<div class="tab">

**<h2>Disclaimer</h2>**

<p><i>*As a writer, it goes against my nature to share the rough notes that constitute this blog. But I've benefited so much from hearing (on podcasts) and reading (on Medium and elsewhere) about other beginners' coding journeys that I'd like to share my experience without getting hamstrung by my hopeless perfectionism.* *If you're interested, better samples of my work are housed on my* [*author page at Slate.com*](http://www.slate.com/authors.melissa_jayne_kinsey.html) *or in this* [*Dropbox folder*](https://www.dropbox.com/sh/1oaatbkdkrig2vo/AADXOLBiTxk1hHj8qXBNjxpma?dl=0)*.*</i><p></div>

**<h2>Day 1</h2>**

**<h3>Thursday, June 14, 2018</h3>**

<p>After just an hour or two of research, I settled on freeCodeCamp (FCC) as my learning platform. I also considered Coursera, Udemy, SoloLearn, Khan Academy, and Codecademy. But I love FCC's straightforward curriculum. You don't have to figure out whether Python is more advanced than C++, say, or whether Java is the same thing as Javascript (hint: it's not). You don't need to sort out the relationship between CSS and SASS and Flexbox and what have you. You just start at the beginning.</p>

<p>As you complete each step in FCC, that challenge is checked off and grayed out on your curriculum page. Instant gratification in a process that's otherwise tedious and protracted. Progressing through the steps quickly turns into a game in which you compete against yourself. It's like trying to reach the next level on Space Invaders, which is probably the last video game I played in earnest.</p>

<p>If you're wondering why I'm learning to code, see the landing page of this repository for an explanation. It boils down to this: I can't see writing about much of anything, let alone something as complex as drug development or aircraft manufacturing, without knowing something about the technology on which it's built. Scientists and engineers rely on high-integrity software and sophisticated algorithms to do cool things like synthesize epidemiologic data, analyze compounds, and develop avionics systems for unmanned aerial vehicles. I don't need to be able to build the software or execute the code in order to write about those things. But I do need to understand the backstory well enough to explain the relevant concepts to others.</p>

<p>I can't devote full days to coding because I need to earn a living. Today, though, I spent the whole morning on it. I made it through the entire Basic HTML and HTML5 portion of FCC's Responsive Web Design curriculum. In the afternoon, I squeezed in the first few steps in the CSS module, too. I've barely begun, but so far, so good.</p>

**<h2>Day 2</h2>**

**<h3>Friday, June 15</h3>**

<p>Not as much time to work today as yesterday, but I'm now halfway through the CSS segment. The website I built with Divi in WordPress (HowtoWriteanOnlineObit.com) offers optional CSS, and I've used it a time or two to customize my page. But I had no idea what I was doing. I just copied and pasted code from various tutorials and articles. With trial and error, I made it work. But it feels great to be able to read some code, even at a rudimentary level for the time being.</p>

<p>I've been a podcast junkie for 6 or 7 years, so naturally I found the [FCC podcast](https://freecodecamp.libsyn.com/) right away. I listen while I clean, exercise, cook, etc. So I've already heard several episodes. I find them really inspiring and encouraging. One episode pointed out that some people in tech don't like code camps because they open the door to anyone, thereby increasing industry competition. Competition eventually lowers compensation.</p>

<p>I'm sympathetic to that kind of gatekeeping. As a self-employed writer, I compete with new writers across the country — around the world, really. The difference between self-taught coders and self-proclaimed writers, though, is that anyone can call himself a writer on Day 1. He doesn't even have to attend freeWritingCamp, much less earn an M.A.

So developers with computer science (C.S.) degrees should take some comfort in knowing that code camp graduates have at least put in a few hundred hours of work to learn the basics. They can't just hang out a shingle. The rigor of any code camp, whether it's formal classroom-style training or an online platform, will deter anyone who's looking for an easy ride. Your mom might still do your laundry, but she can't do your coding for you.</p>

<Besides, Quincy Larson, the founder of freeCodeCamp, would probably say that opening the door to anyone is the whole point. Everyone should learn to code, but not everyone can be a C.S. major or pay thousands of dollars to attend a coding school. Sites like freeCodeCamp make it possible for anyone to learn, no matter their circumstances—strapped for cash, rural, disabled, or otherwise unable to tap into formal educational channels.

<p>I'll probably never call myself a coder or developer or programmer. At best, I might acquire some basic coding skills. But I'm hoping that those skills will help me better understand the science and technology content I write about.</p>

**<h2>Day 3</h2>**

**<h3>Saturday, June 16</h3>**

<p>Today I finished the CSS curriculum and made it halfway through the Applied Visual Design section. So far everything is pretty easy. My publishing background has been helpful, at least in these initial lessons. Many of the challenges familiar concepts like headings, typography, image sizing, and application of RGB, CMYK, or hex colors.</p>

<p>I'm sure, though, that this little head start won't ease my way for very long. As I continue to listen to podcasts, I'm encouraged and intimidated in equal measure. It's uplifting to hear the stories of all those who have taught themselves coding and programming, many while working and raising families. But I feel like a fly drowning in an alphabet soup of programming languages—or do I mean coding languages? Is there a difference? In fact, that's the part that scares me the most: not knowing what I don't know, but being certain that it's a lot.</p>

**<h2>Day 4</h2>**

**<h3>Sunday, June 17</h3>**

<p>Today was Father's Day, and family commitments kept me from coding. That's a choice, not an excuse. A younger person might disdain the idea of taking a day off from coding. But then again, visiting one's in-laws at a nursing home probably isn't how any tech bro or coder girl would choose to spend a day off.</p>

<p>Anyway, in lieu of a report on my coding progress, I have some thoughts on the journey. It reminds me of trying to learn German. After chipping away at the German language for 5 or 6 years, I can read it at a pretty respectable level but can barely hold a conversation. Had I realized how difficult German was before I began studying it, I'm pretty sure I never would have started. The grammar is logical but highly complex, and even if you learn the rules — the four grammatical cases and so on — it's really tough to apply them on the fly during a Tandem or Skype call, when you're dealing with background noise, slang, and colloquialisms. You get frustrated and start to feel stupid faster than you can say unzuzammenhängend.</p>

<p>I'm afraid I might be putting myself in exactly the same situation now. That is, I've arrived at Camp Greenhorn in the foothills of the Codeskill Mountains. There's a framed map at the trailhead and good signage along the way. Quincy Larson and his fellow park rangers have cleared all the low-hanging branches and debris—I needn't worry about snagging my new High Sierra insulated jacket on the underbrush.</p>

<p>As I move forward, every brook has a little wooden footbridge across it, and steep sections of the trail are equipped with concrete steps and sometimes even a little handrail. I feel good—I'm getting some exercise and a change of scenery. I'm breaking in my new HikeBoot Air footwear and looking forward to reaching Lake Tekuhjahbe. It's many miles away, but at this pace, I should be there by sundown, right? Or will I become less surefooted—begin tripping on tree roots, stumbling on rocks? Maybe even lose the trail and never make it to the summit?</p>

<p>OK, I'll drop the metaphor. But what I do know is this: if the analogy to learning German holds up, I'm glad I don't know what I don't know. If I'd given up on learning German, my life would have been much poorer for it. It's true I can't speak German any better than a first-year high school student. But the cliché that "life's about the journey" is true. In trying to learn the language, I've gone from knowing almost nothing about World War II history to knowing a great deal about it. By extension, I've learned about World War I history, post–World War II Cold War history, and German reunification. I've developed an interest in the German Expressionist art of Beckmann, Dix, Schiele, and others. I've watched many outstanding German movies and documentaries. Best of all, in my recent efforts to practice speaking German, I've begun to meet people from Berlin, Nürnberg, and elsewhere via the [Tandem](https://www.tandem.net/) app.</p>

<p>It's hard to guess what, exactly, I might learn from my coding journey other than…well, coding. Then again, I couldn't have imagined how trying to learn a language could teach me so much about art or history. So as I set off from base camp with a backpack full of little more than optimism, I'm hoping to be surprised.</p>

**<h2>Day 5</h2>**

**<h3>Monday, June 18</h3>**

<p>The challenges are getting a little tougher. Today I made it through only ten items in about two hours. Still ten challenges to go to finish the Applied Visual Design section. The content on hsl color and keyframes was new to me. The concepts aren't hard to grasp, but sometimes it's tough to figure out the syntax as I'm writing the code. My coffee consumption has reached alarming levels.</p>

**<h2>Day 6</h2>**

**<h3>Tuesday, June 19</h3>**

<p>I finished my regular work around 2 PM and got started on the ten remaining Applied Visual Design challenges. It took less than an hour to complete them. Reading each challenge out loud really helps me comprehend the instructions and focus on each task, so I'll continue to do that as I move forward.</p>

<p>The next section of the Responsive Web Design certification content is the Applied Accessibility section. This content turned out to be relatively straightforward for me. For years, I've prepared Section 508–compliant documents (that is, documents accessible by people with disabilities) for my medical publishing customers. I'm well versed in the Styles feature in Microsoft Word and use it often to create and apply custom style sets. FreeCodeCamp's accessibility content just requires dropping in a few HTML5 tags. I got through the first 10 accessibility challenges in 40 minutes—that felt good.</p>

<p>I've been taking much more frequent breaks during these challenges than I do when I write. Since I have the luxury of working from home, I can get up and water plants or unload the dishwasher or do some exercises or walk on the treadmill for 10 minutes. When I return to start the next group of challenges, I feel a lot fresher and more focused. Today was my nephew's 20th birthday, so I took a break to give him a call. Then I started back up and finished Applied Accessibility by 5:30 PM. Woo-hoo! That leaves 43 challenges and five projects to complete in order to receive the Responsive Web Design certification.</p>

<p>I just noticed that in order to get a certificate, you don't actually have to complete any of the dozens of challenges I've already finished. Certification is based only on being able to complete the final five projects. Maybe this was a mistake, but I just skipped ahead and looked at those projects. Designwise, they're a little hard on the eyes, but despite the simplicity of the results, I can't imagine being able to code them into existence. Phrases like "fork this CodePen pen" mean nothing to me right now. I'm guessing "pen" is meant in the sense of "playpen" or "sandbox," and "fork" might mean branching off of it somehow. Beyond that, I understand nothing. I'll look for a podcast on codepens. It's 6:00 PM, so I'm calling it a day.</p>

**<h2>Day 7</h2>**

**<h3>Wednesday, June 20</h3>**

<p>Today I completed one measly challenge and then got distracted when I discovered the freeCodeCamp Forum. I created an account and filled out a profile, which earned me a "biographer's badge." I also decided to post this blog and link it to my forum profile, so I took some time to get that set up on WordPress. I wasn't brave enough to build my blog from scratch. I figure I'll have enough of a challenge when I start building the five final projects.</p>

<p>I watched a few YouTube videos on Bootstrap, too. As best I can tell, it seems like a package of shortcuts that help you work more quickly in HTML/CSS, which admittedly seems pretty tedious. You're not supposed to have to use Bootstrap to build your final projects in freeCodeCamp, but I think I'll try it when I get to that point. Actually, since I found out it's not necessary to complete all the challenges in order to receive front-end developer certification, I'm tempted to skip ahead and get started on the capstone projects. But I have only 42 challenges left. Maybe I'll do the projects and these last few dozen challenges at the same time.</p>

<p>I also listened to a few back episodes of the [Data Skeptic podcast](https://dataskeptic.com/). (If you haven't heard the podcast before, check out Kyle and Linh Da's mini-episodes.)</p>

**<h2>Day 8</h2>**

**<h3>Thursday, June 21</h3>**

<p>I was once again distracted by getting a decent-looking blog posted. I'm trying to keep it simple, but I really enjoy this kind of stuff. It's easy to go down the rabbit hole looking for public-domain images and so on.</p>

<p>I rationalized the time away from code camp by changing some of the code in my blog. For example, in the WordPress code editor I deleted the footnote.php file and added some CSS to change the title font. Unfortunately, the customized title doesn't show up on my other devices. Most visitors, then, won't see it. That's because I chose Arca, a font I bought a while back. It won't display if it's not on the visitor's local device. As a workaround, I plan to create a logo that includes my title typography as part of the image. That may hurt the page's SEO, but there's a workaround: I can keep the title type, so that Google can read it when it crawls the site, but make the type transparent or white so it's invisible on the screen.</p>

**<h2>Day 9</h2>**

**<h3>Friday, June 22</h3>**

<p>Got right back to the coding challenges today. I got stuck on the first Responsive Web Design challenge, "Create a Media Query." I spent a long time fiddling with the code, googling it, and trying various "fixes" in Chrome Canary. No luck. Then I read some other campers' comments in the forum. They said this challenge is buggy in Chrome. Once I read that tip, I hopped over to Firefox. There it took me less than a minute to get the code to pass.</p>

<p>The remaining CSS Flexbox challenges took about an hour to complete. Likewise, the final section before the projects, CSS Grid, took an hour. I've arrived at the capstone projects. On Day 7, I wrote that I might skip to the projects and work on the remaining challenges at the same time. Now that won't be necessary.</p>

**<h2>Day 10</h2>**

**<h3>Saturday, June 23</h3>**

<p>Hmmm...No coding today. Just podcasts

**<h2>Day 11</h2>**

**<h3>Sunday, June 24</h3>**

<p>Unfortunately, fear and resulting procrastination coincided with the weekend, and I did very little coding during these two days. I worked a little on my WordPress blog. The header image I had chosen wouldn't fit into the container without stretching and zooming, so I tried adding or changing codes for "max-width" and so on in the WordPress editor. I also removed the footer.php file to get rid of the amateurish tag line "Proudly made with WordPress" (or whatever it says). I succeeded in mucking up the WordPress file to the point that I decided to scrap it entirely and change to a different theme.</p>

<p>The real issue here is my fear of moving on to the project challenges. It's one thing to follow the well-written instructions throughout the freeCodeCamp curriculum. It's another thing to jump into a sandbox and start building a castle. Since I don't plan to code professionally, part of me feels like this exercise is unnecessary. But I know that's a copout. If learning how to code is like learning a foreign language, then students must be able to speak as well as read the language. And building a project with code is the equivalent of speaking the language. I just worry that I'm not analytical enough or mathematical enough or fill-in-the-blank enough to build something with code.</p>

**<h2>Day 12</h2>**

**<h3>Monday, June 25</h3>**

<p>There's no point in blogging about my coding journey if I never finish my certificate because I can't do the projects. So before I waste even more time trying to make my blog look nice, I'd better just do the work.</p>

<p>I began the first coding challenge, which asks campers to build a tribute page. I chose to memorialize the Jews of 15b Caspar-David-Friedrich-Straße, whom Viktor Klemperer immortalized in his masterful war diary.</p>

<p>It made me a little angsty to step into that empty code pen, but I just took it one step at a time to build out the project. I learned almost immediately that although there might be right and wrong answers, there's no single way to arrive at them. It's kind of like writing: you can say the same thing a dozen different ways. None of them is wrong as long as the meaning is conveyed. But certain sentences are more elegant than others. And as with cluttered writing, you can throw in a lot of meaningless code and still produce an intelligible page.</p>

<p>The only difficulty I had in building this project came at the end. As I mentioned in an earlier post, I had no idea what forking was or how to accomplish it. I reread the instructions, which I suspect are deliberately obscure. Then I searched for instructions on Google and YouTube, as well as in the coding forum. Finally I decided to ask for help. An experienced camper stepped up in less than a minute (thank you, @nicknesh!). He even made a little screencast for me. Within 5 minutes, I had submitted my code. *Et voilà!* A green button appeared showing "10/10." Yippee! One down.</p>

**<h2>Day 13</h2>**

**<h3>Tuesday, June 26</h3>**

<p>I now know the meaning of "ugly code." Coding isn't as precise as I thought it would be. You can indent or break lines pretty much wherever you like. As I mentioned in my last post, you can litter the stack with all sorts of superfluous codes, and it will still work — at least that's been my experience so far.</p>

<p>I did discover, however, the "Tidy up code" button in codepen.io. Look for the dropdown arrow at the top right of the HTML section of the codepen. Click the arrow to spiff up your code automatically. Nifty feature for anal-retentive folks like me. You also have the option of analyzing your code for errors.</p>

<p>This second project required building a survey form. To lighten the mood a bit, I created a survey for the Association of Donut Lovers. (If said association doesn't exist, it should.) The survey uses a dropdown menu to let members weigh in on the question of glazed versus cake. Radio buttons ask "would you ever pass up a free doughnut?" The possible answers are "Yes," "Never," and "Only if stale." Finally, I used checkboxes to let ADL members select their preferred toppings.</p>

<p>I didn't spend much time on the layout of this survey, and I left the sections standing cheek by jowl. Maybe once I finish this course I'll go back and tinker with it. I'm just really eager to get at least one certification under my belt.</p>

<p>The first time I ran my code through the forker, it approved just 6 of 17 points (user stories??). I analyzed the code using the menu I mentioned above, made a few changes, and ran it again. Success! It passed all 17 points. Beginner's luck, no doubt.</p>

**<h2>Day 14</h2>**

**<h3>Wednesday, June 27</h3>**

Gotta keep the lights on—spent all day on work-work.

**<h2>Day 15</h2>**

**<h3>Thursday, June 28</h3>**

<p>I spent today, too, working on a paying job—some medical image research. Gotta keep the cashflow moving. But I worked late to finish so that I'll have most of tomorrow to work on project #3.</p>

**<h2>Day 16</h2>**

**<h3>Friday, June 29</h3>**

<p>I wish I had time to approach each of these capstone projects in a leisurely way and make each of them look really good. I know I'd learn more that way, and I'd be able to build a better portfolio. But I really need to finish at least this first certificate to add to my resume in order to land some tech writing gigs. I can always go back and spruce them up later.</p>

<p>For this third project, the task was to design a product landing page. I designed a mock page for a fictitious T-shirt design company. I found a cool illustration of a chicken man standing on a saucer labeled "STANDARD AUTHORITY," so I decided to use that as my logo and adopt that phrase as the name of my company.</p>

<p>As I've explained, my goal was simply to get this page to pass the tests so I could move on to the next challenge. I did, though, find a cool tutorial showing [how to create a navigation menu](https://medium.freecodecamp.org/how-to-easily-build-mashables-navigation-bar-with-html-and-css-9e5007af786) similar to the one used by Mashable. I followed the instructions and created a menu to match my logo. I also figured out how to add internal hyperlinks to the page so that each menu link would actually jump to a particular T-shirt shown farther down on the page. Neither of those extras — the cool menu or the working hyperlinks — was required for the challenge. So I guess I <i>*am</i>* playing around with the code a little.</p>

<p>My code passed 8/16 tests the first time I ran it. My navigation wasn't inside my header section. When I corrected that, the code passed all but one test. The forker reported that my navigation bar needed to be at the top of the page. It was already. I tried deleting my body and section tags so that the code would actually sit on the first line. That still didn't help. It appears to have been kind of a bug. But the Forum came to the rescue. Someone else had had the exact same problem before, and a helpful camper had offered a few lines of code to force the forker to recognize the nav bar. I copied and pasted the code, tested the code again, and this time it passed 16/16. Then I removed those extra lines of code to avoid any confusion later.</p>

<p>This project took the entire day, from 8 AM to 7 PM, with only a few bathroom breaks – no lunch break and no goofing off. Coding is painstaking work. At this point, I'm not really sure why or even if I like doing it. It's definitely a lot like solving a puzzle, and puzzles have always seemed pointless to me. I guess the big difference is that coding does have a point. It actually does produce something. I definitely feel a little adrenaline rush each time something shows up or works as intended — like when I added the little jump links to the menu and got them to go where I wanted them to. Speaking of, it's time to get started on my next fix: project #4.</p>

**<h2>Day 17</h2>**

**<h3>Saturday, June 30</h3>**

<p>I listened to an interview with Quincy Larson, founder of freeCodeCamp. I enjoyed it so much that I searched for other podcasts on which Quincy had been interviewed. (Somehow I think he'd want to be on a first-name basis with his campers.) He gives two standard pieces of advice, neither of which I've followed thus far: one is to code every day, so that you make continual progress and don't forget what you've learned. The other is to hang out with other coders</p>

<p>I've coded most days since I started, so I'll try to be more consistent. As for hanging out at coding Meetups and hackathons and such, that'll be way outside my comfort zone. I'm beyond wallflower status, having crossed over into hermithood long ago. On the other hand, I'm eager to share my enthusiasm about coding and curious whether anyone else my age attends these things. There's a machine learning Meetup tomorrow night. I'll try to step out of my bubble to attend.</p>

**<h2>Day 18</h2>**

**<h3>Sunday, July 1</h3>**

<p>Today was a fun day here at code camp. I started project #4, "Build a Technical Documentation Page." I decided to create a freeCodeCamp glossary. Technically, a glossary might not be technical documentation, but the content of the page is the point of the exercise. The idea is to build out the functionality. Anyway, for the first time I felt some confidence going into it.</p>

<p>I'm still eager to pin down my first certificate, but with this project, I really started to play around with the code. Although I'm just scratching the surface, I've begun to understand that the code is powerful enough to build anything you like on screen, and that the coder is in control of the code. That's not a revelation, I suppose. But up to now I've felt as if I were at the mercy of the code—like it was my job to figure out what the code wanted me to do to make it work. Now I'm beginning to feel like it's the other way around. Yes, I have to follow a particular syntax. But beyond that, I have a lot of creative freedom. It's a lot like writing in that sense: You have to follow the rules of punctuation and grammar, but what you create with your words is up to you.</p>

<p>Also today I discovered Google fonts. (I'm still kicking myself for having paid as much as sixty bucks for a set of fonts.) I chose Raleway and read a couple of blog posts explaining how to use the fonts. Then I entered the HTML and CSS into my codepen. At first I didn't have the right syntax. But I played around with it a bit, and within a few minutes the right font appeared on screen. How exciting is that?? </p>

<p>Then I decided to make each section a different color. The color here is gratuitous, but it's pretty. I like to look at it. So there. I looked up hex colors and chose a muted sunset palette. That was all I had time for. Tomorrow I'll finish building out the page.</p>

<p>As I worked around the house, I listened to a few episodes of one of my new favorite podcasts: [Learn to Code with Me](https://learntocodewith.me/podcast/). The podcast host, Laurence Bradford, is really unpretentious and is a great interviewer. One episode featured Quincy Larson, founder of freeCodeCamp, who introduced himself on the podcast as "a teacher at freeCodeCamp." If only more people of his caliber were that modest…(I'm talking to you, Elon Musk!)</p>

**<h2>Day 19</h2>**

**<h3>Monday, July 2</h3>**

<p>Today I finished project #4. Objectively, it kinda sucks. If I saw a web page that looked like this, I'd think it was totally lame. Nevertheless, building it from scratch gave me a huge sense of accomplishment. Again I did a few little extras. In my last post, I mentioned the color and fonts. But I also used internal hyperlinks and changed their color to complement the page. I added a thin border to my navigation menu and placed a footnote to credit my sources.</p>

**<h2>Day 20</h2>**

**<h3>Tuesday, July 3</h3>**

<p>Work intervened today, and I wasn't able to code. I did listen to a few podcasts and read some pieces on Medium and elsewhere.</p>

**<h2>Day 21</h2>**

**<h3>Wednesday, July 4</h3>**

<p>Once again, no coding happened around here. Family and work obligations intruded. In theory, I get why new coders are advised to code every day. You need to keep things fresh. More important, you need to make incremental progress. It's just like studying German. <i>Deutschsprache</i> wasn't conquered in a day.</p>

<p>On the other hand, it's important to be realistic. I don't think I should feel guilty about neglecting coding in favor of, say, earning a living. I can't ignore paying clients in order to spend time in my codepen. Also, my husband and I both have elderly parents in the area who require attention. But I'm super motivated, so I'm in no danger (at least at present) of losing interest or becoming discouraged if I don't code every day. </p>

**<h2>Day 22</h2>**

**<h3>Thursday, July 5</h3>**

<p>Spent the day on my regular work. Listened to more podcasts while cleaning and exercising.</p>

**<h2>Day 23</h2>**

**<h3>Friday, July 6</h3>**

<p>The Fourth of July holiday fell on a Wednesday this year, so almost everyone has been off work for a 5-day weekend. I had work to do from morning through afternoon and then was away for a few hours.</p>

<p>Whenever an article is mentioned on a podcast, I look it up online and bookmark it for later reading. In *The Economist,* I read "What Is Code?" It really clarified for me why there are different languages, such as JavaScript, Python, and C++. Essentially, if you think of computer instructions as a stack, most of us operate at the very top — the part we can see. The lower you go in the stack, the closer you get to the binary code that actually executes commands. The various coding languages are just intermediaries we use to communicate with the lower end of the stack. Coding languages are like cars: whether you drive Mustang or an Avalon or a Dart, you'll arrive at the same destination. Some cars have a little more style and cleaner lines. Some are faster. Some have a lot of aftermarket parts tacked on to them. But all languages take you to the same destination: the bottom of the stack, where the rubber meets the road.</p>

**<h2>Day 24</h2>**

**<h3>Saturday, July 7</h3>**

<p>Today we drove 2 hours south to the small town where my mother-in-law lives in a long-term care facility. This is a regular thing for us. We pick up my father-in-law, take them both to lunch (not easy, since my mother-in-law can't walk), take my father-in-law back to his house, and then drive 2 hours back home. So no coding today.</p>

<p>I've continued binge-listening to the first four seasons of the "Learn to Code with Me" podcast. I like Laurence Bradford's quirky style. In almost every episode, she says, "I'm probably jumping ahead here, but..." The five-part question is her signature—her curiosity and excitement bubble up and finally spill over as a series of questions, like soda fizzing over the rim of glass. Then she listens to the guest's answer and gives one of her habitual replies, such as "Got it. Thank you for sharing that."</p>

**<h2>Day 25</h2>**

**<h3>Sunday, July 8</h3>**

<p>I've developed e-learning products ever since there were e-learning products to develop. I've written or edited content in various learning management systems (LMSs). I've helped developers implement new features, such as interactive self-check activities. I've also worked with designers to determine the look and feel of an online course or subscription database product (for example, a point-of-care reference tool for doctors or nurses).</p>

<p>But in publishing, there's always more to learn. When I discovered Google Fonts the other day, I made a mental note to go back later and check out [material.io](https://material.io/). It's a fantastic site for information about web design. Best of all, material.io supplies open-source code for all its system icons, navigational elements, surfaces, and so on. You can even pick up code to make your icons move a certain way.</p>

<p>Today I read quite a bit on the site. I knew that icons should be consistent, of course, but material.io explains exactly how to accomplish that. For example, stroke weight should always be two pixels. I learned some new terminology, too, such as "terminal stroke" and "modal navigation drawer." I can't wait to apply some of these principles to my freeCodeCamp projects! </p>

**<h2>Day 26</h2>**

**<h3>Monday, July 9</h3>**

<p>Today I worked on project #5 in the hope of finishing it, but at the end of the day I wasn't quite there. I got a little hung up on trying to make everything look really nice. On [codepen.io](https://codepen.io/mjkinsey/pen/ZMLGQa) you can see how many views your projects have had. Since I realized that projects are searchable and anyone can look at them, I paid attention to the look and feel of my portfolio. Unfortunately, my skills are still rudimentary.</p>

<p>I can't say I put much time into aesthetics on the earlier projects. But I definitely departed from my original plan, which was to finish the five capstone projects with the sole purpose of getting them to pass. The more podcasts I listened to and the more I read on Medium and elsewhere, the more I realized that potential employers would be looking at my project samples, not pulling my certificate.</p>

**<h2>Day 27</h2>**

**<h3>Tuesday, July 10</h3>**

<p>This journal entry should have a big gold star next to it because this is the day I finished project #5 and received my Responsive Web Design developer certification. I'm inordinately proud of this accomplishment, and I even posted the certificate on my Facebook page. Maybe I'll tweet it out later.</p>

<p>This evening I went to my first coding Meetup. The topic was Agile product development. There were about 10 people there, and I met and spoke with almost all of them. The tech community has a reputation for welcoming new people, and it seems that reputation is well deserved. My fellow attendees did not disappoint in terms of friendliness and openness.</p>

<p>As it turns out, Agile product development seems to have about a 90% overlap with the work I've been doing for the past 10 years — that is, shepherding online courses from conception and content creation to beta testing and launch. In publishing, this role used to be called developmental editing. The developmental editor, or DE, would work with the author to develop the content. Then she (nearly always a woman) woud see the manuscript through the production process. In recent years the DE has been called a product owner. I don't know if product owners in publishing follow the Agile process specifically, but the system is at least extremely similar. Anyway, the presentation and Q&A session were interesting to me because the steps were familiar.</p>

**<h2>Day 28</h2>**

**<h3>Wednesday, July 11</h3>**

<p>Today I attended my second tech Meetup. This one was for full-stack developers, and the topic was HTTP. The acronyms were flying — TCP, IP, DDoS, CDN, RFC, etc. In contrast to last night's Meetup, I understood about 10%, and the other 90% went over my head. I did, however, sit next to an interesting guy named Donald — perhaps nearing age 70 — who was eager to help me make some contacts. Right after the meeting, he followed up with links to the profiles of several people he had mentioned. He also told me about some upcoming Meetups that I plan to attend.</p>

<p>Also exciting was my discovery that St. Louis has a freeCodeCamp Meetup with nearly 900 members. I'm super excited to meet other campers in the area! I'm also curious how old they are. I've read that the average age of a person learning to code is about 30. At the Meetups I've attended, I've seen quite a few people older than that. Nice to know I'm not alone. Donald mentioned that he tries not to get hung up on his age, and that seems like good advice.</p>

**<h2>Day 29</h2>**

**<h3>Thursday, July 12</h3>**

<p>Today I signed up to donate five bucks a month to freeCodeCamp even though I'm beyond broke. I also tweeted out my Responsive Web Design certificate. I thanked freeCodeCamp and expressed support for its mission to provide accessible online education for everyone. The donation must have brought me some good karma, because Quincy Larson himself retweeted me! That was followed, of course, by many likes (around 50 so far) and retweets.</p>

<p>I also started the JavaScript course. Turns out JavaScript is way, way, way harder than HTML/CSS. There *is* math involved. I'm spooked but determined. My focus was very spotty today, and I got through only 11 challenges in the Intro section. Hope to do better tomorrow.</p>

**<h2>Day 30</h2>**

**<h3>Friday, July 13</h3>**

<p>Today I completed 23 challenges in the beginning JavaScript section. They weren't easy, but I found that if I re-read the instructions and explanation several times out loud, I could get it.</p>

**<h2>Day 31</h2>**

**<h3>Saturday, July 14</h3>**

<p>I didn't work on any freeCodeCamp challenges today. Instead I listened to the first five episodes of the [CodeView podcast](https://itunes.apple.com/us/podcast/codeview/id1380293751?mt=2), in which one beginner and two veterans talk about coding. I also decided to read the actual [JavaScript documentation](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics). It's really helpful. It has a conversational tone, and there are exercises that allow you to create some simple code and follow along.</p>

<p>To be honest, I didn't build all the little projects because wanted to read the documentation on my iPad or phone, rather than on my iMac. I sit at my computer 10–11 hours a day as it is. I don't need to spend the entire weekend in my office, too. I figure that reading the documentation is already lagniappe for my coding education even if I don't build out the projects (e.g., a "Guess the number" game).</p>

**<h2>Day 32</h2>**

**<h3>Sunday, July 15</h3>**

<p>I did sit at my iMac for a while today, working through 13 challenges. I found myself looking up a lot of stuff, such as string concatenation, return statements, and "passed in" values. Getting the backstory, so to speak, takes time, but I'm building up my coding knowledge bit by bit.</p>

**<h2>Day 33</h2>**

**<h3>Monday, July 16</h3>**

<p>I have to knock off early to go to the dentist (yippee!). But I nevertheless read another section of the JavaScript manual and got through 21 challenges. That leaves 39 to go in this first, depressingly long section of the JavaScript certificate.</p>

**<h2>Day 34</h2>**

**<h3>Tuesday, July 17</h3>**

<p>Donald Trump's soon-to-be-infamous Helsinki Humiliation Summit took place yesterday, and it was a watershed moment. Journalists' heads are exploding, and citizens around the country are aghast at the whole sycophantic spectacle of it. So there were lots of interruptions today.</p>

<p>I really needed to focus, too, because today's coding challenges stumped me. I worked on switch statements, and I'm just not getting it. Some of the language used in JavaScript is frustrating. The worst offender so far is use of the word "un-shift" to mean adding one item to the front of an array. Even "shift" would have been confusing, since the operation doesn't transfer or move anything. But it still would've been better than "un-shift," which suggests that something has already been transferred or moved and is now being removed.</p>

<p>That brings us to use of the word "switch." Apparently it means evaluating or comparing an expression against others and executing a snippet of code if the expressions match. Yet nothing is being switched around or transposed or replaced. Maybe I could think of this function like a lightswitch that turns on a bit of code when it finds matching expressions.</p>

<p>In any case, I found the syntax of these challenges really illogical, inconsistent, and frustrating. There also doesn't seem to be any pattern to follow in determining whether to add or omit spaces. Some expressions are all closed up (no spaces), some include space on one side, and some require space on both sides.</p>

<p>I suppose this is just something you need to remember for each function and operation, the way you have to remember whether German nouns are feminine, masculine, or neuter. There are some patterns in German that can tip you off—for instance, *–ung* words are (usually? always?) feminine. But then, for no reason, we have *der Rock* (masculine) for "skirt" and *das Mädchen* (neuter) for "girl." Argh.</p>

**<h2>Day 35</h2>**

**<h3>Wednesday, July 18</h3>**

<p>The political furor continues. Who can look away? Most of my day was consumed by attending an American Medical Writers Association (AMWA) lunch meeting. Not a tech meetup, but I did evangelize about freeCodeCamp, as I do pretty much everywhere I go now. I'm sure it's annoying to people. Too bad.</p>

<p>I was going to attend an Angular Meetup this evening, but that's just too much human contact for one day. After I decompressed from all the networking, I got back to my freeCodeCamp challenges.</p>

<p>Counting cards… Counting cards…Holy mother of God. This must be the point at which gung-ho would-be coders like me start thinking they're too stupid to do this. If I have a criticism of the freeCodeCamp curriculum, it would be this: Certain challenges are so easy, you could teach them to a 10-year-old in two minutes. Sometimes there are five challenges in a row like that. And then… *WHAM!* You're slammed with an exercise like this.</p>

<p>I get that these megachallenges are intended to incorporate the little challenges that precede it, but sometimes they require a finicky syntax unlike that used in any previous challenge (e.g. Return count + " Bet", with a space before "Bet," which represents the space you'd want to appear on-screen between the card count and the recommendation to the player to place a bet). Sweet baby Jesus. I have 25 challenges left to go.</p>

**<h2>Day 36</h2>**

**<h3>Thursday, July 19</h3>**

<p>Wow. Discouraged. Had trouble understanding the whole concept of testing object properties. I wish the examples were actual things instead of "myObj" or "testObj." Ended up looking up most of the solutions, after hacking at them on my own a long time. Still 19 challenges to go in the BASIC JavaScript section. My progress could be described as plodding, at best. Can I really build something with JavaScript if I can't even do the challenges on my own?? </p>

**<h2>Day 37</h2>**

**<h3>Friday, July 20</h3>**

<p>No coding—tight deadline in my regular work.</p>

**<h2>Day 38</h2>**

**<h3>Saturday, July 21</h3>**

<p>Took the day off from coding to do some cleaning.</p>

**<h2>Day 39</h2>**

**<h3>Sunday, July 22</h3>**

<p>Worked on the project that’s on deadline.</p>

**<h2>Day 40</h2>**

**<h3>Monday, July 23</h3>**

<p>Worked all day—no coding</p>

**<h2>Day 41</h2>**

**<h3>Tuesday, July 24</h3>**

<p> Worked all day—no coding. I did, though, discover a great new podcast called [Developer Tea](https://developertea.simplecast.fm/). The host, Jonathan Cutrell, is as interested in developers' mental health as he is in programming itself. For instance, he has a good episode on the topic of staying focused. He points out that it's more about what you don't focus on than what you do.</p>

**<h2>Day 42</h2>**

**<h3>Wednesday, July 25</h3>**

<p> Worked all day—no coding. I’ve been meaning to set up a GitHub account, and today I discovered that I already had one. I created it back in April 2016, when I downloaded a Google opt-out plug-in. So I got back into my account, updated my password, created a new username, and filled out my profile.</p>

<p>The GitHub platform is pretty unfamiliar to me, so I did some reading up on version control, pushing content, cloning, committing, and so on. There's a whole lot of trial and error. For me, that's the best way to learn any new platform. I read a little, play around a bit, hit another snag, read some more, and try again. And again...</p>

**<h2>Day 43</h2>**

**<h3>Thursday, July 26</h3>**

<p>This evening I attended a recruiting event called WomenHack. No one was much interested in hiring me as a tech writer, since employers were there mainly to recruit developers. But I met some great women. I noticed that they weren't super-young, either. One brainy-looking woman, who'd been a programmer for decades, looked to be in her 60s. Most of the rest of the group were in their 30s or 40s.</p>

**<h2>Day 44</h2>**

**<h3>Friday, July 27</h3>**

<p>Today I heard back again from an instructional designer I'd contacted on LinkedIn. He gave me lots of great advice about revising existing documentation and contributing to freeCodeCamp. He also encouraged me to sign up for GitHub. As mentioned in my July 25 post, I'd already refreshed my GitHub account. So instead of replying on LinkedIn right away, I spent the day trying to get this blog posted on a decent-looking GitHub page. Turns out that's more than a 1-day job for someone who's just learning to code. I ended up thanking my contact and telling him I'd send him my GitHub link when it's ready for prime time.</p>

<p>In trying to get a page up quickly, though, I did a lot of forking, cloning, and—wait for it—coding! It was just CSS, but it was a chance to apply what I've learned in freeCodeCamp. It was fun. Looking forward to playing with it some more.</p>

**<h2>Days 45 to 47</h2>**

**<h3>Saturday, Sunday, and Monday, July 28, 29, and 30</h3>**

<p>Disastrous three days working on my GitHub. I decided forking wasn't the way to go, since it wouldn’t showcase my own work, so I tried to create my own page. I ended up committing something like 400 times. Decided that wouldn't look good, so I deleted that GitHub repository and tried to replace it. You can imagine what kinds of problems ensued. I finally reached out online and got help from some experienced people. They tried to help me conceptualize GitHub and git. I think I get it, but I still have no GitHub up and running.</p>

**<h2>Day 48</h2>**

**<h3>Tuesday, July 31</h3>**

<p>Regular work—no coding.</p>

**<h2>Day 49</h2>**

**<h3>Wednesday, Aug 1</h3>**

<p>No coding today, but I went to a wonderful presentation on cybersecurity at the STL Cyber meetup. The woman who spoke, Kristi Cook, works as a cybersecurity specialist for Peabody Coal. She also serves in the National Guard. Her educational background is in military history, and she says she knew nothing about technology when her boss in the Guard first assigned her to work in as a cyberthreat analyst.</p>

**<h2>Days 50 to 55</h2>**

**<h3>Thursday, Friday, Saturday, Sunday, Monday and Tuesday, Aug 2 through 7</h3>**

<p>No coding during this period. Inspired by Kristi Cook’s presentation on Cold War espionage, I worked on a [Slate story on meddling](https://slate.com/technology/2018/08/meddling-is-the-worst-word-to-use-to-describe-russias-election-interference.html).</p>

**<h2>Day 56</h2>**

**<h3>Wednesday, Aug 8</h3>**

<p>Finally back at it! I started the day by re-reading the posts from people who had helped me with my GitHub questions. One person had sent me a link to an article on the command line. I re-read it and then googled some more articles on that topic. I learned, for example, the rm command and how to use it carefully (with -i, which prompts the user to confirm the deletion).</p>

**<h2>Day 57</h2>**

**<h3>Thursday, Aug 9</h3>**

<p>I spent another few hours plowing through demos and working on my GitHub. I went through a couple of demos on YouTube. I By the end of the day, I had a decent-looking blog template up on GitHub. Now I just need to format these posts and import them from Word.</p>

**<h2>Day 58</h2>**

**<h3>Friday, Aug 10</h3>**

<p>Today I continued to work on my GitHub. I read a lot about floats [here](https://medium.freecodecamp.org/css-floats-explained-by-riding-an-escalator-57fa55232333) (great visualization recommended by a developer who responded to my post) and on this [terrific CSS blog](https://css-tricks.com/all-about-floats/). Yet I still can't make my right column content float to the top right. I tried all of the following: </p>

<BL>

* Adding "float: clear" at the end of the left column content
* Wrapping the right column in a separate section
* Adding "float: right" to the column specs, the profilecard class specs, the <div> specs, and section specs
* Adding "!important" after every "float: right."
* Changing the max widths of the left- and right-column elements
* Reducing the size of the profile picture
* Setting the profilecard to "display: block"
* Setting the profilecard to "align: right"
* Setting the profilecard to "position: absolute"
* Removing "row" html</BL>

<p>I spent hours and used about 35 commits, which is sloppy, but nothing worked. I've asked the FCC community for help.</p>

<p>Then I moved on to one of the conceptualization exercises recommended by one of the developers who responded to my earlier topic (see days 45 to 47). No problem understanding commits and merges, but rebasing?? I don't get it. Still, this is a [nifty interactive exercise](https://learngitbranching.js.org/) if you're struggling with these concepts.</p>

**<h2>Days 59–76</h2>**

**<h3>Saturday, Aug 11–Tuesday, Aug 28</h3>**

<p>During the first week of this period, I continued to wrestle with this GitHub page. Let's just say it wasn't pretty. As I mentioned in my last post, the most difficult part was getting the elements to appear where I wanted them to on the page. Instead of appearing in the upper-right corner, my profile box was showing up at the bottom of the page. I understood the concept of floats behaving like elevators. Nevertheless I couldn't get my code to do what I wanted it to. But with the help of the FCC forum, I was finally able to get it to work!</p>

<p>The next couple of weeks were spent...procrastinating. It's true I was working, and I did go on a short road trip, but primarily I just had cold feet about getting back into JavaScript.</p>

**<h2>Day 78</h2>**

**<h3>Wednesday, Aug 29</h3>**

<p>I don't want to delay my progress on JavaScript any longer, so today I returned to freeCodeCamp and discovered I'd forgotten a lot of what I learned during the first 20 or 30 challenges.</p>

<p>To get back up to speed, I read the online documentation at Mozilla and went back through many of the freeCodeCamp challenges. But I still wanted more backstory. What is JavaScript doing? Why do we need it? What can it accomplish that we can't achieve using HTML and CSS? Can JavaScript actually save me time or make things easier for me once I learn it?</p>

<p>I searched Amazon for a book on the topic. Ultimately, I purchased *Javascript Absolute Beginner's Guide,* by Kirupa Chinnathambi. Normally I check out books from the library, rather than buy them, but I figure I'll need to keep this one handy for a while.</p>

<p>A little digression: In browsing Amazon, I noticed that one JavaScript instruction manual had 421 reviews. As the author of a self-published book with maybe 5 or 6 reviews, I'll go out on a limb and say that unless you're a J.K. Rowling or a Stephen King, 400+ people won't step up to review your book without an incentive. And incentivized reviews are usually positive. So buyers should take those 4.5 stars with a grain of salt.</p>