

DWA_02.8 Knowledge Check_DWA2

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

ES5 (ECMAScript 5): It was released in 2009 and introduced enhancements to JavaScript, including strict mode, JSON support, array manipulation method and improved error handling. ES5 is widely supported by modern browsers and forms the foundation of some JavaScript applications.

ES6 (ECMAScript 2015): ES6, also known as ECMAScript 2015, was a major update released in 2015. It brought new features and syntax enhancements to JavaScript, such as arrow functions, classes, modules, template literals and the let and const keywords. ES6 aimed to improve developer productivity and code readability. It introduced significant changes to the language, marking a significant shift in JavaScript's evolution.

ES2015 (ECMAScript 2015): ES2015 is another name for ES6, reflecting the year it was standardized. It is the same as ES6 and is used interchangeably to refer to the ECMAScript version released in 2015.

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

JScript, ActionScript, and ECMAScript languages are related to JavaScript in the following ways:

Script: JScript is a scripting language developed by Microsoft and implemented in Internet Explorer. It is a dialect of ECMAScript, meaning it shares the core syntax and semantics of JavaScript. JScript was Microsoft's proprietary version of ECMAScript and was used primarily for client-side scripting in web development.

ActionScript: ActionScript is a scripting language primarily used for developing Adobe Flash applications. It is also derived from ECMAScript, specifically based on ECMAScript 4. ActionScript included additional features and functionality specific to the Flash platform but shared a fundamental resemblance to JavaScript.

ECMAScript: ECMAScript is the standardized language specification that JavaScript, JScript, and ActionScript are based on. It defines the syntax, features, and behavior of

the language. JavaScript is the most widely known implementation of ECMAScript, and its compatibility with ECMAScript standards is crucial for web 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 itself. The official specification provides a comprehensive and detailed documentation of the ECMAScript language. The current version of the specification is ECMAScript 2022 (ES12), but you can find specifications for previous versions as well.

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

V8, SpiderMonkey, Chakra, and Tamarin are JavaScript engines that power different web browsers and platforms. They are responsible for executing JavaScript code efficiently. While they all interpret and execute JavaScript, there are differences in their architectures, optimizations, and performance characteristics.

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

The screenshot shows the CanIUse website interface. At the top, there's a navigation bar with 'Home', 'News', and a date 'May 1, 2023 - New feature: CSS Relative colors'. Below this is a search bar with the text 'Can I use' followed by 'html' in a large orange box. To the right of the search bar is a 'Settings' link. Below the search bar, it says '50 results found' and shows two checkboxes: 'CanIuse (8)' and 'MDN (42)'. The main content area is titled '# HTML element: html' and shows a table of browser compatibility. The table has columns for various browsers and their versions. The 'Usage' column shows 'Global' with a percentage of '97.46%'. The table is filtered by 'All' and shows compatibility for various browsers including 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, and KaiOS Browser. The table shows version ranges and usage percentages for each browser.

Browser	Version Range	Usage %
Chrome	4-112	113
Edge	12-112	113
Safari	3.1-16.4	16.5
Firefox	2-112	113
Opera	10-11.5	99
IE	6-10	11
Chrome for Android	113	113
Safari on iOS	3.2-16.4	16.5
Samsung Internet	4-20	21
Opera Mini	all	73
Opera Mobile	12	13.4
UC Browser for Android	2.1-4.3	113
Android Browser	4.4-4.4.4	113
Firefox for Android	113	113
QQ Browser	13.1	13.1
Baidu Browser	13.18	13.18
KaiOS Browser	2.5	3.1