

# Create the HTML for the Home Page

---

## Activity Directions

Estimated Time: 60 minutes

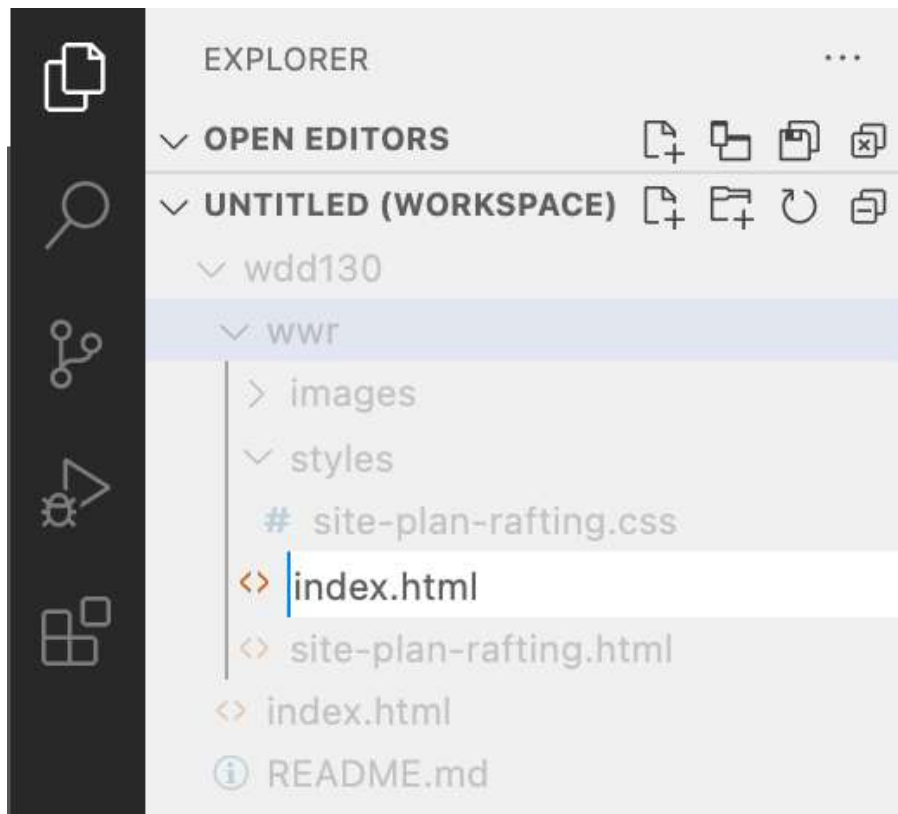
w3schools.com is a great reference site for HTML and CSS. Here are a list of concepts to brush up on for this assignment. These should be a review from past preparation materials and assignments.

- [HTML Basics](#)
- [HTML Elements](#)
- [HTML Attributes](#)
- [HTML Headings](#)
- [HTML Paragraphs](#)
- [HTML Images](#)
- [HTML Lists](#)

## Create the Home page for your wwr project

If you completed the GitHub Pages Hosting activity, you should already have an HTML file ready called index.html. This index.html is for our wdd130 repository's home page. We need a home page for just the wwr project as well. This will be the home page for the White Water Rafting site.

Add a new file within the wwr folder called index.html.



Create the wwr home page index.html

Use the VSCode shortcut of '!' + enter to create an HTML skeleton:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>

</body>
</html>
```

## Change the title

Let's go ahead and change the title next. The `<title>` of the page shows at the top of the browser window or on the tab of a browser window and in the search listing for a site. Let's pick something

that will make sure people know what the site is and what page they are currently on. Something like "Whitewater Rafting Vacations | Dry Oar Boating | Home".

We know from the [wireframe](#) for the home page for the Dry Oar site that there will be a combination of images, headlines, and text on the page. You can refer back to the wireframe and site plan document throughout this activity. The instructions will make much more sense if you are looking at the blueprint!

As you look at the wireframe you will notice a few things. First a box with an X through it indicates an image will go there.

The wireframe also indicates where our headlines and paragraphs will be. Elements are organized into groups. It will be important that we include html markup to represent those groups, as well as the content on the page.

### Note

We will **not** be adding colors and other styles this week, so no CSS. Our only goal is to get all of the content onto the page that is shown on the wireframe. Our Page will **not** look like the wireframe when we finish this week. We are only using it as a guide to get the right HTML elements and content. Each week will get us closer, but our page will not look exactly like the wireframe for several weeks, so patience will be required.

## "Framing" with semantic elements

Building a website is actually a lot like building a house. You start with a blueprint (wireframe) and then you build the foundation and frame for the house. We have already built the foundation. We did that when the `<html>`, `<head>`, and `<body>` elements were added. The framing takes the rooms outlined on the blueprint and creates a real space to put them. We need to do the same thing for our webpage, except for instead of rooms we are looking at content areas.

The way we "frame" in our webpage is by looking carefully at our wireframe to see what "rooms" we have. Then we use some special HTML tags to define them.

Take a moment to review the [wireframe](#) again to help remember what we are trying to do. We need to add some markup to create containers in our HTML.

HTML has several tags specifically for this purpose. Some have specific semantic meanings and others don't. Here is a list of some of the more common of these tags: `<header>`, `<footer>`, `<main>`, `<section>`, `<nav>`, `<article>`, `<figure>`, `<aside>`, `<div>` and `<span>`.

w3schools.com explains what these elements contain in [HTML Semantics](#).

All of these elements are used to show different sections of the page. Try to use the more specific ones first, using a generic `<div>` only if none of the others seems to apply to the content you are boxing in.

## Header

Add HTML elements to your body to start outlining content areas. Make sure and indent child elements within parent elements. As the page gets larger, you will be happy you did. It helps a lot with organization and being able to see mistakes quickly.

The first element we need is at the top of the page. Notice that there is a box around the logo and navigation bar. Those elements are related and form the header of the page. Add the `<header>` tag.

Add an `<img>` element for the logo. Let's give it a class, `class="logo"`. We want to differentiate this `<img>` tag from all others on the page. We will eventually have the logo inside the images folder, and it will be called logo.png. It won't show up on the screen until we do that. There are two required attributes for images. We will add the source attribute and value as `src="images/logo.png"`. And we always need an alt attribute with every image, so add `alt="Dry Oar Logo"`

Users usually expect the logo to take them to the home page of the site, so we will make the `<img>` element the content of an `<a>` element with the hypertext reference of going to the home page `href="index.html"` and add `id="logo_link"`

The `<nav>` element is the perfect semantic tag for the navigation or menu. We will have four pages eventually. So we will add four `<a>` elements inside the `<nav>`

We know the hypertext reference for the home page, and we will all have a contactus.html page eventually, and we'll include our site plan as well, but we don't know the name of our second page so we'll leave the value of the href with a placeholder of a hashtag # and the content as Page 02 for now.

Home pages are always expected as the first link and Contact or About Us pages are usually last on a menu.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Whitewater Rafting Vacations | Dry Oar Boating | Home</title>
</head>
<body>
  <header>
    <a id="logo_link" href="index.html">
      
    </a>
    <nav>
      <a href="index.html">Home</a>
      <a href="#">Page 2</a>
      <a href="site-plan-rafting.html">Site Plan</a>
      <a href="contactus.html">Contact Us</a>
    </nav>
  </header>
</body>
</html>
```

### Tip

There will be a lot of code examples in this and other lessons. Avoid the temptation to just copy and paste them into your page. Type them out! This might seem like a bother, but you will remember the HTML and CSS we use much better if you do.

Also, check your page on Live Server as you go to make sure it is showing up OK. Images will be added later.

## Hero

Let's add a container for the hero image and message. The role of a hero image is to welcome your visitors with the first glimpse of your company or organization.

It is something unique to only the home page of your website. We won't see this section, with a large image like this, on any of the other pages and we are going to also keep it separate from the main section of the page, for ease in applying CSS to it later.

Because there isn't a good semantic tag for the hero section, let's use a `<div>` with `id="hero"`.

Inside our hero `<div>` we will have another `<div>` with `id="hero-box"` holding our hero `<img>` with `id="hero-img"`, `src="images/hero.png"`, and `alt="People enjoying rafting"`

A `<section>`, with the id of 'hero-msg', will hold the hero message consisting of an `<h1>` with `class="home-title"` and 'Have An Adventure' content. Also an `<h4>` with content 'Make Memories with Dry Oar'. Lastly, a `<div>` that will hold a button with a link with `class="button-box"`. The `<a>` will have `class="book"` and `href="contactus.html"` and content of 'Book Now'.

Whenever you want to move multiple elements together, like all the hero text, you should group them by placing them *inside* of another element; like we have with the section hero-msg.

Feel free to change any of the text content between tags. But leave the attributes and values, class, id, and element names as instructed.

```
<body>
<header>
  <a id="logo_link" href="index.html">
    
  </a>
  <nav>
    <a href="index.html">Home</a>
    <a href="#">Page 2</a>
    <a href="site-plan-rafting.html">Site Plan</a>
    <a href="contactus.html">Contact Us</a>
  </nav>
</header>
<div id="hero">
  <div id="hero-box">
    
  <section id="hero-msg">
    <h1 class="home-title">Have An Adventure</h1>
    <h4>Make Memories with Dry Oar </h4>
```

```

    <div class='button-box'>
      <a class='book' href="contactus.html">Book Now</a>
    </div>
  </section>
</div>
</body>

```

So you might be asking, how do you know if you use a <section> or a <div>? If you look at what a section contains on [w3schools.org](https://www.w3schools.org) it says a section is a thematic grouping of content, typically with a heading. So that fits the hero text but not the image. I also know later that I will be using a grid to lay the page out and these groupings will help with the grid. As you become more proficient with HTML and CSS, you will be able to predict these things better.

Here is a [flowchart](#) that might help to pick out the most semantic element. Always use the tag that best describes the content.

## Main

Next let's add an area where the main content for this page will go. Add a <main> element to your page right below the hero <div>. It will have `class="home-grid"`

```

<body>
<header>
  <a id="logo_link" href="index.html">
    
  </a>
  <nav>
    <a href="index.html">Home</a>
    <a href="#">Page 2</a>
    <a href="site-plan-rafting.html">Site Plan</a>
    <a href="contactus.html">Contact Us</a>
  </nav>
</header>

```

```

<div id="hero">
  <div id="hero-box">
    
  <section id="hero-msg">
    <h1 class="home-title">Have An Adventure</h1>
    <h4>Make Memories with Dry Oar </h4>
    <div class='button-box'>
      <a class='book' href="contactus.html">Book Now</a>
    </div>
  </section>
</div>
<main class="home-grid">

</main>
</body>

```

The first things we need inside our new `<main>` element is the three cards, each with an image, icon and title. We'll use `<section>` elements to hold each card's content, which consists of two `<img>`s and an `<h2>`. Use the classes and attributes as shown below.

We will use the same class name for the card and icon images because the CSS will be similar for all of them.

```

<main class="home-grid">
  <section class="rivers-card">
    
    
    <h2>Rivers</h2>
  </section>
  <section class="camping-card">
    
    
    <h2>Camping</h2>
  </section>
  <section class="rapids-card">
    
    
    <h2>Rapids</h2>
  </section>
</main>

```



```
    </section>
</main>
```

Below the card sections we will have an `<img>` and a `<section>` containing an `<h2>`, `<p>`, and `<a>` with the classes and attributes as shown below:

Again, feel free to change the title, paragraph, and button to say whatever you want it to.

```
<main class="home-grid">
  <section class="rivers-card">
    
    
    <h2>Rivers</h2>
  </section>
  <section class="camping-card">
    
    
    <h2>Camping</h2>
  </section>
  <section class="rapids-card">
    
    
    <h2>Rapids</h2>
  </section>
  
  <section class="msg">
    <h2>More Than Just The Thrill</h2>
    <p>Enjoy the breathtaking scenery. From valleys, meadows, canyons, and
    <a class='join' href="rivers.html">Join Us</a>
  </section>
</main>
```

## Footer

Finally we need one more container on our page. This one will hold the footer information. Add one below `</main>` with all the attributes as shown below. To make the copyright symbol that is right

before the date, we use a special code. There are many of these, but we will just learn this one today: `&copy;` Fill in the current year instead of 'XX' and put your first and last name after the year.

```
</main>
<footer>
  <p>Dry Oar &copy; 20XX - Your First and Last Name Here</p>
  <p><a href="site-plan-rafting.html">Site Plan</a></p>
  <p><a href="contactus.html">Contact Us</a></p>
  <div class="social">
    <a href="https://facebook.com">
      
    </a>
    <a href="https://twitter.com">
      
    </a>
    <a href="https://instagram.com">
      
    </a>
  </div>
</footer>
```

If we look at our page now with Live Server, all of our images will be broken (not showing up).

We have referenced all of these images that we don't have in our images folder yet.

## Adding images

There are images provided for this home page, but you are welcome to add your own images to your page. Just remember to change your code to reference your images.

I would suggest using the provided images for this home page, otherwise images you bring in might not have the same aspect ratio (or dimensions) as the provided images causing you to have extra editing or CSS you will have to deal with later.

## Finding your own Images

If you opt to look for your own images, here are a couple of things to keep in mind:

- Make sure the image is not restricted for use. These would include images from other whitewater rafting company sites and photographer's sites.
- You can have Google help you try to avoid those and find images that are free to use by searching something like **rafting** in Google...then select the **Images** tab. At the top of the screen you will see a **Tools** button that will show a sub-menu. Select **Usage Rights** from that menu. Select the **Creative Commons licenses** option to show images that are available to use. A few, however, may still have copyrights, so be careful.
- Many of the images you will find will be very large! Don't worry, you can optimize them. See the videos below for information about images and optimizing images.

Whether you are using the images provided or your own images, watch the two videos below to learn more about images.

You may not have images provided for other pages on this site, and you will not have them provided for your personal website you make. You will need this information about images later.

Here are the links to all the images on the home page. Once you open the link and see the image, just right-click the images and choose 'Save Images As...' and then save them into your images folder in your project folder.

[Logo](#) [Hero](#) [Rivers](#) [River icon](#) [Camping](#) [Fire icon](#) [Rapids](#) [Oars icon](#) [Mountains](#) [Facebook](#) [Twitter](#) [Instagram](#)

Alternatively, there are optimized images we've collected for this course you can use here: [Alternative small, medium, and larger rafting images and logos](#)

You could always link them directly to the external area where they are on the Internet. For example `` But this is called 'hot linking' and is not advised because you have no control over those images and they could be gone or changed at any time. It's always best to download the files into your own project in your images folder. Then your image elements would look like this: ``

Your Git repository will notice all the new files coming in and you'll see the little blue circle with the numbers on your Source Control icon in VSCode. Git will also notice that

we have been editing and adding to index.html. You will need to commit all the changed and added files and push them to the repository. (See the end of last week's assignments if you've forgotten how to do that.)

## **Validate your HTML**

Go to <https://validator.w3.org/> and validate your HTML and fix any error you find. Get help if you don't understand some of the errors. They can be tricky sometimes.

## **Submit your URL in I-Learn**

Once validated, make sure you've pushed the latest version of your project to GitHub. Then submit your URL that shows your home page live on the Internet so your instructor can see it.

Always check the URL before you submit. It will look similar to this URL but with your username replacing where it says 'yourusername': <https://yourusername.github.io/wdd130/wwr/index.html>

Your page won't be pretty yet. It will look something like this:



[Home](#) [Rivers](#) [Rates/Dates](#) [Contact Us](#)



## Have An Adventure

Make Memories with Dry Oar

[Book Now](#)



## Rivers





**Camping**



**Rapids**



**More Than Just The Thrill**

Enjoy the breathtaking scenery. From valleys, meadows, canyons, and high peaks; it's way more than just the rapids. It's a great way to get away from it all and relax amongst all the beauty of the great outdoors.

[Join Us](#)

Dry Oar © 2021

[Site Plan](#)

[Contact Us](#)



