

# DWA\_02.8 Knowledge Check\_DWA2

---

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

> ES5 stands for ECMAScript 5, which is the fifth edition of the ECMAScript language specification. ECMAScript is a scripting language standard that defines the features and syntax that a scripting language like JavaScript should support

> ES6 introduced many new features and improvements to JavaScript, making the language more modern, expressive, and developer-friendly.

Such as = Arrow Functions: A concise syntax for writing anonymous functions, capturing the surrounding `this` value lexically.

>ES2015 means, It brings new features and sugaring for patterns that required significant boilerplate in ES5.

---

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

> ECMAScript (ES): ECMAScript is a scripting language specification standardized by Ecma International. It defines the features and syntax that a scripting language should support. JavaScript is the most popular implementation of the ECMAScript standard. This means that JavaScript adheres to the ECMAScript specification and implements the features and functionalities defined in various ECMAScript versions

> JScript: JScript is Microsoft's implementation of the ECMAScript specification. It was originally developed as a competitor to JavaScript, but over time, it has converged with the ECMAScript standard.

> ActionScript is used for creating interactive animations, games, and web applications. ActionScript was based on an early version of ECMAScript (ECMAScript 4), but it also incorporated some unique features and capabilities specific to Flash development.

---

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

> An example of a JavaScript specification is the ECMAScript Language Specification. This specification defines the ECMAScript language and its features, providing a standardized reference for JavaScript implementations and ensuring consistency across different environments.

---

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

> V8: V8 is an open-source JavaScript engine developed by Google. It is written in C++ and is used in Google Chrome and other Chromium-based browsers. V8 is known for its high-performance, efficient memory management

> SpiderMonkey: SpiderMonkey is an open-source JavaScript engine developed by Mozilla. It was the first JavaScript engine ever created and is used in the Firefox web browser.

> Chakra: Chakra, also known as ChakraCore, was a JavaScript engine developed by Microsoft. It was used in older versions of Microsoft Edge, the web browser that was replaced by the Chromium-based Microsoft Edge. Chakra was known for its performance and featured both a Just-in-Time (JIT) compiler and an interpreter.

> Tamarin: Tamarin is an open-source virtual machine developed by Adobe. It was designed specifically for the ActionScript 3.0 language (a dialect of ECMAScript) and was used in Adobe Flash Player to execute ActionScript code. While Tamarin was not directly used for running JavaScript in web browsers, it shared similarities with JavaScript engines as it interprets and executes scripts written in a dialect of ECMAScript.

> What they do

These JavaScript engines are essential components of their respective web browsers or platforms, and they work to efficiently execute JavaScript code. While they share the common goal of interpreting JavaScript, they may have different performance characteristics, optimizations, and implementation details, resulting in varying levels of execution speed and efficiency.

---

5. Show a practical example using [caniuse.com](https://caniuse.com) and the MDN compatibility table.

---