**JavaScript Notes (03-02-2025)**

**General Information**

* Everything in JavaScript is an object.
* JavaScript is an Object-Oriented scripting language.
* It is **lightweight and interpreted** (memory binding at runtime).
* **Case-sensitive** language.
* **Weakly typed** (does not require explicit data types).
* Supported by all browsers and operating systems.
* Created in **May 1995** by **Brendan Eich**, initially named **Mocha**.

**Features**

* Makes websites dynamic and interactive.
* Executes directly in the browser.
* JavaScript and Java are **completely different**.
* All scripting languages ultimately translate into JavaScript.
* Syntax is similar to C programming.

**Applications**

* Client-side validation.
* Dynamic drop-down menus.
* Displaying date and time.
* Pop-ups and dialog boxes.

**Tokens**

* The smallest unit of code in JavaScript.

**Variables in JavaScript**

* Used to store values with memory references.
* **Keywords:** var, let, const.
  + var was used from **1995-2015**.
  + let and const were introduced in **2015**.

**Hoisting**

* Brings the variable/function declaration to the top of the function.

**Example:**

var b = 100;

function test() {

alert("value of b=" + b);

let c = 100;

window.c = 10;

}

test();

test1();

function test1() {

alert("value of c=" + window.c);

}

**Data Types**

**Primitive**

* String, Number, Boolean, Null, Undefined.

**Non-Primitive**

* Object, Array, Function.

**Stack and Heap Memory**

* **Stack:** Dynamic.
* **Heap:** Static.
* **Non-primitive data types** (objects, arrays, functions) are stored in **heap memory**.

**Arrays in JavaScript**

* Stored in square brackets [].

**Ways to Create an Array**

**By Literal:**

const names = ["Kashish", "Simer", "Diksha"];

for (let i = 0; i < names.length; i++) {

document.write(names[i]);

}

**By Creating an Instance:**

const cities = new Array();

cities[0] = "Jalandhar";

cities[1] = "Patiala";

cities[2] = "Bathinda";

for (let i = 0; i < cities.length; i++) {

document.write(cities[i] + "<br>");

}

**Checking Availability in an Array:**

function available(cities, cityToCheck) {

if (cities.includes(cityToCheck)) {

document.write(cityToCheck + " is available<br>");

} else {

document.write(cityToCheck + " is not available<br>");

}

}

const cityList = ["Jalandhar", "Bathinda", "Patiala", "Amritsar", "Ludhiana"];

available(cityList, "Jalandhar");

available(cityList, "Sangrur");

**Objects in JavaScript**

* Written using curly braces {}.
* Objects are **variables** that can contain multiple values.

**Ways to Access Object Properties:**

* objectName.propertyName; → person.firstname
* objectName["propertyName"] → person["firstname"]

**Example:**

const person = {

id: 101,

name: "Kashish",

age: 21

};

document.write("ID: " + person.id + " Name: " + person.name + " Age: " + person.age);

**Creating Objects Using Constructors:**

function Teacher(id, name, age) {

this.id = id;

this.name = name;

this.age = age;

}

const obj1 = new Teacher(301, "Simer", 21);

document.write(obj1.id + " " + obj1.name);

**Functions in JavaScript**

function test() {

alert("I am a synonymous function");

}

test();

const test1 = function() {

alert("I am an anonymous function");

};

test1();

* **Hoisting** is **not possible** with **anonymous functions**.
* Hoisting is **allowed** in var but **not** in let.

**Switch Case Example:**

function getGrade(marks) {

switch (true) {

case (marks >= 90 && marks <= 100):

document.write("Congratulations, you got O grade");

break;

case (marks >= 80 && marks < 90):

document.write("Congratulations, you got A grade");

break;

case (marks >= 70 && marks < 80):

document.write("Congratulations, you got B grade");

break;

case (marks >= 60 && marks < 70):

document.write("Congratulations, you got C grade");

break;

case (marks >= 0 && marks < 60):

document.write("Congratulations, you got D grade");

break;

default:

document.write("Invalid Marks");

}

}

let marks = 100;

console.log(getGrade(marks));

**Loops**

**For-in Loop:**

const person = {

id: 101,

name: "Diksha",

age: 21

};

for (let p in person) {

document.write(person[p]);

}

**Arrow Functions in JavaScript**

* Also known as **lambda expressions**.
* Java uses **thin arrow (->)**, while JavaScript uses **fat arrow (=>)**.

const test1 = (a) => a;

document.write(test1(100));

**Optional Parameters**

**Default Parameters:**

function sum(a = 5, b = 7) {

return a + b;

}

sum();

sum(10);

sum(10, 10);

**Rest Parameters:**

function show(...args) {

let sum = 0;

for (let i of args) {

sum += i;

}

console.log("Sum = " + sum);

}

show(10, 20, 30);

**Spread Operator:**

* Used for array concatenation, shallow copying, and merging objects.