DWA_02.8 Knowledge Check_DWA2

---

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

ES5 is also known as ECMAScript 2009 that was released in 2009. It is a function contractors focus on how the objects are instantiated. For ES5 you have to write a function, keyword and return to be used to define the function, like normal general JavaScript language.

ES6 is also known as ECMAScript 2015 that was released in 2015. Its class allows the developers to instantiate an object using the new operator, using an arrow function, in case it doesn't need to use function keyword to define the function, also return keyword can be avoided to fetch the computer value.

ES6 and ES2015 are the same thing. ES6 was the popularized name prior to release. However, the committee that oversees ECMAScript specifications made the decision to move to annual updates. With this change, the edition was renamed to ES 2015 to reflect the year of release.

The differences between them is the way the primitive data types, Manipulation of objects, functions and variables are being written.

---

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

JScript - JScript implemented as an Active Scripting engine. This means that it can be plugged into Automation applications that support Active Scripting, such as Internet Explorer, Active Server Pages, and Windows Script Host.

ActionScript - ActionScript is an object-oriented programming language originally developed by Macromedia and later acquired by Adobe. It is influenced by HyperTalk, the scripting language for HyperCard.

ECMAScript - ECMAScript is a standard for scripting languages, including JavaScript, JScript, and ActionScript. It is also best known as a JavaScript standard intended to ensure the interoperability of web pages across different web browsers. ECMAScript is a subset of JavaScript. JavaScript is basically ECMAScript at its core but builds upon it.

---

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

Language syntax,, Error handling mechanisms, Types like boolean, number, string, function, object, etc.. , Built-in objects and functions, including JSON, Math, Array methods, parseInt, etc. , Strict mode, A module system, Basic memory model.

You can find them by MDN (Mozilla) JavaScript Reference or any Coding website

---

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

V8 - V8 is a free and open-source JavaScript and WebAssembly engine developed by the Chromium Project for Chromium and Google Chrome web browsers.

SpiderMonkey - SpiderMonkey is open-source JavaScript and WebAssembly engine by the Mozilla Foundation. It is the first JavaScript engine, written by Brendan Eich at Netscape Communications, and later released as open source and currently maintained by the Mozilla Foundation. It is used in the Firefox web browser.

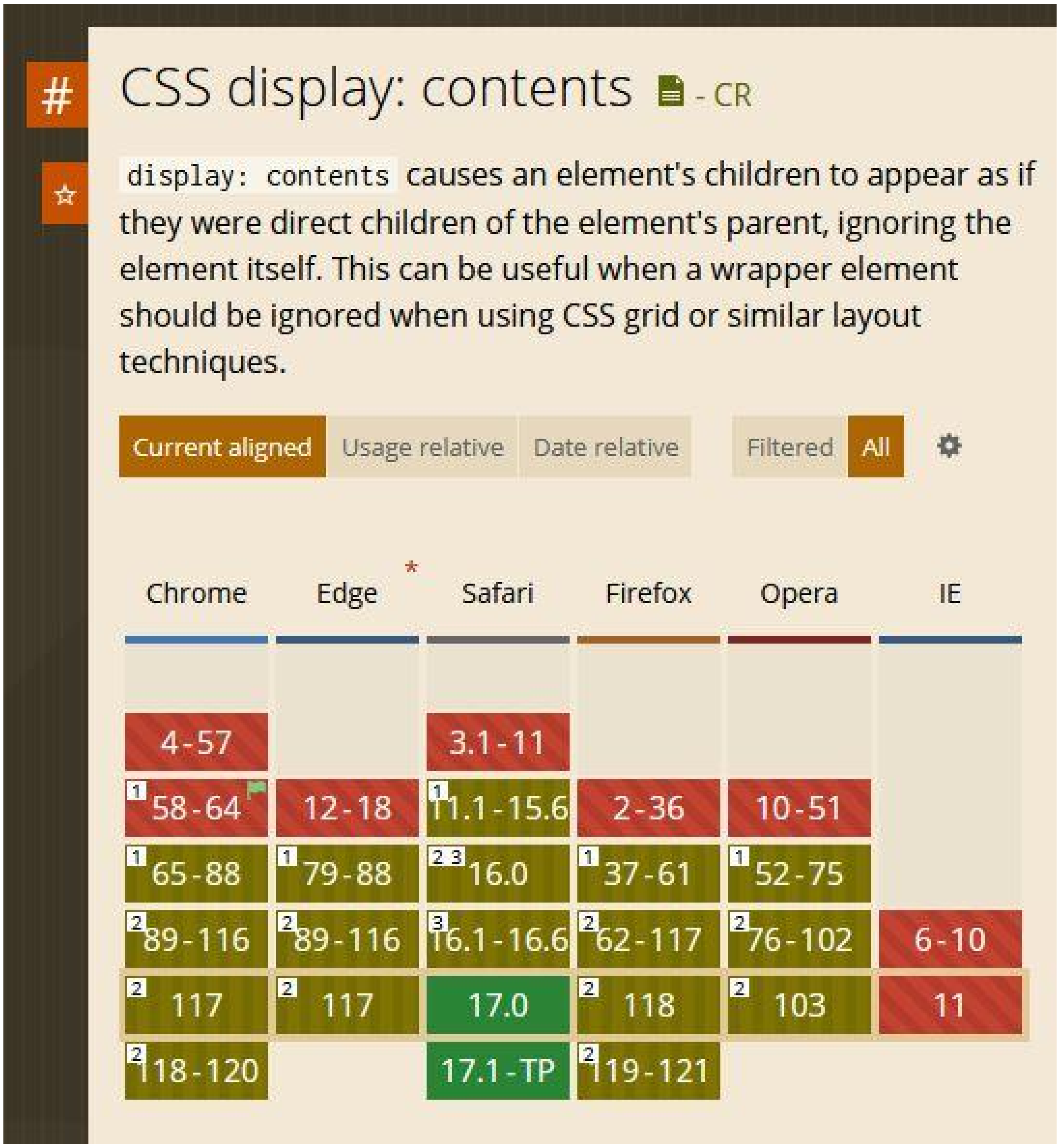
Chakra - Chakra is based on the JScript version, but it has been redesigned to improve performance in Internet Explorer at the expense of proper Active Scripting engine compatibility.

Tamarin - Tamarin is a discontinued free software virtual machine with just-in-time compilation support intended to implement the 4th edition of the ECMAScript language standard.

No, they don't run JavaScript differently because they are JavaScript and WebAssembly engines that run in different web browsers.

---

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

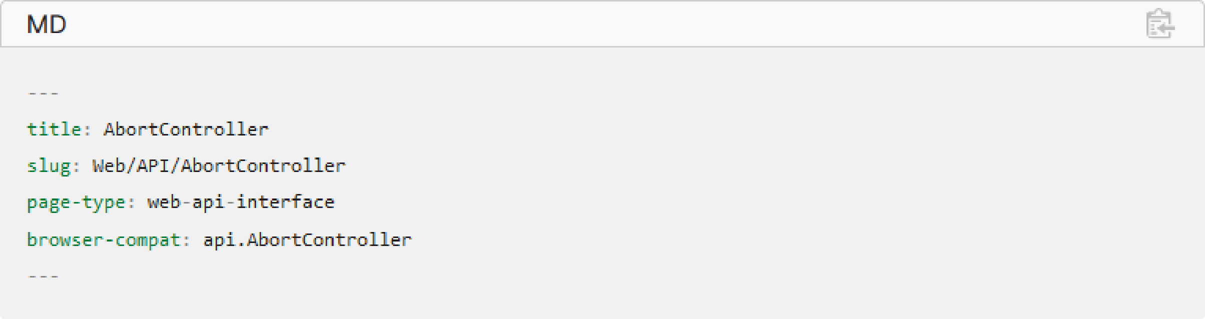


CSS display: contents - CR

`display: contents` causes an element's children to appear as if they were direct children of the element's parent, ignoring the element itself. This can be useful when a wrapper element should be ignored when using CSS grid or similar layout techniques.

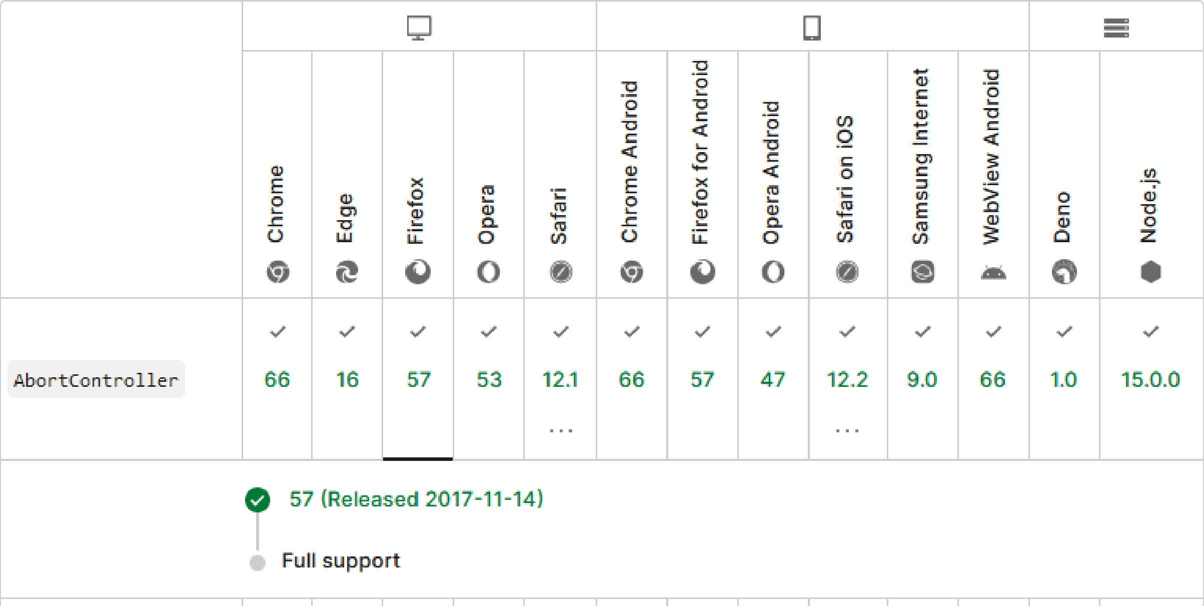
Chrome	Edge *	Safari	Firefox	Opera	IE
4-57		3.1-11			
[1] 58-64	12-18	[1] 11.1-15.6	2-36	10-51	
[1] 65-88	[1] 79-88	[2] [3] 16.0	[1] 37-61	[1] 52-75	
[2] 89-116	[2] 89-116	[3] 16.1-16.6	[2] 62-117	[2] 76-102	6-10
[2] 117	[2] 117	17.0	[2] 118	[2] 103	11
[2] 118-120		17.1-TP	[2] 119-121		

For example, `AbortController` would be added as shown below:



MD

```
---
title: AbortController
slug: Web/API/AbortController
page-type: web-api-interface
browser-compat: api.AbortController
---
```



	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android	Deno	Node.js
`AbortController`	✓ 66	✓ 16	✓ 57	✓ 53	✓ 12.1 …	✓ 66	✓ 57	✓ 47	✓ 12.2 …	✓ 9.0	✓ 66	✓ 1.0	✓ 15.0.0

✓ 57 (Released 2017-11-14)

● Full support