**JavaScript Day 1(09-09-24)**

J. Rahul Yadav

**1. History of JavaScript?**

In September **1995**, a Netscape programmer named **Brendan Eich** developed a new scripting language in just **10 days**. It was originally called **Mocha**, but quickly became known as **LiveScript** and, later, **JavaScript**.

The language derived its syntax from Java, its first-class functions from Scheme, and its prototype-based inheritance from Self. Since then, JavaScript has been adopted by all major graphical web browsers.

**2.How JavaScript named as JavaScript ?**

When Brendan Eich, an engineer at Netscape, created this new scripting language in 1995, he initially code-named it “**Mocha**”. As it developed internally within Netscape, the brand “**LiveScript**” emerged. Yet, when it came time to decide on a public-facing name, “**JavaScript**” was chosen.

The 1990s witnessed a surge in **Java’s** popularity. Developed by **Sun Microsystems**, Java promised a “**write once, run anywhere**” capability. Its applet technology allowed for miniature programs to run within browsers. This buzz around Java led to a **marketing** **strategy**by Netscape: naming their new language in a way that would resonate with the vast sea of Java developers.

**3.What is Relation with Java ?**

**JavaScript** is a beginner-friendly programming language that allows developers to design **interactive websites**. On the other hand, **Java** is a complex language used to build **sophisticated applications** like Android apps, chatbots, and financial software.

Both Has its Use Cases, distinct purposes, strengths, and **limitations**. These factors can influence the types of projects you can create and the careers you pursue.

Beyond the **marketing-driven name** similarity and some **syntactic** elements, there is no **direct technical relationship** between **Java** and **JavaScript**.

**4.Is JavaScript only used for frontend ?**

No, JavaScript is **not** only used for the front end:

* Front end

JavaScript is commonly used for front-end web development, where it can be used to create **interactive** web pages and elements, change text, and calculate math.

* Back end

JavaScript can also be used for back-end development, where it can process **requests** from **browsers** and handle other tasks. **Node.js**, a **JavaScript** runtime, provides the backend **functionality** that makes this possible.

* Mobile apps

JavaScript can be used to create mobile apps.

* Third-party frameworks

JavaScript is often used with third-party frameworks like **React**, **Node.js**, **Vue**, and **Angular**, which provide pre-built functions that extend the use of JavaScript.

JavaScript is a popular programming language that's easy to use and integrates well with other programming languages, such as HTML and CSS.

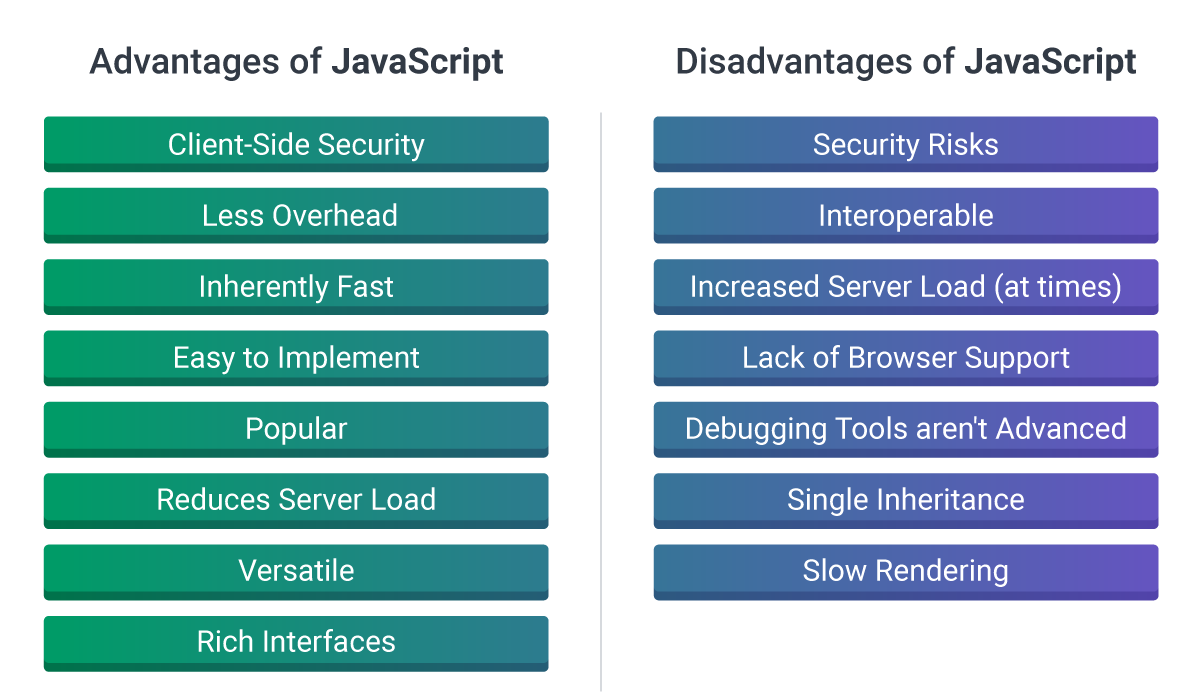
**5.Why and What JavaScript ?**

**WHAT:**

JavaScript is a scripting or programming language that allows you to implement complex features on web pages — every time a web page does more than just sit there and display static information for you to look at — displaying timely content updates, interactive maps, animated 2D/3D graphics, scrolling video jukeboxes, etc.

**WHY:**

* Client-side execution of the logic brings faster user experiences. With the code running **directly** in the browser, the need for **server calls** is **abstracted**, hence a cut in **loading** times. Even with the presence of a server, the fact that JS is asynchronous means that it’s able to communicate with the server in the background without interrupting the user interaction taking place in the frontend.
* Since the very beginning, JavaScript has brought user interface interactivity to the web. It now does the same for applications of all kind, helping to develop the most engaging UX. Today, frameworks like Vue.js are bringing transitions & animations to the next level.
* JavaScript is behind any good responsive web design. More and more, developers need to adapt their design across multiple browsers and devices. Combining HTML5, CSS3 & JavaScript, they can do so within a single codebase.
* For developers, JS is easy to learn and fast to get into active development. Its syntax is easy and flexible for newcomers. It also simplifies the development of complex applications by enabling developers to simplify the app’s composition. The many frameworks & packages out there also ease the life of developers to some extent.

**6.Advantages and Dis-Advantages of JS ?**

**7.Synchronous vs Asynchronous ?**

In synchronous programming, operations are performed one after the other, in sequence. So, basically each line of code waits for the previous one to finish before proceeding to the next. This means that the program executes in a predictable, linear order, with each task being completed before the next one starts.

**Example:**

console.log("Hi");

console.log("Geek");

console.log("How are you?");

**Output:**

Hi

Geek

How are you?  
  
Asynchronous programming, on the other hand, allows multiple tasks to run independently of each other. In asynchronous code, a task can be initiated, and while waiting for it to complete, other tasks can proceed. This non-blocking nature helps improve performance and responsiveness, especially in web applications.

**Example:**

console.log("Hi");

setTimeout(() => {

console.log("Geek");

}, 2000);

console.log("End");

**Output:**

Hi

End

Geek

**8.Diff b/w Scripting and Programming Lang ?**

