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

**🔹Functions in JavaScript**

A **function** in JavaScript is a **block of reusable code** designed to perform a specific task.  
Functions help you **organize code**, avoid repetition, and improve readability.

**✅ Why Use Functions?**

* Write code **once**, use it **many times**.
* Break large programs into **manageable parts**.
* Perform **specific tasks** with **input (parameters)** and optionally **return output**.

**✅ Syntax for Declaring a Function:**

*function functionName(parameters) {*

*// code to be executed*

*}*

* function: Keyword used to declare a function.
* functionName: Name you give the function.
* parameters: Optional. Input values (can be one or more, separated by commas).
* {}: Function body containing the code.

**✅ Syntax for Calling a Function:**

*functionName(arguments);*

* functionName: Name of the function you're calling.
* arguments: Actual values passed to the function.

**✅ Example 1: Simple Function Without Parameters**

*function greet() {*

*console.log("Hello, welcome!");*

*}*

*greet(); // Calling the function*

**🔍 Output:**

Hello, welcome!

**✅ Example 2: Function With Parameters**

*function add(a, b) {*

*let sum = a + b;*

*console.log("Sum is:", sum);*

*}*

*add(5, 3); // Passing arguments*

**🔍 Output:**

Sum is: 8

**✅ Example 3: Function With Return Value**

*function multiply(x, y) {*

*return x \* y;*

*}*

*let result = multiply(4, 5);*

*console.log("Product is:", result);*

**🔍 Output:**

Product is: 20

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

**🔹 Function Declaration vs Function Expression**

| **Feature** | **Function Declaration** | **Function Expression** |
| --- | --- | --- |
| **Syntax** | function name() { ... } | const name = function() { ... }; |
| **Hoisting** | ✅ Yes – can be called **before** it's defined | ❌ No – can only be called **after** it's defined |
| **Function Name** | Required | Can be **named or anonymous** |
| **Called Before Defined** | ✅ Yes | ❌ No |
| **Stored in Variable?** | ❌ No | ✅ Yes |
| **Use Case** | General-purpose named functions | Callbacks, closures, or when assigning to variables |

**✅ Quick Example:**

*// Function Declaration (hoisted)*

*greet(); // ✅ Works*

*function greet() {*

*console.log("Hello!");*

*}*

*// Function Expression (not hoisted)*

*sayHi(); // ❌ Error*

*const sayHi = function() {*

*console.log("Hi!");*

*};*

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

**Parameters and Return Values in Functions**

* Functions in JavaScript often need to work with **inputs** and **outputs**. This is where **parameters** and **return values** come into play.

**✅ 1. Parameters**

* **Parameters** are **placeholders for input values**.
* They are defined **inside the parentheses** when declaring a function.
* You can define multiple parameters, separated by commas.

**🔸 Example:**

*function greet(name) {*

*console.log("Hello, " + name + "!");*

*}*

*greet("Jayesh"); // "Jayesh" is the \*\*argument\*\* passed to the parameter `name`*

**✅ 2. Return Values**

* A **return value** is the **output** that a function gives back after performing its task.
* The return statement is used to **send a value back** to the caller.
* Once return is executed, the function **stops running**.

**🔸 Example:**

*function add(a, b) {*

*return a + b;*

*}*

*let result = add(5, 3); // result stores the returned value (8)*

*console.log(result);*