

DWA_02.8 Knowledge Check_DWA2

1. What do ES5, ES6 and ES2015 mean - and what are the differences between them?

"ES" stands for "ECMAScript," and the number or year following it represents the released version or edition of the ECMAScript standard.. ECMAScript (ES) is a specification or standard that provides a blueprint on how to create a scripting language, and all scripting languages such as JavaScript, JScript, and ActionScript adhere to that standard.

Each ECMAScript standard edition introduces various improvements, additions, and features:

- **ES5 (ECMAScript 5):** Is the 5th edition of ECMAScript standard which was released in 2009. It introduced new 'strict mode', new arrays, and language enhancements.
- **ES6 (ECMAScript 2015):** Is the 6th edition of the ECMAScript standard which was released in 2015. It introduced new features like classes, arrow functions, 'let' and 'const' variable declarations, modules, and async/await for asynchronous programming.
- **ES2015:** Is an alternative name for ECMAScript 2015 (ES6). The ECMAScript committee that oversees the ECMAScript standard/specification decided to switch to yearly updates, and as such the ES6 edition was renamed to ES2015 to reflect the year of release, and any subsequent editions will follow the same naming convention to reflect the year of release.

2. What are JScript, ActionScript and ECMAScript - and how do they relate to JavaScript?

JScript (developed by Microsoft) and ActionScript (developed by Adobe) are scripting languages similar to JavaScript. They are influenced by the ECMAScript standard and share some syntax similarities. However, they are applied in different contexts. For example, JScript is primarily associated with Internet Explorer, while ActionScript is used in conjunction with Adobe Flash Player. Both JScript and ActionScript have become less popular in modern web development.

JavaScript, on the other hand, is the more prominent and popular scripting language, widely used for web development in modern browsers.

ECMAScript is the standard or blueprint for scripting languages such as JScript, ActionScript, and JavaScript. It is not a language itself but rather a guideline or specification for scripting languages.

3. What is an example of a JavaScript specification - and where can you find it?

The ECMAScript 2023 (14th edition) Language Specification, for example.

The ECMAScript Language Specification can be found on Ecma International's official website, specifically in the TC39 (Technical Committee 39) section. However, for the provided language specification, you can find it through the link below:

Link: [ECMAScript 2023 Language Specification](#)

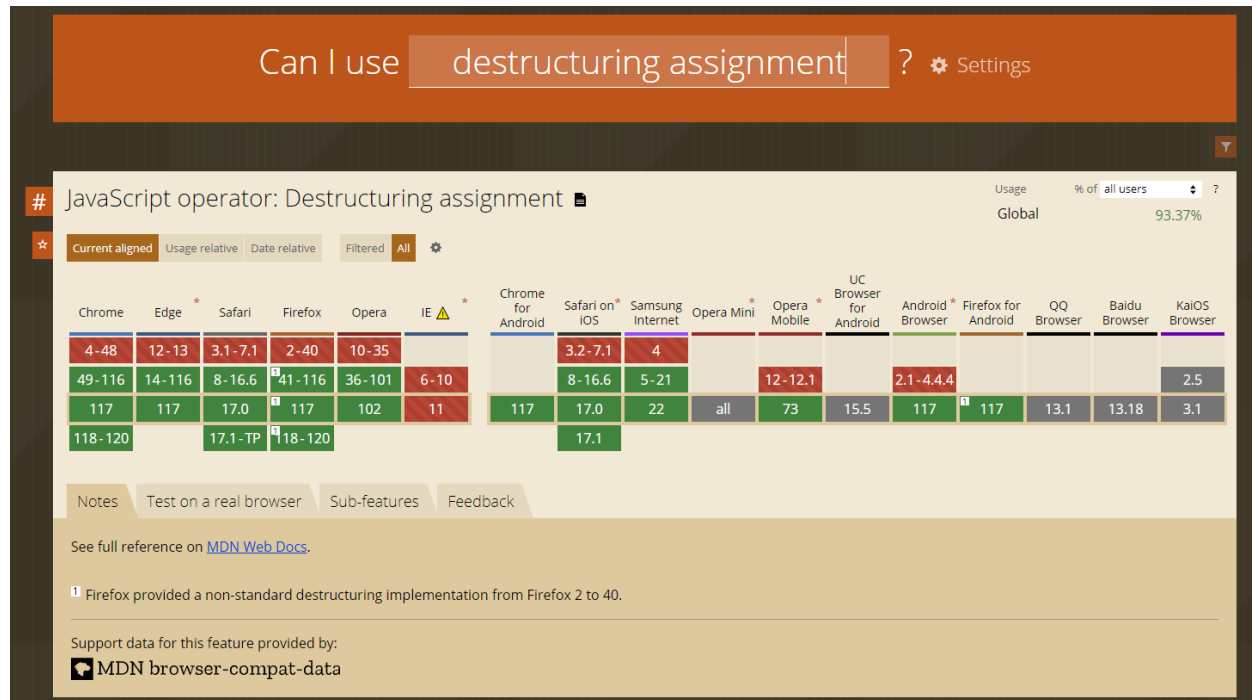
4. What are v8, SpiderMonkey, Chakra and Tamarin? Do they run JavaScript differently?

They are JavaScript Engines or Interpreters that understand, parse, and execute JavaScript code on web browsers.

Yes, they do run JavaScript differently. This is because each JavaScript engine was developed by a different team of developers, and while they all adhere to the ECMAScript standard (which defines the JavaScript language), they may have unique optimization techniques and implementation details that can result in variations in performance and behavior. These differences can impact how JavaScript code is executed and optimized, leading to varying performance levels and compatibility in different browsers.

5. Show a practical example using caniuse.com and the MDN compatibility table.

Below is a snippet from caniuse.com showing the MDN compatibility table for the use of the JavaScript destructuring assignment feature:



Based on the MDN compatibility table, the feature is supported by roughly 93% of web browsers and compatibility issues shouldn't be that much of a concern.