DWA_02.8 Knowledge Check_DWA2

---

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

*ES5, ES6, and ES2015 refer to different versions of the ECMAScript language specification, which is the standard for JavaScript. Here are the key differences between them:

- ES5 (ECMAScript 5): Released in 2009, ES5 introduced significant improvements to JavaScript. It added new features like strict mode, JSON support, array methods (e.g., forEach, map, filter), and better error handling with try-catch statements. ES5 is widely supported by all modern browsers.
- ES6 (ECMAScript 2015): Released in 2015, ES6 brought substantial enhancements to JavaScript. It introduced new syntax, features, and APIs, including arrow functions, classes, modules, destructuring assignments, template literals, and promises. ES6 improved developer productivity and made JavaScript code more concise and expressive. However, full ES6 support may require transpiling or using polyfills in older browsers.
- ES2015: ES2015 is another name for ES6. The year-based naming convention (ES2015) was introduced to avoid confusion caused by the incremental version numbers (ES5, ES6). ES2015 is the official name used in the ECMAScript specification.

---

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

*JScript: JScript is a dialect of JavaScript developed by Microsoft. It is primarily used for client-side scripting within Internet Explorer. JScript is largely compatible with ECMAScript, with some differences and additional features specific to the Microsoft ecosystem.

*ActionScript: ActionScript is a scripting language used primarily for creating interactive content and applications within Adobe Flash. ActionScript is influenced by ECMAScript and shares many similarities with JavaScript, although it has its own syntax and features tailored for multimedia and animation purposes.

*ECMAScript: ECMAScript is the standardized scripting language specification that defines the syntax, semantics, and behavior of JavaScript. JavaScript is the most widely used implementation of ECMAScript and is often used interchangeably with it. Other implementations, such as JScript and ActionScript, are also based on the ECMAScript standard.

---

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, which defines the standard for the JavaScript language. The specification provides detailed information about the syntax, semantics, and behavior of JavaScript, including its core features, data types, control structures, functions, and more

---

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

*V8, SpiderMonkey, Chakra, and Tamarin are different JavaScript engines, each with its own implementation of the JavaScript runtime. These engines interpret and execute JavaScript code. Here's a brief overview:

- V8: V8 is the JavaScript engine developed by Google and used in the Google Chrome browser. It is also used in Node.js. V8 is known for its high performance and efficiency, utilizing just-in-time (JIT) compilation to optimize JavaScript execution.
- SpiderMonkey: SpiderMonkey is the JavaScript engine developed by Mozilla and used in the Mozilla Firefox browser. It was the first JavaScript engine ever created and has evolved over time to support various web technologies.
- Chakra: Chakra is the JavaScript engine developed by Microsoft. It was used in Internet Explorer and Microsoft Edge browsers. However, with the transition to the Chromium-based Edge browser, Microsoft now uses V8 as the JavaScript engine.
- Tamarin: Tamarin was a JavaScript engine developed by Adobe. It was designed specifically for the ActionScript language and used in the Adobe Flash Player. However, Adobe discontinued active development of Tamarin in 2011.

.

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

*CSS Flexible box layout module, this method of positioning elements in horizontal or vertical stacks. It provides a flexible and intuitive way to create dynamic layouts that adapt to different screen sizes and orientations.
*Comprehensive table displaying the support for various browser versions. This table includes specific information on support for properties prefixed with flex, such as flex-grow, flex-shrink, and flex-basis, as well as the widely used display: flex property.