DWA_02.8 Knowledge Check_DWA2

1. What do ES5, ES6 and ES2015 mean - and what are the differences between them?

ES5 (ECMAScript 5) is the fifth edition of the ECMAScript standard and was published in 2009. ES5 was a major revision to Javascript because It introduced several significant changes and additions to the language, both in terms of syntax and functionality. For example: strict mode, new syntax, 'JSON' object, etc.
ES6 is essentially another name for ES2015 which was changed to reflect the year of its release. ES6 introduced many key features such as arrow syntax, 'let' and 'const' variable declarations, template literals, destructuring assignment, etc.
In essence ES5, ES6 and subsequent ECMAScript editions represent different stages of Javascript language development with each edition introducing new features and improvements.

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

JScript is a scripting language that was developed by Microsoft to be similar to JavaScript. JScript was most commonly used as a client-side scripting language in Internet Explorer but with the decline of the web browser due to a movement to more modern browsers, Microsoft shifted its attention away from it in favor of JavaScript.
ActionScript is a scripting language primarily used for creating interactive content within the Adobe Flash platform. ActionScript is closely related to ECMAScript and shares many similarities with JavaScript. Flash technology, including ActionScript, has become less relevant in modern web development due to several factors, including the rise of HTML5, CSS, and JavaScript. Adobe officially ended support for Flash Player in 2020, which further accelerated its decline and its eventual phase out of modern web development.
ECMAScript is a scripting language specification that defines the rules and standards for scripting languages. It serves as the foundation for several programming languages, including JavaScript, JScript, and ActionScript. ECMAScript defines the syntax, semantics, and standard library for these languages, ensuring consistency and interoperability across different implementations

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

A JavaScript specification is a formal document that defines the rules, syntax, semantics, and behavior of the JavaScript programming language. An example of this is the ECMAScript which defines the rules, syntax, semantics, and behavior of the JavaScript programming language. The ECMAScript specification is maintained by Ecma International, a standards organization, and it provides a formalized set of rules and standards that ensure consistency and interoperability for JavaScript across different implementations and platforms. You can go to the ECMA International Website to find the official ECMAScript specification.

---

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

V8, SpiderMonkey, Chakra, and Tamarin are engines inside web browsers with the purpose of understanding and executing JavaScript.

V8 is known for high-performance and speed by using JIT (Just in time) compilation technique to convert JavaScript code into native machine code for faster execution.

SpiderMonkey is the JavaScript engine developed by Mozilla for the Firefox web browser. It was one of the first JavaScript engines created and has gone through various iterations and optimizations over the years.

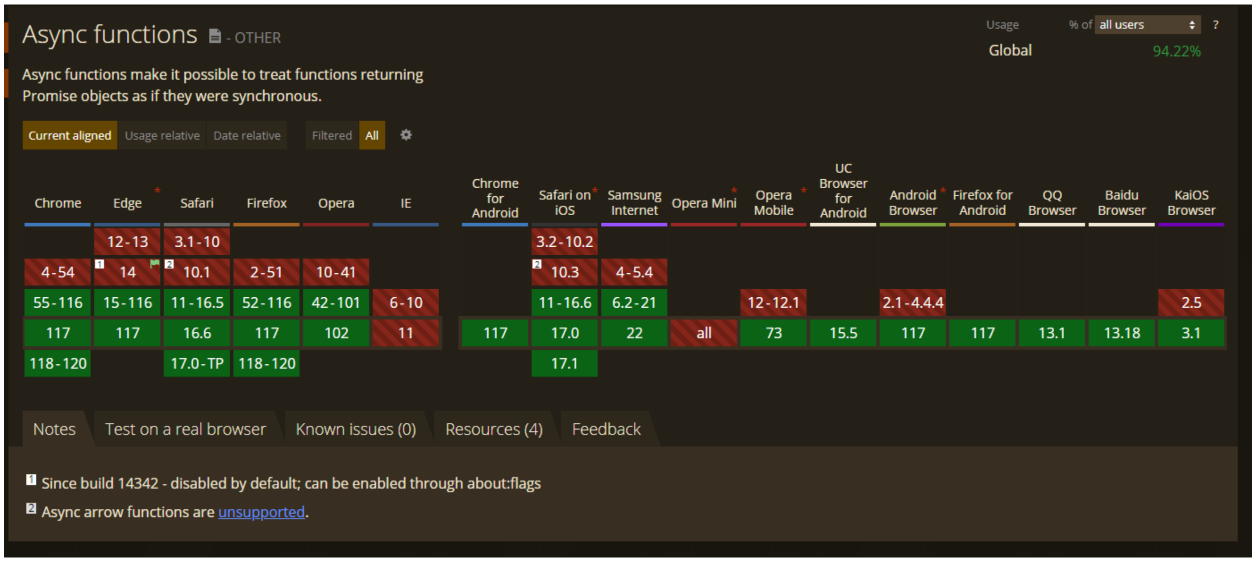
Chakra was the JavaScript engine used in Microsoft Edge browser versions up to 44. It was known for its speed and efficient memory management. However, Microsoft later adopted the Chromium engine, which uses V8 for JavaScript execution.

Tamarin was primarily designed for ActionScript which is used in Adobe Flash Player. Tamarin aimed to provide a high-performance runtime for ActionScript but it is now discontinued.

Each of these engines is designed to execute JavaScript in various contexts, and they have their own strategies for optimizing performance, handling memory management, and providing features. JavaScript code that runs well on one engine may perform differently on another due to the engine-specific optimizations and characteristics. However, they all aim to adhere to the ECMAScript standard to ensure a common baseline of compatibility and functionality.

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

As an example to show how to use caniuse and MDN compatibility table I will use the ‘async’ attribute in html. The `async` attribute is used with the `<script>` element to specify whether the script should be executed asynchronously or not.



Async functions - OTHER

Async functions make it possible to treat functions returning Promise objects as if they were synchronous.

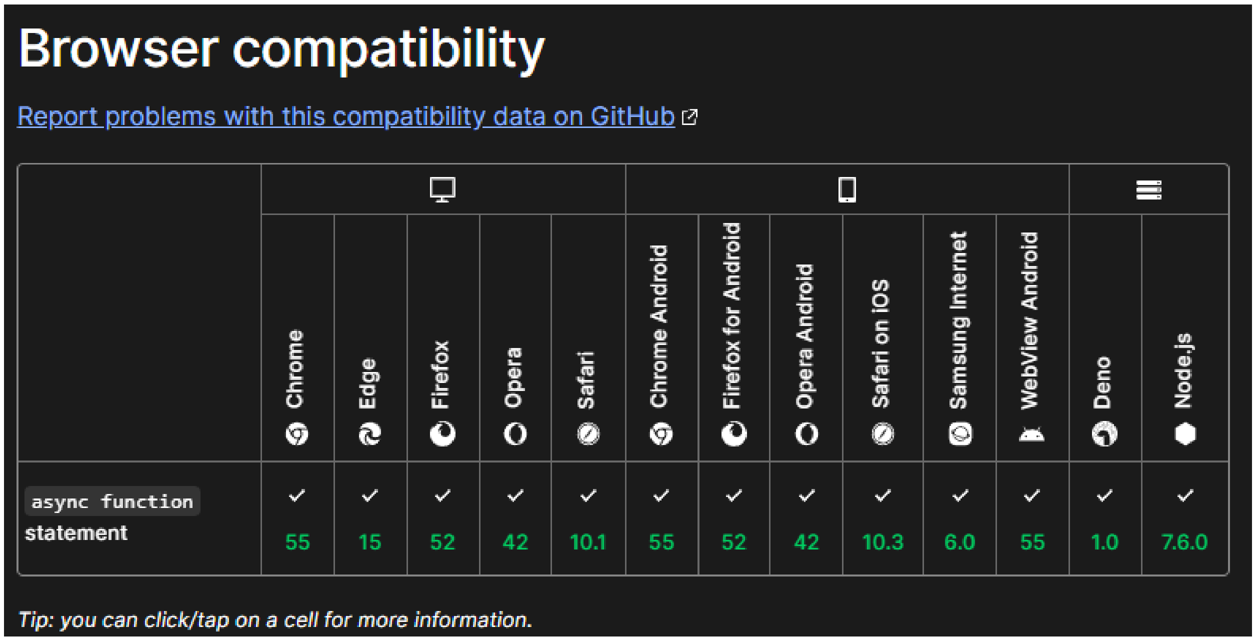
Usage % of all users ?

Global 94.22%

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
	12-13	3.1-10					3.2-10.2									
4-54	[1] 14	[2] 10.1	2-51	10-41			[2] 10.3	4-5.4								
55-116	15-116	11-16.5	52-116	42-101	6-10		11-16.6	6.2-21		12-12.1		2.1-4.4.4				2.5
117	117	16.6	117	102	11	117	17.0	22	all	73	15.5	117	117	13.1	13.18	3.1
118-120		17.0-TP	118-120				17.1									

[1] Since build 14342 - disabled by default; can be enabled through about:flags

[2] Async arrow functions are unsupported.



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
`async function` statement	✓ 55	✓ 15	✓ 52	✓ 42	✓ 10.1	✓ 55	✓ 52	✓ 42	✓ 10.3	✓ 6.0	✓ 55	✓ 1.0	✓ 7.6.0

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

By using Caniuse and the MDN compatibility table I am able to gather information about the browser compatibility of the 'async' attribute. This is important as it allows us to ensure that our code functions as expected across different browsers.