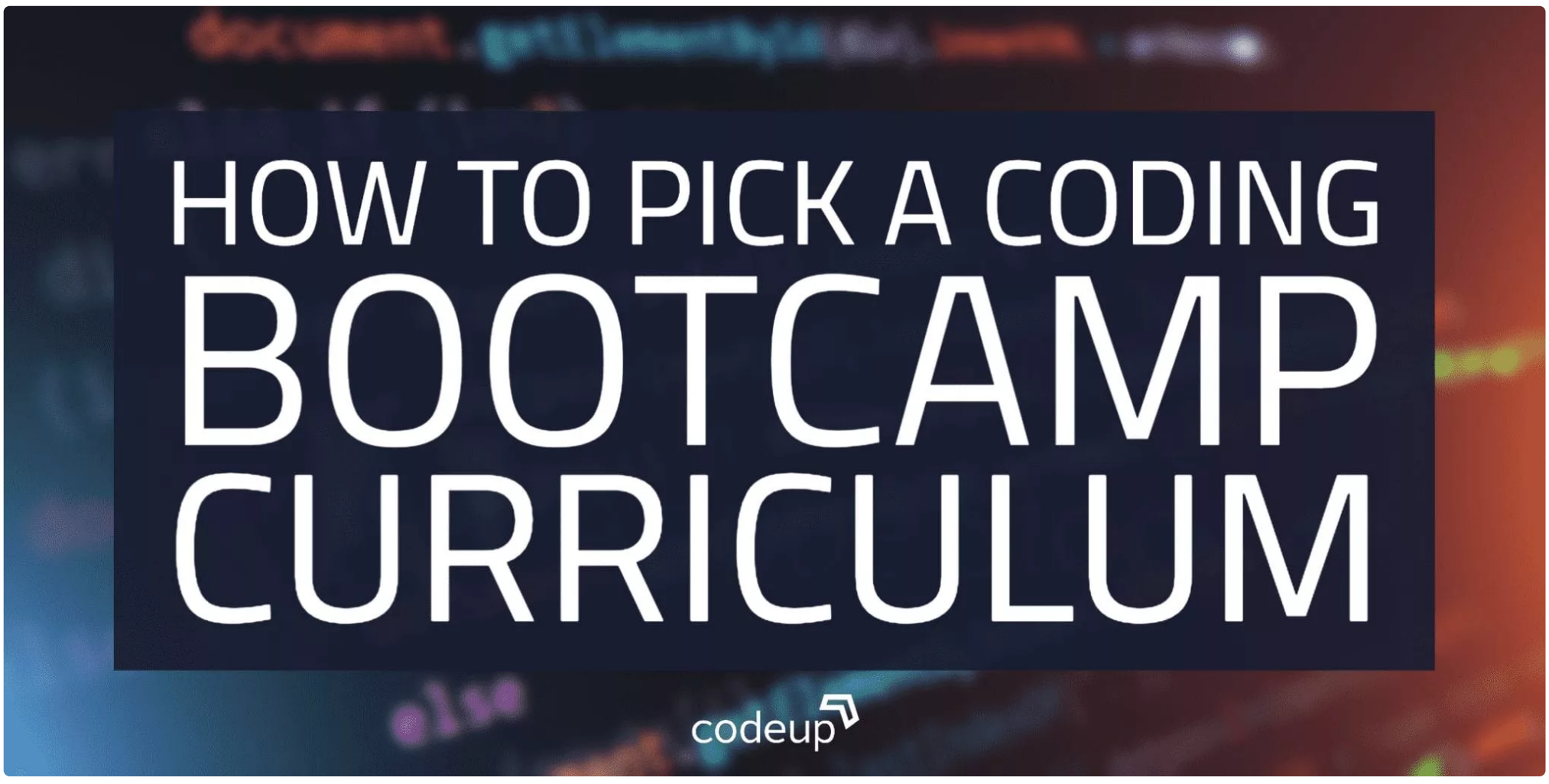
How To Pick A Coding Bootcamp Curriculum

Posted on January 30, 2020 | In Dallas, Data Science, Full Stack Web Development, General, San Antonio



If you're thinking about entering a career as a software developer, you've probably researched a few different bootcamps. During your research, you've probably seen a few different curriculum. Without already BEING a software developer, it's hard to know what's what. In this post, we want to explore how to think about a bootcamp curriculum, and recommend strategies about how to consider the best fit.

Let's start with some terminology. Full-stack web development integrates work on both the front-end and the back-end. The front-end is the user-facing side that you interact with in a web browser. The back-end is the server side that involves the sending and receiving of data. Consider a restaurant website. A front-end only website would show a restaurant menu with prices, dishes, and ordering information. A full-stack web application would allow you to not only view the menu, but place an order and process payment information for that order, interacting with a database and back-end functionality.

Within that understanding, there are a few groupings of technologies:

- **Object Oriented Programming and back-end tech:** This list includes programming languages like PHP, Java, C#, Ruby, and Node.js. These allow you to build functionalities into a web application.
- **Database tools:** Tools like MySQL, MongoDB, PostgreSQL, SQL Server, and Oracle let you store, send, and receive information.
- **Front-end technologies:** Languages and frameworks like JavaScript, Angular, React, HTML, and CSS let you design a front-end interface.
- **Web frameworks:** Spring Boot and Laravel are examples of web frameworks that help you stand up web applications more efficiently.
- **Testing tools:** In production, many companies leverage a methodology called Test Driven Development. This is when developers write tests first, and code second, letting them compare their code against a standard of approval. Common technologies include JUnit, PHPUnit, NUnit, MSTest, Jasmine.

With so many technologies out there, it can be hard to pick what's best to learn. But here's the secret: the specific technologies *do not mat* ow to use these different categories of technologie akeaways are the

fundamental concepts learned. Many Codeup alumni graduate from our Full-Stack Java program and go on to work in PHP, Python, Ruby, Groovy, and other languages. Ultimately, a loop is a loop and an array is an array. Languages differ, but once you've learned an OOP language, the differences become syntactical instead of conceptual.

This leads us to an important point: the more technologies, the *worse*! The quality of a curriculum, and thus the value of it, is not defined by the number of technologies covered. In fact, it's the opposite. Let's give some examples.

Columbia University is one of the premier academic institutions in the world. They are an Ivy League university with a strong reputation. They recently expanded into the bootcamp space, and launched a web development program that covers the following technologies: HTML5, CSS3, JavaScript, jQuery, Bootstrap, Express.js, React.js, Node.js, Database Theory, MongoDB, MySQL, Command Line, Git, and more. All of that in 12 weeks. Let's decode that for you with the terms we've already used. This curriculum promises to teach you:

- Object Oriented Programming and back-end tech: Node.js
- Database tools: Database theory, MondoDB, MySQL
- Front-end technologies: HTML, CSS, jQuery, Bootstrap, Express.js, React.js

Now let's look at Codeup. We teach: HTML, CSS, JavaScript, jQuery, Java, Spring, MySQL. All of that, in 20 weeks. In the terms we've discussed, that's:

- Object Oriented Programming and back-end tech: Java, Spring
- Database tools: MySQL
- Front-end technologies: HTML, CSS, JavaScript, jQuery

The common initial thought is: why spend 20 weeks learning seven technologies when you could spend 12 weeks learning 10? And there lies the misconception. Many bootcamp curricula promise to teach you the latest and greatest technologies: React.js, Angular.js, Express.js, MongoDB, Node.js, etc. etc. etc. That may sound like a better bang for your buck, but it's all a question of priority. Here is the reality of your choices:

- Columbia bootcamp, broad and shallow: gain exposure to a wide variety of technologies in a short amount of time
- Codeup, narrow and deep: gain expertise in software development fundamentals in a narrow scope of technologies

There is no inherent right answer here – it's all about your priorities. That being said, here's what we believe: Learning how to learn, learning how to think like a developer, and learning to program is far more important than gaining exposure to the latest web frameworks. When you understand programming fundamentals, you prepare yourself to learn whatever you want. It's like learning how to work with a car: it's great to know how to drive an Audi, but it's pretty different from understanding how an Audi engine works and how it differs from a Honda.

At Codeup, we focus on crafting you into a software developer. We focus on programming fundamentals, core web technologies, and applied practices. When you graduate, you've landed a job and have the skills to learn any technology. If that sounds like what you're looking for, connect with our Admissions Team and we can tell you more!

Click here to hear our Codeup Alumnus, Po Lin's, story about his journey graduating with a Computer Science degree and how he supplemented Codeup's curriculum to launch a career into software development!

Dallas:

701 Commerce St #100A,

Dallas, TX 75202