

1. What is HTML?

HTML (HyperText Markup Language) is a coding language that can be used to generate content like web pages. It is one of the simplest coding languages, and is often used as the foundation of web content, with other languages used to enhance its capabilities. HTML can also often be used by end users to customize content like blogs, forum posts, and certain social media pages.

Take care to remember that HTML is a language. It's in the name, HyperText Markup **Language**. We learn languages as we are exposed to and use them, so do not be discouraged if you struggle at first.

2. Getting started writing HTML

It is possible to write HTML, or any code for that matter, in any word processor program, but we want to avoid using programs like Microsoft Word to do so. This is because programs like Word apply formatting that can change and augment the code you are writing, which it often will because you won't be following conventional grammatical rules. When you code you must be precise. A single comma out of place can cause all other functions to fail, so you want to avoid a scenario where a program may change a character without you noticing.

Using software like **Notepad** or **Wordpad** will allow you to type your code without worrying about formatting. These programs come in the accessories suite in Windows. There also exist third party software like **Notepad++** and **Sublime Text** (my preferred software at the moment). These are free (as of this writing) and come with features to make writing code easier, like the ability to look up tags and visual indicators of problematic code.

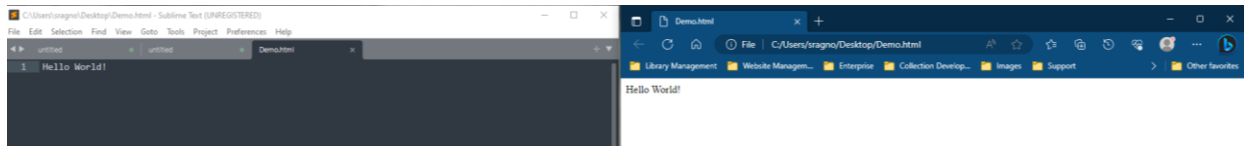
Sometimes you will be able to type HTML in text boxes on websites. Often these text boxes will allow you to toggle between "visual" and "text", with text being intended for code, allowing your HTML to behave as intended, and visual being formatted similarly to a word processor like Microsoft Word.

3. Viewing Your Work as a Web Page

In simple terms, when we access a web page through a browser we are connecting to a series of files on a different computer, usually a "server", connected to a network and permitting other users connected to the same network to access a portion of its files. This is fairly complicated and something we won't be going into here, but we can simulate this process locally (isolated to your own computer).

When begin working on your code in your chosen text editor save your file with the .html extension. You do not need to have any content before you do this, though the generated page

will be blank. Once you've done that you should be able to navigate to the file, right click it, and open it with a browser.



Sublime Text on the left, browser preview on the right

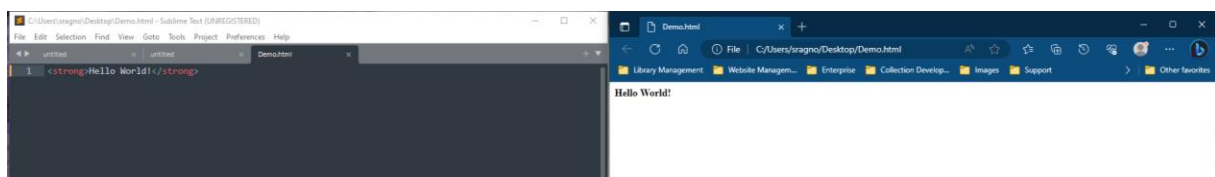
It is wise to have a preview of your website available as you work like this. When you enter new code in your text editor you can save the file and then refresh the browser to see the changes take effect. This can help you spot errors early, before mistakes get buried in other code and it becomes difficult to isolate the problem.

It is also good practice to work on content where it is inaccessible to general users rather than when it is live and in use. This is one way of doing that.

Bear in mind that the context of addresses (locations) on your local computer will be different for visitors from a network. For instance, if I have a picture of a flower with the address `C:\Pictures\Flower.PNG` on my personal computer and give that address to a network user to retrieve the picture in question the request would fail because it would look at `C:\Pictures\Flower.PNG` on that user's computer, not mine where that picture is located.

4. Tags and Elements

You can think of HTML as a set of instructions to a browser informing it how to render a page. These instructions take the form of **Tags**. A tag is a command surrounded by brackets `< >`. There are a finite set of tags, and I've included some of the more useful ones in the appendix of this guide. Tags always come with an open tag `<TAG>` and most come with a close tag `</TAG>`. The content surrounded by open and closed tags is called an element. A tag will augment the element it surrounds.



For instance, the tag `` can be used to bold an element

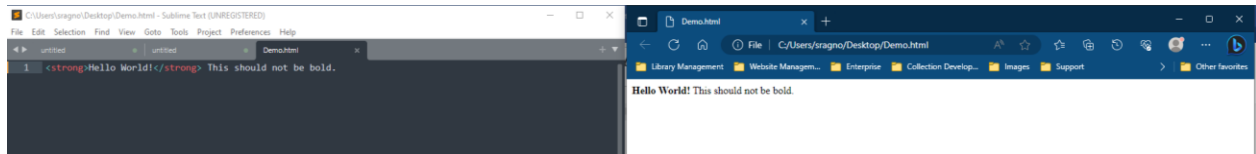
It may be helpful to think of these tags as instructions you may give to a person making the page for you. In the example above, `Hello World!` the instruction

Introduction to HTML

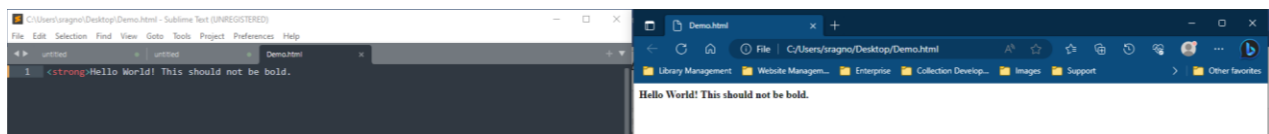
Page | 3

would be “Start bolding. Show ‘Hello World!’. Stop bolding.” Your opening tag tells the browser when to start a command, and the closing tag tells the browser when to stop the command.

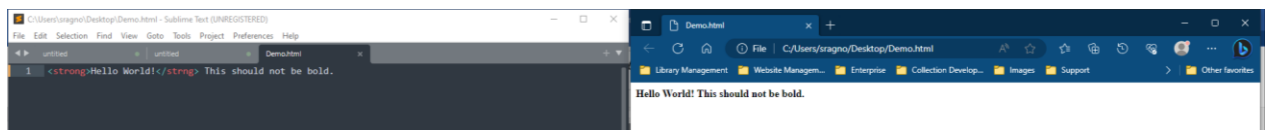
As you may have guessed from that description, omitting a close tag will cause a tag to apply to all following elements. The same will happen if you enter a close tag incorrectly.



This example is displaying correctly

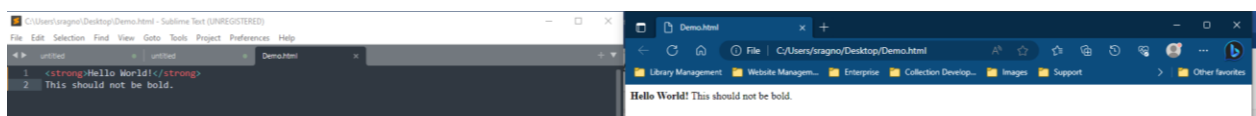


We forgot the close tag here, so the entire page is bold.

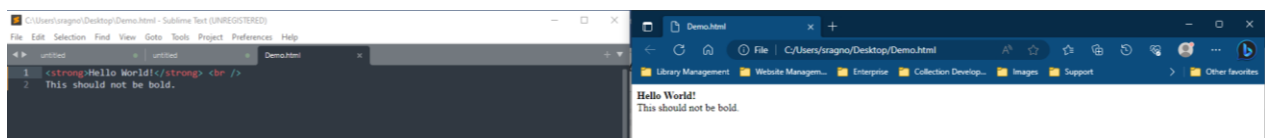


Here there is a typo in the close tag, leading to the same result.

Some tags do not require a close tag because their command does not apply to an element. For instance, the `
` command, or Break, moves content down to the next line.



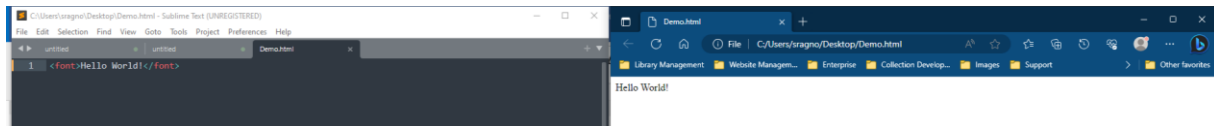
*Moving down to a new line in HTML code is not recognized by a browser (some text editors automatically add code to address this, like in **Wordpress**), it needs instructions.*



The break tag does not need a closing tag because it does not augment anything, it stands alone. Our instructions here are “Begin bolding. Show ‘Hello World!’. Stop bolding. Go to the next line. Show ‘This should not be bold.’”

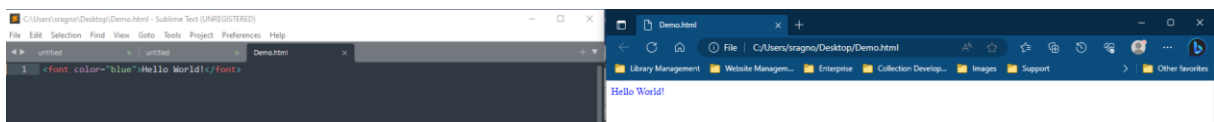
5. Attributes

Many tags can accommodate or require further instructions, known as **attributes**. Take for instance, the tag ``, which instructs a browser to change the font of an element. If I were to instruct you `Hello World!` (“Change the font. Show ‘Hello World!’ Stop changing the font”) you’d likely ask me “Change it to what?”



No change.

Attributes go inside an opening tag, and like tags there are a finite set of tags, though unlike tags they need to be paired with a value. A tag (that closes) complete with attribute will look like this: `<TAG ATTRIBUTE="VALUE">ELEMENT</TAG>`.

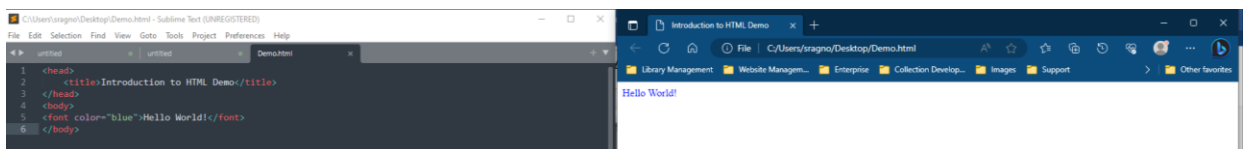


6. Header and Body

There are two regions you want to consider when building your HTML document: the **head** and the **body**.

The head region is tags that define the nature of the page, like metadata (information about the page). The head region is also where more advanced code can be referenced, but we will not be exploring that here.

A good example of a tag that lives in the head region is `<TITLE>`. You may have noticed that the browser tab in my examples is labeled “Demo.html”. The browser is looking for a title for the web page, not finding one, and pulling a title from the file name. Often this will be undesirable, so you can use the title tag to define one.



Note that the page title is displaying in the tab, and not the web page itself. This is the difference between the head and body region. The body will contain the elements that display on your web page.

7. Essential Tags

Now that we have the basics down we can explore tags in more detail. Learning HTML from this point on is really a matter of familiarizing yourself with different tags and attributes. Here are a few that are crucial in almost any web page.

Hyperlinks

A hyperlink allows an element to redirect a browser to another page when it has been clicked. Hyperlinks are the primary way users navigate to other pages, both internal and external. Hyperlinks use the anchor tag `<A>` paired with the source attribute `SRC="VALUE"` where the value is the URL of the desired object. So, for instance, a hyperlink to the library website would be

```
<a href="https://www.hegganlibrary.org/">click here!</a>
```

When a user clicks "click here" they would be directed to the library website.

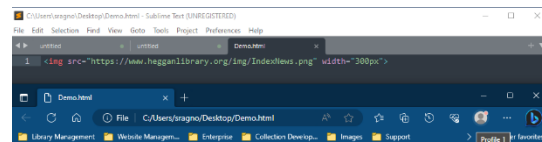
Images

Use the image tag `` with the source attribute to insert an image into your page. Since this is not augmenting an element, it does not need a close tag. Once again the source value is a URL. Any file online will have a URL. In order to obtain an image's URL you can right click an image in most browsers and click "copy image link" or failing that, Inspect to view the page code. For instance, the library has a logo image at <https://www.hegganlibrary.org/img/IndexNews.png> so to insert the library logo into a page we would use

```

```

Often you'll want to adjust the size of an image, and this can be done with some additional attributes, `WIDTH="VALUE"` and/or `HEIGHT="VALUE"`. If you just specify one value the other will adjust to maintain the aspect ratio of the image, so often people just use the width attribute. The value can be in pixels (300px) or a ratio (30%). Keep in mind that browsers come in different shapes and sizes.



Combining Tags

It is possible to nest tags in other tags to compound functions. For instance:

```
<a href="https://www.hegganlibrary.org/"></a>
```

This will produce an image of the library logo that when clicked will redirect a user to the library website.

COLORS

HTML is able to recognize a number of colors by name, such as blue, red, green and so on. However, it is worth being aware of hexadecimal color codes in order to actualize palate that you have in mind. A hexadecimal value is a series of 6 numbers and letters ranging from 0-9 and A-F (#037acf). In colors these values reflect RGB (red, green, blue) and HSL (hue, saturation, lightness). They exponentially expand your color options. Use a color picker tool, such as the one at https://www.w3schools.com/colors/colors_picker.asp to explore your options.

8. Going Further

As I said at the opening of this guide, HTML is a language and you'll learn it by using it and being exposed to it. One handy way to learn code is by reading the code on websites that you like. To do this, visit a site and right click. In most browsers you'll have the options of **View Source** and **Inspect**. View Source will display the source document for a page. Inspect will display a side by side view of the page and the source code, allowing you to hover over tags and see what objects on the page for which they are responsible.

Another resource is to read up on HTML. Many people learn the language just by grabbing a book and studying it for a summer. The library has a number of books on HTML and other coding languages. Another handy resource is the website <https://www.w3schools.com/>. This website contains write ups on many tags and attributes and allows you to try applying them in constructed exercises.

APPENDIX: USEFUL TAGS

`<HTML> </HTML>` Designates the document as HTML code. This should be the first and last line in your code.

`<TITLE></TITLE>` Designates a title for the page.

`<BODY BGCOLOR="X"></BODY>` Modifies the background color of your webpage.

`<BODY BACKGROUND="URL"></BODY>` Use an image as the background to your webpage.

`
` Adds a line break.

`` Establishes a Hyperlink

`TARGET="_BLANK"` Opens the link in a new window

`` Modifies font when used with the following attributes.

`COLOR="X"` Changes the color.

`FACE="X"` Changes the style of font.

`SIZE="X"` Changes the size of the font (Default size is 3).

`` OR ` ` Bold.

`<I></I>` OR ` ` Italics.

`<U></U>` Underline.

`` Embeds an image, modified with the following attributes.

`WIDTH="X"` Defines the width of the image in pixels.

`HEIGHT="X"` Defines the height of the image in pixels.

`BORDER="X"` Defines the border of the image. Set to "0" to eliminate border.

`<H1></H1>` Headline, very large text. `<H2>` through `<H6>` work too, make smaller headlines.

`<DIV></DIV>` Defines styles and functions for a section, or "block" of a page. This has many functions, and is particularly powerful when paired with CSS

`ALIGN="X"` Aligns section to the left, right or center

`<P></P>` Defines a paragraph.

`ALIGN="CENTER"` Aligns paragraph to the center (left and right also work).

`<HR>` Horizontal Rule, creates a bar to divide segments of the page.

`COLOR="X"` Defines color of bar.

`SIZE="X"` Defines size of bar in pixels.

`<TABLE></TABLE>` Establishes a table

`WIDTH="X" HEIGHT="X"` Establishes size of table

`BGCOLOR="X"` Establishes background color for entire table

`ALIGN="X"` Aligns contents horizontally (LEFT, RIGHT, CENTER)

`VALIGN="X"` Aligns contents vertically (TOP, BOTTOM, MIDDLE)

`<TR>` Defines a row

`<TH>` Defines a header cell

`<TD>` Defines a regular cell

`ROWSPAN="X"` cell spans X rows

`COLSPAN="X"` cell spans X columns

`<IFRAME SRC="X" HEIGHT="X" WIDTH="X"></IFRAME>`

Displays source within frame. This can be an image, video, or another website.

`NAME="X"` Assigns a name value to the element

`<!-- X -->` Comment tags. Anything between the two tags will not render on the website, allowing you to hide content or leave messages for yourself or other developers. This will show if a user views source.

`<STYLE></STYLE>` Allows you to enter CSS into an HTML document. CSS is written differently than HTML.

`STYLE="X:Y;"` This attribute allows you to apply CSS to single tags. CSS is written differently than HTML