BRAINJOCKS SCORE™

September 2015

Styling BrainJocks SCORE

Ben Cato & Emily Lord

BRAINJOCKS.COM

Course Overview

01 The value of SCORE

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

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

04 Where to start?
Approaches to project development

05 Understanding SCORE's grid system – page layout, inner structure, and stripes; using responsive grid mixins

06 Styling SCORE components – overriding Bootstrap and using class selections

07 SCORE front-end best practices

01 The Value of SCORE

SCORE Benefits

1. Reusable components
2. Portability
3. Saves time
4. Saves money!

SCORE Markup

- SCORE components are categorized as container, content, or navigation
  - 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

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

02

Overview of Pre-requisites

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

02.1

Bootstrap 3 Overview

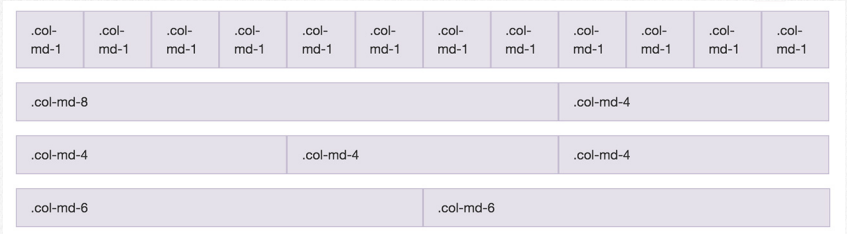
What is Bootstrap?

- 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 class="col-md-1"></div>
        <div class="col-md-1"></div>
        <div class="col-md-1"></div>
        <div class="col-md-1"></div>
        <div class="col-md-1"></div>
        <div class="col-md-1"></div>
        <div class="col-md-1"></div>
        <div class="col-md-1"></div>
        <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 horizontally centered)

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

The Anatomy of Bootstrap's Grid

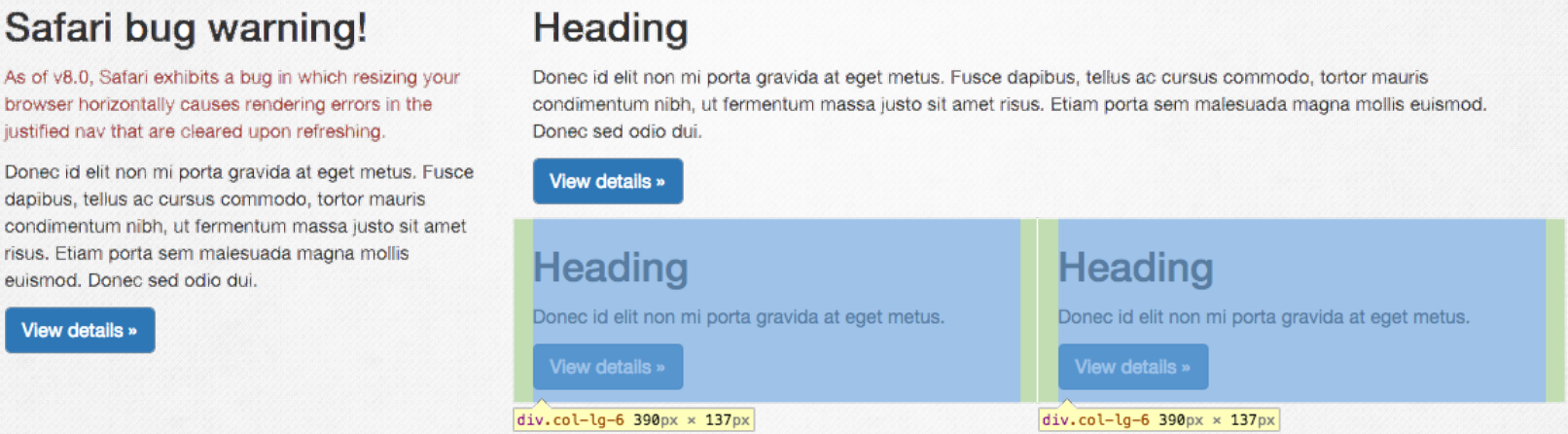
- Column:
  - Width is specified by developer using a class or mixin
    - NOTE: Bootstrap uses 12-column grid
  - 15px padding on left and right (30px between columns)
  - ALWAYS inserted into a row

The Anatomy of Bootstrap's Grid

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

The Anatomy of Bootstrap's Grid

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



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

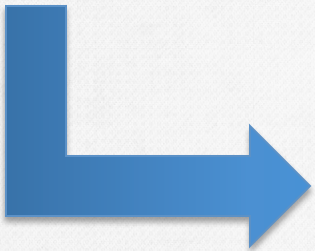
Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

View details »

Heading2

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 »

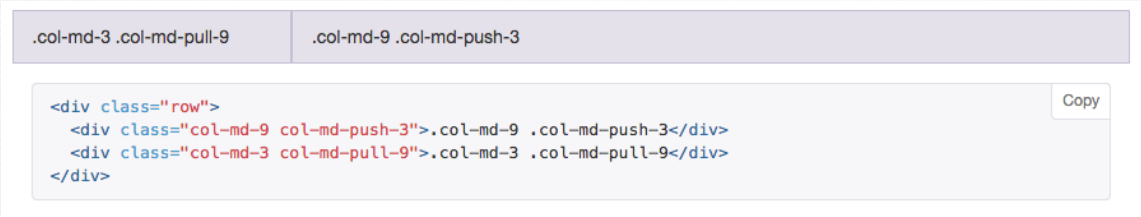
Heading2

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-*



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 (≥768px)	Medium devices Desktops (≥992px)	Large devices Desktops (≥1200px)
`.visible-xs-*`	Visible	Hidden	Hidden	Hidden
`.visible-sm-*`	Hidden	Visible	Hidden	Hidden
`.visible-md-*`	Hidden	Hidden	Visible	Hidden
`.visible-lg-*`	Hidden	Hidden	Hidden	Visible
`.hidden-xs`	Hidden	Visible	Visible	Visible
`.hidden-sm`	Visible	Hidden	Visible	Visible
`.hidden-md`	Visible	Visible	Hidden	Visible
`.hidden-lg`	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

SCORE Component Docs

- Parent page for Bootstrap UI components
  - Containers
  - Content
  - Navigation
- Includes example screenshot and HTML output

02.2

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 nested styles, which is amazing – but can easily get out of control
  - Try to keep nesting no more than 4 levels deep
- Reference: LESS Tutorial

LESS Compiling

Compiler

CSS

```less
//LESS
ul {
    list-style: none;
    li{
        float: left;
        a {
            color: #f00;
        }
    }
}
```

Compiler

```css
//CSS:
ul {
    list-style: none;
}
ul li {
    float: left;
}
ul li a {
    color: #foo;
}
```

LESS Compilers

COMMAND LINE

MAC

WINDOWS

REFERENCE: List of popular compilers

Exercise 1.1: Installing a Compiler

1. Navigate to http://winless.org/
2. Download compiler
3. Open the 1_less_exercise folder from class files
4. Manually compile main.less

Exercise 1 – LESS Exercise

Good Nesting Example

less.less

```less
//LESS
ul {
    list-style: none;
    li{
        float: left;
        a {
            color: #f00;
        }
    }
}
```

css.css

```css
//CSS:
ul {
    list-style: none;
}
ul li {
    float: left;
}
ul li a {
    color: #foo;
}
```

Bad Nesting Example

less.less

```less
//LESS
.container {
    .row-class {
        .column-class {
            ul {
                list-style: none;
                li {
                    float: left;
                    a {
                        color: #foo;
                    }
                }
            }
        }
    }
}
```

css.css

```css
//CSS:
.container .row-class .column-class ul li a {
    color: #foo;
}
```

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

less.less

```less
@brand-color: #foo;

header, footer {
    background-color: @brand-color;
}

h1 {
    color: @brand-color;
}
```

Exercise 1.2: Using a Variable

1. Navigate to less > site > variables.less
2. Define a new variable, @button-color
3. Navigate to site-buttons.less in the same folder
4. Change the background color of all SCORE button components (class of .btn-default) to @button-color
5. After successful compilation, open _html > index.html in a browser
6. Change @button-color to another color value
7. Compile main.less and view index.html in the browser

Exercise 1 – LESS Exercise

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

less.less

```less
.rounded-corners(@radius) {
    -webkit-border-radius: @radius;
    -moz-border-radius: @radius;
    -o-border-radius: @radius;
    border-radius: @radius;
}

.button {
    .rounded-corners(5px);
}
```

Exercise 1.3: Using a Mixin

1. Navigate to variables.less
2. Define a new mixin .fun-transition, which will rotate something 45 degrees
3. Navigate to site-buttons.less
4. Insert .fun-transition into the button's hover state
5. Compile main.less and view index.html in the browser. Hover over a button.
6. Modify .fun-transition to accept a parameter for the angle of rotation, use variable @angle
7. Insert any angle value into the call to .fun-transition in site-buttons.less
8. Compile main.less and hover over a button on index.html

Exercise 1 – LESS Exercise

LESS Operators

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

less.less

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

css.css

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

less.less

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

css.css

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

LESS Operators

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

less.less

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

css.css

```css
//CSS
.class1 > .class2 {
  color: #foo;
}
```

Nesting Media Queries

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

less.less

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

css.css

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

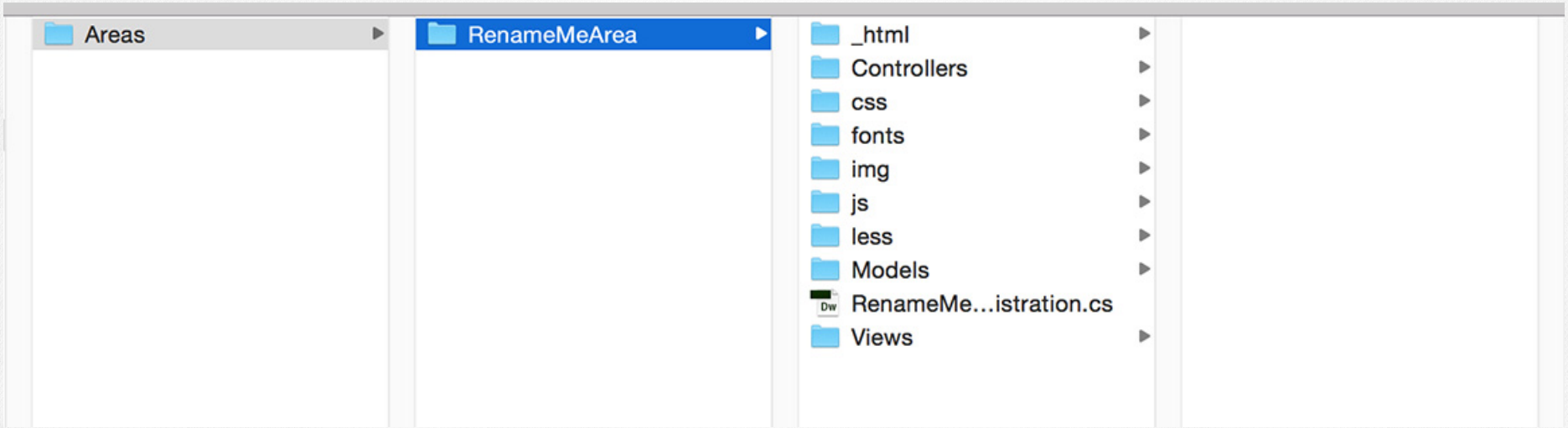
03

Understanding SCORE's File Structure

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

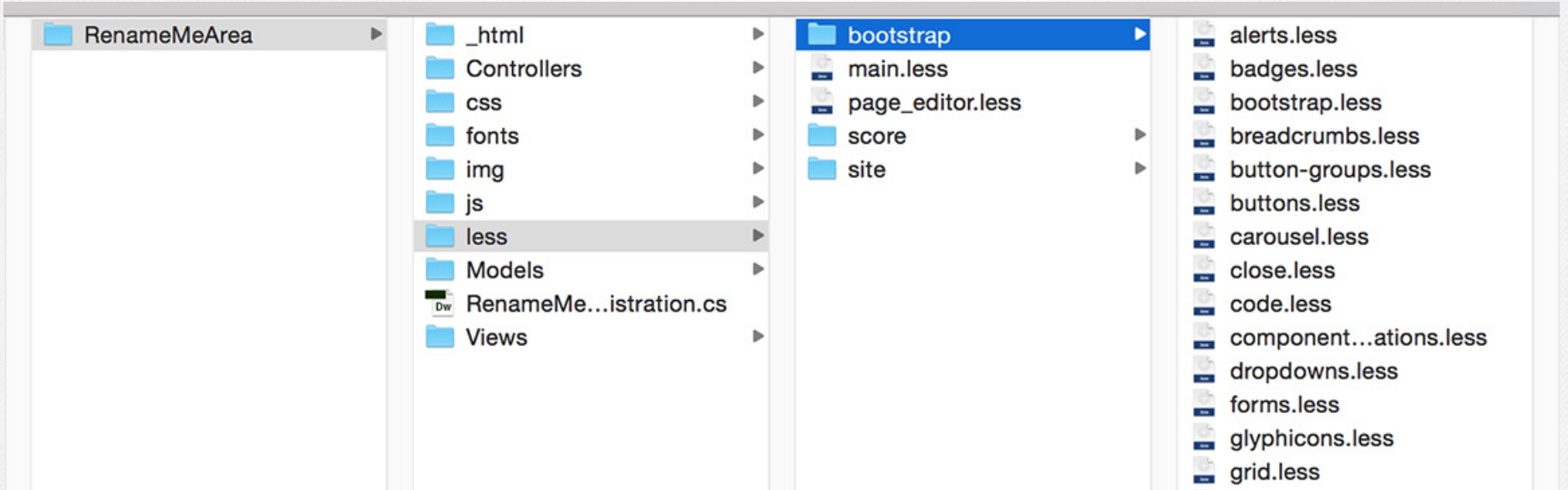
“Areas” Scaffolding

- ALL project files are SHARED between Sitecore and UI developers



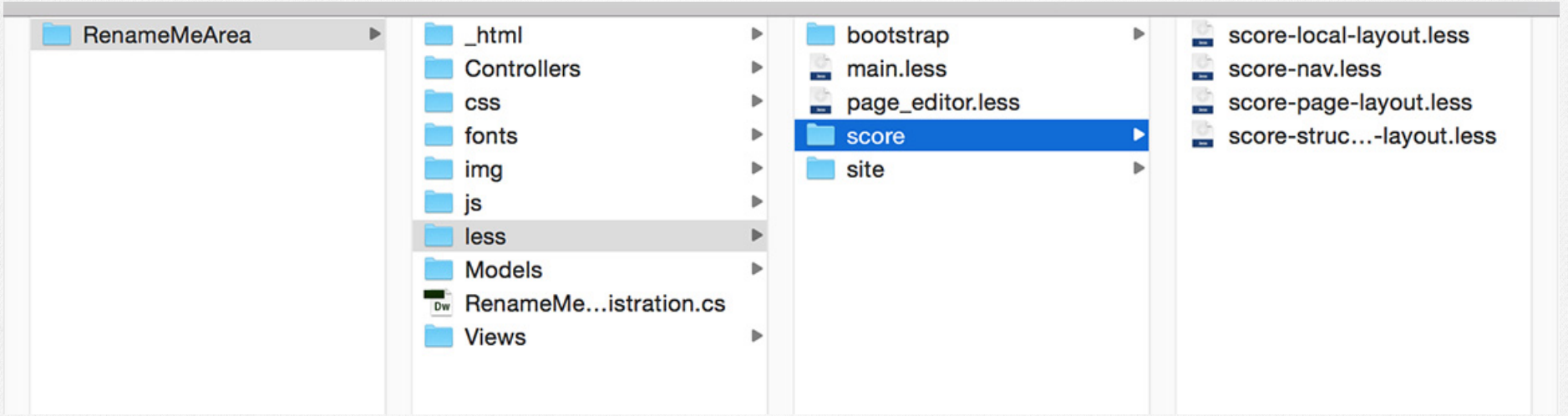
Bootstrap Folder

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



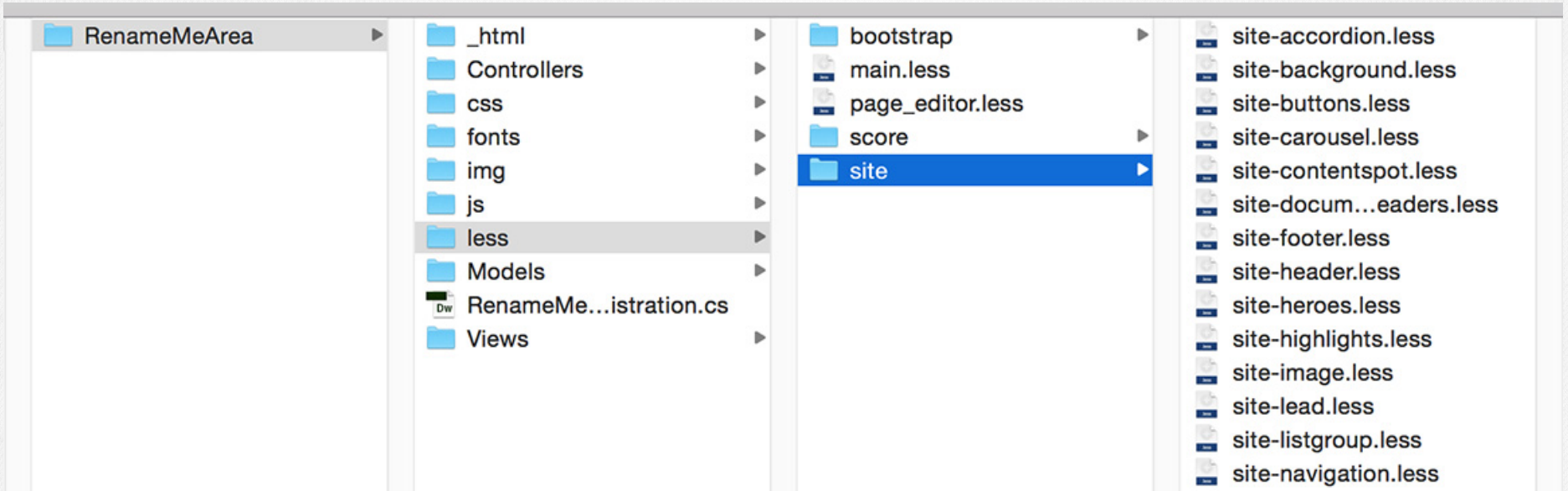
SCORE Folder

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



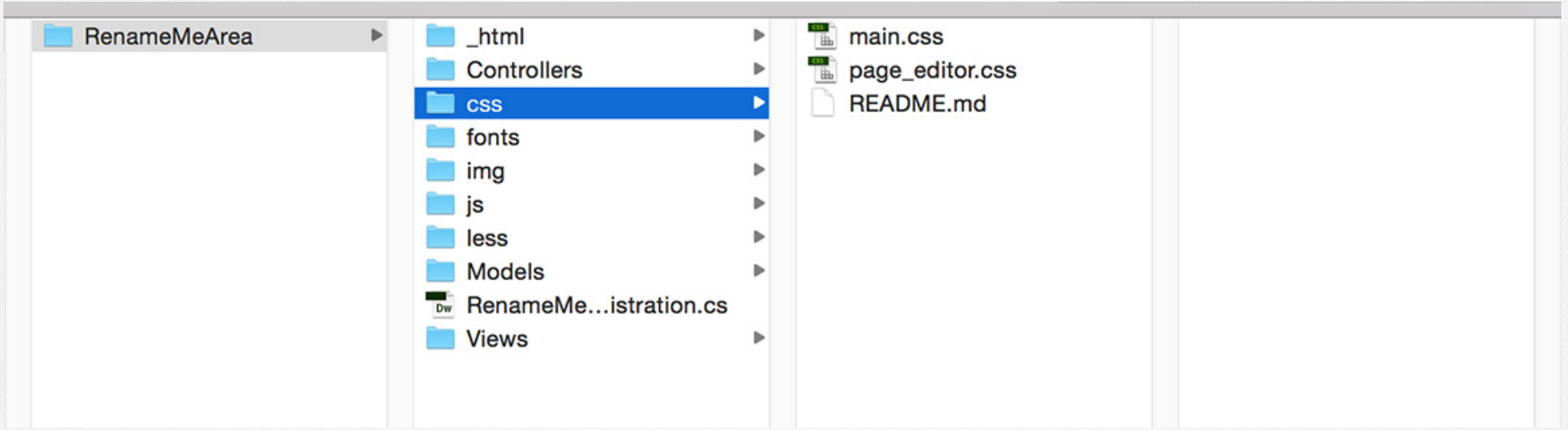
Site Folder

- Location to modify/include all site-specific styles
- 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

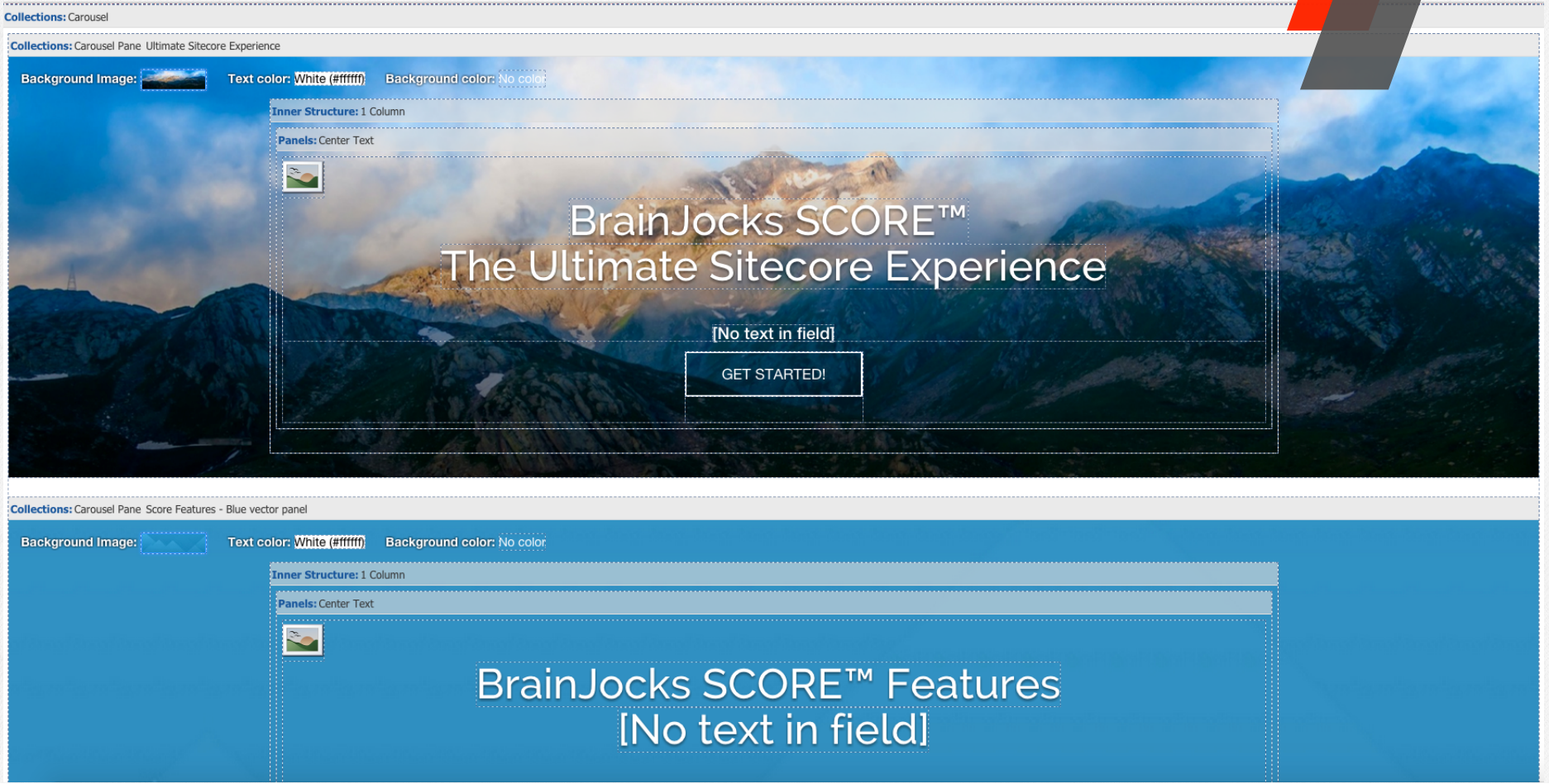
main.less

```
// Core variables and mixins
@import "bootstrap/bootstrap.less";

// Score styles
@import "score/score-nav.less";
@import "score/score-page-layout.less";
@import "score/score-structural-layout.less";
@import "score/score-local-layout.less";

// tenant styles
@import "site/site-variables.less";
@import "site/site-accordion.less";
@import "site/site-background.less";
@import "site/site-buttons.less";
@import "site/site-carousel.less";
@import "site/site-contentspot.less";
@import "site/site-documentheaders.less";
@import "site/site-footer.less";
@import "site/site-header.less";
```

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

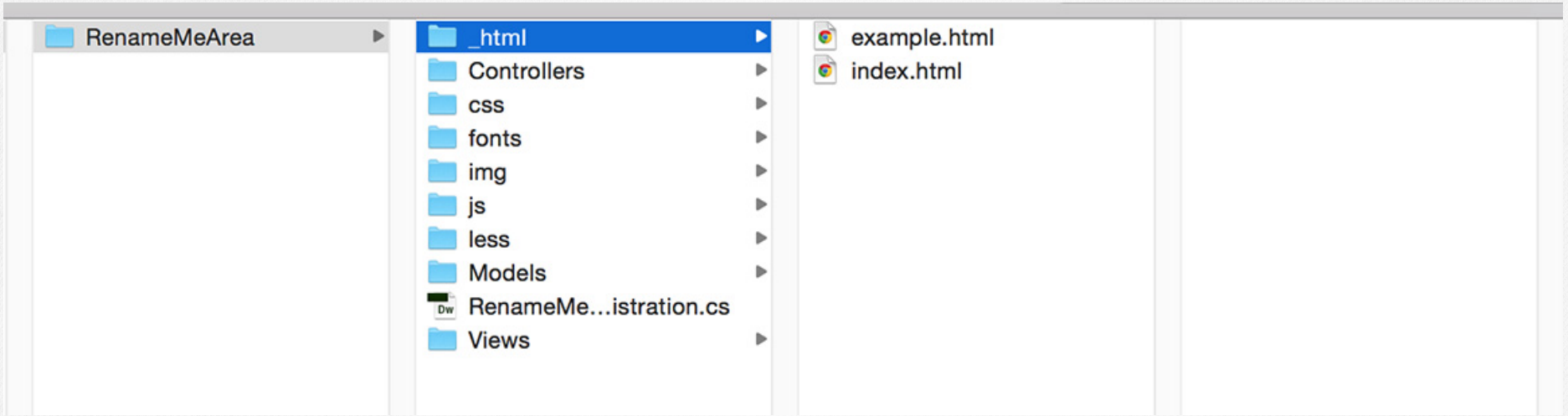
Page Editor CSS



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

HTML Sandbox



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

04

Where to Start?

Approaches to project development

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

05

Understanding SCORE's Grid

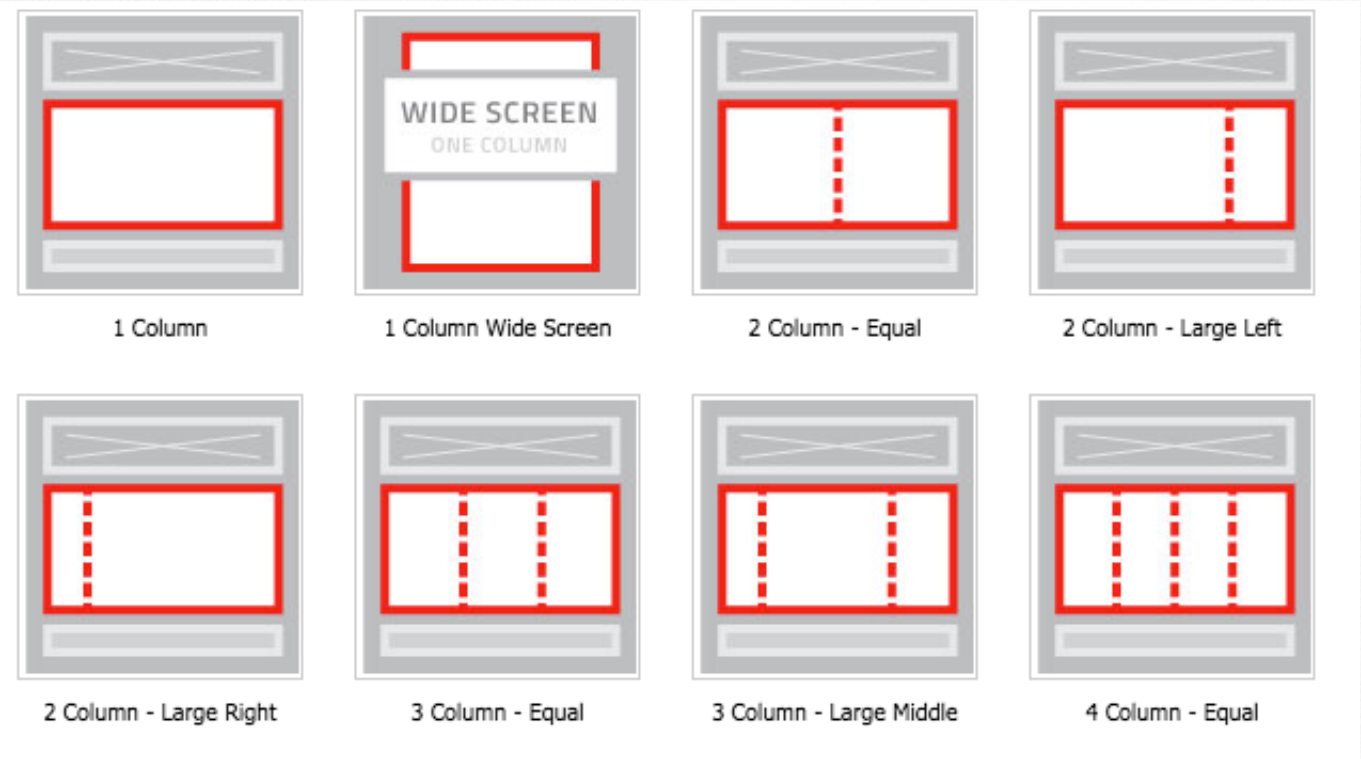
Page layout, inner structure, and stripes; using responsive grid mixins

Container Components

- Page layout = .container + row
- Inner structure = row
- Stripes

Exercise 2 – Grid Examples

SCORE Building Blocks – Page Layout



SCORE Building Blocks – Page Layout

bootstrap.html

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

score.html

```
<div class="score-container">
    <div class="container">
        <div class="score-page score-column1">
            <div class="score-main">
                <div class="score-inner">
                    ...
                </div>
            </div>
        </div>
    </div>
</div>
```

SCORE Building Blocks – Page Layout

bootstrap.html

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

score.html

```
<div class="score-container">
    <div class="container">
        <div class="score-page score-column2-equal">
            <div class="score-side">
                <div class="score-inner">
                    ...
                </div>
            </div>
            <div class="score-side2">
                <div class="score-inner">
                    ...
                </div>
            </div>
        </div>
    </div>
</div>
```

SCORE Building Blocks – Page Layout

bootstrap.html

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

score.html

```
<div class="score-container">
    <div class="container">
        <div class="score-page score-column2-largeleft">
            <div class="score-main">
                <div class="score-inner">
                    ...
                </div>
            </div>
            <div class="score-side">
                <div class="score-inner">
                    ...
                </div>
            </div>
        </div>
    </div>
</div>
```

SCORE Building Blocks – Page Layout

bootstrap.html

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

score.html

```
<div class="score-container">
    <div class="container">
        <div class="score-page score-column2-largeright">
            <div class="score-side">
                <div class="score-inner">
                    ...
                </div>
            </div>
            <div class="score-main">
                <div class="score-inner">
                    ...
                </div>
            </div>
        </div>
    </div>
</div>
```

SCORE Building Blocks – Page Layout

bootstrap.html

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

score.html

```
<div class="score-container">
    <div class="container">
        <div class="score-page score-column3-equal">
            <div class="score-side">
                <div class="score-inner">
                    ...
                </div>
            </div>
            <div class="score-main">
                <div class="score-inner">
                    ...
                </div>
            </div>
            <div class="score-side2">
                <div class="score-inner">
                    ...
                </div>
            </div>
        </div>
    </div>
</div>
```

SCORE Building Blocks – Page Layout

bootstrap.html

```
<div class="container">
    <div class="row">
        <div class="col-md-3">
            ...
        </div>
        <div class="col-md-6">
            ...
        </div>
        <div class="col-md-3">
            ...
        </div>
    </div>
</div>
```

score.html

```
<div class="score-container">
    <div class="container">
        <div class="score-page score-column3-largemiddle">
            <div class="score-side">
                <div class="score-inner">
                    ...
                </div>
            </div>
            <div class="score-main">
                <div class="score-inner">
                    ...
                </div>
            </div>
            <div class="score-side2">
                <div class="score-inner">
                    ...
                </div>
            </div>
        </div>
    </div>
</div>
```

SCORE Building Blocks – Page Layout

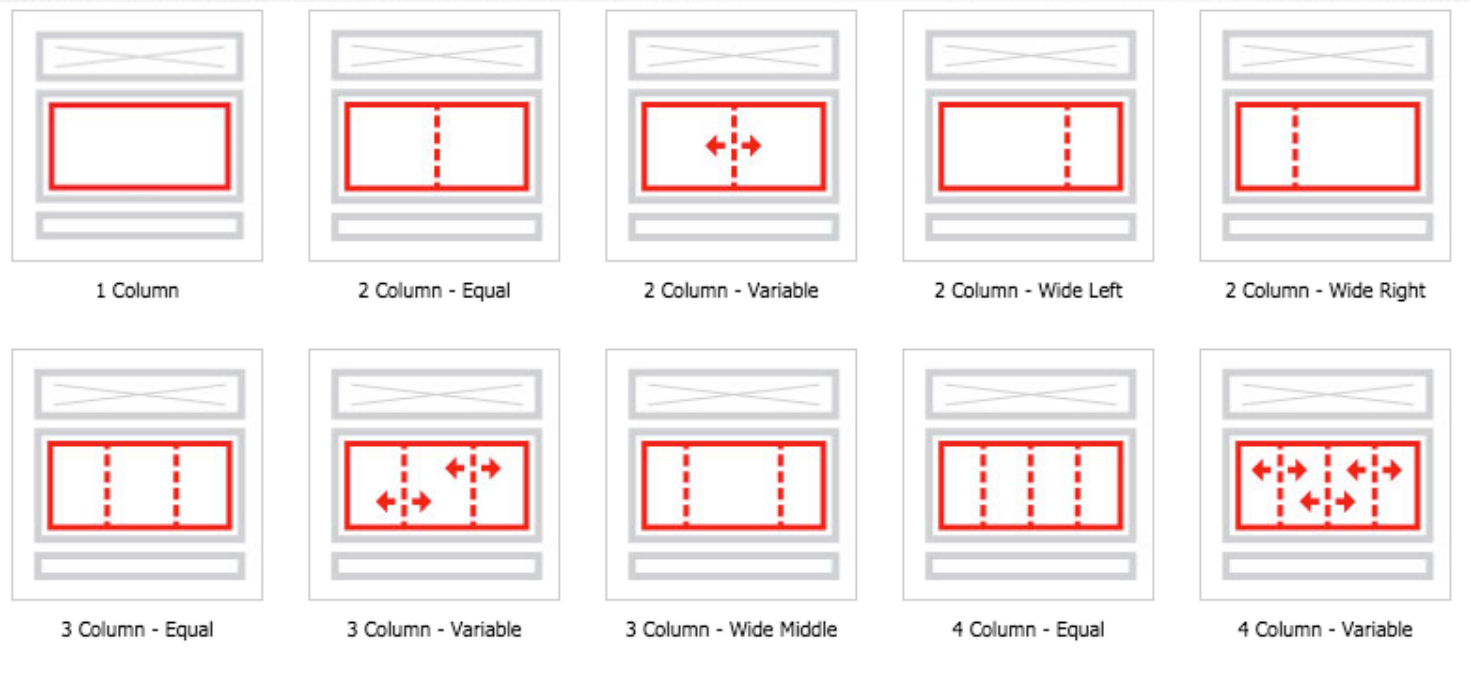
bootstrap.html

```
<div class="container">
    <div class="row">
        <div class="col-md-3">
            ...
        </div>
        <div class="col-md-3">
            ...
        </div>
        <div class="col-md-3">
            ...
        </div>
        <div class="col-md-3">
            ...
        </div>
    </div>
</div>
```

score.html

```
<div class="score-container">
    <div class="container">
        <div class="score-page score-column4-equal">
            <div class="score-side">
                <div class="score-inner">
                    ...
                </div>
            </div>
            <div class="score-main">
                <div class="score-inner">
                    ...
                </div>
            </div>
            <div class="score-main2">
                <div class="score-inner">
                    ...
                </div>
            </div>
            <div class="score-side2">
                <div class="score-inner">
                    ...
                </div>
            </div>
        </div>
    </div>
</div>
```

SCORE Building Blocks – Inner Structure



SCORE Building Blocks – Inner Structure

bootstrap.html

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

score.html

```
<div class="score-container">
    <div class="container">
        <div class="score-page score-column1">
            <div class="score-main">
                <div class="score-inner">
                    ...
                    <div class="score-structural
                                score-column2-equal">
                        <div class="score-left">
                            ...
                        </div>
                        <div class="score-right">
                            ...
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

SCORE Building Blocks – Inner Structure

bootstrap.html

```
<div class="container">
    <div class="row">
        <div class="col-md-12">
            ...
            <div class="row">
                <div class="col-md-8">
                    ...
                </div>
                <div class="col-md-4">
                    ...
                </div>
            </div>
        </div>
    </div>
</div>
```

score.html

```
<div class="score-container">
    <div class="container">
        <div class="score-page score-column1">
            <div class="score-main">
                <div class="score-inner">
                    ...
                    <div class="score-structural
                                score-column2-wideleft">

                        <div class="score-left">
                            ...
                        </div>
                        <div class="score-right">
                            ...
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

SCORE Building Blocks – Inner Structure

bootstrap.html

```
<div class="container">
    <div class="row">
        <div class="col-md-12">
            ...
            <div class="row">
                <div class="col-md-4">
                    ...
                </div>
                <div class="col-md-8">
                    ...
                </div>
            </div>
        </div>
    </div>
</div>
```

score.html

```
<div class="score-container">
    <div class="container">
        <div class="score-page score-column1">
            <div class="score-main">
                <div class="score-inner">
                    ...
                    <div class="score-structural
                                score-column2-wideright">

                        <div class="score-left">
                            ...
                        </div>
                        <div class="score-right">
                            ...
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

SCORE Building Blocks – Inner Structure

bootstrap.html

```
<div class="container">
    <div class="row">
        <div class="col-md-12">
            ...
            <div class="row">
                <div class="col-md-4">
                    ...
                </div>
                <div class="col-md-4">
                    ...
                </div>
                <div class="col-md-4">
                    ...
                </div>
            </div>
        </div>
    </div>
</div>
```

score.html

```
<div class="score-container">
    <div class="container">
        <div class="score-page score-column1">
            <div class="score-main">
                <div class="score-inner">
                    ...
                    <div class="score-structural
                                score-column3-equal">

                        <div class="score-left">
                            ...
                        </div>
                        <div class="score-center">
                            ...
                        </div>
                        <div class="score-right">
                            ...
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

SCORE Building Blocks – Inner Structure

bootstrap.html

```
<div class="container">
    <div class="row">
        <div class="col-md-12">
            ...
            <div class="row">
                <div class="col-md-2">
                    ...
                </div>
                <div class="col-md-8">
                    ...
                </div>
                <div class="col-md-2">
                    ...
                </div>
            </div>
        </div>
    </div>
</div>
```

score.html

```
<div class="score-container">
    <div class="container">
        <div class="score-page score-column1">
            <div class="score-main">
                <div class="score-inner">
                    ...
                    <div class="score-structural
                                score-column3-widemiddle">

                        <div class="score-left">
                            ...
                        </div>
                        <div class="score-center">
                            ...
                        </div>
                        <div class="score-right">
                            ...
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

SCORE Building Blocks – Inner Structure

bootstrap.html

```
<div class="container">
    <div class="row">
        <div class="col-md-12">
            ...
            <div class="row">
                <div class="col-md-3">
                    ...
                </div>
                <div class="col-md-3">
                    ...
                </div>
                <div class="col-md-3">
                    ...
                </div>
                <div class="col-md-3">
                    ...
                </div>
            </div>
        </div>
    </div>
</div>
```

score.html

```
<div class="score-container">
    <div class="container">
        <div class="score-page score-column1">
            <div class="score-main">
                <div class="score-inner">
                    ...
                    <div class="score-structural
                                score-column4-equal">

                        <div class="score-left">
                            ...
                        </div>
                        <div class="score-center">
                            ...
                        </div>
                        <div class="score-center2">
                            ...
                        </div>
                        <div class="score-right">
                            ...
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

SCORE Building Blocks – Stripe

Stripe

SCORE Building Blocks – Stripe

bootstrap.html

```
<div class="container">
    <div class="row">
        <div class="col-md-12">
            ...
        </div>
    </div>
</div>
<div class="full-width-wrapper">
    <div class="container">
        <div class="row">
            <div class="col-md-12">
                ...
            </div>
        </div>
    </div>
</div>
<div class="container">
    <div class="row">
        <div class="col-md-12">
            ...
        </div>
    </div>
</div>
```

score.html

```
<div class="score-container">
    <div class="container">
        <div class="score-page score-column1">
            <div class="score-main">
                <div class="score-inner">
                    ...
                </div>
            </div>
        </div>
    </div>
</div>
<div class="score-stripe">
    <div class="score-container">
        <div class="container">
            <div class="score-page score-column1">
                <div class="score-main">
                    <div class="score-inner">
                        ...
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
<div class="score-container">
    <div class="container">
        <div class="score-page score-column1">
            <div class="score-main">
                <div class="score-inner">
                    ...
                </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

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

Class Exercise

- Create basic SCORE grid example
  1. Navigate to 3_score_grid_exercise > grid-exercise.pdf
  2. Identify page layout
  3. Identify structural layout
  4. Identify stripes
  5. Create page/structural markup
  6. will represent your SCORE component placeholder

Exercise 3 – SCORE Grid

SCORE Component Docs

- Parent page for Bootstrap UI components
  - Containers
  - Content
  - Navigation
- Includes example screenshot and HTML output
- Older versions are child pages of current version

06

Styling SCORE Components

Overriding Bootstrap and using class selections

Styling Content 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

Show/Hide Components

- Content administrators can determine to show or hide components for different screen sizes
- Based on Bootstrap's responsive show/hide classes
- Managed in content editor

	Extra small devices Phones (<768px)	Small devices Tablets (≥768px)	Medium devices Desktops (≥992px)	Large devices Desktops (≥1200px)
`.visible-xs-*`	Visible	Hidden	Hidden	Hidden
`.visible-sm-*`	Hidden	Visible	Hidden	Hidden
`.visible-md-*`	Hidden	Hidden	Visible	Hidden
`.visible-lg-*`	Hidden	Hidden	Hidden	Visible
`.hidden-xs`	Hidden	Visible	Visible	Visible
`.hidden-sm`	Visible	Hidden	Visible	Visible
`.hidden-md`	Visible	Visible	Hidden	Visible
`.hidden-lg`	Visible	Visible	Visible	Hidden

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

07

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.
- Make use of preprocessor source mapping in browser developer tools
  - Chrome Devtools Reference

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

1. Navigate to 4_poc_exercise > design-studios > design-studios.psd or .pdf
2. Identify page layouts
3. Identify structural layouts
4. Identify stripes
5. Identify SCORE components
6. Identify custom components
7. Create markup and styles

Exercise 4 – POC

Ben Cato
ben@brainjocks.com
770.924.6444

Emily Lord
emily@brainjocks.com
770.924.6444

BRAINJOCKS SCORE™