

WDD 130

[Home](#)[W1](#)[W2](#)[W3](#)[W4](#)[W5](#)[W6](#)[W7](#)[I-Learn](#)

W02 Rafting Project: Site Plan

Overview

In this assignment you will create a site plan for the course project, a rafting site. The site plan will be an HTML document with a supporting CSS file and will be located in a folder that will contain all of your future work on the rafting project. This site plan will provide the blueprint and guides from which to design and development the project.

Project Description

 [The Whitewater Rafting Site Project Description](#)

Instructions

Step 1: Rafting Project Folder and File Setup

1. In VS Code, be sure you have the wdd130 directory open and **add** a folder named "**wwr**" (This folder name stands for white water rafting). This folder will contain the rafting website project and you will add to this folder whenever working on the rafting project throughout the course.
 - Click the **New Folder** icon.

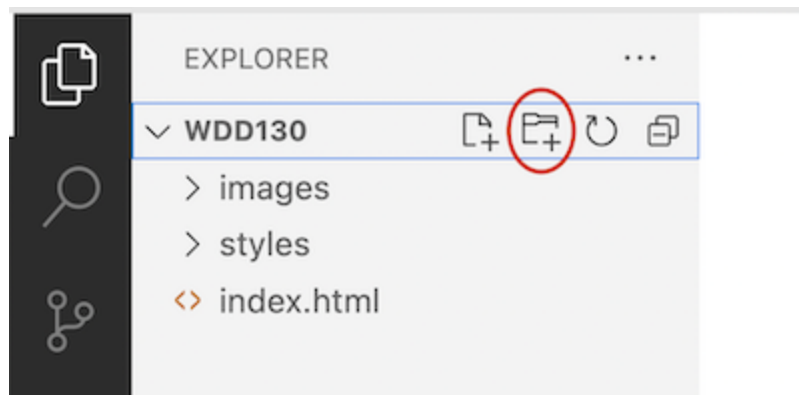


Figure 1: Add new folder icon in the Explorer panel

- Name the new folder "**wwr**".

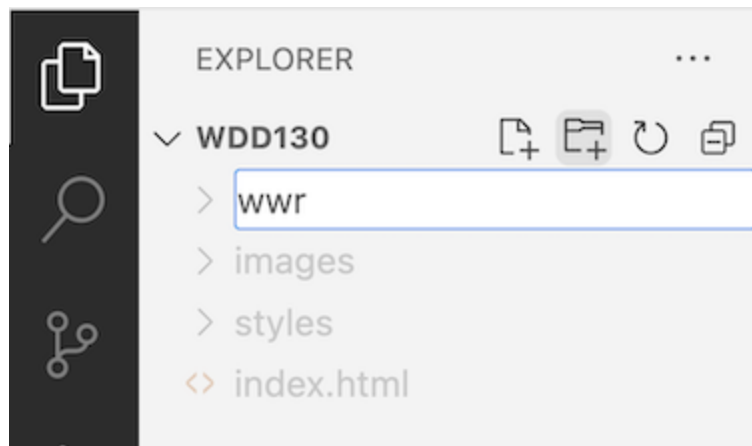


Figure 2: Name the new folder "wwr"

2. **Add** a new file to the **wwr** folder named **"site-plan-rafting.html"**. This will be the site plan.

- Within the wwr folder, click the **New File** icon.

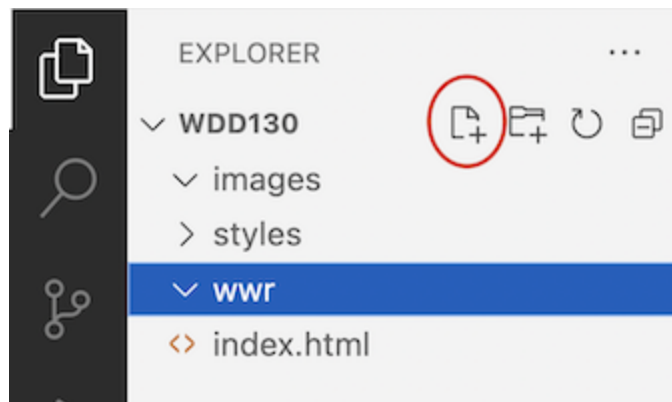


Figure 3: Add a new file

- Name the new file **"site-plan-rafting.html"**

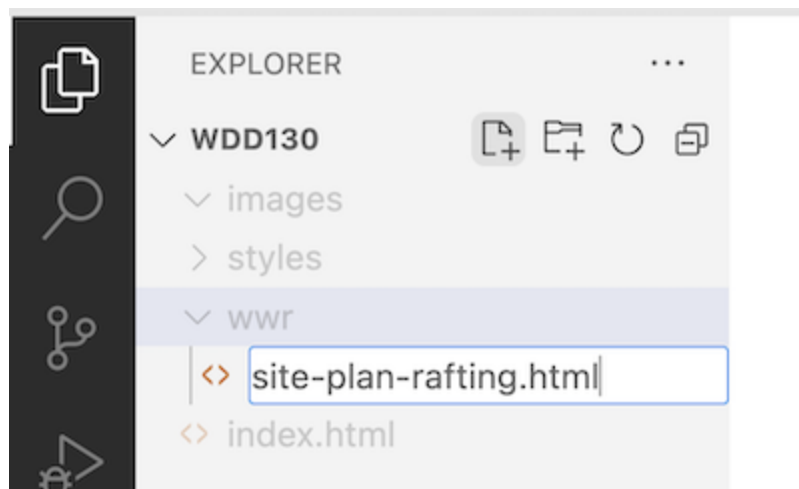


Figure 4: Name the file.

3. Within the **wwr** folder, **add** a folder named "**styles**". This folder will contain the CSS to support the site plan document. Remember to click the **New Folder** icon making sure that you are in the wwr folder.

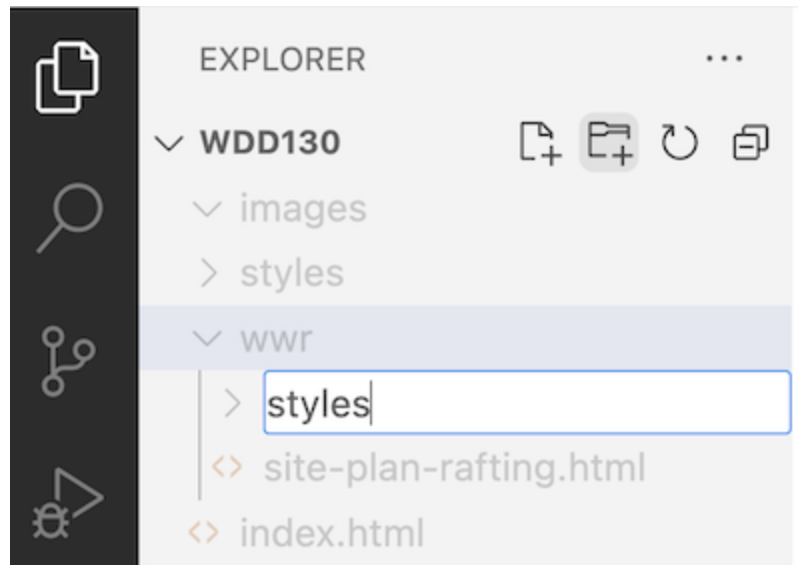


Figure 5: Adding the styles folder to the wwr rafting project folder.

4. **Add** a new file to the **styles** folder named "**site-plan-rafting.css**". Make sure you are in the styles folder when you click the add new file button.

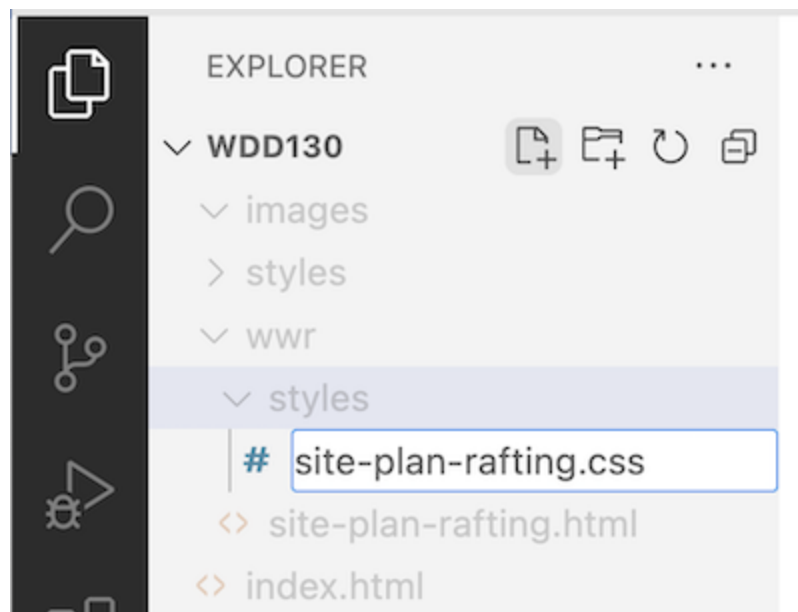


Figure 6: Add a new CSS file

5. Within the **wwr** folder, **add** a folder named "**images**". This folder will contain any images that you used on the rafting site.

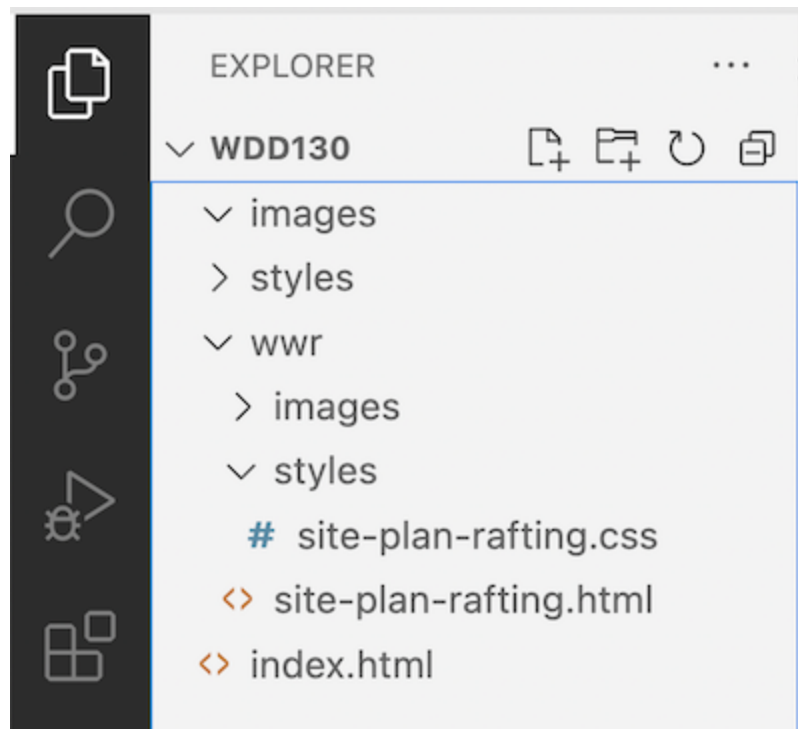


Figure 7: Finished folder structure

Step 2: Write the HTML

1. Open up the **site-plan-rafting.html** file and begin the document by writing the [course standards based](#) HTML markup required in this class for the HTML document and the **<head>** section.

- ☒ Document type definition for HTML5.
- ☒ **<html>** container with a language attribute.
- ☒ **<head>** section with the following elements:
 - ☒ meta charset
 - ☒ meta viewport
 - ☒ title
 - ☒ meta description
 - ☒ meta author
 - ☒ external CSS file reference to styles.css in the styles folder

► Check Your Work

2. The **<body>** section contains the following elements in order:

- ☒ **<header>**. Within the header section, include the following:

- an **<h1>** element that contains the title of the document.
- a **<p>** element that contains your own name as the author.

► Check Your Work

- ☒ **<main>** and include the following elements within the main section. These elements outline the site plan content.

Remember that by definition, an HTML "element" includes an opening and closing tag for non-void HTML elements.

- **<h2>** element with "Overview" as the text content.
- **<h3>** element with "Purpose" as the text content.
- **<p>** element that will contain the purpose text.
- **<h3>** element with "Audience" as the text content.
- **<p>** element that will contain the target audience text.
- **<h2>** element "Branding"
- **<h3>** element "Website Logo"
- **** element that contains these placeholders for the **src** and **alt** attributes

```

```

- **<h2>** element "Style Guide"
- **<h3>** element "Color Palette"
- **<p>** element
- **<h3>** element "Typography"
- **<h4>** element "Headings Example Font"
- **<h4>** element "Normal Text/Paragraph Example Font"
- **<p>** element
- **<h4>** element "Colored Callout Example Font"
- **<p>** element
- **<h3>** element "Navigation"

- **<nav>** element with four (4) children **<a>** anchor tags as shown here with placeholders:

```
<nav>
  <a href="#">Home</a>
  <a href="#">About Us</a>
  <a href="#">Trips</a>
  <a href="#">Contact Us</a>
</nav>
```

- **<h2>** element "Wireframes"
- **<h3>** element "Home Page Wireframe"
- **** element with the **src**, **alt**, and **width** attributes and values set as shown here:

```
** element with the **src**, **alt**, and **width** attributes and values set as shown here:

```


 <h2>Style Guide</h2>
```

```

<h3>Color Palette</h3>
<p></p>
<h3>Typography</h3>
<h4>Headings Example Font</h4>
<h4>Normal Text/Paragraph Example Font</h4>
<p>Example text - Lorem ipsum.</p>
<h4>Colored Callout Example Font</h4>
<p>Example text - Lorem ipsum</p>
<h3>Navigation</h3>
<nav>
 Home
 About Us
 Trips
 Contact Us
</nav>

<h2>Wireframes</h2>
<h3>Home Page Wireframe</h3>

</main>

```

#### ✓ <footer>

- Include an <p> element with a copyright symbol, year, document title, your name, and the course as text content.

#### ▼ Check Your Work (footer)

```

<footer>
 <p>©2024 🌴 Rafting Site Document 🌴 Xavier Rodrique
</footer>

```

## Step 3: Page Content

1. Fill out the <main> section empty paragraphs or other elements with appropriate content.

- ✓ **Purpose:** Imagine that you are the owner of the rafting business and have decided that it is time for you to have an online presence. What is the purpose?

You might need to do a little research if you are unfamiliar with water rafting companies.

- ✓ **Audience:** Write a statement specifically identifying your target audience for a water rafting website. Review the example questions posed about target audiences from the [learning activity](#) on site plan sections. Include three (3) scenario questions. Include both the scenario questions and answers to those questions.
- ✓ **Logo:** Choose or create a logo for your website. Here are some example logos that you can use freely - [Rafting Logos](#). You can adjust your **<h1>** content to match the name of the rafting company that you choose in this document. Remember to save the logo to the **images** folder within your **wwr** folder.

If you choose a logo from those provided in the linked document, you can right click on the image and **Save image as...** a file to the **images** folder within your **wwr** folder.

- ✓ **Color Palette:** Using the [colors.co](#) tool, choose a color scheme for your rafting site, keeping your site purpose and audience in mind. Start by thinking of a color you like or think will work well for a rafting site. Once you have built your palette, note that at the bottom of each color column there is a six digit hex code. This is what is known as a hexadecimal color code (base 16 or 0 to F) and is usually preceded by a "#". Copy that code for the 3–4 colors you would like to use into the site plan document. **List** the colors you have selected in the site plan paragraph using a structure of your choice below the color palette heading. Include the 'name' of the color (given below) and the hexadecimal color code. These are the four (4) labeled 'names' that we will use in the CSS.

- **primary-color**
- **secondary-color**
- **accent1-color**
- **accent2-color**

Do not feel like you will be stuck with these colors. Changes can be made to the site plan after

You should save your chosen palette URL from colors.co in case you need to make any changes later. Copy the URL of your palette. (it might look something



like this: <https://colors.co/b7d3f2-afafdc-8a84e2-84afe6-79beee>) and paste it into this same paragraph area in the site plan.

```
/396e94-e7c24f-a43312-381d2a-aabd8c" target="_blank">https://coo
```

Notice that when working with links in HTML the destination of the link goes in the src attribute. Then the part that is displayed to select goes inside of the `<a>` element. Usually those will be different, however, in this case we are going to display the URL to your color scheme on colors.co.

#### ✓ **Typography:**

Using the [Google Fonts API](#), choose one or two fonts to use for the rafting site and record the names of the fonts and the category of font (serif, sans-serif, display, handwriting, or monospace) in the site plan under the appropriate heading provide for headings, paragraphs, and the callout box.

Do you remember how to work with fonts from the [Website Planning Document](#) learning activity? Here is link to the tutorial on working with fonts: [CSS Fonts](#). Be sure to only use one import statement in your CSS even if you are using more than one font.

- ✓ **Wireframes:** You do not need to change anything in this section. This is an example wireframe for you to study. The given wireframe in this assignment will be used to develop the home page of the rafting site in a future week. In addition, the "About Us" wireframe will be provided when you are assigned to build that page. You will have the chance to design your own wireframe for the other pages.

## Step 4: Write the CSS

1. Open up your **site-plan-rafting.css** file.
2. Add an **@import** reference from the Google Fonts API to use the font(s) you have selected.

#### ▼ Example (@import CSS statement)

```
@import url('https://fonts.googleapis.com/css2?family=font
```

💡 This assignment and project uses the `@import` reference to employ Google Fonts. Another way to load fonts on a page is to use the `<link>` element in the HTML head. Doing so takes advantage of performance and prioritization gains versus `@import`. You will find this method in some videos and other reference sources.

3. Write a **CSS rule** for the **body** of the site plan document to:

- ☒ set the default **font-family** to equal the **paragraph-font** CSS variable value. You will need to use the **var()** function to access the CSS variable.
- ☒ limit the width (**max-width**) of the document output to the screen to **960 pixels**, and
- ☒ set the **padding** around the document to **20 pixels**.

▼ Example (body selector rule)

```
body {
```

4. Next, define CSS variables in the document `:root` pseudo-class selector. In this rule define CSS variables for the following components.

- ☒ primary color
- ☒ secondary color
- ☒ accent color
- ☒ additional accent color
- ☒ heading font
- ☒ paragraph or default font
- ☒ navigation background color
- ☒ navigation link color
- ☒ navigation hover link color
- ☒ navigation hover background color

▼ Example (CSS Variables)

```
:root {
 --primary-color: #XXXXXX;
```

```
--secondary-color: #XXXXXX;
--accent1-color: #XXXXXX;
--accent2-color: #XXXXXX;

--heading-font: '_____', serif;
--paragraph-font: '_____', serif;

--nav-background-color: #XXXXXX;
--nav-link-color: #XXXXXX;
--nav-hover-link-color: #XXXXXX;
--nav-hover-background-color: #XXXXXX;
}
```

You will use your selected color schema and font family choices in your site plan. The following steps help you do that using these CSS variables.

5. Write a CSS rule for the body of the site plan document to:

- ☒ set the default font-family to equal the paragraph-font CSS variable value.  
You will need to use the `var()` function to access the CSS variable.
- ☒ limit the width (max-width) of the document output to the screen to 960 pixels, and
- ☒ set the padding around the document to 20 pixels.

► Example (body selector rule)

6. Group all the built-in headings (h1 - h6) using a grouping selector. Set the group's font-family to the CSS variable established in the document root for heading-font and give the headings the color of your choice.

► Example (headings grouping selector)

7. Set the header to have padding and a background-color using your selected colors for the site.

► Example (header selector)


8. Give the paragraphs appropriate padding.

► Example (p selector)

9. Use the following CSS as a starting point to provide some style and presentation to the <nav> element in the Navigation plan and to its anchor tag children accounting for .

► Example (nav selector and anchor tag children)

## Testing & Submission

1. Test your work continuously as you work through the steps by having your page loaded in your local browser using Live Server. You do NOT need to be on the internet while developing your pages.
2. Validate and correct any errors with your HTML and CSS using the [Web Developer](#) browser extension under Tools and Validate Local HTML and CSS.
3. When you are finished, commit your changes and push them to your wdd130 GitHub Pages repository.
4. Evaluate your own work by using this  [W02: Project Site Plan Page Evaluation](#) tool.
5. Submit the site plan URL (web address) in I-Learn. The URL will look something like this:

`https://githubusername.github.io/wdd130/wwr/site-plan-rafting.html`

