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 version designations for the ECMAScript (ECMA-262) standard, which defines the JavaScript programming language. The latest version of this standard is ES2023.

- ES5 (ECMAScript 5) was released in 2009 and represents a widely adopted version of JavaScript. It introduced features like strict mode, JSON support, and object getters and setters.
- ES6 (ECMAScript 2015) was released in 2015 as a substantial update to JavaScript. It introduced significant enhancements such as arrow functions, classes, modules, and template literals. ES2015 is synonymous with ES6.
- ES2015 is another name for ES6.

Differences between ES5 and ES6
Here are some of the key differences between ES5 and ES6:

FEATURE	ES5	ES6
Arrow functions	NO	YES
Classes	NO	YES
Modules	NO	YES
Template literals	NO	YES
DEfault functions params	NO	YES
Spread operator	NO	YES
Rest operator	NO	YES

---

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

- JScript is a JavaScript implementation by Microsoft, primarily used in Internet Explorer. It was released in 1996 and is a variant of JavaScript.
- ActionScript is a dialect of JavaScript that was used for creating Adobe Flash animations. It was introduced in 1999 but is no longer widely used due to the discontinuation of Flash.

- ECMAScript is the official specification that defines the JavaScript programming language. It is maintained by Ecma International, an international standards organization. JavaScript is an implementation of 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 standard itself, which defines the language's features, behavior, and syntax. You can access the ECMAScript standard on the official Ecma International website at this URL: ECMAScript Standard.

---

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

- v8, SpiderMonkey, Chakra, and Tamarin are all JavaScript engines responsible for interpreting and executing JavaScript code.

While these engines may have some implementation differences, they aim to adhere to the ECMAScript standard. This ensures that JavaScript code is generally compatible across different engines. Any variations usually pertain to performance optimizations and additional features rather than fundamental differences in how JavaScript is executed.

---

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

Let's say you're developing a web application and want to use arrow functions in your JavaScript code. To ensure that arrow functions are supported in the majority of web browsers, you can use caniuse.com and the MDN compatibility table.

Using caniuse.com:
a. Visit the caniuse.com website (https://caniuse.com/).
b. In the search bar, type "arrow functions."
c. The website will display a table that provides information about the support for arrow functions in different web browsers. It will show the percentage of users covered by each browser version.
d. For example, you may see that arrow functions are supported in modern versions of Chrome, Firefox, Edge, and Safari, with a high percentage of users covered.
e. This information helps you gauge whether it's safe to use arrow functions in your project, considering your target audience and the browsers they commonly use.

Using the MDN Compatibility Table:

a. To get more detailed information and documentation about arrow functions, visit the Mozilla Developer Network (MDN) page for arrow functions (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/Arrow_functions).

b. On the MDN page, scroll down to the "Browser compatibility" section.

c. Here, you'll find a compatibility table that shows which versions of major browsers support arrow functions and any potential issues or notes related to their usage.

d. For instance, the MDN table may confirm that arrow functions are supported in recent versions of Chrome, Firefox, Edge, and Safari, and it may provide additional information about specific versions or exceptions.

By using both caniuse.com and the MDN compatibility table, you can make informed decisions about implementing arrow functions in your JavaScript code, ensuring compatibility with your target audience's web browsers. This approach helps create a more robust and user-friendly web application.
