

MONUMENT LAB – PART 2

WEEKLY OVERVIEW

WEEK 3

Lab / Refactoring & Lab

WEEK 4

Forms & Bootstrap / Intro to JS

WEEK 5

jQuery Part 1 / jQuery Part 2

LEARNING OBJECTIVES

- › Be able to apply basic techniques for refactoring code
- › Practice web development by transforming a design comp into a webpage.

AGENDA

Final Project Overview

Exit Ticket Questions

Display Review

Flexbox Review

Monument Lab — Part 2

FINAL PROJECT OVERVIEW

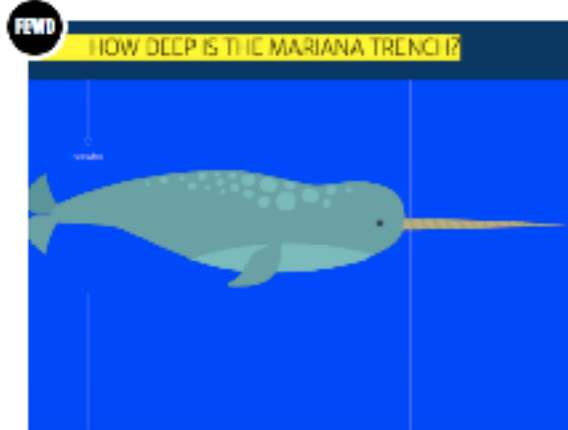
WHERE CAN I FIND PAST EXAMPLES?



Front End Web Development

All Cities

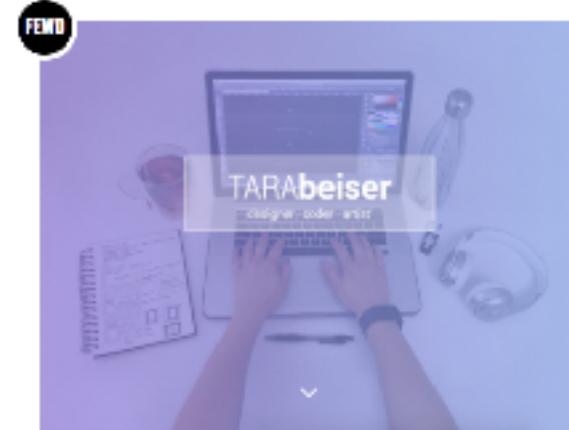
LOGIN



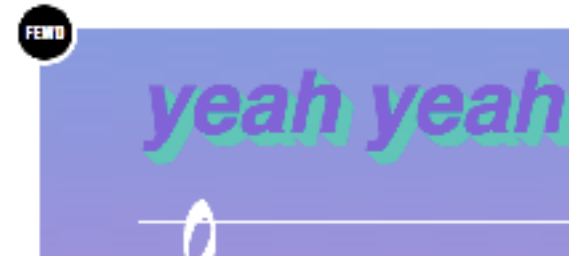
DEPTH OF THE MARIANA TRENCH NYC
by Calie Myers-Wood



HELEN MACDOUGALL STD
by Helen MacDougall

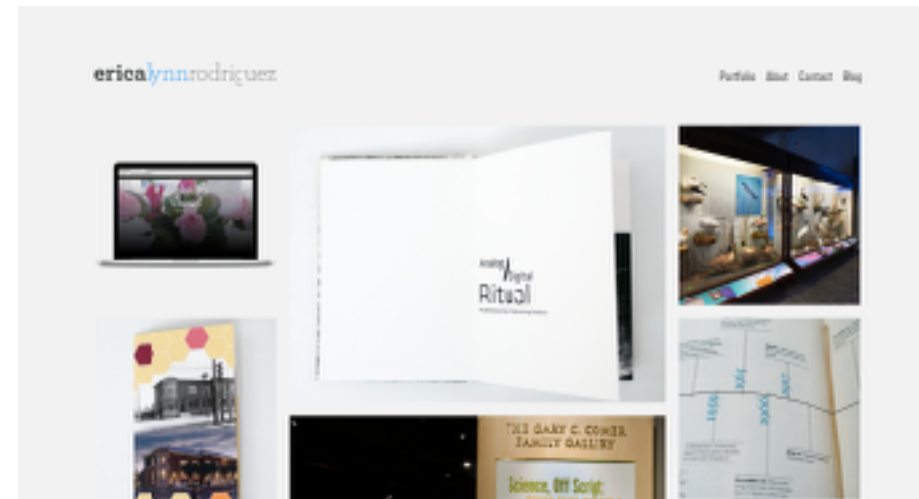
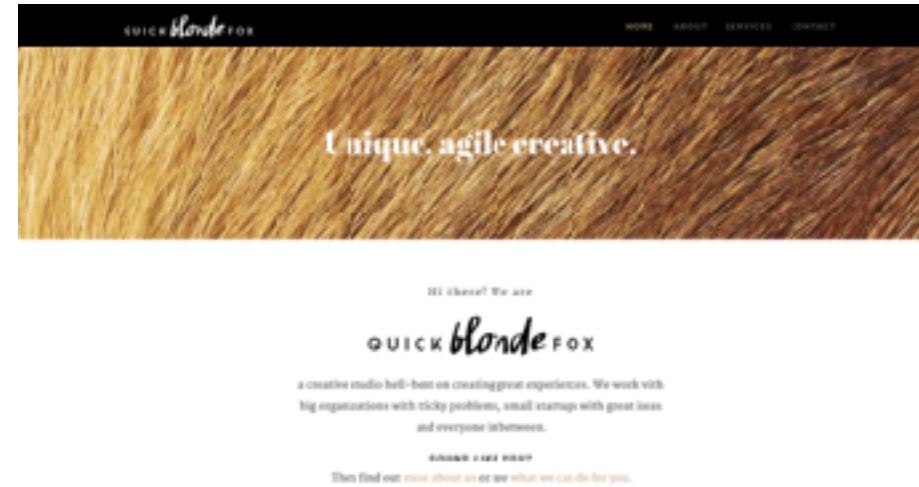


ONLINE PORTFOLIO LA
by Tara Beiser



WHAT KINDS OF PROJECTS TRADITIONALLY MAKE FOR GOOD FINAL PROJECTS?

Portfolio Sites



WHAT KINDS OF PROJECTS TRADITIONALLY MAKE FOR GOOD FINAL PROJECTS?

Sites about hobbies or sites just for fun/practice.



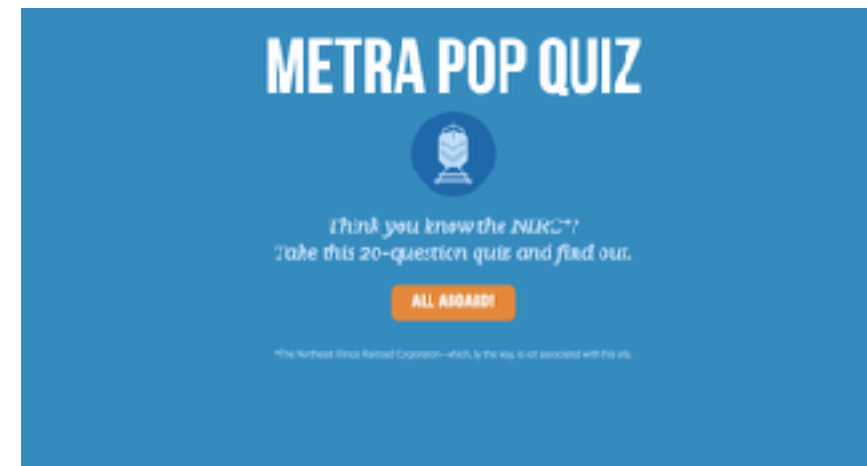
WHAT KINDS OF PROJECTS TRADITIONALLY MAKE FOR GOOD FINAL PROJECTS?

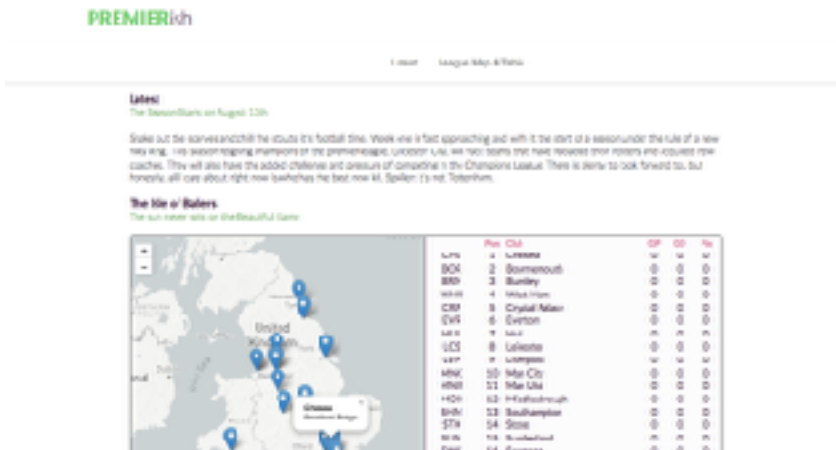
Redesign an existing site for family/friends or just any business with an ugly site :)



WHAT KINDS OF PROJECTS TRADITIONALLY MAKE FOR GOOD FINAL PROJECTS?

Quiz Site / Game





REFACTORING

REFACTOR — HTML

- Always indent nested elements with 2-space tabs
- Close tags immediately after you open them so you don't forget later
- Nothing should be naked — all text should be wrapped in an HTML tag
- Keep project folders organized — the index.html should be in the root directory, images etc should be in sub-directories named as such
- Always opt for the more semantic HTML5 tags instead of plain <div>s when appropriate
- Before you finish a project, run your code through a [validator](#) to catch any errors.
- Don't use
 tags to add space, only when you need a line break. Instead, use CSS padding and margin.
- Headings should be hierarchical (h1 for most important heading, h2 - h6 for subheadings).

REFACTORING CSS

- Organize CSS
- Format CSS declarations with each property on a new line, indented with 2-space tabs (ex: <http://d.pr/i/17ISs>)
- Avoid being super specific with your CSS declarations - keep to two to three levels of indentation when nesting
- Try and write your CSS declarations from the most generic declarations (body, class-less elements), to the most specific (classes, ids, specific sections/pages) from top-to-bottom
- Be wary of creating classes or ID names similar to names used by CSS and HTML e.g. a class named 'body' or 'header'
- When wrapping up a project, run CSS through [validator](#) to check for errors.
- When wrapping up a project, run CSS through [Autoprefixer](#) or [pleeease](#) to add browser prefixes.

TOOLS OF THE TRADE!

- [Sip](#) — Color Picker for Mac
- [Tape](#) — Chrome browser extension (Be sure to go to extensions and make sure it has access to your file URLs in order for it to work!)
- **Inspect Element!!!**

LAB PART II

LAB – MONUMENT



EXERCISE

KEY OBJECTIVE

- Demonstrate the ability to plan and build a website

LAB SESSION

- Until 9:15*
1. Get everything where it needs to be! Add styles for page structure (flexbox-containers, wrappers for columns). Use "*Steps to achieve a multi-column layout*" in review section or your handout for reference
 2. Add base Styles (base fonts, colors, etc.)
 3. Style everything else!

BONUS TASKS (GIFS IN STARTER CODE)

1. Add a hover effect to anchors, buttons, social media icons. Extra bonus: Look up "transition effect hover CSS) to add a smoother transition to hover effects.
2. Create a fixed background effect on scroll for header image
3. Turn the nav into a "sticky nav" so that it stays fixed while the user scrolls.
4. Add an effect to the images in the gallery on hover.
5. Google and implement "Jump Links" so that clicking on a nav item jumps you to that section of the page

LAB – MONUMENT



EXERCISE

REQUIREMENTS

1. Use HTML5 structural elements (nav, header, footer, section, aside, article)
2. Use Flexbox in your CSS to achieve a three-column layout for articles
3. Add a background image to the header.
4. Style your text with the Google Fonts provided by your style guide
5. Add FontAwesome Icons in footer for social media
6. Use indentation in your HTML and CSS files to make your code readable.
7. Use [HTML](#) and [CSS](#) validators to check for errors before submitting.
8. Add browser prefixes using [Autoprefixer](#) or [Pleease](#) before submitting.

BONUS TASKS (GIFS IN STARTER CODE)

1. Add a hover effect to anchors, buttons, social media icons. Extra bonus: Look up "transition effect hover CSS) to add a smoother transition to hover effects.
2. Create a fixed background effect on scroll for header image
3. Turn the nav into a "sticky nav" so that it stays fixed while the user scrolls.
4. Add an effect to the images in the gallery on hover.
5. Google and implement "Jump Links" so that clicking on a nav item jumps you to that section of the page

LEARNING OBJECTIVES

- Be able to apply basic techniques for refactoring code
- Practice web development by transforming a design comp into a webpage.

WEEKLY OVERVIEW

WEEK 3

Lab / Refactoring & Lab

WEEK 4

Forms & Bootstrap / Intro to JS

WEEK 5

jQuery Part 1 / jQuery Part 2

EXIT TICKETS!