

Why Just JavaScript Exists for You

Hello Cai,

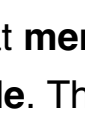
Most courses teach you to *write* code.

That’s obviously important.

But over the years, as I interviewed candidates and answered users on GitHub issues, one thing became clear to me.

So many problems are because we, even many years into programming, don’t *read* code correctly.

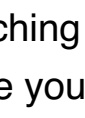
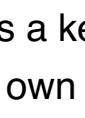
Programming education researchers said this for years. My method is not directly based on this research (I only learned about it a few months ago) but look at [@GregNN’s dissertation](#).

If we don’t teach *reading* code, can we expect people to write it well? 

I wasn’t familiar with the research so of course I had to take a long convoluted route to arrive at a similar conclusion. For me, it was a personal path of fear and confusion. It took me many years of writing JavaScript to feel like I’m *confident* in what a piece of code does.

For me, the key realization was that I read code differently than before—and differently from someone who keeps making the mistakes I used to make. It’s like I have an intuition, a memory of how each concept—variable, object, function—behaves exactly. A mental model!

I realized that **mental models play a huge role in how we read code**. They’re not necessarily visual (they’re more like intuitions—like you “know” how a door handle, a teapot, or a browser address bar works). Still, it’s convenient to express mental models visually for discussion.

To read a program correctly, in **our minds we need to simulate the computer**. But to simulate it correctly, **we need to have a correct intuition** about what each instruction does, and how each concept works. For example, take variables. Here’s two popular ways to “think” about them.  


Most of us probably don’t “imagine” things like this in our heads. (I actually can’t imagine something visual at all.) In our brains these intuitions are a lot more compressed. But they exist, and are often wrong. **A small mistake in a mental model causes many bugs over the years.**



So what’s unique about Just JavaScript? It places the focus on mental models. We revisit the concepts you already know and use every day—expressions, different types of values including null and undefined, objects, variables—and find mistakes that crept into your mental models.

To introduce our mental models, we use illustrations and animations (by Maggie Appleton). Our visualization isn’t the “only right way” to show JS, but we’ve made many intentional choices in it to avoid common misconceptions.

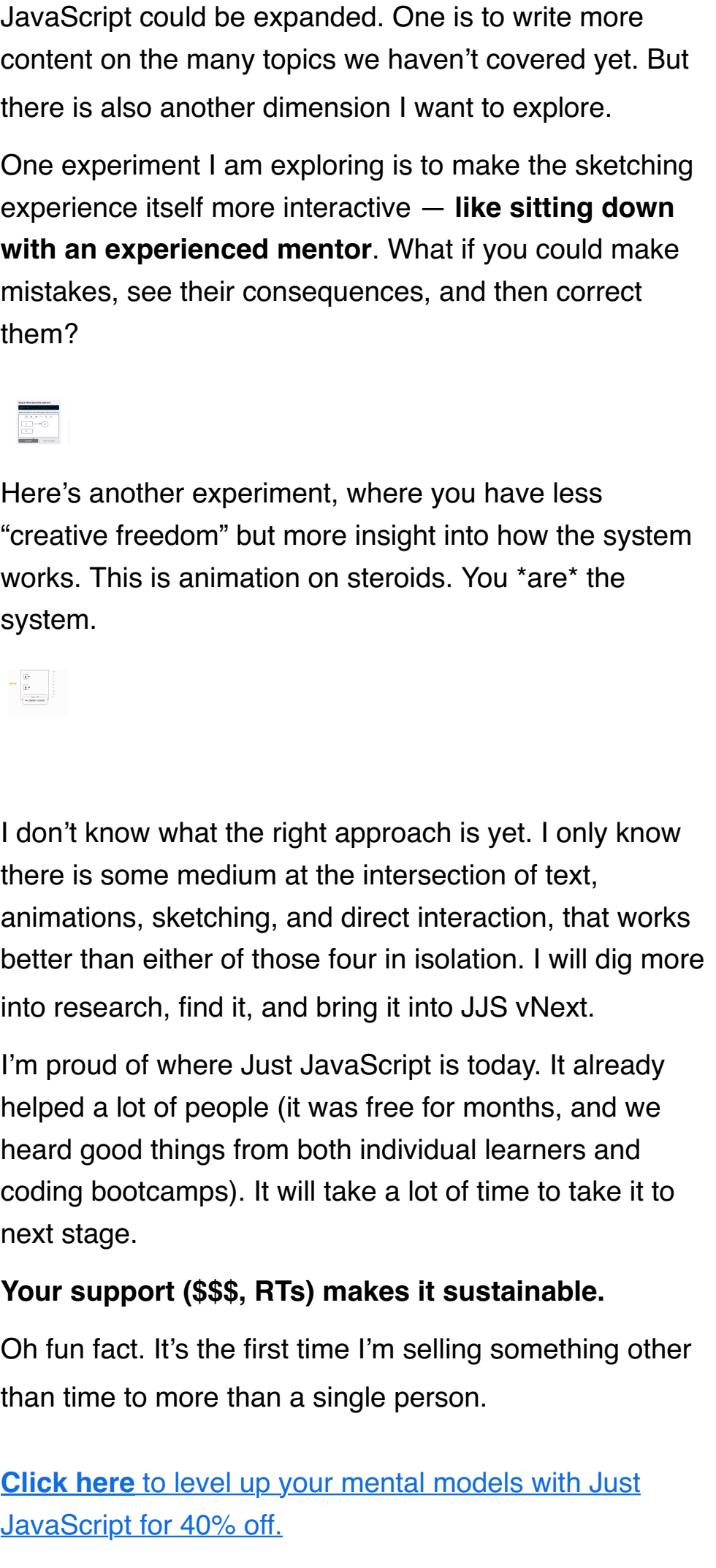



Just JavaScript is not a passive reading experience.

In key places in the text, we ask you to pause reading and *sketch*. Sketching is a key part of Just JavaScript. It’s a way to explore your own ideas about how the code works in your head—and find when they lead you astray. 

Key chapters end with interactive quizzes. Many of these quizzes involve sketching, too. If you have a misconception, it creeps into your sketches. This lets us provide you with feedback and explain *why* you get bugs, and how to think about code differently.  

Now, what’s up with this “universe” thing from the landing page? You see, that’s another way in which Just JavaScript is different. It is not a regular JS course in format or content, and it is also **not a regular JS course in style**. It places you *into* the imaginary JS universe.



Some readers hate the whimsical and quirky aspects of it, but I’m okay with that. (Feel free to get a refund!) Stylistically, it borrows from several genres. Of course, it’s a technical course. But some parts might remind you of a children’s book. Or an astronomy atlas. 

In my head, I kept getting back to texts that inspired me as a child and then as a teen. Alice in Wonderland, Little Prince, GEB, scriptures from different religions, popular science books and math puzzle journal columns. Quirky Systems. What is JS if not another one of those?


So here we are. The first version of the course is out, several years after we first started discussing it with Maggie Appleton. This is an exciting milestone. It doesn’t *feel* like a lot, and yet it took so much work from the whole team. Huge thanks to all.



If the project receives enough support, we hope to work more on it. There are two primary ways the scope of Just JavaScript could be expanded. One is to write more content on the many topics we haven’t covered yet. But there is also another dimension I want to explore.

One experiment I am exploring is to make the sketching experience itself more interactive — **like sitting down with an experienced mentor**. What if you could make mistakes, see their consequences, and then correct them?



Here’s another experiment, where you have less “creative freedom” but more insight into how the system works. This is animation on steroids. You *are* the system. 

I don’t know what the right approach is yet. I only know there is some medium at the intersection of text, animations, sketching, and direct interaction, that works better than either of those four in isolation. I will dig more into research, find it, and bring it into JJS vNext.

I’m proud of where Just JavaScript is today. It already helped a lot of people (it was free for months, and we heard good things from both individual learners and coding bootcamps). It will take a lot of time to take it to next stage.

Your support (\$\$\$, RTs) makes it sustainable.

Oh fun fact. It’s the first time I’m selling something other than time to more than a single person.

[Click here to level up your mental models with Just JavaScript for 40% off.](#)

This is a special release price that ends at the stroke of midnight (Pacific) August 1st, 2021.

Thanks,

Dan

[Unsubscribe from Just JavaScript emails](#) - [Unsubscribe from All Emails](#)

12333 Sowden Rd Ste B, PMB#97429, Houston, TX 77080-2059