JAVASCRIPT

1. JavaScript Introduction

Question 1: What is JavaScript? Explain the role of JavaScript in web development.

Answer:

JavaScript is a high-level, interpreted programming language used to make web pages interactive and dynamic.

Role in Web Development:

Adds interactivity to web pages (e.g., buttons, forms, sliders).

Updates web content dynamically without reloading the page.

Works with HTML and CSS to create responsive and engaging websites.

Question 2: How is JavaScript different from other programming languages like Python or Java?

Answer:

JavaScript runs mainly in web browsers, while Python and Java run on computers or servers.

JavaScript is mostly used for web development, Python for general programming, and Java for apps, web, and software.

JavaScript is interpreted, dynamically typed, and event-driven, whereas Java is compiled and strongly typed, and Python is interpreted but general-purpose.

Question 3: Discuss the use of <script> tag in HTML. How can you link an external JavaScript file to an HTML document?

Answer:

The <script> tag is used to add JavaScript code to an HTML page.

Inline JS: Write code directly inside <script>

<script>

alert("Hello!");

</script>

External JS: Link a separate file using src

<script src="script.js"></script>

2. Variables and Data Types

Question 1: What are variables in JavaScript? How do you declare a variable using var, let, and const?

Answer:

Variables are containers to store data.

Declaration:

var name = "Meet"; → function-scoped, older way

let age = 20; → block-scoped, can change value

const pi = 3.14; → block-scoped, value cannot change

Question 2: Explain the different data types in JavaScript. Provide examples for each.

Answer:

String: Text → "Hello"

Number: Numeric → 100, 3.14

Boolean: True/False → true

Undefined: Variable declared but no value → let x;

Null: Empty value → let y = null;

Object: Collection of key-value → {name: "Meet"}

Array: List of values → [1,2,3]

Symbol: Unique identifier → Symbol("id")

Question 3: What is the difference between undefined and null in JavaScript?

Answer:

undefined → variable declared but not assigned a value

null → variable explicitly set to have no value

3. JavaScript Operators

Question 1: What are the different types of operators in JavaScript? Explain with examples.

Answer:

Arithmetic Operators – Used for math operations

+ Addition → 5 + 2 = 7

- Subtraction → 5 - 2 = 3

\* Multiplication → 5 \* 2 = 10

/ Division → 10 / 2 = 5

% Modulus → 5 % 2 = 1

Assignment Operators – Used to assign values

= → let x = 5;

+= → x += 2; (x = x + 2)

-= → x -= 2; (x = x - 2)

Comparison Operators – Compare values

== Equal → 5 == '5' → true

=== Strict equal → 5 === '5' → false

> Greater → 5 > 3 → true

< Less → 3 < 5 → true

Logical Operators – Combine conditions

&& AND → true && false → false

|| OR → true || false → true

! NOT → !true → false

Question 2: What is the difference between == and === in JavaScript?

Answer:

== → Checks value only, allows type conversion

5 == "5" // true

=== → Checks value and type, no type conversion

5 === "5" // false

4. Control Flow (If-Else, Switch)

Question 1: What is control flow in JavaScript? Explain how if-else statements work with an example.

Answer:

Control flow → The order in which code executes.

If-Else: Executes code based on a condition.

let age = 18;

if(age >= 18){

console.log("Adult");

} else {

console.log("Not Adult");

}

Question 2: Describe how switch statements work in JavaScript. When should you use a switch statement instead of if-else?

Answer:

Switch: Checks a variable against multiple cases and executes the matching block.

let day = 2;

switch(day){

case 1: console.log("Monday"); break;

case 2: console.log("Tuesday"); break;

default: console.log("Other day");

}

Use switch → When checking one variable against many values; cleaner than multiple if-else.

5. Loops (For, While, Do-While)

Question 1: Explain the different types of loops in JavaScript (for, while, do-while). Provide a basic example of each.

Answer:

For Loop – Runs a fixed number of times

for(let i = 1; i <= 5; i++){

console.log(i);

}

While Loop – Runs while a condition is true

let i = 1;

while(i <= 5){

console.log(i);

i++;

}

Do-While Loop – Runs at least once, then checks condition

let i = 1;

do{

console.log(i);

i++;

} while(i <= 5);

Question 2: What is the difference between a while loop and a do-while loop?

Answer:

While Loop: Checks condition first → may run 0 times

Do-While Loop: Runs first, then checks condition → runs at least once

6. Functions

Question 1: What are functions in JavaScript? Explain the syntax for declaring and calling a function.

Answer:

Function: A block of code that performs a task and can be reused.

Declaration:

function greet(){

console.log("Hello!");

}

Calling a function:

greet(); // Output: Hello!

Question 2: What is the difference between a function declaration and a function expression?

Answer:

Function Declaration: Named function, can be called before declaration

function add(a,b){ return a+b; }

Function Expression: Function stored in a variable, cannot be called before definition

const add = function(a,b){ return a+b; }

Question 3: Discuss the concept of parameters and return values in functions.

Answer:

Parameters: Inputs to a function

function sum(a,b){ return a+b; } // a,b are parameters

Return Value: Output from a function

let result = sum(5,3); // result = 8

7. Arrays

Question 1: What is an array in JavaScript? How do you declare and initialize an array?

Answer:

Array: A collection of values stored in a single variable.

Declaration & Initialization:

let fruits = ["Apple", "Banana", "Mango"];

Question 2: Explain the methods push(), pop(), shift(), and unshift() used in arrays.

Answer:

push() → Adds element at the end

fruits.push("Orange");

pop() → Removes element from the end

fruits.pop();

shift() → Removes element from the start

fruits.shift();

unshift() → Adds element at the start

fruits.unshift("Strawberry");

8. Objects

Question 1: What is an object in JavaScript? How are objects different from arrays?

Answer:

Object: A collection of key-value pairs used to store data.

let person = {name: "Meet", age: 20};

Difference from arrays:

Arrays → ordered list of values

Objects → unordered collection with keys

Question 2: Explain how to access and update object properties using dot notation and bracket notation.

Answer:

Dot notation:

console.log(person.name); // Access

person.age = 21; // Update

Bracket notation:

console.log(person["name"]); // Access

person["age"] = 22; // Update

9. JavaScript Events

Question 1: What are JavaScript events? Explain the role of event listeners.

Answer:

Events: Actions that happen in the browser (e.g., click, hover, key press).

Event Listeners: Functions that wait for events and run code when the event occurs.

Question 2: How does the addEventListener() method work in JavaScript? Provide an example.

Answer:

addEventListener(): Attaches a function to an event on an element.

let btn = document.querySelector("button");

btn.addEventListener("click", function(){

alert("Button clicked!");

});

10. DOM Manipulation

Question 1: What is the DOM (Document Object Model) in JavaScript? How does JavaScript interact with the DOM?

Answer:

DOM: A tree-like structure representing HTML elements of a web page.

JavaScript Interaction: Can access, modify, add, or delete elements dynamically.

Question 2: Explain the methods getElementById(), getElementsByClassName(), and querySelector() used to select elements from the DOM.

Answer:

getElementById("id") → Selects single element by ID

let title = document.getElementById("heading");

getElementsByClassName("class") → Selects all elements with the class (returns HTMLCollection)

let items = document.getElementsByClassName("item");

querySelector("selector") → Selects first element matching CSS selector

let firstItem = document.querySelector(".item");

11. JavaScript Timing Events (setTimeout, setInterval)

Question 1: Explain the setTimeout() and setInterval() functions in JavaScript. How are they used for timing events?

Answer:

setTimeout(): Executes a function once after a specified delay.

setInterval(): Executes a function repeatedly at specified intervals.

Question 2: Provide an example of how to use setTimeout() to delay an action by 2 seconds.

Answer:

setTimeout(function(){

alert("This shows after 2 seconds");

}, 2000);

12. JavaScript Error Handling

Question 1: What is error handling in JavaScript? Explain the try, catch, and finally blocks with an example.

Answer:

Error Handling: Managing errors so the program doesn’t crash.

try: Code that may cause an error

catch: Runs if an error occurs

finally: Runs always, after try/catch

try {

let result = 10 / 0;

console.log(result);

} catch(error) {

console.log("Error occurred:", error);

} finally {

console.log("Execution finished");

}

Question 2: Why is error handling important in JavaScript applications?

Answer:

Prevents program crashes

Shows user-friendly error messages

Helps debug and maintain code