DWA_02.8 Knowledge Check_DWA2

---

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

- ES5 introduced significant improvements to JavaScript, while ES6 brought about major changes and added many new features and syntax enhancements. Both ES5 and ES6 are widely used, with Es6 providing a more modern and expressive coding experience.

---

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

- Jscript and ActionScript are scripting languages that are based on ECMAScript specification, with Jscript being a Microsoft-specific implementation and ActionScript being used in Adobe Flash, JavaScript is the most popular and widely used implementation of ECMAScript, and it is the language commonly used for web development.

---

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

- The specification provides in-depth details about the JavaScript language, including its syntax, built-in objects, data types, control structures, and more. It is a comprehensive document that is primarily intended for language implementers and developers who want a deep understanding of how JavaScript works.
- It's important to note that the specification can be quite technical and may not be the most beginner-friendly resource for learning JavaScript. However, it serves as a valuable reference for understanding the inner workings of the language and exploring its finer details

---

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

- V8, Spider Monkey, Chakra, and Tamarin are JavaScript engines, each used by different web browsers or platforms to execute JavaScript code. While they all serve the purpose of running JavaScript, there are differences in their design, implementation, and performance characteristics. Here's an overview of each JavaScript engine:

  - V8: V8 is the JavaScript engine developed by Google and primarily used in the Google Chrome browser. It is a high-performance engine that compiles JavaScript code into highly optimized machine code. V8 is known for its speed and efficient memory management. It utilizes just-in-time (JIT) compilation techniques to dynamically optimize and execute JavaScript code.
  - Spider Monkey: Spider Monkey is the JavaScript engine developed by Mozilla and used in the Firefox web browser. It was the first-ever JavaScript engine created and has gone through significant improvements over the years. Spider Monkey initially used an interpreter, but later versions introduced a JIT compiler for improved performance. It supports various JavaScript features, including those defined in the ECMAScript specifications.
  - Chakra: Chakra, also known as Chakra Core, is the JavaScript engine developed by Microsoft. It was originally used in the Internet Explorer browser and later became the engine for Microsoft Edge. Chakra underwent significant improvements and optimizations over time. In 2019, Microsoft announced that it would transition to using the Blink rendering engine (which uses V8) for its Edge browser, discontinuing active development of Chakra.
  - Tamarin: Tamarin is a JavaScript engine that was developed by Adobe Systems and is based on the open-source Mozilla Spider Monkey engine. Tamarin was specifically designed to execute JavaScript within the Adobe Flash Player, providing enhanced performance for ActionScript (which is based on ECMAScript). However, Adobe discontinued active development of Tamarin in 2012.

- While these JavaScript engines have their unique characteristics, they all aim to execute JavaScript code and adhere to the ECMAScript specifications. However, differences in implementation and optimization strategies can lead to variations in performance and compatibility with certain JavaScript features. The engines constantly evolve and improve to provide better performance and support for modern JavaScript language features

---

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

- caniuse.com:
    - Go to https://caniuse.com.
    - In the search bar, type "grid" and select the "CSS grid" option.
    - The page will display information about the CSS grid feature, including its browser compatibility.
    - You can see a table that shows the level of support for the grid property across different browsers and versions.
    - The table provides a color-coded legend indicating full support, partial support, vendor prefix requirements, and lack of support.

- While these JavaScript engines have their unique characteristics, they all aim to execute JavaScript code and adhere to the ECMAScript specifications. However, differences in implementation and optimization strategies can lead to variations in performance and compatibility with certain JavaScript features. The engines constantly evolve and improve to provide better performance and support for modern JavaScript language features

- MDN compatibility table:
    - Go to the MDN web docs at https://developer.mozilla.org.
    - In the search bar, type "grid" and select the "CSS Grid Layout" option.
    - On the CSS Grid Layout documentation page, scroll down to the "Browser compatibility" section.
    - Here, you'll find a compatibility table that provides detailed information about the support for various CSS grid properties across different browsers and versions.
    - The table includes information about when a feature was introduced, whether it's supported with vendor prefixes, and any known issues or limitations.

- Using these resources, you can easily check the browser compatibility of CSS properties, JavaScript APIs, and other web technologies. This allows you to make informed decisions when choosing which features to use in your web projects and determine if you need to provide fallbacks or alternative approaches for older browsers.