BASIC KNOWLEDGE

- Where does Javascript fit in today's web development world?

JavaScript continues to play a central role in today's web development world. It is a versatile programming language that is primarily used for client-side web development, meaning it runs directly in the web browser on the user's device. Here are some key areas where JavaScript fits in:

1. Front-end Development: JavaScript is essential for creating interactive and dynamic web interfaces. It enables developers to add behaviors, animations, and interactivity to web pages, enhancing the user experience. Popular JavaScript frameworks and libraries like React, Angular, and Vue.js are widely used for building complex front-end applications.

2. Back-end Development: While JavaScript is primarily known as a client-side language, it has gained significant popularity in the server-side domain as well. With technologies like Node.js, developers can use JavaScript to build scalable and efficient server-side applications. Node.js allows JavaScript to run on the server, enabling full-stack JavaScript development and facilitating code reuse between the front-end and back-end.

3. Mobile App Development: JavaScript is extensively used for mobile app development, thanks to frameworks like React Native and Ionic. These frameworks allow developers to build native-like mobile applications using JavaScript and deploy them on both iOS and Android platforms. This approach saves development time and resources, as a single codebase can be used to target multiple platforms.

4. Web APIs and Browser Extensions: JavaScript provides access to various Web APIs, allowing developers to create powerful browser-based applications. It can interact with the Document Object Model (DOM) to manipulate web page content dynamically, handle user events, and make asynchronous requests to servers using AJAX. JavaScript is also used to develop browser extensions, enabling users to customize and extend the functionality of their web browsers.

5. Web-based Games and Interactive Content: JavaScript, along with HTML5 and CSS, has revolutionized the creation of web-based games and interactive content. With technologies like the Canvas API and WebGL, developers can build sophisticated games and graphics-intensive applications directly in the browser without the need for plugins or additional software.

Overall, JavaScript is an indispensable part of modern web development. Its versatility, widespread adoption, and extensive ecosystem of frameworks, libraries, and tools make it a powerful language for creating a wide range of web applications and experiences.

- In which industries is it implemented?

Technology, Media and Entertainment, E-Commerce, Finance and Banking, Healthcare, Education, Travel and Hospitality

- What is the main role of JavaScript in the Front-End?

JavaScript is essential for creating interactive and dynamic web interfaces. It enables developers to add behaviors, animations, and interactivity to web pages, enhancing the user experience. Popular JavaScript frameworks and libraries like React, Angular, and Vue.js are widely used for building complex front-end applications.

- Is JavaScript a programming or scripting language?

Programming Language: JavaScript is considered a programming language because it has all the essential features and capabilities required for programming tasks. It has variables, data types, control structures (like loops and conditional statements), functions, and object-oriented programming features like objects and classes. JavaScript can be used to write complex, structured programs that perform calculations, manipulate data, interact with databases, and more.

Scripting Language: JavaScript is also considered a scripting language because it is often used for scripting purposes, especially in web development. In this context, "scripting" typically refers to writing code that automates or controls specific tasks within a larger system. JavaScript can be embedded in HTML documents and executed by web browsers to provide interactivity, manipulate web page content, respond to user events, and perform other client-side scripting tasks.

The lines between programming languages and scripting languages can be blurry, and the terms are often used interchangeably. JavaScript's ability to be both a programming language and a scripting language is one of its strengths, as it can be used for a wide range of tasks across different domains, from writing large-scale applications to creating small, interactive scripts for web pages.

- Which tags do you use to enclose JavaScript code?

Inline JavaScript: <script type=”module”> //JavaScript code goes here </script>
External JavaScript file: <script src=”script.js”></script>
JavaScript code within HTML: <button onclick=”myFunction()”>Click me</button>

- How can you include JavaScript code from another source into your documents? And where do you place it in the HTML?

JavaScript code within HTML: <button onclick=”myFunction()”>Click me</button>

- How can you create a comment in JavaScript? On both a single line and multiple lines.

Single-line comments: Single-line comments are used to add comments that span only a single line. Anything after the double forward slash (//) is considered a comment and is ignored by the JavaScript interpreter.

Multi-line comments: Multi-line comments, also known as block comments, are used to create comments that span multiple lines. They start with /* and end with */. Any text between these delimiters is considered a comment and is not executed.

- What is the JavaScript string concatenation operator?

In JavaScript, the string concatenation operator is the plus symbol (+). It is used to combine or concatenate strings together. When the + operator is used with two strings, it joins them together into a single string.

- What characters are used to define a JavaScript variable name?

In JavaScript, variable names can be defined using a combination of letters (both uppercase and lowercase), digits (0-9), underscores (_), and dollar signs ($). However, there are a few rules and conventions to keep in mind when defining variable names:

1. The first character of a variable name must be a letter, an underscore, or a dollar sign. It cannot be a digit. For example, firstName, _count, and $price are valid variable names.

2. After the first character, you can use letters, digits, underscores, and dollar signs in any combination. For example, age, _totalPrice, item$1, and my_variable are valid variable names.

3. Variable names are case-sensitive, which means myVariable and myvariable are considered as two different variables.

4. JavaScript reserved keywords, such as var, let, const, function, if, for, etc., cannot be used as variable names.

- Which data types provides JavaScript?

1. Number: Represents numeric values, including integers and floating-point numbers. For example: 42, 3.14, -10.

2. String: Represents a sequence of characters enclosed in single quotes ('') or double quotes (""). For example: 'Hello', "World".

3. Boolean: Represents a logical value that can be either true or false.

4. Null: Represents the intentional absence of any object value. It is a primitive value that indicates the absence of an object reference.

5. Undefined: Represents a variable that has been declared but has not been assigned a value. It is also a primitive value.

6. Object: Represents a collection of key-value pairs or properties. Objects can be created using object literals ({}) or constructed using the new keyword. Examples include { name: 'John', age: 25 } or new Object().

7. Array: Represents an ordered list of values enclosed in square brackets ([]). Arrays can contain values of any data type, and the elements are accessed using their index. For example: [1, 2, 3], ['apple', 'banana', 'orange'].

8. Symbol: Represents a unique identifier. Symbols are often used as keys in objects to avoid naming conflicts.

   Additionally, JavaScript provides some special data types or features:

9. Function: Functions are a special type of object in JavaScript that can be invoked or called to perform a specific task.

10. Date: Represents dates and times. JavaScript provides the Date object to work with dates and perform various operations.

- What is the difference between variable and array?

    1. Variables store a single value, whereas arrays store multiple values in an ordered manner.

    2. Variables are accessed directly using the variable name, while arrays are accessed using the index of the value.

    3. Variables are often used for individual values or small data sets, while arrays are commonly used when there is a need to group and manage multiple related values together.

    4. Variables can be reassigned to hold a different value, while arrays can have their elements modified, added, or removed as needed.

    5. In summary, variables are used to store and access individual values, while arrays are used to store and access collections of values in a structured manner.