DWA_02.8 Knowledge Check_DWA2

---

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

ES5 stands for ECMAScript 5 and was the 5th generation of ECMAScript, which was created to standardise the JavaScript programming language. ES6 was the generation after ES5 and brought updates such as deconstructing into the standardised JS. ES2015 stands for ECMAScript 2015 and highlights the year in which this updated script was released.This is because the ECMAScript committee decided to move to annual updates hence ES6 edition was renamed ES2015 to reflect the year of release.

---

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

JScript was created by Microsoft during the browser wars and is a reverse engineered version of the then ' ‘JavaScript”. ActionScript is a ECMAScript-based object oriented programming language that came out in 1998, later acquired by Adobe. ECMAscript is the standardised version of JavaScript, designed to make it easier to operate web pages across different browsers. This was developed in 1997.

---

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

ECMAScript is a standardised version of javascript that is regularly updated i.e. ES5, ES6, ES7 etc. an example of a specification would be ECMAScript 6 (ES6), also known as ECMAScript 2015. ES6 introduced a number of specifications and updates to the language like Block-scoped variables, Template literals, Arrow Functions.

---

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

These are all examples of different JavaScript compilers. There are more than 39 different compilers being used for JavaScript . A compiler is a software that converts high-level code that developers write into low-level binary code which can be understood by the computer.

---

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

Let's consider a practical example of using Caniuse.com and the MDN compatibility table for the CSS feature "flexbox."

1. Start by visiting caniuse.com in your web browser.
2. In the search bar on the homepage, type "flexbox" and press Enter or click the search icon.
3. The search results will display information about the "flexbox" feature, including browser support tables, usage statistics, and related resources.
4. Scroll down to the support table section. This table shows the level of support for the "flexbox" feature across different browsers and versions.
5. Each browser version is represented by a coloured square in the table. Green squares indicate full support, yellow squares indicate partial support with some limitations, and red squares indicate no support.
6. You can hover over each square to see detailed information about the specific version's support.
7. Now, to access additional compatibility data from MDN, look for the "MDN" tab located below the support table.
8. Click on the "MDN" tab to view the MDN compatibility table for the "flexbox" feature.
9. The MDN compatibility table provides more detailed information about the feature's support, including browser versions, release dates, and compatibility notes.
10. You can explore the MDN compatibility table to gain a deeper understanding of how the "flexbox" feature is supported across different browsers.

By combining the information from Caniuse.com and the MDN compatibility table, developers can make informed decisions about using the "flexbox" feature and ensure its compatibility with the targeted browsers.