DWA_02.8 Knowledge Check_DWA2

---

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

ECMAScript is a trademarked scripting language specification that is defined by ECMA International to standardize JavaScript. ECMAScript is used for client-side scripting of the World Wide Web.

ES5 -an abbreviation of ECMAScript 5, also known as ECMAScript 2009, is the fifth edition of the ECMAScript introduced in 2009. It supports primitive data types that include string, number, boolean, null and undefined. The only way to define variables is by using the var keyword, and both function and return keywords are used to define a function. The most generally used loop is the **for** loop. Formed the basis of modern JS development, array methods.

ES6 -also known as ES2015, is the sixth edition of the EMCAScript introduced in 2015. In addition to the data types implemented in the previous edition, this version introduces a new primitive data type ‘symbol’ for supporting unique values. There are also two new ways to define variables; let and const. The performance of ES6 has a higher performance than ES5 due to the new features and shorthand storage implementation. Furthermore, unlike ES5, which was much more time consuming to use. ES6 introduces destructuring and speed operators, thus object manipulation can be processed more efficiently. In addition to the other new features, ES6 also introduces an arrow function by which developers don’t require the function keyword to define the function. Lastly ES6 introduced the concept of **for..of** loop to perform an iteration over the values of iterable objects.

The names ES6 and ES2015 are sometimes used interchangeably, with ES6 being the more popularized name prior to release. However, the committee that oversees ECMAScript specifications made the decision to move to annual updates, thus changing the name to ES2015 to reflect the year of release. Because of this decision to have annual updates, subsequent releases will therefore also be named according to the year of release.

---

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

JScript is a scripting language owned by Microsoft and was developed in 1996. It can create active online content for the World Wide Web, unlike JavaScript. While JScript is only supported by Microsoft Internet Explorer and therefore can only run in Microsoft Browser, all modern day browsers can support JavaScript. JScript can access objects of Microsoft browser, however due its limitations to Microsoft browser, it is not as popular as JavaScript.

ActionScript is an object-oriented scripting language designed to provide rich interactive abilities to the Adobe Flash Player platform. ActionScript's syntax is very similar to that of JavaScript, as both are based on the same ECMAScript standard. It was originally introduced by Macromedia for its Flash authoring tool, however it is now developed and supported by Adobe Systems. The language is also open source.

ECMAScript is a general-purpose scripting language that is standardized by ECMA International and is implemented in JavaScript and some other languages. Created in 1997, ECMAScript was based on JavaScript, which was created by Netscape, originally known as Mosaic). Although there is some confusion as to which one was developed first -JavaScript or ECMA, according to the timeline, JavaScript was in fact created first in 1995 by Netscape for the Netscape Navigator browser. In 1995, when JavaScript was created, it was mainly used to run on web browsers, specifically the Netscape Navigator browser, which was the most popular web browser at the time. Due to the popularity of JavaScript, competing browser companies began developing scripting languages of their own, because Javascript was not a shared language, thus each browser had to create their own variations of JavaScript.
In 1996, Netscape submitted its JavaScript to Ecma International to make Standards happen for scripting languages. In response, Ecma International set up a Technical Committee, known as TC39, for developers to collaborate and evolve the definition of JavaScript and scripting languages. TC39 was assigned to standardize the language to make it a general purpose, cross-platform and vendor-neutral scripting language.
Thus, in 1997, TC39 came up with the standardized JavaScript called ECMAScript that is specified under ECMA-262 Specification.
JS now classified as an implementation of EMCAScript.

---

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

An example of a JavaScript specification is variable names. Variable names in JavaScript recommend the camelCase method to name variables. This specification can be found in a statement that declares a variable in JavaScript.

ECMAScript specifications are maintained by Ecma International and are freely available on their website. An example would be ES2015, which is the latest version.

Using let or const to define variables which was introduced in ES2015, compared to the previous way of defining a variable by using var.

---

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

These are different JavaScript compilers.
V8- A JavaScript engine used in Google Chrome and other Chromium-based browsers.

SpiderMonkey- a Javascript engine used in Mozilla Gecko applications, including Firefox. It currently includes an IonMonkey compiler and OdinMonkey optimization module.

Chakra- A JavaScript engine previously used in older versions of Microsoft Edge, before being replaced by V8.

Tamarin- An ActionScript and ECMAScript engine used in Adobe Flash.
Different engines execute JS differently based on the browser. May have different features and browser support systems.

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