JavaScript Introduction

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

Ans.

JavaScript is a high-level, interpreted programming language that is primarily used to make web pages interactive and dynamic. It runs directly in the web browser and works alongside HTML and CSS. JavaScript is often called the "brain of a webpage" because it allows you to add logic, behaviour, and interactivity to otherwise static pages.

Role of JavaScript in Web Development: -

1. **Adds Interactivity:**
   JavaScript allows users to interact with web elements such as buttons, forms, menus, or animations without reloading the page. *Example:* Opening a dropdown menu, image sliders, or validating a form.
2. **Controls Web Page Behaviour:**
   It can change the content or style of a webpage in real time. *Example:* Changing text or colour when a user clicks a button.
3. **Handles Events:**
   JavaScript can respond to user actions like clicks, mouse movements, or keyboard input.
   Example: Displaying a message when a user submits a form.

4. **Enables Asynchronous Operations:**
   It allows web pages to load data in the background without refreshing. *Example:* Loading new posts on social media feeds without reloading the page.
5. **Works with APIs and Servers:**
   JavaScript can fetch and send data from servers using APIs, enabling rich web apps. *Example:* Weather apps, maps, and chat systems.
6. **Used in Full-Stack Development:**
   With technologies like Node.js, JavaScript can also run on the server side, making it possible to build entire web applications using just one language.

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

Ans.

JavaScript, Python, and Java are all powerful programming languages — but they differ in purpose, environment, and syntax.

Purpose & Use :-

	JavaScript	Python	Java
Primary Use	Web development (front-end and back-end).	Data science, AI, web backend, automation.	Enterprise applications, Android apps, back-end.

Main Role	Adds interactivity to websites.	Data analysis, AI, scripting, and back-end.	Building largescale, structured software.
Environment	Runs in browser and server via Node.js .	Runs on computer or server not browser.	Runs on JVM Java Virtual Machine.

Syntax & Ease of Learning :-

	JavaScript	Python	Java
Syntax Style	C-like but more flexible	Very clean and readable	Verbose and strict
Ease of Learning	Moderate (needs HTML/CSS knowledge)	Easiest (beginner friendly)	Harder (strict typing and structure)

Execution Environment :-

1. **JavaScript:** Runs directly in browsers (like Chrome, Firefox) or on servers using Node.js.
2. **Python:** Runs through the Python interpreter (not in browsers).
3. **Java:** Compiled into bytecode and runs on the Java Virtual Machine (JVM).

Typing System :-

Language	Typing	Example
JavaScript	Dynamic	Let x=10
Java	Dynamic	X=11
Python	Static	Int x=2003

Platform Independence :-

1. JavaScript **:** Works in all browsers without installation.
2. Python i**:** Needs Python interpreter installed.
3. Java **:** Requires JVM to run.

Performance :-

1. Fastest**:** Java (compiled).
2. Medium**:** JavaScript (JIT compiled).
3. Slower **:** Python (interpreted).

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

Ans.

The <script> tag in HTML is used to add JavaScript code to a webpage. It allows you to either write JavaScript directly inside the HTML file or link an external JavaScript file.

1. **<u>Internal JavaScript Using <script> Tag</u> :-**

We can write JavaScript code directly inside the <script> tag in your HTML document. It is usually placed inside the <head> or at the end of the <body> section.

For example :-

```
<!DOCTYPE html>
<html>
<head>
 <title>Internal Script Example</title>
 <script>
  function greet() {
   alert("Hello, welcome to my website!");
  }
 </script>
</head>
<body>
 <button onclick="greet()">Click Me</button>
</body>
</html>
```

2. **External JavaScript (Linking JS File) :-**

Instead of writing JavaScript directly inside HTML, we can link an external JavaScript file using the <script> tag src attribute.

For example :-

```
<!DOCTYPE html>
<html>
<head>
```

```
  <title>External Script Example</title>
  <!-- Linking external JavaScript file -->
  <script src="script.js"></script>
</head>
<body>
  <button onclick="greet()">Click Me</button>
</body>
</html>
```

Js code (we have to create new file with .js extension)

```
function greet() {
  alert("Hello from external JavaScript!");
}
```
