

5/8/23
 ★ Internet: It is a network of networks which connects computers all over the world which through which we can share information.

A History of internet:

→ In 1969 : ARPA NET (Advanced Research Project Agency Network)

→ IEEE (Institute of Electrical and Electronics Engineers)

PAN:- Personalized Area Networks

CAN:- Campus Area Networks

→ In 1989, internet became available for commercial use.

Logical Address: IP Address

Physical Address: MAC Address

→ Basic services provided by internet:-

Protocols are set of rules.

Service

Electronic Mail

- FTP - File Transfer Protocol

- Telnet - remote login using password

- Usenet News

- HTTP : web access

- HTTPS : provides security

- TCP - Transmission Control Protocol

T.C.P (Transmission Control Protocol):
 T.C.P has a purpose of sending data from one device to another device cell over the internet. It is a set of rules governing how data is send and received over the internet responsible for addressing and routing.

WWW (World Wide Web): It is a huge collection of pages of information linked to each other around one globe.

→ Every page is a combination of text, picture, video, audio, animation and hyperlink.

→ Introduced by Tim Berners Lee.

Web page: It is a collection of normal text, picture, video, audio and hyperlink. It can be designed with HTML, XML or javascript, CSS etc.

HTML - Design structures

Javascript - Validates the field

→ XML is an advanced version of HTML.
 Extensive markup language.

Website: It is a collection of interlinked web pages website is accessed through URL (Uniform Resource Locator). URL is a global address of web document or website. URL are unique in nature, may don't have

copy.

→ Ex: ~~https://google.com~~~~https://google.com~~

Domain

~~https://www.google.com~~

Name

Protocol

URL



URL :- Categories of websites

→ There are mainly three categories of website

(1) Personal (2) Commercial and (3) Government.

(1) Personal - Portfolio website, information website.

(2) Commercial : Amazon, Myntra, flipkart.

(3) Government : ACPC, CBSE, Aadhar

★ Further there are more 2 web types of websites.

(1) Static information Site :- Blogging website, Article

(2) Dynamic interactive site :- Amazon, Myntra

Stock market websites.

★ Web Application :- You can run web application into the browser through URL.

→ There are two types of web application

(1) Service oriented:-

(2) Presentation oriented

Web applications

(1) Service oriented: It is used to implement web services. It is coded using CGI, JSP, ASP. Ex: Currency calculator, Bills.

(2) Presentation oriented web applications: It provides client side services. They are coded using HTML, XML, Javascript etc, for W3School, GeeksforGeeks, Geek4Geeks.

* Web Architecture: It follows two-tier architecture, it is a combination of web server and client.

→ Web servers produce and deliver information where as web clients retrieve and display information.

HTML (Hyper Text Markup Language)

→ HTML is a markup language for creating web pages.

→ It consists of various tags used to structure and present content.

→ HTML is case-insensitive and supports both lowercase and uppercase letters.

→ HTML uses tags to define different types of content.

→ HTML uses tags to define different types of content.

→ HTML uses tags to define different types of content.

→ HTML uses tags to define different types of content.

→ HTML uses tags to define different types of content.

→ HTML uses tags to define different types of content.

→ HTML uses tags to define different types of content.

Markup Language vs Programming Language

Aspect	Programming Language	Markup Language
Purpose	Implements logic, Algorithms, Structures, formats and dynamic behaviour	Formats, presents data.
Execution	Compiled or Interpreted, Executed by computer	Interpreted (by browser or other software) not executed as code.
Logic/Commands	Supports variables, conditionals, loops, functions	Uses tags to annotate and format content
Ex.	Java, C++, Python, JS	HTML, XML, Markdown
Usage	Software development	- Data representation

* Assignment:

- Top few pages
- Photo - Intro - Contact

* Home Page: The main or introductory page of a website.

* Navigation menu: Allows users to move between different sections or pages of the website.

* Content Pages: Pages containing information

~~alt~~, ~~title~~, ~~text~~, ~~host~~, ~~hyperlink~~, ~~reference~~

CLASSMATE
Date _____
Page _____

articles, images and multimedia elements.

* footer: Contains additional links (copyright information), and sometimes contact details.

* HTML:-

<!DOCTYPE html> (HTML document declaration)

<html> (HTML document root element)

</html> (HTML document end element)

<head> (HTML document header element)

</head> (HTML document header end element)

<body> (HTML document body element)

</body> (HTML document body end element)

<title> (HTML document title element)

</title> (HTML document title end element)

<h1> (HTML document heading element)

</h1> (HTML document heading end element)

<h2> (HTML document heading element)

</h2> (HTML document heading end element)

<h3> (HTML document heading element)

</h3> (HTML document heading end element)

<h4> (HTML document heading element)

</h4> (HTML document heading end element)

<h5> (HTML document heading element)

</h5> (HTML document heading end element)

<h6> (HTML document heading element)

</h6> (HTML document heading end element)

 (HTML document bold element)

 (HTML document bold end element)

<i> (HTML document italic element)

</i> (HTML document italic end element)

<s> (HTML document strikethrough element)

</s> (HTML document strikethrough end element)

Cu>

<h> 2

<h> 3

<h> 4

<h> 5

<h> 6

<h> 7

<h> 8

<h> 9

<h> 10

<h> 11

<h> 12

<h> 13

<h> 14

<h> 15

<h> 16

<h> 17

Content: The underline of the specific content.

<u></u>: It underlines the content.

<h></h>: It is used for making headings with different types of web pages. It has different types. <h1><h2><h3><h4><h5><h6>

* Standalone tag

* Links and Anchors:

(1) Links: Creating hyperlinks with <a> anchor tag.

Example: The word "Home" is linked to index.html.

 Visit Example

* Using anchor tag for In Page Navigation

Ex:

 Jump to Section 2

<h2 id="section 2"> Section 2 </h2>

→ Develop recommendation Systems

→ Predictive Analytics

→ Father of AI is Alan Turing
John McCarthy, Marvin Minsky

→ ML is a type of AI that enables systems to learn from data and improve performance on specific tasks

→ Uses algorithms to analyze data, identify patterns

→ DL is a subset of ML that uses neural networks with multiple layers to learn complex patterns

→ AI Pioneers : Durable AI ; Mix AI,
20 Web , XAI , OpenAI

→ Sundar Pichai CEO of Alphabet and Google

→ Satya Nadella : Chairman and CEO of Microsoft

* Inserting Images with `` Tag

Ex:

```

```

→ Image attributes

• `src` = image file name
`alt` = alternative text for the image
`width` = "400" `height` = "300"
`align` = "left"

→ Creating forms with HTML

```
<form> or <form>  
  <i> form elements go here -->  
</form>
```

* HTML Forms

```
<form action="/submit" method="post">  
  <i> form elements go here -->  
</form>
```

→ The `action` attribute specifies where to send the form data.

→ The `method` attribute defines how the data should be sent.

→ collect information of meta-data

* For Text fields:

Syntax: `<input type="text"/>`

→ The `<text area>` tag is an HTML tag used to create a text area within a form.

→ Text areas allow users to enter multiple lines of text (such as comments or messages).

→ To create a text area, use the `<text area>` tag.

→ It is a self-closing tag (meaning it does not require a closing tag).

* form attributes:

`<label for="message">Message: </label>`

`<text area id="message" name="message" rows="4" cols="30"></text area>`

★ Meta data means data about data.

→ It describes information about a file, webpage or dataset, rather than the actual content itself.

→ In a photo metadata includes details like camera type, location, time etc.

→ In a word file metadata includes author name, file size, creation date etc.

★ A meta tag is an `<meta>` tag that provides metadata about a webpage. Placed inside the `<head>` section of an HTML document.

★ Radio Button:

Syntax: `<input type="radio" name="radio_button_name">`

`value="radio_button_value">`

- Red
- Blue
- Green

★ Check box:

Syntax: `<input type="checkbox" name="select_box_value">`

A) Dropdown in html using form tag

<form>

<select name="language">

<option value="Indian"> Indian

<option value="American"> American

</select>

</form>



B) Tables

<table>

<!-- Table rows and columns go here -->

</table>

→ This is a table

→ Define the beginning and end of a table Attribute

e.g. <table border="1" style="width: 100%; border-collapse: collapse;">

width="100%" style="border-collapse: collapse;"

- Background

- bgcolor

- border

- cellspacing

- cellpadding

- width

- height

C) Table row

Attribute row "row" = <tr>

- align

- valign

- bgcolor

- Defines a data cell in a table

Attributes

- align
- valign
- height
- background
- bgcolor
- height
- width
- colspan
- rowspan

Table header <th>

- Acts just like a <td> ... <th> contains text or more tags
- The text is normally centered vertically and horizontally AND the text is bold

★ Iframes :

`<iframe src = "https://www.example.com/"></iframe>`

★ Controlling Iframe Dimensions

`<iframe src = "https://www.example.com/" width = "400" height = "300"></iframe>`

* Adjusting the Border and Frame

frame border Attribute : Use the frame border attribute to control whether the iframe has a border.

Values : frame border = "0" for no border,
frame border = "1" for a visible border

* Scrolling Attribute :

Values : scrolling = "yes" (default), scrolling = "no",
scrolling = "auto" (browser decides).

* src Attribute :

Example : <iframe src = "https://www-example.com"

CSS (Cascading Style Sheets) : It is a simple design language intended to simplify the process of making web pages presentable.

→ CSS is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.

⇒ Style :- Styles is anything that makes a simple object look more attractive and appealing to the viewer. It gives the object some properties like what color it should be of, where it is to be placed, size etc.

* Selected or not "a" = a href="document.html"

- Selectors target HTML elements to apply styles to

h1 { font-size: 24px; }

In this example the selector is "`h1`", which targets all `<h1>` elements in the HTML document.

* Declaration :- Declarations define the styles to be applied to the selected elements.

font-size: 24px; }

In this declaration, we are setting the font-size property to 24 pixels. This means that all `<h1>` elements in the HTML will have a font size of 24 pixels.

* Selector:

It is written as:

Element... → The element name which is to be styled

Example:

P... → Paragraph is styled

* Declaration block

→ Written as:

Element { property: value; }

y

Example:

P2

y

→ These brackets will tell what formatting options to apply on which element.

* Declaration is the part which is inside the declaration block { }. It consists of a property and value and ends with a semicolon

- written as:

Selector → Element {

Declaration → Property: value;

y

Example:

P2

color: red;

y

* Property:

selector → p

scope ↘

color: property

color: value;

font-size: value; ↗ property
 ↓ value ↗ color

*

Value: at its simplest ←

Example:

p

color: red; font-size: 30px;

y

Property
p ↗
color ↗ red;

Value

↓ ↗ blue;

background-color: blue;

→ 3rd value → P would contain both

background

color: blue;

→ 3rd part will

contain custome

Style Sheet (2.2) Inheritance

Inline

In the tag itself

Internal

In the webpage itself

External

In a separate file

Inline styles:

```
<html>
  <head>
    <style>
```

```
  <body>
```

`<h2 style="color: blue">` only this heading will have blue colour

(using inline style) `</h2>`

`<h2> This paragraph will be printed
(internal style) </h2>`

```
<style>
```

```
h2<
```

`color: red;`

`font-family: Arial;`

```
</style>
```

```
</head>
```

```
</body>
```

```
</html>
```

Internal CSS

<html>

<head>

<style>

```
color: #F F7643;  
font-family: Arial;
```

styles{
p{

color: red;

font-family: country;

written in the
webpage where
under the
heading tag

the style

tag tells

the browser

that the

part inside

FLB CSS

<style> inside <head>

<head> in parent

<body> in child

<div> in child

</body>

</html>

; { margin: 0; }

; { font-size: 1em; }

; { border: 1px solid black; }

; { padding: 10px; }

; { background-color: #f0f0f0; }

; { color: black; }

; { font-weight: bold; }

; { text-decoration: underline; }

; { text-align: center; }

; { margin: auto; }

; { width: fit-content; }

; { height: fit-content; }

; { border-collapse: collapse; }

; { border: none; }

; { border: 1px solid black; }

; { padding: 5px; }

Name	Info	Example
Universal	Any element.	* <code><font: 10px Arial; ></code>
Type	Any element of that type.	<code>h1 {text-decoration: underline;}</code>
Grouping	Multiple elements of different types	<code>h1, h2, h3 {font-family: verdana;}</code>
Class	Multiple elements of different types when you don't want to affect all instances of that type	<code>.sampleclass {text-decoration: underline;}</code>
Id	A single element type when you don't want to affect all instances of that type	<code>#sampleid {text-decoration: underline;}</code>
Descendant	An element that is below (in the document tree) another element no matter how many levels below.	<code>#gallery h1 {text-decoration: underline;}</code>
Child	An element that is directly below (in the document tree) another element	<code>#firstchild {font-weight: bold;}</code>

* External CSS :

style.css

h1 { font-size: 2em; color: blue; }

font-family: verdana;

h2 { font-size: 1.5em; color: red; }

color: red; font-family: serif;

font-family: sans-serif; font-size: 160%;

3

* Types of fonts: Using different fonts for your content.

like style="font-family: 'Times New Roman', Times, serif;"> or `font-family: "Times New Roman", Times, serif;`

times new roman

times new roman serif

new times new roman

new times new roman serif

* These are some useful CSS text properties

Property

Description

Example

`text-indent` Applies first-line indent to paragraph

`text-align` Aligns text to left, right, center or both sides
`(justify)`

`text-decoration` Applies text effects namely underline, overline and overlining; line through

`letter-spacing` Specifies spacing between text characters

`text-transform` Capitalizes the first letter of each word (capitalize), converts all letters to uppercase (uppercase) or lowercase (lowercase)

★

Coloring Textbook 227 Notes Done in Visit ☆

Hexadecimal Color Notation

color #6600FF;

66 - specifies the amount of red

00 - specifies the amount of green

FF - specifies the amount of blue

RGB

R - amount of red

G - amount of green

B - amount of blue

color: #6600FF

rgb(100%, 100%, 100%)

color: rgb(255, 255, 255);

100% > 1 and 255 being max values

RGBA

In addition to RGB, A means alpha - it tells the opacity/transparency of the fact - value lies between 0-1 (0 = invisible, 1 = most transparent)

★

HSL and HSLA

H - hue value ranges from 0-360

S - saturation (% color) - 100% (bright)

L - lightness (% black) - 100% (white)

A - Alpha (0-1)

example:

`border: 2px solid black;`

`color: hsl(0, 100%, 50%);`



* Page Border and Margin:

Each of these are CSS properties whose values can be set using appropriate units.

`border: 1px solid black;`

`margin: 10px;`

`padding: 10px;`

`padding-left: 75px;`

`padding-right: 75px;`

`border-radius: 10px;`



* Border Style Values

`solid`

`dotted`

`double`

You can use the `border-radius` property to make rounded edges.



* Using the transform property

`transform: rotate(30deg);`

`transform: translate(200px, 50px);`

`transform: scale(1.5);`

`transform: skew(-15deg, 15deg);`

* styling in CSS

→ inherit - This element inherits the float value of its parent.

- (1) float: left;
- (2) float: right;
- (3) float: none;
- (4) float: inherit;

Left

Right

None

→ This will cause the element to float to the left within its containing element. Other elements following in the document flow will wrap around the left side of the floated element.

Ex:

<head>
<style>
img {

float: left;

→ by inherit

<style>

<head>

<body>

h2 { float: left; }

p { }

This will cause the element to float to the right within its containing element. Other elements following in the document flow will wrap around the right side of the floated element.

→ elevation

<head>

<style>

img {

float: right;

background-color: yellow;

</style>

<head>

<body>

h2 { float: right; }

p { }

</p>

This is like default value, and it means that the element does not float. It remains in the normal flow of the document.

→ Head

<head>

<style>

img {

float: none;

background-color: yellow;

</style>

<head>

<body>

h2 { float: none; }

p { }

</p>

```
<p>
```

```
<body> here is overall width 100%
```

* Navigation Bar:

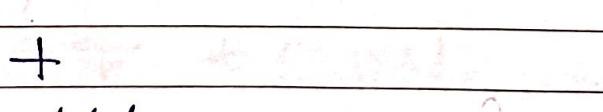
```
→ <body> do not have to put initial value (%)
```

```
<p>
```

```
<ul> position: absolute; left: 18px;
```

```
<li> class="active" href="#">
```

```
<a href="#"> News </a><li>
```

→ Here, we added  to the right

we added have added

Q1 Which tag is used to create a hyperlink in HTML?

Ans <a>

Q2 Which of the following is not an HTML tag?

Ans Sequence

Q3 Which HTML tag is used to display an image?

Ans

Q4 HTTPS is more secure than HTTP because?

Ans It uses SSL/TLS to encrypt communication.

Q5 In the CSS box model, what does padding represent?

Ans Space inside borders but around content.

A Default layout of Navbar

```

<link rel='stylesheet' href='stylesheet.css' />
</head>
<body>
    <div class='page'>
        <div class='menu'> menu </div>
        <div class='sidebar'> sidebar </div>
        <div class='content'> content </div>
        <div class='footer'> Footer </div>
    </div>
</body>
</html>

```

⇒ External css:

stylesheet.css:

html { height: 100%; }

body { font-size: 16px; }

margin: 0;

padding: 0;

box-sizing: border-box;

menu {

height: 60px;

background-color: #B2D6FF;

sideBar {

height: 60px;

background-color: #F0F8D0;

content {

height: 100px; border: 1px solid black;
 background-color: #F5C8E;

2 footer {

height: 100px;

background-color: #D6E9FF;

3 }

How position values Definition > and Syntax

- static
- relative
- fixed
- absolute →
- sticky

It placed nearest element from that element with property

position: absolute;

top: 10px; right: 10px; bottom: 10px; left: 10px;

left: 30px;

Relative:

2

position: relative;

left: 10px; top: 10px; bottom: 10px; right: 10px;

3

Start point

A) Transform Property:

Property in CSS is used to reposition an element on the 2D plane (x and y axis).

B) z-index:

How can we position an element with respect to other elements?

With transform property, we can't do this.

With CSS, we can position elements.

div {
 position: absolute;
 top: 0;
 left: 0;
 width: 300px;
 height: 200px;
 background-color: red;
}

position: absolute; top: 0; left: 0;

top: 0; left: 0; position: absolute;

position: absolute; top: 0; left: 0;

CH-3 Javascript

- ★ Javascript: Runs on interpreter.
- Javascript has object oriented capabilities
- OOPS:
 - Design Pattern, Polymorphism, Encapsulation, Inheritance, Abstraction, constructors, Deconstructors, inline function, Access modifiers.
- Javascript is an untyped language.
- It adds functionality.
- It can accessed internally as well as externally.
- ★ Javascript variables: Javascript variables are containers for storing data values.
- Creating a variable in Javascript is called declaring a variable. (var)
- You declare a javascript variable with the 'var' keyword.
- All javascript variables must be identified with unique names.
- These unique names are called identifiers.
`var a = 3;`

Assignment

* Sum of two numbers:

```
var a = 10;
```

```
var b = 20;
```

```
var c = a + b;
```

```
c = c + a;
```

```
console.log(c);
```

* Data types in javascript:

→ Data types in javascript are undefined and null "hello".

undefined & null: "hello"

① If you want to utilize single ' or double " or apostrophe record in " how could you utilize it in addition to revert "

→ Script:

```
script="text/javascript">
```

```
var a;
```

```
console.log(a)
```

* prompt: to return address + password

```
prompt("password")
```

* prompt Box:

```
prompt("creat image")
```



confirm Box dimension. It will be (5).

confirm ("username")

→ /"Hello World"/

★ conditional statement:

→ if else statement is used to execute the code whether condition is true or false. There are three forms of if statement in javascript.

(i) if statement:

```
var a = 20;
```

```
if a > 10 {
```

```
    document.write("value of a > 10");
```

y

(ii) if else statement:

```
var a = 20;
```

```
if % 2 == 0 {
```

```
    document.write("a is even number.");
```

```
else
```

```
    document.write("a is odd number.");
```

(3) if else if statement in constructor A

~~Var~~

```
var a = 20;
if a == 10;
```

L

"a is greater than 10"

"throw off H"

document.write("a is equal to 10")

: transmission, Inheritance

else if

else if a == 15; ai transmission, inheritance

document.write("a is equal to 15")

else if

a == 20;

: transmission, inheritance (2)

document.write("a is equal to 20")

else

document.write("a is not equal to 10, 15, 20")

→ if else code for tossing coin

if toss coin == head;

else if

document.write("the value is 1")

else

document.write("the value is 0")

→ if statement code:

if age > 18;

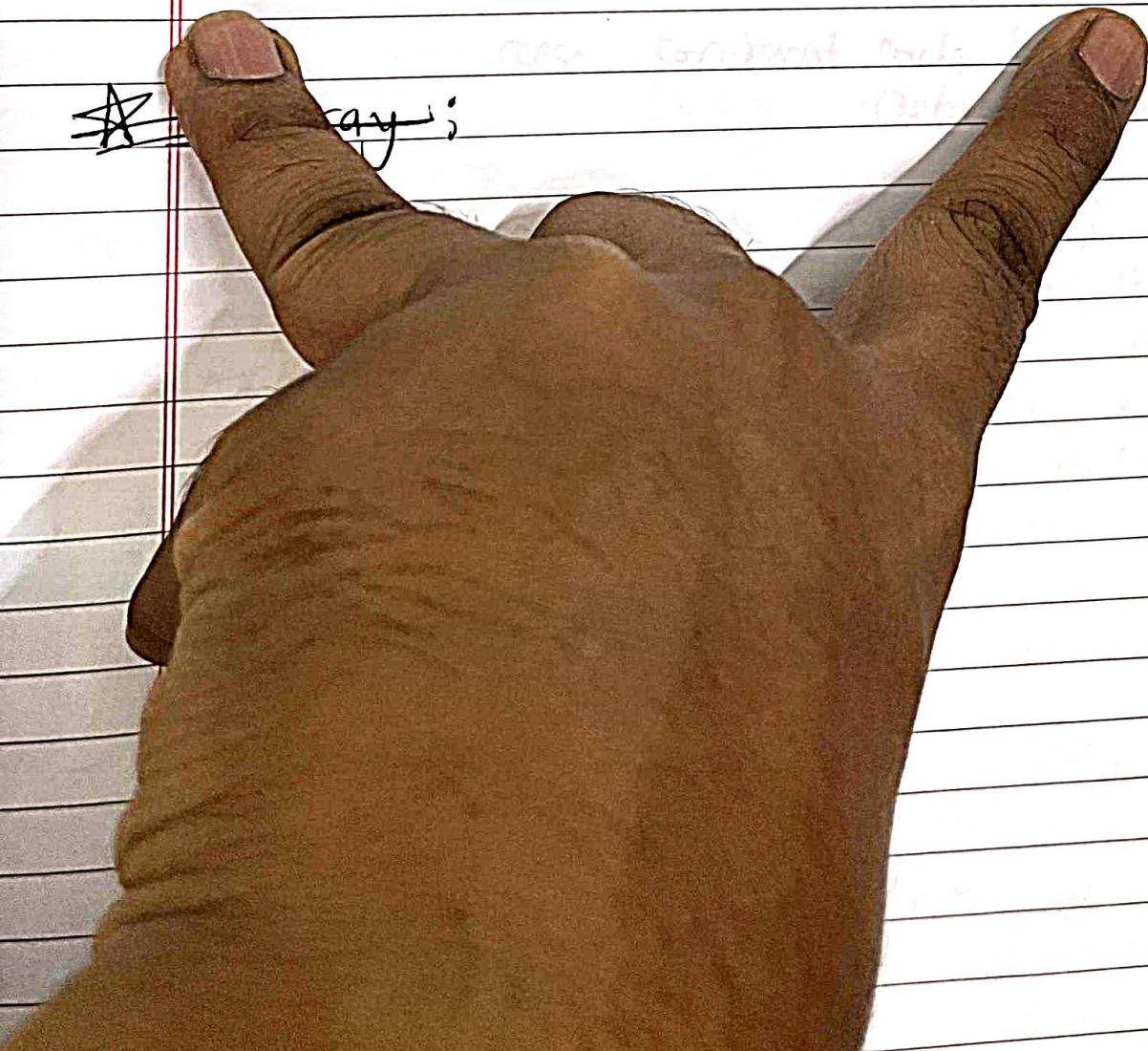
do

statement 2

document.write ("you have voting
rights")

y

~~say;~~



31/1/25

classmate

Date _____

Page _____

* Switch statement: Switch statement is used to select one or many blocks of code to be executed.

Syntax:

```
switch (expression) {  
    case constant1: code1;  
    break;  
    case constant2: code2;  
    break;  
    default: code;}
```

* Iterative statements: Iteration, literally means repetition of a process, so iterative statement or constructs, that helps in repeating some statements for a defined number of times.

The different iterative statements of 3 types:

1, while loop

2, do-while loop

3, for loop

→ Loops can execute a block of code as long as specified condition is true.

① The While loop: The while loop, loops through a block of code as long as a specific condition is true.

Syntax

→ To know what a while condition will do we have to see what happens if the condition is true.

→ Do while loop: Do while loop is just like a while loop, only difference is that it executes at least once irrespective of whether the condition is true or false.

Syntax:

```
(Do While)
```

```
while (condition) { code }
```

All the statements written inside the loop will be executed till the condition is true.

→ for loop: The JavaScript codes are also repetitive and number of times so, for loop can also be used to reduce the complexity of the code.

Ex: for (i = 1; i < 10; i++) {
 document.write(i + " by ");
}

★ Nested loops: Nested loop is nothing but a loop inside a loop.

Actual code: for (i = 1; i < 5; i++) {

for (j = 1; j < 5; j++) {

console.log(i + " " + j);

- ① functions: A javascript function is a block of code designed to perform a particular task.
- A javascript function is executed when something invokes it (calls it).
- Javascript functions are defined with a "function" keyword followed by a name, followed by a "()" parenthesis.
- ② function sum():
 - function names can contain letters, digits, underscores, and dollar sign.
 - The parenthesis may include parameters separated by commas.
- without parameters:

```
function sum (parameters);
```
- with parameters:

```
function sum (var a, var b);
```

var a=10;
var b=20;

```
var add = a+b;
```
- document.write(sum());

★ Parameters passing with function (continued)

Sum(10, 20); is a direct assignment
Function sum(a+b); is doing something

2

the line var add = a+b; is doing something
document.write(add); is doing something

3

function(); is doing something

★ Using return keyword (Code reusability and extensibility).

Ex: make rollab like a calculator

var stored = sum(10, 20); is doing something
document.write(stored); is doing something

function sum(a+b); is doing something

{ Extending it will be difficult

var add = a+b;

return add;

y

★ Arrow function (multiline arrow functions)

Ex: const myFunc = () => {
 console.log("Hello");
 return "Hello";
};

Hello = () => {
 console.log("Hello");
 return "Hello";
};

y

{ (current) after return
return "Hello";

y

(2) Single line arrow function

Syntax

`let myFunction = (a, b) => a * b;`

(3) Arrow function with parameters.

`var hello();``hello = (val) => "Hello " + val;
null ("Hello");`

* Events: The change in the state of an object.
 Is known as an event in Javascript.

→ Event handling is a mechanism that controls events and decides what should happen when an event occurs. There are 3 main events in javascript.

- Mouse event (click) = on(click)
- Key board event (key down and up) = onkeyup and onkeydown

→ Window events (load) = onload

* Keyboard events: works only in form and body tag.

* window events: window event works only in body tag.

★ Arrays in javascript: Array is an object that represents a collection of similar types of elements (A type of data structure).

→ Array is a special variable which can hold more than one value.

→ 2 ways to make an array: ways (E)

~~var emp = ["S", "V", "R"];~~ now
~~emp[0] = "S";~~ ~~emp[1] = "V";~~ ~~emp[2] = "R";~~

(1) ~~var emp = ["S", "V", "R"];~~ ~~it will~~

(2) ~~var emp = newArray();~~ ~~it will~~

(2) ~~emp[0] = "S";~~

~~emp[1] = "V";~~ ~~it will~~ ~~it will~~

→ To delete values in Array use delete

~~delete emp[0] = "S";~~

→ To modify values use var & for.

~~emp[2] = "Rahul";~~

★ Array methods: Array functions or methods are prebuilt or predefined functions for performing

① ~~forEach~~ forEach :- Some specific task like identify, incr., sort, reverse etc. in an array.

(1) `forEach()` :-

```
const numbers = [1, 2, 3, 4];
```

~~numbers.forEach(function(item, index){~~
 ~~console.log(item + " at index " + index);~~
})

→ `forEach()` array function can also be used in this case if function is not called again.

* find sum of elements in an array use `forEach()`.

(2) `map()` :-

```
const numbers = [1, 2, 3, 4, 5];
```

`numbers.map(function(item){`
 return item * 2;

* Objects:- In javascript object is a collection of named values. Its just like an array only.

→ The named values in javascript objects are called properties. Objects are variables containing containing variables. Javascript variables can contain single values. Objects are variables too.

→ But objects can contain many values.

→ The values are returned as name:value pairs
Cname and value separated by 'colon :'

→ (1st) Method

```
<script> var A = {  
    first_name: "Razil",  
    last_name: "Christian",  
    age: 23,  
    profession: "Student"  
};
```

console.log(A); → returns object
A(first_name: "Razil", last_name: "Christian", age: 23, profession: "Student")
console.log(A.first_name); → It will print only first name value.

A.profession = "Software engineer";
console.log(A) → changes the value of property

→ To add new property

A.middle_name = " "

console.log(A) → since outside of this curly braces it is not part of object

→ To delete property

delete A.middle_name;
console.log(A)

→ To add new property

Object.defineProperty(A, "profession", {
 value: "Programmer",
 writable: true,
 enumerable: true,
 configurable: true
});

→ To change value of property

→ (2nd method)

```
var emp = new Object();
```

```
emp.id = 101;
```

```
emp.name = "Ravi";
```

```
emp.salary = 70000;
```

```
document.write(emp.id + emp.name + emp.salary);
```

* Array, functions and object used inside object.

→ Array inside object

(i) var data = {
 std: ['Ram', 'Rahul', 'Shubham'],
 fees: ['Function'
 ↗ return 53000;
 ↗ functions inside object] };

→ object inside object

```
book: { name: "RD Sharma",  
        "500", "y" }
```

* Array object:

```
var emp = [ { Name: "Rahul", Salary: 70000 },  
           { Name: "Razil", Salary: 60000 },  
           { Name: "Shivam", Salary: 60000 } ];
```

```
console.log( emp[2].name + emp[2].salary );
```

