# DWA_02.8 Knowledge Check_DWA2

---

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

JavaScript is built on a standard called ECMAScript, which has evolved through various versions including ES4, ES5, ES6, and ES2015, each marking a significant step in its development.

- Released in 2009, ES5, or ECMAScript 5, was a pivotal update widely embraced in the coding community. It brought to the table crucial features such as strict mode for cleaner code, JSON for easier data handling, and improved array functions, significantly enhancing JavaScript's power and reliability.

- Come 2015, we saw the arrival of ECMAScript 6, also known as ES2015. This release was a game-changer, introducing a wealth of features like arrow functions for more concise code, classes for object-oriented programming, template literals for easier string handling, and block-scoped variable declarations with let and const. It also introduced promises for handling asynchronous operations and modules for better code organization. ES6/ES2015 took JavaScript to new heights, tailoring it more effectively to meet the complex demands of modern web development.

---

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

JScript, ActionScript, and ECMAScript are each scripting languages with a connection to JavaScript, yet they each serve different functions and come from unique backgrounds.

Their Relationship to JavaScript:

- JavaScript, JScript, and ActionScript all stem from the ECMAScript standard, meaning they have similar syntax and structures.
- Among these, JavaScript stands out as the most prevalent ECMAScript implementation, forming a fundamental part of web technology.
- JScript is Microsoft's version of ECMAScript, crafted specifically to integrate with Internet Explorer.
- ActionScript, on the other hand, was developed by Adobe for creating interactive web content, especially for the Flash platform.

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

As a student looking at the JavaScript specification provided in the image, I'd say an example of a specification would be the rules around how single-line comments are handled in JavaScript. The first edition of the ECMAScript specification (ECMA-262) from 1997 details that single-line comments, which start with two forward slashes (//), should not be executed by the JavaScript engine. They can include any characters except for a line terminator, and they end when a line break is encountered. This means that any text following // on the same line is ignored by the compiler, allowing developers to annotate their code with helpful notes and explanations without affecting the execution of the code.

As a student, if I wanted to follow the latest developments in JavaScript and understand the current specifications, I could check out the TC39 GitHub repository. It's where all the discussions and proposals for new JavaScript features happen, and where I could even suggest ideas or changes to the language. It's pretty cool to have such direct access to the evolution of JavaScript!

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

V8, SpiderMonkey, Chakra, and Tamarin are all JavaScript engines. These are the core components that browsers and other environments use to execute JavaScript code.

All these JavaScript engines are designed to run code based on the ECMAScript standard, ensuring that compliant code operates in a similar manner across them. Despite this, they may vary in terms of performance, how they handle newer ECMAScript features, and the additional tools and optimizations they offer. They also differ in their strategies for compiling and managing memory, which can affect the speed and efficiency of the code execution.

Here's a bit about each:

- V8: This is Google's open-source JavaScript engine, used in Chrome and Node.js. It is known for its high performance and is constantly updated with the latest ECMAScript features.
- SpiderMonkey: This is the first-ever JavaScript engine, created by Brendan Eich for Netscape, and it is currently used by Mozilla Firefox.
- Chakra: Developed by Microsoft, it was the JavaScript engine for the Internet Explorer and later for the Edge browser before Edge moved to Chromium and adopted V8.
- Tamarin: It was an open-source JavaScript engine developed by Adobe Systems, which was intended to be used in Firefox, although this collaboration didn't fully materialize.

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

Can I use    prefers-color-scheme    ?    ⚙ Settings

**prefers-color-scheme media query** 📄 - WD

Media query to detect if the user has set their system to use a light or dark color theme.

Usage    % of  all users    ?
Global    96.53%

Current aligned | Usage relative | Date relative    Filtered  All  ⚙

| Chrome | Edge * | Safari | Firefox | Opera | IE | | Chrome for Android | Safari on iOS * | Samsung Internet | Opera Mini * | Opera Mobile * | UC Browser for Android | Android Browser * | Firefox for Android * | QQ Browser | Baidu Browser | KaiOS Browser |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4-75 | 12-18 | 3.1-12 | 2-66 | 10-60 | | | | 3.2-12.5 | 4-11.2 | | | | | | | | |
| 76-119 | 79-119 | 12.1-17.1 | 67-120 | 62-105 | 6-10 | | | 13-17.1 | 12.0-22 | | 12-12.1 | | 2.1-4.4.4 | | | | 2.5 |
| 120 | 120 | 17.2 | 121 | 106 | 11 | | 120 | 17.2 | 23 | all | 73 | 15.5 | 120 | 119 | 13.1 | 13.18 | 3.1 |
| 121-123 | 121 | 17.3-TP | 122-124 | | | | | 17.3 | | | | | | | | | |

Notes | Test on a real browser | Known issues (0) | Resources (6) | Feedback

Support will also depend on whether or not the OS has support for a light/dark theme preference.