CSS BASICS

Introduction:

- CSS provides the styling capabilities.
- Structure alone is not enough; we want the content to be pleasing to the users.
- We can style, color, and do designing in any way we want.

Anatomy of a CSS Rule:

- CSS works by associating rules with HTML elements which govern how they should be displayed.
- The CSS rule contains a selector, after which we have curly brackets and within them is our CSS declaration.
- The declaration contains a property and a value.
- p {color: blue;}
- Stylesheets are files that have multiple CSS rules defined within it.
- Every browser comes with default styles it applies on HTML elements; we overwrite some.

Elements, Classes and ID Selectors:

- CSS selectors are used to determine which set of elements the CSS declarations be applied to.

- **ELEMENT SELECTOR:**
    - We specify only the element name and then declare our CSS.
    - It only styles the elements we target, and doesn't affect other.
    - E.g: **p { color: blue; }**

- **CLASS SELECTOR:**
    - We specify this with a dot and the class name.
    - Targets all that element which have attribute **"class"** with that name.
    - E.g: **.blue { color: blue; }**

- **ID SELECTOR:**
    - We specify this with pound sign and the id name.
    - Targets all the elements with attribute **"id"** with that name.
    - Can only work with 1 element as id is unique for each element.
    - E.g: **#name { color: blue; }**

- CSS allows use to group several selectors into on CSS rule.
- Separate the selectors with comma.
- E.g: **div, .blue, #id1 { color: blue; }**

Combining Selectors:

- This technique allows us to precisely target the elements we want to style.
- There are several ways of doing this:

- **1. ELEMENT + CLASS:**
  - E.g: **p.big { font-size: 20px }, styles every "p" with class "big".**
  - No space between the selectors.
  - This technique is used when we have **"class"** with multiple elements and we don't want to style all of them.

- **2. CHILD SELECTOR:**
  - Syntax: **selector1 > selector2 { … }**
  - It is read from right to left. Selector2 is the direct child and it is targeted.
  - E.g: **article > p { color: blue; }**
  - <article> …. </article> = **Affected.**
  - <article> … </article> = **Unaffected.**
  - … = **Unaffected.**

- **3. DESENDENT SELECTOR:**
  - Syntax: **selector1 selector2 {…}**
  - It is read right to left. Every selector2 that is in selector1 at any level is affected.
  - E.g: **article > p { color: blue; }**
  - <article> …. </article> = **Affected.**
  - <article> … </article> = **Affected.**
  - … = **Unaffected.**

- These combinations aren't limited to element selector only, we can also have combinations of class, id selectors too! We can have any combination with any type of selector.

Pseudo-Class Selectors:

- The ability to target elements (to style them) based on user interaction with the page.
- For e.g., we want the styling to change when use hovers over something.
- Syntax: **selector:pseudo-class {…}**
- There are many pseudo-class selectors such as **:link, :visited, :hover, :active, :nth-child(num)**
- States of Links:
    - **:link =** the style to use when there is no action done with the link. Initial stable state.
    - **:visited =** the style to use when link is visited at least one (opened at least once).
    - **Above two are usually used together to style the link behavior.**
    - **:hover =** style to use when we hover over the link. (Can also use this with normal tags)
    - **:active =** style the state when user has clicked the link with mouse but hasn't let go of the click.
    - **Above two are usually used together to style this bevavior.**

- **:nth-child(num):**
    - It allows to target a particular element in a list. If we want to target the 3rd element in a list we would use: **li:nth-child(3) {…}**
    - We can also use this to for e.g., style all the odd divs: **div:nth-child(odd**

- We can also combine pseudo selectors in any of the ways we learned above with any type of selector.