

**Learning Fast and Slow - How Mentorship Accelerates Mastery**

This past March break, I was fortunate enough to be a mentor to a group of girls taking their first steps into the world of code and design. Hosted at Freshbooks, [Ladies Learning Code](http://ladieslearningcode.com/) became [Girls Learning Code](http://ladieslearningcode.com/program/girls-learning-code/) for the week, fostering an inclusive and collaborative environment in which the girls could build their skills and create their own websites. By applying their newfound skills directly to projects, the girls came away from the week with a new perspective on the power of a career in tech.

As a mentor, I also came away from the event with a new perspective — and it’s changed the way I’ve thought about skills development, time management, and training junior developers.

In those few short days, I learned as much about teaching and working with youth as the girls did about coding, design, and the world of STEM. The lessons I learned— which I’m sharing below— will inform how I teach, mentor, and work with youth and junior developers going forward.

**Lesson 1: Do Less, Get More**

Creating a website in under five days, with no coding knowledge at all, is a lot to ask of anyone. In that short time, the girls designed logos and buttons for their websites; learned HTML and discovered [Mozilla’s X-Ray Goggles](https://webmaker.org/goggles); learned CSS; did market research; made infographics with Canva, and learned about crowdfunding; went on a field trip to Nuvango to see their t-shirts and to get a sense of the production process; visited Ubisoft for a women in tech panel; and gave a final presentation of the websites they built to a large crowd of their parents, mentors, and Freshbooks employees.

If that sounds like a lot, it was. Even as a mentor who (in theory) knew the majority of what was being taught, I was surprised by the pace of it all.

My favourite aspect of STEM professions is the mix of artistry and engineering. I was excited to open the girls’ eyes to that complex balance and show how science and art can combine to solve difficult problems. But as I quickly found out, young learners have short attention spans. Showcasing how far-reaching design and code can be takes a good deal of patience, and abstract concepts and theory do little to spark enthusiasm.

When first learning to code, time to practice is key. It’s important that the kids engage with the practice as soon as possible, so that they start thinking, “I can do that.” Teaching fewer concepts and giving more time to practice and experiment helps forge a connection to what’s being taught.

The same can be applied when training junior developers. More time to practice. More hands on coding - and quick iterative feedback can really help spark enthusiasm, drive engagement and encourage junior developers to learn at a fast pace.

**Lesson 2: They Get Better, I Get Better**

Like anything else, teaching requires practice and patience. I discovered first-hand that this is especially true when teaching children, teenagers, and probably developers of any age. There will be times when we get it wrong, and I definitely did (the look of withering disappointment that one of the girls gave me when I asked her who [Tyler Oakley](http://upload.wikimedia.org/wikipedia/commons/3/3c/Tyler_Oakley_by_Gage_Skidmore.jpg) was will stay with me for a while.)

But just as the girls have to practice the code they’re learning, we as mentors, teachers and developers have to practice and iterate on how we teach, and on how we present ourselves to those who we want to inspire.

It’s not enough to know your code. Teaching it — and teaching it well — requires more than pure knowledge. Instructing youth is not an inborn skill for most of us, but practice and preparation can improve our odds of success quite a bit— and help elevate both parties’ skills in the process.

**Lesson 3: Your Passion Inspires Their Passion**

Holding kids’ attention is tougher than it seems. Convincing them that typefaces and semicolon placement are worth their attention — as opposed to Tyler Oakley, for example — is a particularly tall order. But it can be done!

By the end of the week, it was clear the girls responded best to passionate presenters who genuinely cared about the subject they were presenting. A well-structured presentation and a great knowledge base is only half the battle; it has to be delivered with infectious enthusiasm.

No matter how interesting a subject may be, if we as teachers and developers aren’t excited about it, what hope do we have of sparking interest in someone else? By showcasing our passion, we can inspire others and that goes a long way to making semicolons more enticing.

**Lesson 4: It’s Totally Worth It**

Passing our craft on to someone else—be it design, development, or something else altogether—is a humbling exercise.

It forces us to break our craft down into its fundamental pieces and make sense of it all over again. We could all use a return to the basics every now and then— it reminds us why we fell in love with code in the first place.

Seeing the flicker of recognition in their eyes and knowing that I was responsible for it was enormously satisfying. We all know what those epiphanies mean: an acknowledgment of a problem solved, a puzzle-piece clicking into place. They’re a large part of what keeps us designing and coding. And they come from having the time to experiment, to ask questions of both our teachers and the material, to draw connections and to express our passions through code.

In the end, I believe that we should teach what we’re most passionate about—no matter how specialized the concept—and then take a step back to allow the students (be they kids, designers, or junior developers) to grasp it themselves.

Some won’t be interested, and that’s okay. And if we overreach or miss the mark, that’s okay too. As long as we focus on enabling youth to experiment and discover in-depth the concepts we choose to show them, they will understand the breadth of the industry. And maybe they will begin to see a place for themselves in this wonderful world we inhabit that balances soft skills and quantifiable technique, art and science, creativity and logic.

*Thanks very much to Ladies Learning Code for allowing me to mentor, to Freshbooks for use of their offices (and for the caffeine!), to* [*Myplanet*](http://www.myplanet.com) *for allowing me to take three days off, and to all the girls who made it.*