



SCORE UI

Ben Cato & Emily Lord

# Course Overview

01

Review of  
Fundamentals

02

File Structure

03

Bootstrap 101  
(HTML/CSS/JS)

04

Sass 101

05

RequireJS

06

Component HTML

07

Class Selections

08

Where to Start/  
Design Analysis

09

SCORE vs. Custom  
Components

# Review of Fundamentals

01



# BRAINJOCKS<sup>TM</sup> SCORE Components

SCORE components are categorized as:

- Containers
- Content
- Navigation
- Social

# Class Selections

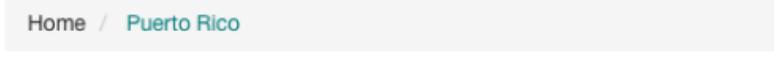
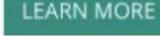
- Determined during design analysis phase
- Used for different variations of SCORE components
- Added by the content admin in the Selections folder
- Applied in the component styles menu

The image consists of two side-by-side screenshots from the Sitecore Content Editor.

**Left Screenshot:** Shows the Sitecore Content tree under the 'Content' folder. A red box highlights the 'Selections' folder under 'TrainingContent'.

**Right Screenshot:** Shows the properties for the 'Primary' item. The 'Value' field in the 'List Item' section contains 'btn-primary', and the 'Text' field contains 'Primary'. Both the 'Value' field and the 'Text' field are highlighted with red boxes.

# Component Styles Document

SCORE components:			
Component	Class Name	Class Value	Screenshot
Breadcrumb	Default		
Button	Default		
	Default <i>*when placed inside highlight/stylebox</i>		
	Primary	.btn-primary	
	Large	.btn-lg	

## Component Styles Guide



02

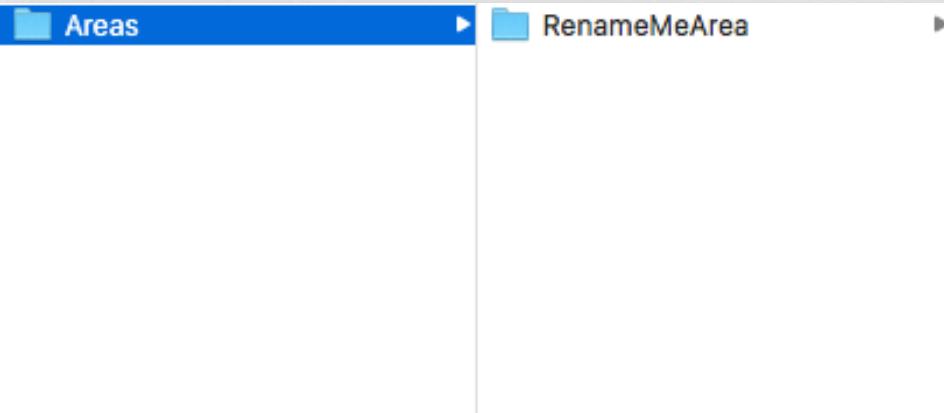
## Understanding SCORE's File Structure

Organization of project files,  
Sass files, and the HTML  
sandbox

# File Structure

## “Areas” Scaffolding

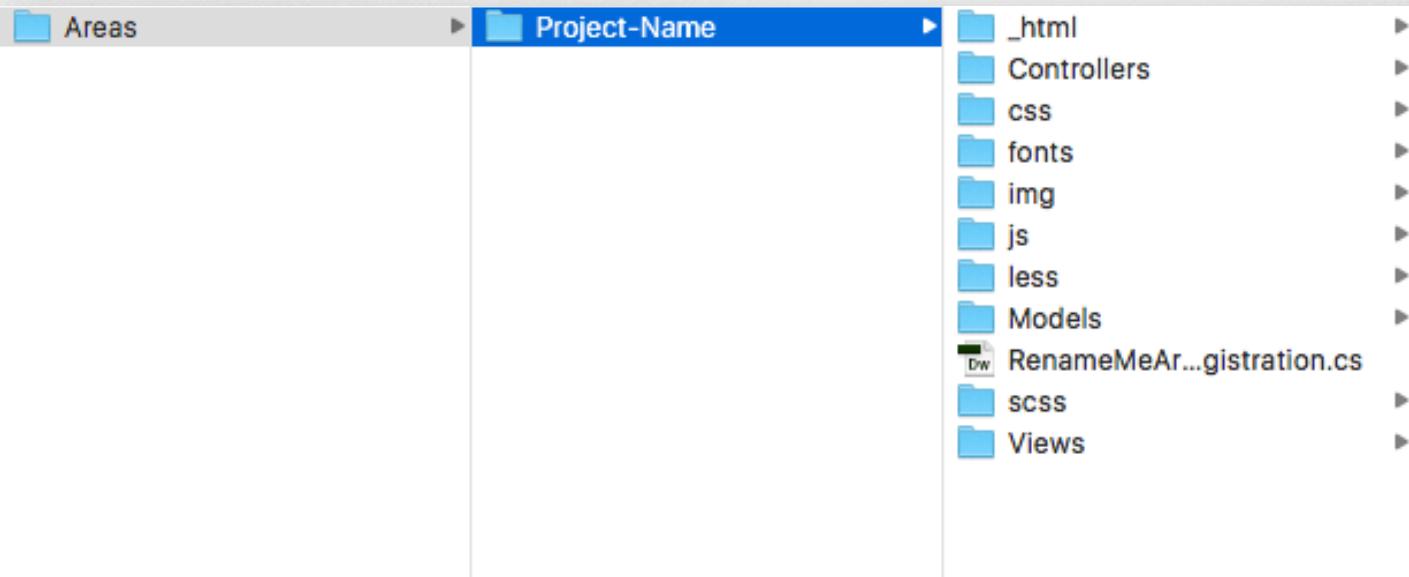
- All project files are *shared* between Sitecore and UI developers



# File Structure

“RenameMeArea” Folder

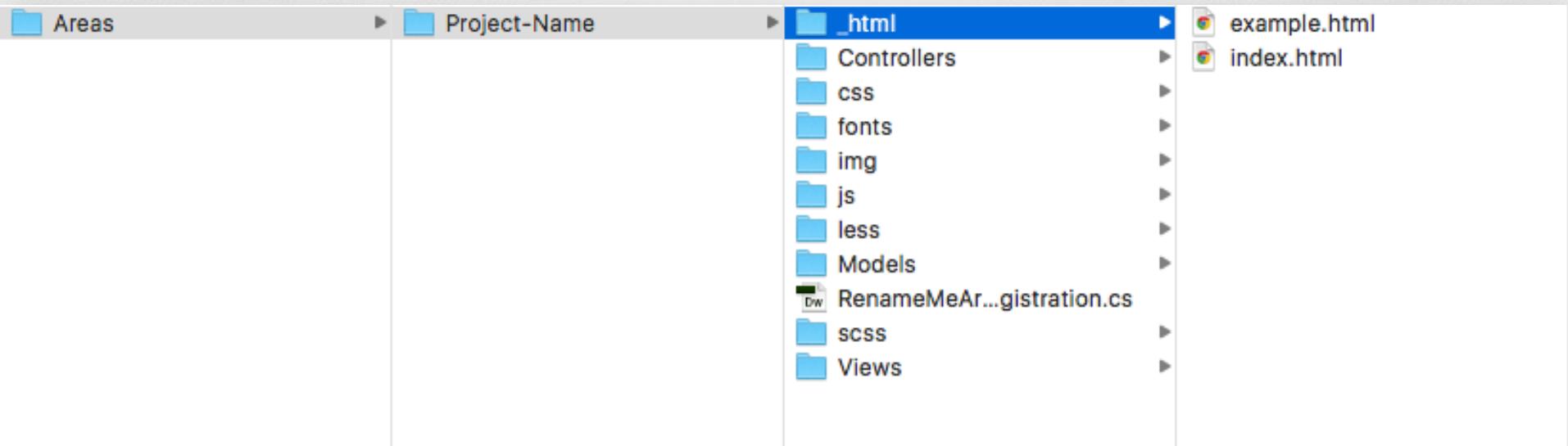
- Renamed to a specific project/site name



# File Structure

## HTML “Sandbox” folder

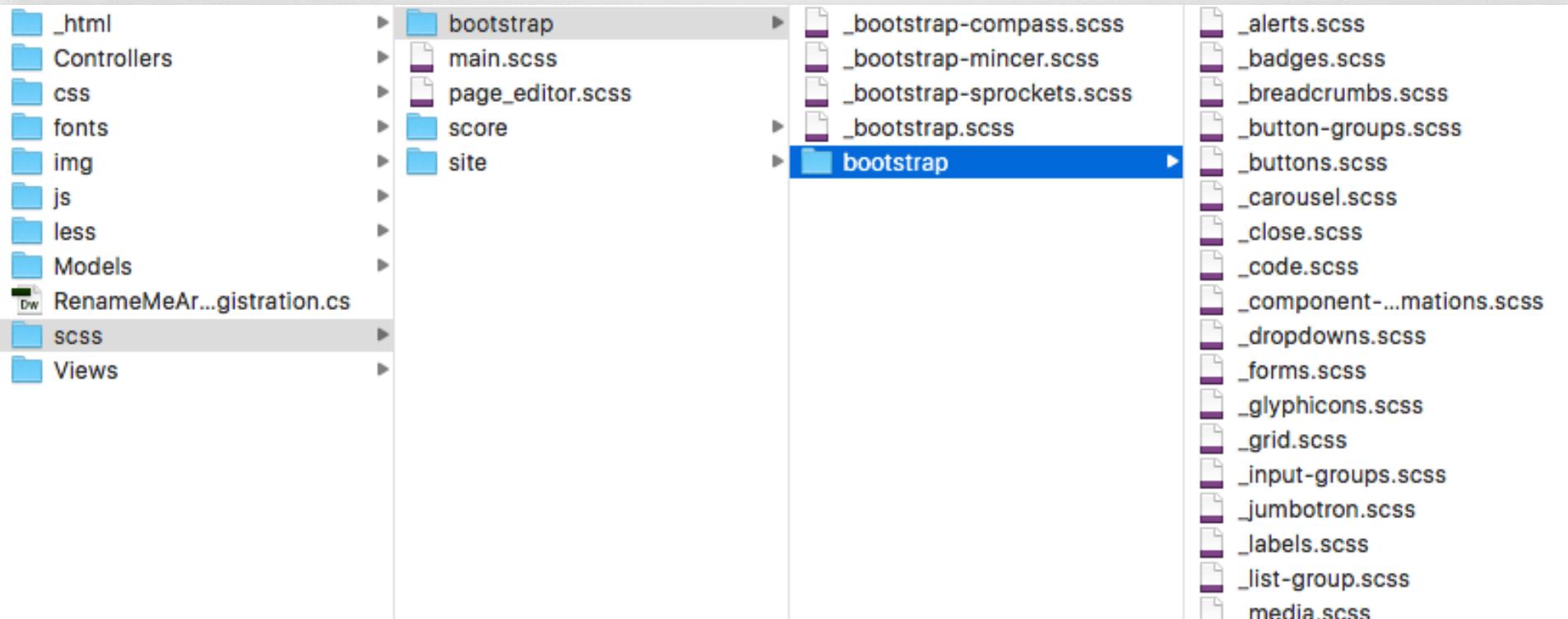
- Local HTML that is for your use - *Does not get deployed*
- example.html provided for “kitchen sink” examples
- index.html “boilerplate” provided for start-up



# File Structure

## Bootstrap Folder

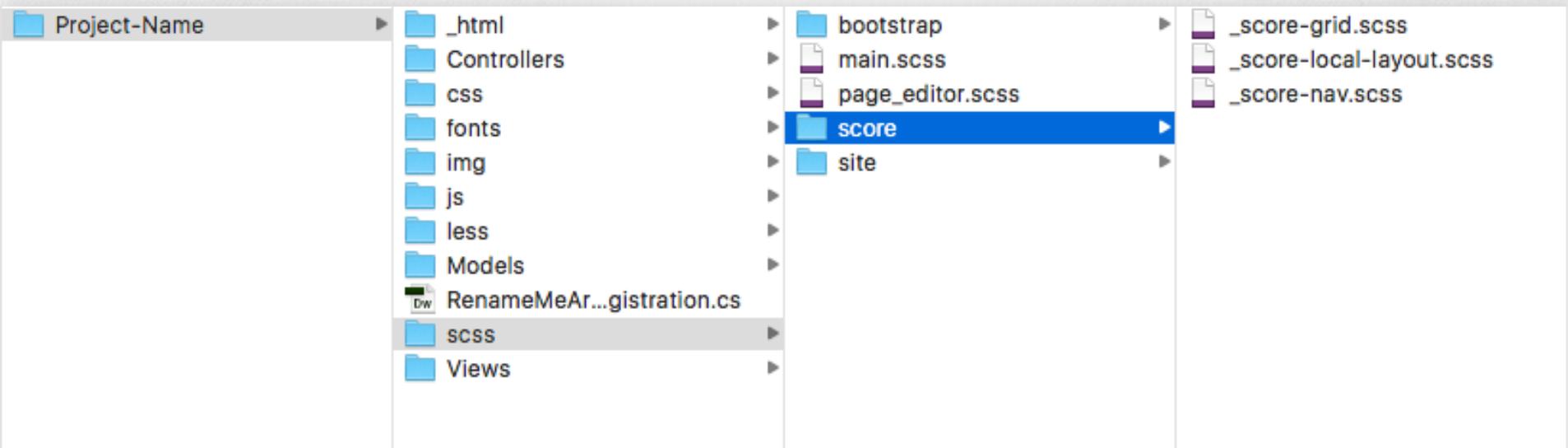
- Contains all of Bootstrap's default Sass files
- Location to access Bootstrap's variables and mixins (if needed)



# File Structure

## SCORE Folder

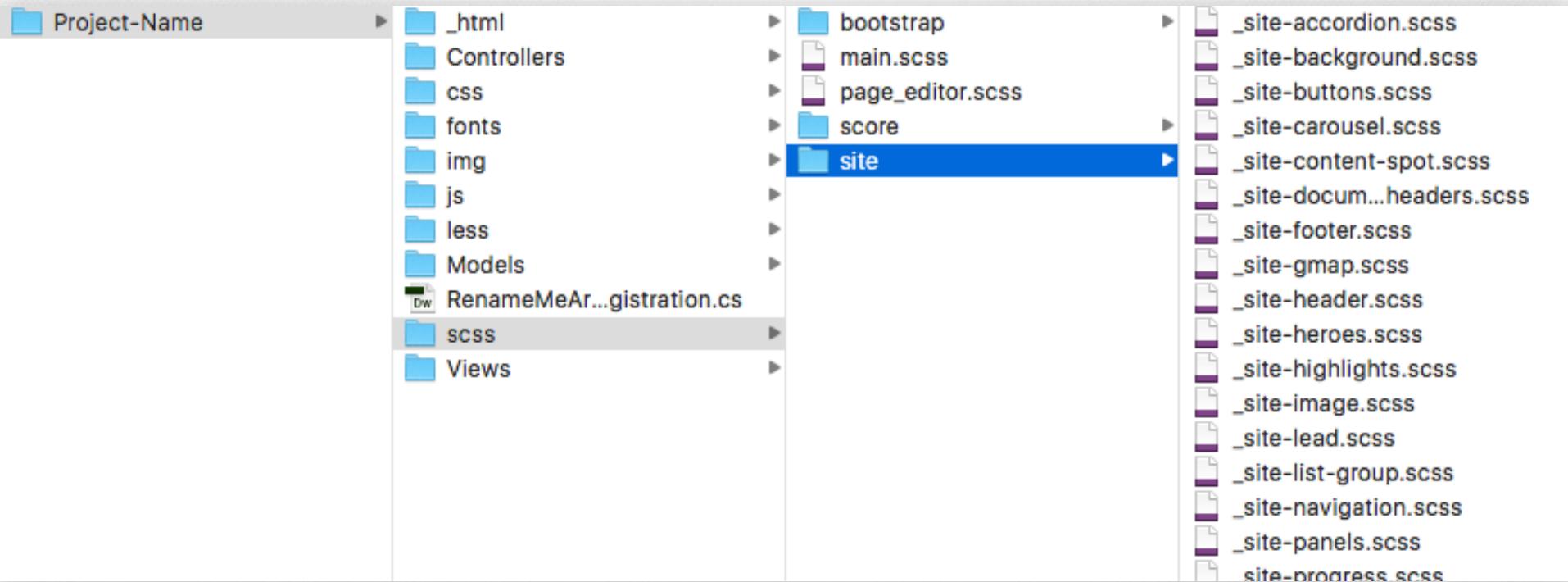
- Contains all of SCORE's default Sass files
- Location to modify SCORE-specific styles (if needed)



# File Structure

## Site Folder

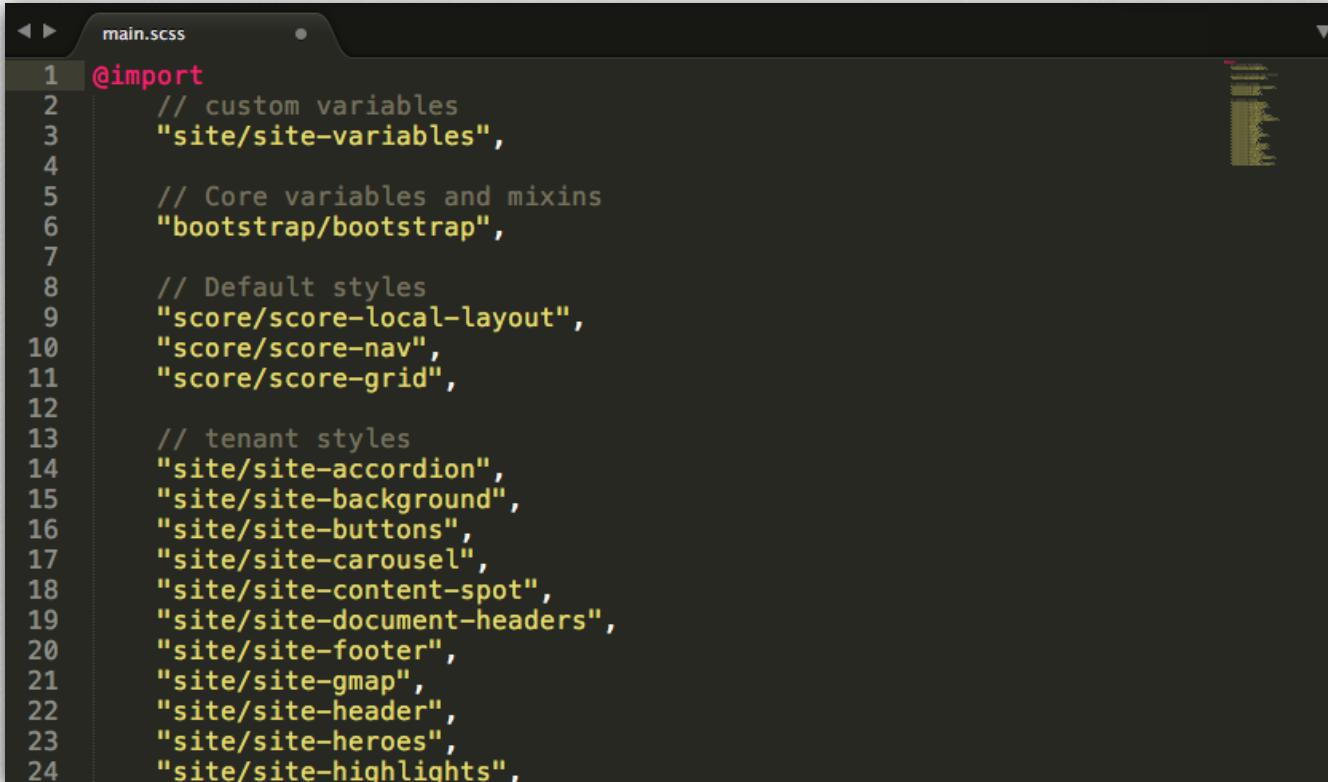
- Location to modify/include all site-specific styles
- Create site-specific variables and mixins



# File Structure

main.scss

- main.scss imports all Bootstrap, SCORE, and site Sass files
- Compiles to main.css



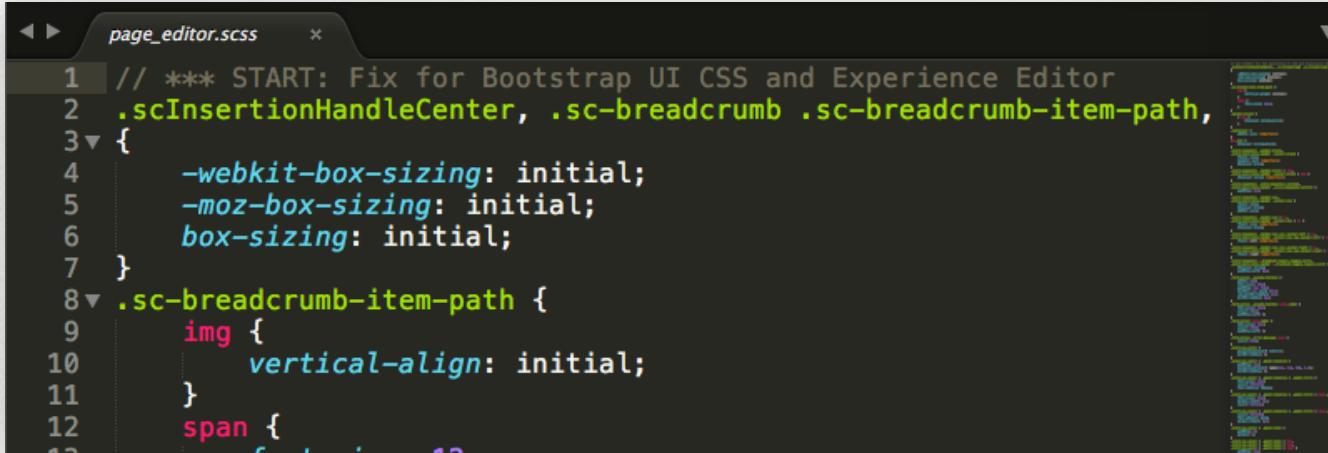
The screenshot shows a code editor window with the file 'main.scss' open. The code is a list of @import statements, each preceded by a line number from 1 to 24. The imports are categorized into three groups: custom variables, core variables and mixins, and tenant styles. The code is written in SCSS (Sass) syntax.

```
1 @import
2   // custom variables
3   "site/site-variables",
4
5   // Core variables and mixins
6   "bootstrap/bootstrap",
7
8   // Default styles
9   "score/score-local-layout",
10  "score/score-nav",
11  "score/score-grid",
12
13  // tenant styles
14  "site/site-accordion",
15  "site/site-background",
16  "site/site-buttons",
17  "site/site-carousel",
18  "site/site-content-spot",
19  "site/site-document-headers",
20  "site/site-footer",
21  "site/site-gmap",
22  "site/site-header",
23  "site/site-heroes",
24  "site/site-highlights".
```

# File Structure

page\_editor.scss

- Styles are specific to the SCORE experience editor
- Components may need to render differently in experience editor via CSS
- Components initially inherit main.css
- page\_editor.css *overrides* main.css and can include additional styles if needed



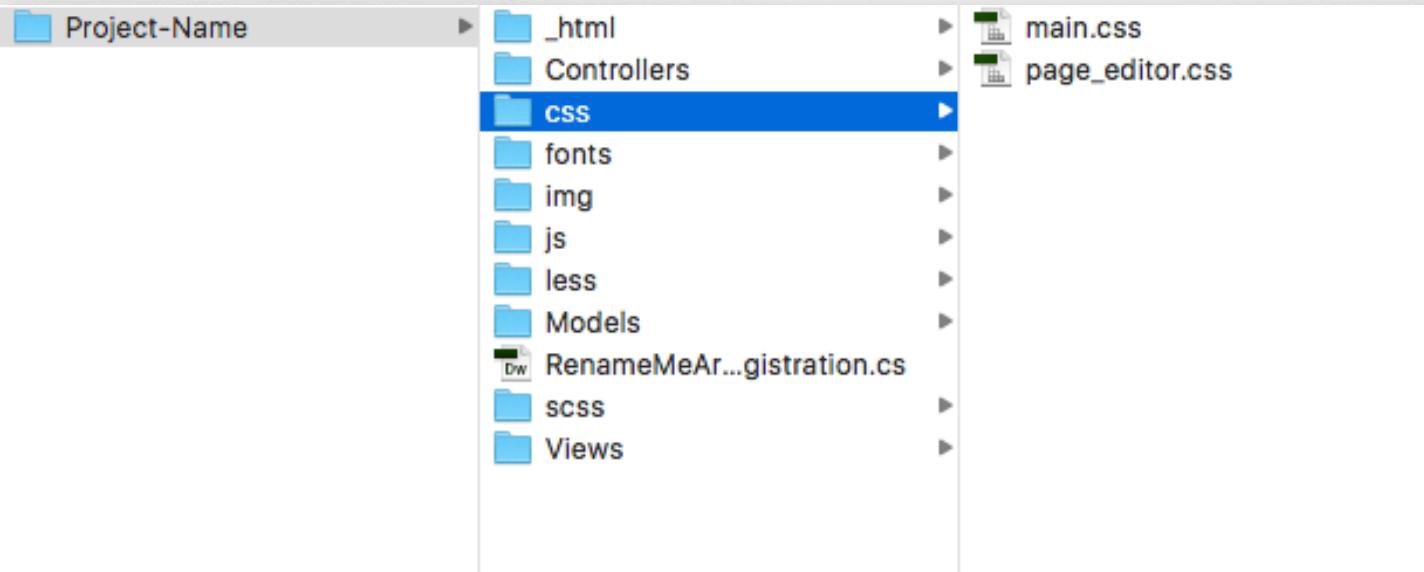
The screenshot shows a code editor window with the file 'page\_editor.scss' open. The code is written in SCSS (Sass) and includes comments and CSS declarations. The code is as follows:

```
// *** START: Fix for Bootstrap UI CSS and Experience Editor
.scInsertionHandleCenter, .sc-breadcrumb .sc-breadcrumb-item-path,
{
    -webkit-box-sizing: initial;
    -moz-box-sizing: initial;
    box-sizing: initial;
}
.sc-breadcrumb-item-path {
    img {
        vertical-align: initial;
    }
    span {
        font-size: 1em;
    }
}
```

# File Structure

## Project CSS files

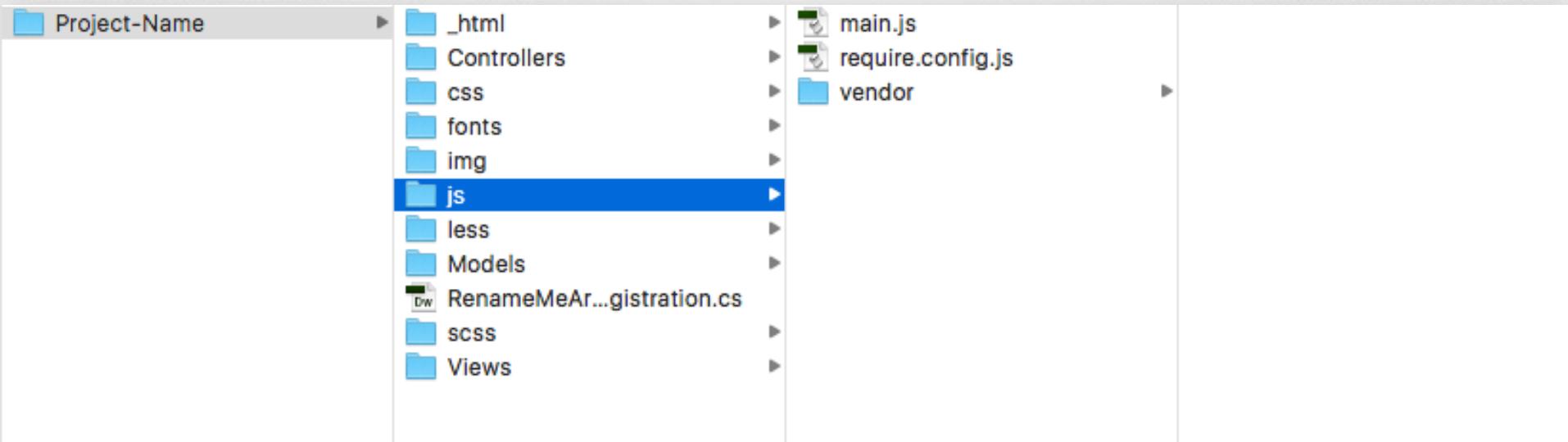
- **NEVER** edit CSS styles directly
- CSS files are **ONLY** to be compiled via Sass



# File Structure

## Project JS files

- SCORE 2.0+ uses [require.js](#) to deliver JavaScript files to your project
- Works in conjunction with main.js



# File Structure

## Project JS files

- Modernizr JS included with SCORE
- Adds classes to the <html> element based on current browser's feature support
- Allows you to use more modern technologies while still creating fallbacks for old browsers

```
<html lang="en" class=" js flexbox canvas canvastext webgl no-touch geolocation postmessage websqldatabase indexeddb hashchange history draganddrop websockets rgba hsla multiplebgs backgroundsize borderimage borderradius boxshadow textshadow opacity cssanimations csscolumns cssgradients cssreflections csstransforms csstransforms3d csstransitions fontface generatedcontent video audio localstorage sessionstorage webworkers applicationcache svg inlinesvg smil svgclippaths">
```

# File Structure

## Project JS files

- matchHeight JS also included with SCORE
- Allows you to match the heights of elements in a row using a simple line of JS

The diagram illustrates the 'matchHeight' JavaScript feature. On the left, three cards are shown side-by-side. Each card has a circular icon at the top left and a title below it. A red arrow points from the middle card to the right, indicating a transformation. On the right, the three cards appear identical, with the same icons and titles, demonstrating that the heights of the elements have been matched.

Implementation  
Methodology

We refer to our approach to managing projects as Agile Transparency. This means everything is visible to our partners and priorities are fluid.

[LEARN MORE](#)

Certified Sitecore  
Expertise

As a leading Sitecore partner in the southeast, we specialize in complex enterprise implementations.

[LEARN MORE](#)

Comprehensive  
Quality Assurance

Quality Assurance is a part of every step in the process from the first sprint to the last.

[LEARN MORE](#)

Implementation  
Methodology

We refer to our approach to managing projects as Agile Transparency. This means everything is visible to our partners and priorities are fluid.

[LEARN MORE](#)

Certified Sitecore  
Expertise

As a leading Sitecore partner in the southeast, we specialize in complex enterprise implementations.

[LEARN MORE](#)

Comprehensive  
Quality Assurance

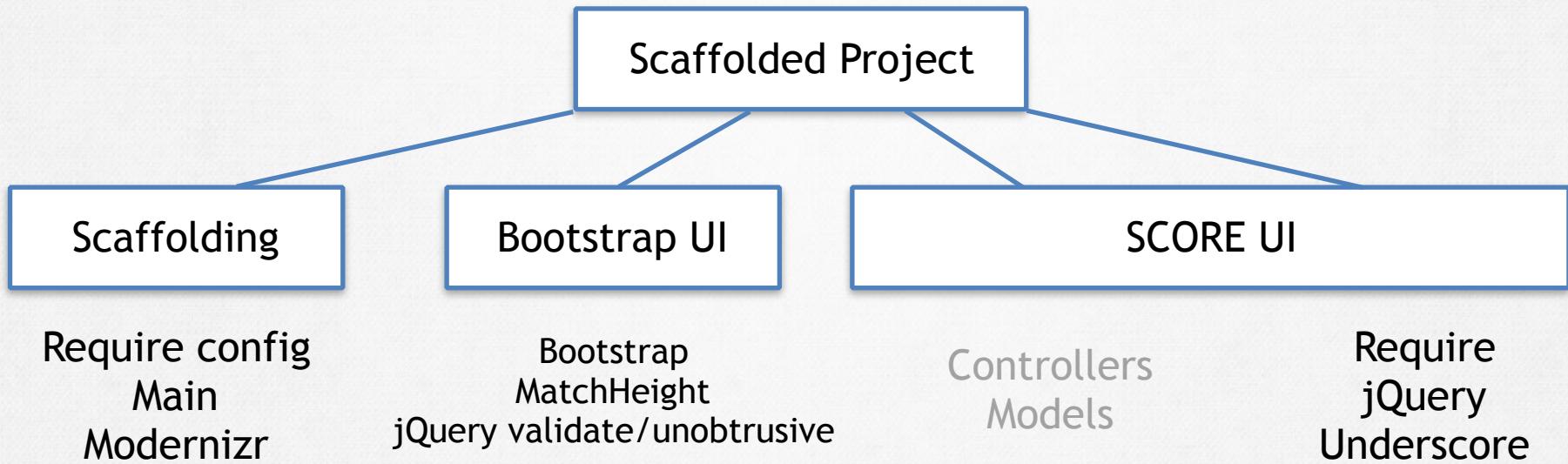
Quality Assurance is a part of every step in the process from the first sprint to the last.

[LEARN MORE](#)

# File Structure

## Project Scaffolding

- Scaffolding layers: Score Scaffolding, Score Bootstrap UI, Score UI
- Some JS files are stored in other layers
- Accessing these JS files if you do not run Sitecore locally:
  - Use jQuery/Bootstrap CDN (included in \_html files)
  - Use local copies of these files and do not commit them



03

# Bootstrap 101

HTML/CSS/JS

# Bootstrap

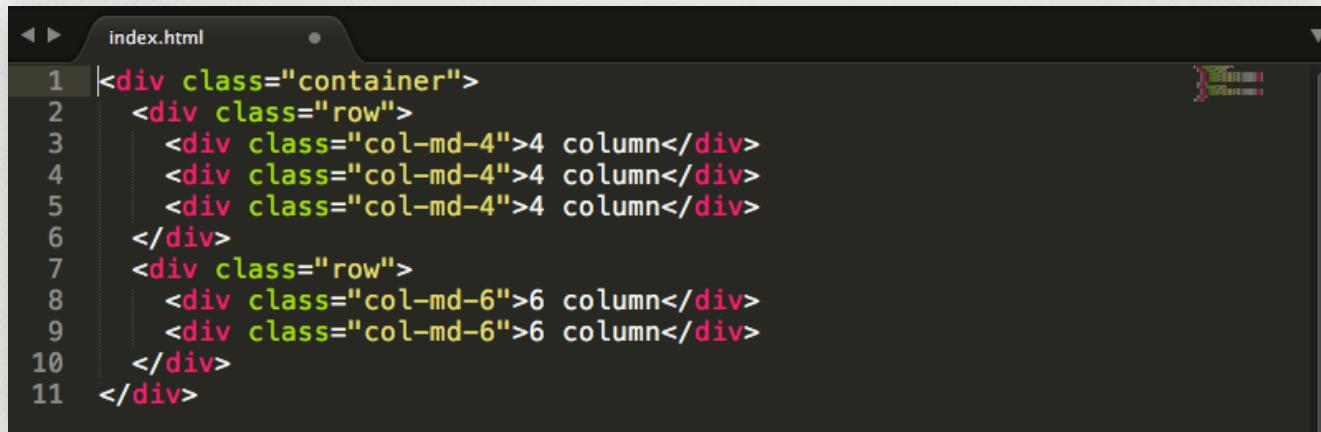
## What is Bootstrap?

- Naturally responsive UI framework
- Based on Sass pre-processor
- Provides a native responsive grid
- Source code includes:
  - Styling for native grid
  - Styling for over 20 native components
  - Javascript file for JS dependent components
- Reference: [Bootstrap 3](#)

# Bootstrap

## Using the grid

- Every page uses a container that holds rows and columns
- Bootstrap uses a 12-column layout per row (by default)
- Each element uses a specific class to pull in native Bootstrap styling
  - Example class names: `.container`, `.row`, `.col-md-4`



A screenshot of a code editor window titled "index.html". The code is written in HTML and shows a basic grid structure:

```
1 <div class="container">
2   <div class="row">
3     <div class="col-md-4">4 column</div>
4     <div class="col-md-4">4 column</div>
5     <div class="col-md-4">4 column</div>
6   </div>
7   <div class="row">
8     <div class="col-md-6">6 column</div>
9     <div class="col-md-6">6 column</div>
10  </div>
11 </div>
```

- Reference: [How Bootstrap's grid works](#)

# Bootstrap

## The Anatomy of Bootstrap's Grid

- “.container”
  - width based on screen size
  - 15px padding on left and right
  - margin: 0 auto; (always horizontally centered)

div.container 1170px x 288px

Heading	Heading	Heading
Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui. <a href="#">View details »</a>	Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui. <a href="#">View details »</a>	Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Vestibulum id ligula porta felis euismod semper. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. <a href="#">View details »</a>

# Bootstrap



## The Anatomy of Bootstrap's Grid

- “.row”
  - width is 100% of parent
  - -15px margin on left and right
  - Parent should either be a container or a column

### Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

[View details »](#)

### Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

[View details »](#)

### Heading

Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Vestibulum id ligula porta felis euismod semper. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.

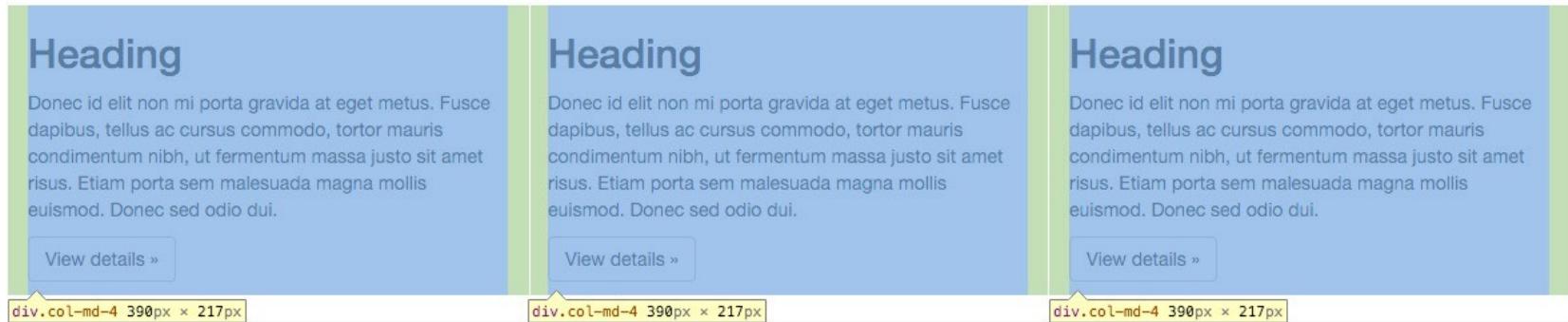
[View details »](#)

div.row 1170px x 217px

# Bootstrap

## The Anatomy of Bootstrap's Grid

- “.col-” (Column)
  - width is specified by developer using a class or mixin (Bootstrap uses 12-column grid)
  - 15px padding on left and right (30px “gutter” between columns)
  - ALWAYS inserted into a row



# Bootstrap



## The Anatomy of Bootstrap's Grid

- Nested row with columns:
  - To break up a column into more columns, first insert a nested row

### Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

[View details »](#)

### Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

[View details »](#)

### Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

[View details »](#)

### Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

[View details »](#)

`div.row 780px x 217px`

# Bootstrap

## The Anatomy of Bootstrap's Grid

- Nested row with columns:
  - Insert columns into the row like normal
  - `.col-lg-6` will be half the width of the parent `.row`

### Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

[View details »](#)

### Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

[View details »](#)

### Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

[View details »](#)

`div.col-md-6 390px x 217px`

### Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

[View details »](#)

`div.col-md-6 390px x 217px`

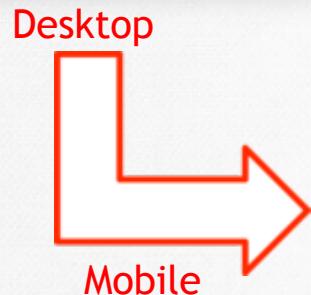
# Bootstrap

## Mobile Breakpoints

- Pixel, em, or rem value representing a screen width
  - Bootstrap uses pixels, defines 4 sizes: xs, sm, md, lg
- Mobile sizes do not have enough horizontal real estate for most columns to appear next to each other
- Grid columns “wrap” when screen width is less than mobile breakpoint size
  - All columns become 100% of screen width
  - Leftmost column becomes top column, columns to the right stack below in order

## Heading 1

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

[View details »](#)

## Heading 2

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

[View details »](#)

## Heading 3

Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Vestibulum id ligula porta felis euismod semper. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.

[View details »](#)

The image shows a single column of three cards, each containing a heading and text, demonstrating the responsive design of Bootstrap's grid system. The cards are arranged vertically and have rounded corners. Each card contains a heading (Heading 1, Heading 2, or Heading 3) followed by a paragraph of text and a "View details »" button. A large red arrow on the left side of the image points from the word "Desktop" to the word "Mobile", illustrating how a desktop layout is scaled down for mobile devices.

## Heading 1

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

[View details »](#)

## Heading 2

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

[View details »](#)

## Heading 3

Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Vestibulum id ligula porta felis euismod semper. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.

[View details »](#)

# Bootstrap

## Column Operations

- Offset
  - Moves columns to the right a specified number of column units
  - Uses class .col-md-offset-\*
    - Example: col-md-offset-4 increases left margin by 4 column units



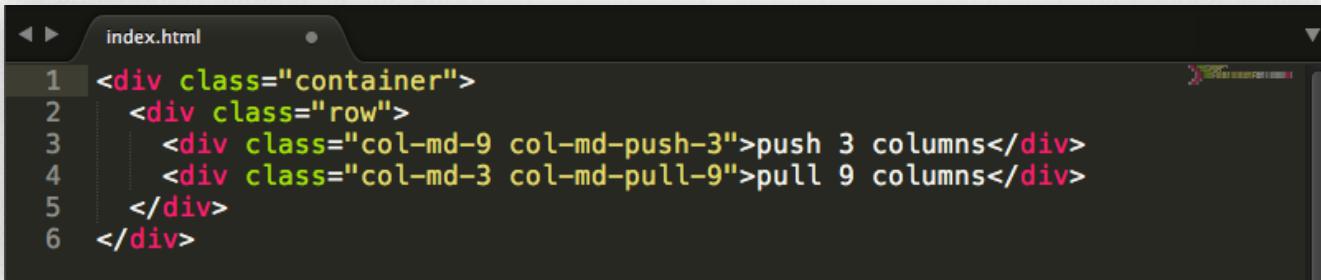
A screenshot of a code editor window titled "index.html". The code is as follows:

```
1 <div class="container">
2   <div class="row">
3     <div class="col-md-9 col-md-offset-3">offset 3 columns</div>
4   </div>
5 </div>
```

# Bootstrap

## Column Operations

- Order
  - Switches order of columns
  - Uses classes `.col-md-push-*` and `.col-md-pull-*`



A screenshot of a code editor window titled "index.html". The code is written in HTML and uses Bootstrap's grid system:

```
1 <div class="container">
2   <div class="row">
3     <div class="col-md-9 col-md-push-3">push 3 columns</div>
4     <div class="col-md-3 col-md-pull-9">pull 9 columns</div>
5   </div>
6 </div>
```

# Bootstrap

## Column Operations

- “Stacking” column sizes
  - Specifying multiple column sizes will allow a column to be different sizes depending on screen size
  - Example: .col-md-3 .col-lg-4 will be a 4-column on large screens & up and a 3-column on medium-large



A screenshot of a code editor window titled "index.html". The code is written in HTML and uses Bootstrap's grid system:

```
1 <div class="container">
2   <div class="row">
3     <div class="col-md-3 col-lg-4"></div>
4   </div>
5 </div>
```

# Bootstrap

## Column Operations

- Responsive utilities
  - Show or hide columns based on viewport width
  - Uses classes `.visible-**` or `.hidden-**`
    - Example: `.hidden-md` is hidden on screen sizes 992px - 1199px wide

	Extra small devices Phones (<768px)	Small devices Tablets ( $\geq 768px$ )	Medium devices Desktops ( $\geq 992px$ )	Large devices Desktops ( $\geq 1200px$ )
<code>.visible-xs-*</code>	Visible	Hidden	Hidden	Hidden
<code>.visible-sm-*</code>	Hidden	Visible	Hidden	Hidden
<code>.visible-md-*</code>	Hidden	Hidden	Visible	Hidden
<code>.visible-lg-*</code>	Hidden	Hidden	Hidden	Visible
<code>.hidden-xs</code>	Hidden	Visible	Visible	Visible
<code>.hidden-sm</code>	Visible	Hidden	Visible	Visible
<code>.hidden-md</code>	Visible	Visible	Hidden	Visible
<code>.hidden-lg</code>	Visible	Visible	Visible	Hidden

# Bootstrap

## Native Components

- Bootstrap styles native components using specific class names:
  - .nav, .navbar, .carousel, .jumbotron, .nav-tabs, .collapse, etc.
  - Note: by default, SCORE components inherit Bootstrap's native component styles (via LESS/Sass as of 2.0)
- bootstrap.js included in project scaffolding for use with components that require JavaScript
  - Carousel, tab sets, accordions, navigation
- References: [CSS components](#) / [JS components](#)

# Sass 101

04



4.1

# Styling with Sass

Writing in scss and how it works for you

# Sass



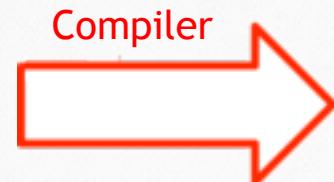
## Sass Basics

- CSS pre-processor (compiles into CSS)
  - All valid CSS is valid Sass
- Makes writing CSS easier with many added features
- Allows nested styles, which is *amazing* - but can *easily* get out of control
  - Try to keep nesting limited to 4 levels deep
- References: [Sass Tutorial](#), [Sass Style Guide](#)

# Sass

## Sass Compiling

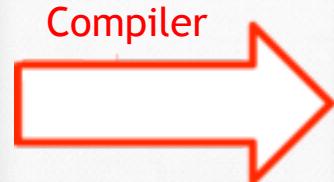
Sass



css



```
_sass.scss
1 //Sass
2 ul {
3     list-style: none;
4     li {
5         float: left;
6         a {
7             color: #ff0000;
8         }
9     }
10 }
```



```
_sass.css
1 ul {
2     list-style: none;
3 }
4 ul li {
5     float: left;
6 }
7 ul li a {
8     color: #ff0000;
9 }
```

# Sass



Command Line

Mac

Windows



Ruby

<CODEKIT>



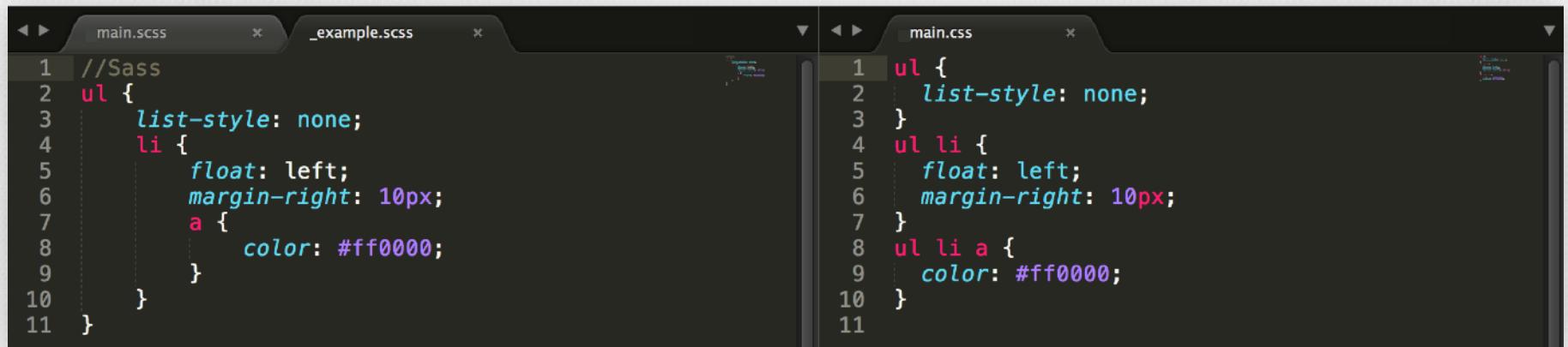
Prepros

- Reference: [Installing a compiler](#)

# LAB 1

# Sass

## Good Nesting Example



The screenshot shows a code editor with two tabs on the left: "main.scss" and "\_example.scss". On the right, there are two tabs: "main.css" and an empty tab. The "main.scss" tab contains the following Sass code:

```
1 //Sass
2 ul {
3   list-style: none;
4   li {
5     float: left;
6     margin-right: 10px;
7     a {
8       color: #ff0000;
9     }
10 }
11 }
```

The "main.css" tab contains the generated CSS output:

```
1 ul {
2   list-style: none;
3 }
4 ul li {
5   float: left;
6   margin-right: 10px;
7 }
8 ul li a {
9   color: #ff0000;
10 }
11 }
```

# Sass

## Bad Nesting Example

The image shows a code editor interface with two tabs: `main.scss` and `main.css`. The `main.scss` tab contains the following SCSS code:

```
1 .container {  
2     .row-class {  
3         .column-class {  
4             ul {  
5                 list-style: none;  
6                 li {  
7                     float: left;  
8                     margin-right: 10px;  
9                     a {  
10                         color: #ff0000;  
11                     }  
12                 }  
13             }  
14         }  
15     }  
16 }
```

The `main.css` tab contains the generated CSS code:

```
1 .container .row-class .column-class ul {  
2     list-style: none;  
3 }  
4 .container .row-class .column-class ul li {  
5     float: left;  
6     margin-right: 10px;  
7 }  
8 .container .row-class .column-class ul li a {  
9     color: #ff0000;  
10 }
```

# Sass

## Nesting Media Queries

- Media queries can be nested to specify mobile behavior for individual elements

The image shows a code editor interface with two tabs: 'main.scss' and '\_example.scss'. The '\_example.scss' tab is active, displaying the following Sass code:

```
1 //Sass
2 .class-1 {
3     @media only screen and (min-width: 768px) {
4         color: #ff0000;
5     }
6     @media only screen and (min-width: 992px) {
7         color: #2b2b2b;
8     }
9 }
```

The 'main.css' tab is also visible, showing the generated CSS output:

```
1 @media only screen and (min-width: 768px) {
2     .class-1 {
3         color: #ff0000;
4     }
5 }
6 @media only screen and (min-width: 992px) {
7     .class-1 {
8         color: #2b2b2b;
9     }
10 }
11 }
```

# Sass

## Tying Up The Markup & Styling

- Nesting structure can mimic HTML hierarchy

The image shows a code editor with two files side-by-side. On the left is `index.html`, which contains the following HTML code:

```
1 <div class="score-highlight">
2   <div class="score-highlight-header">
3     ...
4   </div>
5   
6   <div class="caption">
7     <h2>...</h2>
8     <div class="score-highlight-body">
9       ...
10    </div>
11    <div class="score-call-to-action">
12      ...
13    </div>
14  </div>
15 </div>
```

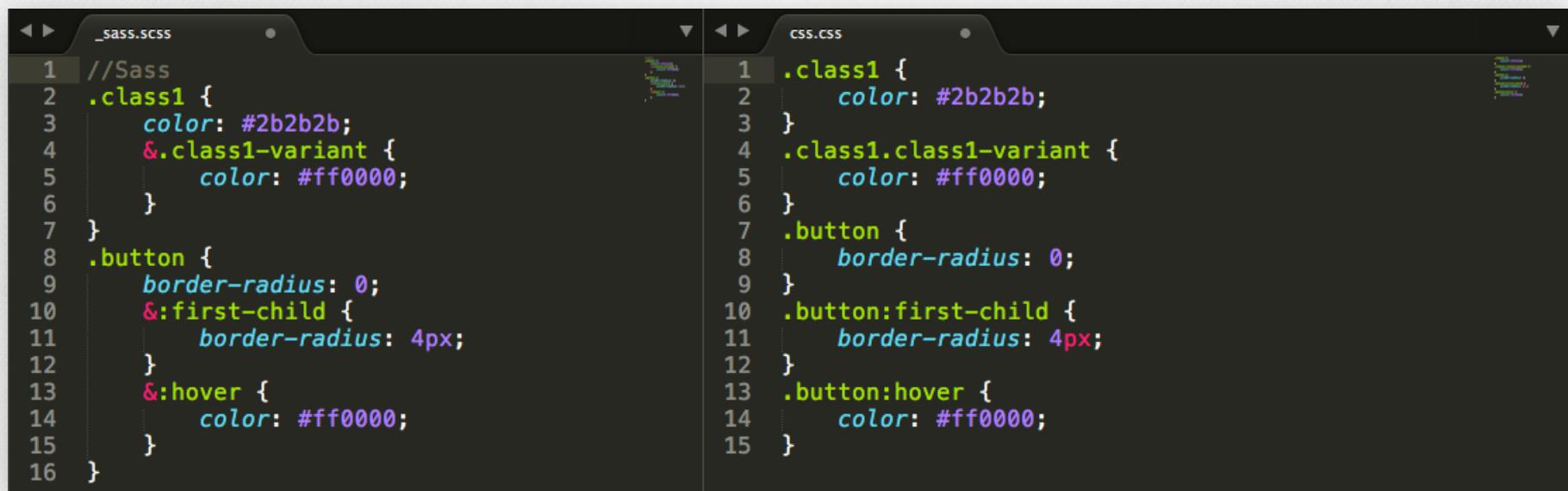
On the right is `_site-highlights.scss`, which contains the following SCSS code:

```
1 .score-highlight {
2   @extend .thumbnail; //inherit bootstrap
3   @include vertical-space($vertical-spacing);
4   .score-highlight-header {
5   }
6   img {
7   }
8   .caption {
9     h2 {
10   }
11     .score-highlight-body {
12   }
13     .score-call-to-action {
14   }
15   }
16   }
17   }
18   }
19   }
20   }
21   }
22 }
```

# Sass

## Sass Operators

- &
  - Appends statement to immediate parent
  - Commonly used for compound classes and pseudo elements



The image shows a code editor interface with two panes. The left pane is labeled '\_sass.scss' and the right pane is labeled 'css.css'. Both panes show the same code, demonstrating the compilation of Sass into standard CSS.

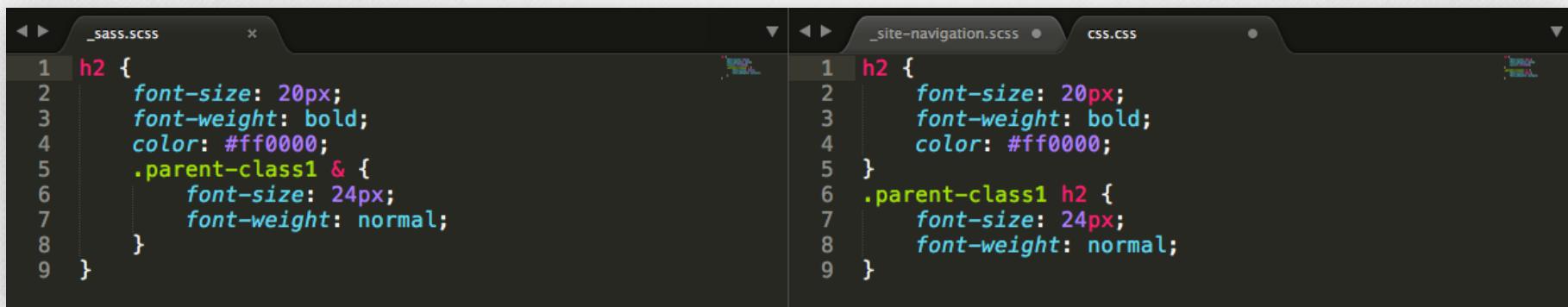
```
//Sass
.class1 {
  color: #2b2b2b;
  &.class1-variant {
    color: #ff0000;
  }
}
.button {
  border-radius: 0;
  &:first-child {
    border-radius: 4px;
  }
  &:hover {
    color: #ff0000;
  }
}

.class1 {
  color: #2b2b2b;
}
.class1.class1-variant {
  color: #ff0000;
}
.button {
  border-radius: 0;
}
.button:first-child {
  border-radius: 4px;
}
.button:hover {
  color: #ff0000;
}
```

# Sass

## Sass Operators

- &
  - Can also go at the end of a selector to append the PARENT statement to the end



The image shows a code editor with two tabs: '\_SASS.scss' on the left and '\_site-navigation.scss' on the right. The '\_SASS.scss' tab contains the following Sass code:

```
1 h2 {  
2   font-size: 20px;  
3   font-weight: bold;  
4   color: #ff0000;  
5   .parent-class1 & {  
6     font-size: 24px;  
7     font-weight: normal;  
8   }  
9 }
```

The '\_site-navigation.scss' tab contains the resulting CSS output:

```
1 h2 {  
2   font-size: 20px;  
3   font-weight: bold;  
4   color: #ff0000;  
5 }  
6 .parent-class1 h2 {  
7   font-size: 24px;  
8   font-weight: normal;  
9 }
```

# Sass

## Sass Operators

- >
  - Direct child selector (same as in CSS)
  - Can be nested and combined with “&”

The image shows a code editor interface with two tabs: '\_sass.scss' on the left and 'CSS.css' on the right. The '\_sass.scss' tab contains the following Sass code:

```
//Sass
.class1 {
  & > .class2 {
    color: #ff0000;
  }
}
```

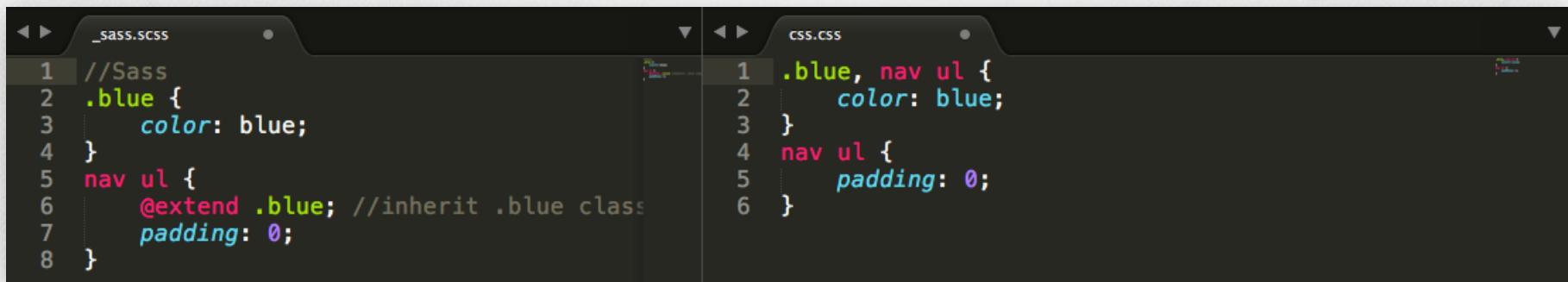
The 'CSS.css' tab contains the generated CSS code:

```
.class1 > .class2 {
  color: #ff0000;
}
```

# Sass

## Extending Classes

- @extend adds the current selector to the selector that is being extended
- Reference: [The extend concept](#)



The image shows a code editor interface with two tabs: '\_sass.scss' on the left and 'CSS.CSS' on the right. The '\_sass.scss' tab contains the following Sass code:

```
//Sass
.blue {
  color: blue;
}
nav ul {
  @extend .blue; //inherit .blue class
  padding: 0;
```

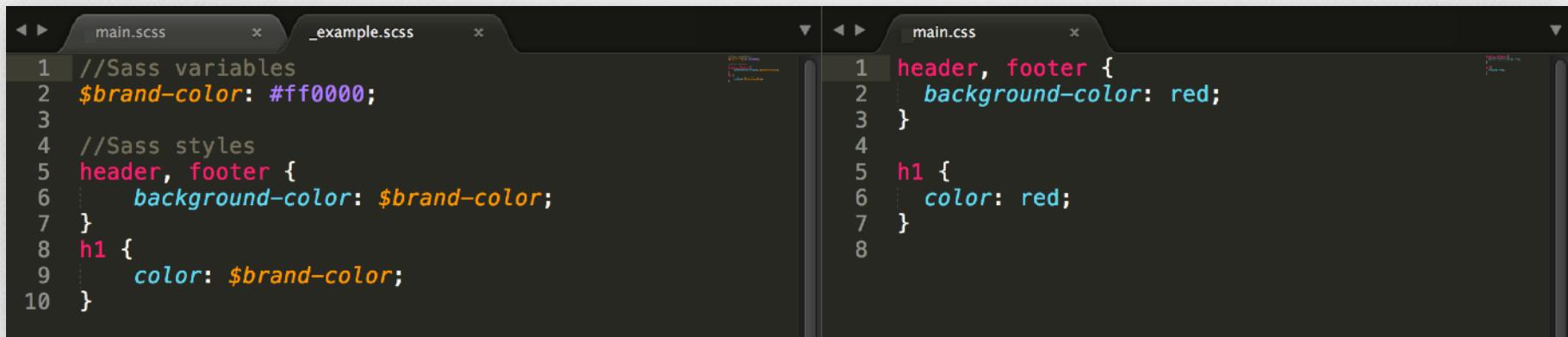
The 'CSS.CSS' tab contains the generated CSS code:

```
.blue, nav ul {
  color: blue;
}
nav ul {
  padding: 0;
```

# Sass

## Sass Variables

- Declared locally or globally and prefixed with \$
- Can be a color, font, another variable, etc.
- Code is easily maintained - changing the value of the variable will change all occurrences
- Bootstrap includes native variables for colors, fonts, and common mobile breakpoints



The image shows a code editor interface with three tabs: 'main.scss', '\_example.scss', and 'main.css'. The 'main.scss' tab contains the following Sass code:

```
1 //Sass variables
2 $brand-color: #ff0000;
3
4 //Sass styles
5 header, footer {
6   background-color: $brand-color;
7 }
8 h1 {
9   color: $brand-color;
10 }
```

The '\_example.scss' tab contains a single line of code:

```
1 //Sass variables
```

The 'main.css' tab contains the generated CSS code:

```
1 header, footer {
2   background-color: red;
3 }
4 h1 {
5   color: red;
6 }
7
8
```

# Sass

## Sass Mixins

- Declared locally or globally and prefixed with @mixin
- Create easily reusable and maintainable blocks of code
- Commonly used for styles that require browser prefixes
- Can accept parameters to use different values in different locations

The image shows a code editor interface with two tabs open. The left tab is titled '\_site-variables.scss' and contains the following code:

```
1 //Global custom variables
2
3 //default margin on content components
4 $vertical-spacing: 30px;
5
6 //Global custom mixins
7 @mixin vertical-space($margin) {
8   margin-bottom: $margin;
9 }
10 @mixin rounded-corners($radius) {
11   -webkit-border-radius: $radius;
12   -moz-border-radius: $radius;
13   -o-border-radius: $radius;
14   border-radius: $radius;
15 }
```

The right tab is titled '\_site-highlights.scss' and contains the following code:

```
1 .score-highlight {
2   @include vertical-space($vertical-spacing);
3   @include rounded-corners(3px);
4 }
5
```

# Sass

## Sass Mixins

- Example of pixel to rem converter mixin

The screenshot shows a code editor with two tabs: '\_SASS.scss' on the left and 'CSS.CSS' on the right. The '\_SASS.scss' tab contains the following SCSS code:

```
1 @mixin pxConvert($px) {  
2     font-size: ($px / 16) + rem;  
3 }  
4  
5 h2 {  
6     @include pxConvert(12);  
7     font-weight: bold;  
8     color: #ff0000;  
9 }  
10  
11
```

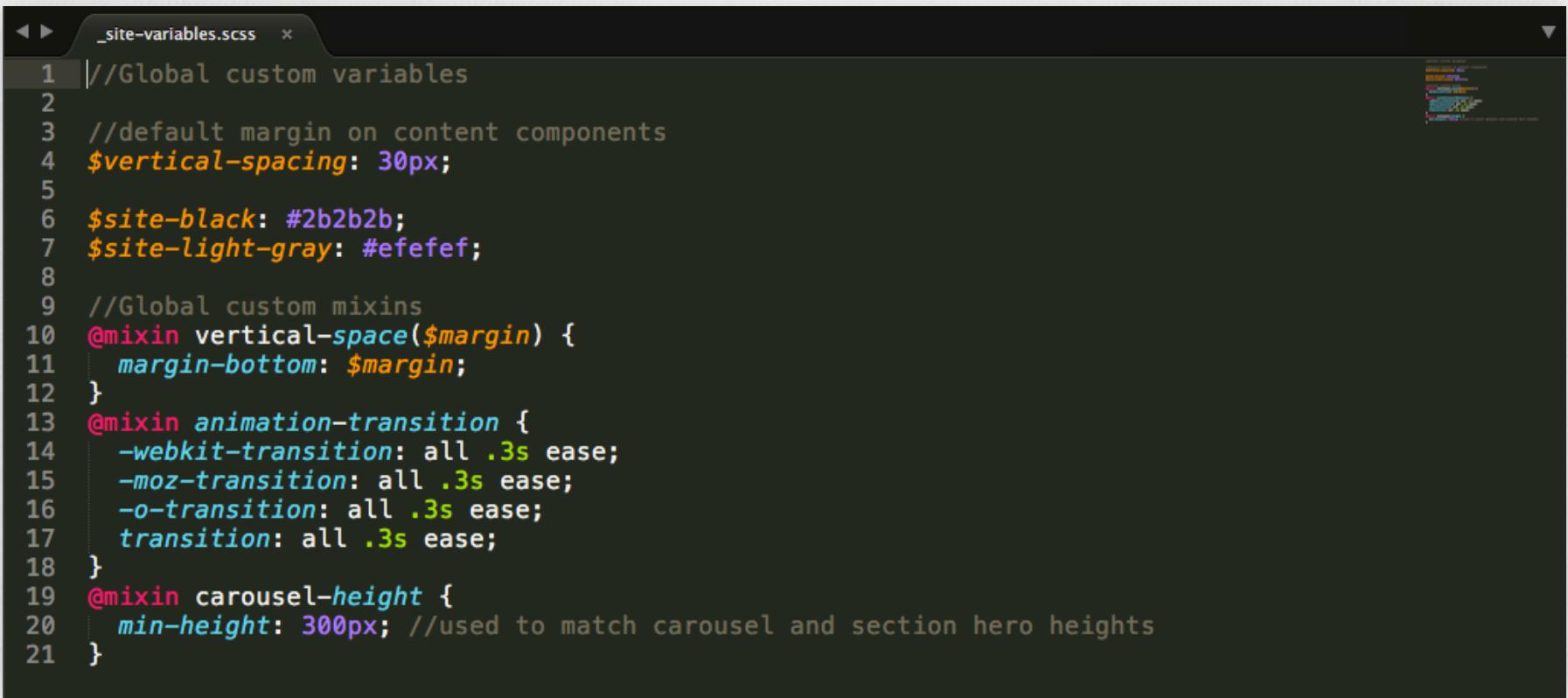
The 'CSS.CSS' tab contains the generated CSS output:

```
1 h2 {  
2     font-size: 0.75rem;  
3     font-weight: bold;  
4     color: #ff0000;  
5 }
```

# Sass

## Sass Mixins and Variables

- Example of \_site-variables.scss w/ mixins and variables  
*(used for training site example)*

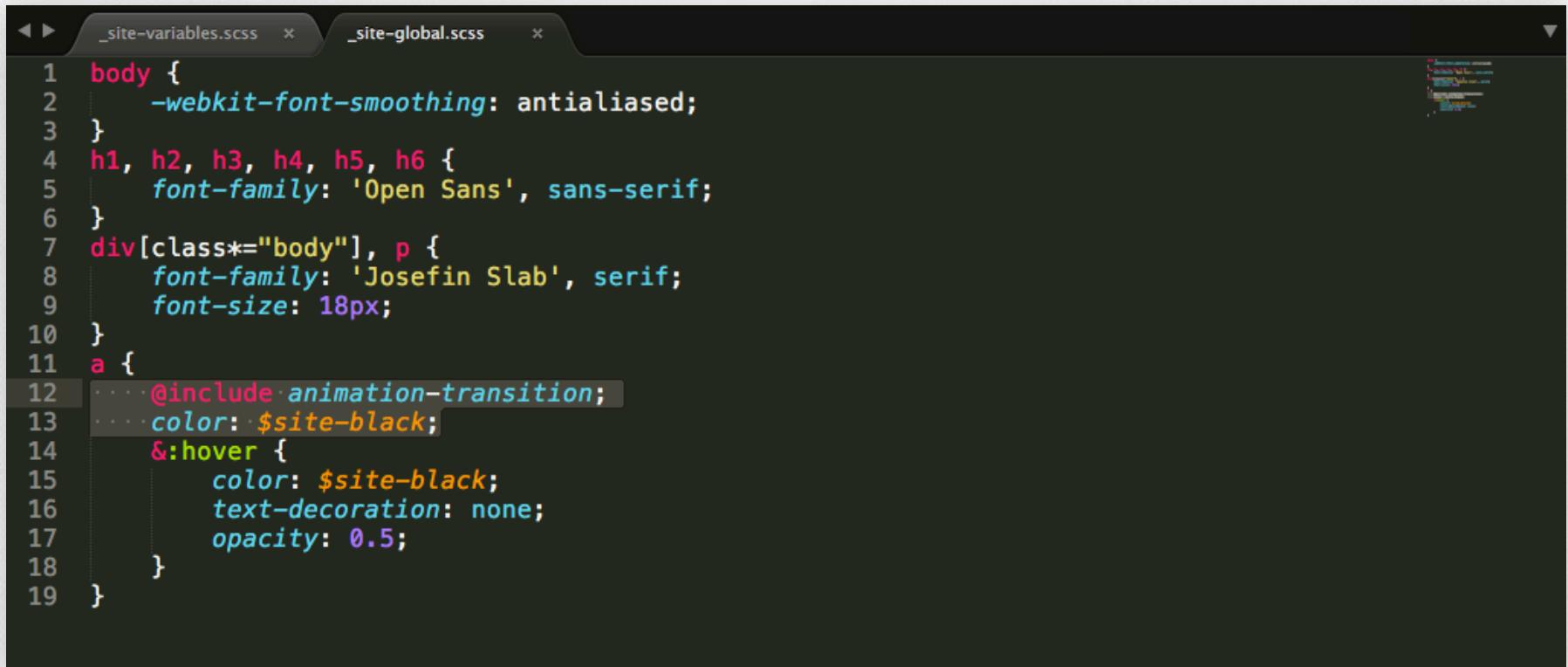


```
1 //Global custom variables
2
3 //default margin on content components
4 $vertical-spacing: 30px;
5
6 $site-black: #2b2b2b;
7 $site-light-gray: #efefef;
8
9 //Global custom mixins
10 @mixin vertical-space($margin) {
11   margin-bottom: $margin;
12 }
13 @mixin animation-transition {
14   -webkit-transition: all .3s ease;
15   -moz-transition: all .3s ease;
16   -o-transition: all .3s ease;
17   transition: all .3s ease;
18 }
19 @mixin carousel-height {
20   min-height: 300px; //used to match carousel and section hero heights
21 }
```

# Sass

## Sass Mixins and Variables

- Example of animation-transition mixin and \$site-black variable  
(used for training site example)



The screenshot shows a code editor with two files open: `_site-variables.scss` and `_site-global.scss`. The `_site-variables.scss` file contains the following code:

```
1 body {
2     -webkit-font-smoothing: antialiased;
3 }
4 h1, h2, h3, h4, h5, h6 {
5     font-family: 'Open Sans', sans-serif;
6 }
7 div[class*="body"], p {
8     font-family: 'Josefin Slab', serif;
9     font-size: 18px;
10 }
11 a {
12     @include animation-transition;
13     color: $site-black;
14     &:hover {
15         color: $site-black;
16         text-decoration: none;
17         opacity: 0.5;
18     }
19 }
```

The `color: $site-black;` line is highlighted in yellow, indicating it's a variable reference. The `animation-transition` mixin is also highlighted.

# Sass

## Vertical Spacing

- Set to 30px for vertical space by default

```
_site-variables.scss
1 //Global custom variables
2
3 //default margin on content components
4 $vertical-spacing: 30px;
5
6 //Global custom mixins
7 @mixin vertical-space($margin) {
8   margin-bottom: $margin;
9 }

_site-highlights.scss
1 .score-highlight {
2   @extend .thumbnail; //inherit bootstrap
3   @include vertical-space($vertical-spacing);
4   .score-highlight-header {
5
6   }
7   img {
8
9   }
10  .caption {
11    h2 {
12
13    }
14    .score-highlight-body {
15
16    }
17    .score-call-to-action {
18
19    }
20  }
21
22 }
```



4.2

# Inheriting Bootstrap via Sass

How SCORE inherits default styles

# Sass

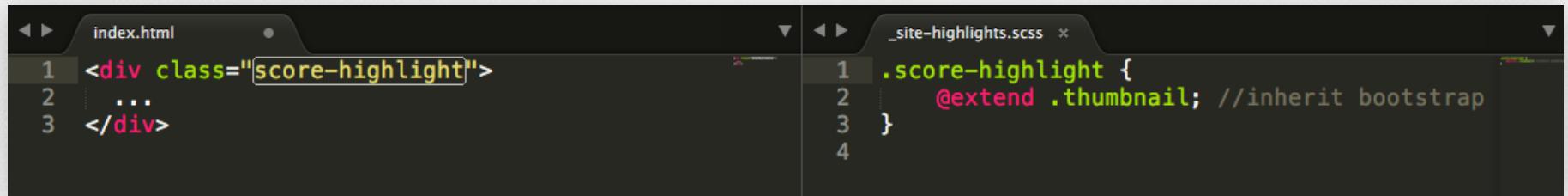
## Inheritance

- Declared with @extend .class-name-you're-extending
- This is the magic that includes your framework's classes into your default component HTML
- Simply remove the extension if you need to remove default styles

```
_main.scss          _site-highlights.scss          _main.css
1  .score-highlight {           1  .score-highlight {           5910
2    @extend .thumbnail; //inherit bootstrap           5911   .thumbnail, .score-highlight {
3  }                                         5912     display: block;
4                                         5913     padding: 4px;
5                                         5914     margin-bottom: 20px;
6                                         5915     line-height: 1.42857;
7                                         5916     background-color: white;
8                                         5917     border: 1px solid #dddddd;
9                                         5918     border-radius: 4px;
10                                         5919     -webkit-transition: border 0.2s ease-in;
11                                         5920     -o-transition: border 0.2s ease-in-out;
12                                         5921     transition: border 0.2s ease-in-out;
13                                         5922   }
14                                         5923   .thumbnail > img, .score-highlight > img,
15                                         5924   .thumbnail a > img,
16                                         5925   .score-highlight a > img {
17                                         5926     display: block;
18                                         5927     width: 100%
```

# Sass

- Default style class names are removed from the HTML
- Default styles are simply inherited via Sass with @extend
- Removing necessary default styles is easy



The screenshot shows a code editor with two tabs: "index.html" and "\_site-highlights.scss".

The "index.html" tab contains the following code:

```
1 <div class="score-highlight">
2 ...
3 </div>
```

The "score-highlight" class is highlighted in yellow.

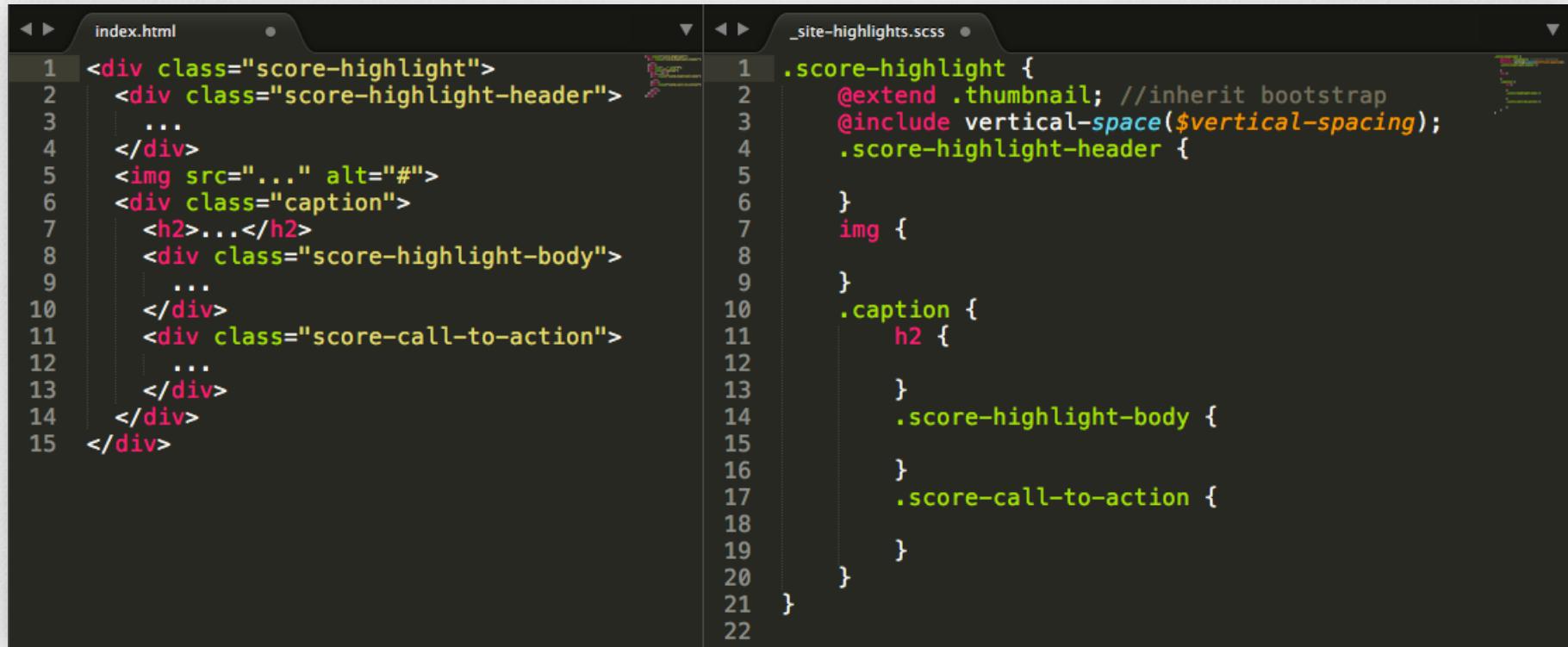
The "\_site-highlights.scss" tab contains the following SCSS code:

```
1 .score-highlight {
2   @extend .thumbnail; //inherit bootstrap
3 }
4
```

# Sass

## Tying Up The Markup & Styling

- Default styles are pre-scaffolded to reflect default markup
- Empty nests do not compile unnecessary CSS!



The image shows a code editor with two tabs: "index.html" and "\_site-highlights.scss".

**index.html:**

```
1 <div class="score-highlight">
2   <div class="score-highlight-header">
3     ...
4   </div>
5   
6   <div class="caption">
7     <h2>...</h2>
8     <div class="score-highlight-body">
9       ...
10    </div>
11    <div class="score-call-to-action">
12      ...
13    </div>
14  </div>
15 </div>
```

**\_site-highlights.scss:**

```
1 .score-highlight {
2   @extend .thumbnail; //inherit bootstrap
3   @include vertical-space($vertical-spacing);
4   .score-highlight-header {
5   }
6   img {
7   }
8   .caption {
9     h2 {
10   }
11   .score-highlight-body {
12   }
13   .score-call-to-action {
14   }
15 }
16 }
17 }
18 }
19 }
20 }
21 }
22 }
```

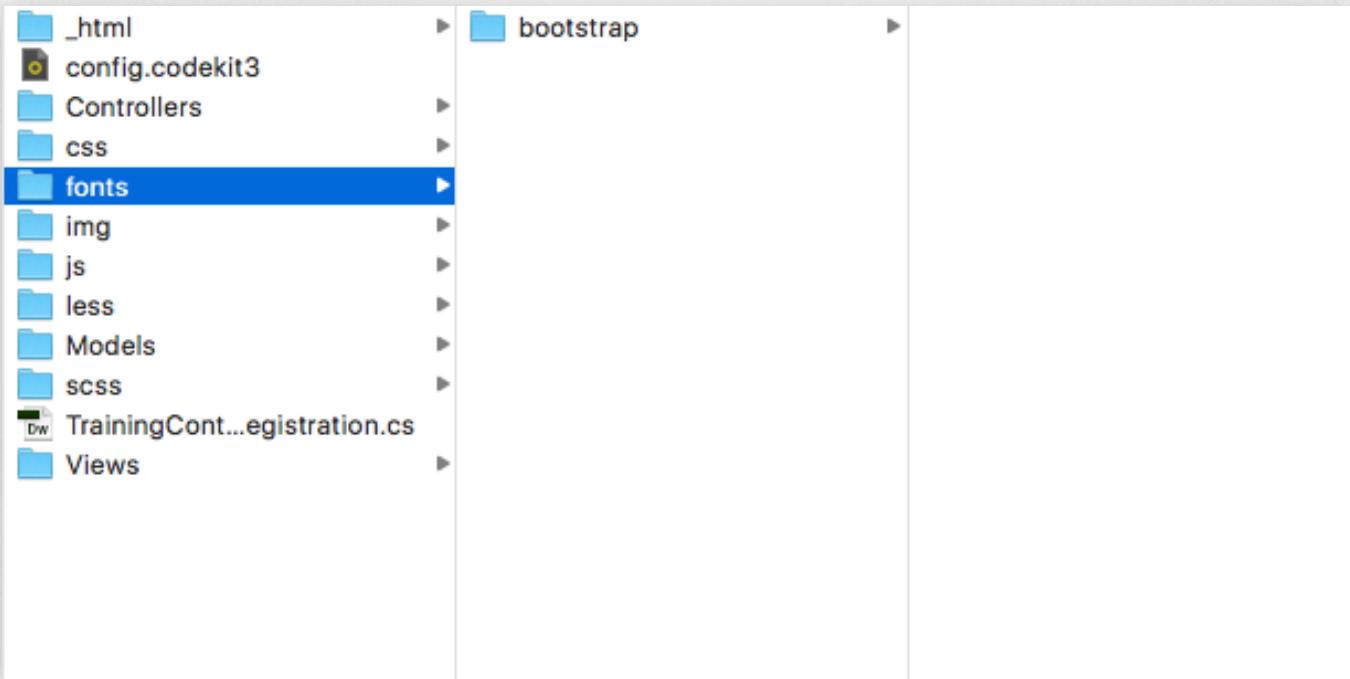


4.3

## Including additional scss files

# Sass

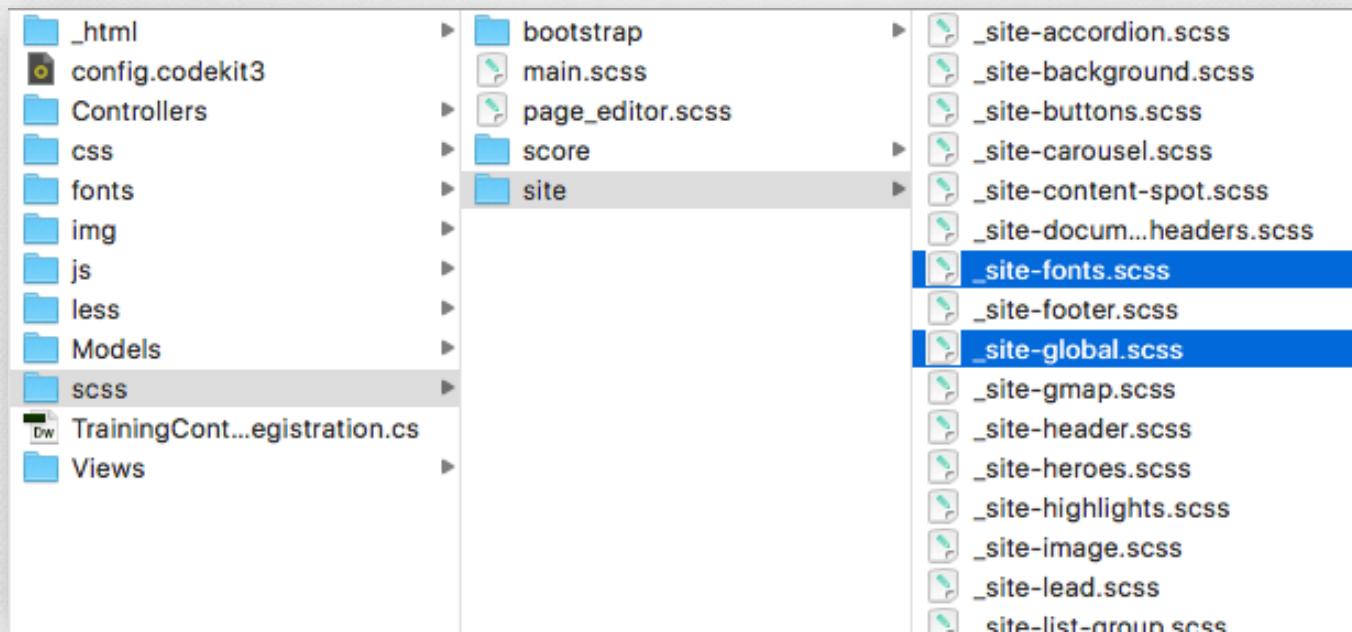
- Include fonts and fonts scss file for designed text treatment



[Google web fonts](#)

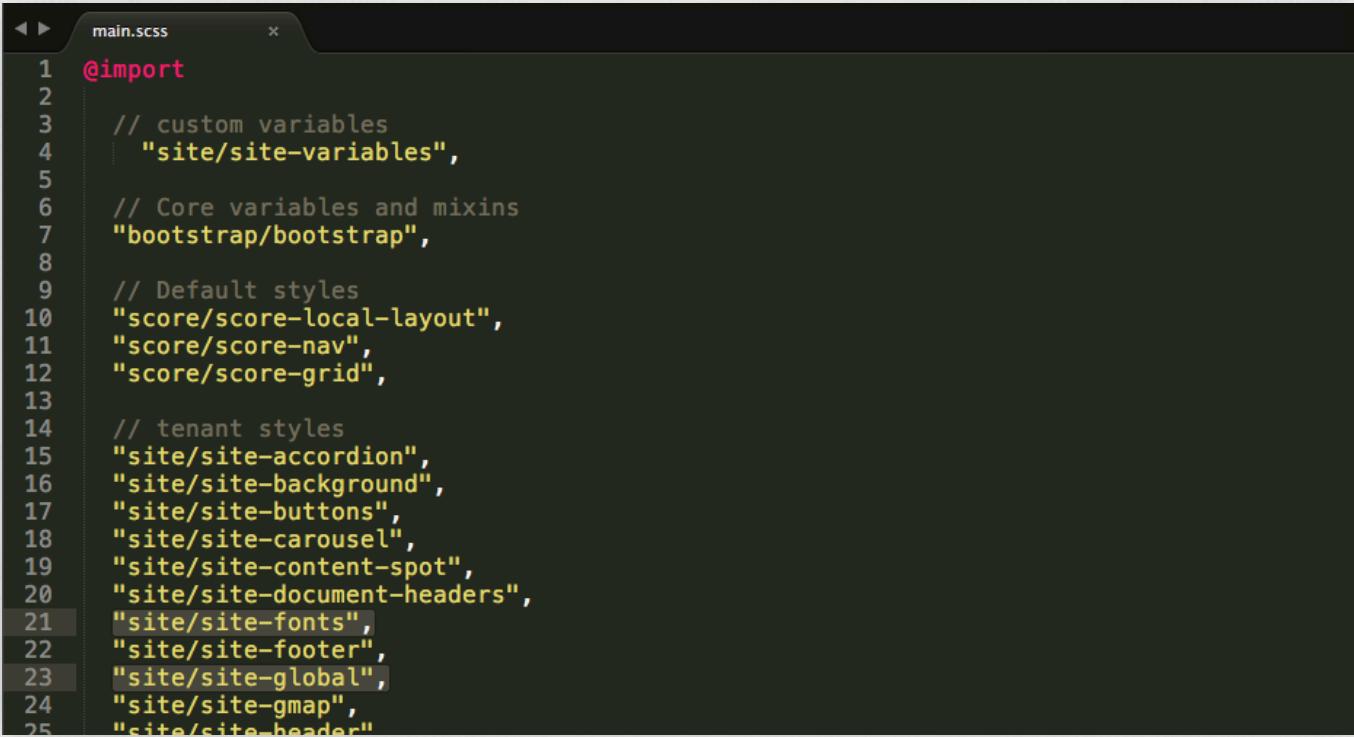
# Sass

- Include any necessary scss files to help keep styles organized  
examples: `_site-fonts.scss`, `_site-global.scss`, `_site-custom-component-name.scss`



# Sass

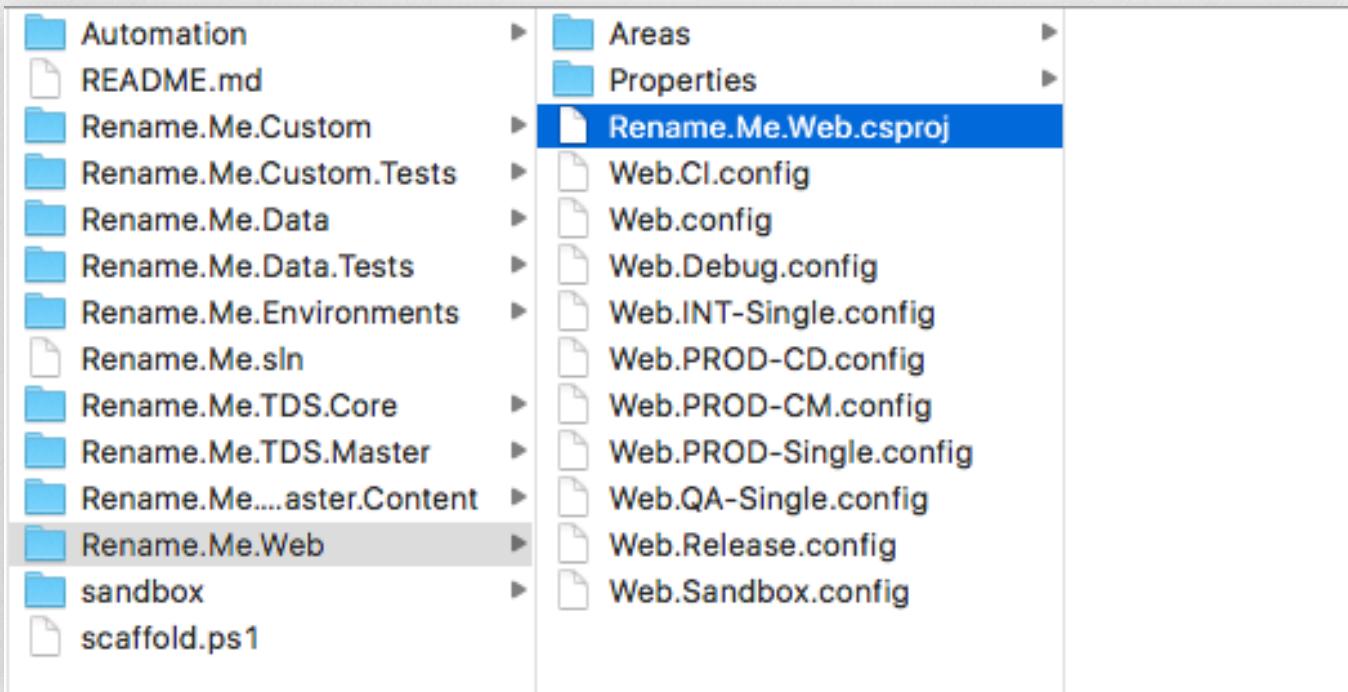
- Must be imported into main.scss



```
main.scss
1 @import
2
3   // custom variables
4   "site/site-variables",
5
6   // Core variables and mixins
7   "bootstrap/bootstrap",
8
9   // Default styles
10  "score/score-local-layout",
11  "score/score-nav",
12  "score/score-grid",
13
14  // tenant styles
15  "site/site-accordion",
16  "site/site-background",
17  "site/site-buttons",
18  "site/site-carousel",
19  "site/site-content-spot",
20  "site/site-document-headers",
21  "site/site-fonts",
22  "site/site-footer",
23  "site/site-global",
24  "site/site-gmap",
25  "site/site-header"
```

# Sass

- Must be included into project file (.csproj)



# Sass

- Must be included into project file (.csproj)



```
Score.Training.Content.Web.csproj
560    <Content Include="Areas\TrainingContent\scss\site\_site-gmap.scss" />
561    <Content Include="Areas\TrainingContent\scss\site\_site-header.scss" />
562    <Content Include="Areas\TrainingContent\scss\site\_site-heroes.scss" />
563    <Content Include="Areas\TrainingContent\scss\site\_site-highlights.scss" />
564    <Content Include="Areas\TrainingContent\scss\site\_site-image.scss" />
565    <Content Include="Areas\TrainingContent\scss\site\_site-lead.scss" />
566    <Content Include="Areas\TrainingContent\scss\site\_site-list-group.scss" />
567    <Content Include="Areas\TrainingContent\scss\site\_site-navigation.scss" />
568    <Content Include="Areas\TrainingContent\scss\site\_site-pannels.scss" />
569    <Content Include="Areas\TrainingContent\scss\site\_site-progress.scss" />
570    <Content Include="Areas\TrainingContent\scss\site\_site-section-header.scss" />
571    <Content Include="Areas\TrainingContent\scss\site\_site-social.scss" />
572    <Content Include="Areas\TrainingContent\scss\site\_site-stripe.scss" />
573    <Content Include="Areas\TrainingContent\scss\site\_site-tabset.scss" />
574    <Content Include="Areas\TrainingContent\scss\site\_site-variables.scss" />
575    <Content Include="Areas\TrainingContent\scss\site\_site-youtube-video.scss" />
576    <Content Include="Areas\TrainingContent\less\score\score-grid.less" />
577    <Content Include=
      Areas\TrainingContent\fonts\bootstrap\glyphicon-halflings-regular.woff2" />
578    <Content Include="Areas\TrainingContent\scss\site\_site-fonts.scss" />
579    <Content Include="Areas\TrainingContent\scss\site\_site-global.scss" />
580    <None Include="Web.config" />
581    <None Include="Web.Debug.config">
582        <DependentUpon>Web.config</DependentUpon>
```

# LAB 2

# RequireJS

05

# RequireJS

## What is RequireJS?

- JavaScript file and module loader
  - Optimized for in-browser use, but it can be used in other JS environments
- References: [RequireJS](#)

# RequireJS

Tabs (desktop) → Accordion (mobile)

Explore the Cities      Enjoy the Outside

Become immersed in city life!

Explore the cities of Puerto Rico and become immediately immersed in its beautiful and vibrant culture. The cities are filled with beatiful buildings, art and delicious food.



Explore the Cities

Become immersed in city life!

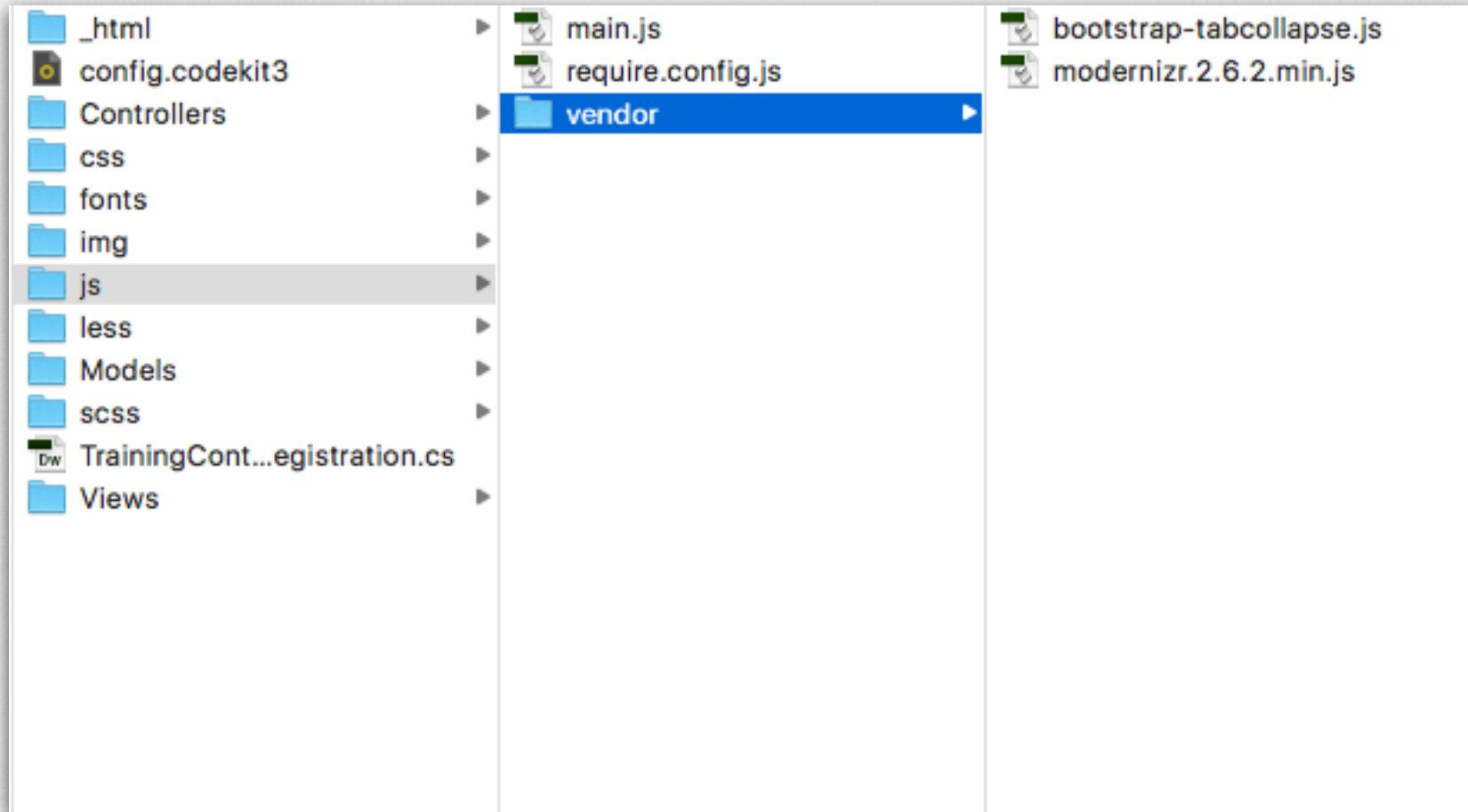
Explore the cities of Puerto Rico and become immediately immersed in its beautiful and vibrant culture. The cities are filled with beatiful buildings, art and delicious food.



Enjoy the Outside

# RequireJS

bootstrap-tabcollapse.js



# RequireJS

- **require.config.js** - Include path to 3rd party JS file



```
require.config.js *  
1 window.require = window.require || {};  
2 window.require.baseUrl = "/Areas/RenameMeArea/js";  
3  
4 window.require.paths = window.require.paths || {};  
5 window.require.paths.bootstrap = "/Areas/ScoreBootstrapUI/js/Vendor/bootstrap.min";  
6 window.require.paths.matchHeight = "/Areas/ScoreBootstrapUI/js/Vendor/jquery.matchHeight-min";  
7 window.require.paths.scorebootstrap = "/Areas/ScoreBootstrapUI/js/BootstrapUI";  
8 window.require.paths.jqueryValidate = "/Areas/ScoreBootstrapUI/js/Vendor/jquery.validate.1.12.0";  
9 window.require.paths.jqueryUnobtrusiveAjax = "/Areas/ScoreBootstrapUI/js/Vendor/jquery.unobtrusive-ajax";  
10 window.require.paths.jqueryValidateUnobtrusive = "/Areas/ScoreBootstrapUI/js/Vendor/jquery.validate.unobtrusive";  
11 window.require.paths.tabCollapse = "/Areas/TrainingContent/js/Vendor/bootstrap-tabcollapse";  
12  
13 window.require.shim = window.require.shim || {};  
14 window.require.shim.bootstrap = { deps: ["jquery"] };  
15 window.require.shim.matchHeight = { deps: ["jquery"] };  
16 window.require.shim.jqueryValidate = { deps: ["jquery"] };  
17 window.require.shim.jqueryUnobtrusiveAjax = { deps: ["jquery"] };  
18 window.require.shim.tabCollapse = { deps: ["jquery"] };  
19 window.require.shim.jqueryValidateUnobtrusive = { deps: ["jqueryValidate"] };  
20 window.require.shim.scorevalidation = { deps: ["jqueryUnobtrusiveAjax", "jqueryValidateUnobtrusive"] };
```

# RequireJS

- **require.config.js** - Include any dependancies (jQuery)



```
require.config.js *  
1 window.require = window.require || {};  
2 window.require.baseUrl = "/Areas/RenameMeArea/js";  
3  
4 window.require.paths = window.require.paths || {};  
5 window.require.paths.bootstrap = "/Areas/ScoreBootstrapUI/js/Vendor/bootstrap.min";  
6 window.require.paths.matchHeight = "/Areas/ScoreBootstrapUI/js/Vendor/jquery.matchHeight-min";  
7 window.require.paths.scorebootstrap = "/Areas/ScoreBootstrapUI/js/BootstrapUI";  
8 window.require.paths.jqueryValidate = "/Areas/ScoreBootstrapUI/js/Vendor/jquery.validate.1.12.0";  
9 window.require.paths.jqueryUnobtrusiveAjax = "/Areas/ScoreBootstrapUI/js/Vendor/jquery.unobtrusive-ajax";  
10 window.require.paths.jqueryValidateUnobtrusive = "/Areas/ScoreBootstrapUI/js/Vendor/jquery.validate.unobtrusive";  
11 window.require.paths.tabCollapse = "/Areas/TrainingContent/js/Vendor/bootstrap-tabcollapse";  
12  
13 window.require.shim = window.require.shim || {};  
14 window.require.shim.bootstrap = { deps: ["jquery"] };  
15 window.require.shim.matchHeight = { deps: ["jquery"] };  
16 window.require.shim.jqueryValidate = { deps: ["jquery"] };  
17 window.require.shim.jqueryUnobtrusiveAjax = { deps: ["jquery"] };  
18 window.require.shim.tabCollapse = { deps: ["jquery"] };  
19 window.require.shim.jqueryValidateUnobtrusive = { deps: ["jqueryValidate"] };  
20 window.require.shim.scorevalidation = { deps: ["jqueryUnobtrusiveAjax", "jqueryValidateUnobtrusive"] };
```

# RequireJS

- main.js - Define your module



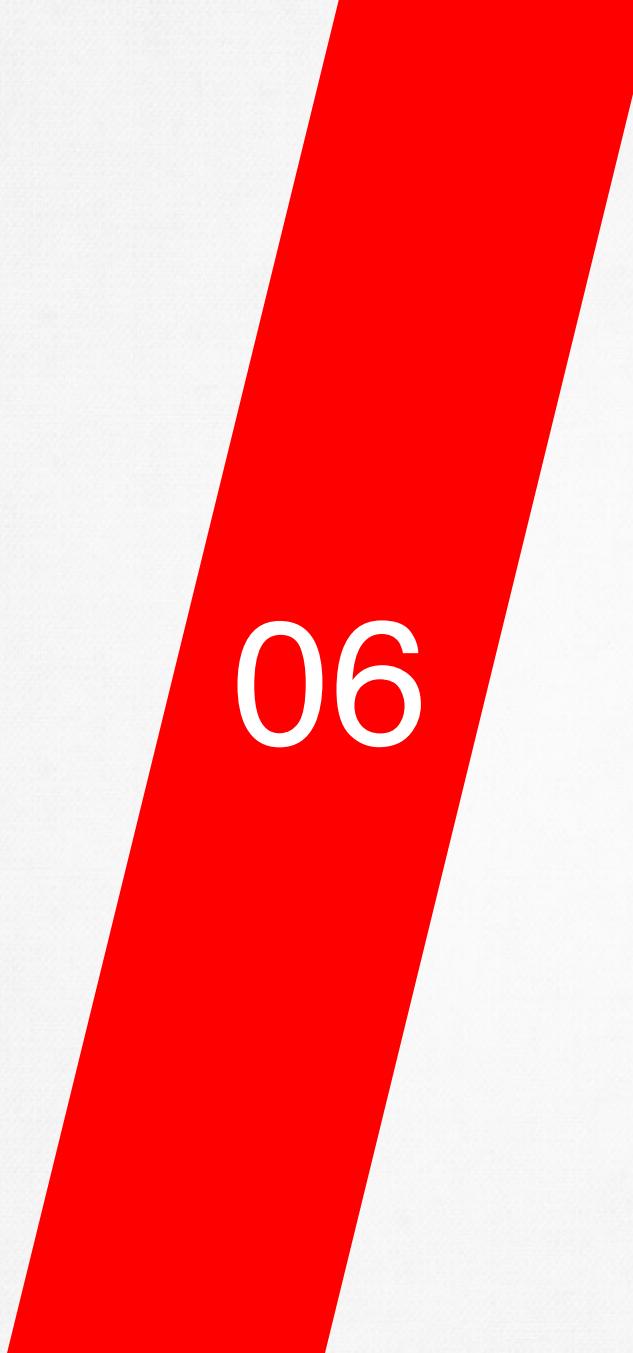
```
main.js
1 define([
2   "jquery",
3   "underscore",
4   "score_ccf/ModuleLoader",
5   "scorebootstrap",
6   "matchHeight",
7   "tabCollapse"
8 ],
9 function ($, _, moduleLoader, scorebootstrap) {
10   "use strict";
11
12   //prevents dropdown from closing when clicked inside
13   $(document).on("click", ".score-megamenu .dropdown-menu", function (e) {
14     e.stopPropagation();
15   });
16 }
```

# RequireJS

- main.js - Initialize

```
main.js
1 define([
2     "jquery",
3     "underscore",
4     "score_ccf/ModuleLoader",
5     "scorebootstrap",
6     "matchHeight",
7     "tabCollapse"
8 ],
9 function ($, _, moduleLoader, scorebootstrap) {
10     "use strict";
11
12     //prevents dropdown from closing when clicked inside
13     $(document).on("click", ".score-megamenu .dropdown-menu", function (e) {
14         e.stopPropagation();
15     });
16
17     //initialize match height
18     $(".score-style-box").matchHeight();
19     $(".score-highlight").matchHeight();
20
21     // if you need to monkey patch anything in Score Bootstrap Initialization logic
22     // here's where you would do it. Before calling to init(). Something like:
23     // scorebootstrap.Accordion.init = function() { ... }
24
25     // init SCORE Bootstrap components (Tabsets, Accordeons, Carousels, etc.)
26     scorebootstrap.init();
27
28     //turn tabs into accordion
29     $(".nav-tabs").tabCollapse();
30
31     moduleLoader.loadPendingModules().done(function() {
32         // anything that needs to run globally when all modules are loaded goes here
33     });
34 });

});
```



06

## Component HTML

Default component markup

# Component HTML

## The Basics

- Markup is intended to be portable
  - No boundaries for the grid
  - Unbound to any specific framework (w/ an exception for JS components)
- All SCORE components are simply prefixed with “score-”
  - Examples:
    - <div class=“score-page”>
    - <div class=“score-structural”>
    - <div class=“score-highlight”>
- All SCORE components have the ability for *class selections*
  - Examples:
    - <div class=“score-page custom-name-1”>
    - <div class=“score-page custom-name-2”>
    - <div class=“score-highlight custom-highlight-class”>

# Component HTML



## Review Confluence Documentation

- [SCORE Documentation](#)
  - Container
  - Content
  - Navigation
  - Social
- Includes example screenshot and HTML output
- Older versions are child pages of current version



6.1

# SCORE Containers

Understanding SCORE's grid system

# SCORE Containers



## Container Components

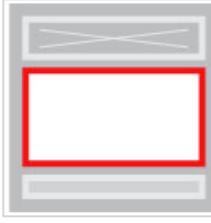
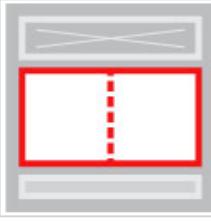
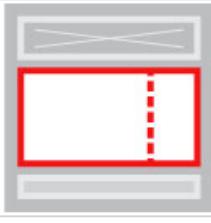
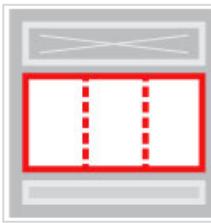
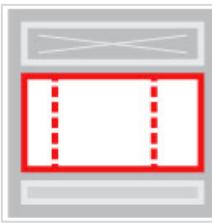
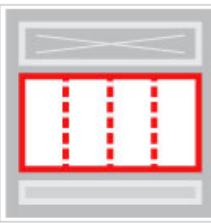
- Container components are simply components that hold other components (i.e. the grid)
- Container renderings:
  - Page Elements (Header, Footer)
  - Page Layout (.container + .row + .col-md-\*)
  - Inner Structure (.row + .col-md-\*)
  - Stripes
- Reference: [Container Documentation](#)

# SCORE Containers

## Page Layout

Select a Rendering  
Select the rendering that you want to use. Click Select to continue.

Open the Properties dialog box immediately.

CONTENT	DOCUMENT HEADERS	PAGE LAYOUT	STRIPES
	 WIDE SCREEN ONE COLUMN		
1 Column	1 Column Wide Screen	2 Column - Equal	2 Column - Large Left
			2 Column - Large Right
3 Column - Equal	3 Column - Large Middle	4 Column - Equal	

# SCORE Containers

## One-Column Page Layout Example:

- Bootstrap version on left - SCORE version on right
- Note the .container wrapper
- Page Layouts **CANNOT** be nested inside of each other

The image shows a code editor with two tabs open. The left tab is titled "bootstrap.html" and the right tab is titled "score.html". Both tabs have a dark background with light-colored text.

**bootstrap.html:**

```
1 <!-- Bootstrap row with columns -->
2 <div class="container">
3   <div class="row">
4     <div class="col-md-12">
5       ...
6     </div>
7   </div>
8 </div>
```

**score.html:**

```
1 <!-- SCORE one column -->
2 <div class="container">
3   <div class="score-column1">
4     <div class="score-center">
5       ...
6     </div>
7   </div>
8 </div>
```

# SCORE Containers



## One-Column Wide Screen Page Layout:

- Stretches edge to edge but still uses .container
- Used for components that can be full-width but can only be added to a container
- Example: Carousel

The screenshot shows a code editor window with the file name "index.html" at the top. The code is written in HTML and uses color-coded syntax highlighting. The code structure is as follows:

```
1 <div class="container full-width">
2   <div class="score-column1">
3     <div class="score-center"></div>
4   </div>
5 </div>
```

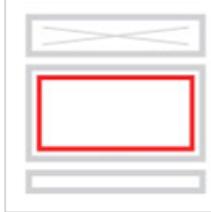
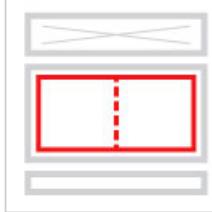
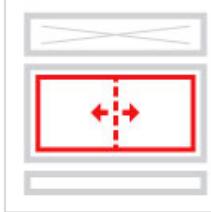
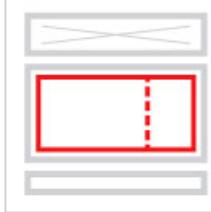
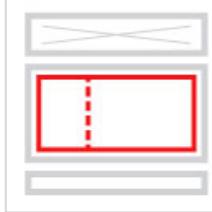
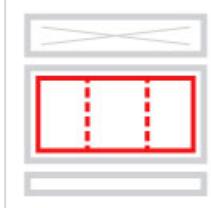
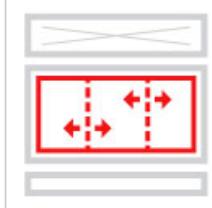
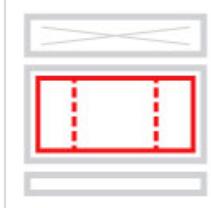
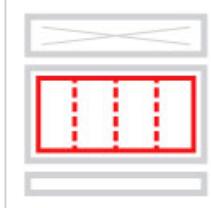
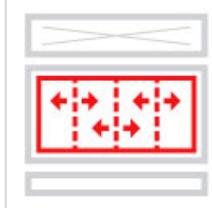
# SCORE Containers

## Inner Structure

Select a Rendering  
Select the rendering that you want to use. Click Select to continue.

Open the Properties dialog box immediately.

Buttons	Collections	Content	Document Headers	Features	Forms	INNER STRUCTURE
Navigation	Panels	Products	Search	Section Headers	Social	Stores

						
1 Column	2 Column - Equal	2 Column - Variable	2 Column - Wide Left	2 Column - Wide Right		
						
3 Column - Equal	3 Column - Variable	3 Column - Wide Middle	4 Column - Equal	4 Column - Variable		

# SCORE Containers

## Inner Structure Example:

- Bootstrap version on left - SCORE version on right
- Inner Structures must be nested inside of Page Layouts
- Inner Structures can be nested inside of each other

```
bootstrap.html
1  <!-- Bootstrap row with nested columns -->
2  <div class="container">
3    <div class="row">
4      <div class="col-md-12">
5        <div class="row">
6          <div class="col-md-6">
7            ...
8          </div>
9          <div class="col-md-6">
10         ...
11        </div>
12      </div>
13    </div>
14  </div>
15 </div>
```

```
score.html
1  <!-- SCORE row with nested columns -->
2  <div class="container">
3    <div class="score-column1">
4      <div class="score-center">
5        <div class="score-column2 equal">
6          <div class="score-left">
7            ...
8          </div>
9          <div class="score-right">
10         ...
11        </div>
12      </div>
13    </div>
14  </div>
15 </div>
```

# SCORE Containers

Stripe

Select a Rendering

Select the rendering that you want to use. Click Select to continue.

□ X

CONTENT DOCUMENT HEADERS PAGE LAYOUT STRIPES



Stripe

Open the Properties dialog box immediately.

Select Cancel

# SCORE Containers

## Stripe Example:

- SCORE stripe is a full-width wrapper
  - Used to separate SCORE page layouts for full-width backgrounds
- Background image/color and text color are applied as inline styles
- Page Layout is placed inside

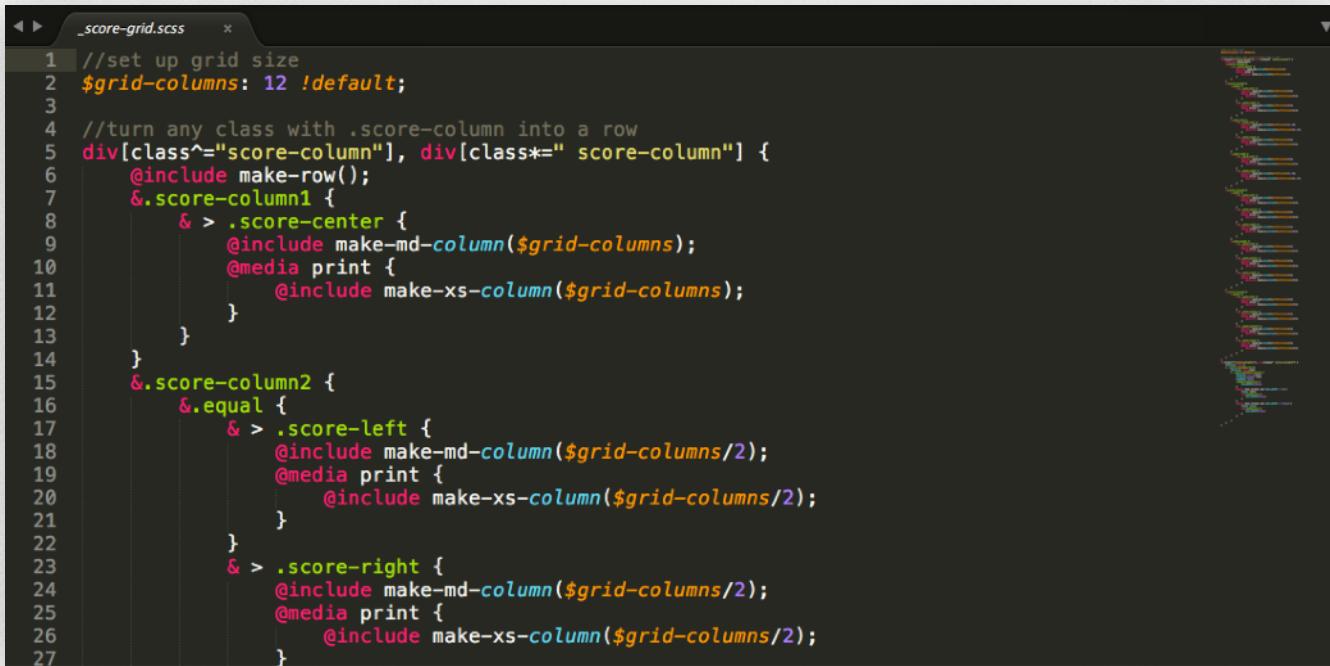
```
bootstrap.html: 1 <!-- Bootstrap full width wrapper -->
2 <div class="full-width-wrapper">
3   <div class="container">
4     <div class="row">
5       <div class="col-md-12">
6         ...
7       </div>
8     </div>
9   </div>
10 </div>
```

```
score.html: 1 <!-- SCORE stripe -->
2 <div class="score-stripe">
3   <div class="container">
4     <div class="score-column1">
5       <div class="score-center">
6         ...
7       </div>
8     </div>
9   </div>
10 </div>
```

# SCORE Containers

## Responsive Grid Mixins

- Grid mixins make or modify rows and columns at the CSS level
- These mixins can be used to create custom layouts
  - Change default 2-column wide-left from an 8/4 pair into a 7/5 or 9/3 pair
  - Make default medium columns into large, small, or extra-small columns



```
//set up grid size
$grid-columns: 12 !default;

//turn any class with .score-column into a row
div[class^="score-column"], div[class*=" score-column"] {
    @include make-row();
    &.score-column1 {
        & > .score-center {
            @include make-md-column($grid-columns);
            @media print {
                @include make-xs-column($grid-columns);
            }
        }
    }
    &.score-column2 {
        &.equal {
            & > .score-left {
                @include make-md-column($grid-columns/2);
                @media print {
                    @include make-xs-column($grid-columns/2);
                }
            }
            & > .score-right {
                @include make-md-column($grid-columns/2);
                @media print {
                    @include make-xs-column($grid-columns/2);
                }
            }
        }
    }
}
```

# LAB 3



6.2

# SCORE Content

Understanding SCORE's content components

# SCORE Content

## Content Components

- Content components cover a large variety of elements to deliver content and drive user experience (mostly comprised of Bootstrap's components as default)
- Content Renderings:
  - Buttons
  - Collections
  - Content
  - Document Headers
  - Features
  - Panels
  - Section Headers

# SCORE Content

Select a Rendering  
Select the rendering that you want to use. Click Select to continue.

BUTTONS    COLLECTIONS    **CONTENT**    DOCUMENT HEADERS    FEATURES    INNER STRUCTURE

NAVIGATION    PANELS    SECTION HEADERS    SOCIAL

Content Spot    Image    Youtube Video    Lead    Anchor Point

Reference a Snippet    Run a Snippet Macro    Simple Map    Map with Location Pins

Open the Properties dialog box immediately.

Select    Cancel

Reference: [Content Documentation](#)

# SCORE Content

REUSE!

- The key to gaining full advantage of SCORE is all about *reuse*.
  - Familiarize yourself with every SCORE component
  - Know when/how/where to use default components
  - Saves you time and money!
- Avoid creating custom components!
  - Custom components are usually inevitable (*but* you'll be surprised with what you can achieve with SCORE components)
  - Strive to make sure that custom components are your last resort
  - Use class selections to modify default components



6.3

# SCORE Navigation

Understanding SCORE's navigation components

# SCORE Navigation

## Navigation Components

- Navigation components create various types of UX navigation
- Navigation renderings:
  - Breadcrumb
  - Main Menu
  - Mega Menu
  - Menu List
  - My Siblings Menu
  - Section Menu Spider
  - Sitemap Spider
  - This Section Menu
- Reference: [Navigation Documentation](#)

# SCORE Navigation

Header

Navigation

Navbar

Menu Item

Menu Item

Menu Item

Navbar

Menu Item

Menu Item

Menu Item

Menu Item

# SCORE Navigation

## Header Rendering for Page Element

Select a Rendering  
Select the rendering that you want to use. Click Select to continue.

Open the Properties dialog box immediately.

CONTENT DOCUMENT HEADERS PAGE ELEMENTS STRIPES

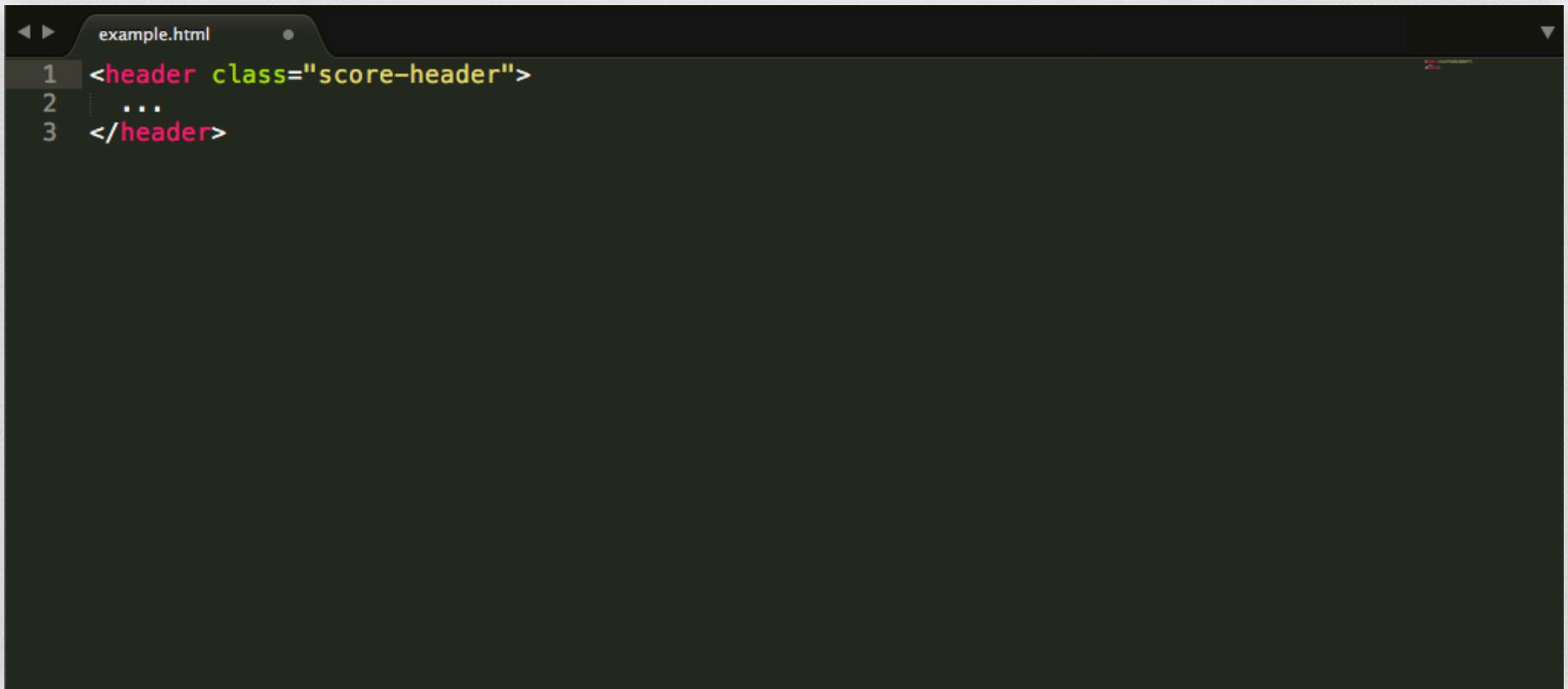
Header

# SCORE Navigation

Header

# SCORE Navigation

Header Page Element:



```
example.html
1 <header class="score-header">
2 ...
3 </header>
```

# SCORE Navigation

## Navigation Renderings for Header

Select a Rendering  
Select the rendering that you want to use. Click Select to continue.

□ X

CONTENT NAVIGATION PAGE LAYOUT

Mega Menu Main Menu Spider

Open the Properties dialog box immediately.

Select Cancel

# SCORE Navigation

Header

Navigation

# SCORE Navigation

## Megamenu Example:

```
example.html
1  <!-- <header class="score-header" -->
2  <nav class="score-megamenu">
3      <div class="score-nav-wrapper">
4          <div class="score-nav">
5              <div class="navbar-header score-navbar-header">
6                  <button type="button" data-toggle="collapse" data-target="#megaMenuID" class=
7                      "score-hamburger navbar-toggle">
8                      <span class="sr-only">Toggle navigation</span>
9                      <span class="icon-bar"></span>
10                     <span class="icon-bar"></span>
11                     <span class="icon-bar"></span>
12                 </button>
13                 <span class="navbar-brand"></span>
14             </div>
15             <div class="collapse navbar-collapse" id="megaMenuID">
16                 ...
17                 </div>
18             </div>
19         </div>
20     </nav>
21 <!-- </header> -->
```

# SCORE Navigation

## Megamenu Item Renderings for Navigation

Select a Rendering  
Select the rendering that you want to use. Click Select to continue.

Open the Properties dialog box immediately.

**MEGA MENU ITEMS**

Mega Menu Navbar Row

Mega Menu Navbar

Mega Menu Button

Mega Menu Button Menu

Mega Menu Split Button Menu

# SCORE Navigation

Header

Navigation

Navbar

# SCORE Navigation

## Megamenu Navbar Example:

```
example.html
1  <!-- <header class="score-header"> -->
2  <!--<nav class="score-megamenu">
3      <div class="score-nav-wrapper">
4          <div class="score-nav">
5              <div class="navbar-header score-navbar-header">
6                  <button type="button" data-toggle="collapse" data-target="#megaMenuID"
7                      class="score-hamburger navbar-toggle">
8                      <span class="sr-only">Toggle navigation</span>
9                      <span class="icon-bar"></span>
10                     <span class="icon-bar"></span>
11                     <span class="icon-bar"></span>
12                 </button>
13                 <span class="navbar-brand"></span>
14             </div>
15             <div class="collapse navbar-collapse" id="megaMenuID">-->
16                 <ul class="nav navbar-nav score-nav">
17                     ...
18                 </ul>
19             <!--</div>
20         </div>
21     </div>
22 </nav>-->
<!-- </header> -->
```

# SCORE Navigation

## Additional Navbar Megamenu Item Renderings for Navigation

Select a Rendering

Select the rendering that you want to use. Click Select to continue.

Open the Properties dialog box immediately.

MEGA MENU ITEMS

Mega Menu Navbar

Mega Menu Button

Mega Menu Button Menu

Mega Menu Split Button Menu

# SCORE Navigation

Header

Navigation

Navbar

Navbar

# SCORE Navigation

## Additional Megamenu Navbar Example:

The screenshot shows a code editor window with a dark theme. On the left, the file `example.html` is open, displaying the following HTML code:

```
1 <!-- <header class="score-header"> -->
2 <!--<nav class="score-megamenu">
3   <div class="score-nav-wrapper">
4     <div class="score-nav">
5       <div class="navbar-header score-navbar-header">
6         <button type="button" data-toggle="collapse" data-target="#score-navbar-collapse" class="score-hamburger navbar-toggle">
7           <span class="sr-only">Toggle navigation</span>
8           <span class="icon-bar"></span>
9           <span class="icon-bar"></span>
10          <span class="icon-bar"></span>
11        </button>
12        <span class="navbar-brand"></span>
13      </div>
14      <div class="collapse navbar-collapse" id="score-navbar-collapse">
15        <ul class="nav navbar-nav score-nav">
16          ...
17        </ul>
18        <ul class="nav navbar-nav score-nav navbar-right">
19          ...
20        </ul>
21      <!--</div>
22      |   </div>
23    </div>
24  </nav>-->
25 <!-- </header> -->
```

To the right of the code editor is a floating "Control Properties" dialog box. It has a dark header bar with a close button (). The main area is titled "Control Properties" and contains the following sections:

- Show Hide**: A dropdown menu currently set to "Menu".
- Menu**: A section containing a "Menu Style" dropdown and a "Position" dropdown set to "Right".
- Navbar**: A section containing a "Position" dropdown set to "Right".
- Caching**: A section containing a "Caching" dropdown.

At the bottom right of the dialog box are two buttons: "OK" and "Cancel".

# SCORE Navigation

## Menu Item Renderings for Navbar

Select a Rendering  
Select the rendering that you want to use. Click Select to continue.

□ X

NAVBAR MENU ITEMS

Basic Menu Item      Mega Menu Item      Mega Menu Item Divider      Drop Down List      Drop Down List Spider

Mega Menu Item with Link

Open the Properties dialog box immediately.

Select      Cancel

# SCORE Navigation

Header

Navigation

Navbar

Menu Item

# SCORE Navigation

## Basic Menu Item Example:

```
2 <!--<nav class="score-megamenu">
3     <div class="score-nav-wrapper">
4         <div class="score-nav">
5             <div class="navbar-header score-navbar-header">
6                 <button type="button" data-toggle="collapse" data-target="#megaMenuID"
7                     class="score-hamburger navbar-toggle">
8                     <span class="sr-only">Toggle navigation</span>
9                     <span class="icon-bar"></span>
10                    <span class="icon-bar"></span>
11                    <span class="icon-bar"></span>
12                </button>
13                <span class="navbar-brand"></span>
14            </div>
15            <div class="collapse navbar-collapse" id="megaMenuID">-->
16            <!--<ul class="nav navbar-nav score-nav">-->
17                <li class="score-megamenu-basic-item">
18                    <a href="/">Basic Menu Item</a>
19                </li>
20            <!--</ul>-->
21        <!--</div>
22    </div>
23 </nav>-->
```

# SCORE Navigation

## Menu Item Renderings for Navbar

Select a Rendering  
Select the rendering that you want to use. Click Select to continue.

□ X

NAVBAR MENU ITEMS

Basic Menu Item    Mega Menu Item    Mega Menu Item Divider    Drop Down List    Drop Down List Spider

Mega Menu Item with Link

Open the Properties dialog box immediately.

Select    Cancel

# SCORE Navigation

Header

Navigation

Navbar

Menu Item

Menu Item

# SCORE Navigation

## Megamenu Item Example:

```
8      <span class="icon-bar"></span>
9      <span class="icon-bar"></span>
10     <span class="icon-bar"></span>
11     </button>
12     <span class="navbar-brand"></span>
13   </div>
14   <div class="collapse navbar-collapse" id="megaMenuID">-->
15   <!--<ul class="nav navbar-nav score-nav">-->
16   <li class="score-megamenu-basic-item">
17     <a href="/">Basic Menu Item</a>
18   </li>
19   <li class="score-megamenu-dropdown">...</li>
20     <a href="#" data-toggle="dropdown">Megamenu Item</a>
21     <a href="#" class="dropdown-toggle toggle-caret">
22       <span class="caret"></span>
23       <span class="sr-only">toggle menu</span>
24     </a>
25     <ul class="dropdown-menu" role="menu">
26       <li>...
27         <div class="score-megamenu-content">
28           ...
29         </div>
30       </li>
31     </ul>
32   </li>
33   <!--</ul>-->
34 <!--</div>
35 </div>
36
```

Megamenu Item ▾

# SCORE Navigation

## Navigation Renderings

Select a Rendering  
Select the rendering that you want to use. Click Select to continue.

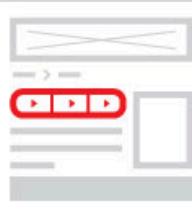
Open the Properties dialog box immediately.

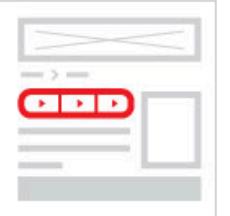
BUTTONS	COLLECTIONS	CONTENT	DOCUMENT HEADERS	FEATURES	INNER STRUCTURE
NAVIGATION	PANELS	SECTION HEADERS	SOCIAL		

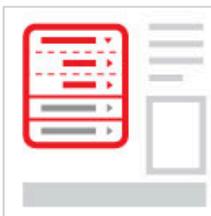
  
Menu List

  
Section Menu Spider

  
Breadcrumb

  
Previous Next Siblings Menu

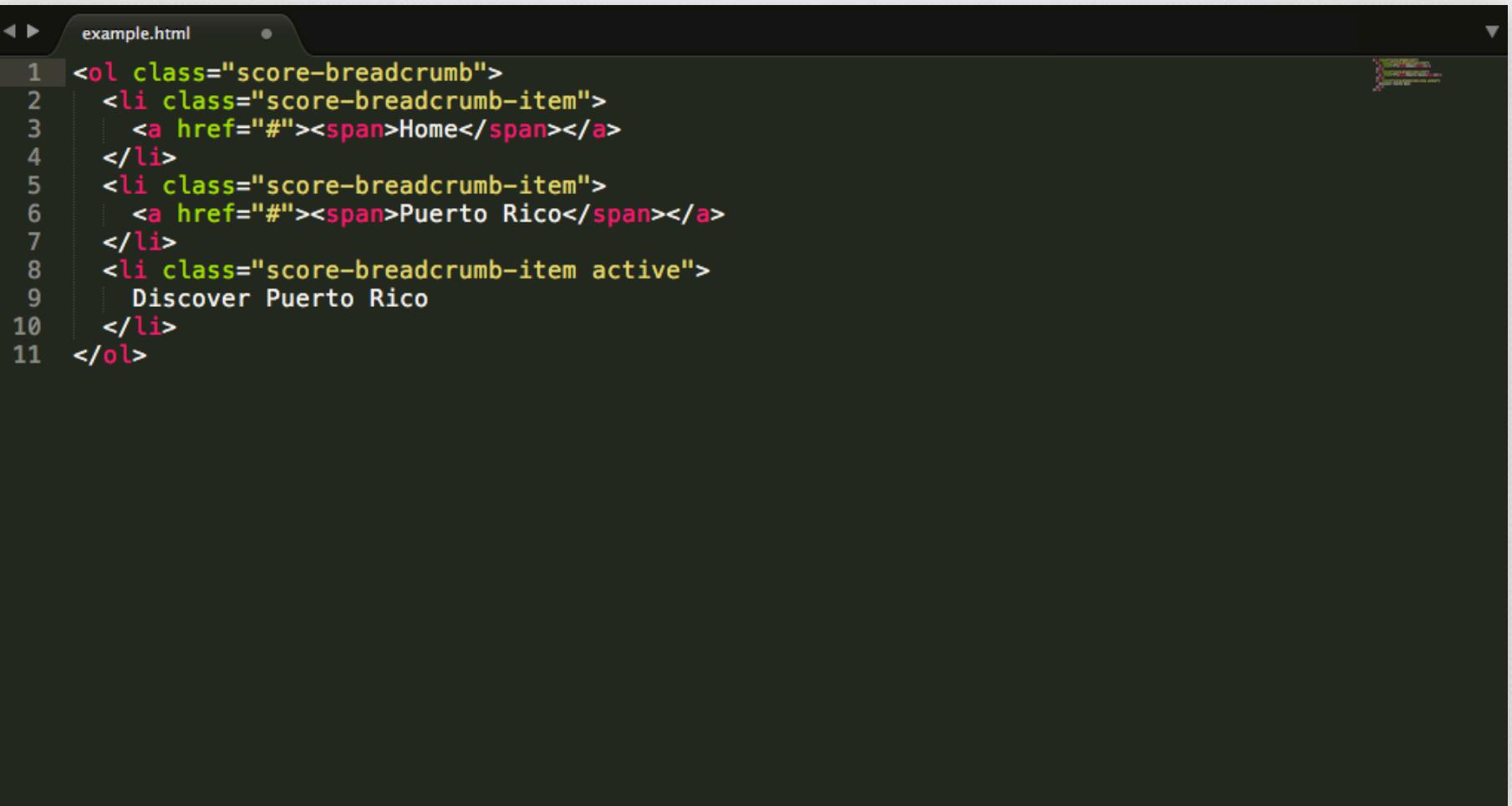
  
My Siblings Menu

  
This Section Menu

  
Siteman Spider

# SCORE Navigation

Breadcrumb Example:



A screenshot of a code editor window titled "example.html". The code editor has a dark theme with syntax highlighting. The code itself is an ordered list (`<ol>`) with the class "score-breadcrumb". It contains three list items (`<li>`). The first two items have the class "score-breadcrumb-item" and contain anchor tags (`<a href="#">`) with spans (`<span>`) containing the text "Home" and "Puerto Rico" respectively. The third item has the class "score-breadcrumb-item active" and contains the text "Discover Puerto Rico". The code is numbered from 1 to 11 on the left side.

```
1 <ol class="score-breadcrumb">
2   <li class="score-breadcrumb-item">
3     <a href="#"><span>Home</span></a>
4   </li>
5   <li class="score-breadcrumb-item">
6     <a href="#"><span>Puerto Rico</span></a>
7   </li>
8   <li class="score-breadcrumb-item active">
9     Discover Puerto Rico
10  </li>
11 </ol>
```

# Class Selections

07

# SCORE Content

## Highlight with Class Selection Example:

- Generated HTML on left - Stylesheet on right

The screenshot shows a code editor with two panes. The left pane displays the generated HTML code for a 'score-highlight' component, while the right pane shows the corresponding SCSS stylesheet.

**example.html**

```
1 <div class="score-highlight all-new-class">
2   <div class="score-highlight-header"></div>
3   
4   <div class="caption">
5     <h2>...</h2>
6     <div class="score-highlight-body">
7       ...
8     </div>
9     <div class="score-call-to-action">
10    ...
11   </div>
12 </div>
13 </div>
```

**\_site-highlights.scss**

```
1 .score-highlight {
2   @extend .thumbnail; //inherit bootstrap
3   @include vertical-space($vertical-spacing);
4   .score-highlight-header {
5   }
6   img {
7   }
8   }
9   .caption {
10    h2 {
11    }
12    }
13    .score-highlight-body {
14    }
15    }
16    .score-call-to-action {
17    }
18    }
19    }
20  }
21  &.all-new-class {
22    //Class selection styles here
23  }
24  }
25 }
```

# SCORE Content

Highlight with Class Selection Example:



Visit Scotland

[LEARN MORE](#)



## Arts and Music

Explore the dozens of museums and art galleries. If you are out at night, there can be live music found everywhere.

# Class Selections

## Background sizes & repeats

- Background styles dropdown is pre-populated with SCORE values
- Classes are stored in `_site-background.scss`
  - **Background classes are only available for components with a field for background image**

```
.score-stripe, .score-style-box, .score-carousel-pane, .score-full-width-hero, .score-hero, .score-section-header, .score-well {  
  &.cover {  
    background-position:center center;  
    background-repeat:no-repeat;  
    background-size:cover;  
  }  
  &.contain {  
    background-repeat:no-repeat;  
    background-size:contain;  
  }  
  &.repeat {  
    background-repeat:repeat;  
  }  
  &.repeat-x {  
    background-repeat:repeat-x;  
  }  
  &.repeat-y {  
    background-repeat:repeat-y;  
  }  
}
```

# Class Selections

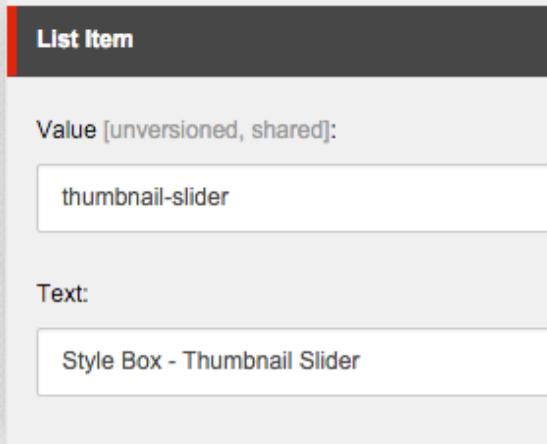
## Style Box Component

- Combination of a content component and a container component
- Will hold almost any other SCORE component
- Has ability for background size/repeat, background image, background color, text color, and class selection
- Can be used simply as a wrapper class

# Class Selections

## Using class selections with JS

- Targeting class selections in JS can further reduce the need for custom components
- Example: Slider on BrainJuice site using [slick.js](#)



```
$('.thumbnail-slider').slick({
  infinite: true,
  slidesToShow: 3,
  slidesToScroll: 3,
  responsive: [
    {
      breakpoint: 1300,
      settings: {
        slidesToShow: 2,
        slidesToScroll: 2,
        infinite: true,
        dots: true
      }
    },
    {
      breakpoint: 870,
      settings: {
        slidesToShow: 1,
        slidesToScroll: 1
      }
    }
  ]
});
```

# LAB 4

08

## Where To Start?

Approaches to project development



# Design Analysis

Communication is key!

- Work with content administrators and designers to determine SCORE and custom components
- Be agile - components may change throughout the project based on requirements
  - Example: dynamic vs. manual components
- Be familiar with components - know what is available in HTML



# Where to start?

Build FIRST and style LAST

- HTML for containers, content, and navigation are ready
- Predefined class selections have been made and are in place
- Foreground and background images



# Where to start?

Build full composition pages **or** standalone components

- All of which are created/stored in \_html “Sandbox” folder for use
- Build/save individual HTML pages as POCs and keep them maintained
- Build/save an HTML page with all components and keep it maintained
- Create custom component HTML to pass along to Sitecore developers

09

## SCORE vs. Custom Components

# Design Analysis - SCORE vs. Custom

- Images here - compare SCORE vs. custom



CHRISTIAN LIVING

Counting Stars When You See None

COUNTING STARS WHEN  
YOU SEE NONE - PART 5

Body text for a custom highlight component. Lorem ipsum dolor sit amet consectetur lorem.



## Explore the Cities

Puerto Rican cities are filled with beautiful colors and architecture.

[LEARN MORE](#)

# Thank you

Ben Cato

[ben@brainjocks.com](mailto:ben@brainjocks.com)

Emily Lord

[emily@brainjocks.com](mailto:emily@brainjocks.com)

