

Portland State CS 410 Summer 2022 - Intro to Web Development Journal

Jonas Persson

August 6, 2022

template

- definition:
- description:

7/31/2022

- Created the GitHub repository
<https://github.com/JonasPerssonDevelops/JonasPerssonDevelops.github.io>
for containing the source code and deploying it to
<https://jonasperssondevelops.github.io>
- Studied the requirements in **Final Project - Summer.pdf**
- Wrote up a top-design design for the web app, deciding to go with multiple HTML files that are populated with elements and content dynamically through JavaScript, especially the footer and navbar
- Researched:
 - Whether to use HTML entities or not for content characters (answer: only for `<`, `>`, and `&`)
 - Storing data in the HTML file to identify the page in order to run custom functions for each page in JavaScript (which can be accomplished using `data-*` attributes in the HTML)
 - JavaScript file, directory, and identifier naming conventions
 - Directory structure for assets for web apps
- Tested with a skeleton HTML file and a `.js` file whether JavaScript can identify a page by `data-*` attributes. Success! Happy days are here again...

8/1/2022

- Continued to study the requirements in **Final Project - Summer.pdf** and top down plan the web app
- Researched semantic HTML for the project cards, and for the overall structure of each HTML page, deciding to go with a article for each card and a header, main, and footer on each page
- Decided that the footer should have social media and github repo links
- Wrote a first version of the GitHub repository README.md and committed it
- Checked out some professional bio websites for content inspiration <https://www.chrisducker.com/>, <https://www.marieforleo.com/>
- Researched:
 - GitHub flavor markdown syntax

- How to use tags header, main, article, section, aside, and footer together on the same page
- Placing the nav outside or inside the header (either seems to be ok)
- Website trends and how standard layouts for 2021 websites look
- How to nest main, section, and article content
- Spent 1 hour looking at different websites with this guide
<https://www.intechinc.com/blog/top-50-examples-best-responsive-designs/> phone viewport sizes
- Wrote up a high-level architecture for each page with the following structure: `<nav>`, `<header>`, `<main>`, `sections`, `articles....`, `<footer>`

8/2/2022

- Researched:
 - Creating entirely dynamic pages versus several pages with dynamic elements; Going with several pages to allow the user to bookmark different pages easier
 - The directory structure of GitHub Pages
 - Using CSS Grid instead of bootstrap to have more control over styling
- Reimplemented the navbar from the homeworks
- Added ARIA accessibility markup to the navigation bar

8/3/2022

- Implemented a responsive header and navbar with CSS grid to display decorative text next to a responsive navbar.
- Researched:
 - How to keep my name as decorative text hidden from screen readers so it isn't read out on every page. Solved with `aria-hidden="true"`
- Added ARIA accessibility markup to the navigation bar.
- Linked a Google font to style the decorative text in

8/4/2022

- Implemented JavaScript functions that dynamically build the navigation bar.
- Hid the social button icons from screen readers with `aria-hidden = "true"` as they don't add any functionality for people who are blind and they aren't part of the content of this personal website.
- Hid the social button icon links from screen readers with `tabindex = -1`

- Researched:
 - How to add link text in JavaScript.
 - How to add a node in JavaScript.
 - Social buttons and rules for using copyrighted images from Meta Platforms and GitHub with them
 - Is the syntax double or single quotes for JavaScript code
 - How to center cell content in CSS grid
 - Method for getting elements by id in JavaScript
 - What the proper method for adding text as a child to an HTML tag is
 - Correct semantic HTML for social button links
 - Hiding redundant links from screen readers with `tabindex = -1`

8/5/2022

- Added, tested, and tweaked responsive font sizing with `rem` and a formula for font size: `font-size: calc(90% + 0.5vw);` in order to achieve about 55 characters / 9 words per line on mobile
- Added, tested, and tweaked responsive image sizing with `rem` and `@media` queries.
- Tried a new CSS grid format for the footer but the complexity didn't add any benefit so abandon ship, and reverted to a flex-box space-evenly solution instead.
- Formatted the markdown file as text wasn't centering properly.
- Fixed the social button links so they open in a new tab.
- Tried to get the footer to stick to the bottom but gave up and saw that major websites don't bother to do that.
- Tested different color schemes.
- Tested resizing images proportionally to the viewport width but ran into problems as svg images don't have sizes so it appears impossible to scale them relatively, couldn't find a solution anyways
- Researched:
 - Semantic footer examples
 - Optimal number of characters per line: 50-75 according to <https://baymard.com/blog/line-length-readability>
 - Project directory structure for storing documentation

8/6/2022

- Styled the navigation buttons to make them look like contemporary CSS web design, but also look professional/conservative in styling (because it's a professional bio / personal website).
- Tweaked the header and footer gradients.
- Gave main a circular look with rounded edges using a transition div
- Added padding to main so that the content of main doesn't touch the sides of the windows.
- Restyled the links in the footer using a CSS descendant selector removing the default colors and underline
- Added major colors for gradients and titles on different pages as variable names in `:root`
- Changed title font to Oswald
- Restyled the `media` queries to position the text better for mobile and desktop
- Researched:
 - Semantic footer examples
 - How to combine `calc` and Viewport width / `vw` to have responsive text sizing and padding for main

template

- definition:
- description: