DWA_02.8 Knowledge Check_DWA2

---

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

ES is the Standardized version of JavaScript which follows the ECMAScript standard and a scripting language.

ES5 is the version of ECMAScript which was released after the termination of ES4 and mostly contains array, date and string features.

ES6 and ES2015 are the same version that the TC-39 decided to release in 2015 changing the naming structure to represent the date it was released instead of a version number. ECMAScript6 (ES6/ES2015 in short) builds on ECMAScript5 and added variables, arrow functions/anonymous functions, ternaries and classes to name a few features.

---

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

JScript is a Microsoft reverse engineered version of JavaScript which was created to compete with the functionality of other browsers at the time while avoiding trademark infringement and capitalizing on the popularity of JavaScript.

ActionScript is Adobe’s version of JavaScript and it mainly took from the ES4 development before it was phased out. It was thus an implementation of JavaScript in Adobe Flash that tried to stay ahead of the market by adopting ES4 standardized features and syntax from early development of the standard.

ECMAScript is a standardization for scripting languages that came about when the race for interactive Web Design was at its peak and caused many companies to develop their own versions of JavaScript. It is now the standard which JavaScript (a dialect of ECMAScript) adheres to.

ECMAScript is thus a set of rules and guidelines which dictate what a language must do or be in order to be considered a dialect of ECMAScript.

---

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

The ECMA-262 is a specification guideline which contain rules such as:
“A conforming implementation of ECMAScript must provide and support all the types, values, objects, properties, functions, and program syntax and semantics described in this specification.” - https://262.ecma-international.org/14.0/#sec-conformance

The specifications can be found in either the current release’s ECMA-262 web page or the changelog and proposed features for the next version on the TC-39 github page.

---

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

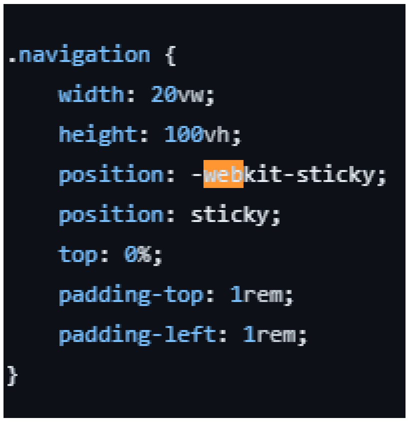
These are all examples of JavaScript compilers developed by different Web Browser companies. These are basically a body with which the commands outlined in a JavaScript document are performed.

They differ in how up to date they are with the current specifications, features and syntax outlined by the EMCAScript versions. This means that some compilers may contain multiple new features while others might still run closely to the ES3 version of JavaScript due to the huge amount of changes introduced in ES5.

V8 is a compiler developed and used by Google Chrome and implemented in things like node.js. SpiderMonkey is developed by Mozilla Firefox and was one of the first compilers to become main-stream. Chakra is the name for a legacy version compiler and the current compiler for Microsoft edge and previously internet explorer. Lastly Tamarin is a discontinued compiler run by Adobe which focused on the ES4 specification.

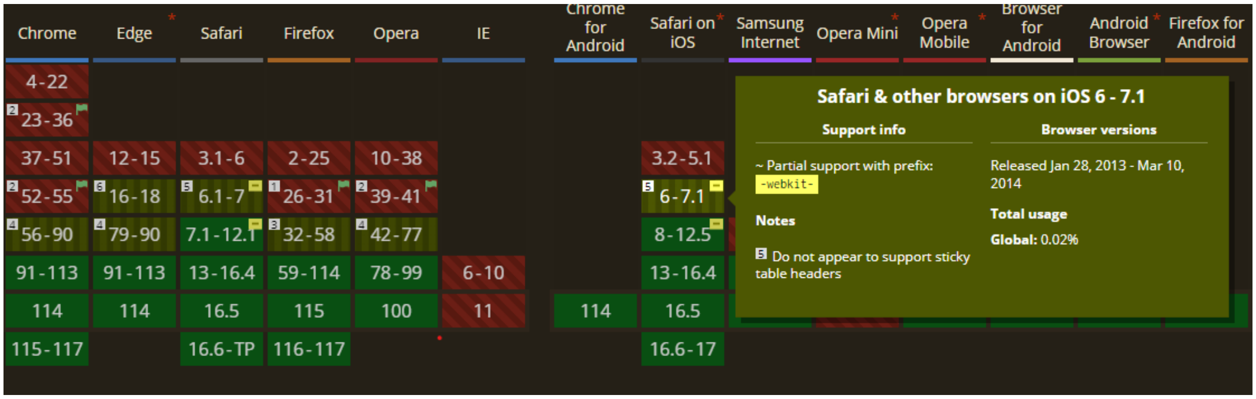
---

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



```
.navigation {
    width: 20vw;
    height: 100vh;
    position: -webkit-sticky;
    position: sticky;
    top: 0%;
    padding-top: 1rem;
    padding-left: 1rem;
}
```

A practical example can be seen in this code from my ITW CSS file:
Here you can see that the position: sticky; feature is duplicated. This is so the prefix -webkit- can be added in order to call the webkit engine which allows some older versions of the browsers to run the sticky position feature.



Chrome	Edge *	Safari	Firefox	Opera	IE	Chrome for Android	Safari on iOS *	Samsung Internet	Opera Mini *	Opera Mobile *	Browser for Android	Android Browser *	Firefox for Android
4-22													
[2] 23-36													
37-51	12-15	3.1-6	2-25	10-38			3.2-5.1						
[2] 52-55	[6] 16-18	[5] 6.1-7	[1] 26-31	[2] 39-41			[5] 6-7.1						
[4] 56-90	[4] 79-90	7.1-12.1	[3] 32-58	[4] 42-77			8-12.5						
91-113	91-113	13-16.4	59-114	78-99	6-10		13-16.4						
114	114	16.5	115	100	11	114	16.5						
115-117		16.6-TP	116-117				16.6-17						

Safari & other browsers on iOS 6 - 7.1

Support info

~ Partial support with prefix: `-webkit-`

Notes

[5] Do not appear to support sticky table headers

Browser versions

Released Jan 28, 2013 - Mar 10, 2014

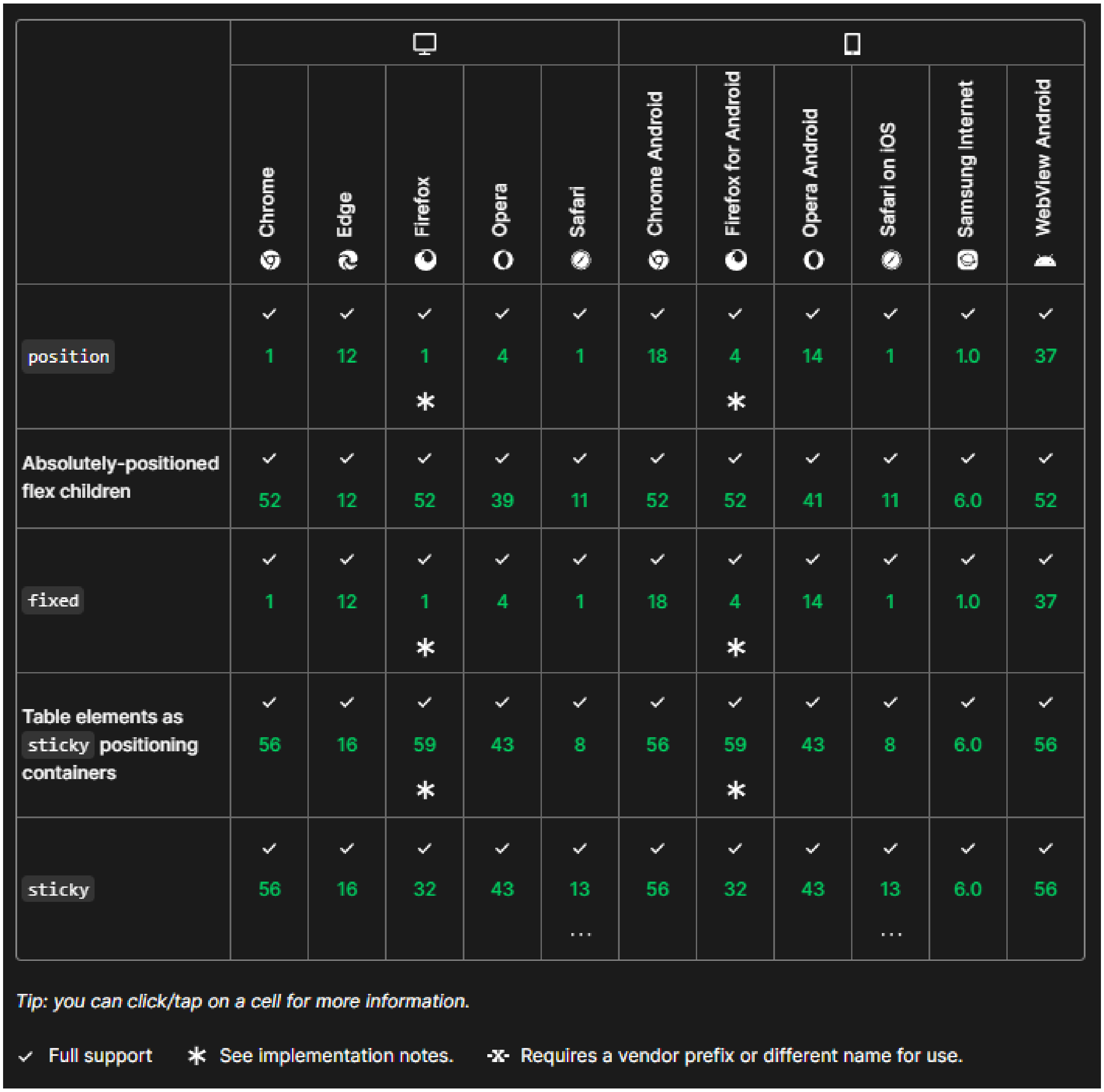
Total usage

Global: 0.02%

However as seen above some browsers still do not support the feature such as Internet Explorer (now deprecated) as well as opera mini.

By looking at this table we can decide if the feature should be implemented at all say for instance if our user base is primarily Internet Explorer users or see how we would need to implement the feature to ensure as many users can use the page as intended (as shown by the webkit example.)

The mdn table shows us that there are still some requirements to the more advanced case uses of sticky and that some browser requires further workaround:



	🖥					📱					
	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android
`position`	✓ 1	✓ 12	✓ 1 ✱	✓ 4	✓ 1	✓ 18	✓ 4 ✱	✓ 14	✓ 1	✓ 1.0	✓ 37
Absolutely-positioned flex children	✓ 52	✓ 12	✓ 52	✓ 39	✓ 11	✓ 52	✓ 52	✓ 41	✓ 11	✓ 6.0	✓ 52
`fixed`	✓ 1	✓ 12	✓ 1 ✱	✓ 4	✓ 1	✓ 18	✓ 4 ✱	✓ 14	✓ 1	✓ 1.0	✓ 37
Table elements as `sticky` positioning containers	✓ 56	✓ 16	✓ 59 ✱	✓ 43	✓ 8	✓ 56	✓ 59 ✱	✓ 43	✓ 8	✓ 6.0	✓ 56
`sticky`	✓ 56	✓ 16	✓ 32	✓ 43	✓ 13 …	✓ 56	✓ 32	✓ 43	✓ 13 …	✓ 6.0	✓ 56

Tip: you can click/tap on a cell for more information.

✓ Full support ✱ See implementation notes. -x- Requires a vendor prefix or different name for use.

MDN goes into far more detail with regards to all use cases when looking for compatibility. For Instance if a table element should have position sticky you would need to make sure that the bug as seen below would not cause major interference or that the work around by displaying like a chrome browser would does not effect your webpage.

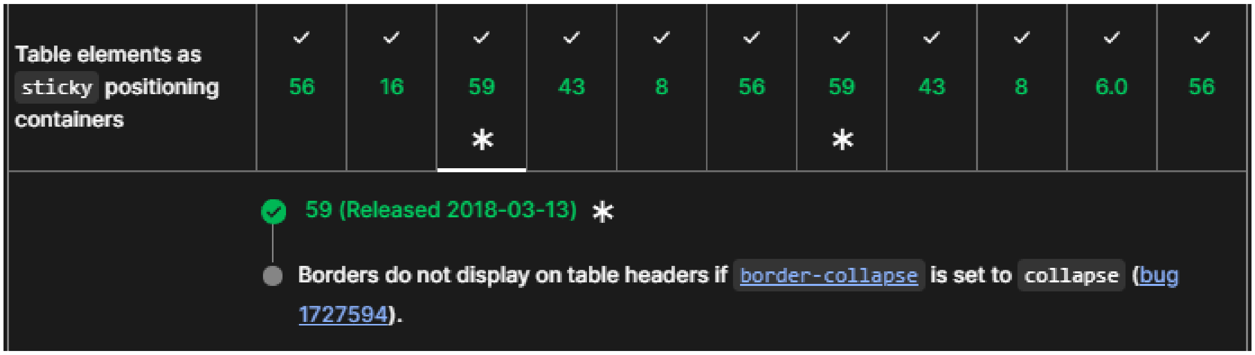


Table elements as `sticky` positioning containers	✓ 56	✓ 16	✓ 59 *	✓ 43	✓ 8	✓ 56	✓ 59 *	✓ 43	✓ 8	✓ 6.0	✓ 56

✓ 59 (Released 2018-03-13) *

● Borders do not display on table headers if `border-collapse` is set to `collapse` (bug 1727594).