



# Always Iterating: How Launch Academy Updates Its Coding Curriculum

by Sarah Pascarella

September 14, 2017

Like

Share 0

At Launch Academy, our curriculum is never static: Because tech is always changing, our curriculum needs to evolve along with the coding languages most desired in the workforce. I recently sat down with several members of the Launch Academy curriculum team to get the inside scoop on how and when Launch Academy adapts its coding bootcamp curriculum.

**Q: Tell me about Launch Academy’s curriculum revision process and how you incorporate student feedback.**

**Ezra Skolnik, Experience Engineer:** During a cohort, the staff will slowly make updates to the curriculum to edit things that are in need of polish. There will typically only be a focus on the next week of curriculum, so the Experience Engineers (EEs) can focus on what they’re currently teaching the students. Between cohorts, we have a week of curriculum development where we make larger revisions—potentially reordering materials, adding and removing weeks—and take a more holistic view of the entire curriculum and any changes we need to make.

LAUNCH

(<https://launchacademy.com/blog/always-iterating-how-launch-academy-updates-its-coding-curriculum/>)

**Joey Huynh, Senior Experience Engineer:** We think about [curriculum updates] on two levels: the minor changes of quality control, including editing and tweaking, and the major changes that are much more architectural in nature. Architectural changes also typically happen in two ways: One is at the content level—like the increased focus on integrating JavaScript into introductory materials and introduction of the React framework. Two, there's the sequence of it. Should we have this topic happen earlier or later [in a cohort]?

**Eben Lowe, Experience Engineer:** Regarding minor changes, we have a smaller, shorter feedback cycle [from students and staff] where we update individual assignments. We have a comment system on every lesson, and every week we'll go through, look at comments that are new, and say agree or disagree. If we agree, we make changes. For major changes, I think of them much more in terms of what we teach, and those [updates] are based on market demands and our own expertise.

**Huynh:** We have ratings on lessons, so students can leave ratings and comments. We as staff may also leave comments on a given lesson. If we can't fix issues right away, during that curriculum week [between cohorts] we have "Sweeps," where we actually go through all the lessons, check out the comments, and make adjustments as needed.

**Q: Do you take feedback from hiring partners into consideration when making curriculum updates?**

**Lowe:** Hiring partner feedback feeds into that larger [revision cycle]. It was the hiring partners' desire for more JavaScript that led to the increasing emphasis on JavaScript, and the front end framework becoming a necessity, which led us to add React in the curriculum—which was not there when I came through the program.

**Huynh:** I get informal feedback from Corinne Babel [Launch Academy's VP of Career Services], and in those discussions, there are different big picture factors that will make us say, OK, we're going to teach this thing.

**Lowe:** With hiring partner feedback, when our Career Services team identifies a feedback trend, it's not so much that we change what we're teaching, but it may cause us to change emphasis. We once got feedback that a particular cohort hadn't done well with a specific data modeling diagram. It was already covered in a lesson, but we made a point to teach it harder in the next cohort so they would not have that [struggle] and not do poorly with it in interviews.

**Q: Can you share some examples of recent changes to the Launch Academy curriculum?**

**Skolnik:** Between Boston cohorts 16 and 17, the on-campus curriculum saw a couple major changes. Previously the order of the curriculum was week one HTTP, week two and three were React, and week four was Advanced Object-Oriented Programming (OOP). So we took the previous week four, which was OOP, and shuffled it in front to week one and pushed everything back. That was actually a reversion to a change that we had made; it had used to be week one, and we pushed it to week four for a few cohorts, decided it wasn't working, and pushed it back to week one. We also reorganized the two React weeks.

**Lowe:** We added at least one more very basic React JSX practice to give students more time with it.

**Skolnik:** Between Boston cohorts 17 and 18, we didn't change too much of the on-campus curriculum, but made some large changes to the Ignition curriculum for the Boston 18 cohort. We updated the introductory JavaScript material, since we had gotten student feedback that pointed to that part of the curriculum as being particularly difficult. We moved the HTML and CSS material to the end of the Ignition program, and integrated it more closely

with the Javascript part of the curriculum. We also removed Object-Oriented Programming in Ruby and Test-Driven Development from Ignition entirely. As a result, we also had to update week one of on campus to include introductory material on Ruby OOP and TDD.

### **Q: Why did Launch Academy start emphasizing more JavaScript in the curriculum?**

**Huynh:** There was definitely an inflection point when job postings were just saying “JavaScript JavaScript JavaScript,” so that was the original impetus.

**Lowe:** It’s no longer possible to be a web developer without being a JavaScript developer. Specifically, it’s no longer possible to be a web developer without working in a complex JavaScript framework. For example, the JavaScript that I first learned comes from a world where your web server, your back end, served up a web page, and there may have been some JavaScript that changed what was on that web page, it may have added a little bit of interactivity. Today, your web server serves up almost nothing on the page, almost a blank page, and a JavaScript framework like React, Angular, Ember, whatever you use, puts up everything that’s on the page, and because they were put up by JavaScript, can immediately be interactive and responsive.

**Skolnik:** The way I like to think of it is our computer hardware has gotten better, so that we are now able to do a lot of fun things in the browser on your computer that didn’t used to be possible. So now instead of the server, the back end, just sending you a thing and doing all of the hard work on its own, now it sends you another...

**Lowe:** A tiny app!

**Skolnik:** Yes, an app! It basically sends you a full-fledged application that does a lot of what the server used to do. So the server is now pushing off a lot of the work, the logic, of what your web page looks like, onto your machine, because we now have the power to do that. Your computer is literally now fast enough to do that. That’s the context of why jobs are looking for JavaScript developers, and particularly these developers who are familiar with frameworks. It’s because everyone wants this new shiny technology that makes the user experience a lot faster and more seamless.

**Lowe:** We no longer think of the web as a series of documents, we think of the web as a series of apps.

**Huynh:** There were a lot of discussions around that decision [to teach JavaScript] because we had initially come out of a purely Ruby program. Ruby is the most accessible to beginners, and we wanted to be accessible to true beginners at Launch Academy. That’s why we started in Ruby, and when we were finding, both from competition and the job market, that JavaScript was really taking a lead, we had to respond. There was discussion around “are we teaching what we think we should teach for software—not just doing what everyone else is doing? Is this still going to truly be accessible to a beginner?” We tried an incremental change to add more JavaScript, and then a much bigger change to embrace the framework.

### **Q: Where do you see the Launch Academy curriculum going in the coming years?**

**Huynh:** I see us taking what we’re already teaching and making it more adjustable, accessible, and helping the true beginners perform better on the curriculum. For example, when we rewrote the Fundamentals to JavaScript and wrote JavaScript into Ignition pre-work, it wasn’t just taking existing curriculum and saying hey, do this earlier. We did a complete from scratch rewrite to mimic how we onboard people to programming using Ruby, but with JavaScript instead so that people actually had the chance to be exposed to basic fundamentals in JavaScript. That foundation allows the student to be truly bilingual by the time they get on campus.

**Lowe:** One structural change worth talking about: If we've been on a journey towards using React as a front end, we're not all the way there yet. We will spend probably the next couple of cohorts finally getting there, to the point where Rails week will be our oldest and most battle-tested curriculum and will be finally ready for its major revision.

In terms of talking about direction, a lot of the recent changes we made to Ignition are very much focused on that last month [of a given cohort]. We have a long-standing group project assignment that we give to students before they go to town on their solo projects. I think the next major architectural curriculum change will be revising that project to be more in line with the stack we're now teaching. We're now considering a major curriculum revision to determine how we can give this project to them so they'll get the most out of the React they have learned, so when they go to work on their solo project, they won't have to decide, "do I want to use more Rails as a front end and just use some React?" We'll set them up to very naturally say "Rails is my back end, React is my front end."

**Skolnik:** I think the focus moving forward will be on polishing and streamlining the direction we're currently moving in. That doesn't mean we'll stop making changes, we may make larger structural changes to the curriculum, [while still supporting] the fundamentals. We have to teach those things.

**Huynh:** We'll continue to perfect the parts of the curriculum to allow people to perform better.

*Sarah Pascarella oversees digital content for Launch Academy.*

## Get Updated

enter your email address...

Subscribe

## Apply

(<https://launchacademy.com/enrollments/new>)

[BLOG \(HTTPS://LAUNCHACADEMY.COM/BLOG\)](https://launchacademy.com/blog)

[CAREERS \(HTTPS://LAUNCHACADEMY.COM/CAREERS\)](https://launchacademy.com/careers)

[CODECABULARY \(HTTPS://LAUNCHACADEMY.COM/CODECABULARY\)](https://launchacademy.com/codecabulary)

[TEAM \(HTTPS://LAUNCHACADEMY.COM/TEAM\)](https://launchacademy.com/team)

[CONTACT US \(HTTPS://LAUNCHACADEMY.COM/#CONTACT-US-SECTION\)](https://launchacademy.com/#contact-us-section)

[PRIVACY \(HTTPS://LAUNCHACADEMY.COM/PRIVACY\\_POLICY\)](https://launchacademy.com/privacy-policy)

[□ \(HTTPS://WWW.FACEBOOK.COM/THELAUNCHACADEMY\)](https://www.facebook.com/thelaunchacademy)

[□ \(HTTPS://TWITTER.COM/LAUNCHACADEMY\)](https://twitter.com/launchacademy)

© COPYRIGHT 2017 LAUNCH ACADEMY. [Terms & Conditions \(https://launchacademy.com/terms\\_and\\_conditions\)](https://launchacademy.com/terms_and_conditions)