**JAVASCRIPT**

Hight level interpreted lang(line by line )

Used to make web page as dynamic

Scripting lang: writing

Founder: developed by Brendon Eich in 1995

First name was Mocha 🡪 LiveScript 🡪 JavaScript(ES-6)

ECMA company created a rules called ECMA script by this js was used by version 6

**JavaScript** is a versatile and widely-used programming language primarily used to add interactivity, functionality, and dynamic behavior to web pages. It's one of the core technologies of web development, along with HTML and CSS.

FEATURES

* Doesn’t require any compilation to execute
* It has full integration with HTML, CSS
* Major browser supports JS which is enabled by default in browser engines 🡪Engine 8
* JS is case sensitive (different ASCII value)

WHERE TO WRITE / TO EXCUTE

We have 3 different ways

1. Directly inside the browser by “inspecting” page using “console”

2. Inside the HTML file by using <script> in both head 🡪 to link externally(we can use for backend) inside head , and body

3. By installing runtime like node.js

TO PRINT

1. console.log()-🡪 displayed output inside console

2. document.writeln() 🡪 directly display output on browser

Node -y

Javascript es6 code snippets

VARIABLES

**Variables** are named storage locations in memory used to store values that can change or remain constant during program execution.

These the memory location / container to the store the values

SYNTAX:

variable\_name=value;

Declaring variables

1. using var

* Used to declare global scoped variables (can use wherever in the code)
* Re initialization and re deceleration is possible
* Syntax: var name=value;
* Real-time we don’t use

2. using let

* To declare block scoped variables
* Re insitization is possible but not re- decelaration
* Syntax: let name=value

3. Using const

* Once declared can’t change
* Re-inst not possible and re-dec not possible
* Syntax: const name=value;

DATATYPES IN JS

* We have 7 types

1. string

2. number

3. BigInt

4. Boolean

5. Undefined

6. Null

7. Object

8.Symbol 🡪 not used

EXPRESSION

Combinations of operands and operations

4 types

1. arithmetic 🡪 +, -,/,\*,%,\*\*,++,--

++(increment) 🡪 2 types pre (increments and later print) and post (prints and later increments)

If a=10 🡪 ++a (11) and a++ (10)

--(decrement) 🡪 2 types pre (decrements and later print) and post (prints and later decrements)

--a and a--

2. assignment 🡪 =,+=,-=.\*=,/+,%=, \*\*=

3. comparison (interview) 🡪 difference btw pre and post and comparison

Compare the value with the variable 🡪 ==, !=,===(imp), !==,>,<,>=,<=, ?(ternary)imp

Diff btw == and ===

Ternary 🡪 works based on condition

Syntax: condition? state1: state 2;

If condition is true then statement 1 or statement 2

4. logical 🡪 AND - && (T – T 🡪 T) , OR || ( F-F 🡪F) , NOT ! (T 🡪 F )

CONDITIONS

Controlling the flow of program by using certain set of rules

1. if condition

2. if-else condition

3. else-if condition

4. nested if

5. switch case

PROMPT () USED TO GET USER INPUT

Define it and use

let prompt = require('prompt-sync')({sigint:true})

LOOP

1.for

For(start;end;step{

/Executable code }

2. for in

For (let iterator in collection){

///execution code }

3. for of

For (let iterator of collection){

///execution code }

4. while

initialization

while condition {

//code

update }

5. do-while

initialization

do {

//code

update

}while condition

FUNCTIONS

block of code that is modular and can be reusable

1. Name (user defined function)

Function name(parameters) {  
body}

Name (arguments)

2. Anonymous 🡪 which doesn’t name

variable=function() {

//body }

3. arrow (imp) 🡪 very fast

variable=() {

//body }

4. IIFE

(anomymous function) (call the function)

5.HOF

Myfunc(){  
//code }

Yourfunct (Myfunc ()) {

//code }

ARRAY DATA TYPE

Array are heterogenous and can store duplicate element, supports positive index

Array datatype is object

We cant sort the number so we use quick sort a-b and we have to conver into hof

Diff btw slice and splice, mutable

STRING

MUTTABLE

Pop up

3 types

1. altert

2. confirm 🡪 confirmation from user before performing some task

3. primpt 🡪 to get input from the user

DOM

DOCUMENT OBJECT MODEL

PROGRAMMING INTERFACE FOR HTML DOCUMENT

BY THIS WE CAN ACCESS AND MANIPULATE THE DATA

IT CAN B DONE BY id, class, tagname, name, more attributes

Which is by command and methods provided by dom

Dom is like tree like structure and contain branches and nodes

DOCUMENT

BODY

HEAD

DIV CLASS”TEXT”

TITLE

DIV ID=”PARA”

H1

object

P

P

Browser engine v8 will understand js and provide the text according the the webpage

COMMAND/METHODS

1. getElementById()

🡪 id is unique cant repeat

* Syntx: document.getElementByID(“id”) 🡪 it should be passed as string
* 4 methods

1. textContent 🡪 gets and send text contenet

2. InneraHTML

3. value

4. src

Difference btw textContenet and innerHTML

getEkementsbyClassName()

aceess and manipulate html elment by its class name

all class will be stored in node list

syntax:var = docment.getElementsByClassName(“class”)

EVENT HANDLING

Change in the state of an object

When js is loaded into html/browswer. JS will start reacting over these events and allows the execuation

This process of execuation is event handling

Js will handle these events via event handler

TYPES

1. onClick()

🡪it can make any tag as clickable ,when we click an event will be fired

2. Ondbclick()

3. onmouseover()

4. onmouseout()

5. onkeydown()

6. onkeyup()

7. onsubmit()

8. onchange()

9.onfocus

10. onblur

11. onload()

We have more

Event listerner

Element.addEventLisener(”event”,fun())

Database

2 types

1. local storage : permanent

Session storage: close the browser the data will be erased

* db provided by browser
* Data wont be send to server
* Stores in the form of of key and value
* Capacity -5 mb, total 10mb

Builtin

setItem()

getItem()

removeItem()

clear()