CSS – Stands for Cascading Style Sheet

* Cascading Style Sheets, is a stylesheet language used to **control the appearance** and **layout of HTML elements** on a webpage.
* It allows developers to apply styles like colors, fonts, spacing, and positioning, helping to **create visually appealing** and **responsive websites.**

CSS rule consists of selector and declaration box:

Ex: h1{color:blue;font-size:12px}

h1 – Selector 🡪 HTML Elements you want to style.

Declaration 🡪 Property : Value;

color:blue;

Three ways to connect CSS to HTML

1. External
2. Internal
3. Inline

**1.External :**

An external style sheet is a separate CSS file that can be accessed by creating a link within the head section of the webpage.

**<link>** - Link tag specifies the relationship between the current document and the linked resource(CSS).

**rel** - attribute denotes the relationship for defining how the linked resource connects to the current html document.

Note :

rel is commonly used with link elements. but is also be used with <a> tag and <area> tag to define relationships.

**2.Internal:**

It is the way of writing the css in same html file by using <style> tag.

**3.Inline:**

* It is the way of writing the css in same line for the specific HTML tag.
* for this we need style attribute .

**Simple Selector :**

**1.Tag Name :**

* To target the element based on tag name itself we have to use tag Name selector.
* The symbol was the **tag name** itself.

**2.ID Name :**

* To target the elements uniquely we have to use an id name.
* id attribute **allows single** identifier name.
* The Symbol used is **hash (#).**

**3.Class Name :**

* To target the specific elements on the basis of class name we have to use class name.
* Class attribute **allows multiple** identifier names in the same attribute.
* Class names can be repetitive also.
* The symbol used is **dot (.)**

**4. Grouping :**

* To target multiple elements at a time we have to use a grouping selector.
* Whenever we need to pass similar properties for **multiple elements** we can use a grouping selector.
* The symbol used to combine all elements is **comma (,).**

**5.Universal :**

* It will target all the elements in the document including body tag too.
* The symbol used is **asterisk (\*).**

**Difference b/w ID and Class**

**ID :**<https://worksheets.clipart-library.com/images2/template-grid-html/template-grid-html-4.gif>

**Uniqueness**: 1 Per Page

**Usecase** : Unique Element

**Class :**

**Uniqueness**: Reusable Multiple Times

**Usecase :** Group of Element

**Margin : (top right bottom left)**

**Margin** is the **space outside the border** of an element. It separates the element from **other elements** on the page.

**Syntax:**

margin: 10px 20px 10px 20px;

margin: top right bottom left;

margin-top: 30px; 🡪 pushes the box 30px down from the top

margin-left: 30px; 🡪 pushes the box 30px from the left

**Padding : (top right bottom left)**

**Padding** is the **inner space** between the **content** and the **border** of an element.

**Syntax:**

padding: 10px 20px 10px 20px;

padding: top right bottom left;

Padding – top : 20px;🡪 add 20px inside the box from the top

Padding-left: 20px;🡪 add 20px inside the box from the left edge

**Border :**

**Border** in CSS is the **line that surrounds the padding and content** of an element. It visually separates the element from others.

Ex:

border: 1px solid black;

1px → Thickness

solid → Style (line type)

black → Color

**Box-Sizing :**

It specifies the behaviours of width and height property.

**1.Content-box :**

* Width and height apply only to the content area.
* Padding and border are added outside the width and height.

**2.Border-box :(Recommended)**

* Width and height include content +padding +border.
* The content area shrinks to make room for padding and border.

**Flex**

* Flexbox is short for flexible box layout.
* A flexbox always consists of flex container and flex items.

**Grid-Layout :**

It is ideal for creating overall layout of the page.

**Grid-element:**

1.Grid-Container

2.Grid-Items

**1.Grid-Container:**

* It defines the outer element of the grid where all the child elements are present.
* It can be defined by setting display :grid or display :inline-grid;

**Grid :**

This value defines **block-level grid container** means it behave like a block element taking full width available**.(similar to div)**

**Inline-Grid :**

This value defines **inline level grid container**, means it behave like a inline element taking as much width as its content **(similar to span)**

**2.Grid – Items:**

* The items are arranged vertically and or horizontally inside the container box based on the needs.
* All grid-items placed inside the grid-container.