



April 2015

Styling BrainJocks SCORE

Ben Cato & Emily Lord

BRAINJOCKS.COM

Course Overview

01

Overview of class pre-requisites – Twitter Bootstrap 3, LESS pre-processor, SCORE component documentation

02

Understanding SCORE's file structure – default project scaffolding and LESS file organization

03

Styling SCORE components – page and structural layouts, responsive grid mixins, overriding Bootstrap

04

General front-end best practices

01

Overview of Pre-requisites

BOOTSTRAP 3, LESS PRE-PROCESSOR, &
SCORE COMPONENT DOCUMENTATION

01.1

Bootstrap 3 Overview

What is Bootstrap?

- Naturally responsive UI framework
- Based on LESS 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
- References:
 - [Bootstrap 3 Site](#)
 - [Bootstrap Tutorial](#)

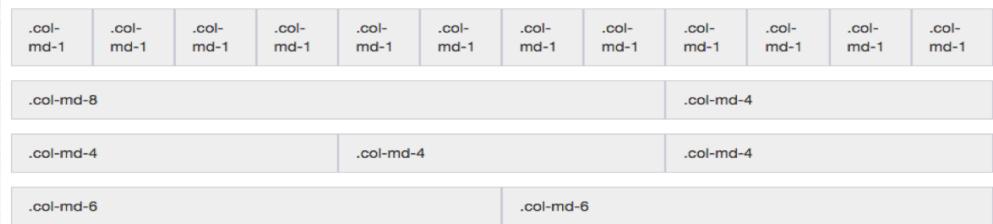


Using the Grid

- 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
- References:
 - [How Bootstrap's Grid Works](#)



Using the Grid



```
<div class="container">
  <div class="row">
    <div class="col-md-1"></div>
    <div class="col-md-1"></div>
  </div>

  <div class="row">
    <div class="col-md-8"></div>
    <div class="col-md-4"></div>
  </div>

  <div class="row">
    <div class="col-md-4"></div>
    <div class="col-md-4"></div>
    <div class="col-md-4"></div>
  </div>

  <div class="row">
    <div class="col-md-6"></div>
    <div class="col-md-6"></div>
  </div>
</div>
```

The Anatomy of Bootstrap's Grid

- Container:
 - Width based on screen size
 - 15px padding on left and right
 - Margin: 0 auto (always at the center of the page)

Safari bug warning!

As of v8.0, Safari exhibits a bug in which resizing your browser horizontally causes rendering errors in the justified nav that are cleared upon refreshing.

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.

[View details >](#)

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

Safari bug warning!

As of v8.0, Safari exhibits a bug in which resizing your browser horizontally causes rendering errors in the justified nav that are cleared upon refreshing.

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.

[View details »](#)

The Anatomy of Bootstrap's Grid

- Column:
 - Width is specified by developer using a class or mixin
 - NOTE: Bootstrap uses 12-column grid
 - 15px padding on left and right (30px between columns)
 - ALWAYS inserted into a row

Safari bug warning!

As of v8.0, Safari exhibits a bug in which resizing your browser horizontally causes rendering errors in the justified nav that are cleared upon refreshing.

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-lg-4 390px x 287px

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-lg-4 390px x 287px

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-lg-4 390px x 287px

The Anatomy of Bootstrap's Grid

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

Safari bug warning!

As of v8.0, Safari exhibits a bug in which resizing your browser horizontally causes rendering errors in the justified nav that are cleared upon refreshing.

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.

[View details »](#)

Heading

Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egestas eget quam.

[View details »](#)

`div.row 780px x 157px`

The Anatomy of Bootstrap's Grid

- Nested row with columns:
 - Insert columns into the row like normal
 - Still .col-md-6, will be half the width of parent row/column

Safari bug warning!

As of v8.0, Safari exhibits a bug in which resizing your browser horizontally causes rendering errors in the justified nav that are cleared upon refreshing.

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.

[View details »](#)

`div.col-lg-6 390px x 157px`

Heading

Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egestas eget quam.

[View details »](#)

`div.col-lg-6 390px x 157px`

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
 - Columns stack in order of leftmost column becomes top column, columns to the right stack below in order



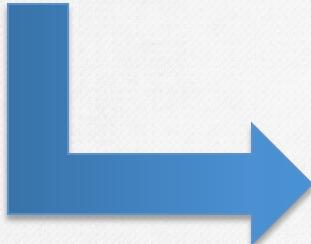
Mobile Breakpoints

Safari bug warning!

As of v8.0, Safari exhibits a bug in which resizing your browser horizontally causes rendering errors in the justified nav that are cleared upon refreshing.

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 »](#)



Heading1

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.

[View details »](#)

Safari bug warning!

As of v8.0, Safari exhibits a bug in which resizing your browser horizontally causes rendering errors in the justified nav that are cleared upon refreshing.

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 »](#)

Heading1

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.

[View details »](#)

Column Operations

1. Offset

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

2. Order

- a. Switches order of columns
- b. Uses classes .col-md-push-* and .col-md-pull-*
 - i. Example:

.col-md-3 .col-md-pull-9	.col-md-9 .col-md-push-3
--------------------------	--------------------------

```
<div class="row">
  <div class="col-md-9 col-md-push-3".col-md-9 .col-md-push-3</div>
  <div class="col-md-3 col-md-pull-9".col-md-3 .col-md-pull-9</div>
</div>
```

Copy

Column Operations

3. Responsive utilities

- a. Show or hide columns based on viewport width
- b. Uses classes `.visible-**` or `.hidden-**`
 - i. Example: `.hidden-md` is hidden from 992px – 1200px

	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

Native Components

- Bootstrap styles native components using specific class names: nav, navbar, carousel, jumbotron, tabs, accordions, etc.
 - Note: by default, SCORE components inherit Bootstrap's native component styles
- Bootstrap.js included in project scaffolding for use with components that require JavaScript
 - Carousel, tabs, accordions, navigation
- References:
 - [Bootstrap 3 Components](#)
 - [JavaScript Components](#)

Bootstrap Markup

- Not portable or fluid when used with components
- Front-end dev has no control over column sizes, etc.
 - Set in HTML for the component and is difficult to change
- Project committed to a 12-column grid



SCORE Markup

- SCORE uses container, content, and navigation components
 - Container components hold other components
- Fixed set of HTML based on Bootstrap
 - HTML generated from inserting component in Sitecore
 - Cannot change for SCORE components
 - Custom components use custom HTML
- Significantly increased flexibility for content administrators
- Markup is portable and can be manipulated by front-end developer

SCORE Component Docs

- [Parent page for Bootstrap UI components](#)
 - Containers
 - Content
 - Navigation
- Includes example screenshot and HTML output



01.2

LESS Overview

LESS Basics

- CSS pre-processor (compiles into CSS)
 - All valid CSS is valid LESS
- Makes writing CSS easier with several added features
- Allows nesting, which is amazing – but can easily get out of control
 - Try to keep nesting no more than 4 levels deep
- Reference: [LESS Tutorial](#)



Good Nesting Example



LESS:

```
1  @ul {
2      list-style: none;
3
4      li {
5          float: left;
6
7          a {
8              color: #foo;
9
10     }
11
12
13
14
15
16}
```

CSS:

```
1  @ul {
2      list-style: none;
3
4
5      li {
6          float: left;
7
8
9      a {
10         color: #foo;
11
12
13
14
15
16}
```

Bad Nesting Example



LESS:

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

CSS:

```
1  .container .row-class .column-class ul li a {  
2    color: #foo;  
3  }  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16
```

LESS 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

```
1  @brand-color: #foo;
2
3  ↪header, footer {
4      background-color: @brand-color;
5  }
6
7  ↪h1 {
8      color: @brand-color;
9  }
10
```

LESS Mixins

- 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

```
1  ↵.rounded-corners(@radius) {  
2      -webkit-border-radius: @radius;  
3      -moz-border-radius: @radius;  
4      -o-border-radius: @radius;  
5      border-radius: @radius;  
6  ↵}  
7  
8  ↵.button {  
9      .rounded-corners(5px);  
10 ↵}  
11
```

LESS Operators

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

```
//LESS:          //CSS:  
.class1 {      .class1.class1-variant {  
  &.class1-variant {    color: #foo;  
    color: #foo;  
  }  
}  
}
```

```
//LESS          //CSS  
.button {      .button:first-child {  
  &:first-child {    border-radius: 4px;  
    border-radius: 4px;  
  }  
  &:hover {        color: #foo;  
    color: #foo;  
  }  
}
```

LESS Operators

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

```
//LESS                                //CSS
.class1 {                            .class1 > .class2 {
    & > .class2 {                      color: #foo;
        |   color: #foo;           }
    | }
}
```

Nesting Media Queries

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

```
//LESS
.class1 {
  @media only screen and (min-width: 768px) {
    color: #foo;
  }
  @media only screen and (min-width: 992px) {
    color: #fff;
  }
}
```

```
//CSS
@media only screen and (min-width: 768px) {
  .class1 {
    color: #foo;
  }
}
@media only screen and (min-width: 992px) {
  .class1 {
    color: #fff;
  }
}
```

LESS Compiling



Compiler
→

CSS

LESS:

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

Compiler
→

CSS:

```
1  ul {  
2      list-style: none;  
3  }  
4    
5  ul li {  
6      float: left;  
7  }  
8    
9  ul li a {  
10     color: #foo;  
11  }  
12 }
```

LESS Compilers

COMMAND LINE



MAC



WINDOWS



REFERENCE: [List of popular compilers](#)



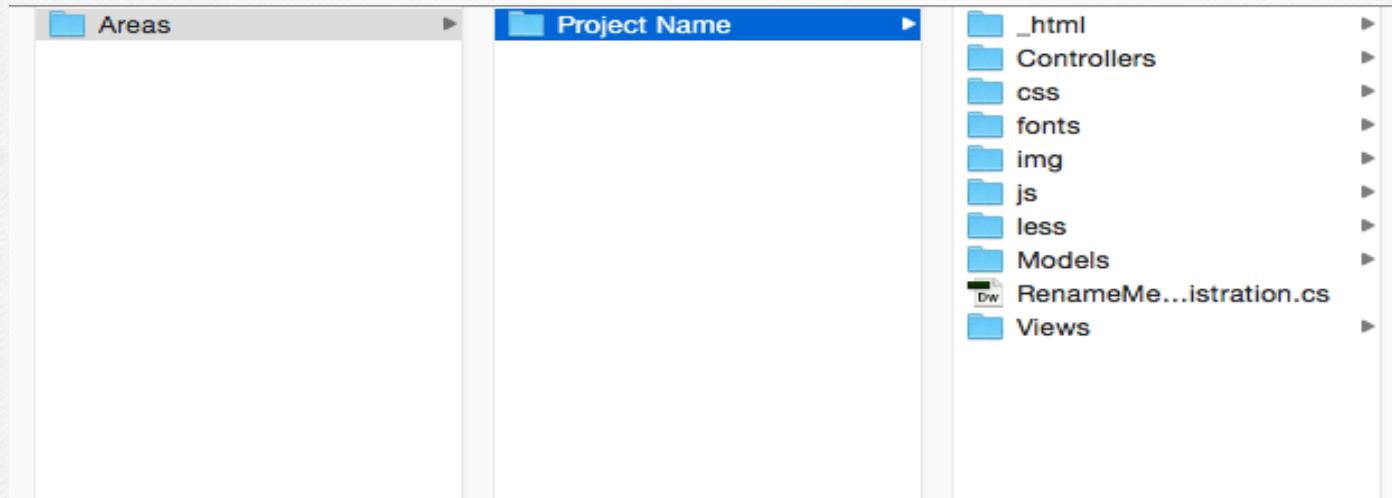
02

Understanding SCORE's File Structure

Organization of project files, LESS files, and HTML sandbox.

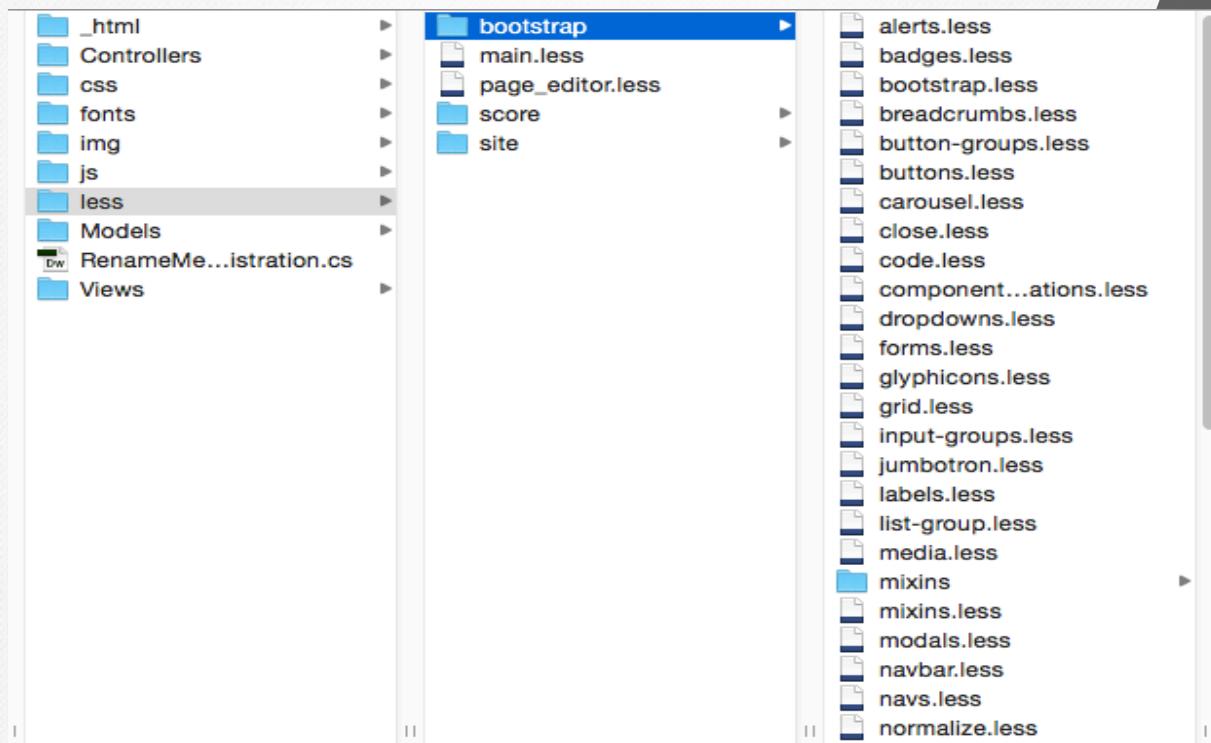
“Areas” Scaffolding

- ALL project files are SHARED between Sitecore and UI developers



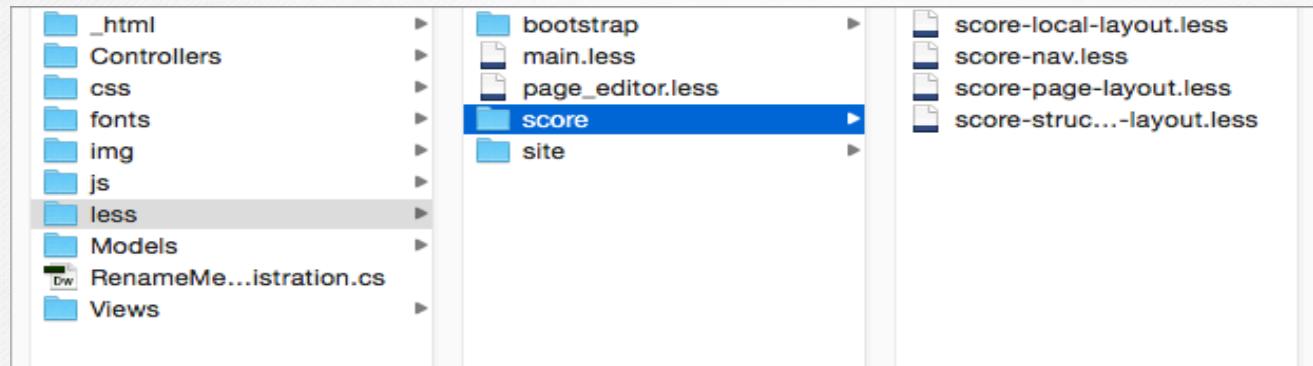
Bootstrap Folder

- Contains all of Bootstrap's default LESS files
- Location to access Bootstrap's variables and mixins, if needed



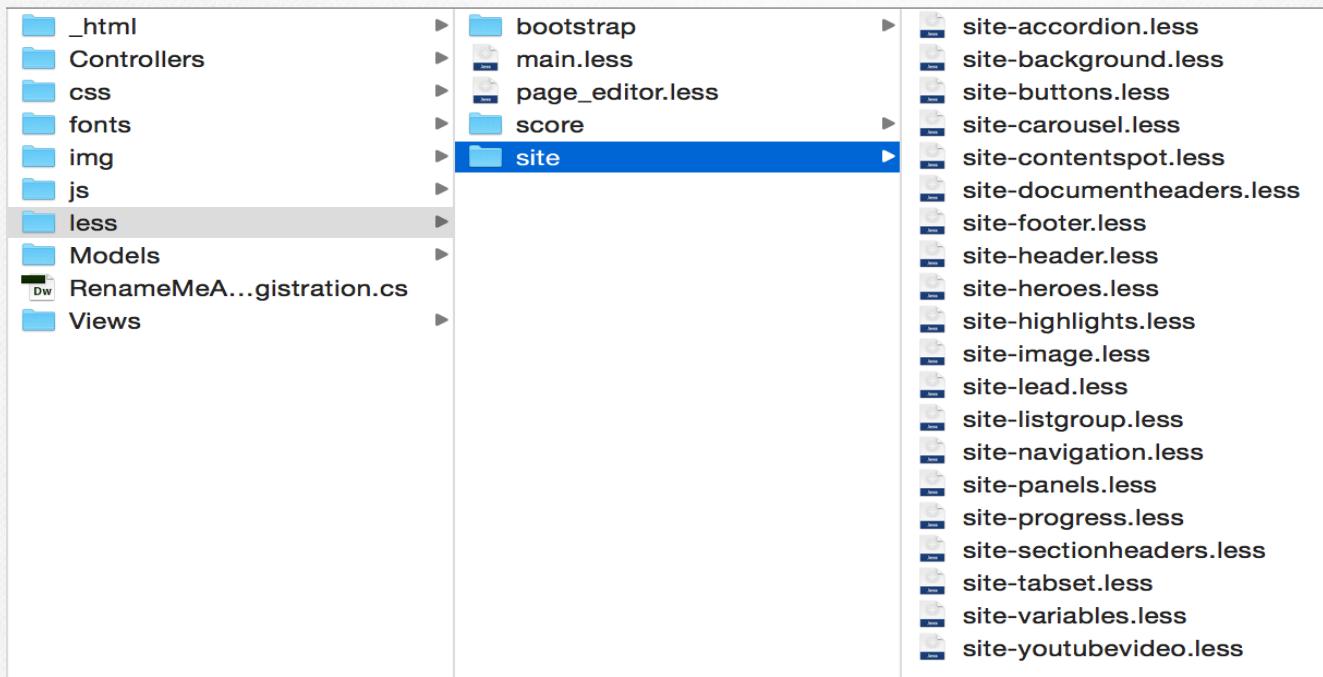
SCORE Folder

- Contains all of SCORE's default LESS files
- Location to modify SCORE-specific styles, if needed
 - Example: score-nav.less



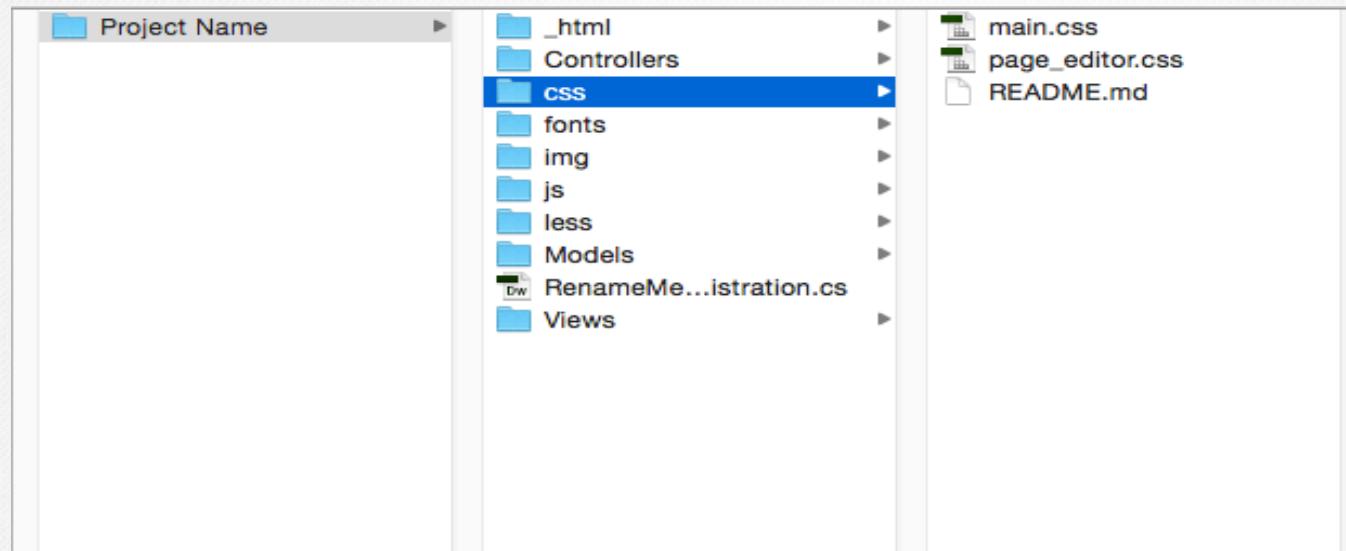
Site Folder

- Modify/include all site-specific styles here
- Create site-specific variables and mixins



Project CSS

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



Main LESS

- Compiles to main.css
- main.less imports all Bootstrap, SCORE, and site LESS files

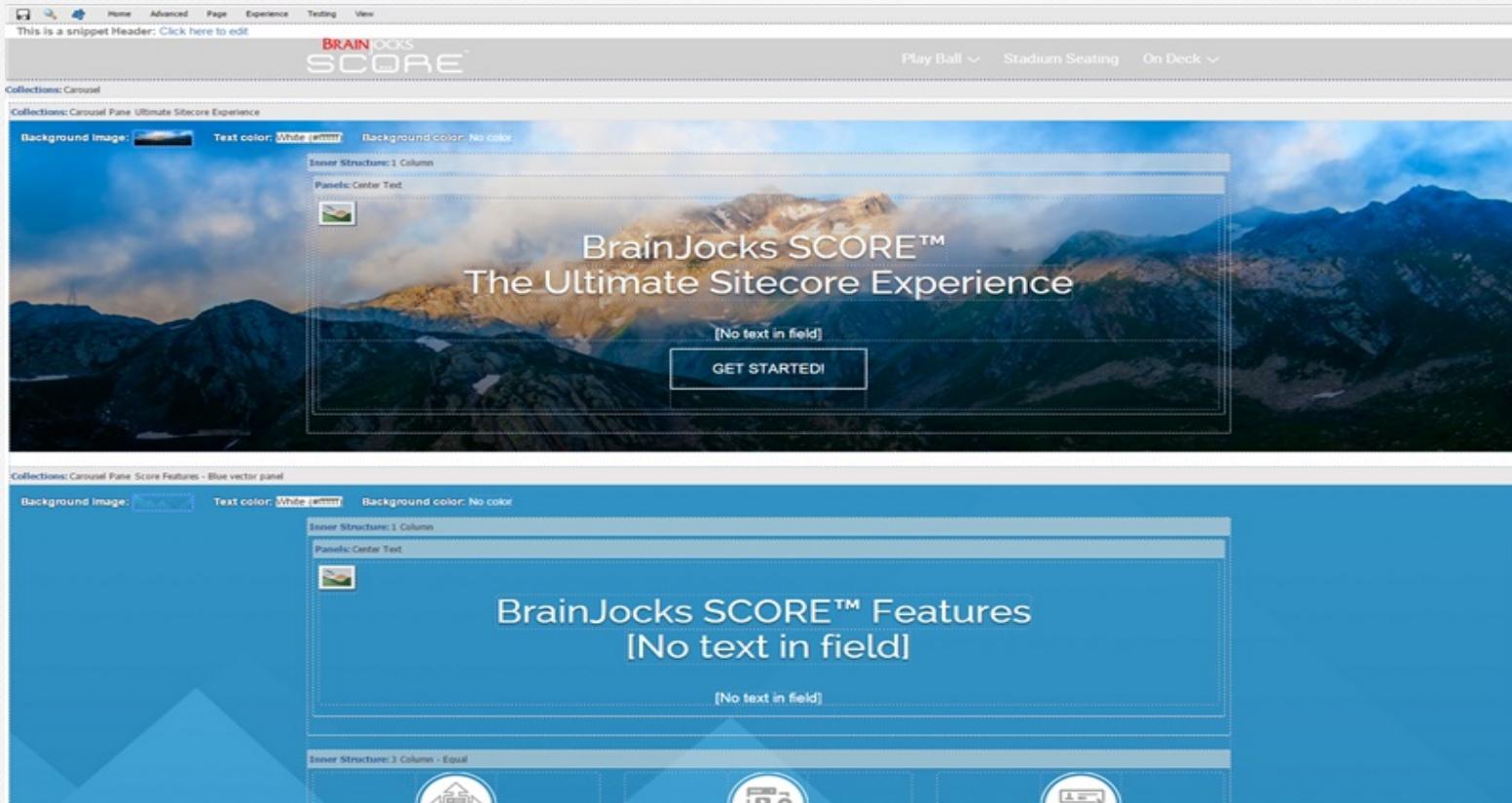
```
1 // Core variables and mixins
2 @import "bootstrap/bootstrap.less";
3
4 // Score styles
5 @import "score/score-nav.less";
6 @import "score/score-page-layout.less";
7 @import "score/score-structural-layout.less";
8 @import "score/score-local-layout.less";
9
10 // tenant styles
11 @import "site/site-variables.less";
12 @import "site/site-background.less";
13 @import "site/site-carousel.less";
14 @import "site/site-heroes.less";
15 @import "site/site-highlights.less";
16 @import "site/site-panels.less";
17 @import "site/site-navigation.less";
18
```

Page Editor LESS

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

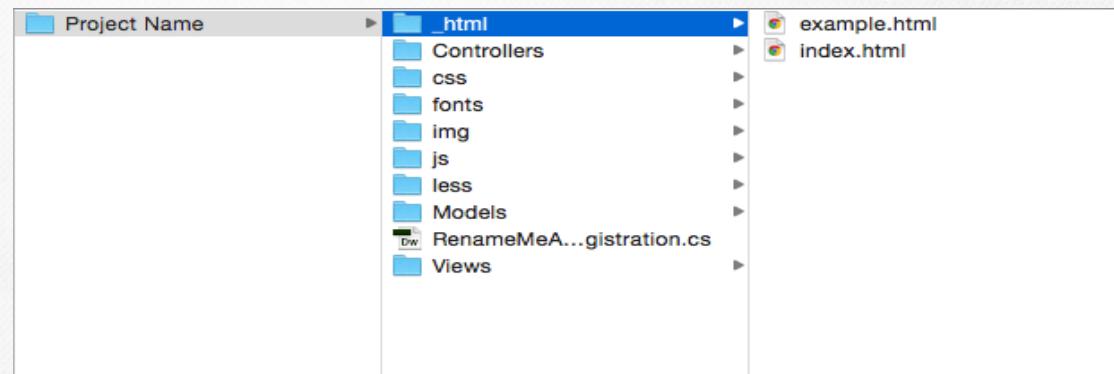
Page Editor CSS

- Example of carousel panes in page editor



HTML Sandbox

- `_html` is your “sandbox” (for POC pages)
- Includes a “kitchen sink” `example.html` file
 - Shows header, `.pageWrapper`, and footer placement
 - Includes example markup of SCORE page, structural, and stripe components
 - Contains all default markup and styles for most common SCORE components



Review `example.html`

Vertical Spacing Strategies

- Bootstrap does not come with vertical spacing for all components
- SCORE makes use of a global variable: @vertical-spacing
 - All SCORE content components have class .score-component which has margin-bottom: @vertical-spacing
 - Alternative: specifying margin for every component individually
 - Global variable is easy to maintain
- Best practice: always use margin-bottom to accomplish spacing rather than margin-top



03

Styling SCORE Components

Understanding SCORE's page and structural layouts, using responsive grid mixins, overriding Bootstrap styles, and customizing site components

SCORE Building Blocks

- All SCORE pages are built with:
 - Header
 - pageWrapper
 - Footer



SCORE Building Blocks



*Bootstrap example

```
<div class="container">
  <div class="row">
    <div class="col-md-12">
      //content
    </div>
  </div>
</div>
```

*SCORE example

```
<div class="score-container">
  <div class="container">
    <div class="score-page score-column1" (.row)
      <div class="score-main" (.col-md-12)
        <div class="score-inner">
          //content
        </div>
      </div>
    </div>
  </div>
</div>
```

SCORE Building Blocks

*Bootstrap example with nested row

```
<div class="container">
  <div class="row">
    <div class="col-md-12">
      //content
      <div class="row">
        <div class="col-md-6">
          //content
        </div>
        <div class="col-md-6">
          //content
        </div>
      </div>
      //content
    </div>
  </div>
</div>
```

*SCORE example with nested row

```
<div class="score-container">
  <div class="container">
    <div class="score-page score-column1" (.row)
      <div class="score-main" (.col-md-12)
        <div class="score-inner">
          //content
        <div class="score-structural score-column2-equal" (.row)
          <div class="score-left" (.col-md-6)
            //content
          </div>
          <div class="score-right" (.col-md-6)
            //content
          </div>
        </div>
        //content
      </div>
    </div>
  </div>
</div>
```

SCORE Building Blocks

*SCORE example -IMPROPER-

```
<div class="score-container">
  <div class="container">
    <div class="score-page score-column2-largeleft" (.row)
      <div class="score-main" (.col-md-8)
        <div class="score-inner">
          //content
        </div>
      </div>
      <div class="score-side" (.col-md-4)
        <div class="score-inner">
          //content
        </div>
      </div>
    </div>
  </div>
<div class="score-container">
  <div class="container">
    <div class="score-page score-column2-equal" (.row)
      <div class="score-side" (.col-md-6)
        <div class="score-inner">
          //content
        </div>
      </div>
      <div class="score-side2" (.col-md-6)
        <div class="score-inner">
          //content
        </div>
      </div>
    </div>
  </div>
</div>
```

*SCORE example -PROPER-

```
<div class="score-container">
  <div class="container">
    <div class="score-page score-column1" (.row)
      <div class="score-main" (.col-md-12)
        <div class="score-inner">
          //content
        </div>
      </div>
    </div>
  </div>
<div class="score-structural score-column2-wideleft">
  <div class="score-left" (.col-md-8)
    //content
  </div>
  <div class="score-right" (.col-md-4)
    //content
  </div>
</div>
<div class="score-structural score-column2-equal">
  <div class="score-left" (.col-md-6)
    //content
  </div>
  <div class="score-right" (.col-md-6)
    //content
  </div>
</div>
</div>
```

SCORE Building Blocks

*SCORE example with score-stripe

```
<div class="score-container">
  <div class="container">
    <div class="score-page score-column1" (.row)
      <div class="score-main" (.col-md-12)
        <div class="score-inner">
          //content
        </div>
      </div>
    </div>
  </div>
<div class="score-stripe">
  <div class="score-container">
    <div class="container">
      <div class="score-page score-column1" (.row)
        <div class="score-main" (.col-md-12)
          <div class="score-inner">
            //content
          </div>
        </div>
      </div>
    </div>
  </div>
<div class="score-container">
  <div class="container">
    <div class="score-page score-column1" (.row)
      <div class="score-main" (.col-md-12)
        <div class="score-inner">
          //content
        </div>
      </div>
    </div>
  </div>
</div>
```



Responsive Grid Mixins

- Grid mixins make or modify rows and columns at the CSS level
- These mixins can be used to create custom page and structural 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
 - Turn pieces of HTML from custom components into rows and columns

```
1  .score-structural {  
2    & .score-column {  
3      &gt; .score-center {  
4        .make-md-column(12);  
5      }  
6    }  
7  }  
8  & .score-column2-equal {  
9    &gt; .score-left {  
10       .make-md-column(6);  
11     }  
12    &gt; .score-right {  
13       .make-md-column(6);  
14     }  
15   }  
16 }  
17 }
```

SCORE Building Blocks

- Bootstrap makes use of containers, rows, and columns
- Containers CANNOT be nested inside other containers
- All rows should be inside of containers
- All columns MUST be inside of rows
- SCORE makes use of containers, score-pages, score-structurals, score-columns and score-stripes
- All score-pages INCLUDE a wrapping container
- A score-page CANNOT be nested inside another score-page
- Score-structurals are ONLY to be nested inside of score-pages (header and footer are exceptions)
- Score-stripes are used to separate score-page(s) for full-width backgrounds

Exercise 1 – Grid Demo

Class Exercise

- Create basic SCORE grid example
 - Download LESS compiler
 - Identify page layout
 - Identify structural layout
 - Identify stripes
 - Create page/structural markup
 - `<div class="score-component"></div>` will represent your SCORE component placeholder

Exercise 2 – SCORE Grid

Where to start?

- Two approaches to begin project development
 - Front-end first, back-end last (you write HTML)
 - Back-end first, front-end last (you are given HTML)
- Determining factors:
 - Budget
 - Time constraints
 - Resources
 - Team preference
- Decide whether to build whole pages with components or build stand-alone components

Pros and Cons: Front-End First

- Pros
 - Solid POC from the beginning (show progress to clients, designers, team members)
 - Point out design flaws early in the process
 - Sitecore devs have a good foundation to start their development
 - Design assets are implemented from the start (image cropping, scaling, saving to project folders)
- Cons
 - Potential re-work after page is built in Sitecore
 - Potential wasted file space with unused styles
 - More pressure on front-end for usable markup

Pros and Cons: Back-End First



- Pros
 - Back-end devs have total control over markup
 - Saves time and effort from front-end perspective
 - Limited front-end involvement at the beginning of the project
- Cons
 - More pressure on back-end for usable markup
 - Sitecore devs have limited visibility with design
 - Potential for HTML re-work after it reaches front-end stage

SCORE Component Docs

- [Parent page for Bootstrap UI components](#)
 - Containers
 - Content
 - Navigation
- Includes example screenshot and HTML output



Styling Components

- Styles are housed in the site LESS folder
- Typically one LESS file per component
- Site styles often need to override native Bootstrap styles
 - Should be declared on site-specific component less file (ex. site-navigation.less, site-carousel.less, etc.)
 - Should NOT be declared in Bootstrap or SCORE less folder

Class Selections

- Components that allow class selections include page headers, sub-headers, stripes, carousels, and highlights
- Enables customization of SCORE components and eliminates the need to create custom components
- Stripes, heroes, and carousels have built-in choices for class selection that control the background size
 - Allows for covered, contained, or repeated background image
 - Reference:
[Proper implementation of background images with text](#)

Demo – Component Examples



04

Best Practices

Ensuring the best project experience for
yourself and your team

Working with Sitecore

- Always have a POC as an HTML file in initial output
 - Preferably shown at page level based upon designed composition
- Stay in sync with Sitecore developers to avoid confusion and re-work
- All project files are shared between Sitecore and UI developers

Working with Sitecore

- Avoid creating new components if possible
 - Utilize option of adding classes to SCORE components
 - If a new component must be introduced, talk to Sitecore developer
 - Any new component styles should be saved in "site" folder (site-component-name.less)
- Keep HTML sandbox files up to date with production
- Use browser extensions to test local CSS on production pages
 - CSS Inject, CSS Reloader, Web Developer, etc.

Code Organization

- Keep LESS files organized and tidy
 - Prevents code bloat and allows code to be easily understood or worked on by another team member
- Stay DRY!!
 - Do your best not to repeat specific styles or blocks of code
 - Use global variables and mixins that can easily be updated and maintained
- Keep nesting under control
 - Maximum of 4 levels deep if possible
 - Styles under highly specific selectors are hard to override in other places
- We recommend NOT changing any Bootstrap LESS files



Final Exercise

- Create home page proof of concept using SCORE components
 - Identify page layouts
 - Identify structural layouts
 - Identify stripes
 - Identify SCORE components
 - Identify custom components
 - Create markup and styles



Exercise 3 – POC

Ben Cato
ben@brainjocks.com
770.924.6444

Emily Lord
emily@brainjocks.com
770.924.6444

