Module 2 Lesson Plans

Weeks 6 - 8 focuses on navigation

Day 15 - Text Styling & Web Fonts

**What to cover:**

Students can do their own research or look up the properties/values for text styling.

Explain the differences between system safe fonts (CSS Font Stack), Google or Adobe font libraries and self hosted fonts. Walk them through implementing a self hosted font using the @font-face rule.

Point out the performance issues with self hosted font - they are often the largest render-blocking resource and Google fonts will always load faster. If self hosted fonts are required, place them in the media query to avoid having them load on mobile.

Explain FOIT and best practice of using ‘font-display: swap’ property. Google fonts should always be brought in as a link in the markup, never as an import in the CSS.

**What to do:**

Have students fill out a document listing possible font style values.

Demo system safe fonts from CSS Font stack (vs relying on VS code to autocomplete) and demo a web font using @font-face.

Day 16 & 17 - Flex Grow & Single Level Navigation

**What to cover:**

Introduce the properties of flex-grow, flex-shrink, and flex-basis. Build a single level navigation that stacks on mobile and re-arranges to horizontal when room allows. Use flex shorthand properties along with flex-wrap to allow the menu items to grow equally to fill the nav (flex row).

Review SVG’s and how to bring them into a project (img tag, inline, CSS background). When brought in as inline elements there is no request to the server for the image and the SVG can be altered with CSS.

When adding SVG’s to the nav point out how to clean them up when they are generated from a library with bloated code. Make sure to remove any id’s inside the SVG for proper validation.

**What to do:**

Demo - Single Level Nav

Demo - Single Level Nav with Icons

Day 18 & 19 - Dropdown Navigation with accessibility

**What to cover:**

How to structure a multi level navigation in the markup.

Introduce CSS positioning properties of relative, absolute, and relative + absolute and when to use these properties. Introduce z-index.

Style the multi level navigation as stacking on mobile (with both levels visible). Style the horizontal layout with a hover state that show/hides the dropdown. Cover accessibility with hidden navigation and the focus:within state. Demonstrate this with the tab select option and browser voice over. Introduce transition states.

**What to do:**

Demo - Positioning Properties

In-class build #2 - Multi Level Nav

Day 20 & 21 - Toggle Navigation with Js

**What to cover:**

A responsive single level navigation hidden behind a hamburger icon on mobile and showing as a default horizontal navigation when space allows.

Following best practices the JavaScript should be in an external file (no inline Js) and it should simply toggle on and off a class name in the markup. That way no inline styling will be added to the markup, and the styling changes are done in the CSS.

When completed, open the inspector tool to demonstrate how the class name is added or removed in the markup when the button is clicked.

Demo version one is a simple toggle nav with a smooth transition state. Additional versions would include extra features.

**What to do:**

Demo - Toggle Nav version 1

Demo - Toggle Nav version 2 (as a overlay nav)

Demo - Toggle Nav version 3 (with the overlay and content using some absolute positioning as another example of using it)

Day 21 & 22 - Multi Level Toggle Navigation

**What to cover:**

Build a multi level, dropdown navigation that hides behind a toggle icon on mobile. Note: this is not ideal or fully functioning without JavaScript accordions. However, this does teach the HTML structure of it and students can be pushed to take the Js course to learn more on how to create tap interactions vs hover states.

**What to do:**

In-Class Build #3 Toggle Navigation

Day 22- Assignment 2

**What to cover:**

Introduce assignment 2 which focuses on loading and styling different types of web fonts. Font styling and more overall page styling. Completely responsive navigation, including a toggle navigation build with Js, is accessible, adjusts the layout on larger screens, and the correct use of relative and absolute positioning.

This assignment will require Photoshop to extract the assets and to match the styling and sizing.

Students should be encouraged to follow best practices when bring in Js, and they will have to follow the method taught in the course for full marks on the rubric.

**What to do:**

Assignment 2 - Responsive Navigation