HTML Notes

* 6 headers <h1> thru <h6>
* <div> divider block
* <b> bold
* <i> italicized
* <a href=<https://websiteurl> [target=“\_blank” for new tab]>Link name</a>
* Href to /otherpage.html to create links
* Lorem# inserts # of random words
* <title> webpage title
* <ul> unordered list
* <li> line item
* ./ to look in same directory
* ../ to go one folder up
* Question: how do you go two folders up?
* <form> for form input on pages
* <label> labels for forms
* <input type=“number,date,password, etc”/>
* <textarea rows=“#”> text area type of input
* Use <div> to wrap groups of text and images together
* Use <figure> before <img> for better formatting control
* <footer> tag for footers
* <a> for anchors

CSS Notes

* style=“color: red” within header to change colors
* Inline CSS
  + Style directly in the HTML element
* Internal CSS
  + Using <style> tags within an HTML file
* External CSS (recommended)
  + Linking an external CSS file
* max-width: sets maximum
* height: 100vh/vw; sets the height of the object to 100% viewport height/width (typically used on landing pages)
* border: 1px solid black; 1 pixel width, solid style, black border
* go to google fonts to download desired fonts and copy the html code and add it within the <head>
* Specificity is the prioritization of the styling
* class = “red” and .red in styles.css allows you define items as red
* class > basic tag
* id > class, but can only be used once in the html code
* id in styles.css is defined through #id
* Display Properties
  + block
    - Take 100% of the width and starts on a new line
    - e.g. <p> <div> tags
  + inline
    - doesn’t allow you to set a width or height for element & starts on the same line
    - e.g. <span> <img> tags
* Position
  + static (default)
    - follows normal flow of the page
  + relative
    - follows normal flow of the page
    - lets you position using top, left bottom, right
  + absolute
    - does not follow the normal flow of the page
    - lets you position with top, left, bottom, right
    - position is based off closest position: relative parent
  + fixed
    - sticks element to a permanent location no the page
    - lets you use top, left, bottom, right
  + sticky
    - sticks element to a permanent location non the page once it hits a threshold
* What are media queries used for?
  + in styles.css @media (max-width: ##px)
* Media Queries
  + Common breakpoints (max-width)
  + 576px cellphones
  + 768px tablets
  + 992px desktops
  + 1200px+ large desktops
* Pseudo Selectors
  + after
  + before
  + styles.css element::after/before to add a conditional object/text before/after the text
* Hover class
  + element:hover {element trait that changes when hovering}
  + best practice to put the transition: element trait that changes time in the a styles.css for that element
* Keyframes
  + used to create floating objects
  + @keyframes action\_name {

0% {transform: translateY(0);}

100% {transform:translateY(10px);}

element{ animation: action\_name time infinite alternate-reverse}

* BEM (Block-Element-Modifier) is a naming convention used for CSS classes to make them easier to read and understand
  + B [block] is the block component
  + E [ element] is dependent on the block component
  + M [modifier] is used to change the style of the block

Flexbox Layout

* display: block; is the default display type for elements
* display: flex; puts elements along the same line, side-by-side
* flex-direction: row; (default)
* flex-direction: row-reverse/column-reverse; to reverse the order at which things are displayed
* align-items: stretch; (default) stretch item to full height of the container
* align-items: center; centers item vertically
* align-items: flex-end; puts item on the floor of the container
* justify-content: flex-start; left aligns items
* justify-content: center; centers the items
* justify-content: space-around; create even space around all items in container
* justify-content: space-between; create even space between all items in container
* align-self: singular/subset item(s) property;
* order: order the items
* flex-wrap: wrap; wraps items into multiple lines. nowrap; wraps items into single line. wrap-reverse; items wrap around to multiple lines in reverse.
* flex-flow: to combine flex-direction and flex-wrap into a single line.

Best Practices

* starting style file with
* @import url('https://fonts.googleapis.com/css2?family=Roboto:ital,wght@0,100;0,300;0,400;0,500;0,700;0,900;1,100;1,300;1,400;1,500;1,700;1,900&display=swap');
* \*{
* font-family: 'Roboto', sans-serif;
* padding: 0;
* margin: 0;
* box-sizing: border-box;
* }

Git Hub

* To update work onto github
  + git add . or git add .\file.html
  + git commit -m “Description of the file”
  + git push