

Eva Morrison
Sprint 2 Research
October 12, 2022

React

Useful Links

- Libraries: <https://reactlibraries.com/>
- Top react component libraries: <https://technostacks.com/blog/react-component-libraries/>
- React home page: <https://reactjs.org/>
- React basics and tutorial: https://www.w3schools.com/whatis/whatis_react.asp
- React Tutorial: <https://reactjs.org/tutorial/tutorial.html>

Background

- Declarative, efficient, and flexible JS library
- Compose complex UIs from small, isolated pieces called “components”

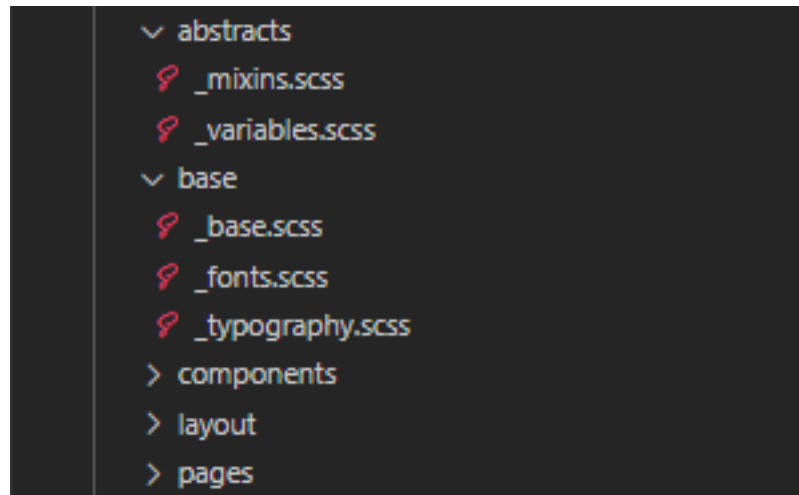
HTML and JavaScript

- HTML Tutorial: <https://www.w3schools.com/html/>
- JavaScript Homepage: <https://www.javascript.com/>
- JavaScript Tutorial: <https://www.w3schools.com/js/>

Front End

Best Practices

- Best practices website: <https://javascript.plainenglish.io/front-end-web-development-best-practices-from-my-experience-in-the-industry-4c5bbc04c9e3>
- Structuring style code
 - Inside the styles folder create separate folders for configurations, base, payout, components, and pages depending on your web application
 - Use separate typography file to define sizes and colors of all html text tags
 - Colors should be defined in a variables file and assign colors to variables with suitable names and use those variables when colors need to be applied through the whole website



-
- These files can then be imported into the main style file
- Main.scss

```

1 // 1. Configuration
2 @import
3   'abstracts/variables',
4   'abstracts/mixins';
5
6 // 2. Base stuff
7 @import
8   'base/base',
9   'base/fonts',
10  'base/typography';
11
12 // 3. Layout-related sections
13 @import
14   'layout/header',
15   'layout/banner',
16   'layout/grey-background',
17   'layout/footer';
18
19 // 4. Components
20 @import
21   'components/button',
22   'components/card',
23   'components/navs',
24   'components/form';
25
26 // 5. Page-specific styles
27 @import
28   'pages/home',
29   'pages/my-page-1',
30   'pages/my-page-2';

```

-
- Structuring JS Code
 - Separate your JS from other sources of JS
 - Plugins, vendors, and scripts are the main three folders
 - Plugins are useful code pieces that are used everywhere in the web app
 - jQuery plugins
 - vendors are 3rd party JS code
 - frameworks and libraries
 - scripts is code written for our web app logic

- Use a CSS Preprocessor
 - Don't write pure CSS
 - Always use a preprocessor like SCSS or Sass
- Don't keep SRC and HREF
 - Empty src and href cause unnecessary server requests
 - This wastes resources and bandwidth
- Avoid using CSS expressions
 - CSS expressions are frequently evaluated degrading performance
- Use a custom parent class for your custom styles
 - Use a custom parent class for when you're writing custom pages and components
 - Prevents accidentally overriding somebody else's styles
 - One parent class will be sufficient, avoid using nested
- Only use globally created responsive typography styles
- Avoid using global JS variables
- Add meaningful comments to your code
- Minifying JS and CSS
- Remove duplicate JS and CSS
- Refactor your code