



CS50B Web Dev 2

Fall 2018 ~ Ethan Wilde

Week 14



Welcome

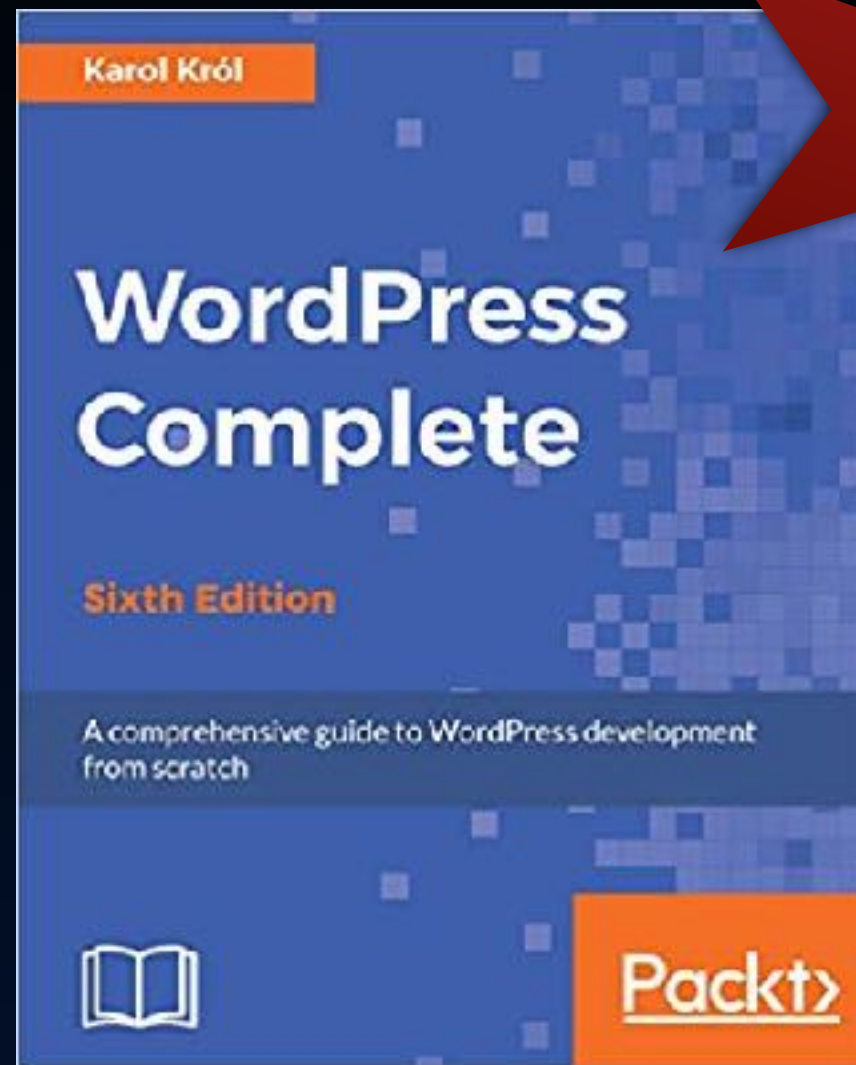
- **Course Outline**
- **Textbook Reading**
- **CSS Pre-Processors: Sass + Less**
- **Working with Bootstrap 4 Sass Files**
- **Creating a New WordPress Workspace**
 - **The WP-SCSS plug-in**
 - **The Understrap theme**
- **Starting Your Final Project**

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.

Textbook Reading This Week



WordPress Complete, Sixth Edition

<https://www.packtpub.com/application-development/wordpress-complete-sixth-edition>

CSS Pre-Processors: Sass + Less



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

CSS Pre-Processors: Sass + Less

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 + Less

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 + Less

1. Variables

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

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


CSS Pre-Processors: Sass + Less

2. Nested Rules

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

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

CSS Pre-Processors: Sass + Less

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.

Creating a New WordPress Workspace

1. **Install the free WP-SCSS plug-in** in your WordPress site in order to enable the compiling of the server-stored Bootstrap Sass files into the final CSS files served to browsers.
2. **Install the free Understrap theme** in your WordPress site in order to utilize the Bootstrap 4 framework and its Sass files.
3. **Make the Understrap theme the active theme** for your WordPress site.
4. **View** your Bootstrap-themed WordPress-powered site.
5. **Edit the Understrap code** so it uses the compiled `/css/theme.css` stylesheet file.

Starting Your Final Project

Week 14

1. Add custom Bootstrap-based Understrap theme to new WordPress site.
2. Begin modifying custom theme to meet requirements of final project proposal.
3. Conduct code validation and accessibility testing of WordPress prototype's content.

Starting 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.

Starting Your Final Project

Week 16

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

Starting 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 *WordPress Complete* (6th ed.)
 - Ch. 8: Developing Your Own Theme
 - Sass Basics Guide
 - Understrap Theme Documentation: Install + Widgets

- **Homework**

- **Assignment 12: Building a Custom Theme**
- **Discussion 7: Sass + CSS Preprocessors**
- **Quiz 2 (WordPress Basics)**

- **Canvas Site**

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