Sass Architecture Structure

<> **sassas.md**   Raw

Sass Architecture Structure

```
sass/
|
|- base/
|   |- _reset.scss        # Reset/normalize
|   |- _typography.scss   # Typography rules
|   ...                   # Etc…
|
|- components/
|   |- _buttons.scss      # Buttons
|   |- _carousel.scss     # Carousel
|   |- _cover.scss        # Cover
|   |- _dropdown.scss     # Dropdown
|   ...                   # Etc…
|
|- layout/
|   |- _navigation.scss   # Navigation
|   |- _grid.scss         # Grid system
|   |- _header.scss       # Header
|   |- _footer.scss       # Footer
|   |- _sidebar.scss      # Sidebar
|   |- _forms.scss        # Forms
|   ...                   # Etc…
|
|- pages/
|   |- _home.scss         # Home specific styles
|   |- _contact.scss      # Contact specific styles
|   ...                   # Etc…
|
|- sass-utils/
|   |- _variables.scss    # Sass Variables
|   |- _functions.scss    # Sass Functions
|   |- _mixins.scss       # Sass Mixins
|   |- _helpers.scss      # Class & placeholders helpers
|
|- vendors/
|   |- _bootstrap.scss    # Bootstrap
|   |- _jquery-ui.scss    # jQuery UI
|   ...                   # Etc…
|
|
`- style.scss            # Primary Sass file
```

BASE FOLDER

The `base/` folder holds what we might call the boilerplate code for the project. In there, you might find the reset file, some typographic rules, and probably a stylesheet, defining some standard styles for commonly used HTML elements.

COMPONENTS FOLDER

For smaller components, there is the `components/` folder. While `layout/` is macro (defining the global wireframe), `components/` is more focused on widgets. It contains all kind of specific modules like a slider, a loader, a widget, and basically anything along those lines. There are usually a lot of files in `components/` since the whole site/application should be mostly composed of tiny modules.

LAYOUT FOLDER

The `layout/` folder contains everything that takes part in laying out the site or application. This folder could have stylesheets for the main parts of the site (header, footer, navigation, sidebar...), the grid system or even CSS styles for all the forms.

PAGES FOLDER

If you have page-specific styles, it is better to put them in a `pages/` folder, in a file named after the page. For instance, it's not uncommon to have very specific styles for the home page hence the need for a `_home.scss` file in `pages/`.

SASS-UTILS FOLDER

The `sass-utils/` folder gathers all Sass tools and helpers used across the project. Every global variable, function, mixin and placeholder should be put in here.

The rule of thumb for this folder is that it should not output a single line of CSS when compiled on its own. These are nothing but Sass helpers.

VENDORS FOLDER

And last but not least, most projects will have a `vendors/` folder containing all the CSS files from external libraries and frameworks – Normalize, Bootstrap, jQueryUI, FancyCarouselSliderjQueryPowered, and so on. Putting those aside in the same folder is a good way to say "Hey, this is not from me, not my code, not my responsibility".

folder is a good way to say "Hey, this is not from me, not my code, not my responsibility".

If you have to override a section of any vendor, I recommend you have an 8th folder called `vendors-extensions/` in which you may have files named exactly after the vendors they overwrite.

For instance, `vendors-extensions/_boostrap.scss` is a file containing all CSS rules intended to re-declare some of Bootstrap's default CSS. This is to avoid editing the vendor files themselves, which is generally not a good idea.

STYLE FILE

The style file should be the only Sass file from the whole code base not to begin with an underscore. This file should not contain anything but `@import` and comments.

Files should be imported according to the folder they live in, one after the other in the following order:

1. `sass-utils/`
2. `vendors/`
3. `base/`
4. `layout/`
5. `components/`
6. `pages/`

Write   Preview

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Comment