**HTML: Semantic Elements**

**Overview**

HTML describes the structure of a web page document. The structure is composed of various categories of elements including the following **content** types: metadata, section, heading, embedded, and interactive elements. Some elements are associated with forms and tabular data. You are **not** expected to memorize all HTML tags. Rather, become familiar with the most commonly used elements and be able to reference others through reliable sources.

"Most HTML elements are a member of one or more content categories — these categories group elements that share common characteristics." - MDN

**Prepare**

**Semantic Elements**

Semantic elements are HTML elements that have a specific meaning and are used to structure the content of a web page. They are used to clearly provide meaning to both the browser and developer.

**Non-semantic Element Examples**: **<div>** and **<span>** are examples of non-semantic elements. These elements do not provide any semantic meaning to the content they contain, but are commonly used with class or id attributes assigned to them.

**Semantic Element Examples**: **<header>**, **<nav>** **<main>**, **<footer>**, **<article>**, **<section>**, **<aside>**, and **<figure>** are some of more common semantic elements. They can be used to identify, with meaning, parts of a web page.

All of these tags are described in detail in the [HTML reference](https://developer.mozilla.org/en-US/docs/Web/HTML/Element) MDN page under the **► HTML elements** tab.

Many web sites contain HTML older HTML constructs like **<div id="header">**, **<div id="navigation">**, **<div id="header">**, etc. These elements are not semantic and should not be used. Students often will even use something like **<header id="header">** or **<header class="header">** which is unnecessary and redundant in most cases. Note that it is OK to use the **<div>** element but if a viable semantic element is available, use it.

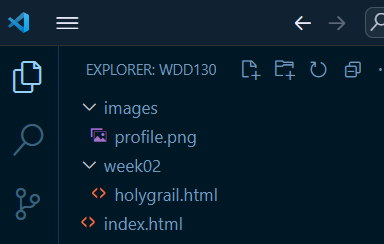
**HTML Headings and Accessibility**

Read the section on [Heading Accessibility Concerns](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/Heading_Elements#accessibility_concerns) - "Properly ordered headings that do not skip levels convey the semantic structure of the page, making it easier to navigate and understand when using assistive technologies." - MDN

**Activity Instructions**

In this activity you will use HTML to build a common page layout with a **header**, **nav**, **main**, **aside**, and a **footer** element.

**Folder and File Setup**

1. In VS Code, create a "**week02**" folder if you have not already.  
   This new folder should be a subfolder of your **wdd130** main folder.
2. Create a new HTML file name "**holygrail.html**" in the **week02** folder. Make sure this HTML file is located in the **week02** sub-folder and **not** on the **wdd130** root/parent folder.File Structure for Week 02 Holy Grail Assignment

**HTML Head Content**

Start building the **holygrail.html** page by including these, baseline, required HTML elements:

Use the first two sections of the [Development Standards Checklist](https://byui-cse.github.io/wdd130-ww-course/dev-standards.html) reference, which you should bookmark, to help you with the how and why on each of these requirements.

* A valid document type. **<!DOCTYPE html>**
* An **<html>** root container with the proper language attribute.
* A **<head>** container.
* The **meta charset** element.
* The **meta viewport** element.
* A **title** element will relevant content.
* The **meta description** element.  
  The content could be: *A common layout example on the web called the 'holy grail' layout*.
* The **meta author** element.

Check Your Understanding

**<!DOCTYPE html>**

**<html lang="en-US">**

**<head>**

**<meta charset="utf-8">**

**<meta name="viewport" content="width=device-width,initial-scale=1.0">**

**<title>Xavier Rodriquez | Holy Grail Layout Example</title>**

**<meta name="description" content="A common layout on the web called the 'holy grail'">**

**<meta name="author" content="Xavier Rodriquez">**

**</head>**

**<body>**

**</body>**

**</html>**

It is OK to use built-in snippets in VS Code that write common code segments for you. For example, in a blank html document, type **html5** and then click the **tab** key. The basic HTML framework will be written automatically.

**HTML Body Content**

Build the common page layout within the **<body>** using the following semantic elements:

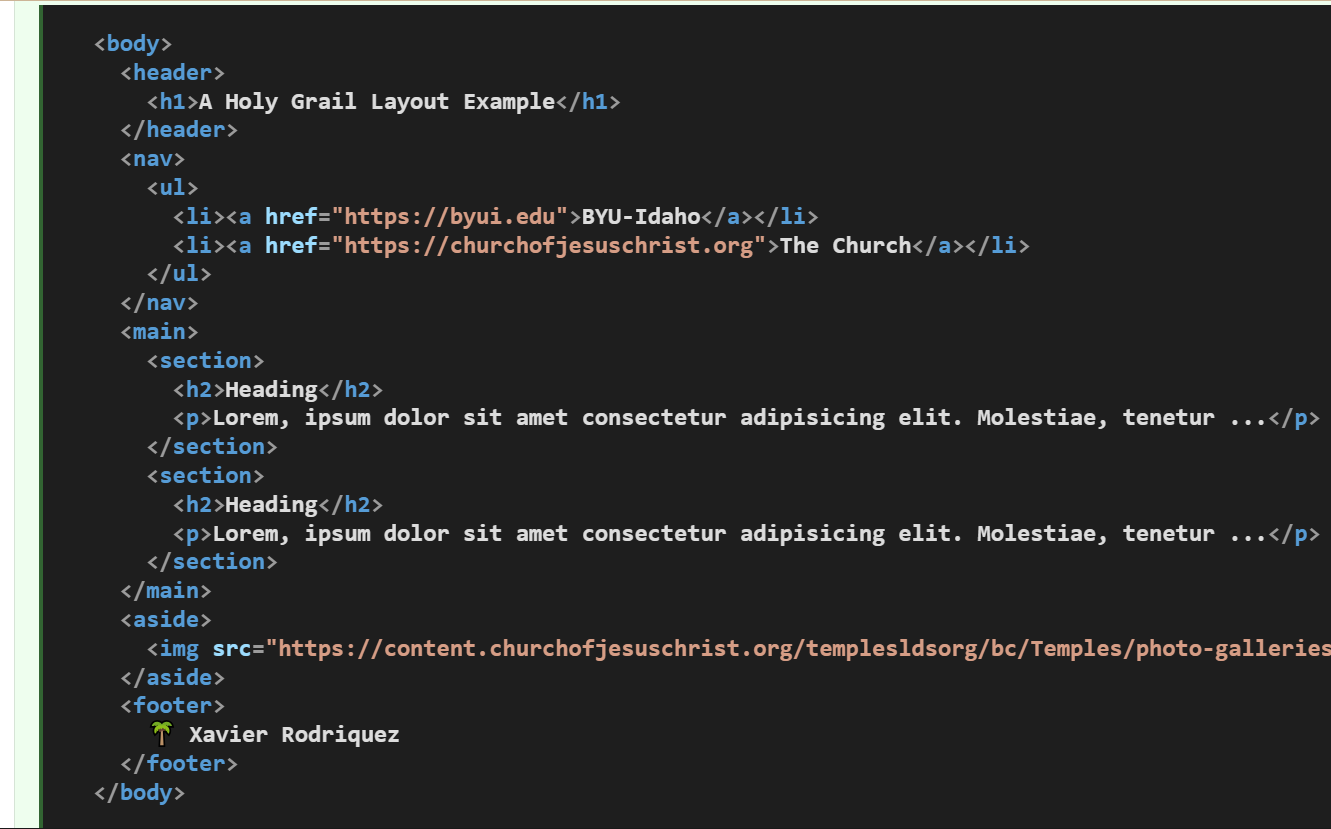
This page will not have any layout or style at this point. It is just the structure. We will add CSS styling and layout in another activity.

1. Start with a **<body>** element that will contain all the page content.
2. Add a **<header>** element.
   * Include a first level heading **<h1>** within the **header**.
   * The content of the heading can be "*A Holy Grail Layout Example*".
3. After the **header** ends (**</header>**), add a **<nav>** element.
   * The **nav** element contains a single unordered list **<ul>** with two list items.
   * The two (2) list items **<li>** each contain a hyperlink **<a>** to an external site of your choice.
4. Next, include a **<main>** element.
   * The **main** element contains two (2) **<section>** elements.
   * Each **section** element contains:
     + A second level heading **<h2>** with the placeholder language of "*Heading*", etc.
     + a single paragraph **<p>**

The paragraph content can be **filler/placeholder** text. To automatically fill the paragraph with [Lorem ipsum text](https://en.wikipedia.org/wiki/Lorem_ipsum), in VS Code type **lorem10** and then press the **Tab** key and ten (10) words will be written automatically. You can change the number 10 to the desired size.

1. Next, include a **<aside>** element.
   * In this container element, include an image element **<img>**.
     + Reference (**src**) a temple image from the [Church Media Gallery](https://www.churchofjesuschrist.org/temples/photo-gallery).  
       Copy the image URL of the image you choose by right mouse clicking on the image and selecting **Copy image address** from the popup menu.
     + Set the width attribute (**width**) to a value of "**200**" which means 200 pixels.
2. Next, include a **<footer>** element.  
   The content of the footer element can just be your name.

Check Your Understanding



**<body>**

**<header>**

**<h1>A Holy Grail Layout Example</h1>**

**</header>**

**<nav>**

**<ul>**

**<li><a href="https://byui.edu">BYU-Idaho</a></li>**

**<li><a href="https://churchofjesuschrist.org">The Church</a></li>**

**</ul>**

**</nav>**

**<main>**

**<section>**

**<h2>Heading</h2>**

**<p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Molestiae, tenetur ...</p>**

**</section>**

**<section>**

**<h2>Heading</h2>**

**<p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Molestiae, tenetur ...</p>**

**</section>**

**</main>**

**<aside>**

**<img src="https://content.churchofjesuschrist.org/templesldsorg/bc/Temples/photo-galleries/payson-utah/800x500/payson-utah-temple-celestial-room-1458545.jpg" alt="Payson Utah Temple Celestial Room" width="200">**

**</aside>**

**<footer>**

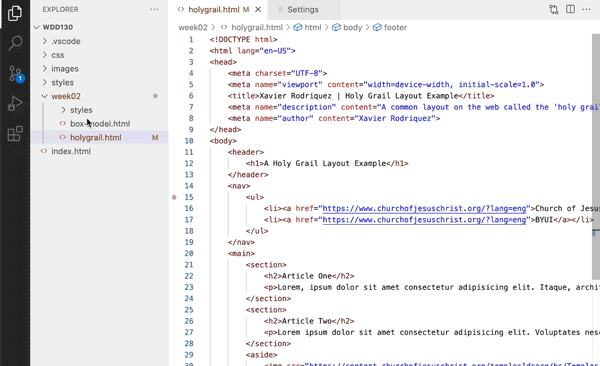
**🌴 Xavier Rodriquez**

**</footer>**

**</body>**

**Submission**

1. Remember to always **Save** your work
2. **Commit** and **Push/Sync** your work to your **wdd130** GitHub Pages remote site.
   * In VS Code, click on the **Source Control** icon.
   * Type a commit **Message** in the message input field provided.  
     This is for your own reference and is required.
   * Commit the changes by clicking the blue **✔ Commit** button.
   * Sync your committed changes to your local **wdd130** GitHub repository by clicking the **Sync Changes 1 ↑** button.

Commit and Push to GitHub from VSCode

1. Test your work by running this [Page Evaluation](https://byui-cse.github.io/wdd130-ww-course/grader/w02-holygrail.html) tool.
2. Share your work with class in Microsoft Teams by posting your URL to this html file:

https://yourgithubusername.github.io/wdd130/week02/holygrail.html

**Optional Resources**

* [HTML Content Categories](https://developer.mozilla.org/en-US/docs/Web/HTML/Content_categories) - MDN