**W02: Enhancing Your Course Home Page**

**Overview**

In this assignment you will apply the learning activity concepts to enhance your home page with HTML standards, CSS, and testing with the W3C validators.

**Instructions**

**Learning Activity**

If you have not completed the [An Introduction to CSS](https://byui-cse.github.io/wdd130-ww-course/week02/prepare-css-introduction.html) learning activity, do so now or you will have difficulty following the instructions below.

**HTML: Add <head> meta Elements**

1. Open your course home page, **index.html** for editing.
2. Add two new meta tags to your **head** section, namely after the **title** and before the CSS external file **link** element. These will be the meta description and the meta author.
3. <meta name="description" content="Describe your home page here using keywords like your name, the course, the purpose of the page.">

<meta name="author" content="Your name">

**Description**  
"Specifying a description that includes keywords relating to the content of your page is useful as it has the potential to make your page appear higher in relevant searches performed in search engines" - MDN

**Author**  
"Specifying an author is beneficial in many ways: it is useful to be able to understand who wrote the page, if you have any questions about the content and you would like to contact them. Some content management systems have facilities to automatically extract page author information and make it available for such purposes." - MDN

**HTML: <body>**

1. In the **body** section, add an **aside** element after the existing **main** element and before the **footer**.
2. In the **aside** element, add the following child elements and element content:
   * **h2** - Your state or country.
   * **img** - A picture that can represent your state or country.

Remember that all images must be [optimized](https://byui-cse.github.io/wdd130-ww-course/week01/prepare-images.html).  
This image cannot be more than 100kB in file size.

* + **p** - A brief paragraph describing your state or country.

Check Your Understanding - Example

...

</main>

<aside>

<h2>Utah</h2>

<img src="images/utah.jpg" alt="Utah landscape">

<p>Utah is a state in the western United States. It is bordered by Colorado to the east, Wyoming to the northeast, Idaho to the north, Arizona to the south, and Nevada to the west. It also touches a corner of New Mexico in the southeast. The state is a center of transportation, education, information technology, and research, and a major tourist destination for outdoor recreation. Salt Lake City, the state capital and largest city, is home to the headquarters of The Church of Jesus Christ of Latter-day Saints (LDS Church).</p>

</aside>

<footer>

...

**CSS**

1. Open the **styles.css** file for editing.
2. Add a CSS declaration to the **main** selector to limit its width for now.Check Your Understanding - Example
3. main {
4. max-width: 840px;

}

1. Add CSS declarations for the following properties for an **aside** element selector.
   * width
   * margin
   * border
   * padding
   * background color
   * color

Be sure to check the contrast ratio between your foreground and background colors. [contrast-ratio.com](https://contrast-ratio.com/)

1. Check Your Understanding - Example
2. aside {
3. width: 20rem;
4. margin: 1rem;
5. border: 1px solid rgba(0, 0, 0, 0.2); */\* 1️⃣ \*/*
6. padding: 1rem;
7. background-color: #eee;
8. color: #333;
9. text-align: center;
10. }
11. 1️⃣ The border color uses the rgba color method in order to take advantage of a subtle, transparent border effect. The RGBA value consists of four values: red, green, blue, and alpha. The first three values (0, 0, 0) represent black, and the last value (.2) represents the alpha or opacity of the color. An alpha value of .2 means that the color is mostly transparent, allowing the background to show through. By using this border color, the border appears very light and barely noticeable, while still providing a subtle visual cue that helps to separate elements on a webpage. Additionally, it can help to add a layer of visual interest to the design without being overly distracting or detracting from the overall layout.
12. Set the **img** inside the **aside** to have a width of 200 pixels and a height of auto.

This auto setting ensures that we do not change the images aspect ratio.  
Why is that important?

Check Your Understanding

aside img {

width: 200px;

height: auto;

}

Changing the native aspect ratio of an image will distort or pixelate the image.

1. To not allow the existing **img** CSS selector to affect this new image, change the selector for the existing profile **img** to be more exclusive by adding the **main** parent selector.
2. main img {
3. width: 150px;
4. height: auto;
5. ...

}

1. Change the existing paragraph **p** padding property value to 0 pixels.  
   *Note that any 0 value does not need to specify the units.*

p { padding: 0;}

1. Apply CSS to the **<footer>** element to 1️⃣add a top margin to separate it from the **main** element content, 2️⃣add a top border to design additional distinction from the **main** content, and 3️⃣align any text in the **footer** to the center of its content box.Check Your Understanding
2. footer {
3. margin-top: 15px;
4. border-top: 1px solid #000;
5. text-align: center;

}

**Explanation**  
**margin-top** specifies just the top of the element to have a margin of 15 pixels.  
**border-top** uses a [shorthand](https://www.w3schools.com/csS/css_border_shorthand.asp) property list to set the border-width, border-style, and border-color in one declaration versus three.  
**text-align** sets the text alignment to center.

1. Be sure to test your page locally often using the **Live Server** extension in VS Code. Remember that server tool will render the page in your default browser as a web page.

You do not need to be on the internet to test your page locally when running it with Live Server.

1. **Save** your work.

Video Demonstration: ▶️ [Home Page Enhancement](https://video.byui.edu/media/t/1_5xptpphz) | (10.00 mins)

**Testing**

* Every page in this course will be expected to pass the [development standards checklist](https://byui-cse.github.io/wdd130-ww-course/dev-standards.html).  
  Review your rendered page and markup against the checked items.
* Validate the HTML by running the **Web Developer** extension in your browser. Correct any errors.
* Validate the CSS. Correct any errors.
* Commit and sync your changes to your wdd130 GitHub Pages enabled repository.
* Use this [page evaluation](https://byui-cse.github.io/wdd130-ww-course/grader/w02-homepage.html) tool to self check some of the basic requirements.

**Submission**

1. Return to I-learn to submit your wdd130 GitHub Pages home page URL:

https://yourgithubusername.github.io/wdd130

[**Back**](https://byui-cse.github.io/wdd130-ww-course/week02/index.html)[**I-Learn**](https://byui.instructure.com/calendar/)

**An Introduction to CSS**

**Overview**

CSS (Cascading Style Sheets) is a rule set language that defines the style and presentation of an HTML document. CSS rules select elements from the document and contain style declarations.

"Like HTML, CSS is not a programming language. It's not a markup language either. CSS is a style sheet language. CSS is what you use to selectively style HTML elements ... CSS is used to style and layout web pages — for example, to alter the font, color, size, and spacing of your content, split it into multiple columns, or add animations and other decorative features." - MDN

**Prepare**

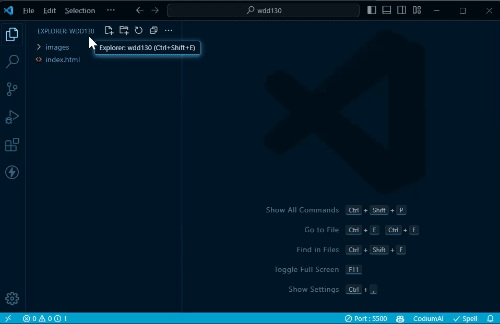
* Video Demonstration: ▶️ [What is CSS?](https://youtu.be/CDQobxknrlE" \t "_blank) | (4:07 mins, [Transcript](https://byui-wdd.github.io/wdd130/text/what_is_css.pdf" \t "_blank))

**Activity Instructions**

In this activity, you will add some style declarations to your course home page.

Currently your home page (index.html) has default styling applied by the browser that renders the page. For example the <h1> element is styled as larger and bolder. In this activity, we will apply custom styling.

**Create CSS Folder and File**

1. In VS Code, with your **wdd130** folder open, create a new folder in the wdd130 root directory named "**styles**".
2. In the new styles folder, add a new file named "**styles.css**"

**Write some CSS**

1. Open the styles.css file and add the following CSS rules.
2. nav {
3. background-color: darkblue;
4. padding: 10px;

}

This CSS rule names the **<nav>** element as the selected element to apply two style declarations.

* 1. The background color will be the color "darkblue", which is a valid, named color.
  2. Padding, or area between the content and border is set to 10 pixels for readability.

nav a {

color: white;

display: block;

padding: 20px;

}

This CSS rule selects all the **<a>** (anchor / link) elements in the HTML that are children of the <nav> element and applies the following declarations:

* 1. The foreground or font color will be white.
  2. The display will change from the default of inline to block.
  3. Each anchor tag will have padding of 20px;

h1 {

color: darkblue;

}

This CSS rule references the **<h1>** (heading 1) element as the selected element and applies the following declaration:

* 1. The foreground, font color is set to darkblue.

img {

width: 150px;

height: auto;

border: 1px solid black;

padding: 2px;

box-shadow: 0 0 30px gray;

}

This CSS rule references all the **<img>** (image) elements as the selector and applies the following declarations:

* 1. a set width with a height that will adjust dynamically to fit the image's aspect ratio.
  2. A border shortcut that names the width, type of border, and color of the border.
  3. Padding between the image and the border.
  4. A box shadow around the image border with zero offset and depth of 30px in gray.

p {

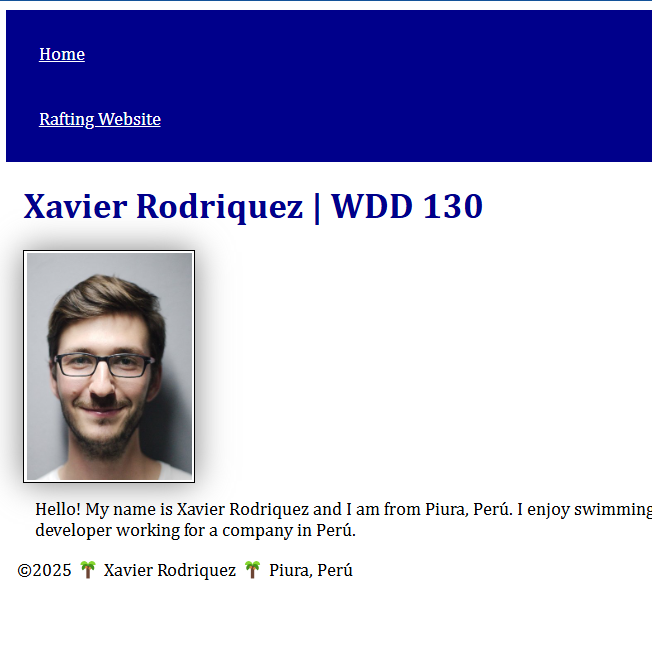
padding: 10px;

}

This CSS rule selects all the **<p>** (paragraph) elements as the selector and applies the following declaration:

* 1. Padding between the paragraph content and the border is set to 10 pixels.

1. Be sure to save your styles.css changes.

Example home page with CSS applied

**Link the CSS file to the HTML page**

1. Open the **index.html** file and add the following line in the **<head>** section of the HTML document.
2. <link rel="stylesheet" href="styles/styles.css">

1. Test the application of the style to your homepage by using **Live Server**. To do this, right mouse click on the file name in the Explorer Panel in VS Code and select **Open Live Server**.
2. Make corrections if needed.

Video Demonstration: ▶️ [Activity Walkthrough](https://video.byui.edu/media/t/1_axrugcro" \t "_blank)

**Remote Render and Test**

1. In the VS Code **Source Control** area, **publish** the site with these additions and changes by committing with a message and syncing to your wdd130 GitHub Pages remote site.

Publishing activities is good practice and the process will become more routine. Remember that the message you write with the commit is for you to give the commit some context.

1. Open up your GitHub Pages rendered URL to test the page.

**Optional Resources**

* [CSS](https://developer.mozilla.org/en-US/docs/Web/CSS) - MDN
* [CSS basics](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics) - MDN

[**Back**](https://byui-cse.github.io/wdd130-ww-course/week02/index.html)[**I-Learn**](https://byui.instructure.com/calendar/)