**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 }