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 versions of the ECMAScript specification, which is the standardized specification for JavaScript.

- ES5 (ECMAScript 5) was released in 2009 and introduced significant enhancements to the language, including strict mode, JSON support, and array methods like forEach, map, filter, etc.
- ES6 (ECMAScript 2015) is a major update to the language that was released in 2015. It introduced many new features such as arrow functions, classes, modules, template literals, and destructuring assignments, among others. ES6 is also known as ES2015.
- ES2015 is just another name for ES6, reflecting the year in which it was released. After ES6, the ECMAScript committee adopted a new naming convention that uses the year instead of version numbers to avoid confusion caused by multiple version numbers (like ES5.1, ES6, etc.).

---

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

- JavaScript is the most popular implementation of the ECMAScript Standard. ActionScript and JScript are other languages that implement the ECMAScript . The core features of Javascript are based on the ECMAScript standard, but Javascript also has other additional features that are not in the ECMA specifications/standard.
- JScript is Microsoft's dialect of the ECMAScript specification, which is used in Internet Explorer. It is essentially the same as JavaScript with some additional features specific to Microsoft's implementation.
- ActionScript is a scripting language derived from ECMAScript, primarily used for creating interactive content in Adobe Flash. It has a syntax similar to JavaScript and supports object-oriented programming.
- ECMAScript is the standard specification for scripting languages, with JavaScript being the most popular implementation of it. ECMAScript defines the syntax, types, and behavior of the language, while JavaScript is the actual programming language that conforms to the ECMAScript specification.

---

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. The latest version of the specification can be found on the ECMA International website https://www.ecma-international.org/publications/files/ECMA-ST-ARCH/ECMA-262,%201st%20edition,%20June%201997.pdf

---

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

- V8 is a free and open-source JavaScript and WebAssembly engine developed by the Chromium Project for Chromium and Google Chrome web browsers.[5] The project's creator is Lars Bak.[6] The first version of the V8 engine was released at the same time as the first version of Chrome: 2 September 2008. It has also been used on the server side, for example in Couchbase and Node.js.

- SpiderMonkey is an open-source JavaScript and WebAssembly engine by the Mozilla Foundation.It is the first JavaScript engine, written by Brendan Eich at Netscape Communications, and later released as open source and currently maintained by the Mozilla Foundation. It is used in the Firefox web browser. SpiderMonkey was released in 1995. It implements the ECMA-262 specification (ECMAScript) and is written in C/C++.

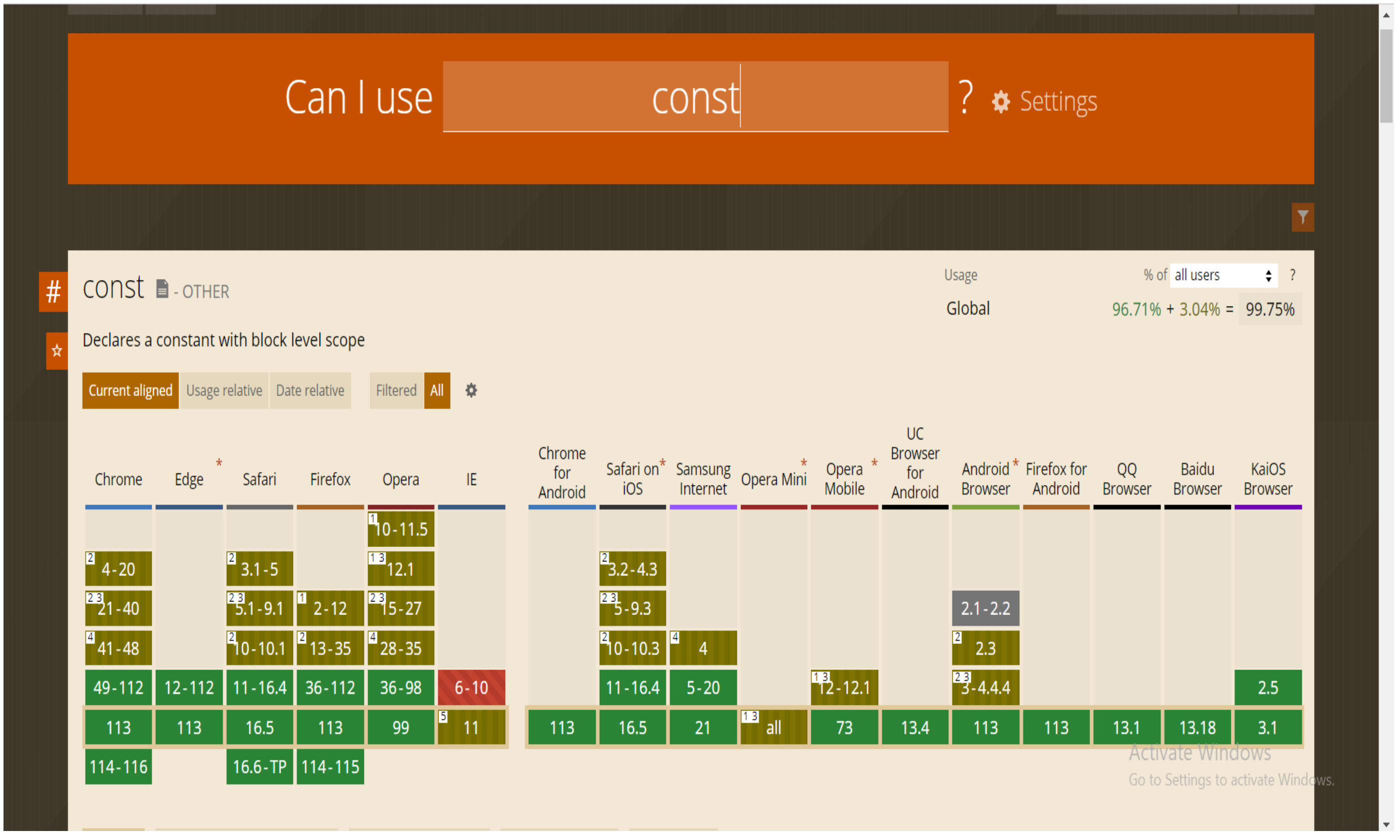
- Chakra is a proprietary JScript engine developed by Microsoft. It is used in the Internet Explorer web browser. Microsoft later developed a new JavaScript engine for its Microsoft Edge browser, which is confusingly also called Chakra. Microsoft Edge switched to the V8 JavaScript engine in 2020.

- Tamarin is a discontinued free software virtual machine with just-in-time compilation (JIT) support intended to implement the 4th edition of the ECMAScript (ES4) language standard. Tamarin source code originates from ActionScript Virtual Machine 2 (AVM2) developed by Adobe Systems, as introduced within Adobe Flash Player 9, which implements ActionScript 3 scripting language. ActionScript Virtual Machine 2 was donated as open-source to Mozilla Foundation on November 7, 2006, to develop Tamarin as a high-performance virtual machine, with the support from broad Mozilla community, to be used by Mozilla and Adobe Systems in the next generation of their JavaScript and ActionScript engines with the ultimate aim to unify the scripting languages across web browsers and Adobe Flash platform and ease the development of better performing rich web applications.

- Each JavaScript engine may have different performance characteristics, optimizations, and compatibility with JavaScript features. They are continuously improved to provide faster execution and better support for the language.

---

5. Show a practical example using caniuse.com



Can I use const ? Settings

const - OTHER

Declares a constant with block level scope

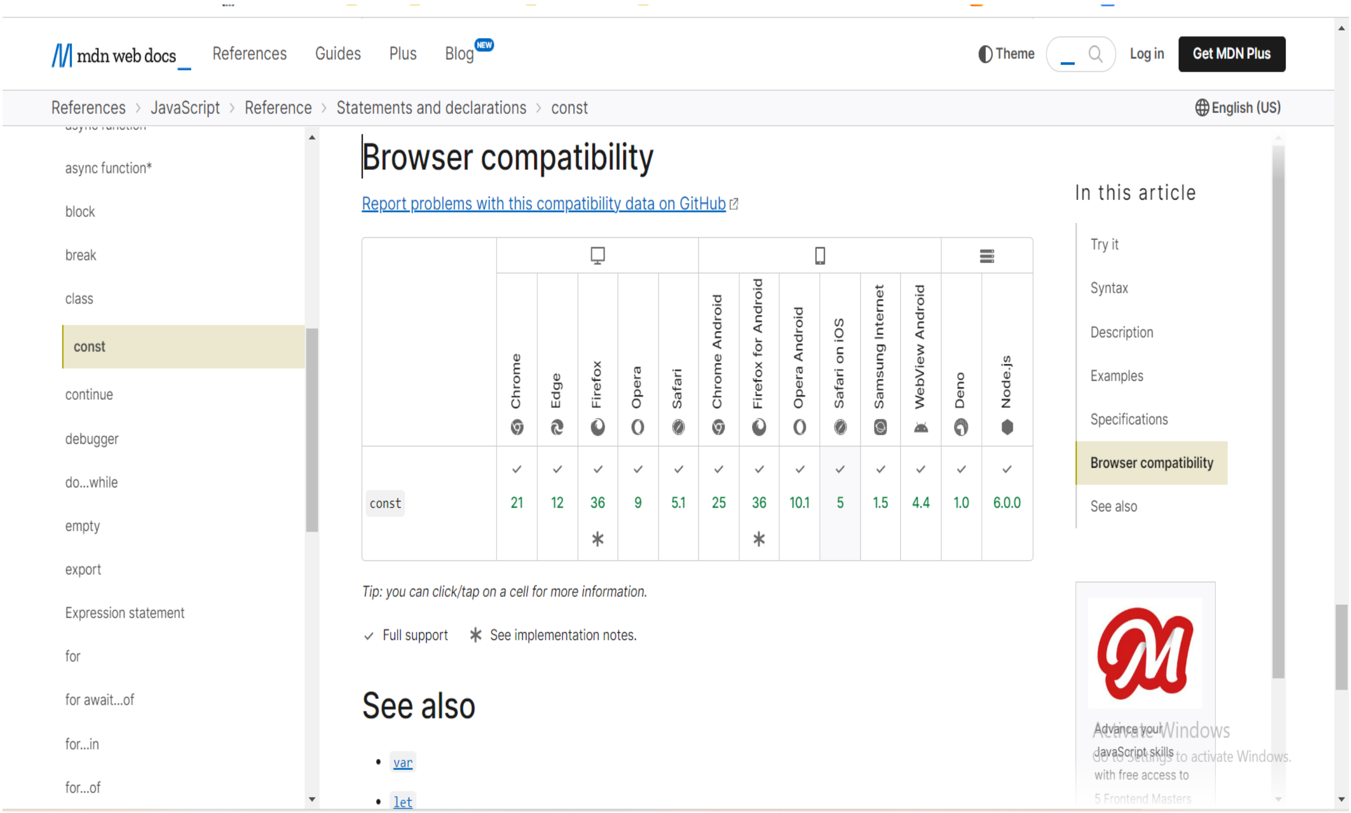
Usage % of all users ?

Global 96.71% + 3.04% = 99.75%

Current aligned | Usage relative | Date relative | Filtered | All

Chrome	Edge	Safari	Firefox	Opera	IE	Chrome for Android	Safari on iOS	Samsung Internet	Opera Mini	Opera Mobile	UC Browser for Android	Android Browser	Firefox for Android	QQ Browser	Baidu Browser	KaiOS Browser
				[1] 10-11.5												
[2] 4-20		[2] 3.1-5		[1 3] 12.1			[2] 3.2-4.3									
[2 3] 21-40		[2 3] 5.1-9.1	[1] 2-12	[2 3] 15-27			[2 3] 5-9.3					2.1-2.2				
[4] 41-48		[2] 10-10.1	[2] 13-35	[4] 28-35			[2] 10-10.3	[4] 4				[2] 2.3				
49-112	12-112	11-16.4	36-112	36-98	6-10		11-16.4	5-20		[1 3] 12-12.1		[2 3] 3-4.4.4				2.5
113	113	16.5	113	99	[5] 11	113	16.5	21	[1 3] all	73	13.4	113	113	13.1	13.18	3.1
114-116		16.6-TP	114-115													

MDN compatibility table.



Browser compatibility

Report problems with this compatibility data on GitHub

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android	Deno	Node.js
const	✓ 21	✓ 12	✓ 36 *	✓ 9	✓ 5.1	✓ 25	✓ 36 *	✓ 10.1	✓ 5	✓ 1.5	✓ 4.4	✓ 1.0	✓ 6.0.0

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

✓ Full support * See implementation notes.

See also

- var
- let