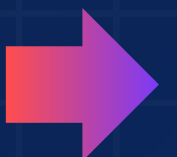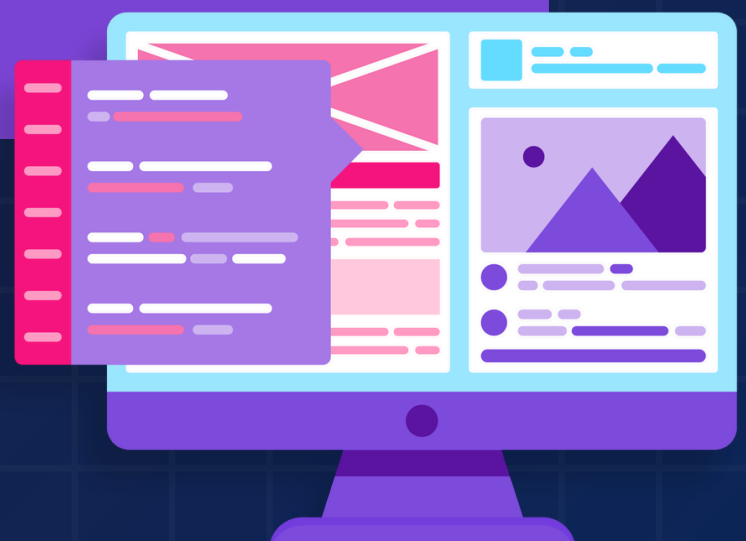EXPLORING

History of Javascript

Jacob Rajan

@jacobrajan

1990

- **Tim Berners** Lee introduced HTML and Web

- 1990's early browsers were **Mosaic, Netscape**

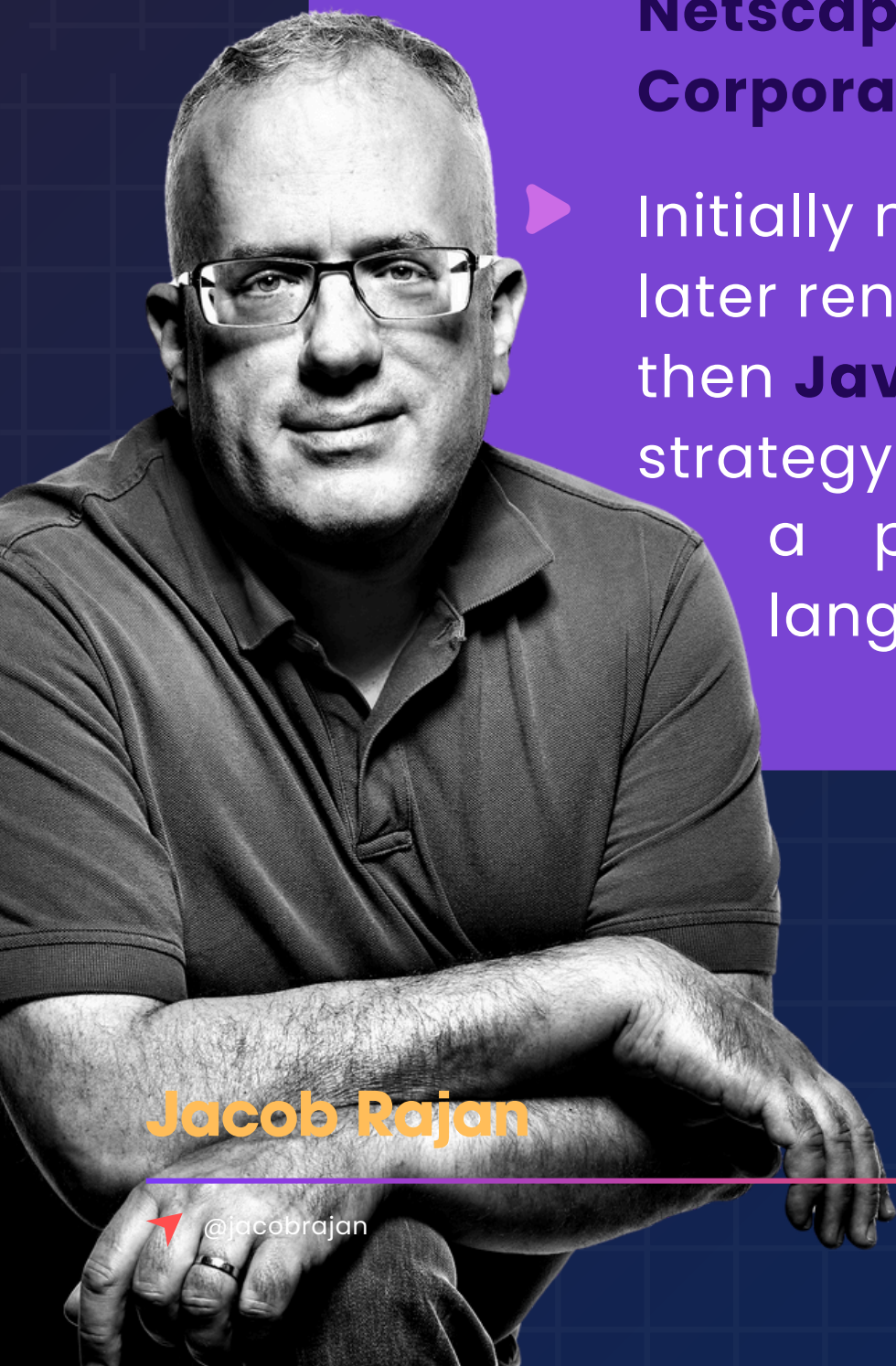- These browsers used HTML as Markup language and ECMAScript as client side script

Jacob Rajan

@jacobrajan

1995

▶ JavaScript was created by **Brendan Eich** while working at **Netscape Communications Corporation.**

▶ Initially named **Mocha**, it was later renamed to **Live Script** and then **JavaScript**, as a marketing strategy to associate it with java a popular programming language at the time.

Jacob Rajan

@jacobrajan

1996

▶ Microsoft introduced a similar language called **JScript** in **Internet Explorer 3.0**, leading to the first browser wars.

▶ Netscape given the rights of maintaining script to a company called **"Sun Micro Systems"**.

Jacob Rajan

@jacobrajan

1997

- JavaScript was submitted to **ECMA International** for standardization, resulting in ECMAScript (ES). ECMAScript 1 (ES1) was the first official version.

- **1998-1999:** ES2 and ES3 were released, with ES3 introducing key features like regular expressions, better string handling, and more robust error handling.

ecma

Jacob Rajan

@jacobrajan

2009

▶ **ECMAScript 5 (ES5)** was released, introducing significant features such as strict mode, JSON support, and new array methods, which greatly improved the language's usability and performance.

▶ **2009: Node.js was introduced by Ryan Dahl**, allowing JavaScript to be used for server-side scripting. This expanded the scope of JavaScript beyond the browser

JS

Jacob Rajan

◥ @jacobrajan

2015

▶ **ECMAScript 6 (ES6), also known as ECMAScript 2015 or ES2015,** brought substantial improvements and new features such as arrow functions, classes, template literals, destructuring, modules, and more. ES6 marked a significant leap forward in making JavaScript a more powerful and developer-friendly language.

JS

Jacob Rajan

@jacobrajan

future

> JavaScript continues to evolve, driven by an active community and ongoing contributions to the ECMAScript standard. Future developments focus on improving performance, enhancing developer productivity, and expanding JavaScript's capabilities both on the client and server sides.

Jacob Rajan

@jacobrajan

What other info would you add?

LEAVE A COMMENT

TO LET US KNOW!

Jacob Rajan

@jacobrajan