🧙‍♂️ The Tale of JavaScript: From Netscape to the World

In the early 1990s, the World Wide Web was still a newborn — a collection of simple, static pages connected by hyperlinks. People could *read* information, but they couldn't interact with it. Clicking a button didn't do anything fancy. Forms couldn't validate themselves. There was no magic in the browser.

Then came a company named **Netscape**. In 1994, they released a groundbreaking web browser called **Netscape Navigator**, and it quickly became the most popular way to access the web.

Netscape had a bold vision:

> "What if websites could do more than just display text and images? What if they could *respond* to users, change in real time, validate forms without reloading, animate, interact?"

To bring this vision to life, Netscape needed something the web didn't yet have — a **scripting language** that ran *inside* the browser.

🧠 Enter Brendan Eich

In 1995, Netscape recruited a talented young programmer named **Brendan Eich**. He had deep experience in systems programming and functional languages, but Netscape didn't hire him to build anything complex. They wanted a **simple, lightweight scripting language** — something web designers could learn easily.

And they needed it fast.

> "You have 10 days," they told him.

Brendan sat down and started working. He created a language that borrowed syntax from **Java and C** (to look familiar), included **first-class functions** like in Lisp, and used **prototypal inheritance** instead of traditional classes.

In just **10 days**, the first version of JavaScript was born.

🧪 From Mocha to LiveScript to JavaScript

Initially, Brendan called his creation **Mocha**.

Then, during testing and internal release, it became **LiveScript**.

But then something curious happened. Netscape struck a marketing deal with **Sun Microsystems**, the creators of the popular new language **Java**. Java was the hot buzzword of the tech world — sleek, powerful, and object-oriented.

So for pure branding purposes, **LiveScript was renamed to JavaScript**.

> This had *nothing* to do with the Java language. But the name stuck.

🕸️ Meanwhile, at Microsoft…

Microsoft, seeing the success of Netscape Navigator and the rise of JavaScript, decided to create their own version. They called it **JScript** and embedded it in **Internet Explorer**.

But this led to a new problem — **incompatibility**. Developers started writing scripts that behaved differently in different browsers.

To solve this, Netscape submitted JavaScript to a standards body — **ECMA International** — so the language could be formally specified and implemented consistently.

In **1997**, JavaScript was standardized as **ECMAScript**. The first version was called **ECMAScript 1**.

⚡ The Rise of JavaScript

At first, JavaScript was considered a toy — good for small tasks like alerts and button clicks. But developers started to see its potential.

Over the next decade, JavaScript grew up:

- It powered **AJAX**, which made apps like Gmail possible.
- Libraries like **jQuery** made cross-browser scripting easier.
- Frameworks like **Angular**, **React**, and **Vue** revolutionized frontend development.
- In **2009**, JavaScript left the browser when **Node.js** arrived — bringing JS to the server.
- And in **2015**, **ES6 (ECMAScript 2015)** gave JavaScript a modern makeover with `let`, `const`, classes, arrow functions, modules, and more.

🌍 JavaScript Today

From humble beginnings in a corner office at Netscape, created under pressure in just 10 days, JavaScript has become the **most widely used programming language in the world**.

It runs in every browser, powers millions of websites, fuels real-time apps, builds mobile apps (React Native), desktop apps (Electron), and even servers and APIs (Node.js).

And all of this began with one question at Netscape:

> “Can we make the web *interactive*?”

🧙‍♂️ The Creator’s Legacy

Brendan Eich went on to become CTO and later CEO of Mozilla (the makers of Firefox), and eventually founded **Brave**, a privacy-focused browser.

But his most enduring legacy is JavaScript — a language created in 10 days, now powering the modern web.