

HTML Introduction

Before we begin, there will be assigned roles for this assignment. Each of the roles will be responsible for helping the group get through the document

Roles

Reference Guide: Role specific for person who knows HTML*: When the group is stuck on a question, your job is to guide them towards the answer without outright telling them.

Manager: Keeps the team on task, makes sure that the conversation doesn't go too off topic

Quality Control: Makes sure there is an accurate record of correct/final answer for the team and that all other members in the group understand the right answer

Timekeeper: Makes sure that the activities are being done in a timely manner and vocalizes to the instructor if more time is needed

Speaker: Reads through the packet out loud for the group so everyone can follow along. Communicates with the facilitator (in this case Sean) in case the group has a question or is confused about a part of the assignment

Reference Guide:

Manager:

Quality Control:

Timekeeper:

Speaker:

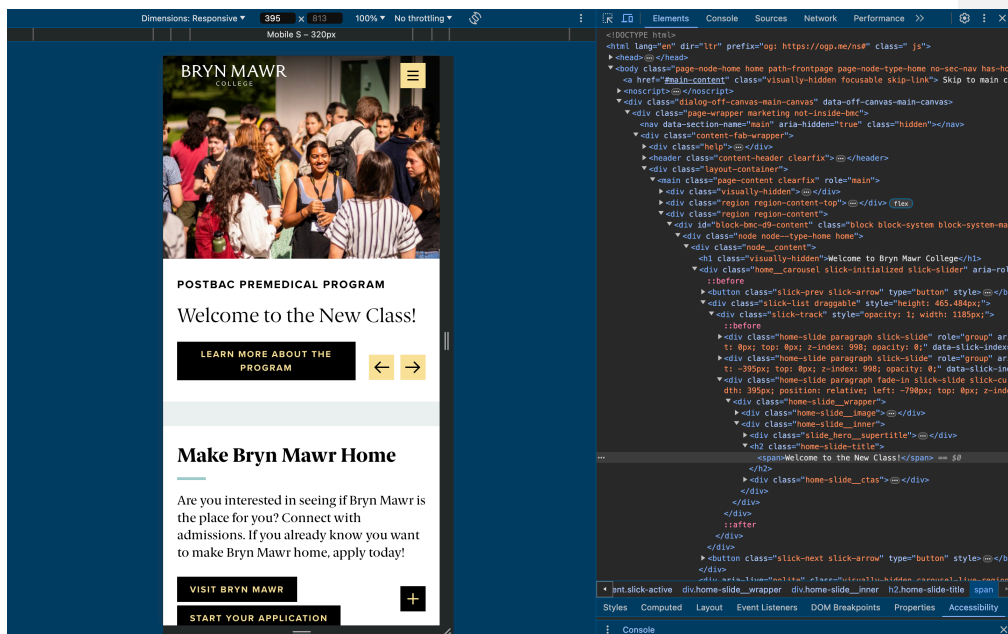
Now to begin the assignment....

What is HTML?

HTML = HyperText Markup Language

- HTML creates the structure of a web page and tells the browser how to display the content

Visit brynmawr.edu on your browser, right click on your trackpad and choose “inspect” – all of that code is HTML!...very, very complicated HTML



Rather than trying to actually read this, let's start with the basics

Syntax of HTML

HTML is made up a series of **elements**, which can be made up of **opening and closing tags**, **content**, and **attributes**

Great! OK but what does that mean?

Take an opportunity to fill out what you THINK the syntax should look like – talk through this as a group and come to a decision of what the syntax looks like (and why you think so). Feel free to use the white board.

Write out a line of HTML that says “My cat is grumpy” using these 7 pieces

<	>	>	
p	p	/	>

___ ___ ___ My cat is grumpy ___ ___ ___

What does the formatting of this is tell you? What do you think “p” stands for?

Now that you had a chance to guess the syntax of this, take the tape off this section and label the parts of this HTML

Commented [SK1]: TAPE

<p> My cat is grumpy </p>

What part of this do you think is the **opening tag**?

What part of this do you think is the **closing tag**?

What part of this do you think is the **content**?

***Note: **attributes** are also a part of an element, but we will go over what they are later

Altogether, all of these parts are called an **element**

HTML elements labels the content by using tags to describe whether an item is a paragraph, a link, an image, ect

The element above is an example of a **paragraph element** – it's one way we label text in an HTML document.

As individuals (so not as a group), take a minute to imagine what the element would look like for...

An image?

A heading?

A link?

(Don't worry about being right, we will get to these!)

Paragraphs, Headings, and Images


Now we have an understanding of what an element is and the syntax of HTML, we can learn about some other types of often used HTML elements

Paragraphs

The paragraph tag `<p>` is used to define the text within the document. It is the most often used tag in most HTML documents.

While you can directly type text into an HTML document and it will appear in your document, we have to remember that by adding the **opening and closing tag** to encase the **content** it makes it readable to the computer

Example

`<p> The DSSF Fellows are super cool </p>` →  That makes sense to me!

The DSSF Fellows are super cool → What is this gibberish?

Headings

Headings are used in a website in order to create organized sections that make the website easy to navigate

Think about if you were reading a scientific article – usually there is the title of the article, chapters, and subsections. This allows you to easily find the information you're looking for.

In this case it would look like this

```
<h1> Title of your article </h1>
<h2>Each of your chapters </h2>
<h3> Each of your subsections </h3>
```

You can go all the way down to `<h6></h6>` depending on how many subsections your HTML document has

Similarly to how an article only has one title, your HTML document can only ever have one `<h1></h1>` element, but the amount of `<h2></h2>` through `<h6></h6>` elements can be unlimited

Label the web page below with the correct headings depending on what is needed (hint – using the spacing to help determine between sections and subsections)

```
<____> My website <____>

  <____> About Me <____>

    <____> My Life Story <____>

  <____> My Work Experience <____>

    <____> My Resume <____>

      <____> My Personal Statement <____>

    <____> My Portfolio <____>

  <____> Contact Me <____>
```

Images

With the introduction of images, we are also going to get introduced to two new concepts: **attributes** and **empty elements**

Below is an example of an **image element**

```

```

Let's break this down!

An image element by itself looks like this ``

But why does it only have an opening tag and not a closing tag? Let's compare.... What's the differences between an image element and a paragraph element?

```

```

And

```
<p> Hello! I am a paragraph element </p>
```

The big things you may have noticed is that the **image element** doesn't have **content**, and that it has **attributes**.

The structure of an attribute is typically <element attribute-name ="feature of said attribute" >

So in the example above our attributes are:

src= "https://upload.wikimedia.org/wikipedia/commons/1/15/Cat_August_2010-4.jpg "

and

alt = "an image of a cat from Wikimedia"

Attributes contain extra information about the element that won't appear in the content.

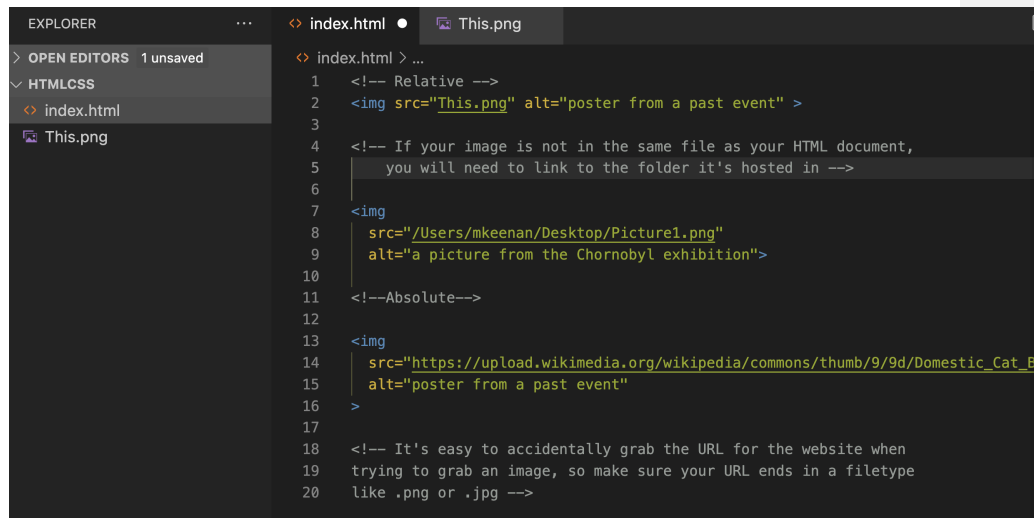
The only factor determining if it is an empty element is tied to if it has content or not, however empty elements often are only useable if they have **attributes**

You'll notice that the above image element has two attributes: **src** and **alt**

Src stands for source. You're actually unable to input an image into your HTML document, but you can link to its source. There is two different ways of creating a link – **Relative or Absolute**

Relative Link – it's for files that are hosted locally on your computer. The easiest way of creating a link to an image would be to input the image into the same file as your HTML document

Absolute Address – it's for files that are hosted on another server (such as a website) and requires the specification of the full, absolute URL

A screenshot of a code editor with a dark theme. The Explorer panel on the left shows a file named 'index.html' and a folder named 'This.png'. The main editor area shows the content of 'index.html'. The code includes comments and two tags. The first tag is a relative link to 'This.png'. The second tag is an absolute link to a file on a user's desktop. The third tag is an absolute link to a file on a website. The code is as follows:

```
<!-- Relative -->


<!-- If your image is not in the same file as your HTML document,
you will need to link to the folder it's hosted in -->



<!--Absolute-->



<!-- It's easy to accidentally grab the URL for the website when
trying to grab an image, so make sure your URL ends in a filetype
like .png or .jpg -->
```

Alt stands for Alternative text – it's used to describe the image when it is unavailable (say, if the image gets taken down on the original website) and is used for those who use screen readers to understand the content of the photo

<!-- Comments -->

Included in the image above is the example of some comment elements

They allow you to put information into your HTML document without the code editor thinking that it's useable

They're useful if you want to leave comments for yourselves or others in your HTML document

Challenge:

Now that you have an understanding of certain elements, each of you should individually create an HTML document in VSCode that includes three different types of heading elements, two paragraph elements, and two types of image elements (one Relative and one Absolute)

HTML Document Syntax

Amazing! Now you have a bunch of great content but we're not quite done – we currently have a bunch of elements free floating in the document. We have to structure the code so your code editor is able to read your code properly

<DOCTYPE! HTML>

At the top of your HTML document, you're going to put <DOCTYPE! HTML>

This is because we will be working with the fifth version of html (HTML5) – this tells the html document we're working with that we're using this version of HTML and not a past version of HTML

<html></html>

Below that we're going to include the opening tag for <html>

The closing tag for this will go at the very end of the document

This is because for now we want everything in the document to be encompassed by this <html></html> element. This makes it so your computer knows where the document starts and ends.

`<head></head>`

Inside of the html element, we're going to include the head element (this should go above all the elements you produced in the last challenge)

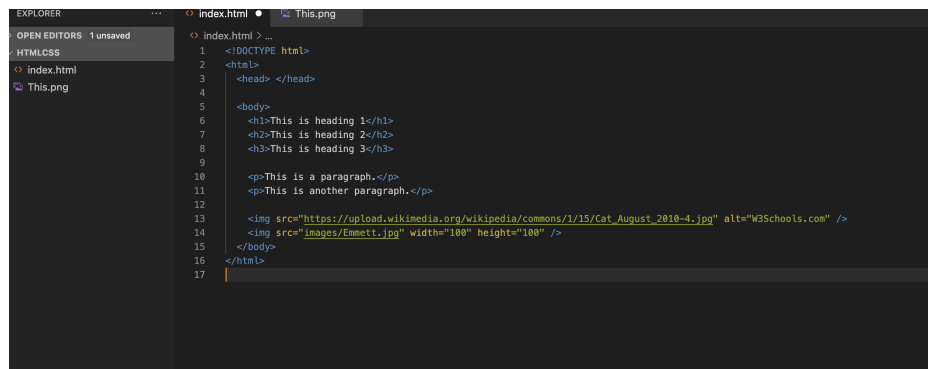
The head is where all your elements that aren't going to appear on the web page will go. Most of the content you will include here will include information about your HTML page, your title, and links to your stylesheets (which we will go over during the CSS session)

`<body></body>`

Finally, we have the body element, which will engulf all the content you produced from the last challenge (headings, paragraphs, and images)

This section is where you want to include all the content you want to appear directly in your web page.

Your document should look similar to this when you're finished

A screenshot of a code editor with a dark theme. The Explorer sidebar on the left shows a file named 'index.html' selected. The main editor area displays the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3   <head> </head>
4
5   <body>
6     <h1>This is heading 1</h1>
7     <h2>This is heading 2</h2>
8     <h3>This is heading 3</h3>
9
10    <p>This is a paragraph.</p>
11    <p>This is another paragraph.</p>
12
13    
14    
15  </body>
16 </html>
17
```

Head elements

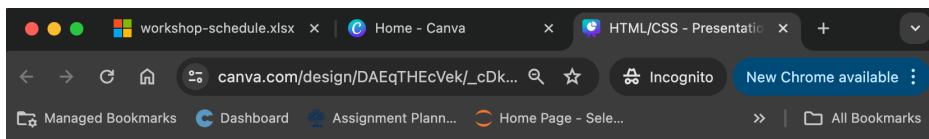
Now that we've covered how to create a very basic HTML document, let's start building up our document

`<title></title>`

The title element gives a title to your document. This title is important since all documents put online are required to have a title.

The title is shown in the browser's title bar or in the page's tab when you're surfing the web

Circle the examples of different website titles below



`<meta>`

Meta is an element that is short for “metadata” – this is also another example of an empty element since all the different types of information you put in the meta element are defined by different attributes

The information put in the meta elements is often used by the web to understand the contents of your website

Below are different examples of meta elements

`<meta charset = “UTF-8” >`

`<meta name = “keywords” content = “HTML, CSS”`

`<meta name = “author” content = “Sean Keenan”`

`<meta name = “viewport” content = “width = device-width, initial scale = 1.0” >`

Try to interpret as a group a meaning of these meta elements (you can google the meaning of UTF-8). What would be another example as well of information you think would fit the meta element?

Body elements part 1

As we've discussed, the content you've included in your site (images, headings, paragraph elements) all belong in the body of your document since they appear on your website

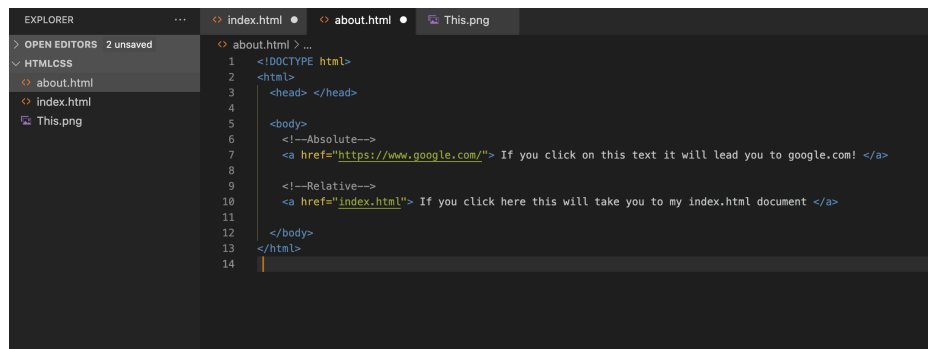
There are lots of other HTML elements that will be very useful as we look to build up our document

`<a>`

The anchor element is used with hyperlinks – similar to with image elements there are **relative** and **absolute** links

The attribute used in order to attach the links is href (not src!)

Unlike with images, anchor elements are regular so they include content



```
1 <!DOCTYPE html>
2 <html>
3   <head> </head>
4
5   <body>
6     <!--Absolute-->
7     <a href="https://www.google.com/"> If you click on this text it will lead you to google.com! </a>
8
9     <!--Relative-->
10    <a href="index.html"> If you click here this will take you to my index.html document </a>
11
12  </body>
13 </html>
14
```

Give an example below of how you think you could use an image as a link (feel free to play around for an answer in your HTML document)

Document Object Models

One thing you may have noticed once you got the answer for the above question is that it is possible to wrap elements inside of other elements.

You may have just noticed it but you've already been doing this inside your HTML document.

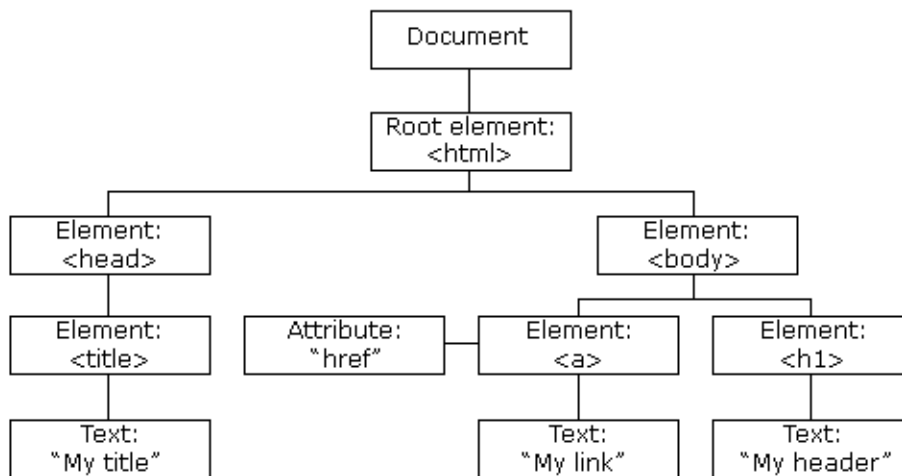
For example: Your paragraph elements are enwrapped inside of your body element. Your title element is enwrapped inside of your head element. Both your body and head element are enwrapped inside your html element.

This relationship (between elements are that are enwrapped and those doing the wrapping) are known as **parent** and **child** relationships

Parents = wrapping

Children = being enwrapped

Sibling = both being enwrapped by the same parent



Discuss as a group and label the above with two examples of parent and child relationships, and one example of a sibling relationship

This knowledge will come in handy later once we cover CSS

Body elements part 2

`` and ``

Speaking of parent and child relationships, another element that needs to utilize this is the **ordered** and **unordered** lists

The only difference is that default ordered lists use numbers and unordered lists use bullets

To create either of these, inside of the `` and `` elements need to be **list items** ``

The list elements are the **children** of the unordered/ordered list elements

In your HTML document, take a moment to individually create an example grocery list with at least three items on it. Compare with the rest of the group.

Style attributes

Finally for our HTML worksheet, we will talk about the style attribute

This is a way for us to stylize our HTML without having to use CSS (which is our next session)

This only works on a single element, and later on, we will learn that by using the style attribute, this will trump any changes made in the CSS document. It's a way to make sure certain elements follow certain style rules.

Some examples would be

`<h1 style = "color:blue"> Now my header is blue! </h1>`

`<p style = "text-align:center"> Now my text is centered rather than left aligned </p>`

Try taking an element in your HTML document (header, paragraph, link) and use the style element to turn the "background-color" orange

