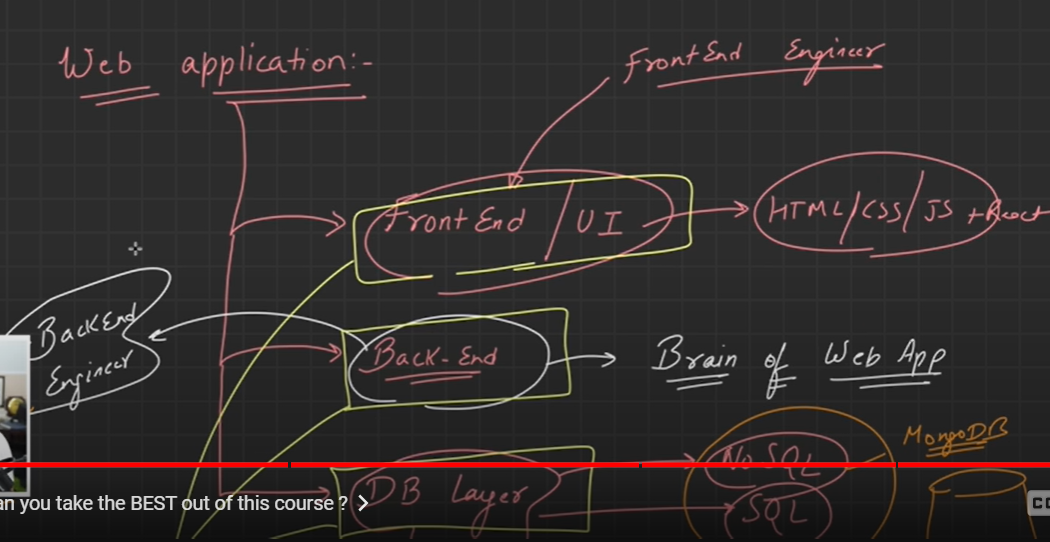
**Web Development**:

1. Web: It is a system jahan par bahut sare documents or resources aapas mein interconnected hote hein, and ye documents and resources can be accessed by using internet.
2. Web is a small entity which belongs to the bigger entity called internet.
3. **Development**: It is an entity through which you can create a lot of websites or web applications can be created.
4. Web development is a process basically with which you can create websites and web applications.

**Website:**  It contains static data and user interaction is negligible.

It contains read only data. Ex- blogs/Wikipedia

**Web Applications:** It contains dynamic data which can be changed after certain interval of time and a lot of user interaction is there.

For eg: 

**Network:**  Bahut sare resources aapas mein connect hote hein or ek system jisme sare k sare resources connected form mein exist karte hein usko network kehte hein. Ye sab aapas mein communicate kar sakte hein, resource sharing kar sakte hein or wo wired ya wireless form mein connect ho sakte hein.

**Types of Network**:

1. LAN: Chote scale ke network ko hum local area network kehte hein.

For eg: Network formed in a small room or building.

1. MAN: When cities are connected to each other and forming a network is called metropolitan area network.
2. WAN: When countries are interconnected to each other and forming a network is called a wide area network.

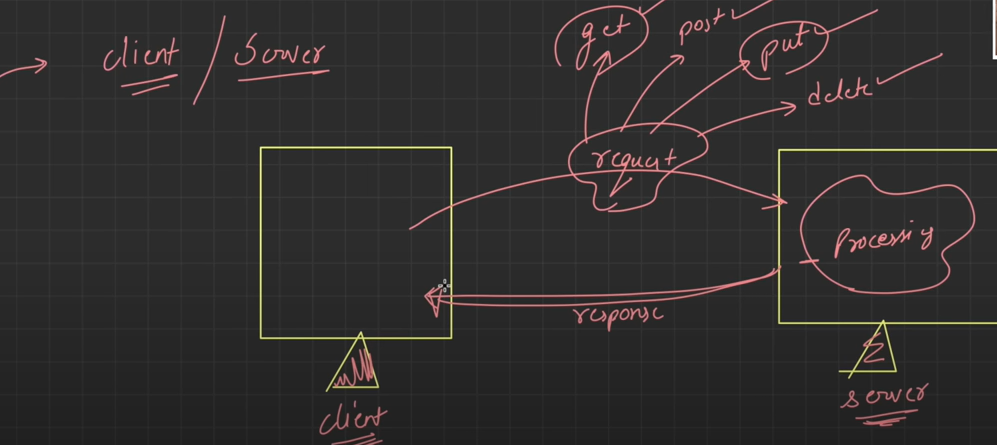
**Internet**: A global system in which billions of devices are connected with each other where everything exists, or a collection of networks of network is called internet.

**IP Address**: Agar aapki device kisi network par connected he toh us device ka ek unique identification address hota he toh us address ko hum IP Address kehte hein. Agr aap kisi device ko uniquely identify karna chaahte hein over a network we need IP address.

**Browser:**  As we know on internet there are so many resources available for eg. Documents, websites, web applications, so to access all these things we need a tool, that tool is our browser. It is just a software to access all these resources.

**Browser engine**: It is a tool or brain behind the browser which enables the execution of services for the browser. Jo bi code yahan available he usko run karne ke liye humein browser engine ki jaroorat padti he.

**Client Server Model:**

******

1. This is a client server model where client requests some resources from the server and server fulfils the request by sending the response over internet.
2. Requests can be of several types:
3. get: In this client requests some resources from the server or want to get some resources from the client. In simple words client server se kuch maang rha he ki mujhe ye data laakr do. Humlog (fetch) type ki request kar sakte hein using get.
4. post : In this type client server ko data de rha he ki is data ko rakhlo. Humlog (create) type ki request kar sakte hein using post.
5. put : In this type of request humlog data ko update karte hein for eg: Love babbar ko hum Lovely Babbar mein update karte hein. Humlog (update) type ki request kar sakte hein using put.
6. delete: We can make (remove) type of request using delete.

Client Server model: In this client pc which is located in india makes a request for resource from another pc which is in US fulfils its request by sending the response .

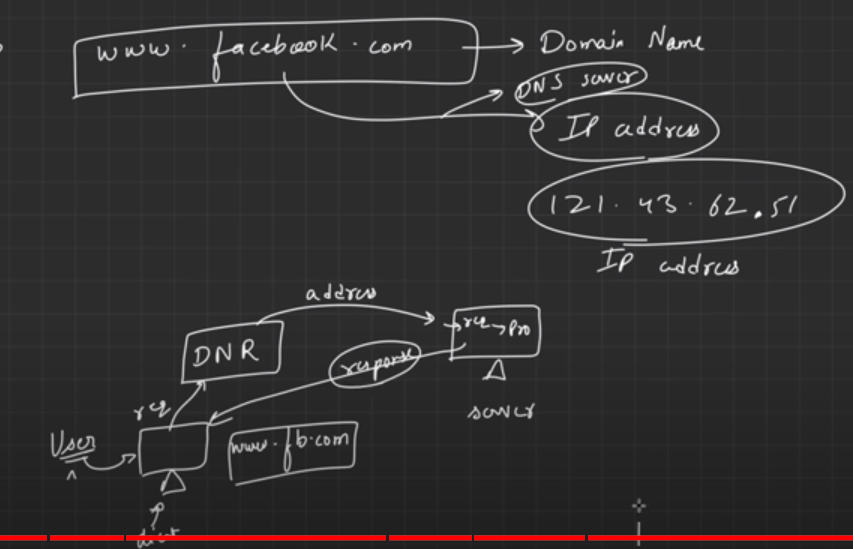
Different types of client: Laptop, Pc , mobile

Browser, App, CLI

Different types of server: DB server, file server, app server, web server,

Email server etc.

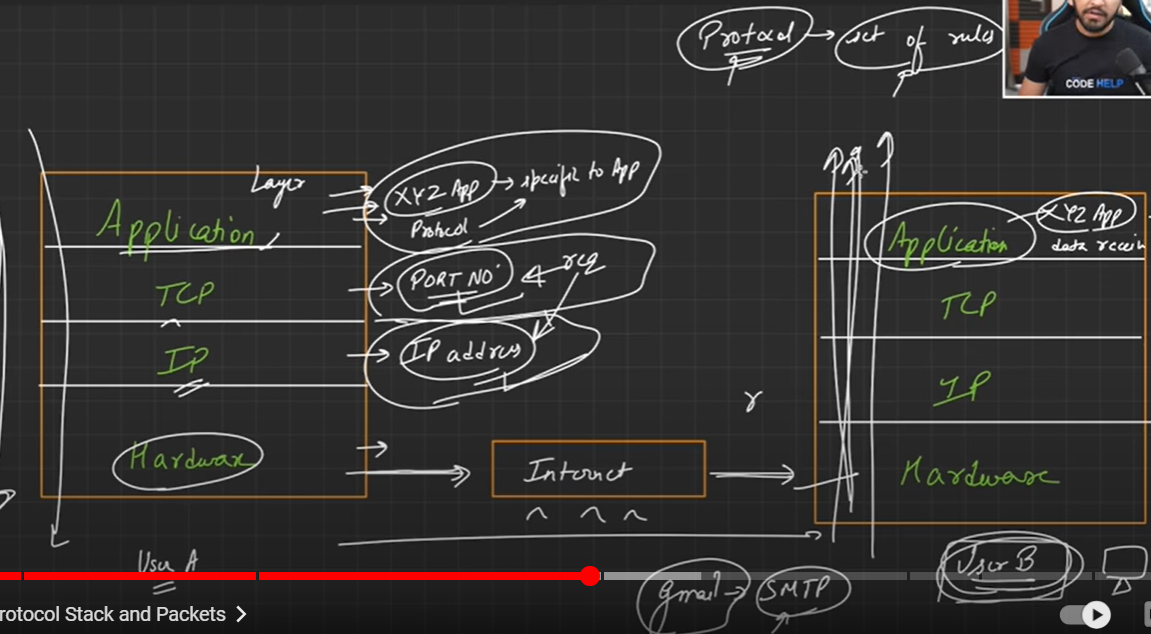
#Understand how the things look when we request any website on internet



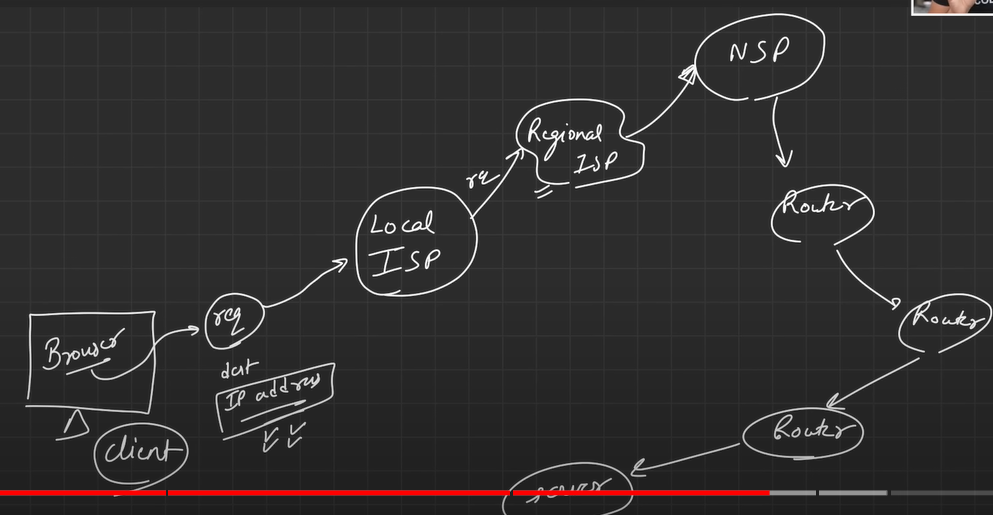
1. Jab hum [www.facebook.com](http://www.facebook.com) type karte hein toh toh [www.facebook.com](http://www.facebook.com) ek domain name hota h but humein domain name ni ip address chaiye hota he to reach any website. Ye domain name ko ip address mein badalne ke liye humein DNS server ki jarurat padti he jo domain name ko IP address mein convert karta h or phir us IP address ki madad se humlog us website par pahunch jaate hein.
2. So process start hota he ek client computer/browser se jo request karta he koi website, Phir Domain Name Resolution ki help se humko ek IP address mil jata he, Phir ye address ki request server ke pass jati he server request process karta he and server phir response wapas bhej deta he client ko.
3. In simple words hum koi bi request karte hein internet par us domain name ke corresponding Ip address generate hota h phir server response wapas bhej deta h us client ko .

#**Internet**: Inter stands for interconnected and net stands for network.

1. So basically internet stands for large number of interconnected networks where they can share information or resources with each other.
2. The client makes the request, that request goes to ISP which provides an IP address and now ISP access the internet.
3. ISP provides us with an IP address to work on with in a network.
4. **Packets:**  Suppose we want to send 100 mb file over a network from one pc to another, we divide the 100 mb file into smaller chunks, these smaller chunks are called Packets.
5. **When two applications are interacting with each other:**



1. Application layer is specific to application and uses a protocol to connect with another application layer of different user.
2. TCP layer provides us with the port number with which application you have to interact with.
3. IP layer gives us with the IP address of the given application with whom we have to interact and also tells us with which machine we have to interact.

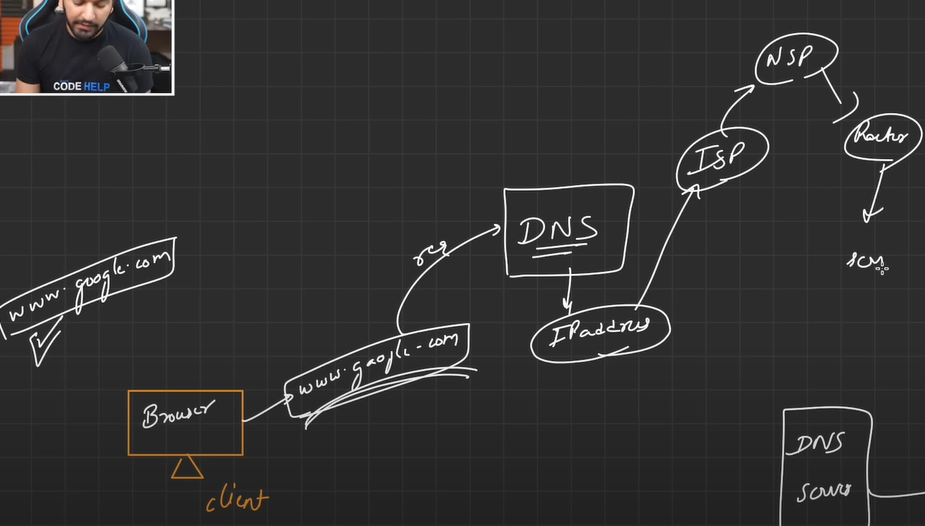


1. Browser makes the request in which he tell the destination IP Address which is requested, the request goes to the router, the router gives it an IP address, the request goes to the Local ISP where it finds that if the requested IP is available with it or not , if not it goes to the regional ISP and checks if the IP address is there or not, if the regional ISP also does not contain the IP address, it goes further to the NSP which is the main server and provides IP to all the ISP’s then it goes to the router, to several routers and then it ultimately goes to the server.

A black screen with white text

Description automatically generated

1. The client makes a request which goes to the router and form the router it goes to the ISP to check if the IP address is available or not, then from the ISP request travel back to the router and then it goes ultimately to the server.



1. The browser makes request [www.google.com](http://www.google.com) , this request is converted by DNS server into IP Address, this IP address is gone to the ISP and checks if that IP is available in the routing table or not, if not the request goes to the NSP and it checks in the routing table and if the IP address is available in it fulfills the request made by client and goes back to the router to the server.

**#HTML:**

**1.** Html is used for labelling the website.

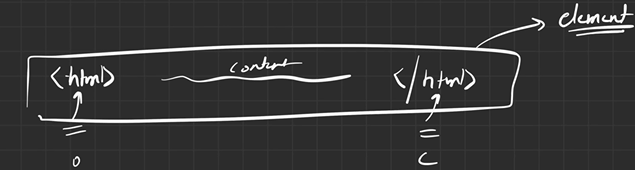
2.Html is used for creating the structure of the website.

3. Html is the skeleton of the website.

4. Html tells us what is present in the website and how it will look.

5. Jo bi humlog site par visible he uski labelling karta h.

**Element:**  An element in html consists of a opening or closing tag and in between there is some content written. Individual <html> is a tag and combined is called an element. <html> tag is the root element and in between this tag the code is written.



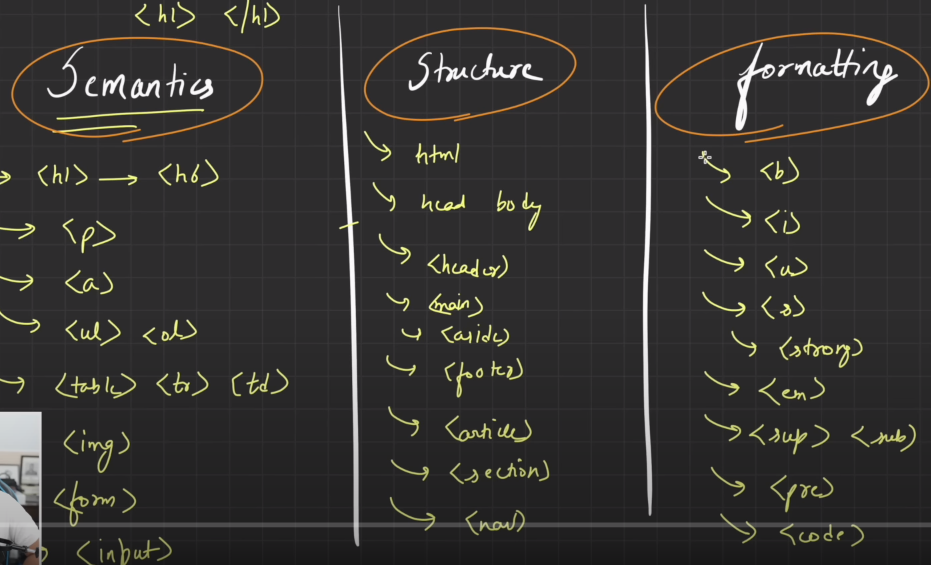
**<HEAD> tag:**  Head tag is used to show only <metadata>, <SEO>, <title>, ya kisi ki prakar ki stylesheet use karni he usme humlog head tag use karte hein.

**<BODY> tag:**  Body tag is used to display the content which we used to display. Sare ke sare labelling content or tags jo use karte hein hum wo body ke andar rakhte hein.

**TAGS**:

1. Tag is a markup entity, with which we can define the structure of a content.
2. It is used to define the meaning of the content of a webpage.
3. It can be used to format the content of a webpage.

**Types of Tags in HTML:**

****

**#Heading tag:**  This is used to define the heading in a webpage, the difference is in the semantics of the heading. All is dependent on the semantics of the page, humare H1 ko jyada importance milta he H6 ko kam milta he.

Ex.     <h1>This is my first heading</h1>

    <h2>This is my second heading</h2>

    <h3>This is my third heading</h3>

    <h4>This is my fourth heading</h4>

    <h5>This is my fifth heading</h5>

    <h6>This is my second heading</h6>

Rest all tags for ex. <H7> or <H8> all will behave like a paragraph tag.

**#Anchor Tag:** This tag is used to create a hyperlink from the webpage. If we want to send some mail or create bookmarks, hyperlink or call someone we use <a> tag. Ek webpage se dusre webpage par pahunchna chaahte hein toh hum is tag ka istemal karte hein.

For eg:     <a href="https://www.google.com">Click here</a>

    <a href="mailto:abc@gmail.com">Email</a>

    <a href="tel:982382736823">Phone number</a>

**#List:**  We have three types of list available in html.

Unordered list:     <ul>

        <li>Paani laayo</li>

        <li>Pani garam karo</li>

        <li>You are useless</li>

    </ul>

Ordered list:     <ol>

        <li>Lays</li>

        <li>Chips</li>

        <li>Kurkure</li>

    </ol>

**#Table Tag:**  This table is used to create a table in html. It has main <table> tag, then<tr> tag which is used to denote (table row) , then we have <td> tag which is used to denote (table data) in a table.

For eg: <table>

    <tr>

        <td>Row no.</td>

        <td>Column</td>

        <td>Name</td>

    </tr>

**#Image Tag:** This tag is used to insert image in webpage.

For eg:< img src =”Copy path of image or image address” alt =”Image address”>

# Here src and alt are the attributes of image tag.

**#Bookmark in Html:** To create a bookmark in html we use

1. First, use the id attribute to create a bookmark:
2. Then, add a link to the bookmark ("Jump to Chapter 4"), from within the same page:

For eg: <h2 id="C4">Chapter 4</h2>

<a href="#C4">Jump to chapter 4</a>

**#Style Tag:**

1. This tag is used for styling the html tags or beautifying it.
2. It is used to beautify the html tags.

For eg: <h2 style="background-color: aqua;">This is Heading2</h2>

<h2 style="color: crimson;">This is Another Heading 2</h2>

**#Paragraph Tag:** If we want to write a paragraph of 100 words or 50 words without typing the words we can use the (LOREM) keyword.

For eg: <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Optio harum dolorum placeat suscipit porro, eius vero id sit blanditiis ipsam?</p>

**#Strong tab:**  This tag is used to make the word bold and will give higher emphasis on the word.

For eg: <strong>This is strong tab</strong>

**#Em tag:**  This tag will make the word or sentence italic.

For eg: <em> This is emphasis tag</em>

**#S tag:**  This tag is used to make the word strike of or will make a cut line on the word.

For eg: <s> This is strike tag</s>

**#Sup tag:**  This tag is used to define the power of any word.

For eg: <p> 2<sup>5</sup></p>

**#Sub Tag**: This tag is used to define the subscript of any tag.

For eg: <p> O<sub>2</sub></p>

**#Pre tag:**  This tag is used to keep the sentence, words or paragraph in same sequence or humlog sentence ko jaise print karna chaahte hein waise hi print hota he.

<pre>

    THis

    tag

    is

    pre

    tag

</pre>

**#Structural tags:**

1. These tags are used to provide structure to the webpage.
2. There are different types of structural tags available in HTML.
3. <header> This tag is used to define the header or a webpage.
4. <footer> This tag is used to define the footer of the webpage.
5. <main> Apart from header or footer all the things in a webpage are written in the main tag.
6. <nav> This tag is used to define the navigation links in a webpage.
7. <article> If there is an independent article then we use the article tag.
8. <section> If there is multiple data and the data is interrelated among each other then we use this tag.
9. <aside> If we want to give a sidebar to a webpage we use this tag.

**Difference Between Article and Section Tags:**

* Both elements can be used to divide the content on your web page
* The HTML article element is used to wrap independent and self-contained content.
* The HTML section element is used to divide thematically defined pieces.

**#Emit:** This is a special method in html or shortcut present in VS which we can use to write the html code very quickly.

**1.p>p>p>p** – This will create 4 paragraphs.

**2.p>img** – This will create image tag inside the paragraph tag.

**3.Article+section** – this will create first an article tag then section tag.

The first tag will behave as parent tag and the second tag will act as section tag.

4.When we use the + tag it acts as a siblings for each other while when we use the > keyword it acts a parent child relation.

5. p{Content is inserted}- This tag helps us to insert text inside this paragraph.

**#ID attribute:**  Id jo attribute hota he wo kisi bi element ko uniquely identify karta h. For eg. Ek class mein 40 bacche hein unka sab ka roll number alag hota he waise hi ID hota he. Toh agar aap webpage create karte hein toh us webpage mein har kisi element ki ID alag hogi. To create a ID attribrute we use (#hastag sign). For eg- (p#p1)

**#Class attribute:** Class ek parent class identity hoti he, Class mein apke multiple elements exist karte hein, class humlog multiple elements ko grouping karne ke liye hota h. For eg. Dog class or uske andar labra, rotweler, german shepherd aa sakte hein. For eg: To create a class we use . (p.para)

1. We can use + symbol to make multiple insertions.

For eg:- p+article+section

**#DIV Tag:**

1. It is used to create division.
2. It is used as a generic container.
3. It is a wrapping tag or used as a container to store several small elements or tags.
4. We can create different different sections in a code or make divisions in a code.
5. To create a div tag we use .wrapper it will create a div tag syntax.
6. <div class="wrapper"> </div>

.class for eg- <div class="class"> </class>

**#Block elements:**

1. Block elements ese elements hote hein jo hamesa next line mein aate hein.
2. Left or right se jitna bi space hota he ye by default le lete hein.
3. Jo puri ki puri screen le rha he usko hum block element kehte hein.

**Eg:** <div style= background-color:aqua >This is my first div</div>

**#Inline elements:**

1. Inline elements ese elements hote hein jo next line mein nahi aate hein.
2. Jitna isme content hota he us hisaab se space le lete hein.
3. Jo srf jitna content he utni hi width le rha he usko hum inline element kehte hein.
4. Inline elements same line mein hi aate hein.
5. <span style= background-color:yellow> This is my first span tag </span>

**#Is it possible to convert block element to inline element**

1.You can set a block-level element to display like an inline element by setting the display property to inline. You can also cause inline elements to behave like block-level elements using the display property.

**#Tables In Html:**

1. There are four main tags in html tables-
2. <Table> tag used to create table.
3. <Thead> tag – used to create headings in a table.
4. <Tbody> – main content is put in body tag.
5. <Tfoot> – Summarization of the table is put under this tag.
6. <Tr> tag – It is used to create table row .
7. <Td> tag – It is used to put data in particular cells in a row.
8. <Th> tag – It is used to put data in table headings column.

For eg: <table border="1">

        <thead>

            <th>Student ID</th>

            <th>Student no.</th>

            <th>Gender</th>

            <th>age</th>

        </thead>

<tbody>

    <tr>

        <td>1234</td>

        <td colspan="3">Bill</td>

    </tr>

    <tr>

        <td>1235</td>

        <td>John</td>

        <td>Male</td>

        <td rowspan="3">18</td>

    </tr>

    <tr>

        <td>1236</td>

        <td>Olivia</td>

        <td>Female</td>

    </tr>

    <tr>

        <td>1237</td>

        <td>Mia</td>

        <td>Female</td>

    </tr>

<Rowspan> - To expand the given cell rowwise.

For eg:             <td rowspan="3">17</td> Here 3 represent the number of rows we have to expand the given cell.

<Colspan> - To expand the given cell columnwise.

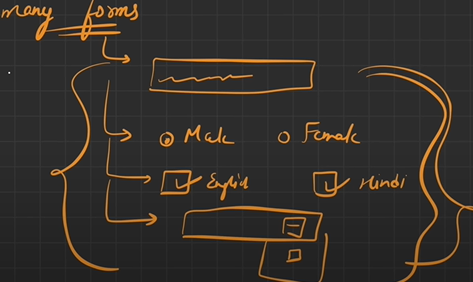
For eg:             <td colspan="3">Maly</td>

**#HTML Forms:**

1. They are used to collect user input data.
2. To make a form in Html we use the <form> tag.

**<input>** tag – This can be used to display in multiple forms.

1. This single tag can be used in multiple ways to take input from user.
2. Input tag jab bhi hum use karte hein toh uske sath type attribrute hum jarur use karte hein, type atrtibrute humein btatate he ki input kis type ka hoga radio hoga ki checkbox hoga ya or kuch.
3. Input tag by default type=”text” hota h.
4. For eg: It can be checkbox, radio button, a box where you have to input data and many more ways.
5. Kya check karna chaahte ho wo Type attribrute batata he humein.



**<label tag>:**

1.This tag tells us what we have to insert in the box, jo saath mein hum log text attach karte hein box ke side mein ki box mein kya bharna he use hi hum label tag kehte hein.

2. Labels are also used by screen readers which tells us what user has to enter by reading out aloud.

**Very Imp** 3. Har label ke andar ek for attribrute hota he jisko aapko complete karna hote he “ for = “ \_\_\_ “ jo ki input element ki id ke barabar hoti he. Label ka “for =\_\_\_” or input ki “ id = \_\_” attribrute ki value same hona chaiye.

**#How to create buttons in HTML:**

1. It can be created with the help of input tag.

For eg:         <label for="Form:">New Form</label>

        <input type="text" id="Form:">

        <br>

        <br>

        <label for="Enter Details">Details</label>

        <input type="text" id="Enter Details">

        <br>

        <br>

        <input type ="Button" value="Login here">

1. It can also be created with the help of button tag.

For eg: <input type="radio" value="Click here">

        <button>2nd button</button>

# If input type =”Submit” then pure form ka data submit ho jata he or ek submit button create ho jata he.

**#Placeholder attribute:**  This attribute gives us a faded text when we use it inside the html inputs.

For eg:         <label for="email"> Enter Email Id:</label>

        <input type="text" id="email" placeholder="Enter your email here">

**#Select tag:**  With select tag we can create drop down menus.

**#Option tag:**  This tag is used to create drop down options.

    <select name="dishes" id="Selector1">

        <option value="maggi">Maggie</option>

        <option value="kurkure">Kurkure</option>

        <option value="chips">Chips</option>

    </select>

**#Text area:**  This tag can be used to create multi line comments or enter larger text in one single box.

<textarea name="Textarea1" id="textarea" cols="20" rows="10" placeholder="Enter text here"></textarea>

<br>

<br>

<textarea name="Textarea2" id ="textarea2" cols="10" rows="5" placeholder="Enter your text here"></textarea>

**#Field set:**  This is used for grouping of multiple elements in a form.

**#Legend:**  This is used to create captions in a form, koi bi fieldset ka caption dena ho we use legend.

<fieldset>

<legend>This is dummy html!</legend>

<label for="cars">Choose a car:</label>

<select name="cars" id="cars">

    <option value="volvo">volvo</option>

    <option value="jaguar">jaguar</option>

    <option value="audi">Audi</option>

</select>

<fieldset>

**#How to select multiple drop-down options:**

<label for="cars">Choose a car</label>

<select name="cars" id="cars" multiple>

    <option value="volvo">volvo</option>

    <option value="jaguar">jaguar</option>

    <option value="audi">Audi</option>

</select>

To select multiple options we use Ctrl option.

**#Types of Input Tags:**

1. Text – This is by default in input tag even if we don’t specify this type html uses it by default. For eg: <input type=”text”>
2. <label for="Textfiled"> Enter your text here</label>
3. <input type="submit" name ="textfield" id="button">
4. <br>
5. <br>
6. Radio: type<radio>
7. <label for="rad">Radio1:</label>
8. <input type="radio" name="Radio1" id="rad">

**# How to select one radio button at a time:**

To achieve the desired behavior (i.e., selecting one button at a time), you should use the same **name** attribute for all the radio buttons that you want to act as a single group.

        <label for="rad1">Radio1:</label>

        <input type="radio" name="RadioGroup" id="rad1">

        <br>

        <br>

        <label for="rad2">Radio2:</label>

        <input type="radio" name="RadioGroup" id="rad2">

        <br>

        <br>

        <label for="rad3">Radio3:</label>

        <input type="radio" name="RadioGroup" id="rad3">

        <br>

1. **How to create a checkbox in HTML:**
2. To create a checkbox in html we use the type=”checkbox”

For eg:         <label for="Checkbox1">Checkbox1:</label>

        <input type="checkbox" name="Checkbox1" id="checkbox1">

        <br>

        <br>

        <label for="Checkbox2">Checkbox2:</label>

        <input type="checkbox" name="Checkbox2" id="Checkbox2">

1. **File- select:**

To create a browse button or upload file button we use the file-select option.

With the help of this button we can upload any file from the system. We can use the type<file>.

For eg:

<label for="Fileselect1">Upload a File</label>

<input type="file" name="File-select1" id="Fileselect1">

1. **To make image a button:**

To create the image a button we use the type<image> tag.

For eg: <label for="Image1" >Click here</label>

<input type="image" name="Image1" id="Image1" alt="DemoImage" src="55008.png">

1. **To hide the password:**

To hide the password we use the type<password> tag. Then the password will show dotted.

For eg:

<label for="Password1">Enter your password</label>

<input type="password" name="Password1" id="Password1">

1. **To create a Date type dropdown:**

To create a date type menu or to open the calendar so that we can choose the date from the options we use the type<date>.

For eg: <label for="Calendar">Choose the date</label>

<input type="date" name="Calendar" id="Calendar">

1. **How to select a by default value in a Dropdown:**

To set a default value in a dropdown we have to use the tag as <selected>.

9.**To make a dropdown in html** :

To make a dropdown we use :

1. <details> as closing and opening tag. </details>
2. The inside details tag we use <summary> tag.
3. Then inside <summary> tag we use the <p> tag.

Eg:               <details>

                       <summary>

                        What is your name?

                    </summary>

                       <p>My name is Ankit.</p>

                   </details>

                   <details>

                       <summary>What is your age?</summary>

                       <p>My age is 27.</p>

                   </details>

**# Media Tags in HTML:**

1. To put a image in html we use the <img> tag.

**For eg:** <img src="DSC\_9367 copy.jpg" alt="Demo Image" width="100px" height="100px" loading="lazy">

1. To put a audio file in html we use <audio> tag. It is very important to put controls tag in audio tag, if we don’t use it the audio tag will not work.

For eg:

<audio src="Audio.mp3" controls></audio>

1. To put a video file in html we use <video> tag. In this tag also it is very important to put <controls> tag, if we don’t use it the video tag will not work. We can also define the width and height of the given video tag with the help of <width=”100px”> and <height=”100px”>.

For eg:           <video src="Video.mp4" controls width="500px" height="500px"></video>

1. **Iframe tag:**  If we used to refer some external document in an existing document we use the iframe tag. With the help of this tag humlog current page ke andar external document ko show ka sakte hein.

For eg:   <iframe src="../Html\_Project.html" frameborder="1" width="800px" height="300px"></iframe>

**#How to add Youtube Videos with the help of <iframe> tag:**

1. To add youtube videos with iframe tag we use <embed> tag, then extra information which is irrelevant in the youtubelink, for eg- watch, remove ? and v and put the video /id make it in a sequence like this - <https://www.youtube.com/embed/zFMJZro-zmc> . Then the video will be visible on the webpage.

For eg: <iframe src="https://www.youtube.com/embed/zFMJZro-zmc" frameborder="1"></iframe>

**#How to create a bookmark in HTML:**

1. To create a bookmark first we have to assign the id to the element which we want to assign the bookmark and then in the <href> section we give the same <id> we want to create for the bookmark.
2. <section id="DSA Video">
3. <li><a href="#DSA Video">DSA Video</a></li>

2.We can also assign the autoplay option and mute option in a video with the help of – (?autoplay=1) and (&mute=1)

For eg:     <td><iframe src="https://www.youtube.com/embed/0oe7CF-TY\_U?autoplay=1&mute=1" width="250" height="250"></iframe></td>

**# Inside Head Tag:**

1. **Title tag:** Title tag is used to set the title of the webpage as same, if you gave the <Title> as Web Development <Title> it will display the same.

**For eg:**     <title>My Webpage</title>

1. **Favicon:** The image that is beside the title of the webpage we give is given by the favicon tag. Jo image hum log ko add karni hoti he favicon tag ke andar usko hamesa hum log root directory ke andar save karte hein.

For eg:    <link rel="shortcut icon" href="download.png" type="image/x-icon">

**#How to link a CSS file with HTML:**

1. To link an CSS file with html we use the link rel=”stylesheet” href=”style.css” tag.

For eg:     <link rel="stylesheet" href="style.css">

**# How to link a JS file with HTML:**

1. To link a JS file with html we use the <script> tag.

For eg:     <script src="script.js"></script>

**# SEO – Search Engine Optimisation:**

1. Sari fight humari website ko jab search karein tab website sabse upar aani chaiye, toh search engine optimization se esi technique lagate hein jisse hum jab apni website search karein toh wo search karte time sabse upar aaye, usme kuch percent yogdaan humare <meta> tags dete hein jo ki <head> tag ke andar aate hein.
2. <meta charset="UTF-8">

Ye line humko bta rhi he ki ye code jyadatar languages ko support karta h.

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

Ye line humko batati he jo bhi code humlog render kar rhe hote hein Internet Explorer par toh jo bhi sabse latest rendering engine hoga usko use kar rhe hoge.

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

Ye line humko batata he jab hum browser kholte hein toh default webpage khulta he uska initial zoom level by default 1 rakhna he yani 100% or uski width device ke width ke barabar hogi.

    <meta name="description" content="This website is used to build India's best coders">

This tag gives the description about the website and is visible when we inspect the code and search the meta tag in description option.

   <meta name="keywords" content="codehelp, codehelp love babbar, codehelp github, codehelp love">

This tag is used to add keywords to the website so that if these keywords are searched on the browser, this website pops up first.

**CSS**

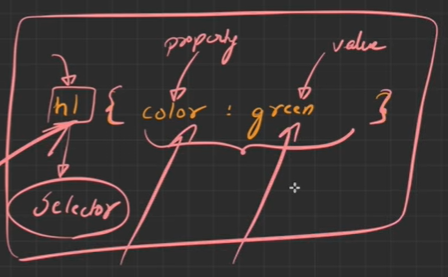
**# Inline CSS:**

1. This type of css happens when we defines the styling tags inside the tags itself, jab hum code element ke andar hi de dete hein.
2. This type of method should not be used more because its not considered good practice and jab code complex ho jaata he toh ye code ki readability khatam kar deta he.

For eg:     <p style="background-color: aqua;">Lorem ipsum dolor, sit amet consectetur adipisicing elit. Facere numquam ad cum laborum accusamus esse hic voluptate. Ratione, a ipsam.</p>

**# Internal CSS:**

1. This type of tags are given inside the <head> tags, and they gives styling to the webpage.
2. <head>
3. <style>
4. h1{
5. color: blueviolet;
6. }
7. </style>
8. </head>



1. Selector: Ye humko batata he ki kis element ko style karna he humko, or mera target element kya he jisko humko style karna he.
2. Brackets ke andar jo bi likha hota he unke andar hum consider karenge styling ke liye.
3. Left side par jo likha he usko hum property kehte hein.
4. Right side wala ko hum value kehte hein.

**\*\* Very Important Point:**

1. Jab bhi inline css or internal css mein do property ka clash hoga toh jo tag inline css mein he wo hi consider hoga hamesa.

For eg:

p{

            background-color: brown;

        }

 <p style="background-color: aqua;">Lorem ipsum dolor, sit amet consectetur adipisicing elit.</p>

Is wale case mein inline css wala bg-color consider hoga.

**\*\* Important:**

1.Jab bhi inline css or internal css mein do property ka clash hoga toh jo tag inline css mein he wo hi consider hoga hamesa ye point bhi reverse ho sakta he.

1. Jab humko inline css mein jo tag likha he usi ko consider karwana he chahhein koi bi rule ho tab humlog ( !important ) tag ka use karte hein.

For eg:     <style>

        p{

            background-color: brown !important;

        }

**#External CSS:**

1. Jab bhi humlog external css ki baat karenge toh humlog ko do points ka dhyaan rakhna padega-
2. Pehle ek css file ko create karna padega.
3. Us css file ko link karna padega humein us html file se.

**# How CSS tags(Internal, Inline, External) work in orderwise:**

1. Code mein jis sequence mein (Internal, Inline, External) codes hum daalte hein usi sequence mein ye code work karte hein.
2. Suppose pehle external css ka code likha huya he and uske baad internal css ka code likha huya he toh jo last wala internal css wala code consider hoga.
3. Agar humein kisi code ko jo jaruri run karna hota he usko hum (!important) wale tag se run kar sakte hein or jo bi style humko dena hota he usko de sakte hein.

**#Selectors:**

1. Ye ek esa tareeka he jiske dwara hum kisi bi HTML element ko target kar sakte hein or usko style kar sakte hein.

There are 4 types of selectors in HTML:

1. Element
2. ID
3. Class
4. Group
5. **Element Selectors**: In this we use the whole tag as selector to assign a property to the element.

For eg: <style>

p{

    color:cadetblue

}

</style>

    <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Consequuntur adipisci nulla reprehenderit illum aliquam ipsa eligendi assumenda vitae quaerat earum.</p>

Here <p> is the selector because with the help of this tag we can assign the styling property to the whole code wherever we have used <p> tag.

1. **ID selector:** Lets say aapne kisi element ko id assign kar rakhi he toh us id ko use karke hum code ko style kar sakte hein.
2. Isme ek rule hota he ki id se phle aapko ek (#) dena hota he.

**For eg:** #article1{

    background-color:coral;

}

    <p id="article1">Lorem ipsum dolor, sit amet consectetur adipisicing elit. Consequuntur adipisci nulla reprehenderit illum aliquam ipsa eligendi assumenda vitae quaerat earum.</p>

1. **Class Selector:**  Humlog multiple elements ko same class assign kar sakte hein, so agar hum class selector ka use karte hein toh hum class element ko lekar style assign kar sakte hein or jis jis ki class same hogi usko style kar sakte hein hum.
2. Isme ek rule hota he ki class se phle aapko ek (.) lagana padta he then hum style attribrute assign karte hein.

**For eg:** .paraclass{

    background-color: crimson;

}     <h1 class="paraclass">Welcome to the Codehelp!</h1>

    <p class="paraclass">Lorem ipsum dolor, sit amet consectetur adipisicing elit. Consequuntur adipisci nulla reprehenderit illum aliquam ipsa eligendi assumenda vitae quaerat earum.</p>

1. **Group Selector:**  Group selectors mein hum multiple tags ko comma(,) lagakar select kar sakte hein or usko style property de sakte hein.

**For eg:** p,article,h1{

    width: 500px;

}     <h1 class="paraclass">Welcome to the Codehelp!</h1>

    <p class="paraclass">Lorem ipsum dolor, sit amet consectetur adipisicing elit. Consequuntur adipisci nulla reprehenderit illum aliquam ipsa eligendi assumenda vitae quaerat earum.</p>

1. Hum log specific tag ki class ya id ko bhi group attribrute ki madad se style attribrute assign kar sakte hein.

**For eg:** p.paraclass{

    align-content: center;

}

    <p class="paraclass">Lorem ipsum dolor, sit amet consectetur adipisicing elit. Consequuntur adipisci nulla reprehenderit illum aliquam ipsa eligendi assumenda vitae quaerat earum.</p>

**#Fonts&Colors:**

1. Font-size: With this we can decide the size of the text or word how big or small it will be.

           font-size: 20px;

1. Font-style: With this we can decide how the font will look like for eg: italic, bold,
2. font-style: italic;
3. Font-weight: This will tell how much the thickness or thinness of the text or will decide the degree of blackness.

For eg:             font-weight: 800;

1. Line-height: This will tell how much will be the gap between two words, or how much will be the gap height between two words.

For eg:             line-height: 90px;

1. Font-family: Particluar letter ko render karne ka tarika alag hota he wo aap yahan se kar sakte hein. Font-family mein fallback system hota he agar koi font available ni he toh dusra le le uski jagah.

For eg:             font-family: Cambria, Cochin, Georgia, Times, 'Times New Roman', serif;

1. **We can also make our own fonts and import fonts from external site:**
2. In css we can make our own fonts or import fonts from external site,

Lets suppose you like a font from external site and you want to use it in your own site then you can do this, lets go to google fonts and clink on the desired font go to copy link and then add the above font in the head section of the code.

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=Lobster&display=swap" rel="stylesheet">

1. **Colors:**

We can add colours to our code in 4 ways:

1. **Named colors:**  This can be done by giving the colour name simply in the style tag in bg-color or text color.

For eg: #Firstpara{ background-color: yellow;

            color: brown;

        }

    <p id="Firstpara">Lorem ipsum dolor sit amet consectetur adipisicing elit. Consectetur temporibus nesciunt blanditiis eum iste esse aliquam deserunt nulla quis minus.</p>

1. **Rgb values:** We can assign colour to the code by simple giving the rgb values. In this the value ranges between ( 0-255) and rgb stands for Red, Green, Blue.

**For eg:**             background-color: rgb(0, 0, 0);

1. This means that red ka component 0 , green ka component 0, and blue ka component bi 0.
2. #Thirdpara{
3. background-color: rgba(39, 49, 184, 0.5);}
4. In rgb values the rgba component there is red, green, blue, alpha component where Alpha component decides what will be the transparency level of the color. Alpha ke saath hum transparency ka level set kar sakte hein.
5. **Hexadecimal Values:** In html colors can also be assigned with the help of hexadecimal values.

**For eg:** .fourthpara{

   background-color: ffffff;

        }

In this pehle do ff – red , dusre do ff- green, teesre do ff- blue ka component decide karte hein.

Isme hexadecimal values dene se pehle hum log # lagate hein.

1. **HSL Value:(Hue Saturation Level)**
2. Hue is a degree on the color wheel from 0 to 360. 0 (or 360) is red, 120 is green, 240 is blue.
3. Saturation can be described as the intensity of a color. It is a percentage value from 0% to 100%.
4. The lightness of a color can be described as how much light you want to give the color, where 0% means no light (dark), 50% means 50% light (neither dark nor light), and 100% means full light.

For eg: .fifthpara{

            background-color: hsl(0, 50%, 50%);

        }

**#Box Models:**

1. **To add border in HTML:**

For eg: h1{

            background-color: aqua;

            border-style: solid;

            border-width: 2px;

            border-color: coral;

        }

1. **Shorthand-notation:**
2. Ek hi field ke andar multiple values specify kar sakte hein.

For eg:         h1{

            border: solid brown 15px;

1. Border-radius: Ye humko jo border ki golayi hoti he wo set karne ke kaam aati he, or corners ko rounded kr skte hein.

For eg:             border-radius: 20px;

This we will need more while creating buttons.

If we want to make only one radius rounded then we will use border-radius top-left=”5px”.

For eg:             border-top-left-radius: 5px;

If we want to make bottom border only change. Then

For eg: border-bottom:10px;

**#Width in HTML elements:**

1. width: 500px;
2. With this we can gave width to html elements as per our need.

**#Height In HTML content:**

        height: 100px;

With this we can gave height to html elements as per our need.

**#** **Why on span tag height and width doesn't work:**

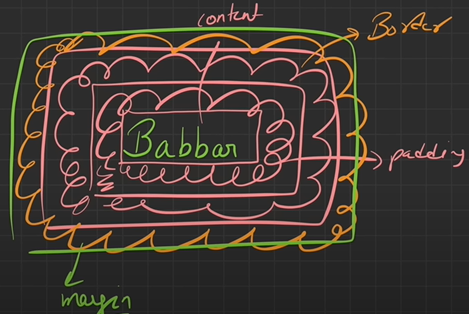
Inline elements, such as a <span> , will completely ignore the width and height as well the the top and bottom margin properties because, well, the content is what determines the dimensions.

**#CSS Box Model:**

1. Jab bhi aap koi website design kar rhe hote hein yap hir browser par kuch bhi display kara rhe hein toh har ek cheez jo bhi page par he wo ek box ke form mein render(display) hongi ya phir ek rectanglular shape ke andar render hogi.

Box model consist of 4 parts:

1. Content
2. Padding
3. Border
4. Margin



**#Padding:**

1. Content or border ke beech mein jo space hota he use padding kehte hein.
2. Padding zero bhi ho sakti he.

For eg:             padding: 20px;

1. Ese likhte hein hum jab padding toh wo chaaron direction mein apply hota he.

For eg:             padding-left: 10px ;

            padding-top: 20px;

            padding-right: 30px;

            padding-bottom: 40px;

1. We can add padding specific to the direction also, suppose for right,left,top,bottom to.
2. We can add padding for x axis and y axis too. Jo isme pehli value hoti he wo top or bottom padding batata he or jo dusri value he vo right or left padding batata he.

For eg: padding: 20px 40px;

**#Margin:**

1. Do boxes ke beech ka jo gap he use hum margin kehte hein.

For eg:             margin: 50px;

Ese likhte hein hum jab padding toh wo chaaron direction mein apply hota he.

1. Margin humlog specific direction mein bi add kar sakte hein.

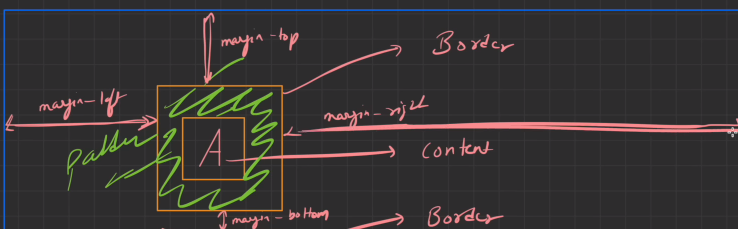
For eg: margin-top: 10px;

            margin-bottom: 20px;

            margin-right: 30px;

            margin-left: 40px;

Reference:



**#Best Code Practices:**

1. Jab bhi hum log ache codes ki bata karte hein to usme by default margin or padding ko zero rakhte hei.

**#Universal Selector:**

1. Jitna bi humara code hein(elements,tags) us sabme ye properties apply hoti hein use hum universal selctor kehte hein.
2. Isko humlog \*{} se represent karte hein.

For eg:         \*{

            margin: 0;

            padding: 0;

        }

**#CSS Display Property:**

1. Display property ke dwara kisi ek element ko display par ya viewport ya webpage par kaise render karna he ya visible karna he wo darshaya jaata he, wo logic hum is property ke andar define karte hein.

**#Block Element Property:**

1. Block element for eg:
2. div
3. <p>
4. <h1> etc
5. **Block elements ki properties**:
6. They always comes in new line.
7. Width – Extreme left -> Extreme right, Ye pura line cover karte hein extreme left se extreme right.
8. Custom Width/Height – Hum log block elements mein custom width or height assign kar sakte hein.
9. Custom Margin/Padding – Hum log block elements mein custom margin or padding bi assign kar sakte hein.
10. 10 different block elements example:

<html>

<body>

<para>

<pre>

<fieldset>

<hr>

<headings>

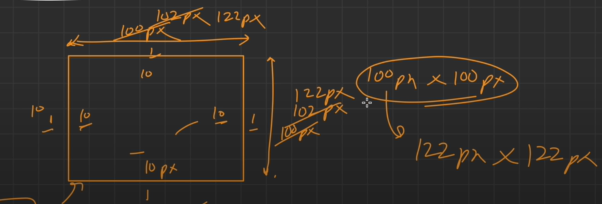
<ul>

<ol>

<form>

**#Box-Sizing 🡪 Border-Box**

1. Without border box ke aapke box ki jo actual width hota he wo increase ho jaata he.

For eg: 

1. Is box mein 1px-1px ki custom width or height dedi he humlog ne toh total dimension of box 102-102 px ho jaati he.
2. Ese hi hum log ne 10px-10px ki margin or padding de dete hein toh actual box ki dimension 122px-122px ho jaati he.

**#Conclusion:**

1. Box size or border box ka use karne se actual jo box he apna size retain karta he, uska dimension change ni hota srf content shrink hota he.
2. Box size or border box ka use nah karne se actual jo box he wo apna size ni retain kar paata he, uska dimension change ho jaata he or content expand karta he.

For eg:     .box{

        width:300px;

        height:300px;

        border-style: solid;

        border-color:blueviolet;

        border-width: 2px;

        padding: 100px;

    }

#After applying border-box:

    \*{

        margin:0;

        padding:0;

        box-sizing: border-box;

    }

**#Properties of inline elements:**

1. They always comes in same line.
2. Ye pura line ka space ni lete hein jitna humara content hota he bas utna hi space lete hein.
3. Custom width or height isme apply ni hota he.
4. Custom margin isme srf horizontally work karta he isme, custom margin horizontally work ni karta he.
5. Custom padding isme apply kar sakte hein hum.

For eg: span{

        border-style: double;

        border-color: aquamarine;

        border-width: 2px;

        width: 400px;

        height: 300px;

        margin: 50px;

        padding:100px;

    }

**#Difference between block and inline elements:**

**Block elements Inline elements**

a. New line. Same line.

b. width🡪 EL 🡪ER width🡪 Content.



c. Custom W/H - Custom W/H -



d. M/P 🡪 all directions Margin 🡪only horizontal , padding🡪all dir padding 🡪 all directions.

**#Conversion of Inline Elements To Block Elements:**

Set: display = block

OR

**#Conversion of Block Elements To Inline Elements:**

Set: display = inline

**\*\*List of 10 inline elements:**

button,

input,

label,

a,

bdo,

br,

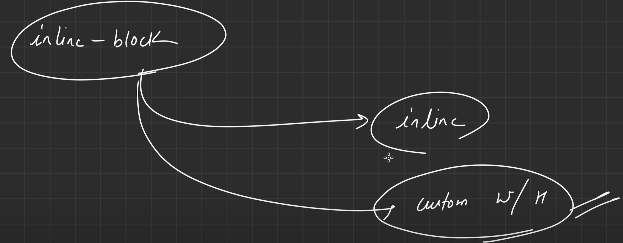
img,

map,

object

**#Inline-Block elements:**

1. This property behaves as inline elements but we can give custom-properties of block elements also.

For eg: 

        display: inline-block;

1. So in inline-block elements we can give custom width and height properties.

**# CSS Position-Property:**

1. With this property we can position different different elements on the screen like top se kitna dur hoga content, bottom se kitna space hoga, left or right se kitna space hoga, konsa content kiske upar overlap hoga etc ye sab decide karenge.

For eg: Lets say humlog ne 4 images bna di unko kaise or kahan position karna he ye css-position property batata he.

1. By default jo position property apply hoti he wo static hoti.

**\*\* Do images or content ke beech ka gap hum log margin property se decide karte hein.**

**# Position-property:** Relative

1. Jab humlog relative position property ka use karte hein toh wo current window ke related karte hein.
2. Hum log isme spacing top, left, right, bottom kar sakte hein.
3. Jis image ya content ko hum move karte hein uska gap maintained rehta he ya uski position intact rehti he.
4. Relative position ye hota he jo humari current window he uske related us window ke corresponding hum kisi bi element ko position kar sakte hein using top, left, right and bottom property.

For eg:         .Image1{

            position: relative;

            top: 100px;

            left: 200px;

            bottom: 300px;

            right:400px;

        }

**#Absolute Positioning:**

1. Jab bhi humlog ko overlapping karwani hoti he tab hum use karte hein absolute positioning ka.

For eg: .Image2{

            position: absolute;

            top: 100px

        }

1. Absolute positioning mein gap maintained ni hota he.
2. Absolute positioning is relative to its current positioning window.
3. This can be positioned to left, right, top and bottom.

**#Fixed-Postioning:**

1. Jab hum fixed positioning apply karte hein toh wo element fixed ho jaata he chaahein kuch bhi karlo wo apni position se nhi hatti he.
2. Fixed wali image ya element pure window mein fixed rahegi chaahein kitne bhi container bane ho.

**For eg :** .Image4{

            position: fixed;

        }

**#Sticky-Positioning:**

1. Jab hum sticky positioning apply karte hein toh element apni position par stick ho jaata he.
2. When we do sticky positioning then we have to give the target position like (top,left,right,bottom) corresponding to which the element will be sticked.
3. Sticky positioning tab tak hi rehti he jab tak wo apne parent container mein rehti he agr wo parent container se bahar gyi to wo fixed ni rehti he.
4. Aapne koi element pakda or kisi spot par Rakha(top,left,right,bottom) phir hum usko sticky mark kar sakte hein.
5. .Image3{
6. position: sticky;
7. top: 100px;
8. }

**# CSS Size-Units:**

1. Jo percentage hum use kar rhe hote hein wo humare immediate parent ke upar depend kar rhi hoti he ki kitni width,height ya or kuch show karna he.

For eg: .parent{

            border: 1px solid black;

            background-color: beige;

            width: 50%;

        }

        .child{

            border: 2px solid red;

            background-color:aqua ;

            width: 20%;

        }

    <div class="parent">

    <div class="child">

        Child

    </div>

    </div>

1. **.vh dimensions:**
2. If you want to give the dimensions of your viewport to full viewport height then we can give by this unit.
3. .container{
4. background-color: beige;
5. border: 2px solid red;
6. height: 100vh;
7. }
8. <div class="container">
9. Welcome to codehelp
10. </div>
11. So it will give the height of the container on whole screen equal to the size of viewport.
12. Similarly we can give the width to the viewport also in (vw) format.

For eg:             width: 50vw;

**#Pixel unit:**

1. Difference between logical pixel and physical pixel

Logical pixels are defined as the number of physical pixels in a device's screen divided by the CSS pixel ratio, and logical pixels are what you see when you look at your device (and more importantly, what your browser sees).

1. Difference between ppi and dpi

DPI refers to the number of printed dots contained within one inch of an image printed by a printer. PPI refers to the number of pixels contained within one inch of an image displayed on a computer monitor.

1. Pixel is also an unit for descripting the size for any element, with this we can increase or decrease the size of any object or element.

For eg: .box1{

            background-color: bisque;

            border: 2px solid black;

            font-size: 100px;

        }

**#Absolute Units:**

1. Em unit humare parent par dependant hoti he jo bi uska font size hoga, jo child hota he seedhe seedhe wo parent unit se multiply ho jaati he.

**For eg:**         .parent{

            font-size: 10px;

        }

        .box{

            font-size: 10em;

        }

Isme jo parent ka size he 10px wo sidhe child wali 10em\*10= 100px se multiply hokar size ho jaata he.

**#REM Unit:**

1. Ye unit humare root element ke size ke upr dependant hota he,

Root element ka by default size 16px hota he.

1. Jo bi property aapka html element show kar rha hoga uske according humari dimension set ho jaayegi.

For eg:         .Jee{

            border: 1px solid black;

            font-size: 5rem;

**#Float property in html:**

1. It is used for positioning and layout on web pages.
2. A common usage is floating an image to one side and letting text wrap around it.

**For eg:**         #img1{

            float: right;

            height: 400px;

        }

        #img2{

            float: left;

            height: 400px;

        }

**#Clear keyword:**

1.The clear property specifies what should happen with the element that is next to a floating element.

2. It can have values : none|left|right|both|initial|inherit;

For eg:         #img1{

            float: right;

            height: 400px;

        }

        #img2{

            float: right;

            height: 400px;

            clear: right;

        }

**#The great collapse:**

1. It is a scenario in float situation where the parent box collapses when the elements placed inside it are floating in the same direction.
2. To overcome this situation we can do:

**Overflow: auto**

For eg:         .box{

            border: 1px solid black;

            overflow: auto;

        }

.item{

            width:200px;

            height: 200px;

            margin:2px;

            background-color: aquamarine;

            border: 1px solid black;

            float: left;

**#Overflow in css:**

1. The CSS overflow property controls what happens to content that is too big to fit into an area.
2. The overflow property specifies whether to clip the content or to add scrollbars when the content of an element is too big to fit in the specified area.

Toh jis prakar se aapka content apne parent element se bahar nikal jaata he us situation ko hum overflow situation kehte hein.

For eg:         .container{

            border: 1px solid black;

            width: 500px;

            height: 500px;

            overflow: scroll;

        }

In this situation the content that is coming outside the container will not be gone and we can scroll to show the full content.

They can be of several types for eg: hidden,auto,scroll.

**#Flexbox:** Ek bahut hi powerful tool he jisse hum different types ke layout create kar sakte hein.

1. To make a box, container, or element a flexbox we have to set

Display=flex

1. Jab bhi hum kisi container ko flex mark karte hein to uski by default direction row hoti he.
2. With the help of flex-direction property we can decide the direction of the flex box. It can be of four types:
3. Flex-direction: row
4. Flex-direction: row reverse
5. Flex-direction: column
6. Flex-direction: column reverse

For eg:         .container{

            background-color:beige;

            border: 1px solid black;

            border-width: 2px;

            margin: 2px;

            padding: 2px;

            display: flex;

            flex-direction: column;

        }

**#Flex-wrap:** The **flex-wrap** [CSS](https://developer.mozilla.org/en-US/docs/Web/CSS) property sets whether flex items are forced onto one line or can wrap onto multiple lines. If wrapping is allowed, it sets the direction that lines are stacked.

1. If you don’t want the content to be squeezed and make it auto adjustable according to the screen until the size is available.
2. By default property is no-wrap.
3. It can be of two types:
4. Wrap- It wraps the element until there is space left.
5. wrap-reverse: It wraps the element in reverse direction.

**#Flex-flow:**

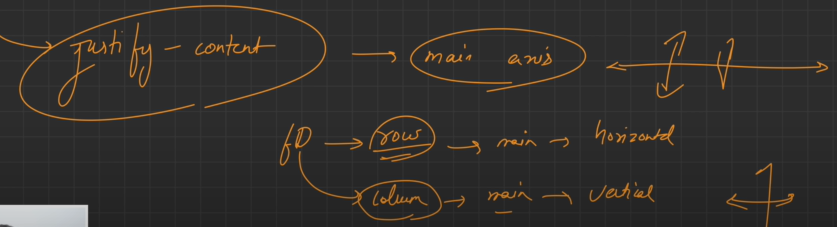
The **flex-flow** [CSS](https://developer.mozilla.org/en-US/docs/Web/CSS) [shorthand property](https://developer.mozilla.org/en-US/docs/Web/CSS/Shorthand_properties) specifies the direction of a flex container, as well as its wrapping behavior.

For eg: flex-flow: row-reverse nowrap;

flex-flow: column wrap-reverse;

**#Justify-content:**

1. Justify-content ek esi property he jo ki contents ko main axis ke hisaab se place karta he.
2. Flex-direction🡪row🡪main axis🡪horizontal.(is direction mein elements place karta he).
3. Flex-direction🡪column🡪main axis🡪vertical. .(is direction mein elements place karta he).

****

1. **Justify-content: Center**
2. Isme elements horizontal axis ke corresponding centre mein placed ho jaayenge.

For eg:  justify-content: center;

            height: 500px;

1. **Justify-content: Start**
2. This will place the elements at the starting of the website corresponding to horizontal axis.
3. justify-content: start;
4. **Justify-content: End**
5. This will place the elements at the end of the website corresponding to horizontal axis.
6. justify-content: end;
7. **Justify-content: Space-between**
8. In this scenario pehle wala box and aakhiri wala box aapki boundary se sata huya hota he or beech wale box apas mein equal distance par hote hein.
9. justify-content: space-between;

**5.Justify-content: Space-around**

a. Isme har ek dabbe ke left or right mein ek equal space assign kar dete hein.

For eg:             justify-content: space-around;

**6.Justify-content:Space-evenly:**

a. Isme hard dabbe ke left or right mein equal space hota he.

For eg:             justify-content: space-evenly;

**#Flex-direction: Column**

**a.Justify-content: center**

1. Isme elements vertical axis ke corresponding centre mein placed ho jaayenge.
2. flex-direction: column;
3. justify-content: center;

**#Align-Items:**

1. Align-items hamesa humare cross-axis ke corresponding alignment karni hoti he.

**Foreg:** flex-direction:row;

        /\*justify-content: space-between; moves the content row horizontal wise \*/

        /\*align-items:center; moves the content column vertical wise \*/

        align-items:stretch;

        display: flex;

**\*\*VImp:** (Difference between justify-center and align items):

1. justify-content — controls alignment of all items on the main axis(columns)
2. align-items — controls alignment of all items on the cross axis.(rows)

**#Align-items(stretch):**

1. Align-items work according to the cross-axis which is the vertical axis and is called rows.
2. By default is we are using (align-items: stretch;) then if we have given hard-coded height we have to remove it because by-default align-items is stretch only.

**#Element placed at center:**

1. If we want to place the element just at the centre of the website then we use the following tag.

For eg: align-items:center;

        justify-content: center;

**#Align-Content:**

1. Jab hum wrap on karte hein or humein multiple contents ke beech mein spacing define karni hoti he tab hum align-content ka use karte hein.
2. Is property ki help se hum iske beech ke space se kam kar sakte hein.

For eg: align-items:start;

        justify-content:start;

        flex-wrap: wrap;

        align-content: start;

**#Align-content: end**

1. This will place the content at the end of the website.

For eg:         align-content: end;

**#Align-content: center**

1. This will place the content at the center of the website.

For eg:         align-content: center;

**# Properties of Flex-Items:**

1. **Order:** With this property we can set the order of the flex items placed in a website.

For eg:  #box2{

        background-color: blueviolet;

        width: 100px;

        height: 100px;

        border: 2px solid black;

        margin: 2px;

        order:-3;

1. With this property we can decide the sequence of the element.

**#Flex-shrink:**

1. With this property we can decide how much fast or slow the flex-item will shrink.

For-eg:     #box1{

        background-color: aqua;

        width: 100px;

        height: 100px;

        border: 2px solid black;

        margin: 2px;

        order: 500;

        flex-shrink: 4;

This will shrink the first box with 4x speed.

**#Flex-Grow:**

1. At first most with this property flex-items will grow and take the whole available space inside the container.
2. This property will decide how fast the flex-item or element will grow and take the available space in the container.

For eg:  #box2{

        background-color: blueviolet;

        width: 100px;

        height: 100px;

        border: 2px solid black;

        margin: 2px;

        order:-3;

        flex-grow: 5;

**#Flex-basis:**

1. This is used to set the initial dimension of the flex-item.
2. This will set the initial size of the flex-item.

For eg:     #box1{

        background-color: aqua;

        width: 100px;

        height: 100px;

        border: 2px solid black;

        margin: 2px;

        order: 500;

        flex-basis: 400px;

This will set the initial dimension of the flex-box 1 to 400px.

**#Align-Self:**

1. This property decides how the item or element aligns himself.
2. By default you are aligning the items across the vertical axis.

**For eg:**     #box2{

        background-color: blueviolet;

        width: 100px;

        height: 100px;

        border: 2px solid black;

        margin: 2px;

        order:-3;

        align-self: start;

**This will align the box2 in the starting of the website.**

**#Media-Query:**

1. Website ko jab hum grow or shrink kar rhe hein ya different- different websites par hum usko jab view karte hein toh wo website break ni hota he usi website ko hum responsive kehte hein.
2. Media-Query ek esa tool he jisse hum kis display-size par element ko kaise place kar sakte hein wo decide karte hein.

For eg:         .box{

            background-color: aqua;

            width: 400px;

            height: 400px;

            border: 3px solid black;

        }

        @media (min-width:500px){

            .box{

            background-color: green;

            }

        }

In above example we have set the bg-color of box1 to green when the screen resolution is above 500px and as soon the display size is below 500px it will turn blue.

**#Shadows in CSS:**

1. Box-shadow: The **box-shadow** [CSS](https://developer.mozilla.org/en-US/docs/Web/CSS) property adds shadow effects around an element's frame. You can set multiple effects separated by commas. A box shadow is described by X and Y offsets relative to the element, blur and spread radius, and color.

For eg: box-shadow:             box-shadow: 10px 10px rgb(201, 130, 7);

.box{

            background-color: aquamarine;

            width: 330px;

            height: 330px;

            margin:5px;

            padding: 5px;

            border: 2px solid brown ;

            box-shadow: -10px 10px rgb(201, 130, 7);

        }

The first value is X offset, Second value is Y offset, Third value is blur and fourth is the spread radius.

1. Box-shadow using blur value: This will tell how much will be the blur value in the tag.

For eg:             box-shadow: 10px 10px 10px orangered;

1. Box shadow using spread radius: This will tell how much will be the spread radius of the blur value in the tag.

For eg:             box-shadow: 10px 10px 10px 20px orangered;

1. Box-shadow-inset: This will bring the shadow inwards.

For eg:             box-shadow: inset 10px 10px 10px 20px green;

1. You can also add multiple shadows in the same box by giving (,)

For eg:             box-shadow: 10px 20px red, 30px 40px green;

**#Custom-Variable:**

1. To create a custom variable we follow:

Syntax: --primary-color: brown;

--custom-width:330px;

And we call this variable with the help of:

Width: var(--custom-width);

Background-color: 10px 20px var(--primary—color);

For eg:         .box{

            --primary-color:blue;

            --custom-width:330px;

            background-color: aquamarine;

            Width: var(--custom-width);

            height: 330px;

            margin:5px;

            padding: 5px;

            border: 2px solid var(--primary-color) ;

            box-shadow: 10px 20px red, 30px 40px green;}        }

**\*\*Important-point(Custom-Variable):**

1. This custom variable will work within those brackets only, if we want to use it on any other place we have to create it again.
2. If we want to use the property or color accessible to all the elements then we have to define that property in the root element.

For eg:             :root{

                --primary-color:blue;

                --custom-width:330px;

            }

            Width: var(--custom-width);

            border: 2px solid var(--primary-color) ;

**#Text-Shadow:**

1. Text-shadow has all the same properties as box-shadow except the spread radius.

For eg: h2{

            text-shadow: 10px 10px 2px blueviolet;

        }

**Different ways of same syntax in text-shadow:**

1. text-shadow: yellowgreen 10px 10px 3px;
2. text-shadow: 10px 10px;

**#By default shadow colour is black.**

**#Animations in CSS:**

1. Whenever an element or component shift from one styling to second styling, this process is called animation.
2. **Animation-name:** In this tag we give the name of the animation and then we call the animation- name tag with the help of @keyframe tag and tag name. After this we give the from(source) and to(source) till where the animation is to be given.

**Syntax:**  animation-name: rightmovement;

        }

        @keyframes rightmovement{

            from{

                top: 0;

                left:0;

            }

            to {

                top:0;

                left: 1200px;

            }

        }

1. **Animation-duration:** This tag will tell you how much time or duration will the animation exist. From one place to another how much time will the element take is given by animation-duration.

**For eg:**             animation-duration:5s ;

1. **Animation-iteration-count:** Kitni baar animation karwana chahhte ho wo is tag se decide karte hein.

**For eg:**             animation-iteration-count: 2;

1. **Animation-delay:** Animation start hone se kitna delay karwana chhahte ho wo ye tag decide karta he.
2. animation-delay:2s;

**\*\*Vimp:**  It is necessary to give the position of the element such as relative, fixed, absolute so that we assign top, left, bottom or down to that element.

            position: relative;

**#Animation-timing-function:** The **animation-timing-function** [CSS](https://developer.mozilla.org/en-US/docs/Web/CSS) property sets how an animation progresses through the duration of each cycle.

* **ease:** With this property value, the animation starts slowly, then fast, and then finally ends slowly (this is default).
* **linear:** If this value is specified for the property then the animation plays with the same speed from start to end.
* **ease-in:** If this value is specified then the animation begins with a slow start.
* **ease-out:** If this value is specified for the property then the animation plays normally but ends slow. This is similar to ease-in.
* **ease-in-out:** With this property value, the animation both starts and ends slow.

For eg:             animation-timing-function: linear;

**#Animation-direction:** We can control the direction of the element with this tag for eg: to and fro direction can be given with this.

For eg:             animation-direction: reverse;

**#Animation-fill-mode:** With this tag we will tell ki first frame se aage or last frame ke peeche kya karna he ye hum fill mode se decide karna he.

For eg:             animation-fill-mode: both;

**#Animation-Play-State:** Play state se aap apni animation ko pause kar sakte hein.

For eg:             animation-play-state:running

**#In this type also we can give animation ratio:**

            0%{

                top:1%;

                left:5%;

            }

            30%{

                top:5%;

                left:90%;

            }

            60%{

                top:70%;

                left:10%;

            }

            100%{

                top:75%;

                left:90%;

            }

1. This 0% tells of the animation time.

**#Transitions in CSS:**

1. Ek state of styling se jab hum dusre state of styling par switch ya move karte hein use hum transition kehte hein.
2. Transition process ek single stage mein hota he.
3. Transition dekhne ke liye humein hover karna padega ya ek pseudo class define karna padega.

\*\*Animation mein series of movements hote hein jabki transition mein ek single stage mein movement hoti he.

Transition animation ki ek child entity hum maan sakte hein.

1. Transition-property: all;

This will apply transition to all the properties that are defined inside the transition-element.

1. Transition: all 1s 2 s ease-in;

This is a shorthand notation to give all the properties in a single line.

We can also give multiple properties by separting commas.

For eg: transition: width 2s, height 2s;

**#Transform in CSS:**

1. It is the process where an element, div or box can be transformed from one state to another, for eg: rotated, size changed, movement of object, scaling of an object etc.
2. To apply smooth movement of box or elements we use transition property whereas to apply change of state from one state to other we use the transform property.

For eg: .box:hover{

    transform: rotate(360deg);

}

Same is in transition we have to hover over the box or element to make changes in the box.

1. Transform: Rotate(45deg) : This will make the box rotate by 45 degrees.
2. Transform: rotate(-45deg) : This will make the box rotate 45 degrees but in negative direction.
3. Transform: Scale(2) : This will increase the size of the box two times bigger than the original content. With this property we can increase or decrease the size of the box.
4. Transform: Scale(0.5): this will decrease the size of the box to half.
5. Transform: skex(45deg) : This will tilt the box by 45 degrees.
6. Translate: This property tells how the box will move left, right, inside, outside. It has two parameters which will tell how the box will move in x axis and y axis.

For eg: .box:hover{

    transform: translate(0px, 500px);

}

This will move the box 0 in x axis and 500 px down on Y axis.

# To remove underline from anchor-tag we use text-decoration property.

a{

    text-decoration: none;

}

#Kisi bi cheez ko humko overlap karwana he toh humlog position absolute.

#Kisi bi element ko transparent ya opaque karne ke liye hum opacity property ki use kar sakte hein.

For eg: opacity:1;

**Javascript Notes**

1. To declare a variable in js we use (let) and (var) keyword.

For eg: let a = 26;

1. Whenever a variable is declared with (let) keyword it is having block scope and can’t be used globally.
2. Whenever a variable is declared with (var) keyword it is having global scope can be accessed within the block and outside the globe too.
3. Var can be redefined or changed while changing the value of let it will throw an error.
4. Const can neither be changed nor be redeclared.

We will use const when the indentifier value will not be changed.

#Object And Primitive Data types:

1. Objects are key value pairs where keys can be any number, string etc and values can be any data types. It is a not primitive data type.
2. Primitive data types are of 7 types:

Eg: NN SS BB U - null, number, string, symbol, Boolean, BigInt, Undefined.

1. When we try to add two things in a string then both the things are concatenated.
2. We cannot change the value of a const. variable but if we make a const. object and give it a reference then we can change the values inside that object but if we try to assign any other value to that const. object it will throw an error.

For eg: const a1 = {

    name : "Ankit",

    section : 1,

    isPrincipal : false

}

 a1["name"] = "Shivam"

 a1["section"] = 2

 a1["isPrincipal"] = true

//  a1["name"] = "lovish"

 console.log(a1)

 a1 = {}

 console.log(a1)

Error: JS1.js:31 Uncaught TypeError: Assignment to constant variable.

**# Expressions and Operators in JS**

1. Expression is a value in JS which you can assign to a variable, constant.
2. For converting a string to a number we use (parseint) method.
3. (prompt) method will always give us a string as output.
4. Prompt humse sawaal puchta he or alert humko batata he.
5. Switch or case statement mein jo bhi condition true hoti he wo execute hoti he or uske baad ka sara code execute ho jata he.
6. For- in loop loops through the object keys.
7. For-of loop loops through the object values.
8. Do-while loop is a type of loop which will execute its body atleast once.
9. In do-while loop first the block runs then the condition is checked whereas in while loop first the condition is checked and then the loop runs.
10. Function ek blackbox he jo ki input leta he or phir us condition ko analyse karke output deta he.

**#Important Concept of Javascript Functions:**

1. Jab bhi hum koi cheez return kara dete hein uske baad kuch bi print ni hoga.
2. Jab bi aap koi value return karte hein toh aap usko kisi variable mein store kar sakte hein srf console.log se ni hogi.

For eg: function addtwonumbers(number1,number2){

    let result = number1 + number2

    console.log("Hitesh")

    return result

}

const result = addtwonumbers(3,5)

console.log("Result : ", result)

1. To print all the variables in a table format we use, console.table([]) keyword.

For eg: const var1 = "Harry"

let var2 = 1223

var var3 = "Sinku"

var4 = "AMdhsk@gmail.com"

console.table([var1,var2,var3,var4])

1. “use strict” – When we type this keyword in the starting of the code then my js code will be treated as newer version of JS.
2. “33” = 33 can be converted to a number with keyword (Number) in Js.
3. But when we convert “33abc” = NaN, it can be converted to a number but will give a value Number not Available.