JavaScript for ABAP Programmers

Introduction

Chris Whealy / The RIG

JavaScript's Name – And the Resulting Confusion…

JavaScript was first developed by Brendan Eich at Netscape in 1995 (in 10 days). Originally this new scripting language was called "Mocha".

When Netscape Navigator 2.0 was shipped in September 1995, the language was renamed to "LiveScript".

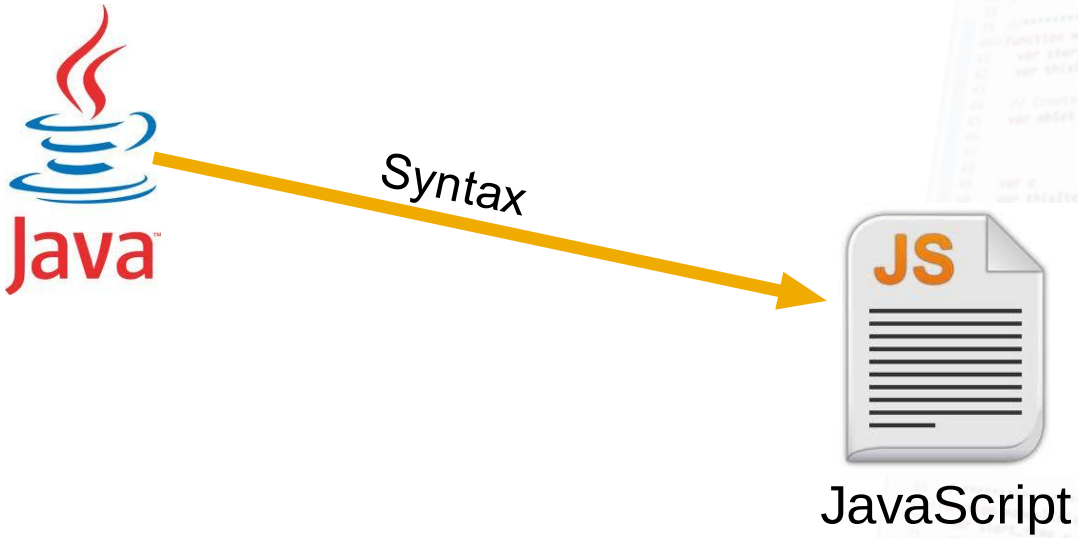
However, when Netscape Navigator 2.0B3 was released on December 4th, 1995, Netscape made a joint announcement with Sun Microsystems to the effect that Netscape's new scripting language would complement Sun Microsystem's new Web application language Java; therefore, the new scripting language would be called "JavaScript" – and this has caused no end of confusion ever since…

The bottom line is simply this…

JavaScript is NOT Java!

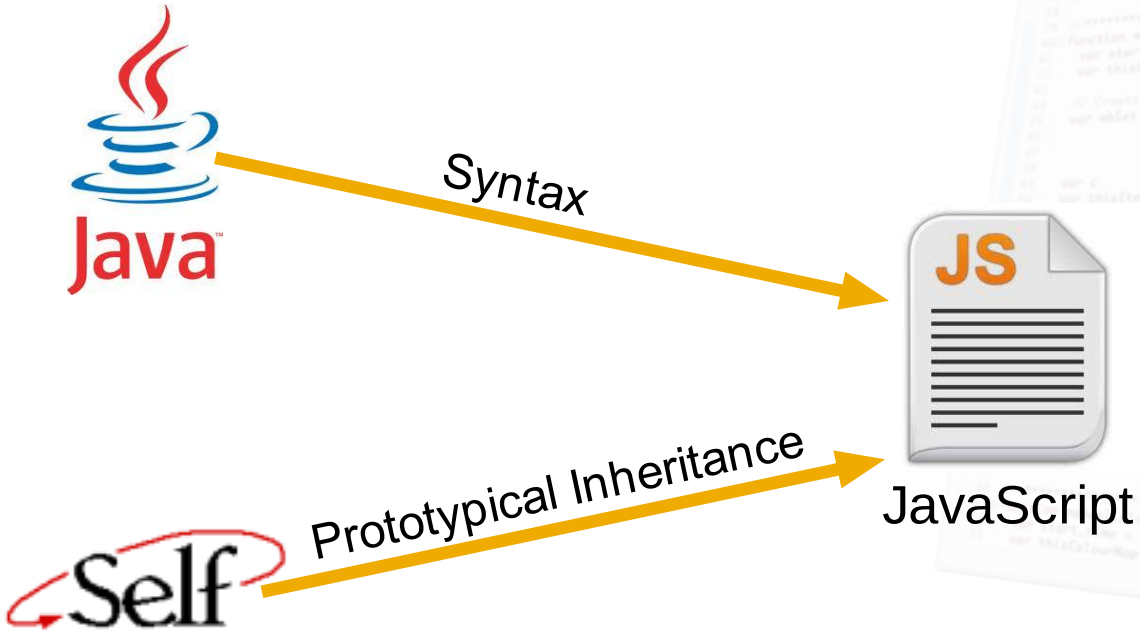
JavaScript's Linguistic Heritage

Over the course of the language's development, various conceptual elements have been borrowed from other languages to form what is now called JavaScript.



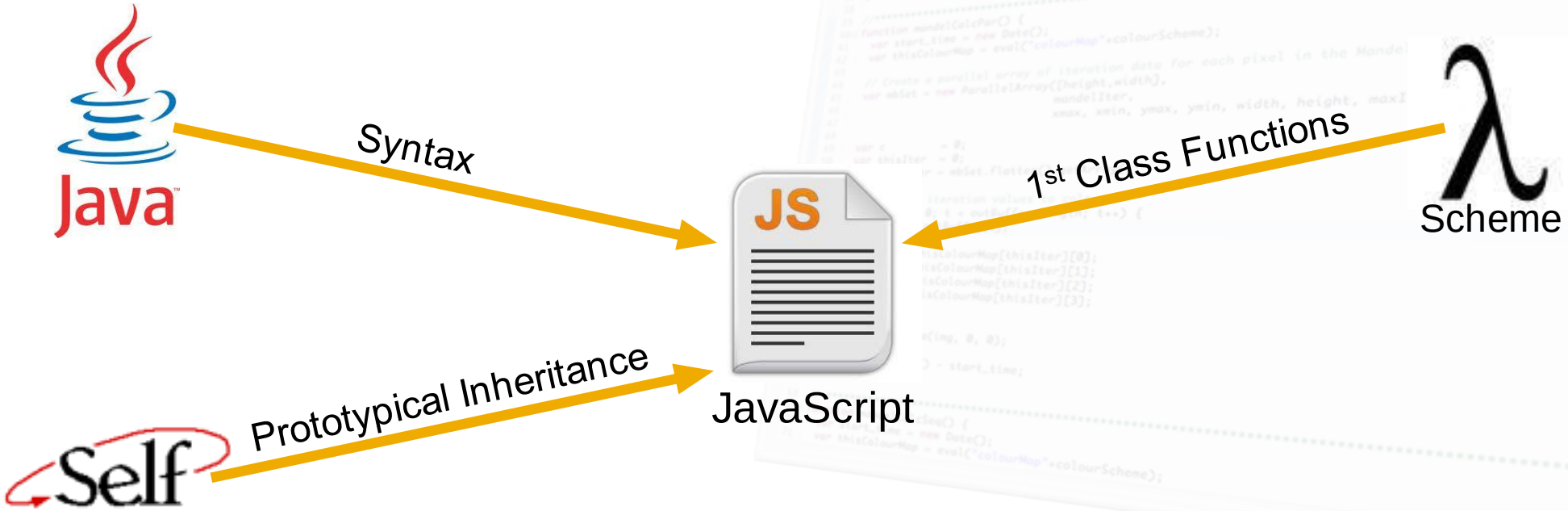
JavaScript's Linguistic Heritage

Over the course of the language's development, various conceptual elements have been borrowed from other languages to form what is now called JavaScript.



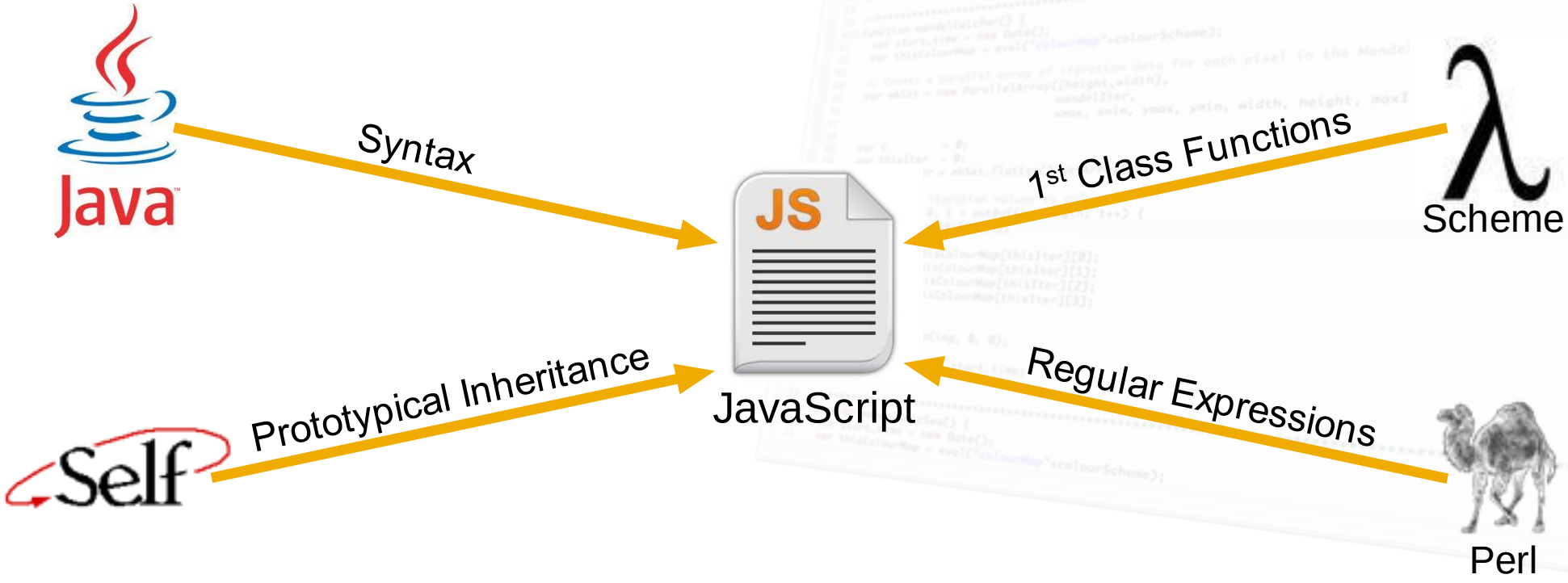
JavaScript's Linguistic Heritage

Over the course of the language's development, various conceptual elements have been borrowed from other languages to form what is now called JavaScript.



JavaScript's Linguistic Heritage

Over the course of the language's development, various conceptual elements have been borrowed from other languages to form what is now called JavaScript.



JavaScript Standards

JavaScript was standardised by the European Computer Manufacturer's Association (ECMA) in 1997. JavaScript used in browsers is now considered to be only one implementation of the ECMA-262 specification.

Edition	Date
1	June 1997
2	June 1998
3	December 1999
4	Abandoned
5	December 2009
5.1	June 2011
6 (Harmony)	Work in progress
7	Work in progress

JavaScript Runtime Engines

In order to execute a JavaScript program, a JavaScript Engine is needed.
Multiple implementations of JavaScript engines are available, with the most widely used being:

JavaScript Engine

Rhino (Mozilla)
SpiderMonkey (Mozilla)

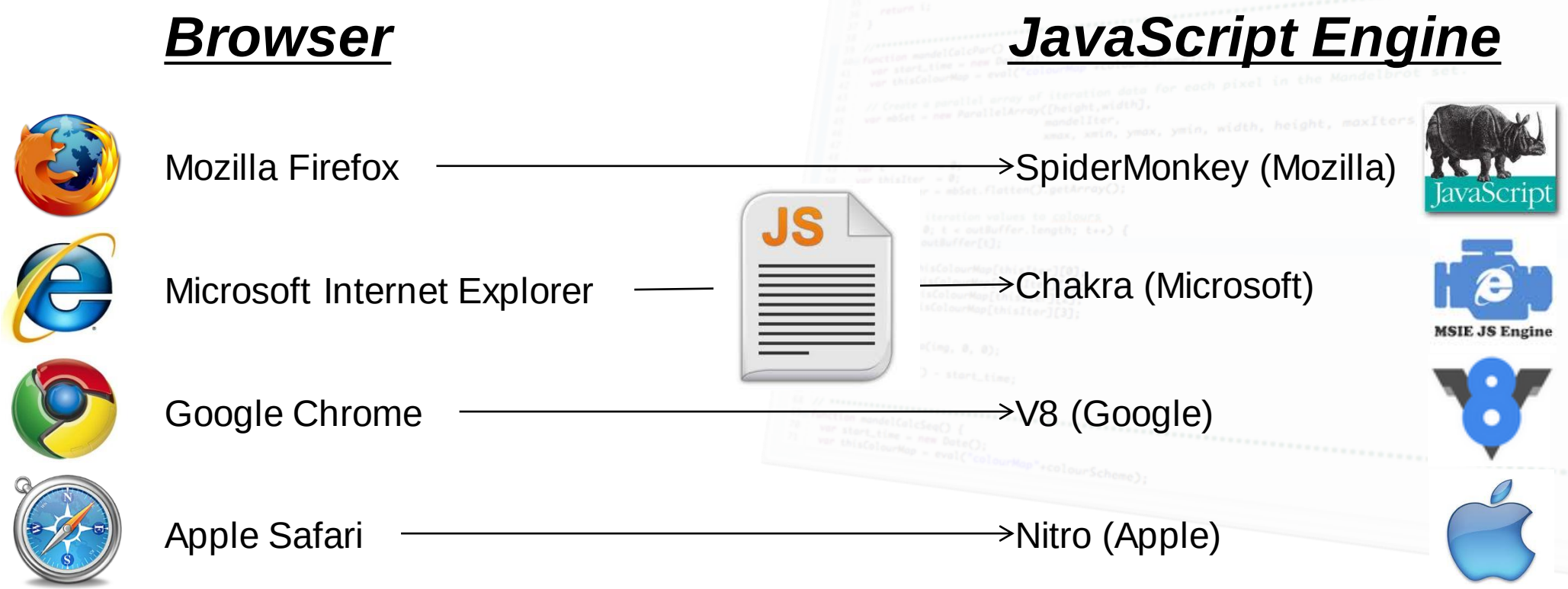
Chakra (Microsoft)

V8 (Google)

Nitro (Apple)

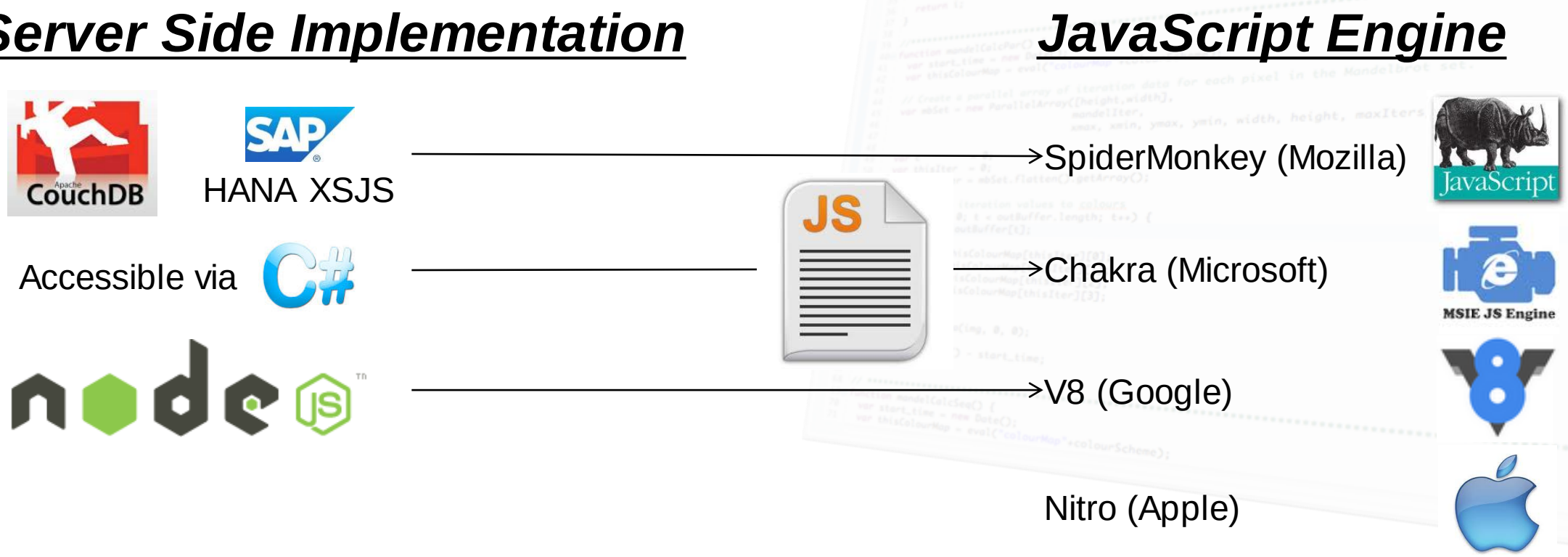
JavaScript Runtime: Client Side Implementations

Modern JavaScript engines have been designed to run either from within a web browser or as stand alone, server-side execution engines.



JavaScript Runtime: Server Side Implementations

There are multiple server-side implementations of JavaScript, many of which are based on Mozilla Rhino (Java based). When speed is required however, Mozilla SpiderMonkey or the Google V8 engine are used instead.



In A Nutshell – The Main Differences Between ABAP & JavaScript

Understanding the differences between ABAP and JavaScript is fundamental to transferring your existing programming skills into JavaScript. Some of the main differences are as follows:

ABAP	JavaScript
Strongly typed	Weakly typed
Syntax similar to COBOL	Syntax derived from Java (and C)
Block scope	Lexical scope
No equivalent concept	Functions are 1st class citizens
OO using class based inheritance	OO using referential inheritance
Imperative programming	Imperative or Functional programming