Keotshepile Maje

DWA_02.8 Knowledge Check_DWA2

---

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

ES5 is ECMAScript 5 also known as javascript5. This version was released in 2009 with some improvements. Features and Improvements made:

- 'Use strict' =>
  - It enforces stricter rules that help catch common coding mistakes.
- JSON support =>
  - (JavaScript Object Notation) is a lightweight data-interchange format that is widely used for storing and transmitting structured data between a server and a client, or between different parts of an application.
- string.trim() =>
  - trim() method removes whitespace from both sides of a string.
- Array.isArray() =>
  - It checks whether an object is an array.
- Array iteration methods =>
  - Allow you to perform operations on each element of an array. These methods make it easier to work with arrays and perform common tasks without explicitly writing loops.
- Trailing commas for object literals =>
  - It adds the ability to place a comma after the last property value in an object literal without causing a syntax error.

ES6 is a major update to the ECMAScript specification. This version was released in 2015 and its official name is ES2015 (ECMAScript 2015) after it was ratified by ECMAScript in June 2015. Many new features and syntax enhancements were introduced to JavaScript, making the language more expressive and powerful. Some of the features that were introduced in ES6:

- Block-scope variables =>
  - let
  - const
- Arrow functions =>
  - It allows a short syntax for writing function expressions.

- Template literals =>
    - Provides a convenient way to create a string that can include dynamic variables or expressions.
- Classes =>
    - These are templates for JavaScript Objects.
- Modules =>
    - The module is used to import and export javascript file/ code from one to another.
- Destructuring assignment =>
    - This feature allows you to extract values from arrays/ objects and assign them to a variable in a more concise and intuitive way. It provides a convenient syntax for unpacking values from data structures.

---

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

They share a common foundation rooted in ECMAScript.

JScript

- ➢ JScript shares many similarities with JavaScript but it is not the same thing. It is Microsoft's implementation of ECMAScript, which standardized specifications for JavaScript.

ActionScript

- ➢ It was developed by Macromedia and later acquired by Adobe.
- ➢ ActionScript is an ECMAScript-based language closely related to JavaScript.
- ➢ It lets you add complex interactivity control, and data display to your application and websites. It shares similar syntax and core features with JavaScript.

ECMAScript

- ➢ It is the standardized specification for JavaScript programming language.
- ➢ It provides a common language specification that browsers and other environments can implement.

---

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

ECMA specification (ECMA-262) => It contains the standardized specification for javascript. Which defines the syntax, semantics, and behavior of the language, ensuring consistent implementation across different JavaScript engines and platforms.

You can find them at:
The ECMA international website and the GitHub repository.
https://tc39.es/ecma262/ or https://github.com/tc39

---

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

These are JavaScipt engines developed by different organizations, responsible for interpreting and executing JavaScript code. All these engines strive to interpret and execute JavaScript code according to ECMAScript specifications.

V8

- ➢ Developed by Chrome team at Google
- ➢ V8 utilizes just-in-time(JIT) compilation techniques to optimize JavaScript during runtime, providing fast execution speeds and high performance.

Spider-monkey

- ➢ Developed by Mozilla Foundation
- ➢ Spider-monkey was the first ever JavaScript engine. It employs various optimization techniques, including JIT compilation, to execute JavaScript efficiently.

Chakra

- ➢ Developed by Mircosoft
- ➢ It utilizes both JIT compilation and background compilation techniques to optimize JavaScript execution

Tamarin

- ➢ Developed by Adobe Systems and Mozilla
- ➢ It was specifically designed for the ActionScript and Javascript languages. It uses a JIT compiler known as NanoJIT aimed to provide efficient execution speeds.

---

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

As a developer when you want a particular feature that functions in a desired way, and want it to work on every browser that the user uses. You have to check which browsers use or are compatible with that feature on caniuse and MDN before using it.
