

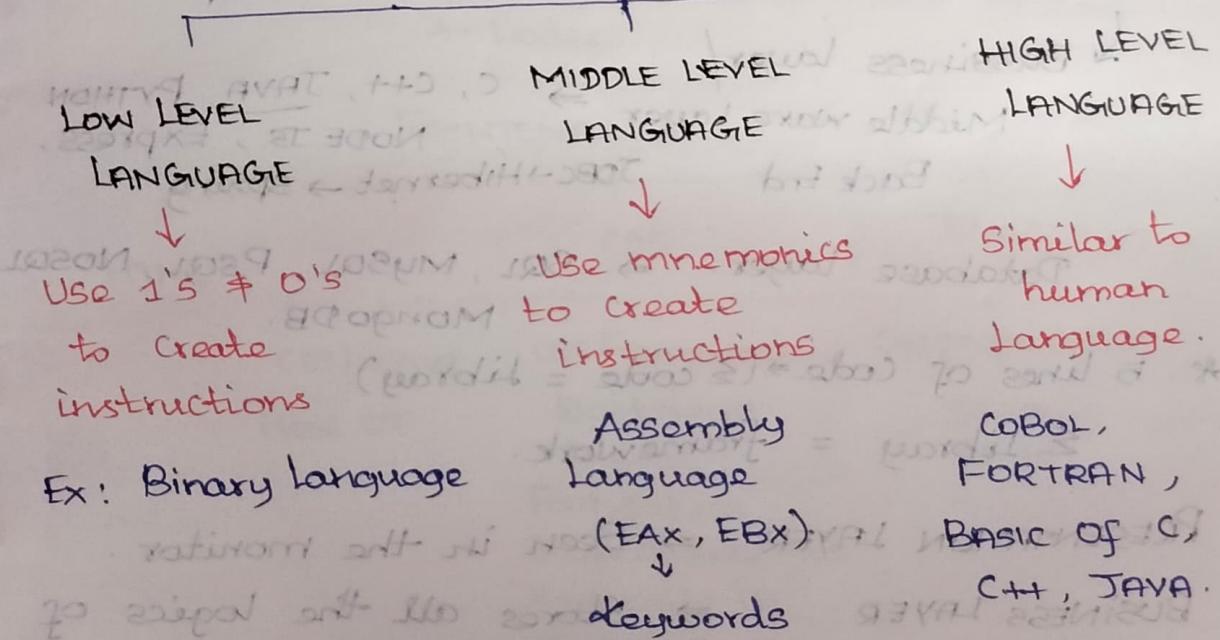
LANGUAGE :

It is a tool to communicate with someone.

PROGRAMMING LANGUAGE :

- * It is a tool to communicate with a machine.
- * Computer programming is a collection of instructions.
- * These instructions come in the form of many different languages such as C++, Java, Javascript, Python, Ruby and rust.

COMPUTER LANGUAGE



SOFTWARE :

It is a set of instruction which is used to reduce manual work.

There are two types of Software.

⇒ SYSTEM SOFTWARE

⇒ APPLICATION SOFTWARE.

SYSTEM SOFTWARE :

It is used to control the device.

Ex : Operating SIm.

APPLICATION SOFTWARE

It is used to perform Specific task.

Ex : OLA, zomato.

LAYER'S To BUILD APPLICATION

ROAD MAP OF TECHNOLOGY

→ Presentation layer → HTML, CSS, JS

Serveleted. Front End

Bootstrap

→ Business layer /

Middle ware Layer → C, C++, JAVA, PYTHON
NODE JS, Express.

Back End

JDBC → Hibernate → Spring

Database Layer. → SQL, MySQL, PSQL, NOSQL
MongoDB.

* 5 lines of code ⇒ (≤ code = Library)

≤ Library = framework.

PRESENTATION LAYER ⇒ Seen in the monitor.

BUSINESS LAYER ⇒ Stores all the logics of application.

DATABASE LAYER ⇒ Store data on database.

TECHNOLOGY :

Tools and techniques implement application.

Software

Set of programs that governs the functioning of computer.

System Software

Control the internal computer operation.

Application Software

Carries out necessary operations for specified application to function.

System Management Program

1. Operating Syst
2. Device Drives
3. System utilities

Developing Software

1. Programming Language.
2. Language Translator
3. Linker
4. Loader.

General Purpose Software

Word Processor

1. Word Processor
2. Presentation

Spreadsheet

Image editor

1. Reservation
2. Attendance
3. Billing
4. Report card, etc.

Specific Purpose Software

-Ware

FRONTEND

HTML

CSS

JAVASCRIPT

css library

Pure CSS

css Framework

Bootstrap

Bulma

Foundation

Materialize CSS

Semantic UI

Tailwind CSS

CSS preprocessor

SASS

LESS

ES6

TypeScript

Package Manager

npm

yarn

JS Framework

Angular JS

React JS

Vue JS

JS Library

JQuery

Script.js

US

P5 JS

D3 JS

Underscore.js

Jodash

JQuery UI

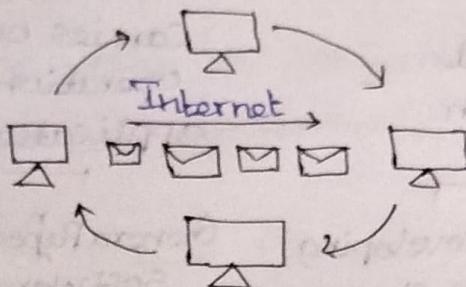
JQuery Mobile

JQuery Plugins

INTERNET :

* Internet is the interconnection of world wide device that can able to share datas.

Data's → Information → Media's / files.



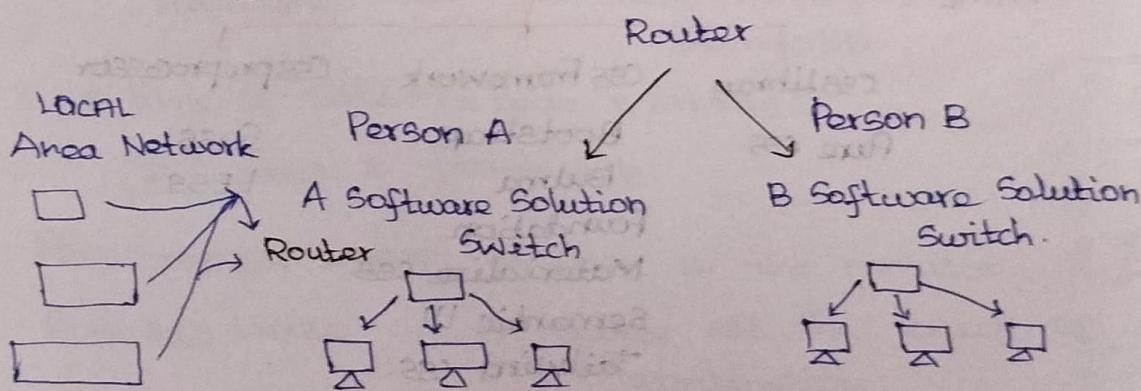
* Internet follows a mechanism is Packet routing Network.

* It works by using a packet routing Network that follows.

1. INTERNET PROTOCOL.

2. TRANSPORT CONTROL PROTOCOL.

Packet Routing Network.



Packet routing Network : [Data sent over the internet is called a message]

⇒ Before messages get sent, they are

broken up into tiny parts called "PACKETS"

IP : It specifies that Computer Should send data to other Computer with an attached numerical address (IP address)

TCP :

It works with IP to ensure transfer of data is dependable and reliable. (No packets lost, No delay affecting data quality).

PACKET :

The message which sends to over the internet. The message will connected into tiny Packets.

ROUTING :

Establish a connection b/w different network.

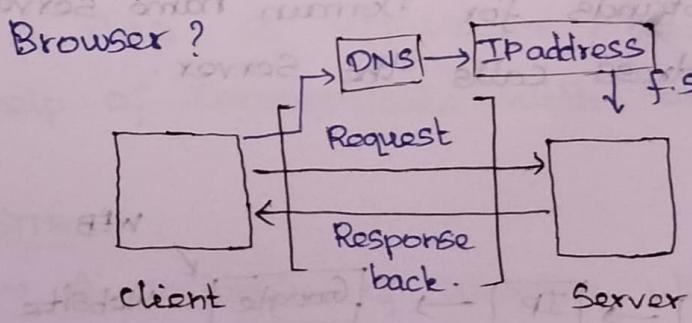
Network :

A group of device which is connected with each other.

NOTE :

Every devices which is connected to internet it will have IP address.

What happen's while Surfing Internet through Browser?



Browser Url

http://www.google.com
Google
facebook
Submit

http://www.facebook.com
Login

Sign Up

URL : Uniform resource locator

HTTP : Hypertext transfer protocol

HTTPS : Hypertext transfer protocol Secured

http : www.google.com / Search = facebook / - / -

HTTP : Protocol

www : Web

google.com : Domain name.

facebook : Parameter 1

Parameter 2

Parameter 3

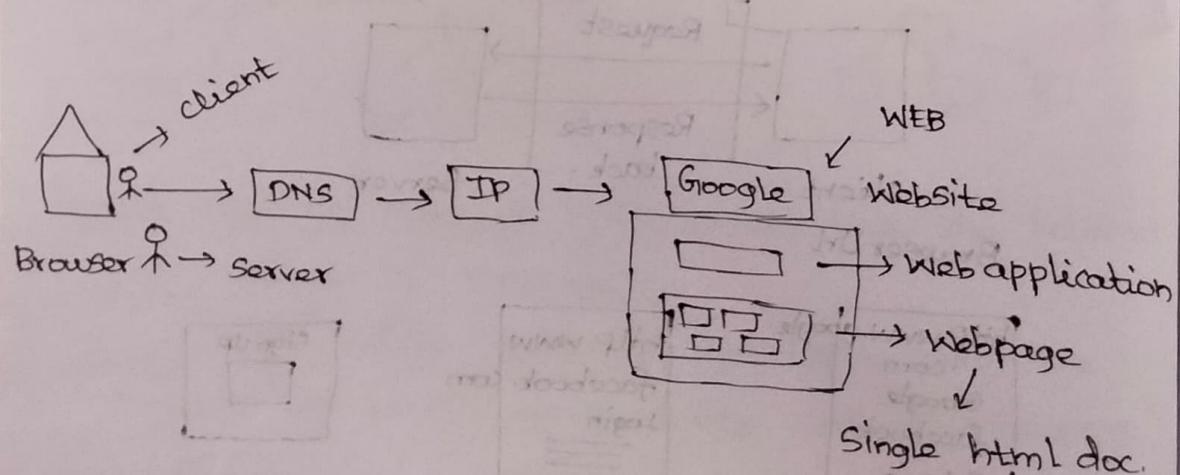
SEREVER → Use to Response with the client.

CLIENT → Use to Request the Server.

* Client and Server talking to medium called HTTP.

DNS AND ITS JOB :

- * It will get the domain name and it will converted into IP address.
- * DNS Stands for Domain name server
- * IP address calls the Server.



WEB : A collection of Information
(Infinite no of Information)

WEB PAGE : A Single html document is considered.

* A document which can be displayed in a Web browser.

* These are also often called just "Pages".

Internet is a subset of Web SUBSET → It is part of web
WEB APPLICATION :

→ Application program (or) application Software is a computer program designed to help people perform an activity.

→ A web application is a computer program that utilizes web browsers to perform tasks over the Internet.

→ A collection of Webpages.

BROWSER :

→ It is a piece of software application.

→ A Web browser (or) browser is a program that retrieves and displays pages from the Web, and lets users access further pages through hyperlinks.

A piece of application / mediator with a help of communicate with Server.

WEB SITE :

→ A collection of webpage which are grouped together and usually connected together in various ways.

→ Often called a "Website" (or) a "site".

A collection of Web application and Webpages.

WEB SERVER :

⇒ As we saw, the internet which allows billions of computer to be connected all together.

⇒ Among these computers, some computer (called webserver) can send messages to web browsers.

⇒ A Web Server is a computer hosting one or more websites.

⇒ "Hosting" means that all the webpages and their supporting files are available on that computer.

⇒ The Web Server will send any webpages from the website it is hosting to any user's browser, per user request.

⇒ How to host : <http://www.wikihow.com/Host-your-own-Website-for-Free>.

NAVIGATION :

It is used to navigate the module.

WEB :

⇒ Web is a subset of Internet.

⇒ The Web consists of pages that can be accessed using a web browser.

⇒ Web is an interconnected Sm of Public Webpages accessible through the internet.

⇒ The WWW depends on a number of protocols, Sm and technologies all working together in harmony.

SYNCHRONOUS

APPLICATION

Only one thread will be executed at a time

Ex: Bank Application

SINGLE PAGE

APPLICATION

Entire Application written on Single html document.

Does not required page reload time

This kind of building Application is modern approach.

Ex : Gmail, Facebook, Instagram, Googlemap

STATIC WEBSITE

- * Content of webpage can't be change at runtime
- * No interaction with database.
- * It is faster to load as compared to dynamic website.
- * No feature of content management.
- * Cheaper development costs
- * HTML, CSS is used for developing websites
- * Same Content will be delivered every time.

ASYNCHRONOUS

APPLICATION

Multi threads will be executed at a time

Ex : Hotstar.

MULTIPLE PAGE

APPLICATION

Entire Application written on multiple html document

Required More page reload time.

The kind of building Application is Traditional approach.

Ex : Zudio.

DYNAMIC WEBSITE

- * Content of a web page can be changed.
- * Interaction with a database is possible.
- * It is slower than Static Website.
- * Feature of content management.
- * More development costs.
- * Server, Side language such as PHP, JAVA, NODEJS are used.
- * Content may change every time the page is loaded.

HTML → It creates a structure of your webpage & application.

CSS → Designing for webpage.

JS → Function of the webpage.

HTML :

* HTML stands for HYPERTEXT MARKUP

LANGUAGE.

Hypertext → Reference link

text → Information on the webpage.

markup → Presentation of the text

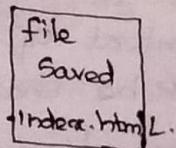
Language → Browser Language. (communicate with Browser)

* Browser can understand HTML language.
Year : 1993

Inventor : Tim Berner's Leege.

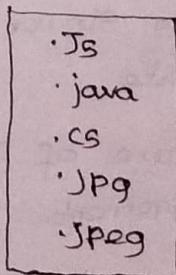
File can be saved to .html extension.

Ex :



. is used to instruct the OS what file format.

Other extensions :



- (x) 1. Structure of html? → To instruct the browser we use declaration and structure of html.
2. List of void tag?
3. What is doctype html?

STRUCTURE OF HTML :

<!DOCTYPE html> → Version of html 5

<html> → Root of html.

<head> → It contains Meta data.

Meta data

<title> Webpage </title>

</head>

<body>

<p> Content </p>

</body> → The content that you want to display on the web page

</html>

TAG : → Tag will have some information.

* Anything which is enclosed with angular bracket.

* There are two types of tags.

⇒ PAIRED TAG / CONTAINER TAG

⇒ UNPAIRED TAG / VOID TAG / EMPTY TAG,

ENCLOSED TAG. (Self closing)

PAIRED TAG

UNPAIRED TAG

It test
is placed <html>

<head>

b/w a <title>

tag & its

companion

tag. <p>

 break

<hr> horizontal line.

If doesn't have a
companion tag (or) closing
tag.

Heading tags ⇒ h₁, h₂, ..., h_n

Tag →

 Emphatic!

Element contents

BITS OF TEXT

→ close Tag

Element contents

BITS OF TEXT

TEXT FORMATTING TAGS :

- * Text formatting tag is an paired tag.
- * It is open and closed tag.
- * There are various types of tags.

There are 13 tags:

 bold text

<big> big text

 emphasized text

<i> italic text

<small> smaller text

 Important (or) weight text

<sub> subscript text

<sup> superscript text

<ins> inserted text

 deleted text.

Deprecated Tags:

This 3 tags can be used but developer we should not used.

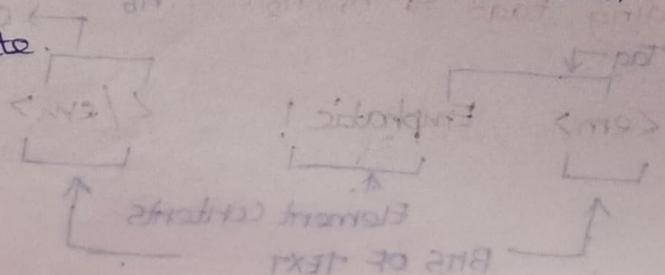
<s> used to strike a text

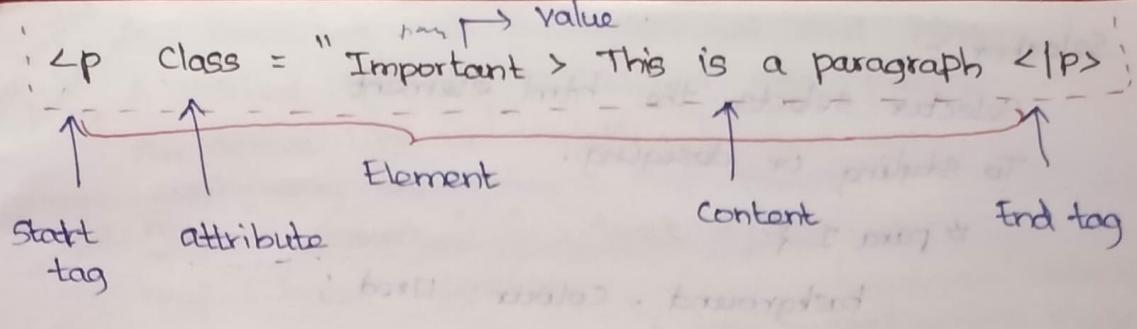
<strike>

<u> - used to underline the text.

ATTRIBUTES :

- * Additional information about Tag / Element
- * It always Specified in Opening tag.
- * **SYNTAX:** Attribute name = "Value".
- * Each tag will have their own attribute.





CORE ATTRIBUTES :

Frequently using attributes called core attributes.

- * ID
- * CLASS
- * STYLE
- * TITLE

ID : (Unique)

ID is an core attribute

ID Value always be Unique

It used to target particular / specific element.

Way to Telling CSS :

Inline Style

Internal Style

External Style.

Internal style :

Style is a tag inside the head tag is

nothing but Internal style. Inside the head tag

We create style.

<style> </style>

Selector {

Property and Value

}

<style>

background-color: black; color: white; font-size: 16px; font-weight: bold; text-align: center; width: 100%; height: 100%; position: absolute; top: 0; left: 0; z-index: 1000; opacity: 0.8; border-radius: 10px; padding: 10px; margin: auto; transition: all 0.5s ease-in-out;

Selector :-

Selector selects the html element.

To styling or designing.

para 1 {

background - colour : red ;

}

How to comment the line in HTML ?

<!-- Content --> shortcut Ctrl + /

How to comment the line in CSS ?

/*

=====

*/

| tag | attributes |
|------------|------------------------|
| Style
: | Internal
Style |
| title | Metadata tool tip |

INLINE STYLE :

style as tag is said to be Internal

↳ Style is an attribute is nothing but inline

Style. Ex : border : 1px solid black

↳ Written inside opening tag, width style colour.

CLASS :

To target multiple (or) All the element.

I used to target multiple element

STYLE : common design for all element, same name can be used.

Inside the head tag we create style.

TITLE :

It's a part of a document which contains information about the document.

It's create a Metadata to information about the information.

Title is an attribute is nothing but ToolTip.

ANCHOR TAG : => Paired Tag

* <a> tag defines a hyperlink, which is used to link from one page to another.

* By default, links will appear as follows in all Browsers.

characteristics
 → An Unvisited link is underlined and blue.
 → A Visited link is underlined and purple.
 → An active link is underlined and red.
 SYNTAX : attributes of anchor tag is href , target
` target `
 href : hyper reference . Webpage | Websites
Target :
 → text → converted into hypertext it visits
 → Self → Replace old tab & produce o/p
 → blank → open in new tab
 → Parent
 → top .

C : DESKTOP : WORKSPACE : HTML : INDEX.html .
 Parent node child node child node

Starts to parent node and ends with child node
 is called as Absolute path.

Starts to child node and ends with child node
 is called as Relative path.

ELEMENT :

Content inside the tags are called HTML Content
 or Inner HTML.

Entire Content which is enclosed with tags
 are called element.

`<p class = "important"> This is a paragraph </p>`
 ↑ ↑ ↑ ↑
 Start tag Attribute value content end tag

IMAGE TAG : ⇒ Unpaired Tag .

- Import Image to your Webpage.
- Import Image into html file.

``
 Tag Name Attribute Name Attribute Value Attribute Name Attribute Value .

~~GROUPING ELEMENTS~~ GROUPING html Element

< p id="Para1" > Content </p>

Element :

Consists of a opening tag and closing tag with attributes and content.

elements :

↳ Multiple Element.

Nested Element :

↳ Element inside element

Grouping ↗ **Block level element**

↘ **Inline level element**.

Block Level Element :

Paragraph is a block level element it

will occupy 100% of width

Ex :

<p> This is my content - block level element </p>

Inline Level Element :

↳ In Occupy Only a content of width.

↳ → It is an Inline level element.

We can group the elements in 2 types.

html elements

Block level Elements

Inline level Elements

Both are Content Wrappers

Means wrapping the Contents.

Grouping Content :

- ↳ <div> - block level
- ↳ - Inline level.
- ↳ Both are content wrapper
(wrapping content)

↳ Block level occupy entire width of the screen.

↳ Inline occupy only content width.

HTML ATTRIBUTES :

| ATTRIBUTE | DESCRIPTION |
|-----------|--|
| alt | - Specifies an alternative text for an image. |
| disabled | - Specifies that an i/p element should be disabled. |
| href | - Specifies the URL (web address) for a link. |
| id | - Specifies a Unique id for an element. |
| src | - Specifies the URL (web address) for an image. |
| style | - Specifies an inline CSS style for an elements. |
| title | - Specifies extra information about an element (displayed as a <u>tool tip</u>) |
| value | - Specifies the Value (Text Content) for an i/p elements. |

LIST :

- ↳ Order List.
- ↳ UnOrder List
- ↳ Description List.

Order List :

- ↳ List will maintaining Some Sequence

Order list.

- ↳ It is an paired tag.

SYNTAX :

To provide list elements

↳

To change bulletin sequence.

- ↳ we use type as a tag in

A → uppercase

a → lower case

i, I → roman.

Ex :

i. Apple

ii. Orange

iii. Mango.

UnOrder List :

- ↳ List it will not maintain any Sequence it follows.

- ↳ Unorder list start with the tag

- ↳ It is an paired tag.

Each list item starts within the tag

Attributes type :

(•) disc, circle, square, none.

default → "disc" (□)

Description List :-

If you want to describe the term with some meaning.

- ↳ Start with `<dl>`
- ↳ Definition term starts with `<dt>`
- ↳ Definition details starts with `<dd>`

Ex :-

`<dl>`

`<dt> TD </dt>`

`<dd> Software development </dd>`

TABLE :-

- ↳ HTML table allow web developers to arrange data into rows and columns.
- ↳ Table is a collection of rows and columns.
- ↳ HTML table is made of cell's
- ↳ It is an paired tag.

Table cell :- Each table cell is defined by
`<td> </td>`

Table row :- Each table row starts with `<tr>` and ends with `</tr>` tag.

Table head :- Table header starts with `<th>` and ends with `</th>` tag.

Table foot :- Table foot starts with `<tfoot>` and ends with `</tfoot>` tag.

Table Cell - padding :-

- ↳ Space between cell content and cell edge
- ↳ Padding is 0 by default.

Ex :-

`th, td {`

`padding: 15px;`

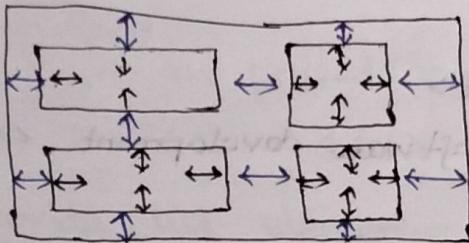
Table Cell-Spacing :

- ↳ Space between each cell is cell spacing
- ↳ spacing is space by default.

Ex :- table {

 border-spacing: 30px;

}

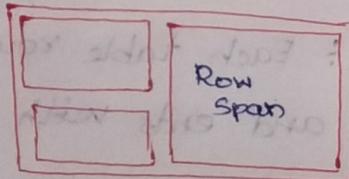
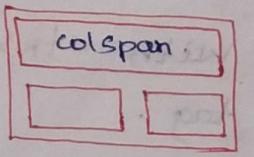


cell padding

cell spacing.

Table Span :

- ↳ Col Span (Column Span) Occupies next column cell. `<td colspan="2">`
- ↳ Row Span (Row Span) Occupies next row cell. `<td rowspan="2">`
- ↳ Default Value 1.



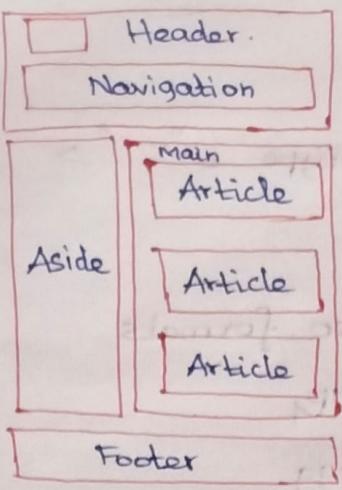
Semantic Tags :

- ↳ Tags which has some meaning is known as Semantic tags.

- ↳ These tags will be understood by both developer and browser.

Ex :-

`<form>`, `<table>`, `<Aside>`,
`<Section>`.



<div id="header">

<header>

<div id="nav's">

<nav>

<div id = "content">

<div id = "aside">

<content>

<aside>

<div id="footer">

<footer>

HTML 4

VS

HTML 5

Semantic Tag.

Way to remember Semantic tag :-

H → Header

D → Details

A → Aside

M → Main

N → Nav

S → Summary & Section

T → Time

A → Article

F → Figure

F → Fig caption / Footer.

Multimedia Tags :-

<audio controls>

<source src = " " type = " " >

</audio>

↳ Import audio and video to the webpage.

Audio Attributes :- Audio Format's

↳ Auto play

↳ MP3

↳ Muted

↳ Ogg

↳ Loop.

↳ WAV.

Video Tag :

< Video Control >

< Source Src = " " type = ". " >

< /video >

Video Attributes

↳ Auto play

↳ Muted

↳ loop

Video formats

↳ MP4

↳ Ogg

↳ WebM

Iframe :

An HTML iframe is used to display a webpage within a web page.

<iframe Src = "#" height = "200" width = "300"

title = "iframe Example" > </iframe>.

HTML forms :

HTML forms are required to collect data from the user.

HTML form is a document which stores information of a user on a webserver.

Collecting information from the user through electronic format.

| Name | Value |
|--------------------------------|---|
| Name | <input type="text" value="John"/> |
| Sex | <input type="radio"/> Male
<input type="radio"/> Female |
| Eye Color | <input checked="" type="checkbox" value="red"/> red |
| Check all that apply | <input type="checkbox"/> Over 6 feet tall
<input type="checkbox"/> Over 200 pounds |
| Describe your athletic ability | <input type="text" value="Enter my information."/> |

HTML form elements :

Tag

Description

<form>

Defines an HTML form for user input.

<input>

Defines an input control.

<text area>

Defines a multiline input control
(text area).

<table>

Defines a table for an <input>
element.

<fieldset>

Groups related elements in a form.

<select>

Defines a drop-down list.

<option>

Defines an option in a drop-down
list.

<button>

Defines a clickable button.

Value

Description

button

Defines a clickable button (mostly
used with a Javascript to activate
a script).

checkbox

Defines a checkbox.

color

Defines a color picker.

date

Defines a date control (year, month,
day (no time))

date-time -

Defines a date & time control.

local

(Year, month, day, time (no time zone))

Email

Defines a field for an e-mail address.

file

Defines a file-select field & a "Browse"
button (for file uploads).

hidden

Defines a hidden input field.

image

Defines an image as the submit button.

| | |
|----------|--|
| month | Defines a month & year control
(no timezone). |
| number | Defines a field for entering a number. |
| Password | Defines a password field. |
| Radio | Defines a radio button. |
| range | Defines a range control (like a slider control). |
| reset | Defines a reset button. |
| Search | Defines a text field for entering a search string. |
| Submit | Defines a submit button. |
| tel | Defines a field for entering a telephone number. |
| text | <u>Default</u> defines a single-line text field. |
| time | Defines a control for entering time
(no time zone). |
| url | Defines a field for entering a URL. |
| week | Defines a field week & year control
(no time zone). |

CSS

CSS - Cascading Style Sheet

↳ For styling and designing.

Ways to link CSS :

Types of CSS :

1) Inline CSS - style as an attribute.

2) Internal (or) Embedded CSS - style as an tag.

3) External CSS.

Inline Style : 1st priority.

Ex :

<p style = "color : red"> Hello CSS </p>

Internal Style :

Ex :

<style>

P { color : Value }

</style>

External CSS Style :-

External CSS is used to apply CSS on multiple pages.

Pages. Inverted → (1) related public for CSS files.

Extension must be .css for CSS files.

Selector :

A CSS selector selects the HTML element

which is able to style (or) design the

Selector {

Property : Value ;

}

[student] relates to student

Types of Selector :

- ↳ simple Selector
- ↳ Combinator Selector
- ↳ Attribute Selector
- ↳ Pseudo class Selector
- ↳ Pseudo Element Selector.

Simple Selector :

- ↳ Id Selector (#) - 1st Priority
- ↳ class Selector (.) - 2nd Priority
- ↳ Universal Selector (*) - 4th Priority
- ↳ Element Selector (tag) - 3rd Priority.
- ↳ Grouping Selector (combination of id, class, tag)

Combinator Selector :

Based on relationship targetting an element

- ↳ Descendent Selector (space) - direct and indirect children.
- ↳ child Selector (>) - direct children.
- ↳ Adjacent Sibling Selector (+) - Next Sibling.
- ↳ General Sibling Selector (~) - General Sibling.

Attribute Selector :

- * Attribute provides extra information to the tag.
- * Selector is used to select elements with a specified attribute.
- * Attribute selector with the help of attribute in to target an element.
 - ↳ Selector with attribute.
 - ↳ Element Selector [attribute]

Ex : img [src]

7 types of attributes

↳ Selector With attribute - Img [alt]

Select the element with value.

↳ Attribute with Value - Img [alt = "Value"]

Select the element with value.

↳ Attribute with caret ^ Symbol -

Img [attribute ^ = "Value"]

(check for prefix value).

↳ Attribute with \$ Symbol

Img [attribute \$ = "Value"]

(check for suffix value).

↳ Attribute with ~ Symbol

Img [attribute ~ = Value]

(select value containing a specified word

followed by space)

↳ Attribute with vertical ^ Symbol

Img [attribute ^ = "Value"]

(select value followed by hyphen).

↳ Attribute with * Symbol - Img [attribute * = "Value"]

(Select all value)

Difference b/w Pseudo class & Pseudo Element

Pseudo - class

Assist the Selection
Which can not be expressed
by Simple Selector.

Pseudo Element

Target Specified Part
of Element.

SYNTAX :

Selector : Pseudo Element

Selector : Pseudo class {

Prop - Value ;

}

Prop - Value ;

}

(Property value pair)

Pseudo classes :

Pseudo class is a selector that assist in the selection of something that cannot be expressed by a simple selector.

SYNTAX :

Selector : Pseudo-class {

Property : Value

}

Types :

↳ Anchor Pseudo classes

↳ UI element Pseudo classes

↳ Structural Pseudo classes

Anchor Pseudo classes :

↳ link

↳ Visited

↳ Active

↳ hover

↳ Enabled

↳ Disabled

↳ Focus

Structural Pseudo classes.

- ↳ First-child
- ↳ Last-child
- ↳ nth-child
- ↳ First-of-type
- ↳ Last-of-type.

Pseudo Elements :

styles specified (target) parts of an element.

SYNTAX :-

Selector :: Pseudo-Element {

 Property : value;

↳ ::first-line.

↳ ::first-letter

↳ ::before

↳ ::after

↳ ::marker

↳ ::selection.

hover : Pseudo class (:)

inside the hover use do operation.

a:hover {

 font-size : x-large;

}

* hover is used, when we touch the word it will perform the operation property & after that it will back to old form.

(Noot Noot)

In this we have opacity.

Opacity : 1 → It is used to when not transparent at all. (default)

opacity : 0.1 → completely transparent.

Text Property :

This text is styled with some of the text formatting Properties.

Text formatting :

- color (change the color of the text)
- Text-align.
- Text-transform (capitalize, upper case, lower case)
- Text-shadow (add shadow to text)
- Text-decorating [decorate the content
(line, style, color)]
- letter-spacing [Space b/w individual letters]
- word-spacing [Space b/w words]
- Text-indent.

Task :

- Text-Overflow.
- word-wrap [allow long words to break & wrap in next line]
- word-break.
- writing-mode [series of text in horizontal
(or) vertical]

Background Property :

- Background-image : url("./image.jpg")
- Background-repeat : no-repeat | repeat-x | repeat-y | round
- Background-size : contain | cover | 100% | x y Values
(100% 100%)

- Background - Position : right | left | center | top | bottom.
- Background - attachment : scroll | fixed.
- Background - clip : content | border | padding box.

Font Property :

- Font - Size : large | small | medium
- Font - weight : bold | bolder | lighter | normal .
- Font - style : italic | normal .
- Font - family : serif | sans-serif
- Font - variant : small - caps | normal .

Height and Width Property :

- ↳ Max - width : px
- ↳ Max - height : px
- ↳ Min - width : px
- ↳ Min - height : px
- ↳ width : px
- ↳ height : px .

Max - height }
height } overflow (fixed)

Min - height } default
Min - width }

Overflow Property :

- ↳ default in visible state .
- ↳ To hide overflow text (hidden)
- ↳ To use hidden text (scroll)

Max - Width } → Same.

Width

Min - Width - Not apply for block & inline level element but we can use.

[display : inline-block] → allows to set a width and height of an element

- * This property can able to give min-width
- * We can't give width to the block and inline but we can able to give to inline-block element.
- * width not get overflow instead it will increase the height.

Font - Property :

- ↳ Font - Size : large | small | Medium
- ↳ Font - weight : bold | border | lighter | normal
- ↳ Font - Style : italic | normal
- ↳ Font - Family : serif | sans-serif
- ↳ Font - Variant : small - caps | normal .

Box - Model :

If you want to space in and around HTML elements.

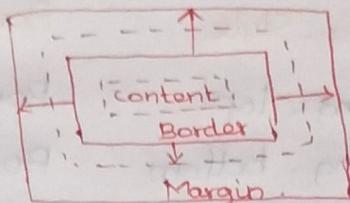
Box model allow us to design an around html element and give space b/w the element.

Three Types :-

Padding → Space b/w Content & element border.

Margin → Space b/w imaginary line & element border.

border → Lines around html element.



Padding :-

Space b/w Content and element border.

Four Types of Padding :-

Padding - left

Padding - right

Padding - Top

Padding - bottom.

Short hand Property :-

Set several isolated properties with one declaration.

	Top	Right	Bottom	Left
Padding	45px	25px	30px	25px

Top Left

45px 25px

Bottom Right

Padding 45px (common for all sides)

Margin :

Space b/w imaginary line and element border.

Properties :

Margin - Style

Margin - top
Margin - bottom
Margin - left
Margin - right

Padding - Style

Padding - top
Padding - bottom
Padding - left
Padding - right

Border - Style

Border - color
Border - width
Border - radius
(The lines around html element)

Position :

If you want to position the element anywhere in the webpage.

The position property specifies the type of position method used for element.

Default : Static. positioned elements are not affected by the

- ↳ top
- ↳ bottom
- ↳ left
- ↳ Right

Relative : positioned relative to its normal position.

Absolute : Positioned relative to the nearest positioned ancestor.

TYPES :

Absolute → Fixed

Relative → Sticky.

Margin :

Space b/w imaginary line and element border.

Properties :

Margin - Style

Margin - top
Margin - bottom
Margin - left
Margin - right

Padding - Style

Padding - top
Padding - bottom
Padding - left
Padding - right

Border - Style

Border - color
Border - width
Border - radius
(The lines around
html elements)

Position :

If you want to position the element anywhere in the webpage.

The position property specifies the type of position method used for element.

Default : Static - positioned elements are not affected by the

↳ top

↳ bottom

↳ left

↳ Right

Relative : positioned relative to its normal position.

Absolute : Positioned relative to the nearest positioned ancestor.

TYPES :

Absolute → Fixed

Relative → Sticky.

Absolute

Consider Grandparent as Margin Opp.

Absolute Share the Space for next element.

Relative

Consider parent margin as Opp.

Relative will not Share Space for next element.

Fixed → In this the position we can't move it by margin-size (or) other position like absolute.

Relative.

Sticky → If we scroll the page the element will sticky to that position.

Position Property :

The position property specifies the type of positioning method used for element.

Static : Elements are positioned static by default.
Static positioned elements are not affected by the top, bottom, left, and right properties.

Sticky : Positioned based on the user's scroll position.

Fixed : It always stays in the same place even if the page is scrolled.

Float → float is used for positioning and formatting content.

Ex : Set an img float left to the text in container

Float, Display & Overflow :

↳ Float - Right / Left / none

↳ Display - none / block / inline / inline-block

↳ Overflow - hidden / visible / scroll.

Transform :

* If you want to move the element from one place to another place.

* The property value which is working on 2 dimensional (2D)

* The property value which is working on 3 dimensional (3D)

* Translate accept 2 Value x Value y value.

Translate :

It will translate the element from one place to another place.

↳ Scale also used for 2D, we give value at this points. By default it is in (1,1)

[Scale increase the size of the particular element]

If we want to change the angle we use Skew we pass this in degree format.

Rotate () :

→ It is for 3D

→ If we give values in degree it rotate the particular element.

Box-Shadow : x-axis, y-axis, darkness color.

2D, and 3D Transforms :

Transforms allow you to move, rotate, Scale, Skew the element.

Difference :

	circle	Square	Rectangle
2D			
3D			
	Cylinder	cube	cuboid.

2D Transforms :

- 1) Translate () - Moves an element from its current position.
- 2) Rotate () - Rotate an element clockwise / counter-clockwise according to degree.
Using -ve value will rotate the element counter-clockwise.
- 3) Scale () - Increase / Decrease the size of element.
- 4) Scale x () - Increase (or) decrease the width of an element.
- 5) Scale Y () - Increase (or) decrease the height of an element.
- 6) Skew () - Skew an element along the x and y-axis by the given angles.
- 7) Skew x () - Skews an element along the x-axis by the given angle [degree 0-360]
- 8) Skew y () - Skews an element along the y-axis by the given angle.

3D Transforms :

Rotate x () - Rotates an element around its x-axis.

Rotate y () - Rotates an element around its y-axis.

Rotate z () - Rotates an element around its z-axis.

ANIMATION :

→ animate transitions from one CSS style configuration to another.

(or)

→ CSS allows animation of HTML elements an animation lets an element gradually change from one style to another.

Animation Property :

→ @ keyframes

→ animation-name

→ animation-duration

→ animation-delay

→ animation-iteration-count

→ animation-timing-function

→ animation-direction

If you want to use animation we have to use
on Syntax.

@ keyframes identifiers :

from {

}

to {

}

@ keyframes identifiers {

0% {

100% }

80%

}

}

→ animation-duration is important

animation-duration : 2s

animation-iteration : count-infinite.

animation-name : identifier;

TRANSITION :

Time given for the element to perform some action.

(or)

Transition gives you the period of time to make some action on elements.

Transition Property :

↳ Transition

↳ Transition-delay

↳ Transition-duration

↳ Transition-timing-function.

Easye → Slow - fast - slow

Ease in → Slow - start

ease-out → Slow - end

ease-in-out → Slow start, Slow end

Linear → Same Speed throughout.

FLEX :

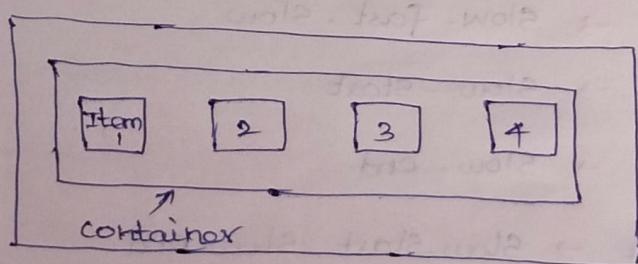
- ↳ The flex is a layout Module.
- ↳ When you trying to use display : flex; You can access of flex property is known as layout module.
→ (One Dimensional)
- 1D = Flex will help to aligned the element either in x axis (or) y axis.
- ↳ Flex is responsive Webpage.
[Based on Screen size will automatically align the Element]
- ↳ For Every flex, flex item, height and width is important.

1. parent div is considered as flex container.
2. child div is considered as flex item.

Flex : (Rules)

The flexible box layout, makes it easier to design flexible responsive layout structure without using float (or) positioning.

Flex box :



Flex Properties :

Display : flex

Flex-direction : row / column

Flex-wrap : wrap / no-wrap / wrap-reverse

Justify-content : flex-start / flex-end / center / space-around / space-between / space-evenly.

Align-items : flex-start / flex-end / center / stretch.

Flex-basis : <length>

Align-self : flex-start / end / center.

Flex-grow : <number> 0

Flex-shrink : <number> 1

Order : <integer>.

layout module :

When you trying the display flex then only we can have the process of flex property.

GRID :

- * Any horizontal line or any vertical line cross each other and it is performing one square shape is known as grid.
- * Grid is a layout module.
- * Whenever we want to use grid display : grid then only you can access grid property is known as layout module.

- * Grid will work on 2D (2 Dimension)
- * Grid is a responsive Webpage.
[Based on ScreenSize will automatically aligns the elements]
- * Every Grid template is important.

Row Line → The line between two rows.

Column Line → The line between two columns.

Row Gap → The space between two rows.

Column Gap → The space between two columns.

Fr → Fractional Unit.

Grid :

CSS Grid Layout module provides grid based layouts with rows & columns. It is to design web pages without having to use floats and positioning.

Grid Rows & Columns :

* The vertical lines of grid items are called columns.

The horizontal lines of grid items are called rows.

The space between each column/row are called gaps.

Grid Lines :

- * The lines b/w columns are called column lines.
- * The lines b/w rows are called row lines.

Grid Property :

- display : grid;
- grid-template-columns : auto auto auto auto;
- grid-template-rows : auto auto auto auto;
- justify-items : space-around, evenly, around
- column-gap : 20px.
- row-gap : 20px.
- gap : even space for column and row.
- Grid-row and Grid-column
- Grid-area.

.item1 {

grid-area : header;

}

.grid-container {

grid-template-areas : "header header header"

"Menu Menu Menu"

"right footer footer";

