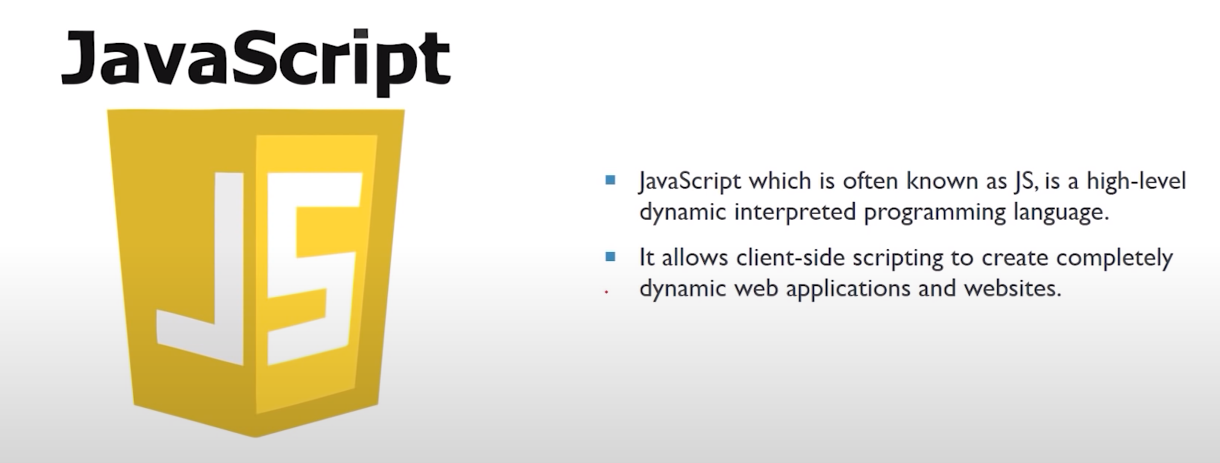
***HTML :***



***CSS :***

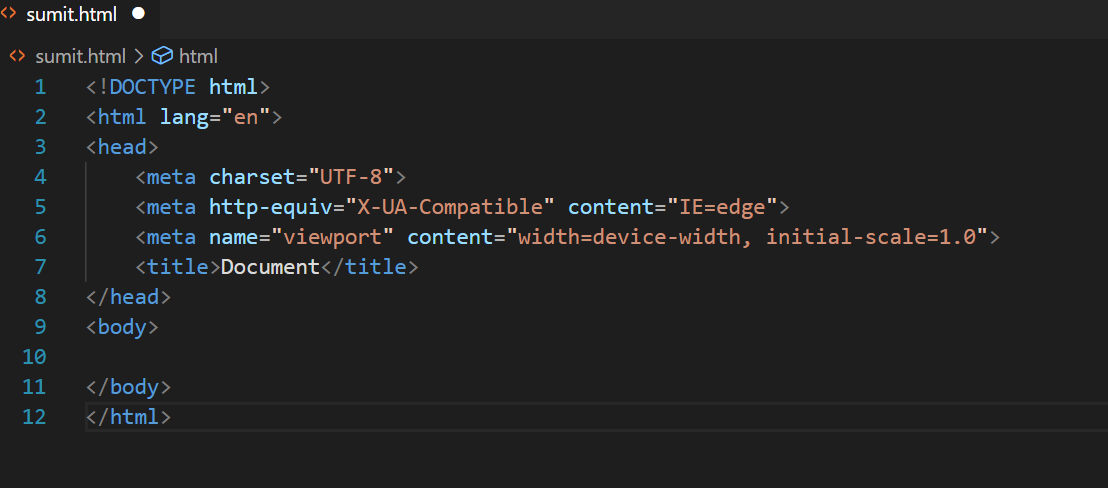
******

***JAVASCRIPT :***

******

***HTML:***

***Boiler Plate code of HTML :***

******

**To get it type:**

**! + enter**

<!DOCTYPE html>     <!-- It is Document Type -->

    <html lang="en">    <!--  It is Opening tag -->

    <head>

        <meta charset="UTF-8">

        <meta http-equiv="X-UA-Compatible" content="IE=edge">

        <meta name="viewport" content="width=device-width, initial-scale=1.0">

        <title>Document</title>

    </head>

    <body>

    </body>

    </html>       <!--  Closing tag -->

An opening tag begins a section of page content, and a closing tag ends it

The <head> element is a container for metadata (data about data) and is placed between the <html> tag and the <body> tag

 <!-- Including css file in html -->

    <link rel="stylesheet" href="sumit.css">

    <!-- including javascript file -->

    <script src="sumit.js"></script>

**The <body> tag defines the document's body. The <body> element contains all the contents of an HTML document, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.**

**We can use 6 heading from h1 to h6 and many paragraph p. Another paragraph tag starts new line.**

**Here, if we don’t want to start new line span tag is used.**

**Html cuts all the space and consider as it only one space.**

**The <strong> tag is used to define text with strong importance. The content inside is typically displayed in bold. Also <b> tag is used for bold the text.**

**The <em> tag is used to define emphasized text. The content inside is typically displayed in italic. Also <i> tag is used to convert text into italic.**

**<br> tag is used for line break and start new line. It is self** **closing tag. For self closing tag we does not need to close it.**

**In paragraph <p> tag if we want to create dummy words Lorem is used. If we want 100 dummy words just write Lorem100.**

**Tag <a> is used to open any link on internet . And it attribute target is used to whether link will be opened in new webpage or existing webpage.**

<a href="https://www.google.com/" target="\_blank">Click to go Google </a>

**Tag <img> is used to display image. It has two main attributes first is src which has link of image to be displayed and second alt which executed when image has some errors .**

<img src="https://source.unsplash.com/random" alt="Image" >

**Lists:**

**HTML lists allow web developers to group a set of related items in lists.**

**There are Two types of lists**

1. **Unordered list:**

**The <ul> tag defines an unordered (bulleted) list. Use the <ul> tag together with the <li> tag to create unordered lists.**

1. **Ordered List:**

**For ordered lists, use the <ol> tag is used.**

**+**

<ul>

        <li>This is first item of Unordered List</li>

        <li>This is Second item of Unordered List</li>

    </ul>

<ol type="i" >

        <li>This is first item of ordered List</li>

        <li>This is Second item of ordered List</li>

</ol>

**Table:**

**Tag <div> is used to create new line in webpage. The content of line must be written in opening and closing div tag.**

**Form tag:**

The <form> tag is used to create an HTML form for user input. The <form> element can contain one or more of the following form elements: <input> <textarea> <button>

Input tag is used to take various inputs from user. There are various types of input tag like text, search, url , email, password, date pickers, number, checkbox, radio, and file.

radio is used to give many options to user and select one of them. And all others take input as their names.

In attribute we write the keyword for what the value is stored and then accessed by backend.

submit is used to submit all above information in form. reset is used to reset all above information in form.

The <label> tag defines a label for several elements.

The <select> tag is used to select between the various options. Here, options given by <option> tag.

Id of every element must be unique. We can give many classes to element by giving space between them.

Shortcut -> After element . is for class and # is for id.

<div class="MyBG" id="unique">

        Hi Sumit

    </div>

    <!-- div.class1.class2.class3 + enter -->

    <div class="class1 class2 class3"></div>

    <!-- It inherit the property of that classes -->

    <!-- span#id1 + enter -->

    <span id="id1"></span>

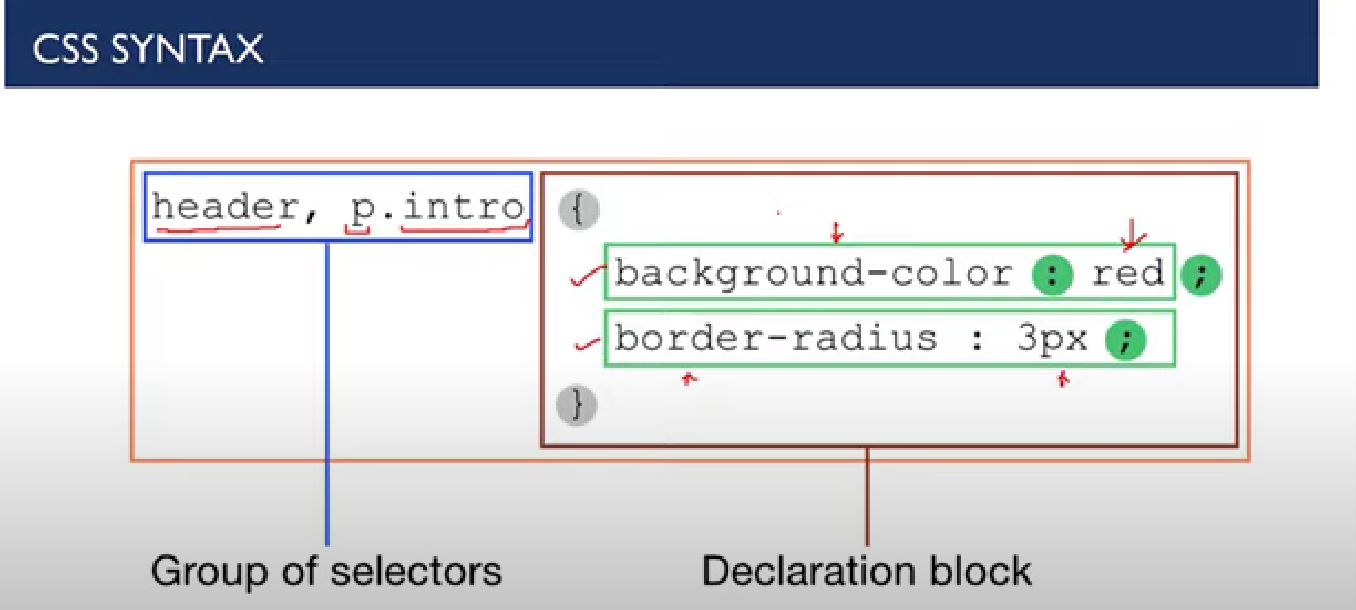
**Html entities:**

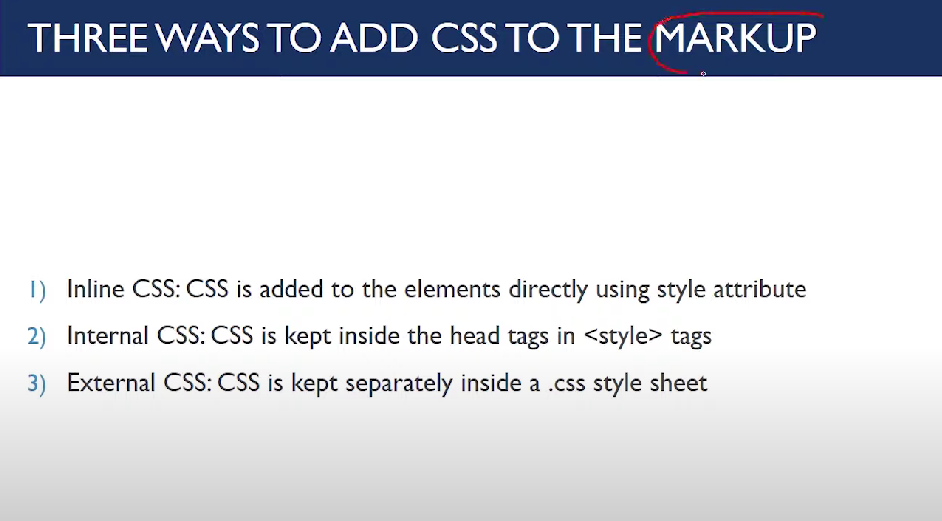
Some characters are reserved in HTML. If you use the less than (<) or greater than (>) signs in your text, the browser might mix them with tags. Character entities are used to display reserved characters in HTML.

&nbsp; is used for space.

***CSS:***







Inline css has more priority then external and internal css.

 In internal vs external the style which written later is executed. (Mhanje je ekdam last la ahe te).

No one can override important attribute

( !important ).

***Inline CSS:***

<p style="background-color: yellow; color: red;">

        Lorem ipsum

    </p>

***Internal CSS:***

<style>

        p{

            color: brown;

            background-color: burlywood;

        }

    </style>

***CSS Selectors:***

Selectors are used to target the HTML elements on our web pages that we want to style.

***Types of selectors:***

<style>

        /\* Element selector \*/

        p{

            border: 2px solid yellow;

        }

        /\* id selector \*/

        #first{

            color: tomato;

        }

        /\* class Selector \*/

        .bglightblue{

            background-color: lightblue;

        }

        /\* Grouping Selector \*/

        span , footer{

            background-color: pink;

        } </style>

id selector is used to give style to particular unique element.

class selector is used give style to multipe elements.

Element selectors are use to give style to that all same type elements.

Grouping Selector is used give that style to multiple elements.

The asterisk (\*), also known as the CSS universal selector, is used to select all items in an HTML file.

***FONT:***

<!-- importing font from the google -->

    <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Sofia">

<style>

        p{

            /\* font family can have multiple font styles  \*/

            font-family: "Sofia", 'Times New Roman', Times, serif;

            font-size: 35px;

            line-height: 2.2em;   /\* It is space between two lines \*/

            color: blue;

        }

        span{

            font-family: 'Trebuchet MS' , sans-serif;

            font-style: italic;

            font-weight: bolder;

        }

    </style>

***Border:***

<style>

        #firstpara{

            background-color: rgb(13, 203, 13);

            height: 300px;

            width: 500px;

            /\* border: 2px solid black; \*/

            border-width: 2px;

            border-style: solid;

            border-color: black;

            /\* border-radius: 40px; \*/

            border-top-right-radius: 50px ;

            border-bottom-left-radius: 50px;

            font-size: large;

        }

        #secondpara{

            background-color: rgb(117, 76, 204);

            height: 300px;

            width: 500px;

            border-top: 4px solid red;

            border-left: 5px solid orangered;

        }

        #thirdpara{

            height: 300px;

            width: 500px;

            border: 2px solid black;

            background-image: url(\tree.jpg);

            background-repeat: no-repeat;

            background-position:  center;

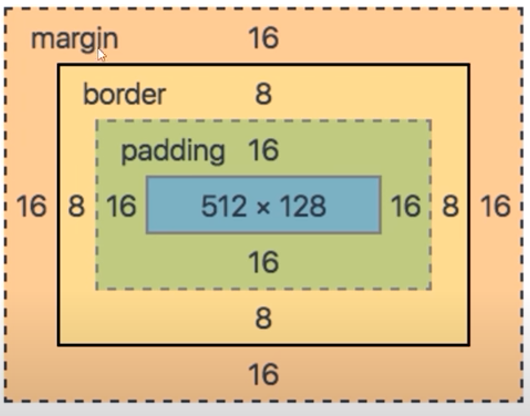
        }

    </style>

<!--Sourse = border.html -->

**Margin and Padding:**

In CSS, a margin is the space around an element's border, while padding is the space between an element's border and the element's content.



**The float Property:**

The float property is used for positioning and formatting content eg. let an image float left to the text in a container.

The float property can have one of the following values:

* left - The element floats to the left of its container
* right - The element floats to the right of its container
* none - The element does not float (will be displayed just where it occurs in the text). This is default
* inherit - The element inherits the float value of its parent

**The display: inline-block Value:**

Compared to display: inline, the major difference is that display: inline-block allows to set a width and height on the element.

Also, with display: inline-block, the top and bottom margins/paddings are respected, but with display: inline they are not.

Compared to display: block, the major difference is that display: inline-block does not add a line-break after the element, so the element can sit next to other elements.

visibility: hidden;

It hides the element but reserve the space for the element at that place.

**display: none;**

It removes the element from the screen.

**z-index:**

It only works for the position: relative, absolute, fixed or sticky.

The element which has highest z index is on the top.

Flex-wrap:

It adjust itself according to the container. If row is full start with new row.

**Justify-content:**

It is used to align main axis elements.

**align-items:**

It is used to align cross axis elements.

**order:**

Higher the order later it shown in the container

flex-grow:

 This property specifies how much the item will grow relative to the rest of the flexible items inside the same container.

* When flex direction set to row flex basis control width
* When flex direction set to column flex basis control height

**JAVASCRIPT :**

* We can write the javascript code in the both area of html head as well as body
* To print in the console of the website we use

console.log("Hello World!!");

## When to Use JavaScript const?

If you want a general rule: always declare variables with const.

If you think the value of the variable can change, use let.

JavaScript has 3 types of scope:

* Block scope
* Function scope
* Global scope

## Local Scope / Block Scope :

Variables declared within a JavaScript function, become **LOCAL** to the function.

### **Example**

// code here can NOT use carName  
  
function myFunction() {  
  let carName = "Volvo";  
  // code here CAN use carName  
}  
  
// code here can NOT use carName

## Function Scope:

Variables defined inside a function are not accessible (visible) from outside the function.

## Global JavaScript Variables

A variable declared outside a function, becomes **GLOBAL**.

var has a global scope and let has block scope

Redeclaring a variable using the var keyword can impose problems.