**VSCode Extensions List**

* LiveServer is required
* Bolded extensions are heavily suggested

-**Live server** (lets us right click an HTML file and open quickly “open with live server”)

-**vscode-icons** (lets us see better icons in our vscode)

-**Prettier** (code formatter, makes your code look more sane)

-**Intellicode** (gives us predictive options while writing our files, like how IntelliJ does)

**-Auto Rename Tag** (quality of life, automatically changes the other half of a tag you change)

-Better comments (gives us more options on comments)

-HTML CSS support (id and class attribute completion)

-Material Icon Theme (like vscode-icons but more options)

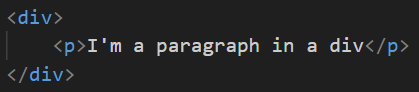
-Atom Keymap (gives support for useful keyboard shortcuts)

-Live Share (enables you to collaboratively edit and debug with others in real time)

**What is HTML?**

* **HyperText Markup Language (HTML)** is used to create web pages (front end!)
  + A markup language (HTML, XML, etc.) is **descriptive not programmatic.**
    - HTML is not a programming language. It’s a markup language!
* HTML is the standard for **displaying web pages on the internet.** 
  + **It’s the face of your webpage.** It creates what a web page looks like (along with CSS), whereas **Javascript** is what gives webpages functionality.
    - HTML = face (structure)
    - CSS = makeup (style)
    - JS = brain (logic – what makes everything work dynamically)
* **HTML syntax** is broken down into two subsets: **elements and attributes.**

**Syntax 1.) HTML Elements**

* Elements provide the **structure of the document** and are defined by tags.
  + E.g. <div> this is a div element, surrounded by the div tag </div>
* HTML must have its tags **properly nested** to be valid.
  + **Correct:**
    - 
  + **Incorrect:**
    - 
* There are two types of elements: (QC loves to ask about the differences)
  + **Block elements:** These elements will **render as a block on the page**. In other words, block elements **block** an entire line on the page for themselves.
  + **Inline elements:** These elements can sit next to each other on the same line when rendered. Can be rendered **in** the same **line** as each other.
* Common HTML elements (NOT a comprehensive list):
  + **<div>** defines a “division” of the page. Often contains other elements. (block)
  + **<h1>** through **<h6>** are header tags. 1 is the largest 6 is the smallest. (block)
  + **<p>** defines a paragraph, typical text elements (block)
  + **<span>** standard inline element, like an inline p (inline)
  + **<br>** line break. Creates a new blank line (doesn’t need closing tag) (inline)
  + **<img>** displays an image (does not need closing tag) (inline)
  + **<a>** anchor tag used to make a hyperlink. (inline)

**Syntax 2.) HTML Attributes**

* Attributes are metadata used to **describe the elements**; you can think of them as parameters defining the specific element tag.
* **Attributes are defined** **within a tag** of an element as a key/value pair.
  + For instance, in an image tag, the browser needs to know the location of the image: <img src=”[www.myimage.com/puppy/”](http://www.myimage.com/puppy%E2%80%9D/) />
    - src is the attribute in this case, telling the browser where to find the image
* **Global attributes** are those that can be applied to any tag. These will be very useful when we start messing with CSS. Some include:
  + **class** - you can group elements together under a class name
  + **id** - a good way to uniquely identify an element
  + **hidden** - used to hide an element
  + **style** - if you want to use inline styling (not best practice for CSS)
  + **title** - lets you add a text flag that appears when you scroll over an element

**HTML Doctype Declaration & Root Tag**

* <!DOCTYPE html>
  + This is the **doctype declaration.**  It informs the browser what type of document we are displaying (HTML) as well as the version if you want. The one above defines an HTML 5 document.
  + **THIS IS NOT CONSIDERED A TAG!** It’s the doctype declaration.
* The root tag of an HTML document is <html>. **All your HTML must go in between the opening and closing tags of this element.**
  + The HTML page is then further divided into a **<head>** section, usually for metadata, and the **<body>** section for actual content to be displayed.

\*Started HelloHTMLCSS\*

\*If you haven’t installed any of the extensions above, please at least install Live Server… It’ll let you launch your HTML onto a browser from VSCode, instead of having to go into your file explorer and find the HTML file to open in a browser\*

**HTML Lists**

* HTML has built-in functionality for displaying lists of items.
* There are two types of lists: **Ordered and Unordered**
  + **Ordered lists** have a progression such as 1 2 3… etc.
  + **Unordered lists** have no order. They are all represented by the same character - usually a bullet.
* Ordered lists use the **<ol>** tag
* Unordered lists use the **<ul>** tag.
* Items in both types of lists use the **<li>** tag (list item).
* Note: Lists of either type can be nested inside other lists.

**HTML Tables**

* Tables are an easy way to **structure our HTML pages and display information**. There was a time when almost all HTML pages were just giant tables.
* To create a table, we begin with a **<table>** tag
* **<thead> will define header rows**
  + Inside that, we can have the **<tr>** which creates actual rows
    - **<th>** will then give the actual cell data for that header cell
* **<tbody> defines the body of a table**
  + Each row of the table will have a **<tr>** tag
    - Individual cells in the row will have a **<td>** tag
* **<tfoot>** can be used to add a footer element to your table.

**HTML Input Types (HTML Input Conversation)**

* **Don’t use <forms> for full stack development; they’re tricky to use with the technology we’re using. Input boxes are probably your best bet at this stage. We’ll still cover forms because the different input types are important!**
* HTML has built-in functionality for forms, which take various inputs and **submit them to a server or handle them with Javascript etc.**
* **Input elements** include:
  + Text - general use text input
  + Password - prevents the inputs from being displayed on the screen.
  + Radio Buttons - select one from a list of options.
  + Checkboxes - select multiple from a list of options
  + Select Boxes - drop down menu to select an option
  + File select boxes - Allows you to upload a file from your local machine.
  + Text Area - A multi-line text field.
  + Reset and Submit buttons.
* **Attributes** used in forms:
  + **Type** - type of input you’re creating
  + Action - how (or where) the form data will be processed.
  + Target - Where the result will open: ie same window (self) or in a new tab/window (blank)
  + **Name** - each input needs a name value so the data can be labeled.
  + Method - What HTTP method the form will use to send the form. GET or POST.
  + **Value** - the initial value of the field.
  + **Placeholder** - this is a “hint” that describes what should be entered in the field.
  + **Required** - This field must have an input for the form to be submitted.
  + **Min and Max** - give minimum or maximum values that can be imputed into the field.
* The <label> tag can be used to give a text label to an input field, also helpful for extracting data from certain inputs like radio or checkbox.

**Not in the Curriculum but potentially useful------------------------**

* **Semantic tags** - HTML 5 added many semantic tags beyond what had already existed.
  + **“Semantic tags are element tags that are descriptive of the tag’s intended use beyond just their functionality”.** In other words, their name is self-explanatory
    - <article>
    - <header>
    - <footer>
    - <summary>
    - <section>
  + These contrast with typical tags like <div> and <span> because they clearly define what they do in the name.
    - These are good for society! They benefit **accessibility features** and are essential for screen readers for the visually impaired.
  + <https://www.tutorialspoint.com/html5/html5_new_tags.htm>
* HTML 5 also introduced elements to natively embed audio and video
  + <audio> will let you include an audio file (mp3, wav) to play on your page
  + <video> will let you include a video (mp4, webm)

**HTML 5 Specific Features**

* HTML 5 introduced the DOCTYPE declaration.
  + Before HTML 5, the DOCTYPE still existed, but was longer and uglier
  + Also, you can just type “!” in a new HTML file and it’ll give you skeleton tags thanks to emmet
* Character encoding declaration: <meta charset=”UTF-8”>
  + Covers pretty much every character in the world