

## \* Web Technology

1) HTML

2) CSS

3) Java Script

## \* Applications

1) Applications are the set of software programs which is used to perform some task or functionality to the users and the functionality will be different for different users.

### \* Types of Applications:

#### 1) Desktop application:

- \* These applications are ~~relat~~ limited to the particular system or desktop and data transfer is not possible in case of desktop application.
- \* We can access desktop application without internet connection.
- \* eg :- MS Paint, MS powerpoint, Excel, etc...

#### 2) Web Application:

- \* These are the applications which we can access only through browsers with the help of an internet.
- \* We can transfer the data with the help of web application.
- \* Using With the help of web technology we are going

to create only web application.

eg:- Facebook, Instagram, flipkart etc.

### 3) Mobile application:

\* These are the applications which is available on small computing devices such as smartphone, tablets, using mobile.

\* Using mobile application we can access without internet, but in order to transfer the data we should connect it with the internet.

\* eg:- calendar, contacts, calculator etc.

### \* Terminologies

1) Web → Interconnection of data

2) Internet → Intermediate / Interface between two media.

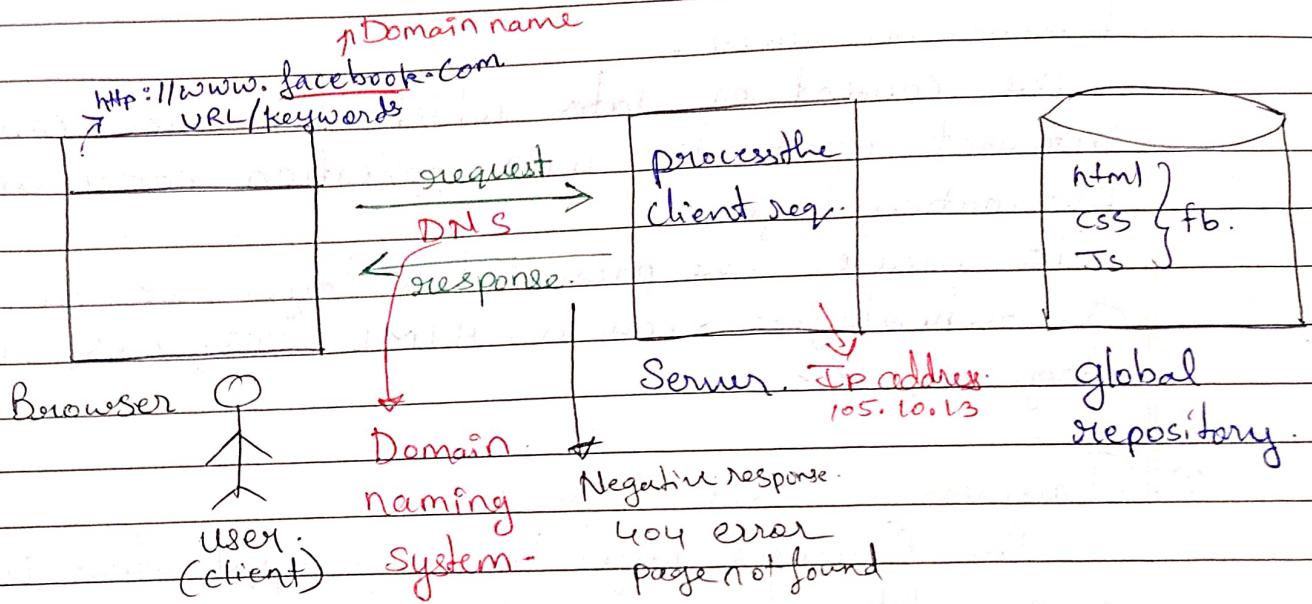
3) Browser → Software application which is used to access the Web pages.

4) Server → Software, process the client request and send back the response.

5) Web page → document written in hypertext (HTML)

6) Website → Collection of web pages

## \* How Web Works?



- \*) When we will send the request to server through the URL the server will process the client request. That is the server will check it in the global repository for the respective web page HTML file.
- \*) If the files are there in the global repository then it provides positive response and if the files are not present in the global repository then it provides the negative response that is 404 error / page not found.
- 3) Server understand only IP address not the domain name.
- 4) There is an intermediate call DNS (Domain naming system) which converts domain name into ip address when we are sending the request to the server.

## \* Different types of layers in web application

### 1) Presentation layer :

The content or data which the user can interact with the web application comes under the presentation layer.

To create the presentation layer we are using the technologies such as HTML, JS, CSS, react, Angular, vue, etc.

### 2) Business / Logic layer :

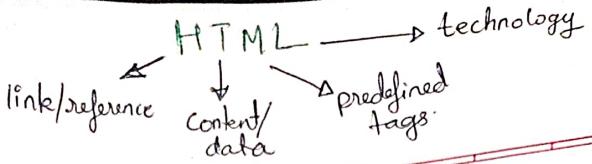
The Business layer in web application is responsible for processing business rules, executing the logic and provide the functionalities and also acts as the intermediate between presentation layer and Database layer.

The technologies which we are using are JavaScript, Java, Python, PHP, etc.

### 3) Persistence / Database layer :

It is used to store the data into the database.

Some of the examples for database are MS SQL, Oracle SQL, Mongo DB.



Date \_\_\_\_\_  
Page \_\_\_\_\_



## HTML

- Creating a reference or link to visualize the content or data to the users with the help of predefined tags we are using the technology called as HTML which is abbreviated as "Hyper Text markup Language".
- HTML is used to create the structure of web application.
- We can create the HTML file using the extension either .html (Preferred) or .htm
- Without html file we cannot create any web pages.
- Cmd is the command to open the command prompt and code space. is the command to open the VS code.
- Within the HTML file we can take the structure of HTML using by two ways with ! or html:Boilercode.

\* DocType declaration :- (`<!DOCTYPE html>`)

This is the first statement in a html file which defines the version of html.

The current html version we are using is html:5

\* HTML tag: (`<html></html>`)

It is called as root element because inside html tag we are going to write the entire html code.

html tag takes one attribute that is lang which defines the language of the web page. and the value should be the language code.

There are two children of html tags

1) Head

2) body.

\* Head tag : (`<head></head>`)

We should pass the information related to the web page within the head tag.  
The children of head tags are meta, title, style, etc..

\* Body tag : (`<body></body>`)

We should pass the content or data which we want to display into the web page within the body tag.

\* Title tag :

It is used to give the title to the web page.

\* meta tag :

It is used to pass the meta data about the webpage. (`<meta></meta>`)

## \* Tags in html :

In html every tags are predefined and we can't able to create the custom tag in html.  
Every tags are having different functionalities.

### Syntax:

→  $\langle \text{tag-name} \rangle$  → opening tag  
 $\langle /\text{tag-name} \rangle$  → closing tag

There are two types of tags :-

- 1) paired tag : It is having both opening and closing tag.
- 2) unpaired tag : It is having only opening tag or self closing tag.

## \* Basic tags in HTML

### 1) heading tags :

heading tags are used to display the heading into the web page.

All the heading tags are block level elements and paired tags.

There are 6 variants of heading tags that is h1 to h6.

All the heading tags are only differ in the font size.

h1 - 2em

h2 - 1.5em

h3 - 1.17em

h4 - 1em

h5 - 0.83em

h6 - 0.67em

### 2) Paragraph tag. (<p></p>)

This tag is used to display the paragraph into the web page and it is a block level element and paired tag.

To generate the random paragraph we can used the ~~text~~ embed abbreviation "lorem"

## \* Pre tag.

These tag displays the content into the web page exactly how we have written in the code.

We can give ~~white~~ spaces and line breaks within the pre tag.  
These is also block level element.

4) Div tag :

div tag access the container for HTML elements or parent element.

div is a block level element and also a paired tag.

5) Span tag :

Span is the inline level element and it is used to highlight some part of the text within the paragraph or other html element.

6) Img tag :

Image tag is used to ~~the~~ display the image into the web page.

image is the ~~2~~ inline block level element.

By default image tag takes 2 attributes ① src (source) & ② alt (alternative text)

We can provide the path for the image in 2 ways

① absolute path - we have to download the image and provide the path.

② relative path - here to directly we can fetch the image from the browser. by passing the URL

If the URL or <sup>image</sup> path is wrong then alternate text will get displayed.

We can set the height and width using height attribute and width attribute.

## \* Elements in html.

Elements is the combination of opening tag, closing tag along with content or children.

## \* Types of elements.

### 1) Inline level element.

Inline level element takes only the content and also it will not take the height & width CSS property.  
eg :- span tag, A tag. (anchor tag)

### 2) Block level element.

Block level element takes whole width of the browser window and also it takes height & width CSS property.

### 3) Inline block level element.

Inline block level element takes only the content width not whole width of the browser window and also it takes height & width CSS property.

## ★ Formatting tags (inline)

- 1) `<b>` or `<strong>`
- 2) `<u>` or `<ins>`
- 3) `<i>` or `<em>`
- 4) `<mark>`
- 5) `<q>`
- 6) `<sup>`
- 7) `<sub>`
- 8) `<small>`
- 9) `<del>` → it is used to strike the text.

## ★ Deprecated:

- 1) ~~strike~~ `strike`
- 2) `<big>`
- 3) `<center>` → block.

## \* Void elements in html)

void elements are self closing tags and using void elements we could not able to display any content into the web page.

eg :- ① <br/> - it gives the line break.

② <hr/> - it displays the horizontal line.

③ <input/> - it creates the Input field.

## \* Attributes in html :

Attributes provides the extra information or additional information about the html element.

Attribute must be present in the opening tag of the html element.

**Syntax:** attribute name = "attribute value" or only attribute name.

Every attributes are predefined that is we cantot create our own attribute.

There are two types of attributes :-

1) core attributes & 2) Element Specific attribute.

1) Core attributes → we can use in every html element.

① id

The id value must be unique and only one values we can pass.

id attribute is used to target the single html element in the css and Java script.

2)

Class :-

Class attribute value need not to be unique and we can pass multiple value for class attribute & It is used to target one or more html (multiple) html element in css & Java script.

3)

Style :-

It is used to give the css of html elements and this way of giving css is called as inline css and inline css gives priority.

4)

title :-

Title attribute is used to display the pop-up text when the user ~~over~~ <sup>hovers</sup> over the html element.

2)

Element Specific attribute :-

These attributes are limited to particular html elements and we can't able to use this attribute for every html element.

Eg :- src and alt in image tag, href in anchor tag.

src (attribute)

alt (attribute)

and other attributes like id, class, style etc.

id (attribute)

class (attribute)

style (attribute)

## \* Anchor tag <a> </a> (inline level element)

Anchor tag is used to display the hyperlink into the web page along with the href attribute. We should pass URL or file path as a value for the href attribute.

If we are not passing any value for href attribute then the hyperlink will redirect to the same web page.

If we want to redirect the webpage in different tab then we can pass the attribute target = "\_blank".

We can make image, heading, paragraph as a hyperlink by making this tag as the children of anchor tag.

## \* Marque tag <marque> </marque> (block level element)

Marque tag is used to display the scrolling text into the web page.

By default it takes two attributes

① Behavior

② Direction

- i) Behavior takes 3 values
  - ① scroll (default value)
  - ② alternate
  - ③ slide,

- ii) Direction takes 4 values
  - ① up
  - ② down
  - ③ left (default value)
  - ④ right.



Other attributes of Marquee tag

1) height & width.

It is used to set the height & width for the Marquee text.

2) bg color.

It is deprecated and it is used to set the background color for the Marquee text.

3) loop.

It defines how many times the text should scroll & it takes the number as a value.

4) scroll amount

This attribute defines the speed of the scrolling text & the default value is 6.

If the value is 0 the text will not scroll.

If the value is -ve than it will take default value as 6.

If we pass the value more than 6 then it will increase the speed of the scrolling text & if less than 6 the speed will decrease.

## \* Lists in html :

### 1) Order list :

It displays the list items with some order by default it will be numerical order.

To create the order list we can use `<ol></ol>` tag and to display the list item `<li></li>` tag.

`<ol>` tag takes one attribute that is `type = " "` which defines the type of the order and it takes multiple values such as: `ord-1` (default value), `A, a, I, i,`.

`start` attribute defines from where we have to start the order.

`reverse` Reversed attribute reverse the order for the list but it will not change the content.

### 2) Unorder list :

Unorder list doesn't come with any order.

To create the unorder list we can use `<ul></ul>` tag and to display the list item we can use `<li></li>` tag. `ul` tag also takes `type` attribute & it takes the values. `disk` (default), `circle`, `square`, `none`.

### 3) Description list :-

Description list is used to create the list along with the discription.

Within discription list we can use `<dt>` tag (discription term) to display the heading & we can use `<dd>` tag (discription detail) to display the content.

#### 4) Nested list :-

List which is present inside another list is called as nested list.

Note :- all the tags in the list are block level elements.



#### Table :-

We can create the table using table tag and it is a block level element.

We can create the table row using `<tr> </tr>` tag

we can create the table heading using `<th> </th>` tag.

We can pass the table data using `<td> </td>` tag.

`<thead>` & `<tbody>` are used to separate the table heading & table data.



#### Attributes for table tag :-

##### 1) `border` :-

It is used to set the border of the table and we should pass the number as a value which is thickness.

If gives the border for each & every cell & to merge the border we can use css property. `border-collapse`.

"`border-collapse: collapse`".

~~style = border = "1" style = border-collapse: "collapse"~~.

`border = "1" style = border-collapse: "collapse"`

### \* Cellpadding :-

It is used to give the space between content & the border.

### \* Cellspacing :-

It is used to give space between the cells.

### \* Attributes in <th> & <td> tags :-

- (1) Rowspan :- It is used to merge two or multiple rows.
- (2) Colspan :- It is used to merge two or multiple columns.



### \* Form HTML form :-

form is used to collect the data from the user.

To create the form in the web page we can use form tag `<form></form>` which is a block level element.

form tag takes one attribute that is action which defines which web page we want to redirect after submitting the form & we should pass the html file path as a value.

If we want to redirect into new tab then we have to use target = "-blank".

## \* Input tag <input/>

- It creates the input field in the form
- It is a self closing tag & void element
- It takes by default 3 attributes
  - ① type : it defines what type of input field we want to create and it takes multiple values such as text, password, email, number, date, file, check box, radio, etc.
  - ② name : name attribute access the container for the input value which is used when we are storing the data into the database.
  - ③ Id :

## \* Label tag <label></label>

- It is used to define the label text for the input field and it takes one attribute that is for.
- To make the connection between label & input tag we should pass the same value for id attribute in "input" tag and for attribute in label tag.

## \* Button tag.

- It is used to create the button & it takes one attribute that is type which takes 3 values submit, reset, button.

Note: In case of input type as range mandatorily we should pass max & min attribute.

## \* Other attributes in Input tag.

- \* Place holder : It is used to display the label inside the input field.
- \* Required : This makes the input field as mandatory.
- \* Textarea tag : <textarea> This also creates input field but bigger in size.  
In textarea we can give the line break.  
Textarea tag also takes two attributes that is rows & cols which defines no. of rows & no. columns.  
Style = "resize: none"  
This property makes the user not to resize the text area.
- \* Input tag
- \* <input> type = "radio" and type = "checkbox"
  - In radio the user can able to select only one option and in checkbox the user can able to select one or multiple options.
  - Name attribute value must be same for all the input text for both radio & checkbox.
  - Value attribute : When submitting the form this value attribute data will be stored in the database (Mandatory)
  - We should make all the input tag as a child of one parent element then the parent element id value must be equal to for attribute value in

table tag.

- Select tag : It is used to create the dropdown to display the option we use option tag .
- Select tag takes two attribute that is name & id .
- Option tag takes one attribute that is value .
- The optgroup tag is used to group the option which takes one attribute label . & it gives the label text for the options .

• Datalist tag (`<datalist>`)

- . It is used to create the dropdown along with the user can search for the options .
- . To display the options we should use option tag .
- . Mandatorily we should use input tag and to make the connection between input tag and datalist tag we should pass the same value for list attribute in input tag and id attribute in datalist tag .

## Semantic tags : (Block level element)

- Every semantic tags are block level elements .
- Semantic tags acts as the container for html elements .
- Semantic tags was introduce from the version html 5 .
- This tags describes its own meaning to both browser & developer .
- If we are using semantic tag in the webpage it is easy to debug the code & also improve the SEO (Search engine optimization) performance .

## \* Semantic tags :

- <section>
  - <form>
  - <table>
  - <article>
  - <figure>
  - <main>
  - <aside>
  - <header>
  - <footer>
  - <nav>
- |           |           |
|-----------|-----------|
| <header>  | <header>  |
| <aside>   | <aside>   |
| <article> | <article> |
| <figure>  | <figure>  |
| <footer>  | <footer>  |

## \* Media tags in html. (Inline block element)

- (1) Image tag : It is used to display images on the web page.
- (2) Audio tag : audio tag is used to embed audio strings such as music or other audio strings.
  - It takes one attribute that is src and we should pass the audio path.
  - controls attribute is used to give the play and pause button. that is audio controls.
  - Autoplay attribute is used to play the audio automatically once the user visit the web page.
  - loop attribute is used to play the audio again & again that is infinite times.
- (3) Video tag : It is used to display the video into the web page and it takes one attribute that is src and we should pass video path as a value.
  - We can also pass controls, autoplay, & loops for the video tag. <video>

④

iframe :

It is used to display one webpage inside another webpage that is nested webpage.

By default it takes 2 attributes ① src where we should pass the URL or file path of the nested webpage

② frameborder : It takes 2 values 0 & 1 if we pass 0 it will not give the border for the frame and if the value is one then it gives the border.

Note :- We can set the height & width for all the media tags with the help of height & width attribute.

Q\*

How to embed a map & youtube video?

→ to embed a map :-

① go to google maps

② search for the location

③ click on the share button

④ click on embed a map & copy the html code and paste within the body tag.

→ To embed a youtube video .

① Search for the video and click on that video.

② Right click and copy embed code paste inside the body tag .

Q.

How can we display the favicon?

→ Using link tag inside the head tag which takes 3 attribute . ① rel - it defines the relationship . Relationship between the current document & the linked resources .

② href we should pass the url or file path for the image .

③ type : It defines the type of favicon and the value is image/x-icon .

Q.

## How to display the icons into web page?



Using font awesome icons.  
It is a library which is having a number of icons  
and we can use this library by downloading it  
or CDN (content delivery network).

To display the icon in the web page.

- ① go to the browser & search for font awesome  
~~CDN~~.
- ② click on the first link & copy the link tag &  
paste inside the head tag.
- ③ Search for font awesome icons click on the first  
link & search for the icons copy the code snippet  
and ~~copy~~ paste inside the html code, body tag.

## \* CSS

- CSS stands for Cascading style sheet.
- CSS is used to give styling for the HTML element.
- We can also provide interactive animations with the help of keyframes.
- We can arrange the HTML elements in row direction & column direction (using flex & grid).
- We can also provide the responsive web design using CSS media queries.

## \* What is responsiveness in web page

- Depends on the screen width we can change the alignment of the web page this process is called as responsiveness.
- How many ways we can give the CSS for the HTML elements.

### (1) inline CSS.

- We can give the CSS using style attribute in the opening tag of the HTML element.

• Inline CSS gives more priority.

### (2) Internal CSS.

We can also provide the CSS using style tag `<style></style>` and this style tag should be present in the head tag.

### (3) External CSS.

- We can also give CSS by creating new CSS file & we have to make connection between HTML & CSS file using link tag.

• link tag takes 2 attributes  
① REL - which defines the relationship between the HTML document & this external CSS file and the value should be `stylesheet`.

② `href` - in href we should provide the external file path.

④ `@import` :  
We can also pass the css by importing the one file inside another css file.

Syntax : `@import url ("file path")`  
import statement should be present at the Top of css file

Note : Inline css give more priority

Note: The priority between internal & external css depends on the declared position that is if we define internal css after external css then internal css gives more priority and vice versa.

## ~~CSS Selectors~~

Selectors are used to target the html elements

① and give the CSS.

i) Simple Selector :

① `Id` : i) `Id` selector is used to target the html element unique.

ii) It should be prefix with `#` that is id value.

iii) `Id` selector gives more priority.

② class Selector :

i) It is used to target 1 or multiple html elements and we can give the CSS.

ii) The class value should be prefix with `.`(dot).

3) Tagname Selector :

i) We can target the html element using tag name but the disadvantage is if target all the element having the same tagname.

## (ii) Groupname Selector:

i) We can also group the html elements by using tag name, id, class name & we can give the CSS but it should be separated by ,.

## (iii) Universal selector:

i) It targets all the html elements & it should be used ~~prefix with \*~~.

Universal selector must be defined ~~as the~~ at the top of CSS code that is after import statement.

Universal selector gives least priority.

## 2) Combinator selector:

### i) Descendent selector:

This selector targets both direct & indirect child.

Syntax :- parent element + target element

### 2) Child selector:

This selector targets only the direct child

Syntax :- parent element > target element.

### 3) Adjacent sibling selector:

This selector targets the first nearest sibling.

Syntax :- sibling element + target element.

### 4) General sibling selector:

This selector targets all the sibling.

Syntax :- sibling element ~ target element.

### 3) Attribute Selector.

We can also target the html element with the help of other attribute.

**Syntax :-** target-element[att-name = "value"]  
(or)

target-element[att-name]

### 4) Pseudo Element Selector:

gt is used to target the content of the html element.

**Syntax :-** target-element :: pseudo-element { }

- ① First-letter :- gt targets the first letter of the content.
- ② First-line :- gt targets the first line of the content.
- ③ Before :- gt is used to add some content before the previous content.
- ④ After :- gt is used to add some content after the previous content.

⑤ Selection :- If we want to give any css while selecting the content we can give selection selector.

⑥ Marker : It is used to give css for the list items.

### 5) pseudo class selector -

#### 1) Dynamic pseudo class selector:

It is used to give the css dynamically.

**Syntax :** target-element : pseudoclass selector.

- ① link
- ② visited
- ③ hover
- ④ active
- ⑤ focus

<a> <a>

multiple

<a> <input>

- \* 2) Structural pseudoclass selector:
- 1) first-child.
  - 2) last-child.
  - 3)  $n^{\text{th}}$  child. → (Odd & even also we can pass).  $n$  number.

\* CSS units :-

① Absolute unit :

Absolute unit are fixed that is it will not change depends on other parameters such as parent element, root element, screen height & screen width.

The absolute unit are :- mm, cm, in, px, pt, pc.

$\frac{1}{100}$  point picas.

$$1 \text{ cm} \rightarrow 10 \text{ mm}$$

$$1 \text{ in} \rightarrow 2.54 \text{ cm.}$$

$$1 \text{ px} \rightarrow \frac{1}{96} \text{ of 1 inch (1mp)}$$

$$1 \text{ pt} \rightarrow \frac{1}{72} \text{ of 1 inch}$$

$$1 \text{ pc} \rightarrow \frac{1}{6} \text{ of 1 inch. (or) } 12 \text{ pt.}$$

② Relative unit :-

Relative unit are not fixed that is the length of the element will get modified depends on other parameters.

The relative units are.

font-size } em → relative to the element (parent)

rem → relative to the root element.

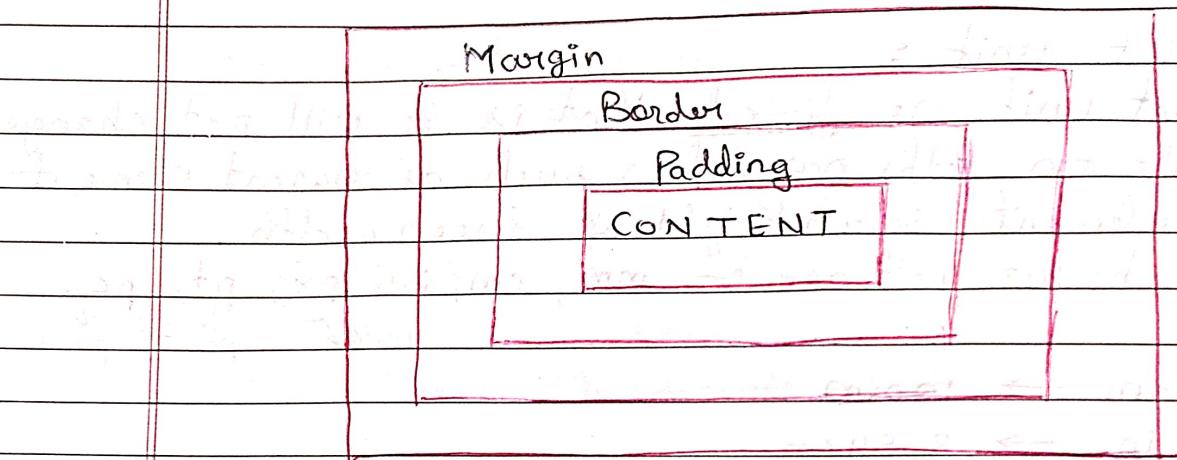
% → relative to the element (parent).

height & width. } vh → viewport height. that is screen height.

vw → viewport width that is screen width.

## \* CSS Box Model :

- i) The CSS Box is a fundamental concept that describes how elements are rendered on a webpage.
- ii) Every HTML elements are considered as a Box, and the Box model consists of content, Padding, Border, margin.



### i) Border →

This property is used to set the Border for HTML elements and it takes 3 values.

i) Thickness of the Border (in px)

ii) Type of the Border

solid  
 dotted  
 dashed

iii) Color of the Border

Also we have Border-top, Border-left, Border-right, Border-bottom properties.

### ii) Padding →

i) This property gives the space between Border and the content.

ii) We can pass. @ one value (All four sides).

b) two values  $\rightarrow 10\text{px} \quad 20\text{px}$   
 $\downarrow \quad \downarrow$

Top/Bottom left & right.

c) four values  $\rightarrow 10\text{px} \quad 20\text{px} \quad 30\text{px} \quad 40\text{px}$   
 $\downarrow \quad \downarrow \quad \downarrow \quad \downarrow$   
 Top Right Bottom Left

(clockwise direction)

③ Also, we have padding-top, padding-left,  
 padding-right, padding-bottom property.

iii) Margin:

i) Margin gives the space outside the border

ii) This property also takes

a) 1 value  $\rightarrow 20\text{px}$  (All four side)

b) 2 value  $\rightarrow 10\text{px} \quad 20\text{px}$

c) 3 value  $\rightarrow 10\text{px} \quad 20\text{px} \quad 30\text{px}$

③ Also, we have margin-top, margin-right, margin-bottom, margin-left property.

\* Formatting Text:

1) Color  $\rightarrow$  It is used to give the CSS for text & it will change the text color.

2) text-align  $\rightarrow$  It will align the text in 3 ways:  
 $\rightarrow$  centre

left (default)

right

3) font-weight  $\rightarrow$  It will use to bold the text or to normalize the heading text.

4) Text-decoration → This will decorate the text using 3 ways →  
(a) overline → give line on over / above the text.  
(b) underline → gives underline to the text.  
(c) line-through → it strikes out the text.

5) text transform → This css property is used to convert text to uppercase, lowercase, capitalize.  
(Controls the capitalization)

lowercase - converts all text to lowercase

uppercase - converts all text to uppercase.

capitalize - converts only the initial letter to uppercase.

6) letter-spacing :- This css property is used to give space between each letter.  
e.g. letter-spacing: 3px;

7) font family :- This, css property is used to give font style to the text.  
e.g. font-family: cursive;

8) text-shadow :- This css property is used to give shadow to the text.

It takes 4 values.

1<sup>st</sup> value - x direction

2<sup>nd</sup> value - y direction.

3<sup>rd</sup> value - opacity.

4<sup>th</sup> value - color of shadow.

y-direction :- If value is positive the text shadow goes down, if value is negative the text shadow goes up.

x-direction :- If value is positive the text shadow goes right if value is negative the text shadow goes left.

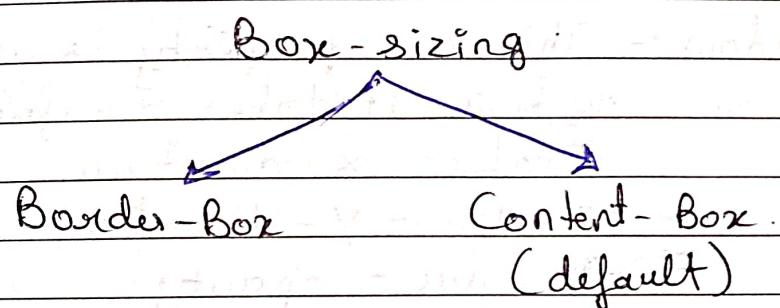
- 9) word-spacing :- This CSS property is used to give space between each words.  
e.g. word-spacing: 10px;
- 10) text-indent :- This CSS property is used to give indent/space in the beginning of first line of a paragraph or text.
- 11) list-style :- This CSS property is used to give marker/list style to the list items.(disc, square, circle, none).
- 12) border :- This CSS property is used to give border to an element, it takes 3 values ; thickness, type, color.
- 13) box-shadow :- This CSS property is used to give shadow to box, it takes 4 values:  
1<sup>st</sup> value - x-direction  
2<sup>nd</sup> value - y-direction  
3<sup>rd</sup> value - opacity  
4<sup>th</sup> value - color.
- 14) font-size :- This CSS property is used to give size of the font.
- 15) height →
- 16) width -

\* Why we are targeting universal selector (\*) and passing padding: 0, margin: 0 and box-sizing: border-box?

- Padding: 0 and margin: 0 is to prevent the default padding and margin value which is coming from default CSS i.e. user agent stylesheet.

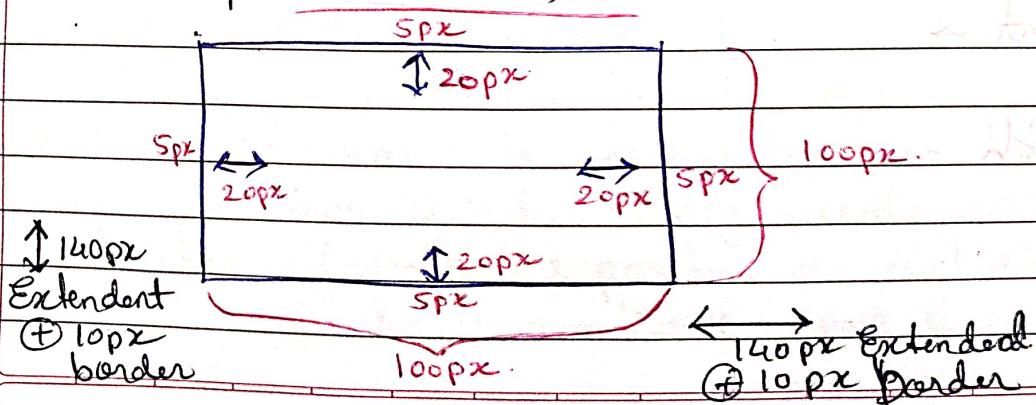
→ Box-sizing property :-

- This property takes two values i.e. border-box & content-box (default value).
- If the value is content box, if we pass any border or padding then the value will get added to the height and width of the html element.



→ Content box :-

padding: 20px;  
border: 5px solid black;



If the value is border-box, then if we pass any border or padding then the value will not be added to the height and width.

### \* Background properties :-

#### 1. Background color :-

It is used to set the background color for the html elements and only one background color we can set.

#### 2. Background :-

- It is used to set one or multiple background colors with the help of linear gradient function.
- This function takes three or multiple values.
  - i. direction - to top, to right, to bottom, to left and the rest of the values are colors.

#### 3) Background-image :-

This property is used to set the background image with the help of url function. and we should provide the path or url as a value.

Background-image : url (" " );

#### 4) Background-size :-

i) We can set the height & width of the background image using this property & it takes 2 values.

Eg :- Background-size : 300px 200px.

+                    +  
width              Height

2) Also we can pass contain & cover values ;

Background-size = contain;

Background-size : cover.

3) If the value is cover then the Background image

will be cropped depends on the aspect ratio & will display. Also the image will not repeat.

4) If the value is contain, then the entire image get displayed but if the aspect ratio doesn't match the image will get repeated.

### 5) Background-repeat :-

It takes 2 values.

a) Background-repeat : repeat ;

It repeats the background image.

b) Background-repeat : no-repeat ;

It will not repeat the background image.

20/04/25

### \* Display properties in CSS :-

- This property specifies how an element is displayed in the document layout.
- It is one of the most important properties for controlling layout behaviour.
- It takes multiple values such as block, inline, inline-block (flex), grid, none.

Q. What is the difference between display:none & visibility: hidden ?

- display: none :-

It will disappear the entire html element and it will not occupy the space of the html element in the webpage.

- visibility: hidden :-

It occupies the space of the html element in the web page & it disappear only the content of the element in the web page.

### \* display : flex :-

- Flex is used to arrange the html elements in the row direction or column direction.
- Flex is used to give the alignment for the html elements.
- Every flex properties (including display : flex) must pass to the parent container.
- By default every elements will be arranged in the row direction.

### \* Flex properties :-

#### 1) Flex direction :-

This property defines which direction the elements should be arranged and it takes four values.

- i) row (default)
- ii) column
- iii) row reverse
- iv) column reverse

#### 2) Align items :-

- This property is used to align the html elements within the parent container.
- If flex direction is row then it will align the html elements in vertical direction and if flex direction is column then it will align the html elements in horizontal direction.
- This properties takes three values,
  - i) start (default)
  - ii) end
  - iii) center.

### 3) Justify content :-

- This property also used to align the html elements within the parent container.
- If flex direction is row than it will align the html elements in horizontal direction and if flex direction is column than it will align the html elements in vertical direction.
- This property also takes multiple values such as
  - i) start (default)
  - ii) end
  - iii) center
  - iv) Space between - It will give the space between file item.
  - v) Space around - It will give Space all around the div but not equally.
  - vi) space evenly. - It will give equal space.

### 4) gap :- gap: 5px;

- This property gives row gap if flex direction is row and column gap if flex direction is column.
- It is used to give custom gaps.

## \* display grid:-

- It is used to arrange the html elements in both row & column direction.
- Every grid properties we should pass to the parent element.

## \* grid properties :-

### 1) grid template column

It is used to set no. of columns and length of the columns.

The no. of values for this property depends on no. of columns and we should pass the length of the column as a value.

Eg:- grid-template-columns: 150px 150px 150px;

It creates 3 columns with the length 150px.

We can also pass for (fractional unit) to set the length of the column or length of rows which divides the parent element and set the length for child element.

We can use repeat function as a value if we want to set the same length of the row & column. repeat function takes 2 values

① no. of rows or column.

② length of the row or column.

### 2) grid template rows :-

It is used to set the no. of rows & length of the rows.

gap :-

gt is used to set the row gap & column gap  
it takes 2 values.

① row gap:

② column gap.

Alternatively we can use the property rowgap & columngap.

## \* Position:

The position property determines how an element is positioned in the document. There are 5 values in position

1) static (it is default value)

2) relative

3) absolute.

4) fixed.

5) sticky.

1) Static → It is normal flow & top, bottom, right, left is not used means no effect on it.

2) relative → It is normal flow & we can use top, bottom, right, left in it.

3) absolute → It is not a normal flow and we can use top, bottom, right, left in it, & it takes viewport but if we want to set the position into the element only, then we have to pass position relative in the parent element.

4) fixed :- It is not a normal flow we can use top, bottom, right, left in it but it takes viewport & it is fixed in position.

5) sticky :- It is a normal flow we can use top, bottom, right left in it but it sticks to a position within the element only & while scrolling it is stick in one position & when it reaches the position.

#### \* z-index :-

- z-index is used to overlap one element on the other element & it take number as the value.
- In z-index, <sup>the</sup> element which have greater value overlap the other element which has the lower value.

## 8 Media queries CSS.

- It is a powerful tool that is used to create responsive web design for the web application depends on device characteristic such as screen size, resolution, orientation, etc.

Syntax: @media (max-width: value), and  
(min-width: value) {  
 code  
}

★

## Transition & transform:

★

### Code:

#### :CSS:

\* {

padding: 0;

margin: 0;

box-sizing: border-box;

}

#### body {

display: flex;

height: 100vh;

width: 100%;

align-items: center;

justify-content: center;

}

#### :container {

height: 200px;

width: 200px;

border: 1px solid black;

transition: 3s;

}

#### : container : hover {

cursor: pointer;

background: deeppink;

transition: 3s;

/\*

height: 250px;

width: 250px;

transform: translateX(200px);

transform: translateX(-200px);

transform: translateY(200px);

transform: translateY(-200px);

transform: translate(200px, 200px);

transform: rotateX(-360deg);

transform: rotateY(-360deg);

transform: rotate(-360deg);

transform: scalex(2);

transform: ScaleY(2);

transform: scale(0.8);

transform: Scale(1.5, 3);

transform: skewX(40deg);

transform: skewY(-40deg);

transform: translate(150px, -150px)

rotate(360deg);

scale(0.7) skew(40deg);

3}

function: transform

transform: rotate(45deg);

transform: rotate(-45deg);

transform: rotate(45deg);

transform: rotate(-45deg);

transform: rotate(45deg);