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 all references to different versions of the ECMAScript (ES) standard, which is the standardized specification for JavaScript. Here's what each of these terms means and the differences between them:

- ES5: ES5, short for ECMAScript 5, was released in December 2009. It introduced significant enhancements to the JavaScript language and formed the basis for modern JavaScript development. ES5 introduced features like strict mode, JSON support, array methods (e.g., forEach, map, reduce), object property manipulation methods (e.g., Object.keys, Object.defineProperty), and more. ES5 is widely supported by all major web browsers.

- ES6: ES6, short for ECMAScript 2015, is the next major version of the ECMAScript standard. It was released in June 2015. However, ES6 is often used interchangeably with ECMAScript 2015. ES6 introduced numerous new features and syntax enhancements to JavaScript, including let and const declarations, arrow functions, classes, modules, template literals, destructuring assignments, and more. ES6 greatly improved JavaScript's readability, expressiveness, and provided new programming patterns. While ES6 introduced many new features, not all web browsers supported them immediately, leading to the need for transpilers like Babel to convert ES6 code into ES5 code for broader compatibility.

- ES2015: ES2015 is another name for ECMAScript 2015, which is equivalent to ES6. The term "ES2015" is derived from the year the ECMAScript 2015 specification was finalized. The ECMAScript standards have been named by year since then, such as ES2016, ES2017, ES2018, and so on, to indicate yearly updates to the language.

In summary, ES5 is the version of JavaScript introduced in 2009 with significant enhancements. ES6 or ES2015 refers to the next major version released in 2015, which brought significant changes and improvements to the language. The term ES2015 is used to indicate the specific version released in that year, while ES6 is a more general term commonly used to refer to the ECMAScript 2015 standard.

---

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

JScript, ActionScript, and ECMAScript are all programming languages that are related to JavaScript. Here's an overview of each:

- JScript: JScript is a scripting language developed by Microsoft. It was introduced as the company's implementation of the ECMAScript specification, which is a standardized specification for scripting languages. JScript was primarily used in Microsoft's Internet Explorer browser and Windows scripting environments. JScript is very similar to JavaScript and shares many features and syntax with it.

- ActionScript: ActionScript is a scripting language developed by Macromedia (later acquired by Adobe). It was primarily used for creating interactive content and animations in Adobe Flash. ActionScript was based on an earlier version of ECMAScript, specifically ECMAScript 4, but it evolved separately and introduced additional features specific to Flash development. While ActionScript and JavaScript have some similarities, they are not fully compatible.

- ECMAScript: ECMAScript is a standardized scripting language specification. It defines the core features and syntax that scripting languages like JavaScript, JScript, and ActionScript are based on. JavaScript is the most well-known implementation of ECMAScript and is often used as a synonym for ECMAScript. ECMAScript defines the basic features such as variables, data types, control structures, functions, and object-oriented programming constructs that are common to these languages.

In summary, JScript and ActionScript are specific implementations of the ECMAScript specification, each with its own unique features and use cases. JavaScript, on the other hand, is the most widely used implementation of ECMAScript and has become the de facto scripting language 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 specification, which defines the standard for the JavaScript programming language. The ECMAScript specification provides detailed descriptions of the syntax, semantics, and behavior of JavaScript.

You can find the ECMAScript specification at the following locations:

- Official website: The latest version of the ECMAScript specification, along with previous versions, can be found on the official ECMAScript website at https://www.ecma-international.org/standards/ecma-262/.
- Ecma International: The organization responsible for maintaining and publishing the ECMAScript standard is Ecma International. You can visit their website at https://www.ecma-international.org/ for more information and access to the specification.
- GitHub: The ECMAScript specification is also available on GitHub. The official repository can be found at https://github.com/tc39/ecma262, where you can access the source files, proposals, and discussions related to the evolution of the JavaScript language.

---

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

V8, SpiderMonkey, Chakra, and Tamarin are all JavaScript engines that are used to run JavaScript code. These engines are responsible for interpreting and executing JavaScript code in web browsers, server-side environments, or other applications.

- V8: V8 is an open-source JavaScript engine developed by Google. It is primarily used in the Google Chrome web browser and the Node.js runtime environment. V8 compiles JavaScript code into machine code and utilizes various optimization techniques to improve performance. It employs just-in-time (JIT) compilation to dynamically translate frequently executed code into highly efficient machine code.

- SpiderMonkey: SpiderMonkey is the original JavaScript engine created by Brendan Eich for the Netscape Navigator web browser. It is an open-source engine developed by the Mozilla Foundation and is used in the Mozilla Firefox web browser. SpiderMonkey interprets and executes JavaScript code and has undergone significant improvements over the years to enhance performance and support modern JavaScript features.

- Chakra: Chakra, also known as ChakraCore, is a JavaScript engine developed by Microsoft. It was initially used in the Internet Explorer web browser and later evolved into the EdgeHTML rendering engine used in Microsoft Edge. ChakraCore is an open-source engine and can be embedded into other applications. Microsoft has transitioned to using the Chromium rendering engine in its Edge browser, which utilizes V8 as its JavaScript engine.

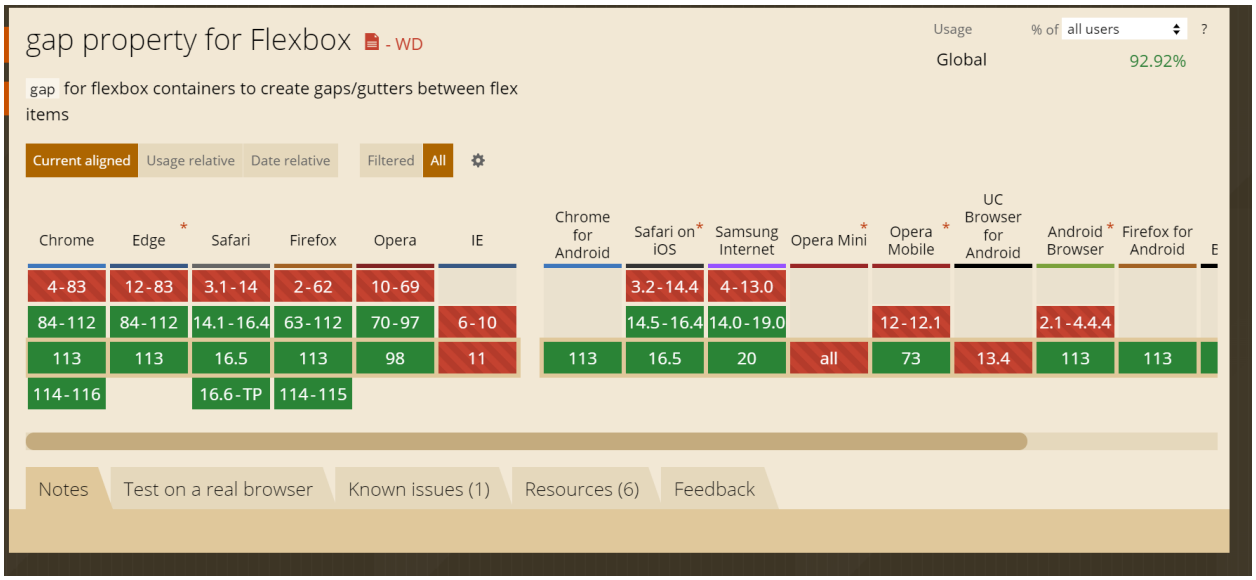
- Tamarin: Tamarin is a high-performance JavaScript engine developed by Adobe Systems. It was specifically designed for the Adobe Flash Player, enabling efficient execution of ActionScript (a scripting language similar to JavaScript) and other dynamic languages. Tamarin incorporates a technique called trace-based just-in-time compilation, which optimizes code execution based on runtime profiling information.

While all these engines run JavaScript, there may be differences in their implementation details, performance characteristics, and support for specific JavaScript features. Each engine has its own set of optimizations and strategies to execute JavaScript code efficiently, leading to variations in performance across different engines. Additionally, different versions of these engines may have distinct features and capabilities, so it's important to consider the specific version and context in which they are used.

---

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

Practical Flexbox example.



MDN compatibility table.

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android
`css`	✓ 28	✓ 12	✓ 22	✓ 15	✓ 9	✓ 28	✓ 22	✓ 15	✓ 9	✓ 1.5	✓ 4.4
`Hz`	✓ 66	✓ 79	✗ No	✓ 53	✓ 16.4	✓ 66	✗ No	✓ 47	✓ 16.4	✓ 9.0	✓ 66
`Q`	✓ 66	✓ 79	✗ No	✓ 53	✓ 16.4	✓ 66	✗ No	✓ 47	✓ 16.4	✓ 9.0	✓ 66
`ch`	✓ 66	✓ 79	✗ No	✓ 53	✓ 16.4	✓ 66	✗ No	✓ 47	✓ 16.4	✓ 9.0	✓ 66