JavaScript Introduction

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

JavaScript is a versatile, high-level programming language primarily used to create dynamic and interactive effects on websites

**Client-Side Interactivity:** JavaScript is primarily used to create interactive elements on a webpage, running directly in the user's browser. It allows developers to add features

**Event Handling:** JavaScript is crucial for responding to user actions like clicks, mouse movements, keyboard presses, and more. This allows developers to create dynamic user experiences based on user interaction

**Asynchronous Programming (AJAX and Fetch API):** JavaScript can send and receive data asynchronously (without refreshing the page) using tools like **AJAX** or the more modern **Fetch API**. This enables features

**Manipulating the DOM (Document Object Model):** The DOM represents the structure of an HTML document.

**Backend Development (Node.js):** JavaScript isn't limited to the client side; with **Node.js**, JavaScript can also be used for server-side programming.

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

* **JavaScript**: Primarily used for web development (both frontend and backend with Node.js). It's great for interactive, dynamic web pages and runs in the browser. It's event-driven and supports asynchronous programming.
* **Python**: A general-purpose language known for its simple syntax, making it great for beginners. It's widely used for data science, AI, web development, and automation. It's slower than JavaScript and Java but has powerful libraries.
* **Java**: A general-purpose, object-oriented language often used for large-scale enterprise applications and Android development. It’s platform-independent (via the JVM) and offers strong performance and multi-threading capabilities.

Each language excels in different areas: JavaScript for web, Python for data and automation, and Java for enterprise-scale applications.

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

The <script> tag in HTML is used to include JavaScript code in a webpage. You can either write JavaScript directly inside the tag or link to an external JavaScript file using the src attribute.

inline JavaScript:

<script>

console.log("Hello, World!");

</script>

linking an external JavaScript file:

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

It's best to place the <script> tag just before the closing </body> tag for better page load performance.

Variables and Data Types

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

**var**:

* + **Scope**: Function or global scope.
  + **Reassignable**: Yes.
  + **Hoisting**: Hoisted but initialized as undefined.

**let**:

* + **Scope**: Block scope (e.g., within loops or conditionals).
  + **Reassignable**: Yes.
  + **Hoisting**: Hoisted but not accessible before declaration.

**const**:

* + **Scope**: Block scope.
  + **Reassignable**: No (value can't be reassigned, but contents of objects/arrays can change).
  + **Hoisting**: Hoisted but not accessible before declaration.

**Best Practice:**

* Use let for variables that may change.
* Use const for constants.
* Avoid using var in modern code.

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

In JavaScript, there are **7 basic data types**:

**1. Primitive Types:**

* **Number**: Represents numbers (integer or floating-point).
* let age = 25;
* **String**: Represents text.
* let name = "John";
* **Boolean**: Represents true or false.
* let isActive = true;
* **Undefined**: Variable declared but not assigned a value.
* let a;
* **Null**: Represents no value or an empty reference.
* let user = null;
* **Symbol**: Unique, immutable identifier.
* const id = Symbol('id');
* **BigInt**: For large integers beyond Number limit.
* let bigNumber = 123456789012345678901234567890n;

**2. Object Types:**

* **Object**: Collection of key-value pairs.
* let person = { name: "Alice", age: 30 };
* **Array**: Ordered list of values.
* let fruits = ["Apple", "Banana"];
* **Function**: Block of code for tasks.
* function greet() { return "Hello"; }

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

Here's a comparison of **undefined** and **null** in a table format:

| **Feature** | **undefined** | **null** |
| --- | --- | --- |
| **Meaning** | Represents a variable that has been declared but not assigned a value. | Represents an intentional absence of any value or object. |
| **Type** | Primitive type undefined. | Primitive type, but typeof null returns "object". |
| **Default Value** | Automatically assigned to variables that are declared but not initialized. | Explicitly set by the programmer to indicate no value. |
| **When it's used** | - Uninitialized variable.- Missing function return.- Missing function parameter. | - Explicitly indicating empty or no value.- Resetting or clearing a value. |
| **Example** | let x; console.log(x); // undefined | let user = null; console.log(user); // null |
| **Type of** | Type of undefined returns "undefined". | typeof null returns "object". |

JavaScript Operators

**What are the different types of operators in JavaScript?**

**Explain with examples.**

**• Arithmetic operators**

**• Assignment operators**

**• Comparison operators**

**• Logical operators**

**What is the difference between == and === in JavaScript?**