



# CS50B Web Dev 2

Fall 2018 ~ Ethan Wilde

*Week 15*



# Welcome

- **Course Outline**
- **No Textbook Reading**
- **Working with Bootstrap 4 Sass Files**
- **The Understrap theme**
- **Continuing Your Final Project**

# Course Outline

1 Responsive HTML5 + CSS3	10 WordPress: Intro to CMS
2 Introduction to Bootstrap	11 WordPress Posts
3 Page Elements + Components	12 WordPress Pages + Menus
4 Navigation	13 WordPress Plugins + Themes
5 Forms + Decoration	14 WordPress Bootstrap Theme
6 Customizing Bootstrap CSS	15 Intro to Sass: CSS Preprocess
7 Introduction to JavaScript	16 Customizing Bootstrap Sass
8 Bootstrap + JavaScript	17 Mastering Sass + Bootstrap
9 Midterm Review + Midterm	18 Final Exam (online)

Get all of the details in the complete syllabus on Canvas.

No Textbook Reading  
This Week

# CSS Pre-Processors: Working with Sass Rules



<https://sass-lang.com/>

# CSS Pre-Processors:

## Sass

**The syntax for Sass is almost identical to CSS. The authors of Sass were smart and made Sass compatible with CSS, meaning you can write CSS in your Sass files.**

**But do remember to save your Sass files names with the suffix `.scss` and not `.css`.**

# CSS Pre-Processors:

## Sass

**There are so many amazing capabilities that Sass introduces to make writing style rules easier and more efficient.**

**Here is a list of my top three.**

# CSS Pre-Processors:

## Sass

### 1. Variables

```
$color1: #00214D;
```

```
body {  
  color: $color1;  
}
```



# CSS Pre-Processors:

## Sass

### 2. Nested Rules

```
<header class="x">  
  <nav> ... </nav>  
</header>
```

```
header.x {  
  nav { color: red; }  
}
```

# CSS Pre-Processors:

## Sass

### 3. Mix-ins

```
@mixin mixit1() {  
  border:2px solid red;  
  background-color:yellow;  
}
```

```
#element1 {  
  font-size:20px;  
  @include mixit1();  
}
```

# Working with Bootstrap 4 Sass Files

**The style rules in the Bootstrap 4 framework are actually created using Sass.**

**Software on the server reads the Sass SCSS files and creates derivative CSS file(s) from them.**

**The browser reads the CSS.**

# Continuing Your Final Project

## **Week 15**

1. Begin writing Sass code in WordPress-powered site for CSS generation.
2. Create draft project documentation for your project, as presentation.

# Continuing Your Final Project

## **Week 16**

1. Conduct final user testing of your WordPress site and report changes (human-centered design practices from [designkit.org](https://designkit.org)).
2. Get feedback on draft project documentation.
3. Continue writing Sass code in WordPress-powered site for CSS generation.

# Continuing Your Final Project

## **Week 17 + Week 18 (finals week)**

1. Edit Sass rules and content in site to implement changes from final user testing.
2. Edit draft project documentation for your project based on feedback.
3. Deliver final site files, URL and documentation; minimum count of ten (10) pages / posts.
4. Optional: migrate your site to a the Pantheon hosting platform.
5. Participate in Final Project Presentations discussion.

# What to Do Next

- **Reading + Doing**

- Read "WP-SCSS: A Wordpress Sass Plugin" found at <https://connectthink.com/developers-corner/wp-scss-wordpress-sass-plugin/>
- Read "Why Bootstrap is a bad fit for WordPress Themes" at <https://theshaper.com/2014/08/19/why-bootstrap-is-a-bad-fit-for-wordpress-themes/>
- Homework
- **Assignment 13: Sass Basics**
- **Discussion 7: Sass + CSS Preprocessors**

- **Canvas Site**

- All materials available there
- <https://canvas.santarosa.edu/courses/32995>