**History of JavaScript**

Netscape was the first browser to support static websites. At that time, even simple tasks like adding two numbers (e.g., 2 + 2) required a server request. The founder of Netscape, **Marc Andreessen**, decided to develop a programming language for their browser to perform basic tasks locally without sending requests to the server every time. He hired **Brendan Eich**, the creator of JavaScript. Eich developed the first version of JavaScript, initially called **“Mocha”**, in just 10 days. It was tested and implemented in Netscape. Later, it was renamed to **“LiveScript”** and eventually to **“JavaScript”**, mainly because Java was a popular programming language at the time. Despite the name, JavaScript has no direct connection to Java.

At first, JavaScript was only used within browsers. The founder of Stack Overflow famously remarked:

"Any application that can be written in JavaScript will eventually be written in JavaScript."

Later, Microsoft reversed the concept of JavaScript and developed a new language called **JScript**. Netscape filed a complaint with **ECMA** (European Computer Manufacturers Association), requesting standardization of JavaScript. ECMA established rules for the language and introduced **ECMAScript**, adding advanced features. Since then, various versions and updates of JavaScript have been released by ECMA. Initially, Microsoft refused to accept the ECMAScript standards but eventually adopted them.

In **2008**, when Google Chrome entered the market with its **V8 JavaScript engine**, JavaScript's performance was significantly improved. Other browser companies followed, developing their own JavaScript engines, such as Firefox's **SpiderMonkey** and Safari's **Chakra**.

In **2009**, **Ryan Dahl** extracted the V8 engine from the browser and developed **Node.js**, which enabled JavaScript to run not only in browsers but also on servers. This innovation paved the way for modern JavaScript frameworks like **Angular**, **React**, and others.

**Question: Is JavaScript interpreted or compiled?**

Initially, JavaScript was purely an **interpreted language**. However, modern JavaScript engines now use a combination of interpretation and **Just-In-Time (JIT) compilation** for better performance.