

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 specification (also known by its standard number: ECMA-262), which is the reference standard for scripting languages like JavaScript. ES5 refers to the 5th edition of this specification, released in 2009, and represents the first major revision implemented. ES6 refers to the 6th edition which was released in 2015 and brought further major changes.

ES2015 is actually a synonym for ES6. ECMA International decided to switch to annual releases of ECMAScript, with each edition being named based on the year they were released. ES6 was therefore renamed to ES2015 and there is no difference between them.

ES6 differs from the previous ES5 version in several major ways. ES6 introduced new syntax and language features such as arrow functions, classes, template literals, destructuring assignments, default function parameters, and the spread operator. It also introduced "let" and "const" keywords which replaced the "var" keyword of ES5 and allowed for block-scoping of declared variables. It further introduced a standardized module system using "import" and "export" and other features such as promises, iterators/generators and enhanced object literals. ES5 and ES6 also differ in terms of their levels of browser support - while ES5 is almost universally supported by all major browsers, some features of ES6 are not.

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

ECMAScript is a standardized version of JavaScript that was created to ensure compatibility among different implementations of the language. It is a general-purpose, cross-platform, vendor-neutral specification that lays out the syntax, semantics, and core features that should be used for JavaScript as well as other similar off-shoot languages. Languages that follow the rules and guidelines defined in the ECMAScript specification are said to be implementations of the specification. JavaScript, although it was initially the basis for ECMAScript, is therefore now classified as an implementation

of ECMAScript and is the the most widely used and well-known language that adheres to this standard.

JScript is another implementation of the ECMAScript standard, which originated as a reverse-engineered version of the original JavaScript used in the Microsoft Internet Explorer web browser. Because both JavaScript and JScript are derived from ECMAScript, they share a lot of the same syntax and features. However, because JScript was developed specifically for Microsoft technologies it has some features that are not supported in other environments. JavaScript, on the other hand, is more cross-platform and vendor-independent and is supported by all major web browsers.

Like JScript, ActionScript was developed as a knock-off version of JavaScript created by Macromedia Inc (now Adobe Systems). It is also classified as an implementation of ECMAScript and is thus very similar to JavaScript in terms of syntax and core programming features. It differs from JavaScript in terms of its primary purpose and the main usage environment, as it was developed for applications targeting the Adobe Flash platform and needed to have functionality for handling interactive media content. JavaScript, on the other hand, was developed for web browsers and more general use.

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

The most recently published example of a JavaScript specification is the ECMAScript 2021 (ES2021) specification. ECMAScript specifications are maintained by Ecma International and are freely available on their website, under the Standards & Publications section. All versions of the ECMAScript specification can be accessed using the standard reference number: ECMA-262, and then browsing through the various versions available, including ES2021 (12th edition).

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

V8, SpiderMonkey, Chakra and Tamarin are all JavaScript engines (also called JavaScript interpreters). They are essentially programs that understand and execute JavaScript code. Different JavaScript engines are used on different web browsers. V8 is used in Google Chrome, SpiderMonkey is used in Mozilla Firefox, Chakra is used in

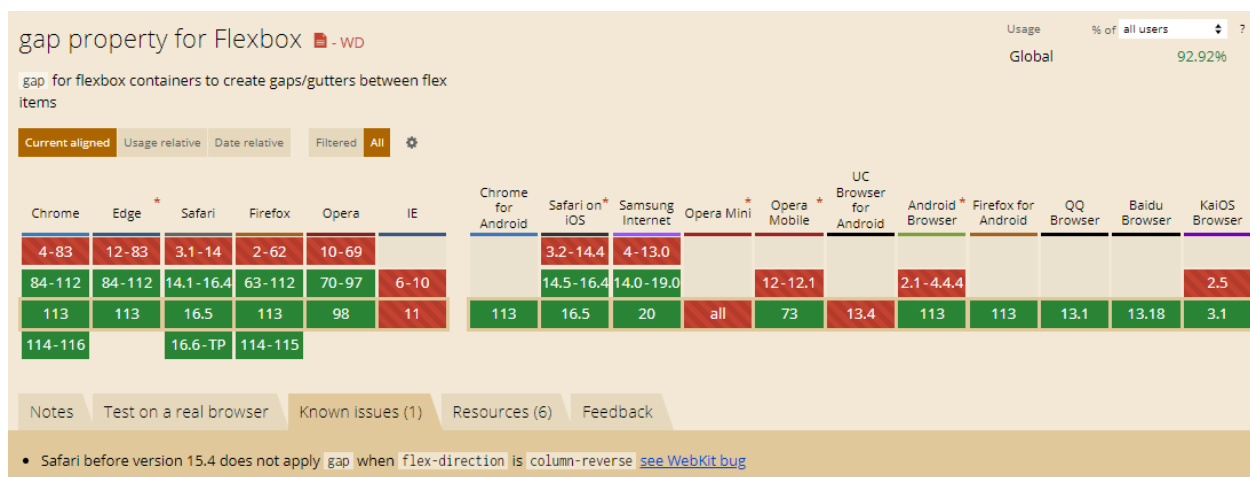
Microsoft Edge and Tamarin was used in Adobe Flash Player and Adobe AIR, but was discontinued in 2012.

Different engines can execute JavaScript code differently. While all do adhere to the ECMAScript specification, they are often designed to only support a certain subset of the language based on their environments. They may also have introduced additional language features that go beyond the specification, and may also use different strategies for parsing, optimizing, and executing the code. Differences in optimization techniques can result in significant changes in performance. There are also differences in browser support for some features, as the different engines use slightly different “dialects” of JavaScript. Finally, different JavaScript engines have different levels of compliance with new ECMAScript standards as changes are introduced incrementally at different paces and to different extents.

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

Example - checking support for gap property when using Flexbox in CSS

Canibase:



MDN Browser Compatibility Table:

	Desktop					Mobile					
	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android
<code>gap</code>	✓ 57	✓ 16	✓ 52	✓ 44	✓ 10.1	✓ 57	✓ 52	✓ 43	✓ 10.3	✓ 7.0	✓ 57
Supported in Flex Layout	✓ 84	✓ 84	✓ 63	✓ 70	✓ 14.1	✓ 84	✓ 63	✓ 60	✓ 14.5	✓ 14.0	✓ 84
Supported in Grid Layout	✓ 66 ...	✓ 16 ...	✓ 61 ...	✓ 53 ...	✓ 12 ...	✓ 66 ...	✓ 61 ...	✓ 47 ...	✓ 12 ...	✓ 9.0 ...	✓ 66 ...
<code>calc()</code> values	✓ 66	✓ 16	✓ 52	✓ 53	✓ 12.1	✓ 66	✓ 52	✓ 47	✓ 12.2	✓ 9.0	✓ 66
<code><percentage></code> values	✓ 66	✓ 16	✓ 52	✓ 53	✓ 11	✓ 66	✓ 52	✓ 47	✓ 11	✓ 9.0	✓ 66
Supported in Multi-column Layout	✓ 66	✓ 16	✓ 61	✓ 53	✓ 14.1	✓ 66	✓ 61	✓ 47	✓ 14.5	✓ 9.0	✓ 66

Tip: you can click/tap on a cell for more information.

✓ Full support 🚧 Uses a non-standard name. ... Has more compatibility info.