GENERAL ASSEMBLY 6.2

Front-End Web Development

Albert Penticoss
Web Designer, University of Melbourne

This is week 8

- 5 lessons to go!
  - Tonight - Review/Refactoring/Debug CSS and JavaScript
  - Tue 15th - In class Final Project time
  - Wed 16th - Students choice topic
  - Mon 21st - Instructors choice topic
  - Wed 23rd - Final project presentations

Start your engines!

- Connect to wifi
- Open your text editor
- Open GitHub desktop and GitHub website
- Open schoology

Final Project Milestone 3

- 2nd Draft HTML/CSS
- Draft JavaScript
- Due this week - Wednesday 9th March

Submit your assignment

- Assignment 6 - Due this week - Wednesday 9th March
- REMINDER: No “auto” extensions - if you need an extension please do me the courtesy of asking me personally

Reminder – Graduation requirements

- To graduate from the course, students must:
  - 1. complete at least 80% of assigned homework
    - 8 assignments so you can only miss one!
  - 2. attend 80% of classes - so you can miss up to a maximum of four
  - 3. satisfactorily complete the final project (as determined by the course instructor and stated on the class syllabus).

15 MIN CHALLENGE

```
<!doctype html>
<html lang="en">
    <head>

        

    </head>
    <body>

        

        

    </body>
</html>
```

```
<!doctype html>
<html lang="en">
    <head>
        <title>Blank html file</title>

    </head>
    <body>

    </body>
</html>
```

```
<!doctype html>
<html lang="en">
    <head>
        <title>Blank html file</title>
        <meta charset="utf-8">

    </head>
    <body>


    </body>
</html>
```

FEWD – Before we get started


```
<!doctype html>
<html lang="en">
    <head>
        <title>Blank html file</title>
        <meta charset="utf-8">
        <link type="text/css" rel="stylesheet" href="css/main.css">
    </head>
    <body>

    </body>
</html>
```

```
<!doctype html>
<html lang="en">
    <head>
        <title>Blank html file</title>
        <meta charset="utf-8">
        <link type="text/css" rel="stylesheet" href="css/main.css">
    </head>
    <body>

        <header>

        </header>
        <main>
            <h1>Heading 1</h1>
            <p>Paragraph</p>
        </main>

        <footer>

        </footer>


    </body>
</html>
```

```
<!doctype html>
<html lang="en">
    <head>
        <title>Blank html file</title>
        <meta charset="utf-8">
        <link type="text/css" rel="stylesheet" href="css/main.css">
    </head>
    <body>

        <header>

        </header>
        <main>
            <h1>Heading 1</h1>
            <p>Paragraph</p>
        </main>

        <footer>

        </footer>


        <script src="https://code.jquery.com/jquery-1.12.0.min.js"></script>
        <script src="js/main.js"></script>
    </body>
</html>
```

FEWD – LESSON 16

REVIEW REFACTOR DEBUG

What are we going to cover today?

- Refactor
- The Switch statement
- “this” keyword
- Debug techniques
- Revisit HTML5 Boilerplate

LEARNING OBJECTIVES

- Apply switch blocks as a replacement for if/else if/else
- Describe the concept of "this" as it applies within jQuery anonymous functions
- Define refactoring and describe why it is important.
- Learn the basics of CSS/JS refactoring and be able to apply these concepts to their own code
- Know the different ways to debug code and how to apply the concepts to their own code

FEWD – LESSON 16

REFACTOR

What is Refactoring?

- Making code more efficient without changing functionality.

Why Refactor?

- The process of rewriting code without changing functionality
  - To reduce or eliminate redundancy
  - Make code easier to read
  - Make code more maintainable

CSS Refactor

- Remove inline styling
- Replace repeated styles with classes
- Rename classes/ids for readability
- Organize CSS - (https://smacss.com/ worth a look)
- Group by section
- Order by precedence (tag selectors at top, id selectors at bottom)
- Create classes for large CSS changes in JS
- Remove unnecessary css

JS Refactor

- Use functions
- Use variables
- Use arrays
- Use a switch statement instead of a long line of if/else
- Combine jQuery selectors
- Combine jQuery property changes into objects
  - .css,.attr,etc
- Chain jQuery function calls

Refactor code along

- Let’s have a look at some examples

FEWD – LESSON 16

SWITCH

Switch Statement

- A switch statement can replace multiple if/else if /else if statements.
- Can only do so if the condition being tested is the same in each case.

Switch Statement

This can be replaced with a switch statement because all continual tests are the same

```
if (userEntry === 'NYC') {
 ...
} else if (userEntry === 'SYD') {
 ...
} else if (userEntry === 'LA') {
 ...
}
```

Switch Statement

We can’t use it here though:

```
if (userEntry === 'NYC') {
 ...
} else if (invoiceTotal >= 100) {
 ...
} else if (heaterIsON) {
 ...
}
```

Switch Statement

```
switch (<thing to test>) {
    case <test value 1>:
        ...
        break;
    case <test value 2>:
        ...
        break;
    default:
        ...
}
```

Switch Statement Example

```
switch (userEntry) {
   case 'NYC':
      ...
      break;
    case 'SYD':
      ...
      break;
    default:
      ...
}
```

Switch Statement Example

```
switch (userEntry) {
    case 'New York':
    case 'NYC':
        ...
        break;
    case 'Sydney':
    case 'SYD':
        ...
        break;
    default:
        ...
}
```

KEYWORD: THIS

Keyword: “This”

- jQuery: “this” refers to the selected object

Keyword: “This”

- How about this code?

```
$(“p”).on(“click”,function(e){
    $(this).fadeOut(500);
});
```

- Rule of thumb (ROT): If I don’t know what thing will be acted on, then I should consider using “this”

Keyword: “This”

- Let’s see how we can improve on some code with “this”
  - color_scheme

FEWD – LESSON 16

DEBUGGING

Debugging

- Why isn't this working?

Debugging

- Always start be defining the problem.
  - “The image is not moving”
  - “None of my code works”

Debugging

- This will tell you where to start your hunt
  - Image not moving
    - find the code that makes the image move
  - None of my code works
    - Syntax error, check console

Debugging: Level 1

- Check for errors (red text, aligned right) in console To access debugging consoleImage not moving

  PC: CTRL+SHIFT+J

  Mac: COMMAND+OPTION+J

- Click the error
- The location may not be correct but is a good place to start
  Ex: Unbalanced brackets or parentheses

Debugging: Level 2

- So no red errors but not getting the right answer? Try console.log
- Ex:

```
var stringOfNames="";
["Bob","Joe"].forEach(function(element){
    stringOfNames-=element+",";
    console.log(stringOfNames);
});
```

Debugging: Level 3

- Use the debugger in Chrome
- Set a breakpoint
- Run the code
- Step through the code until you get to the error
- Variable values display on the right
- You can switch to the console to run code or check value of variable

Debugging: Level 4

- Get help!
  - Try “Your preferred search engine” search
  - Be ready to clearly articulate the problem (Write out what your problem is)
  - If nothing, ask instructor

Debugging

- Let’s have a look at an example.

FEWD – LESSON 16

HTML5 BOILERPLATE