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 all refer to different versions of the ECMAScript specification, which is the standard upon which JavaScript is based.

1. ES5 is the fifth edition of the ECMAScript specification, published in December 2009.
   - **Key Features:**
     - Introduced "strict mode" to help catch common coding errors and prevent the use of certain error-prone features.
     - Added new methods for arrays (e.g., `forEach`, `map`, `filter`, `reduce`).
     - Improved support for JSON (JavaScript Object Notation).

2. ES6 is the sixth edition of the ECMAScript specification, published in June 2015.
   - **Key Features:**
     - Introduced significant enhancements to the language, including new syntax features and additional functionality.
     - Added let and const for block-scoped variable declarations.
     - Introduced arrow functions for concise function syntax.
     - Provided new ways to create and manipulate objects, such as the class syntax.
     - Enhanced template literals for more flexible string interpolation.
     - Introduced destructuring assignment for extracting values from arrays and objects.
     - Added the spread/rest operator for handling arrays and function arguments more conveniently.
     - Introduced Promises for improved handling of asynchronous operations.

3.
   ES2015 is essentially another name for ES6. The "ES2015" name was introduced to reflect the year of publication, addressing the concern that using version numbers might imply a fixed, once-per-year release schedule.

---

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

1.
ECMAScript is a scripting language specification that serves as the basis for several scripting languages, including JavaScript, JScript, and ActionScript. It defines the core features that a scripting language should provide.
JavaScript is the most well-known implementation of ECMAScript. When people refer to JavaScript, they are often talking about the implementation of ECMAScript in web browsers. However, other scripting languages, such as JScript and ActionScript, also implement the ECMAScript specification.

2. JavaScript is a high-level, interpreted programming language that conforms to the ECMAScript specification. Originally developed for web browsers, JavaScript is widely used for client-side scripting, enabling interactive and dynamic content on websites.
JavaScript follows the ECMAScript specification, implementing the features and rules defined by ECMAScript. The terms "JavaScript" and "ECMAScript" are often used interchangeably, but it's essential to recognize that JavaScript is a specific implementation of the broader ECMAScript standard.

3. JScript is a scripting language developed by Microsoft. It is the company's implementation of ECMAScript and is primarily used for client-side scripting in Internet Explorer browsers.
JScript adheres to the ECMAScript specification, but it may have Microsoft-specific extensions that are not part of the standard. JScript is distinct from JavaScript in terms of its implementation and is specific to the Internet Explorer browser.

4.
ActionScript is a scripting language developed by Adobe Systems (formerly Macromedia) for use with the Adobe Flash platform. It is primarily used for creating interactive content, animations, and applications within the Flash environment.
Like JavaScript and JScript, ActionScript is an implementation of the ECMAScript standard. However, ActionScript may have features specific to the Flash platform. The syntax and core functionality of ActionScript align with the ECMAScript specification.

In summary, ECMAScript is a standard that defines the core features of scripting languages, and JavaScript, JScript, and ActionScript are implementations of this standard. JavaScript is commonly associated with web browsers, JScript with Internet Explorer, and ActionScript with Adobe Flash. While each implementation follows the ECMAScript specification, there may be platform-specific features or extensions.

---

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

One example of a JavaScript specification is the ECMAScript Language Specification. To find the ECMAScript specification, you can visit the official Ecma International website.
- [ECMAScript® Language Specification](https://www.ecma-international.org/ecma-262/)

---

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

V8, SpiderMonkey, Chakra, and Tamarin are all JavaScript engines, which are components responsible for executing JavaScript code in web browsers or other environments. Each of these engines has its characteristics, and they are associated with different web browsers or platforms.

- While all these engines are designed to execute JavaScript code, there can be differences in how they optimize and execute certain JavaScript features. They may implement various optimization techniques such as just-in-time (JIT) compilation, garbage collection strategies, and other performance enhancements.
- Each engine may have its own set of features, capabilities, and compliance with ECMAScript standards. However, efforts are made to align with ECMAScript specifications to ensure consistency across browsers.

- Differences in the development and maintenance of these engines can lead to variations in performance, memory management, and support for newer JavaScript language features.

In summary, V8, SpiderMonkey, Chakra (deprecated), and Tamarin are JavaScript engines associated with specific web browsers or platforms. While they share the common goal of executing JavaScript, they may employ different strategies and optimizations, resulting in variations in performance and behavior.

---

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