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 are different versions of the ECMAScript (ES) language specification, which defines the standard for JavaScript. Each version introduces new features, syntax, and improvements to the language.

ES5 was released in 2009 and brought significant enhancements to the JavaScript language. Some notable features introduced in ES5 include:

- Strict mode: A stricter mode of JavaScript that enforces better coding practices and eliminates certain error-prone behaviors

ES6, also known as ES2015, was a major update to JavaScript released in 2015. It introduced numerous new features and syntax improvements, making JavaScript more powerful and expressive. Some notable features introduced in ES6 include:

- Block-scoped variables: The `let` and `const` keywords for declaring variables with block scope, improving variable scoping rules.

---

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

JScript is Microsoft's legacy dialect of the ECMAScript standard that is used in Microsoft's Internet Explorer 11 and older.

JavaScript is technically titled ECMAScript . while **ActionScript**, IS A JavaScript knock-off language created by **Macromedia**.it was initially designed for controlling simple two-dimensional vector animations made in Adobe Flash (formerly Macromedia Flash). Initially focused on animation, early versions of Flash content offered few interactivity features, thus had very limited scripting capability. Later versions added functionality allowing for the creation of web-based games and rich web applications with streaming media (such as video and audio). Today, ActionScript is suitable for desktop and mobile

development through Adobe AIR; it is used in some database applications and in basic robotics as in Make Controller Kit.

---

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

The specification is a collection of documents describing how JavaScript and its variants should work in the context of JavaScript and its variants. For example, here is an extract from the very **first specification created by ECMA** explaining how single-line commenting should work:

Because a single-line comment can contain any character except a "LineTerminator" character, and because of the general rule that a token is always as long as possible, a single-line comment always consists of all characters from the "//" marker to the end of the line. However, the "LineTerminator" at the end of the line is not considered to be part of the single-line comment; it is recognized separately by the lexical grammar and becomes part of the stream of input elements for the syntactic grammar. This point is very important because it implies that the presence or absence of single-line comments does not affect the process of automatic semicolon insertion — ECMA 262: 1st Edition (1997)

In short:

> The compiler should not execute single-line comments.
> Singe-line comments should always start with "//"
> Single-line comments accept any character except line breaks.
> Single-line comments end before the first line breaks

---

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

V8, SpiderMonkey, Chakra, and Tamarin are all JavaScript engines, which are responsible for executing JavaScript code in web browsers or other environments. While they all serve the same

purpose of running JavaScript, they are developed by different organizations and may have unique features and optimizations.

V8 is the JavaScript engine developed by Google for the Chrome web browser. It is also used in other projects, such as Node.js. V8 is known for its high-performance execution and innovative optimization techniques, including just-in-time (JIT) compilation. It compiles JavaScript code into machine code, resulting in faster execution times.

SpiderMonkey is the JavaScript engine developed by Mozilla for the Firefox web browser. It is one of the oldest JavaScript engines and has been actively maintained and improved over the years. SpiderMonkey has a long history of supporting JavaScript and has implemented various features and optimizations, including JIT compilation.

Chakra is the JavaScript engine developed by Microsoft. It was initially used in Internet Explorer and later became the foundation for the JavaScript engine in Microsoft Edge. Chakra underwent significant improvements with the release of Microsoft EdgeHTML and introduced features like just-in-time (JIT) compilation and advanced garbage collection techniques.

Tamarin was a JavaScript engine developed by Adobe Systems. It was designed to be used in the Adobe Flash Player and Adobe AIR runtime environments. Tamarin was notable for its use of just-in-time (JIT) compilation and advanced garbage collection algorithms. However, development on Tamarin was discontinued in 2011, and its codebase was eventually merged into the Mozilla SpiderMonkey project.

While these JavaScript engines have different origins and may have specific optimizations or features, they all aim to execute JavaScript code effectively. However, it's worth noting that each engine can have its own performance characteristics and may vary in terms of execution speed, memory usage, and support for specific JavaScript features.V8, SpiderMonkey, Chakra, and Tamarin are all JavaScript engines, which are responsible for executing JavaScript code in web browsers or other environments. While they all serve the same purpose of running JavaScript, they are developed by different organizations and may have unique features and optimizations.

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