HTML CSS SUMMARY - 16/09/2022

IMPORTANT POINTS

- **HTML TAGS** : starting and ending part of HTML element, anything between < and > is a tag. **** is a tag.Most tags must be opened and closed in order to function but there are singleton tags as well. for example
.
- **Elements** : Building blocks of the language. Defined by a start tag, some content, and an end tag. Eg. ** this is element .** 2 types of elements :
    - Block
    - inline
- **Attribute** : Describe characteristics of HTML element. It is placed in the opening tag of the element. It provides Additional properties (attribute) to the element.
  Eg. **<img src="mydog.jpg" alt="A photo of my dog.">**
- **Block elements** : these always start on a new line. Always take full width available. eg . , ,
- **Inline elements** : these do not start on a new line. Inline elements only take up as much width as necessary. Eg. <a>, <img>, <button>
- **Semantic elements** : These elements have meaningful names which tell us about the type of content for example <header>, <footer>, <table>, <nav> etc. These make the code easier to write and understand for the developer as well as instruct the browser on how to treat them.
- **Non semantic elements** : they tell nothing about its content. example ,
- **HTML entities** : Some characters are reserved in HTML and they have special meaning when used in HTML documents. Eg. we cannot use < or > in HTML text as the browser might mix them up with tags. Character entities are used to display reserved characters in HTML. They look like **&entity_name** or **&#entity_number.** Entity names are easier to Remember but they are not supported on all browsers whereas entity numbers are supported widely.

Non-breaking space		
<	&lt	&#60
>	&gt	&#62
&	&amp	&#38

  the < p > define a paragraph.
  Output — `<p>the <p> tag defines a paragraph.</p>`

- Built in HTML validations :
    - Use validation when we want to get the right data, in the right format and is not omitted.
    - Use when we want to protect users data and ensure secure passwords
    - When we want to protect ourselves from malicious users.
- **Some validation attributes :**
    - Required
    - Minlength and maxlength
    - Min and max : Specifying minimum maximum values of numerical input types
    - Type : Specifies the type of data to be take as input. Ex. number, email address.
    - Pattern : Specifies a regular expression that defines a pattern the entered data needs to follow.

- Always use double quotes in tags

CSS

- **CSS selectors** : Helps us to find or select the HTML elements we want to style. There are five categories of selectors
    - **Simple** : based on name, class, id
    - **Combinator** : based on certain relation between them
        - Descendant : space.
        - Child : >
        - Adjacent sibling : + (elements 1 after another and have same parent)
        - General sibling : ~ (selects all elements that are next siblings of a specified element). Eg. div ~ p selects all <p> elements that are next siblings of elements.
    - **Pseudo-class selectors** : based on certain state. Eg. **:link, :hover, :active**
    - **Pseudo-element selectors** : select & style part of an element. Eg. **::before, ::after, ::marker**
    - **Attribute** : based on attribute or attribute value. Eg. **a[target="_blank"],** selects <a> elements with a target="_blank" attribute.
- **Specificity**: If there are two or more CSS rules that point to the same element, the selector with the highest specificity value will "win", and its style declaration will be applied to that HTML element. Priority order –
    - Inline styles > IDs > Classes, pseudo-classes, attribute selectors > Elements and pseudo-elements
    - To calculate specificity – id =100, class = 10, elem=1, inline style=100
    - If we use **!important** it will override even inline styles
    - If same specificity, latest rule wins
    - * and inherited values have 0 specificity
- **BOX MODEL** : wraps around every HTML element, consists of :
    - Margin
    - Padding
    - Borders
    - The Actual content

When you set the width and height properties of an element with CSS, you just set the width and height of the content area. To calculate the full size of an element, you must also add padding, borders and margins.

- **POSITION :** Specifies the type of positioning method used for an element. It can have 5 different values: Static, relative, fixed, absolute, sticky. We position elements using top, bottom ,left, right properties however these properties will not work unless position property is set first
    - **Static** : not affected by the top, bottom, left, and right. Not positioned in any special way, it is always positioned according to normal flow of the page (default position)
    - **Relative** : It is similar to Static positioning but we can use top bottom left and right properties to position the element relative to the position it would have originally been (in the doc flow if it was statically positioned). Generally used when we want to position something absolutely inside the current element.
    - **Absolute** : It completely removes the element from the document flow and all the other elements act as if the current element was present there; we can use this to position something relative to its parent (parent must not be statically positioned).
    - **Fixed** : Similar to absolute but the elements which are declared fixed are positioned relative to the HTML document and not related to it any of its parent. Stay fixed while scrolling
    - **Sticky** : It is the combination of relative and fixed. the element will stay in relative position until it moves out of the scroll window and then it will become fixed afterwards. Just like a sticky navbar.

- **TRANSFORM :**
  - Rotate
  - Scale : change size relative to the original size
  - Translate : move object left, right, up, bottom
  - Skew : stretch a object