

3 FULL STACK CURRICULUM

# Software Development

## Full-Time Online

16+ weeks, 70-90 hours/week



**Full-Time**  
class commitment



**Career Path Focus**  
built into curriculum



**Learn by Doing**  
real projects, real datasets

Join our 13,000+ global alumni and kickstart your career path in tech.

# Program Overview

**Your career path into software development** begins on your first day of class. Within 16 weeks, you'll study to become a self-sufficient, versatile developer who has the critical skills to pursue a career path in tech.

**Anyone can learn to code**, but the path to becoming a developer isn't easy. Most successful students dedicate 70-90 hours/week to bootcamp diving deep into their studies and building friendships along the way.

**You'll start coding from day one.** At Coding Dojo, our learning environment fosters collaboration and deep learning; not competition.



**Up Next:** The Whole Curriculum



# The Whole Curriculum



## Week One Programming Basics

To kickoff the program, you'll explore habits, computer basics, and fundamental programming concepts and skills necessary to be successful in your bootcamp!

- What You'll Focus On:**
- Basic computer literacy
  - Algorithmic foundations
  - Learning stamina



## Weeks Two to Four Web Fundamentals

You'll then move to Web Fundamentals—a three week course that starts with the basics to provide a good overview before jumping into specific languages.

- What You'll Focus On:**
- HTML
  - CSS
  - Git/Github
  - jQuery (optional)
  - Wire-framing (optional)



## Weeks Five to Eight Python Full Stack

We'll then dive into our first full stack language: Python. We'll start slow with small projects, then work our way up to designing a full framework project with your instructor and classmates.

- What You'll Focus On:**
- Python Fundamentals
  - Python OOP
  - MySQL
  - Flask
  - MVC
  - Deployment



## Weeks Nine to Twelve Javascript Full Stack

Mid-program, we'll start on Javascript—You'll examine a wide-range of applicable formats and projects to help you get ready for real-world application.

- What You'll Focus On:**
- JavaScript
  - Node.JS
  - Express.JS
  - [Socket.io](#)
  - MongoDB
  - React
  - Deployment



## Weeks Thirteen to Sixteen Java Full Stack or C#/.NET Stack

At the very end, you'll make a choice to study Java or C#/.NET the last four weeks of the course.

- What You'll Focus On (Java):**
- Java
  - Java Fundamentals
  - Java OOP
  - Java Web Development
  - Java Spring
  - Deployment

**OR**

- What You'll Focus On (C#Net):**
- C#/.NET
  - C# Fundamentals
  - C# OOP
  - [ASP.NET](#) Core
  - Object Relational Mapping (ORM)
  - Identity Framework Core
  - Deployment



**Up Next:** A Day in the Life

# An Example Day's Schedule in a Full-Time Program



## Morning

**8:50 AM - 9 AM** Login to Zoom session for morning Algorithms

**10 AM** Recap Algos & Discussion Lecture

**11 AM** Group Activities & Setting the Day's Expectations

## 24/7 Cohort Access

Your access to our LEARN Platform and Discord is available 24/7. Access your materials at whatever time you need them.

## Self Study

Most students dedicate 70-90 hours a week to self-study, though you may need more or less depending on your learning style and experience.

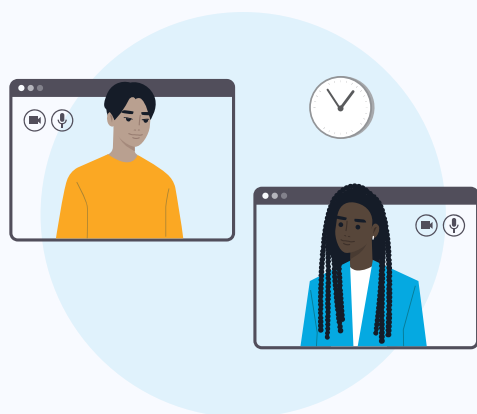
## Lectures

Live participation is held

**Monday - Friday** from 9am - 5pm MST. Students should plan for 12+ hour days with 8 hours of instruction.

## Optional Office Hours

Need more assistance understanding a concept? Optional office hours are held Monday - Friday when class is in session an hour prior to the morning kickoff between 8am - 9am MST.



## Mid-Morning

**12 PM** Enjoy lunch!

**1 PM - 5 PM** Labs including demos, code reviews, and extra sessions



## Evening

Additional Assignments & Self-Study

**5 PM - 9 PM** Complete daily assignments, read lessons for following day



**Up Next:** Let's Dive Into the Stacks!



# Let's Dive Into the Stacks!

## What does 3 stack mean?

A **stack** refers to a programming language, and when we refer to 'full stack', we mean you'll study every facet of that programming language.



### Stack One: Python

Python is one of the most popular languages in the industry<sup>1</sup>. Its diversity, adaptability, and easy-to-master basics makes it the perfect language to start with at bootcamp.

#### What Python is used for:

- Web Applications
- Web Development
- Machine Learning
- Data Science
- Cloud Infrastructure



### Stack Two: Javascript

JavaScript is ideal for building dynamic websites and applications. It runs on every application level making it an efficient, modern approach to web development.

#### What Javascript is used for:

- Web Applications
- Mobile Applications
- Game Development
- Web Servers
- Animation



### Stack Three: Java or C#/.NET

Java is a high-level language which revolutionized language development post-release.

#### What Java is used for:

- Web Applications
- Mobile Applications
- Game Development
- Web Servers

### C#/.NET (as an optional third stack)

C#/.NET covers both the programming language C# and the .NET Framework which is an application framework library. It's extremely versatile, making the language popular for writing desktop apps, background services, and apps.

#### What C#/.NET is used for:

- Web Services Applications
- Client-Server Applications
- Console Applications
- Web Applications
- Games

<sup>1</sup> <https://www.tiobe.com/tiobe-index/> (visited 3/9/2023)



# Programming Basics

**WEEK 1**

To kickoff the program, you'll examine habits, computer basics, and fundamental programming concepts and skills necessary to be successful in your bootcamp! During this section, students study basic computer literacy skills, such as how to install and navigate basic programming tools. Students apply algorithmic thinking to make predictions of common programming skills, such as variables, arrays, conditionals, functions, and loops.

Additionally, students experience the rigor and intensity of the bootcamp, strengthening their cognitive processing stamina, resiliency, and other behavioral skills necessary for a bootcamp. By the end of the course, students should walk away with the basic computer literacy, algorithmic foundations, and learning stamina needed to find success in a bootcamp.

**Up Next:** Web Fundamentals

# Web Fundamentals

## Front-End Development & The Web

**WEEKS 2-4**

### HTML

#### Intro to HTML

- Basic Nesting Practices, Indentation
- The Head & Body
- Body Tags (lists, tables, etc.)
- Building Forms & Declaring Input Values
- Containers, Elements, Attributes, & Classes

### CSS

#### Intro to CSS

- CSS Selectors & Declarations
- Inspecting Element
- Inline, Block, Float, and Positioning
- Div Layout & Formatting
- Styling Text & How Fonts Work
- Using Properties & Backgrounds
- Replicating Complete User Interfaces

#### More Styling\*

- Intro to Bootstrap

### Git/Github

#### Git & Version Control

- Using Terminal Commands\*
- How to Create & Utilize a Repository
- Git Workflow Overview & States\*

#### Github

- How to Use a Github Repository

### Javascript

- Functions & Debugging
- Event handling
- Parameters
- Implementing Dynamic Content
- Traversing DOM Elements

### jQuery\*

#### Intro to jQuery

- jQuery Functions
- Essentials of the jQuery Library

### Responsive Web Design\*

#### Intro to Responsive Web Design (RWD)

- Breakpoints, Units, & Media Queries
- Basics to Typesetting & Scaling
- Cross-device RWD
- Grid System, Fluid Grids, & Adaptive Layouts

#### CSS Frameworks

- Responsive Typography
- Using CSS Reset & Boilerpoint

### Wireframing\*

- Wireframing Fundamentals

\*Optional Topics

**Up Next:** Python



# Python

## Stack One: Full Stack Development

WEEKS 5-8

### Python

#### Intro to Python

- Variables, Data Types & Best Practices
- Using Strings & Built-in String Functions
- List Creation & Manipulation
- Dictionaries in Python
- Nested Dictionaries & Lists
- Conditionals, Operators, & Nested Loops
- Functions in Python

### Python OOP

#### Intro to Object Oriented Programming

- Classes, Constructors and Creating Object Instances
- Setting and Updating Attributes
- Adding and Using Methods
- Chaining Methods
- Implementing Static and Class Methods
- Setting Up Associations Between Classes
- How to Use Modules & Packages in Python
- Introduction to Inheritance, Polymorphism, Encapsulation and Abstraction

#### Python Test Driven Development (TDD)\*

- Unit Testing in Python & Outcome
- How to Use Assertions
- TDD Methods: setUp & tearDown

#### Advanced Python

- Variable Length Arguments
- Ternary Operators in Python
- Using Anonymous Functions (Lambdas) in Python

### MySQL

#### Intro to MySQL

- Database Design & Relationships
- Entity Relationship Diagrams (ERDs)
- Conventions & Common Data Types
- Normalization
- Basic MySQL Queries for CRUD
- MySQL Functions
- Joins

### Flask

#### Intro to Flask

- Routing in Flask Applications
- Building & Using Forms
- Rendering Templates & View
- Delivering Static Content
- The Different HTTP Methods
- Implementing Cookies & Session
- Hidden Inputs & Form Validation

#### Flask w/ MySQL

- Using PyMySQL to Connect to a Database
- Basic Data Security
- SQL Injection, Hashing Passwords & Bcrypt
- Back-end Validation and User Authentication Logic

### MVC

- Creating the MVC Design Pattern in Flask
- Modularization, Using Models & Controllers
- Building Full-Stack Flask Applications

### Deployment

- Amazon Web Services (EC2)
- Linux

### AJAX\*

- Fetching Data and Parsing JSON
- Using External APIs and API Keys
- Sending JSON Responses to the Client
- Intro to Asynchronous vs Synchronous Execution
- Manipulating the DOM to Display Dynamic Data

\*Optional Topics

# Java

**WEEKS 13-16**

## Stack Three: Full Stack Development

### Java Fundamentals

#### Intro to Java

- Java Development Kit Installation
- Executing Java Programs
- Variables, Data Types, & Type Casting
- Control Structures & Exceptions

### Java OOP

#### Intro to Object Oriented Programming

- Creating Objects & Classes
- Methods, Member Variables & Constructors
- Overloading & this
- Inheritance & Packages

#### Advanced Java OOP

- Use of Static
- Interfaces & Abstract Classes
- Annotations
- Java Beans

#### Data Structures\*

- Doubly Linked Lists
- Tries

### Java Spring

#### Spring Intro

- Routing
- Java Server Pages
- Session
- Form Submission
- GET vs POST
- Dependency Injection

#### Spring MVC

- Model, View, and Controller (MVC) Design Pattern
- Java Persistence API (JPA)
- MySQL Connections
- Persistent Model Annotations
- Relationships
- Advanced Queries

#### Spring Security

- Spring Security Overview
- Authentication & Authorization
- Servlet API Integration
- Spring MVC Integration

### Deployment

- Amazon Web Services (EC2)

\*Optional Topics

**Up Next:** C#/.NET (optional stack 3 instead of Java)

# C#/.NET

**ALTERNATIVE:  
WEEKS 13-16**

Optional Stack Three, in Place of Java

## C# Fundamentals

### Intro to C#

- .NET Console Applications
- Variables, Types, Type Casting, & Functions
- Control Structures
- Debugging .NET Applications (VS Code)

## C# OOP

### Intro to Object Oriented Programming

- Classes & Objects
- Access Modifiers
- Inheritance & Polymorphism
- Encapsulation with Properties

### Advanced C# OOP

- Interfaces
- Abstract Classes

## ASP.NET Core

- Dependency Injection with ASP Services
- MVC Architecture
- Razor
- ViewModels
- Custom User Authentication/Authorization

## Object Relational Mapping (ORM)

### Working with ORMs

- LINQ
- Entity Framework Core
- User Authentication/Authorization
- Identity Roles

## Deployment

- Amazon Web Services (EC2)
- Production Environments
- Hosting with Nginx/Supervisor



**Up Next:** Career Services



# Career Services

**Career Services Support.** Career Services is dedicated to providing quality career preparation and job readiness services. This provides students with knowledge and skills to navigate the job search process in the tech industry. Milestones:

## 1

### Professional Profile & Portfolio Building

Career services will guide you through creating your digital footprint, industry exploration, and building out a professional profile. Milestones:

- ✓ Profile Creation and Optimization
- ✓ Github Portfolio Production
- ✓ Resume Development & Curation

## 2

### Job Prospecting & Application Guidance

Discover tactics and job search strategies to help you prepare for the job search post-bootcamp. Milestones:

- ✓ Real Job Searches
- ✓ Sample Applications
- ✓ Hiring Manager Communications
- ✓ Job Title Refinement

## 3

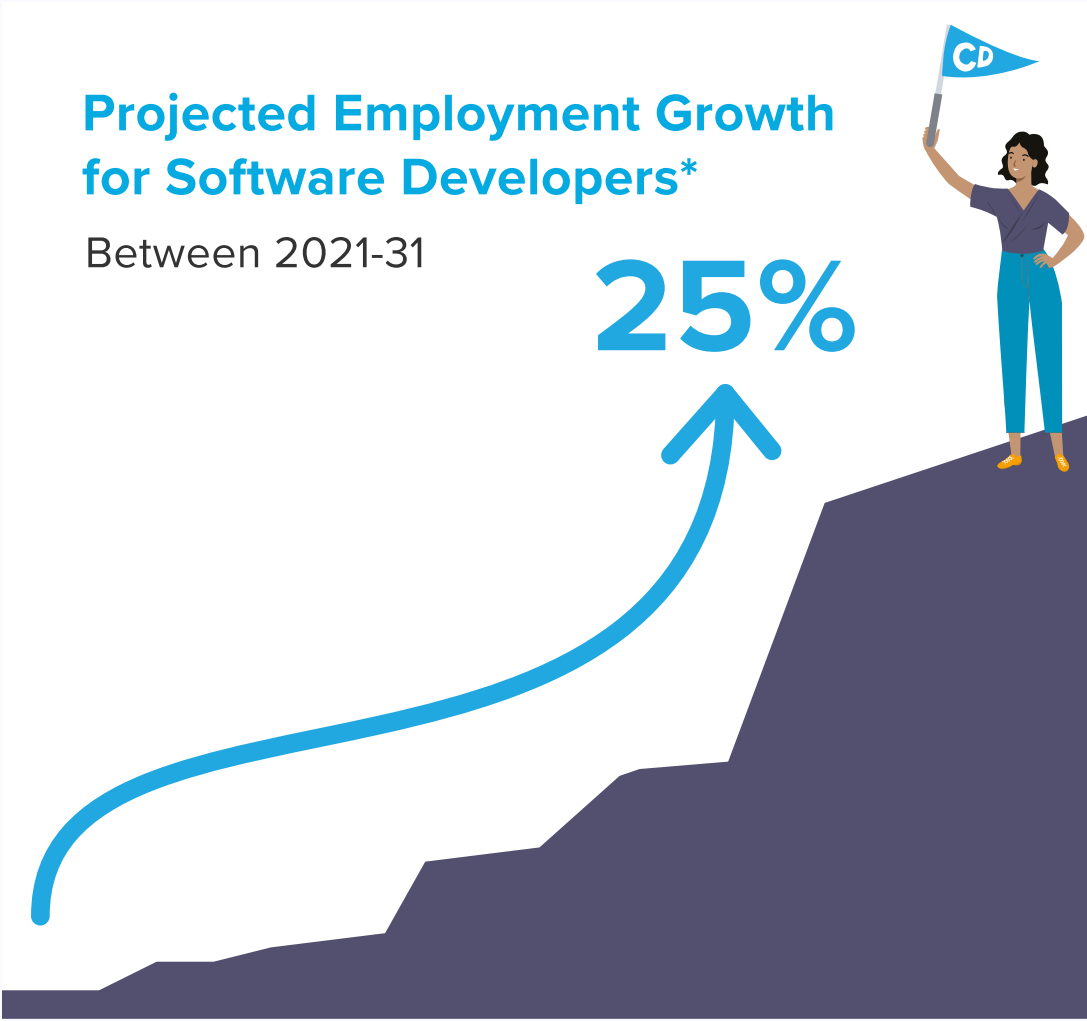
### Interview Prep & Job Search Strategies

Exit the program with your new skills, Coding Dojo Certificate of Achievement, and job search strategies to begin your search for a new career path. Milestones:

- ✓ Interview Preparation and Negotiation Strategies
- ✓ Technical Job Skills Tests
- ✓ Target Compensation Management

Coding Dojo cannot guarantee employment, salary or career advancement.

# Industry Trends



SOURCE: \*Bureau of Labor Statistics, U.S. Department of Labor, Occupational Outlook Handbook, Software Developers, Quality Assurance Analysts, and Testers, at <https://www.bls.gov/ooh/computer-and-information-technology/software-developers.htm> (visited April 19, 2023). This data represents national figures and is not based on school-specific information. Conditions in your area may vary. Some career paths may require further education or job experience.



**Up Next:** How to Enroll

# How to Enroll



## Do Your Research

- Explore our programs on our website and view other program overviews.
- Schedule a call with one of our Admissions Advisors who will talk through your future career path goals and what program may best suit you.
- Attend an Open House to meet directly with our Instruction and Career Service Managers.



## Submit Application

- Submit your application! The application process takes less than 5 minutes and does not include a technical assessment.
- Complete a quick 30-minute interview with our Admissions team.
- Receive your decision within 2-3 business days.



## Explore Financing Options

- Our Admissions Advisors will help you explore our financing options.
- Coding Dojo offers a variety of payment options, financing, and partial-scholarships for those who qualify.



## Finalize Your Enrollment

- Submit your deposit, confirm your financing, and sign your Enrollment Agreement to reserve your seat in class!
- Your Admissions Advisor will introduce you to your Student Experience Manager who will help you get ready to start bootcamp.



**Up Next:** Financing Options



# Financing Options



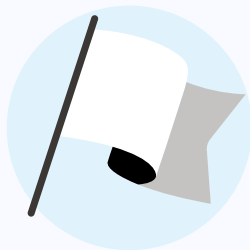
## Installment Plans

Spread tuition payments out over your course with customizable installment plans.



## Third-Party Financing

Finance your bootcamp with a third-party loan from a variety of vendors or source your own.



## Pay in Full

Pay your tuition in full and get started.

Schedule a call with an Admissions Advisor to discuss which payment or financing option is right for you.

[Chat with Admissions](#)

# Software Development

## Part-Time Online

18 - 34 weeks, 30 hours/week (Accelerated Program)



**Part-Time**  
class commitment



**Career Path Focus**  
built into curriculum



**Learn by Doing**  
real projects, real datasets

Join our 13,000+ global alumni and kickstart your career path in tech.



# Program Overview

**Your career path into software development** begins on your first day of class. In 18 to 34 weeks, you'll study to become a self-sufficient, versatile developer who has the critical skills to pursue a career path in tech.

**Anyone can learn to code**, but the path to becoming a developer isn't easy.

**You'll start coding from day one.**

Dive into a fast, project-based learning environment that fosters collaboration, not competition.



**Up Next:** Choose Which Part-Time Program



# A Part-Time Option to Fit Your Schedule:

## Accelerated Program

Our accelerated program allows you to choose your own adventure! Choose 1, 2, or 3 full stacks at a part-time pace.



18 - 34 Weeks



30 Hrs/Week

Includes complete web fundamentals,  
then choose from the following stacks:



Python



Javascript



Java



**Up Next:** About the Accelerated Program

# About the Accelerated Program

**Learn to build applications** in some of the top programming stacks of 2023. Pick between Python, JavaScript, or Java as your stack, or choose to extend the program and explore multiple languages.



# An Example Day's Schedule in an **Accelerated Program**



## Morning

Head to Work

## 24/7 Cohort Access

Your access to our LEARN Platform and Discord is available 24/7. Access your materials at whatever time you need them.



## Evening

Lecture, Office  
Hours & Self Study

## Self Study

Most students dedicate 30-35 hours a week to self-study, though you may need more or less depending on your learning style and experience.

## Lectures

Live lectures are held **three times per week** for an hour from 6pm-7pm MST. Lecture days are **Tuesday, Wednesday, and Thursday**.

## Optional Office Hours

Need more assistance understanding a concept? Optional office hours are held an hour prior to lecture times between 5pm-6pm MST.



Up Next: Stacks

# Let's Dive Into the Stacks!

## What does 3 stack mean?

A **stack** refers to a programming language, and when we refer to 'full stack', we mean you'll study every facet of that programming language.



### Stack One: Python

Python is one of the most popular languages in the industry<sup>1</sup>. Its diversity, adaptability, and easy-to-master basics makes it the perfect language to start with at bootcamp.

#### What Python is used for:

- Web Applications
- Web Development
- Machine Learning
- Data Science
- Cloud Infrastructure



### Stack Two: Javascript

JavaScript is ideal for building dynamic websites and applications. It runs on every application level making it an efficient, modern approach to web development.

#### What Javascript is used for:

- Web Applications
- Mobile Applications
- Game Development
- Web Servers
- Animation



### Stack Three: Java

Java is a high-level language which revolutionized language development post-release. It's adopted widely in the industry and going strong for 20+ years.

#### What Java is used for:

- Web Applications
- Mobile Applications
- Game Development
- Web Servers

<sup>1</sup> <https://www.tiobe.com/tiobe-index/> (visited 3/9/2023)





# Programming Basics

To kickoff the program, you'll examine habits, computer basics, and fundamental programming concepts and skills necessary to be successful in your bootcamp! During this section, students study basic computer literacy skills, such as how to install and navigate basic programming tools. Students apply algorithmic thinking to make predictions of common programming skills, such as variables, arrays, conditionals, functions, and loops.

Additionally, students experience the rigor and intensity of the bootcamp, strengthening their cognitive processing stamina, resiliency, and other behavioral skills necessary for a bootcamp. By the end of the course, students should walk away with the basic computer literacy, algorithmic foundations, and learning stamina needed to find success in a bootcamp.



**Up Next:** Web Fundamentals

# Web Fundamentals

## Front-End Development & The Web

### HTML

#### Intro to HTML

- Basic Nesting Practices, Indentation
- The Head & Body
- Body Tags (lists, tables, etc.)
- Building Forms & Declaring Input Values
- Containers, Elements, Attributes, & Classes

### CSS

#### Intro to CSS

- CSS Selectors & Declarations
- Inspecting Element
- Inline, Block, Float, and Positioning
- Div Layout & Formatting
- Styling Text & How Fonts Work
- Using Properties & Backgrounds
- Replicating Complete User Interfaces
- Using CSS Reset & Boilerpoint

#### More Styling\*

- Intro to Bootstrap

### Git/Github

#### Git & Version Control

- Using Terminal Commands\*
- How to Create & Utilize a Repository
- Git Workflow Overview & States\*

#### Github

- How to Use a Github Repository

### Javascript

- Functions & Debugging
- Event handling
- Parameters
- Implementing Dynamic Content
- Traversing DOM Elements

### jQuery\*

- Essentials of the jQuery Library
- jQuery UI Library & More Libraries\*

### Responsive Web Design\*

#### Intro to Responsive Web Design (RWD)

- Breakpoints, Units, & Media Queries
- Basics to Typesetting & Scaling
- Cross-device RWD
- Grid System, Fluid Grids, & Adaptive Layouts

### Wireframing\*

- Wireframing Fundamentals

\*Optional Topics



**Up Next:** Python

# Python

## MySQL

### Intro to MySQL

- Database Design & Relationships
- Entity Relationship Diagrams (ERD)
- Database Normalization
- MySQL Workbench & Querying
- Conventions & Common Data Types
- How to Use ERDs
- Using a Database with Your UI  
Recreating ERDs\*

## Python

### Intro to Python

- Variables, Data Types & Best Practices
- Using Strings & Built-in String Functions
- List Creation & Manipulation
- Using Tuples & Built-in Tuple Functions
- How to Use Dictionaries in Python
- Conditionals, Operators, & Nested Loops
- Constructing Functions in Python

## Python OOP

### Intro to Object Oriented Programming

- Creating Objects & Classes
- Adding Properties/Attributes to Classes
- Constructing & Adding Methods to Classes
- Chaining Methods & Using Magic Methods
- How to Use Modules & Packages in Python
- Creating Multiple Objects
- Updating Methods with ‘Super’
- Overriding Inheritance & Polymorphism

### Python Test Driven Development (TDD)\*

- Unit Testing in Python & Outcomes
- How to Use Assertions Using
- TDD Methods: setUp & tearDown

### Advanced Python\*

- How to Use Multiple Arguments
- Ternary Operators in Python
- Using Lambda\*
- Using Composition Over Inheritance\*

## Flask

### Intro to Flask

- Routing in Flask Applications
- Building & Using Forms
- Rendering Templates & Views
- Delivering Static Content
- The Different HTTP Methods
- Implementing Cookies & Sessions
- Hidden Inputs & Form Validation

### Flask w/ SQL

- Import, Export, & Connect Your Database
- Connecting & Running Python Across Files
- Database Communication & Validation
- Encryption & Data Security Basics

## MVC

### Intro to Model View Controller (MVC)

- Views, Session Classes & Session Data
- How to Use Models with Controllers
- Data Validation
- Using Bcrypt with MVC
- How to Use Multiple Controllers & Models

## Deployment

- Amazon Web Services (EC2)
- Linux

\*Optional Topics



**Up Next:** Javascript

# JavaScript

## JavaScript

### Fundamentals

- Declaring & Referencing
- Variables Variable Hoisting in JavaScript
- Conditionals, Operators, & Nested Loops
- Using Arrays & Loops in JavaScript
- Objects, Functions, & Function Scoping
- Variable Hoisting with Scoping
- Return Statements in JavaScript
- Function Hoisting

### JavaScript OOP

- How to Use Object Constructors
- Common Constructors: 'This' & 'New'
- Private Methods & Variables
- Creating Prototype Objects in JavaScript
- Best Practices for JavaScript OOP

### Advanced JavaScript

- How to Use Callbacks
- Delegating Functionality & Event Handling

## Node.JS

### Intro to Node

- How to Use Package Managers (NPM/Bower)
- Making a Full Web Server
- How to Work with Node Modules
- Common & Useful Node Modules
- Node.JS

### Modularization

- Using Require & Module.exports
- How to Modularize Existing Projects

## Express.JS

- HTTP Methods: Forms, Data Transfers, & Routing
- RESTful Routing

## Socket.io

- Applications with Real-time Communication

## MongoDB

### MongoDB & Mongoose

- MongoDB Overview, CRUD Ops
- Intro to Mongoose
- Dependencies in Mongoose
- Mongoose Communication with MongoDB
- Mongoose Methods
- Data Validation with Mongoose
- Create Associations Between Mongo Objects

## React

- Create React App
- Class Based Components
- Props, Children, Synthetic Events
- State, LifeCycle Methods
- Functional Components
- useState, useEffect, useReducer
- context API
- Manage application state using hooks: useState, useEffect
- useReducer, useContext\*

## Deployment

- Amazon Web Services (EC2)
- Linux
- Production Environments

\*Optional Topics



**Up Next:** Java



# Java

## Java Fundamentals

### Intro to Java

- Java Development Kit Installation
- Executing Java Programs
- Variables, Data Types, & Type Casting
- Control Structures & Exceptions

## Java OOP

### Intro to Object Oriented Programming

- Creating Objects & Classes
- Methods, Member Variables & Constructors
- Overloading & this
- Inheritance & Packages

### Advanced Java OOP

- Use of Static
- Interfaces & Abstract Classes
- Annotations
- Java Beans

### Data Structures\*

- Doubly Linked Lists
- Tries

## Java Spring

### Spring Intro

- Routing
- Java Server Pages
- Session
- Form Submission
- GET vs POST
- Dependency Injection

### Spring MVC

- Model, View, and Controller (MVC) Design Pattern
- Java Persistence API (JPA)
- MySQL Connections
- Persistent Model Annotations
- Relationships
- Advanced Queries

### Spring Security

- Spring Security Overview
- Authentication & Authorization
- Servlet API Integration
- Spring MVC Integration

## Deployment

- Amazon Web Services (EC2)

\*Optional Topics



**Up Next:** Career Services

# Career Services

**Career Services Support.** Career Services is dedicated to providing quality career preparation and job readiness services. This provides students with knowledge and skills to navigate the job search process in the tech industry. Milestones:

## 1

### Professional Profile & Portfolio Building

Career services will guide you through creating your digital footprint, industry exploration, and building out a professional profile. Milestones:

- ✓ LinkedIn Profile Creation and Optimization
- ✓ Github Portfolio Production
- ✓ Resume Development & Curation

## 2

### Job Prospecting & Application Guidance

Discover tactics and job search strategies to help you prepare for the job search post-bootcamp. Milestones:

- ✓ Real Job Searches
- ✓ Sample Applications
- ✓ Hiring Manager Communications
- ✓ Job Title Refinement

## 3

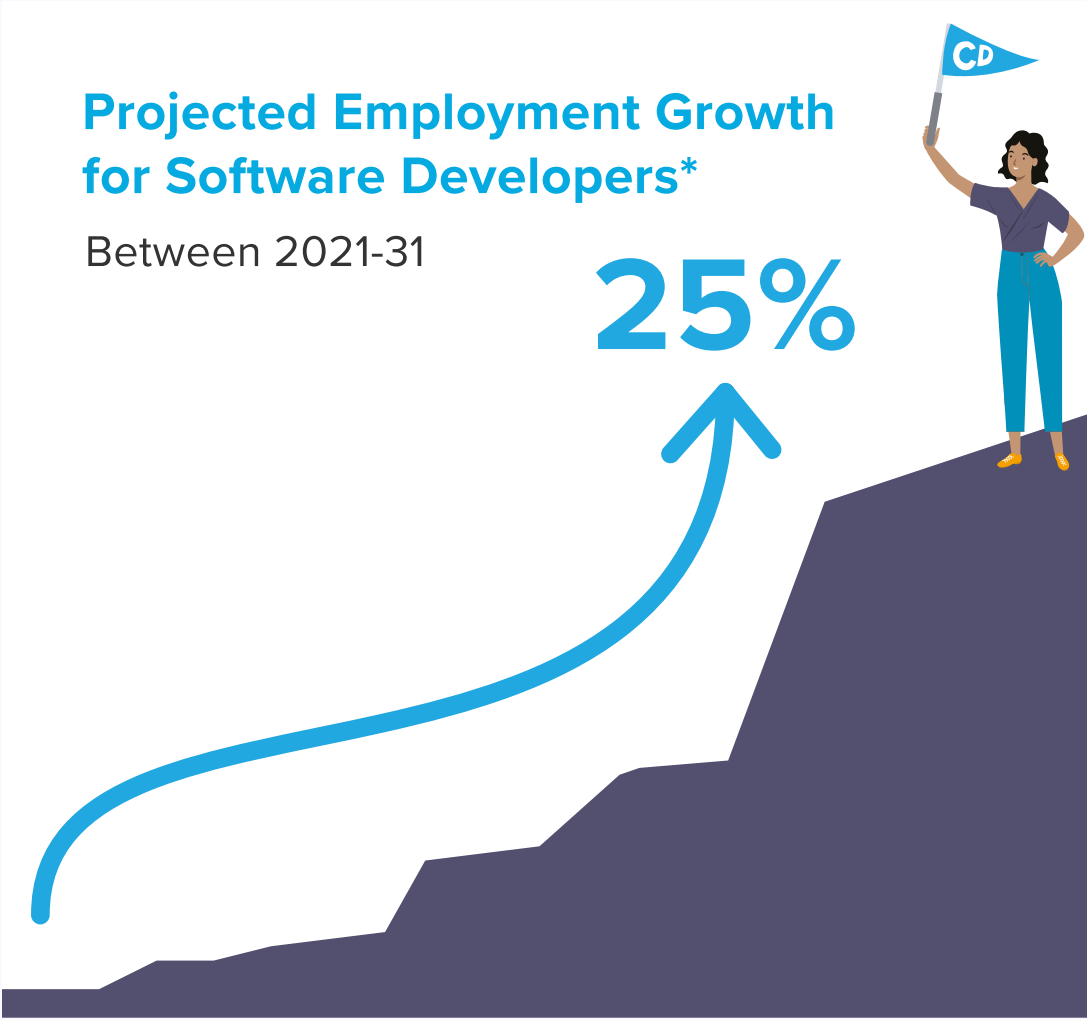
### Interview Prep & Job Search Strategies

Exit the program with your new skills, Coding Dojo Certificate of Achievement, and job search strategies to begin your search for a new career path. Milestones:

- ✓ Interview Preparation and Negotiation Strategies
- ✓ Technical Job Skills Tests
- ✓ Target Compensation Management

Coding Dojo cannot guarantee employment, salary or career advancement.

# Industry Trends



SOURCE: \*Bureau of Labor Statistics, U.S. Department of Labor, Occupational Outlook Handbook, Software Developers, Quality Assurance Analysts, and Testers, at <https://www.bls.gov/ooh/computer-and-information-technology/software-developers.htm> (visited April 19, 2023). This data represents national figures and is not based on school-specific information. Conditions in your area may vary. Some career paths may require further education or job experience.



**Up Next:** How to Enroll

# How to Enroll



## Do Your Research

- Explore our programs on our website and view other program overviews.
- Schedule a call with one of our Admissions Advisors who will talk through your future career path goals and what program may best suit you.
- Attend an Open House to meet directly with our Instruction and Career Service Managers.



## Submit Application

- Submit your application! The application process takes less than 5 minutes and does not include a technical assessment.
- Complete a quick 30-minute interview with our Admissions team.
- Receive your decision within 2-3 business days.



## Explore Financing Options

- Our Admissions Advisors will help you explore our financing options.
- Coding Dojo offers a variety of payment options, financing, and partial-scholarships for those who qualify.



## Finalize Your Enrollment

- Submit your deposit, confirm your financing, and sign your Enrollment Agreement to reserve your seat in class!
- Your Admissions Advisor will introduce you to your Student Experience Manager who will help you get ready to start bootcamp.



**Up Next:** Financing Options



# Financing Options



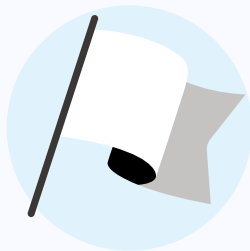
## Installments

Spread tuition payments out over your course with customizable installment plans.



## Third Party Financing

Finance your bootcamp with a third party loan from a variety of vendors or source your own.



## Pay in Full

Pay your tuition in full and get started.

Schedule a call with an Admissions Advisor to discuss which payment or financing option is right for you.

[Chat with Admissions](#)