**JavaScript Introduction**

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

**Answer:**

* **JavaScript** is a **programming language** used to make web pages **interactive and dynamic**.
* It runs directly in the **web browser**.
* It helps in doing things like:
  + Showing **alerts or pop-ups**
  + **Validating forms**
  + **Creating animations**
  + **Updating content** without reloading the page
* In short, it makes websites **come alive** and **respond to user actions**.

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

**Answer:**

* **JavaScript** runs **inside the browser**, while Python and Java usually run on the **computer or server**.
* JavaScript is **loosely typed**, meaning you don’t have to declare data types.
* It is mainly used for **web development**, while Python and Java are used for many other purposes like AI, data analysis, or building apps.
* JavaScript is **interpreted**, not compiled like Java.

**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 in HTML to **add or run JavaScript code**.
* You can write JavaScript **inside the <script> tag** or **link an external file**.
* <script>

alert("Hello!");

</script>

* This is how we can write javascript in script tag
* <script src="script.js"></script>
* This is how we can link external .js file in our html file in head section.

**Variables and Data Types**

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

**Answer:**

* **Variables** in JavaScript are used to **store data or information** that can be used later in the program.
* They act like **containers** that hold values such as numbers, text, or objects.
* JavaScript provides **three ways** to declare variables:
  + **var** – used in older versions, has a wider scope.
  + **let** – used in modern JavaScript, has a limited (block) scope.
  + **const** – used to declare variables whose values **cannot be changed**.

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

**Answer:**  
JavaScript has several basic data types used to represent different kinds of values:

* **String** – used for storing text.
* **Number** – used for numeric values.
* **Boolean** – represents true or false values.
* **Undefined** – means a variable has been declared but not given a value.
* **Null** – represents an empty or non-existent value.
* **Object** – used for storing multiple values in a structured way.
* **Array** – a special type of object used to store a list of values.

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

**Answer:**

* **Undefined** means a variable has been declared but no value is assigned to it yet.
* **Null** means the variable has been assigned an **empty or “nothing” value** on purpose.
* In simple words, **undefined** is something not yet given a value, while **null** is an intentional empty value.

**JavaScript Operators**

**Question 1: What are the different types of operators in JavaScript?**

**Answer:  
Operators in JavaScript are symbols used to perform operations on values or variables.  
There are several types of operators:**

**1. Arithmetic Operators**

* **Used to perform mathematical operations like addition, subtraction, etc.**
* **Examples include:  
  + (add), - (subtract), \* (multiply), / (divide), % (modulus)**

**2. Assignment Operators**

* **Used to assign values to variables.**
* **Examples include:  
  = (assign), += (add and assign), -= (subtract and assign), \*= (multiply and assign)**

**3. Comparison Operators**

* **Used to compare two values and return either true or false.**
* **Examples include:  
  == (equal to), != (not equal to), > (greater than), < (less than), >= (greater than or equal to), <= (less than or equal to)**

**4. Logical Operators**

* **Used to combine or test multiple conditions.**
* **Examples include:  
  && (AND), || (OR), ! (NOT)**

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

**Answer:**

* **== is the loose equality operator. It checks only the value, not the data type.**
* **=== is the strict equality operator. It checks both value and data type.**
* **So, == allows type conversion, while === does not.**

**In short:  
== → compares value only  
=== → compares value and type both**

**Control Flow (If-Else, Switch)**

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

**Answer:**

* **Control flow in JavaScript is the order in which statements are executed in a program.**
* **Normally, JavaScript runs code from top to bottom, but control flow statements like if-else help make decisions and change that flow.**

**How if-else works:**

* **The if-else statement checks a condition.**
* **If the condition is true, one block of code runs.**
* **If the condition is false, another block of code runs.**
* **It helps in decision-making within a program.**

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

**Answer:**

* **A switch statement is used to check one value against many possible cases.**
* **Each case has some code to run if it matches the value.**
* **The break statement stops the switch from checking more cases.**
* **If no case matches, the default case runs.**

**When to use switch:**

* **Use a switch statement when you have many conditions based on the same value.**
* **It makes the code cleaner and easier to read than using many if-else statements.**

**Loops (For, While, Do-While)**

**Question 1: Explain the different types of loops in JavaScript (for, while, do-while).**

**Answer:  
Loops in JavaScript are used to repeat a block of code multiple times until a condition becomes false.  
There are three main types of loops:**

**1. for loop**

* **Used when you know how many times you want to run the code.**
* **It has three parts — initialization, condition, and increment/decrement.**

**2. while loop**

* **Used when you don’t know how many times the loop will run.**
* **The code runs only if the condition is true.**
* **The condition is checked before running the loop body.**

**3. do-while loop**

* **Similar to the while loop, but the code runs at least once, even if the condition is false.**
* **The condition is checked after the loop body.**

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

**Answer:**

* **In a while loop, the condition is checked before the loop runs — if the condition is false, the loop does not run.**
* **In a do-while loop, the code runs once first, and then the condition is checked — even if the condition is false, it executes at least once.**

**In short:**

* **while loop: test first, then run.**
* **do-while loop: run first, then test.**

**Functions**

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

**Answer:**

* **A function in JavaScript is a block of code that performs a specific task and can be reused whenever needed.**
* **It helps to avoid repeating code and makes programs more organized.**

**Syntax explanation (theory only):**

* **A function is declared using the function keyword followed by a name, parameters (if any), and a block of code inside curly braces { }.**
* **To use or run a function, it must be called by its name followed by parentheses ().**

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

**Answer:**

* **A function declaration defines a function using the function keyword and a name. It can be used before it is defined because of hoisting.**
* **A function expression means assigning a function to a variable. It cannot be used before it is created because it is not hoisted in the same way.**

**In short:**

* **Function declaration: hoisted and can be called before definition.**
* **Function expression: not hoisted and must be defined before calling.**

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

**Answer:**

* **Parameters are like placeholders for values that are passed into a function when it is called. They allow the function to work with different data.**
* **A return value is the output a function gives back after finishing its task.**
* **The return statement sends the result back to where the function was called.**

**In short:**

* **Parameters → Input to the function.**
* **Return value → Output from the function.**

**Arrays**

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

**Answer:**

* **An array in JavaScript is a special variable used to store multiple values in a single name.**
* **It can hold values of different data types like numbers, strings, or objects.**
* **Each value in an array has a position number called an index, which starts from 0.**
* **Arrays help in organizing and managing data easily.**
* **They can be declared and initialized by listing the values inside square brackets [ ].**

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

**Answer:  
These methods are used to add or remove elements from an array:**

**push()**

* **Adds a new element at the end of the array.**
* **Increases the array’s length.**

**pop()**

* **Removes the last element from the array.**
* **Decreases the array’s length.**

**shift()**

* **Removes the first element from the array.**
* **Moves the remaining elements one position forward.**

**unshift()**

* **Adds a new element at the beginning of the array.**
* **Shifts existing elements to the next index.**

**Objects**

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

**Answer:**

* **An object in JavaScript is a collection of key–value pairs, where each key (called a property) has a specific value.**
* **It is used to store and organize data in a more detailed and structured way.**
* **Objects can contain values like numbers, strings, arrays, or even other objects.**

**Difference between Objects and Arrays:**

* **Arrays store data in an ordered list using index numbers (0, 1, 2, ...).**
* **Objects store data in key–value pairs, where each value is accessed by its property name.**
* **Arrays are best for lists of items, while objects are best for describing a single item with multiple details.**

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

**Answer:**

* **JavaScript provides two simple ways to access and update the values of object properties:**

**Dot Notation ( . )**

* **You use a dot (.) followed by the property name to get or change its value.**
* **Example (theoretical): objectName.propertyName**

**Bracket Notation ( [ ] )**

* **You use square brackets [ ] and write the property name inside quotes.**
* **Example (theoretical): objectName["propertyName"]**

**In short:**

* **Dot notation → Simple and easy to use when property names are known.**
* **Bracket notation → Useful when property names are dynamic or contain spaces.**

**JavaScript Events**

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

**Answer:**

* **JavaScript events are actions or occurrences that happen in the web browser, such as:**
  + **Clicking a button**
  + **Moving the mouse**
  + **Typing in a text box**
  + **Loading a web page**
* **Events allow web pages to respond to user actions and make them interactive.**
* **An event listener is a function that waits and reacts when a specific event occurs.**
* **It helps in handling user interactions without changing the HTML code directly.**

**In short:**

* **Event → What happens (like a click or keypress).**
* **Event Listener → The code that responds when it happens.**

**Question 2: How does the addEventListener() method work in JavaScript?**

**Answer:**

* **The addEventListener() method is used to attach an event listener to an HTML element.**
* **It listens for a specific event type (like click, mouseover, or keydown) and runs a function when that event occurs.**
* **This method helps keep JavaScript separate from HTML, making the code cleaner and easier to manage.**

**In short:**

* **addEventListener() connects an event with a function that should run when the event happens.**

**Basic idea (theory only):**

**element.addEventListener("eventType", functionName);**

**Example meaning (without code):**

**When a button is clicked, the function linked to that event will run automatically.**

**DOM Manipulation**

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

**Answer:**

* **The DOM (Document Object Model) is a programming interface that represents the structure of a web page as a tree of objects.**
* **Every part of a web page — such as headings, paragraphs, buttons, and images — becomes an object in the DOM.**
* **JavaScript can access, change, add, or remove these elements using the DOM.**
* **This allows web pages to become dynamic and interactive.**

**In short:**

* **The DOM acts as a bridge between HTML and JavaScript, allowing JavaScript to manipulate web content in real time.**

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

**Answer:  
These methods are used by JavaScript to select and work with HTML elements in the DOM:**

**getElementById()**

* **Used to select a single element by its unique ID name.**
* **It returns only one element because IDs are unique in HTML.**

**getElementsByClassName()**

* **Used to select all elements that share the same class name.**
* **It returns a list of elements (not just one), so you can work with each item separately.**

**querySelector()**

* **Used to select the first element that matches a CSS-style selector (like ID, class, or tag). It is very flexible and modern for selecting any kind of element.**