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 (ES) specification, the standard underlying JavaScript. Each version introduces new features and improvements.

ES5 (ECMAScript 5):

- Released in December 2009.
- Introduced strict mode, array methods (e.g., `forEach`, `map`, `filter`), JSON support, and various string manipulation methods.
- Added `bind`, `call`, and `apply` methods for function context manipulation, and `Object.create` for object creation with custom prototypes.

ES6 (ECMAScript 2015):

- Released in June 2015.
- Major update with new features and syntax enhancements, making JavaScript more powerful.
- Added `let` and `const` for block-scoped variable declarations, arrow functions, classes, template literals, destructuring assignments, spread and rest operators, and enhanced object literals.
- Introduced Promises for better asynchronous programming and new array methods like `find`, `findIndex`, and `entries`.

ES2015 (ECMAScript 2015):

- Another name for ES6, reflecting the yearly release cycle starting from 2015. Subsequent versions are named by their release year (e.g., ES2016, ES2017).

---

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

JScript:

- Microsoft's implementation of the ECMAScript specification, used primarily in Internet Explorer.
- Similar to JavaScript but with some differences and additional features specific to JScript.
- Less relevant today due to the decline of Internet Explorer.

ActionScript:

- Developed by Adobe Systems for Adobe Flash applications.
- Rooted in ECMAScript (specifically ECMAScript 3), but a separate language with additional features.
- Used for creating interactive content in Flash, now largely obsolete with the decline of Flash.

ECMAScript:

- The standardized scripting language specification defining JavaScript.
- JavaScript is the most widely known and used implementation of ECMAScript.
- ECMAScript serves as a language specification for creating JavaScript engines in various environments like browsers and Node.js.

In summary, JScript and ActionScript are specific implementations of ECMAScript used in Internet Explorer and Flash, respectively. JavaScript is the most popular ECMAScript implementation used in modern 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. It defines the syntax, semantics, and standard library for JavaScript. You can find it online on the Ecma International website. Ecma International, a standards organization, maintains and regularly updates this specification.

---

4. What are V8, SpiderMonkey, Chakra, and Tamarin? Do they run JavaScript differently?

V8:

- Developed by Google, used in Chrome and Node.js.
- Written in C++ with Just-In-Time (JIT) compilation for high performance.
- Uses optimization techniques like inline caching, hidden classes, and adaptive compilation.

SpiderMonkey:

- Developed by Mozilla, used in Firefox.
- First JavaScript engine, created by Brendan Eich.
- Uses JIT compilation and optimizations like JIT warm-up, type inference, and parallel compilation.

Chakra:

- Developed by Microsoft for Internet Explorer and later Microsoft Edge (replaced by V8 in the Chromium-based Edge).
- Used a combination of JIT and pre-compilation.

Tamarin:

- Developed by Adobe Systems for Adobe Flash Player and ActionScript.
- Used tracing JIT compilation for high-performance execution of ActionScript.

These engines implement the ECMAScript specification but may have different performance optimizations and execution speeds.

---

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

caniuse.com:

1. Open your web browser and go to caniuse.com.
2. In the search bar, type "fetch" and select "Fetch API".
3. The page will display detailed browser support information for the Fetch API.
4. You'll see a table indicating support status (fully supported, partially supported, or not supported) across different browsers and versions.

MDN compatibility table:

1. Open your web browser and go to the MDN web docs.
2. In the search bar, type "fetch" and select "Fetch API".
3. The documentation page will provide usage, syntax, and compatibility information.
4. Scroll to the "Browser compatibility" section to see a table displaying support status for various browsers and versions.
