4. Basics of CSS

4.1 Introduction

- **What is CSS (Cascading Style Sheets)?**

  - CSS defines rules specifying styles that should be applied to HTML elements or groups of elements. With CSS, you can control the color, font, the size of text, the spacing between elements, how elements are positioned and laid out, what background images or background colors are to be used, different displays for different devices and screen sizes, and much more. Simply speaking, CSS is a set of instructions telling the browser what to do with the HTML elements. Ex.

    - What the color of an element should be

    - Where exactly an element should be placed on the browser

    - How wide or tall an element should be

    - As the name indicates, the styling in general

    - Note: CSS is useless without the HTML elements to apply the instructions on

- **What does cascading mean?**

  - The word cascading means that a style applied to a parent element will also apply to all children elements within the parent. So, if you set the color of the body text to "blue", all headings, paragraphs, and other text elements within the body will also get the same color (unless you specify something else).

4.2 Understanding CSS (CSS versions and sending styling instructions to the browser)

- **CSS Versions**

  - CSS3 is the official latest version of CSS

  - CSS3 is developed on top of the main principles of the previous versions. It just made things simpler and added a couple of additional features.

  - Just focus on CSS3 as everything from the previous versions is included in it

  - If you want to know more about the differences between versions, refer to this link

- https://hackr.io/blog/difference-between-css-css2-and-css3

- **Sending styling instructions to the browser**

  - Your job is to make the browser understand what you are trying to accomplish. Your browser is the builder. You are only allowed to send instructions on how it should build it, or technically speaking, render it. If you follow the standard way of telling the browser what to do with your HTML elements, it translates it to the right way and renders it in the way you want it.

  - You use the combination of HTML and CSS to send instructions to the browser. It is like sending a written instruction to someone who is supposed to rearrange your house in a way you want it without showing them pictures or videos, just written instruction. Let's say the person rearranging your house only speaks English. You don't have to know or use every single English word to effectively communicate with this person. You just use the words you know effectively. In this case, you are directly communicating with the browser. Same concept when using CSS to instruct the browser to style our HTML. You just use the CSS rules you know to instruct the browser.

4.3 Adding CSS instructions to HTML documents

- There are multiple ways of adding CSS instructions into your HTML. We will discuss the three ways below.

  - **1. Internal (adding CSS instructions on the HTML file/ on-file)**

    - An internal CSS is defined in the <head> section of an HTML page, within a <style> element. Ex: add an on-file style that changes the background of the body tag to red

  - **2. Inline**

    - The main difference between the inline styling and internal styling is that inline styling is directly applied inside the opening tag of the HTML element. For that, we use the "style" attribute of an HTML element. Ex: add an inline style on the <body> tag to change the background color to blue

  - **3. External CSS file**

- This is the recommended way of including your CSS instructions on your HTML file. The main concept is to create a separate CSS file containing all the CSS instructions and to link this CSS file to the HTML file. We add a <link> tag in the <head> section to link our CSS file to our HTML file. Ex: add an external CSS file to change the background color to purple

4.4 CSS units

- Having a clear understanding of the measuring units is the bases for any design and styling job. There are a couple of commonly used measurement units in CSS.

  - **The most common one is the pixel (px)**. It is the size of 1/96th of an inch (0.26mm). You must train your brain to think in terms of pixels from now on.

  - **Other commonly used units include**:

    - **Percentage %**: this is when you want to describe your element in relation with other elements, usually the containing elements

    - **em and rem**: these are usually used to define font sizes, and both are scalable units of size

      o The em unit is relative to the font size of its parent element. In the example below, the child will have a font-size of 36px (2 * 18px = 36px)

      ```
      .parent { font-size: 18px;}
      .child {font-size: 2em;}
      ```

    - The rem unit, short for root em, is a relative unit that'll always be based upon the font-size value of the root element, which is the <html> element. And if the <html> element doesn't have a specified font-size, the browser default of 16px is used. So that means that, by using the rem unit, the values of parent elements are ignored, and only the value of the root is taken into consideration. If look at the following example, the child will have a font size of 32px (2 * 16px = 32px)

      ```
      .html {font-size: 16px;}
      ```

```
.parent {font-size: 15px;}

.child-rem {font-size: 2rem;}
```

- For further reading about CSS units, refer to this link:

  - https://www.w3schools.com/cssref/css_units.asp

4.5 Thinking in terms of containers

- This concept of wrapping elements in containers is key in mastering CSS. Note that the wrappers or containers are HTML elements. The styling is mostly applied on them. Commonly used HTML containers are <header>, <section>, <div> and <footer>

- Before you even start writing a single CSS instruction, make sure to focus on organizing your HTML elements. Organizing your HTML means bundling related elements together with in a container. Try to think of the best way to group your elements together

- To help you understand what we mean by "thinking in terms of containers" let's work on a small project called "The Puppy Lover's Page" together. First thing first, every web page development starts with a design. Designers are responsible to provide that to you Look at the Puppies page design: https://evangadi.com/courses/phase1-docs/css-class/puppy-Lovers-Designes-updated.zip)

  - It is based on the design that you start to think about the structure of your HTML

  - Look at the conceptual containers you should think of before building your HTML

  - Then you build your HTML matching the containers you imagined

  - Look at the demo video on this section

4.6 Naming containers and elements using id and class

- One way of styling specific elements in CSS is to name your HTML elements. Naming is applied on the HTML elements. When you name, make sure to name every container in a way that is uniquely identifiable. It is these names we later use to specifically identify the element we want to style. We use classes and ids for naming elements. Classes and ids are always placed in the opening HTML tag. Please note that you can give any name you think is appropriate

- **Naming elements with classes**

- We use the class attribute to style multiple elements on our web page. Classes are like family names. Each element can have more than one class attribute value

- Ex.

```
<div class="puppy-container" > Hello  </div>
```

- **Naming elements with Ids**

  - Ids are like unique names. Use the id attribute to style one specific element on your web page. Each element can have only one id attribute value, and the attribute value must appear only once within the HTML file. Please remember that:

    - You shouldn't have similar id names in a single page

    - No element can have more than one id attribute value

    - Ex.

```
<div id="puppy-wrapper" > Hello  </div>
```

4.7 Selecting elements using id, class and element type

- Like we have said earlier, CSS is just a set of instructions the browser should follow to render the HTML elements. To write CSS instructions for an HTML element, we need to first correctly identify the element, this is what selecting means. There are multiple ways of selecting a specific HTML element. Here is a useful link listing out the 30 most used CSS selectors: https://code.tutsplus.com/tutorials/the-30-css-selectors-you-must-memorize--net-16048

- **Class selector**

  - This is a way of selecting an HTML element using their class name. We use the dot (.) identifier for that

Ex:

```
.className {

  /* Instruction goes in here */

}

.bold-title {

  color: red;

  font-weight: bold;

}
```

- **ID selector**

  ▪ This is a way of selecting an HTML element using their id. We use the hashtag (#) identifier for that

    Ex:

```
#id {

/* Instruction goes in here */

  }

#header-wrapper{

background-color: black;

}
```

- **Type selector (element type selector)**

  ▪ A type selector (sometimes referred to as an element type selector) matches elements with the corresponding element name, such as <p>, <span>, and <div> tags. Type selectors are generally used to make "broad stroke" changes to the style of a site. Example, if we want all the links on our website to be of color red, we use the following selection and styling

```
        a  {
```

```
        color: red;

    }
```

4.8 Selecting elements using star (wild selector), hover and descendant selectors

- **The star/wild/universal selector**

    - The * selector selects all elements. This is when we want to apply a certain instruction on every single HTML element. Example, if we don't want any of the HTML elements to have a margin (margin is just the space outside of an element's boarder), we use the * selector to make the margin bordering ALL elements 0 like below:

    ```
    * {

      margin: 0;

    }
    ```

- **Descendant selector**

    - Sometimes, just using the class name or the element type might not provide a specific selection as we want. In those cases, we use descendant selector style for that.

    - The descendant selector matches all elements that are descendants of a specified element. You can assume this as using the full name of a person. If we have 2 friends with similar first name, we can use their last name to identify one from the other.

    - **The Parent and Child concept**

        - The outer wrapper of an HTML tag is called the parent

        - The one in the inside the parent HTML is the child. Look at the example below to make every anchor link on the website red, but if the anchor link is inside a div, to make the that specific link white

        ```
        a {

        color: red;

        div a {

        color: white;
        ```

```
}/* Note: CSS here uses parent div first and the
child next    */
```

o **Hover selector**

- This is more of an action selector. It gets activated when you hover your cursor on an HTML element. For example, look at the following example to change the color of the anchor link to green and underline it when people hover over the line.

```
a:hover {

color: green;

text-decoration: underline;

}
```