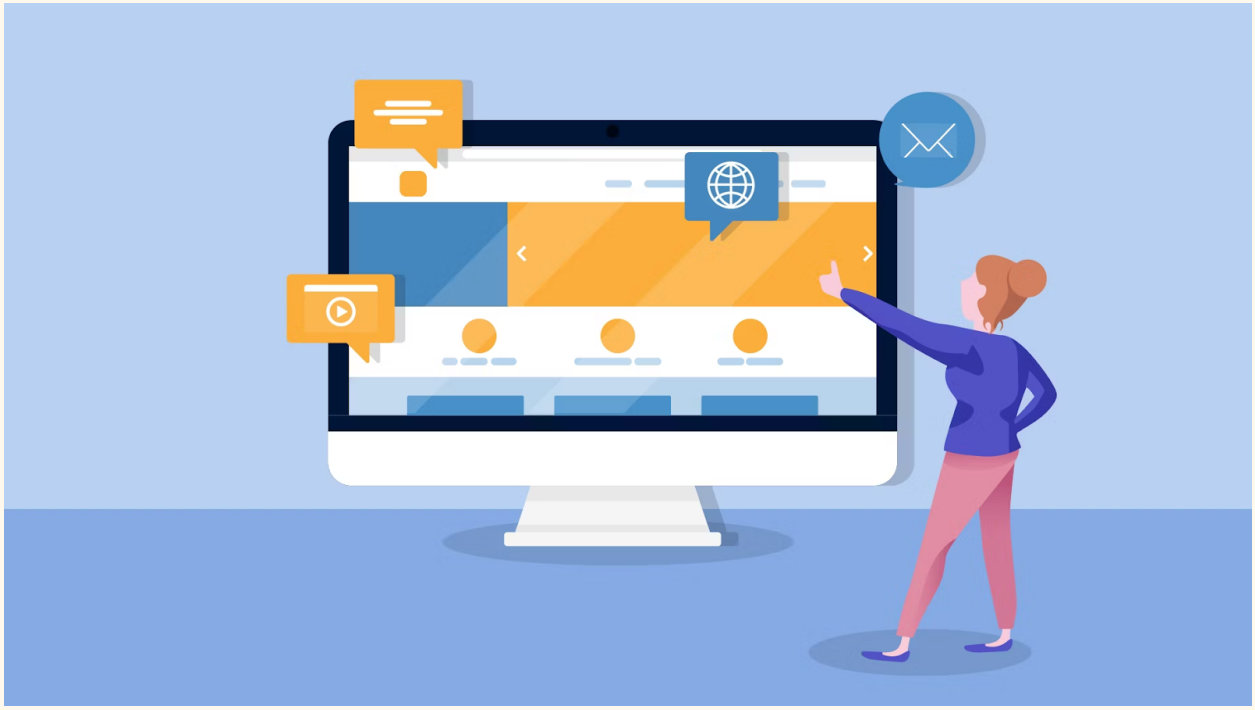
FINAL PROJECT JOURNAL
PEDRAM KABOLIAN

GOAL



My goal was to learn the basics of HTML and CSS as programming languages alongside coding applications to a proficiency where I can build a small website while also grasping the fundamentals well enough to be able to expand upon my skills in the future. During this project, I used Codecademy to learn HTML and CSS. Unfortunately, as I was using the free version, I was not given any certification for completing the lessons. However, I have included the small exercises I did while learning the languages to demonstrate my learning and growth.

Table of Contents

FINAL PROJECT JOURNAL
PEDRAM KABOLIAN

HTML

What is HTML?

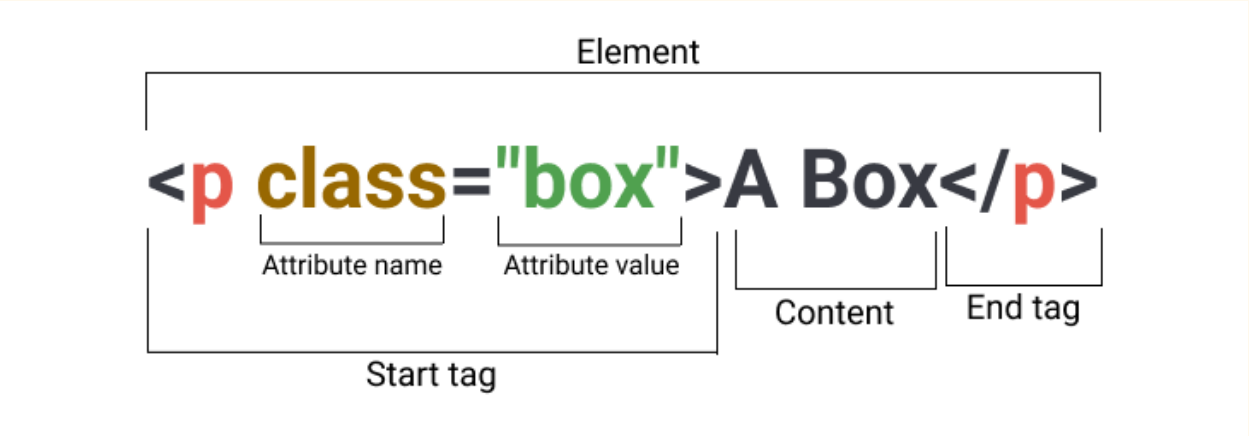
HTML stands for HyperText Markup Language. It is a language used to structure web pages and their contents. HTML was invented by Tim Berners-Lee in 1993, and has since evolved into HTML 5. HTML was based off of SGML; another markup language, and is technically not a programming language. HTML lacks many features that define programming languages; such as variables and loops, as such, it is rather dissimilar to Java and C++. Both HTML and Java/C++ follow code as a procedure, meaning they run through code from top to bottom, but the similarities end there.

FINAL PROJECT JOURNAL
PEDRAM KABOLIAN

MAY 25, 2023

The Basics

___



Elements:

HTML is composed of elements. Everything exists within an element with the exception of the <html> element itself. HTML is a language that relies on nesting: elements are placed within elements within elements. An element with an element inside it is called a parent, grandparent, etc. and an element within another is a child, grandchild, etc. Elements do everything in html. They are the building blocks of the website format.

Attributes:

Attributes are the next major part of HTML. If elements are totems on a totem pole, attributes are the etchings. Attributes can be used to assign classes and IDs, that can be used for styling and organization. They can also be used to format elements, and determine their nature. The dimensions of an image, the controls on a video, and what happens when the user clicks on a link are all determined by the attributes on their respective elements.

Element tags and attributes are the basics of HTML. It's worth it to collect and memorize these to expand your arsenal and formatting abilities. When you're confused about a tag or an attribute, search it up; there are tags and attributes for everything.

FINAL PROJECT JOURNAL
PEDRAM KABOLIAN

MAY 26, 2023

Differences and Similarities

—

Differences:

Elements in HTML are enclosed in triangular brackets < >, as opposed to Java and C++, which typically use parentheses and squiggly brackets { }.

HTML and Java/C++ are fundamentally different as they serve different purposes. HTML is a markup language, used to format web pages, while Java/C++ are programming languages, used to create programs and processes.

Similarities:

All elements in HTML must be closed with closing tags, similar to semicolons in Java/C++.

Code in both HTML and Java/C++ is read from top to bottom, in the order it was written.

Attributes are equivalent to variables in Java/C++. They are assigned values and called on.

HTML greatly relies on importing external files, like Java/C++ with imports.

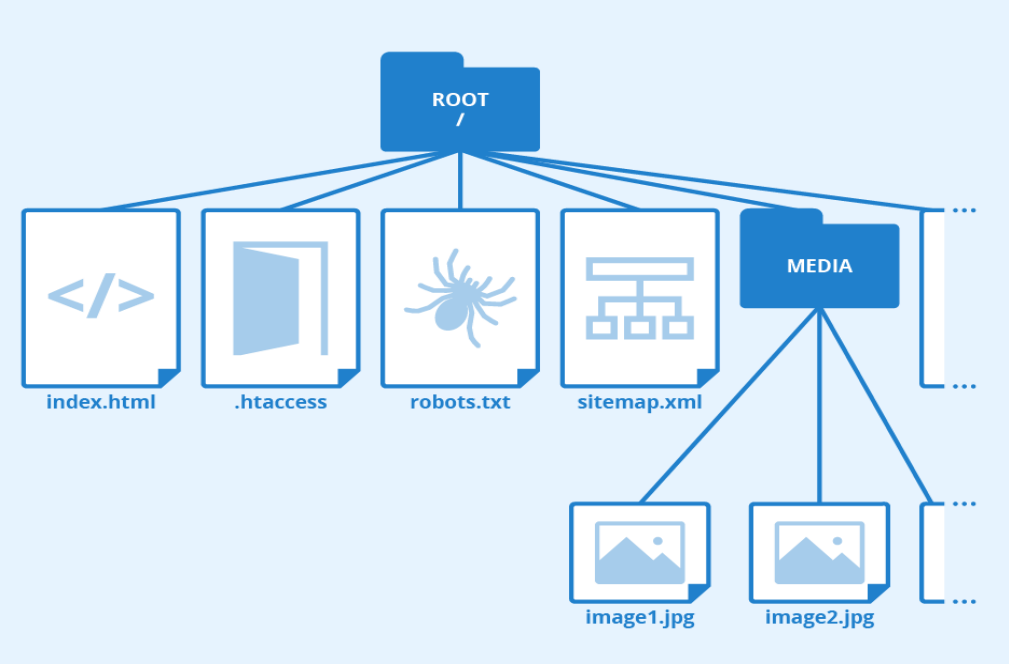
HTML code is nested in parent elements, similar to methods in Java/C++.

FINAL PROJECT JOURNAL
PEDRAM KABOLIAN

MAY 26, 2023

Links and Root Directory

—



Displaying

Displaying is an integral part of web design. Any form of media that is shown on a page is displayed. Media can be displayed in two ways: links, or directory destinations. Links are URLs that lead to a website that has the media uploaded. Directory destinations are directions to files located within the same root directory as the HTML file.

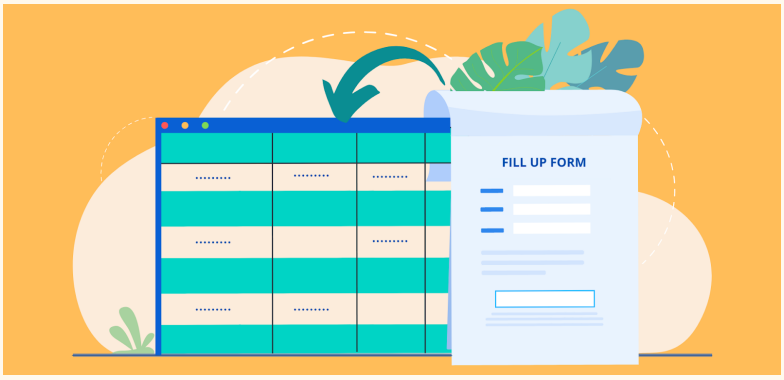
Navigation

Navigation can be used to change to different pages within the same site. This is typically done using directory destinations. It can also be used to switch websites entirely, which is done using links. It is important to properly manage links and directory destinations, because the entire site may cease to function properly if even one link or directory destination is slightly incorrect.

FINAL PROJECT JOURNAL
PEDRAM KABOLIAN

MAY 30, 2023

Tables and Inputs



Forms:

Computers communicate with each other via HTTP requests. HTTP stands for Hypertext Transfer Protocol. The modern version is HTTPS, the S being Secure. Forms can be created using the <form> tag, and inputs can be created by nesting an <input> tag inside it. All user inputs that are received by the website require a form. Inputs can take many forms, from usernames to passwords to sliders to numbers. Inputs can be regulated by using pattern recognition, which requires the input to match a certain format, by limiting the input forms, or by adding minimum and maximum values. Input fields can also be made required. A special submit input can be made for the user to submit the information they entered. Once sent, information takes the form of variables and their corresponding values. ie. name='Pedram'.

Tables and Lists:

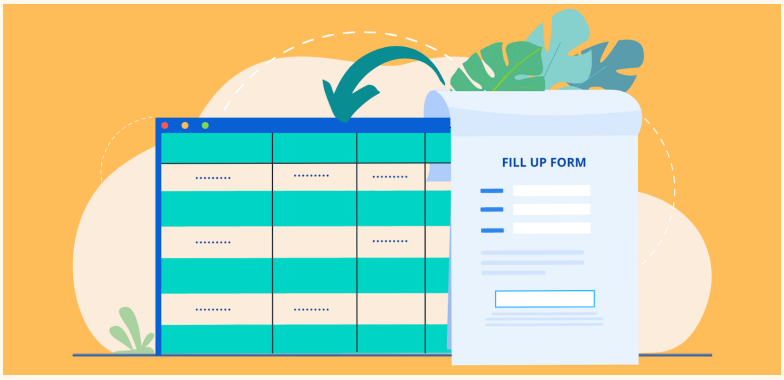
Tables in HTML function differently from other tables. Every row in an HTML table must be manually added through a data cell or title cell element. Cells can also be made to span multiple rows or columns. There are two main types of lists in HTML: ordered and unordered. Unordered lists are in no particular order, while ordered lists are numbered.

FINAL PROJECT JOURNAL
PEDRAM KABOLIAN

MAY 30, 2023

Commenting, Navigation, and Accessibility

___



Commenting:

Comments in HTML function similar to the */ commenting mechanic in Java. Begin comments with <!-- and end them with -->.

On-Page Navigation:

Navigation can be accomplished using the <nav> element. Use an ID to have the web page automatically navigate to the part of the page that holds the ID. This is useful for large web pages.

Accessibility

Pages can be made more accessible with alt text that can be read by screen readers. When inserting an image, add alt text just in case the image doesn't load or if the user is visually impaired. The same principle applies to most other media embedded in the page.

FINAL PROJECT JOURNAL
PEDRAM KABOLIAN

MAY 31, 2023

Convention

Use double spaces or the indent key when nesting elements, similar to Java/C++.

Use semantic HTML rather than non-semantic. Non-semantic is when everything is nested in or elements, while semantic nests elements in semantic elements like <header> and <footer>. These elements don't actually add anything, but make code more readable and help with styling and organization later on.

Use descriptive class and ID attributes. This makes code more understandable and makes writing complicated pages more manageable.

Keep the styling to CSS. This is much more efficient and keeps code readable.

Keep everything lower case unless necessary. This is similar to Java/C++ convention.

Each individual page should have its own separate HTML file.

Keep styling off HTML. leave all styling to CSS. The most styling that should be done in the actual HTML file is text color/font.

Use comments in complex code to make code understandable, like in Java/C++.

CSS

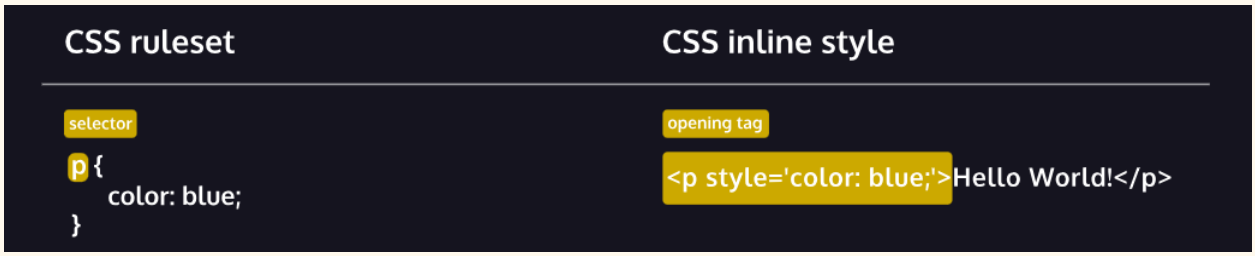
What is CSS?

CSS stands for Cascading Style Sheets. HTML code by itself doesn't actually display anything, it just dictates the format of the page. It's up to the default style sheet of the browser the HTML file has been uploaded to to actually display the page. This default style, however, is very simple and quite ugly. Fortunately, HTML has a sister language: CSS, that can style web pages. CSS shares a few more similarities with Java and C++ than HTML, but it is still a completely different type of language. As with Java and C++, CSS code is often held within curly brackets, and variables are assigned values within these brackets. HTML and CSS have extremely different appearances and syntaxes, despite being so closely related. CSS can do just about anything with a website, and is necessary to create a web page that is pleasing to look at.

June 1, 2023

Basics

___



Styles:

There are two types of CSS: Ruleset and Inline Style. Inline Style is CSS that is implanted directly into HTML using the style attribute within a <style> element. Ruleset can be used in HTML as well, but more importantly it can be used in CSS. Never use Inline Style. Ruleset is the standard.

Syntax:

CSS needs a selector. A selector determines which HTML elements the style will affect. The selector can be a tag, a class, or an ID; or even a combination. There is also a universal selector that selects all elements, the universal selector is asterisk *. When two styles conflict, the one that has a more specific selector will apply, like with methods in Java/C++.

Styles are made using properties and values, similar to variables and values in Java/C++. Properties can be anything from color to font to positioning. Values can also take many forms, but depend on the property they apply to.

Code in CSS is contained within squiggly brackets, similar to Java/C++.

All lines of code in CSS must be ended with a semicolon, like in Java/C++.

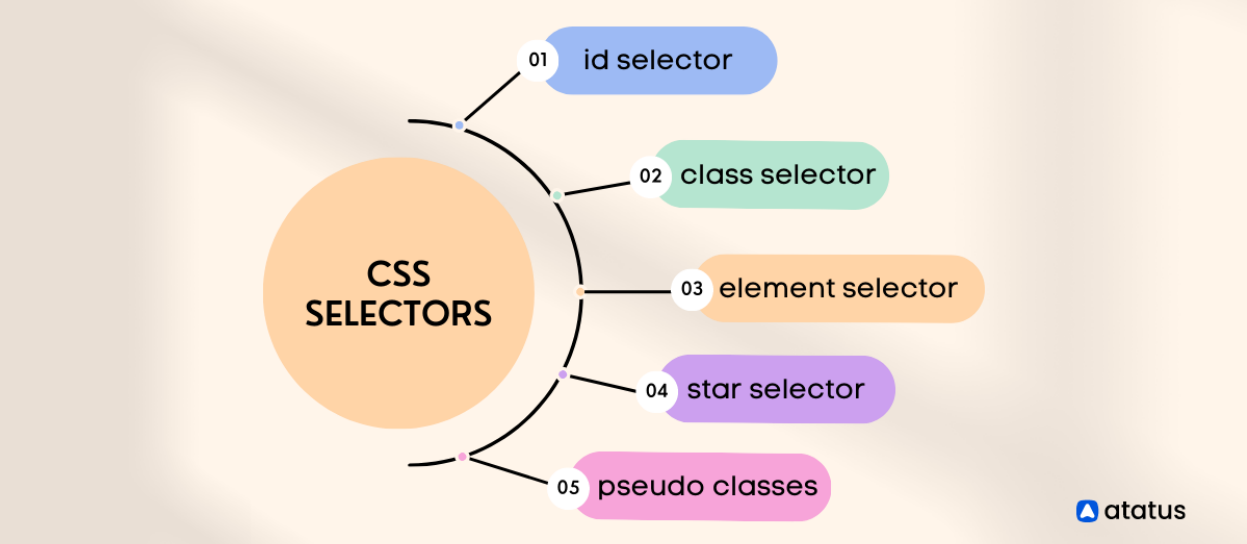
There is no nesting in CSS, and the order of blocks doesn’t matter.

FINAL PROJECT JOURNAL
PEDRAM KABOLIAN

June 1, 2023

Selectors

—



Selectors are what determine which elements will be affected by a style.

The 4 main types of selectors are: Universal Selector, Element Selector, Class Selector, ID Selector [ranked from least specific to most specific].

Class selectors must have a . at the beginning, and ID selectors must have a #.

When two styles are in conflict, the one whose style is more specific will apply.

Pseudo Classes can also be added to selectors. These will only affect elements when they're in a certain state. For example the hover pseudo class will only apply when the cursor is hovering over the element. Add a colon to the end of a selector to add a pseudo class.

Selectors can be combined by adding them one after the other. This increases specificity.

A block can apply to multiple selectors. Simply separate selectors with a comma.

FINAL PROJECT JOURNAL
PEDRAM KABOLIAN

June 1, 2023

Styling

___

Colors:

Colors can be changed using the color property. There are several predefined colors to choose from using simple names like ‘red’ and ‘blue’, but CSS also recognizes color hex codes, which should be used instead as they’re more specific.

Positioning:

There are 5 different types of positioning in CSS. Static, Fixed, Absolute, Sticky, and Relative. Static is the default nature of objects. Fixed and Absolute function similarly, where the object they’re applied to doesn’t move regardless of scrolling. Sticky elements function as though they are fixed until a certain point. Relative allows for movement of a static object in relation to something, typically the viewport. The viewport is the monitor that is being used to view the website, and relative positions are set by putting how far the element should be from the edges of the viewport.

FINAL PROJECT JOURNAL
PEDRAM KABOLIAN

June 1, 2023

Styling Continued

Fonts:

Fonts can be altered using the font-family property. Only fonts that are recognized by the browser should be used. Fonts may not do what their input names imply. For example, the ‘cursive’ font on google chrome sets the font as comic sans. Font weight can be changed using the font-weight property, and font type can be changed using font-type. Font size can be set using the font-size property.

Gradients:

Gradients can be applied by using the linear-gradient property and setting the values of the starting color, the end color, and the angle. Gradients look weird when the website exceeds the viewport dimensions of the viewing monitor.

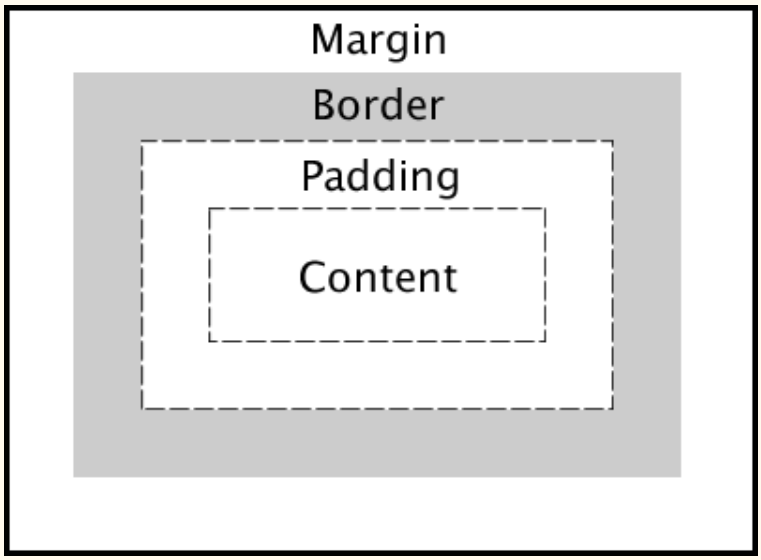
Text Organization:

Text can be aligned using the text-align property. All the standard text alignment features are present, including justify which evenly spaces text. Note that text boxes in HTML take up all horizontal space allotted to them, so text-align will move the text to the other side of the page.

FINAL PROJECT JOURNAL
PEDRAM KABOLIAN

June 5, 2023

Box Model and Page Formatting



CSS elements follow the box model, as seen above. The actual content of the element is surrounded by a border, with padding on the inside and a margin on the outside.

The box-sizing property is set to content box by default. This can be hard to work with so set it to border-box to get a more accurate idea of what the size of elements will be.

Padding and margin behave similarly when sizing. To size them, you can address individual sides with padding-top and padding bottom, or use shorthand to address all in one line. Shorthand for padding and margin sizes follows this format: when 4 size values are given, values are assigned to top, right, bottom, left. When two are given: top/bottom and right/left. 1 value is universal.

It is important to note that vertical margins do not combine. The larger of the two applies itself.

Sometimes elements can flow off the page. Use the overflow property to determine what should be done with these elements. Set it to hidden to hide it, collapse to add a scrollbar, and expand to zoom out the page.

FINAL PROJECT JOURNAL
PEDRAM KABOLIAN

June 5, 2023

Browser

___

Ordering:

When elements overlap each other, important information may be covered or the website may just look ugly. Using the z-index property, order important or unimportant elements. The higher the z-index, the higher up on the page it is.

Browsers:

Different user agents (nerd for browsers) will view pages in different ways. Fonts and named colors are up to the browser's discretion. To avoid this, use universal language, like hex codes for colors instead of names.

Viewport:

The viewport is the monitor that is showing the page. To make sure the website looks the same regardless of monitor sizes, do not use pixels to set element dimensions. Instead, use viewport width (vw) and viewport height (vh) to set dimensions. 1vw is 1% of the viewport width, 100vw is 100% of the viewport width.

Publishing

There are two different ways to publish a website. Free publishing and paid publishing.

Paid publishing is very easy and mainstreamed. Simply lease or purchase a domain name off GoDaddy or Google Sites, then use a file manager to add your files to the domain.

Free publishing is also easy, but there are less options. There are plenty of ways to publish a website, but they don't allow you to upload custom code. The only way to easily publish a custom website is through github. On github, create a repository, name it [username].github.io, add your files, then publish it.

After publishing, it may take a while for changes to be made visible.

The very first file that will be opened when the website is accessed is the file named 'index.html'. Make sure that one and only one file have this name.

HTML and CSS are very easily coded using notepad and notepad++. Notepad provides no assistance, and notepad++ is a very minimalist editor. The basic interface of notepad++ is very similar to eclipse and codeblocks, and is quite intuitive to use.

Evaluation

Overall, I enjoyed learning HTML and CSS. It felt very rewarding learning a coding language I could very easily apply in the real world. The base fundamentals and conventions were quite similar between HTML, CSS, Java, and C++; so I was able to make connections between my past learning and my new learning. The payoff of putting together my own website and publishing it was also very rewarding.

I feel that by now, at the end of my learning, I have a very good grasp on the fundamentals of HTML, CSS, and web design. I obviously have quite a ways to go before mastery, but I feel confident just knowing how the languages function, and I believe this will make my learning going forward much easier.

I was able to achieve most of my necessary goals, as well as quite a few of my wanted goals. However, there were a couple wanted goals I wasn't quite able to achieve. I wasn't able to learn how to use Atom because it got sunsetted by Microsoft and was no longer usable, but I did learn how to use notepad and notepad++. I believe the fact that many of my wanted goals were achieved while a few of my needed goals went unachieved was because of my lack of knowledge of the workings of HTML and CSS. For example, embedding videos is one of the easiest things you can do, but I assumed it would be more difficult so I put it in the wanted goals section.

HTML is a constantly evolving language, and I'll have to continue using it in order to improve and maintain my skills. HTML and CSS are very useful languages and I plan on using them a lot for personal and professional purposes in the future. I believe I had a very good work ethic while working on this project. I came in during lunch and worked without distraction. I even took up an extra project (the website) to show off my skills.

The idea to make a website for my project was a fantastic one. Not only was it a way for me to showcase what I had learned, but I also learned a lot of stuff I could never have learned from tutorials and online lessons. I'll be sure to apply what I learned while building this website on my future websites.

In the future, I will definitely be practicing and improving my skills in HTML, CSS, and web design. My next benchmark goal will be to become competent enough to design a personal website and a professional website that isn't ugly which I can use to upload things to.