

THE ACTUALIZE CURRICULUM 2020

AN OVERVIEW OF THE
TECHNICAL TOPICS COVERED
AT ACTUALIZE



01

OUR PROCESS

“We're not just software engineers - we're educators. Our curriculum is designed first and foremost using the lens of how people learn and think.”

Actualize was founded upon the premise that teaching people to code requires true educational expertise. Software engineering is a complex and vast subject, so it's not enough to just present students with information. Each concept needs to be taught using clear and in-depth explanations, poignant analogies, and exercises that solidify the concept in the student's mind.

As educators, we value teachability over trendiness. Our choice of technologies for the curriculum is based on what allows our students to best grasp the core concepts at hand and then be able to teach themselves any new technology. Teaching the shiniest JavaScript framework that came out yesterday is less important to us.

Indeed, most Actualize grads go on to jobs that require languages and frameworks that are outside our curriculum. And that's because we've prepared our students to be able to teach themselves any new technical topic.



02

TECHNOLOGY OVERVIEW

The Actualize curriculum covers many topics, grouped into following broad categories:

1. **Programming Language.** Subtopics: Ruby, OOP
2. **Version Control.** Subtopics: Git, Github
3. **Back-End Development.** Subtopics: Rails
4. **Database Design.** Subtopics: Postgres, SQL
5. **Front-End Development.** Subtopics: HTML, CSS, JavaScript, Vue.js
6. **Computer Science.** Subtopics: Data structures, algorithms

In the next sections, we will explain many of the subtopics covered in the curriculum. Don't worry if you don't understand everything - it'll all be explained in depth when you join the course!

03

PROGRAMMING LANGUAGE THROUGH RUBY

RUBY

Ruby is a multipurpose programming language developed in Japan during the 1990's. It became popularized through its use in conjunction with the Rails web development framework and has become a popular choice for back-end web development.

We find that Ruby's natural, English-like syntax softens the learning curve for students learning the fundamentals of programming. Our students begin learning Ruby from Day 1 of the Prework phase, and use it to learn critical concepts such as variables, code flow, methods, conditionals, loops, arrays, hashes, and functions.



04

OBJECT-ORIENTED PROGRAMMING (OOP)

OOP is a popular programming technique that allows for code simplification and reuse. Using Ruby, which is an object-oriented language, we explore important OOP topics including objects, classes, methods, modules, inheritance, and composition.

VERSION CONTROL WITH GIT AND GITHUB

Version control is what allows software engineers to keep track of every version of the codebase that ever existed. This makes it easy to roll back mistakes or discover at what point a bug was introduced. Git is a popular tool that makes version control manageable.

Github is a website that connects with Git to store all your versions of code online. In addition to the storage of code in the cloud, Github also makes it easy for an entire team to work on a single codebase without everyone overwriting each others' work. Lastly, Github is a great site for discovering new and exciting open-source projects created by other developers.

05

RUBY ON RAILS

BACK-END WEB DEVELOPMENT WITH RAILS

Rails is a web framework for back-end web development. It is used in conjunction with Ruby, and is popularly known as Ruby on Rails.

Ruby on Rails is one of the most powerful and productive web frameworks available, powering websites from major companies such as GitHub, AirBnB, Groupon, Hulu, Kickstarter - and the list goes on.

There are many different web frameworks to choose from. However, from an educational standpoint, we believe that Ruby on Rails is the best choice as the first framework for a student to learn. This is for two reasons.

First, it is written in the Ruby programming language, which is an ideal first language for beginners due to its readable syntax and practical functionality.

Second, Ruby on Rails has had a huge influence on the web frameworks that came after it, from Ruby Sinatra, Node Express, Java Spring, PHP Laravel, and Elixir Phoenix, to name a few. This means that learning Ruby on Rails as a first web framework enables a student to leverage their knowledge to learn a multitude of other frameworks.



06

API DEVELOPMENT

MICROSERVICES THROUGH APIS (APPLICATION PROGRAMMING INTERFACES)

APIs are programs that deliver data across the web. This data is offered in a variety of formats, with JSON being among the most popular.

Modern full-stack web development often uses a back end comprised of APIs and a JavaScript-powered front-end that communicates with these APIs. The full-stack apps that you will be building at Actualize take this form as well.

The concept of Service-Oriented Architecture (SOA) is that your web API can serve not just as the back end for your web application, but also mobile apps and other interfaces as well.

You will learn how to interact with existing web APIs as well as how to build your own APIs. Using Ruby on Rails, you will build web APIs that serve as the back-end of your web applications.

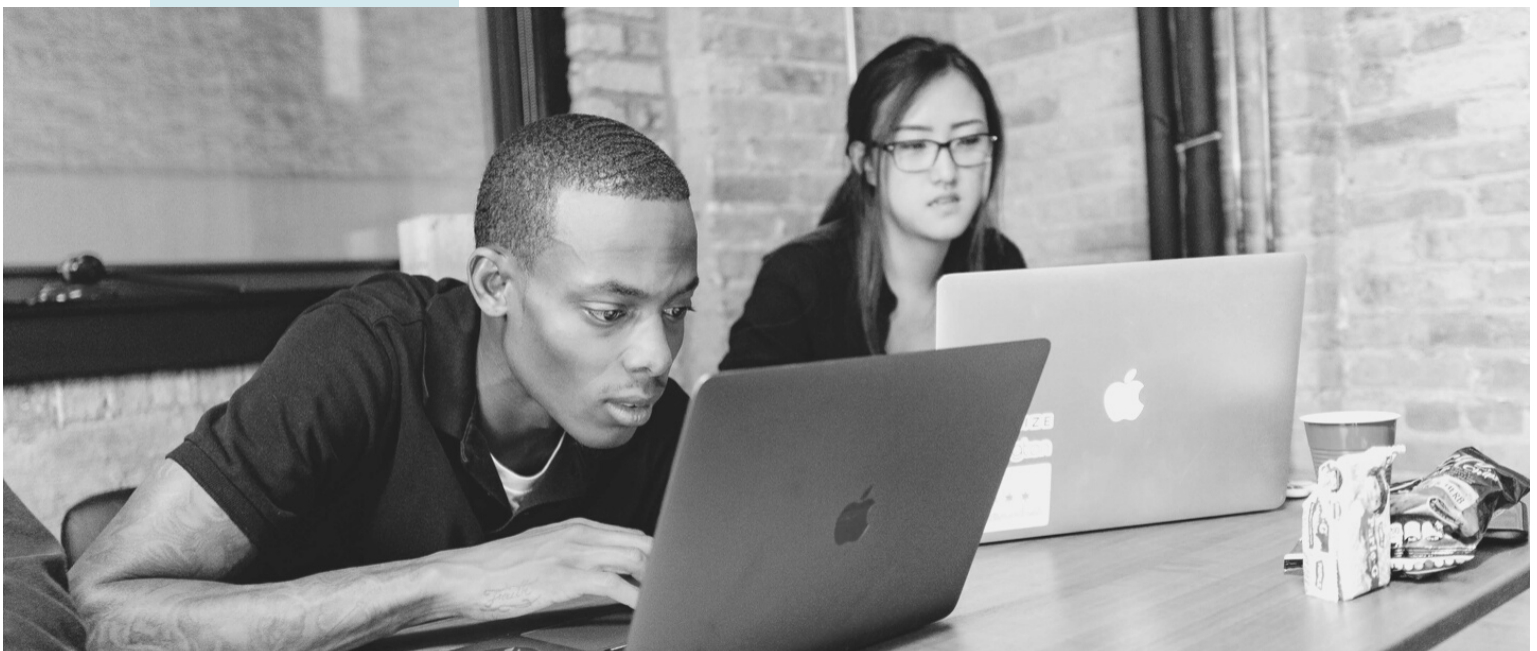
07

DATABASE DESIGN WITH POSTGRES AND SQL

RELATIONAL DATABASE THEORY

The database is a critical component of the web applications you will build, as the database contains all the information that your app deals with. Using the Postgres database, you will learn how to design a database schema to set up database tables and their associations with one another.

You will learn SQL, the language that powers most relational databases. From there, you will learn to use ActiveRecord, the ORM (Object-Relational Mapper) that translates your Rails code into SQL so your web app can store and retrieve information from your database.



08

THE FRONT END

Front-End Development with HTML, CSS, JavaScript, and VueJS

Full-stack web development means creating both the back end and front ends of a web application. Simply put, the back end is code that exists on a web server and delivers data and code across the web to your computer.

The front end is the code that your browser receives (from the back end) and then displays as a webpage that you can view and interact with.

HTML, CSS, and JavaScript are used together to and comprise the bulk of front-end web development.

HTML & CSS

HTML (Hypertext Markup Language) is the language that declares the structure of a webpage, and CSS (Cascading Style Sheets) is the language that describes how the webpage will appear visually. Using a simple analogy, you can think of HTML as the bones of a webpage, and the CSS as its flesh.

With HTML and CSS alone, however, a website is just a pretty thing to look at. JavaScript is what makes the webpage dynamic, allowing the user to interact with it. (Does that make JavaScript the webpage's blood?)

09

JAVASCRIPT

JavaScript is the key language of front-end web development, as it is the language understood by all web browsers. While back-end development can be accomplished with a variety of languages (such as Ruby on Rails), functional front-end development is dominated almost exclusively by JavaScript.

In addition to the core features of JavaScript, you will learn about ES2015+, DOM manipulation, and the modern JavaScript project setup, including Babel, Webpack, NPM, and NPM Scripts.

FRONT-END FRAMEWORKS WITH VUE.JS

Most front-end JavaScript development is coded in conjunction with a front-end framework (otherwise known as a JavaScript framework). Front-end frameworks make it convenient to organize your JavaScript code in a predictable and reusable format.

Launched by Evan You and a team of passionate volunteer contributors in 2014, Vue.js is one of the most popular JavaScript frameworks today. It specializes in creating SPAs (Single Page Applications), dynamic web applications that feel more like a desktop app than a website.



10

THE TEACHABILITY OF VUE.JS

There are many popular front-end frameworks out there, with React and Vue currently getting a lot of buzz. Again, our choices of technologies always involves teachability over trendiness. We choose to teach Vue.js specifically because it is a framework that is designed in a progressive manner - one can use it in a simple way to get started and layer on the complexity as needed.

Therefore, we see Vue.js as being the best educational choice for our students, as it enables students to see and understand the fundamentals of how JavaScript frameworks work in general.

After learning Vue.js with a focus on these fundamentals, it becomes easy to pick up additional JavaScript frameworks thereafter. In fact, as soon as the bootcamp is over many of our graduates build their own side-projects using React and other frameworks. The grads often remark on how learning Vue.js made it simple for them to teach themselves these other technologies.

11

DATA STRUCTURES & ALGORITHMS

COMPUTER SCIENCE THEORY

While data structures and algorithms are part of basic computer programming, we delve further into more complex analyses of these topics to broaden our students' perspectives on theoretical computer science.

Actualize Founder & CEO Jay Wengrow is the author of the popular book "A Common-Sense Guide to Data Structures and Algorithms," and our curriculum covers important aspects of these topics including time complexity, Big-O analysis, algorithm optimization, and more.



12

LEARN HOW TO LEARN

The most important skill needed to become a software engineer is the ability to learn how to learn new things. Everything we do at Actualize is focused on this.

Our goal is that our grads shouldn't need us anymore. They will have the ability to learn any new technology through research, tinkering, and building. Because that is exactly what professional developers do every day.

QUESTIONS?

If you have any questions about the Actualize or our curriculum, we'd love to hear from you!

Website: <https://actualize.co>

Email: hello@actualize.co

Phone: 773-766-5965

