

Discussion Board 3.1 – JavaScript: The Bad Parts

Eric McCool

Bellevue University School of Science and Technology

Web231-339A Enterprise JavaScript I

Professor Krasso

October 31, 2022

This week we look at the bad parts of JavaScript. To be fair, I have found bad parts in the few languages I have studied so far. So why should JavaScript be any different? Almost everything in life will have a con to its pro, right?

The first, and most glaring deficiency, turns out to be its security. JavaScript is used extensively in web browsers to present us with happy places to surf. Being that we can examine the source code in the browser apparently allows the more unscrupulous types to thwart its intentions and make life horrible for the rest of us. This leads developers to create 'defensive code' (Witt, n.d.), as if 'regular' coding can't be frustrating enough by itself at times.

Next, JavaScript struggles with global variables, as we saw last week. We are encouraged to use `let` and `const` to avoid this trouble. They can also cause some of the 'side effects' I just read about in chapter 4 when we use a global variable in a function (Flanagan, p. 71, 2020). This is too bad. I tended to use globals a lot when I was learning about C and Python. I will just have to adjust.

Then there is the number accuracy thing. In last week's reading about numbers, it turns out that $.3 - .2$ is NOT equal to $.1$, but $.2 - .1$ IS equal to $.1$ in JavaScript (Flanagan, p. 30, 2020). Go figure. JavaScript struggles with rounding floating-point numbers. This is confirmed in this week's link about the bad parts of JavaScript (Cyrille, 2018). Probably not a good thing if you are into data science or statistics.

One more, for good measure. When I first learned that JavaScript used 'automatic type-conversions' (Flanagan, p. 45, 2020), I thought this was a cool thing. Turns out, not so much. The struggles with this eventually led to the creation of TypeScript and FlowType (Cyrille, 2018).

While flipping through pages in a search about JavaScript flaws, I discovered that JavaScript was created in only 10 days! That is pretty amazing. But it might also explain why most critics complain of it being too complicated. Whatever the critics say, it looks like JavaScript is here to stay, at least until TypeScript or PyScript become more popular.

References

Cyrille, A., 2018. *"JavaScript: The Bad Parts and How to Avoid Them."* Published in SunStack.

Presented in Medium.com, 2022. <https://medium.com/sunstack/javascript-the-bad-parts-and-how-to-avoid-them-1a7c9bc5a0dd>

Flanagan, D., 2020. *"JavaScript: The Definitive Guide – Seventh Edition."* O'Reilly Media, Inc.,

2020. 1005 Gravenstein Hwy North, Sebastopol, CA 95472.

Witt, M., (n.d.). *"Why is JavaScript Considered a Bad Programming Language?"* Quora.com.

Quora, Inc. 2022. <https://www.quora.com/Why-is-JavaScript-considered-a-bad-programming-language>

Max Witt's page:

<https://www.quora.com/profile/Max-Witt-3>