.

With that, we have our SaaS structure ready to use.