Onsite Bootcamp

14 Week Immersive Bootcamp
3 Full Stack Curriculum

4500+ grads to date
$60k - $125k avg alumni salary*

Over 4,500 alumni, hired by tech companies worldwide

*As of Feb 2018 alumni data
Onsite Bootcamp

Your career as a software developer starts on your first day in class.

Within 14 weeks we’ll turn you into a self-sufficient, versatile developer who has all the critical skills to have a long, healthy career in tech.

Learn by Doing

You’ll start coding from day one on campus. Dive into a fast, project-based learning environment that fosters collaboration, not competition.

Anyone Can Learn to Code

Anyone can learn to code, but the path to becoming a developer isn’t easy. The most successful students dedicate at least 70-90 hours/week to the bootcamp.

A Typical Day at the Dojo

Activities subject to change based on campus and curriculum
3 Full Stack Curriculum

We’re here to maximize your career opportunities and coding mastery. You’ll learn 3 full stacks, have a portfolio to show, and 3x the job prospects.

Level Up, Stack by Stack

Web Fund.
- Terminal
- Git/GitHub
- HTML5
- CSS3
- Javascript
- jQuery

Python
- Python 3
- OOP
- Flask
- Django
- MySQL
- Ajax

C#.NET
- C#
- ASP.NET Core 2
- LINQ
- Dapper
- Entity Framework
- Identity

Ruby on Rails
- Ruby
- Rails
- RSpec
- Capybara
- PostgreSQL
- Active Record

JAVA
- Java 8
- MySQL
- JSPs
- Spring Data JPA
- Spring Boot
- Spring Security

MERN
- Javascript ES6
- Mongodb
- Express.js
- React
- Node.js
- Socket.io

Awards & Recognition

Curriculum subject to change during attendance due to mid-course improvements
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

Intro to CSS3 & More Styling*
- Building Shapes
- Constructing Complex Tables
- Intro to Bootstrap
- CSS Preprocessors, LESS, & SASS

Git / Github
Git & Version Control
- Using Terminal Commands
- How to Create & Utilize a Repository
- Making, Tracking, & Reverting Changes
- Git Workflow Overview & States*
- Advanced Git Commands & Concepts*
- Branching, Merging, & Conflicts*

Github
- How to Use a Github Repository
- Forking, Cloning, & Pulling*
- Github Collaboration & Workflow*

jQuery
Intro to jQuery
- jQuery Functions & Debugging
- Parameters & Getters/Setters
- Essentials of the jQuery Library

Advanced jQuery
- Implementing Dynamic Content
- Callbacks in jQuery
- Traversing DOM Elements
- Forms in jQuery
- 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

CSS Frameworks
- Responsive Typography
- Using CSS Reset & Boilerpoint

Wireframing*
- Balsamic Overview
- Wireframing Fundamentals

*Optional topics
Python Full Stack Development

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 Dictionaries in Python
- Conditionals, Operators, & Nested Loops
- Constructing Functions in Python

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
- Overriding Inheritance & Polymorphism
- 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

Pylot MVC
Intro to Pylot Model View Controller (MVC)
- Views, Session Classes & Session Data
- How to Use Models with Controllers
- Data Validation with Pylot
- Using Bcrypt with Pylot MVC
- How to Use Multiple Controllers & Models

Deployment
- Amazon Web Services (EC2)
- Linux
- PostgreSQL

*Optional topics
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 Web Development
Java on the Web
- Servlets & Web Containers
- Query Parameters
- Java Servlet Pages
- Light MVC Patterns
- Session & POST Patterns

Java Spring
Spring Fundamentals
- Spring Overview
- Spring Tool Suite
- Intro to Spring Boot
- Spring MVC Apps

Spring Data I & II
- MySQL Connections
- Repositories & Spring Data - JPA
- Persistent Model Annotations
- Relationships
- Advanced Queries

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

Deployment
- Amazon Web Services (EC2)
- Linux
- PostreSQL

*Optional topics
MERN
Full Stack Development

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)
- File System Module & HTTP
- Making a Full Web Server
- How to Work with Node Modules
- Common & Useful Node Modules

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

Express.JS
- Render Templates With Express View Engines
- HTTP Methods: Forms, Data Transfers, & 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
- RESTful Routing with Mongoose & Express

React
- Create React App
- Class Based Components
- Props, Children, Synthetic Events
- State, LifeCycle Methods
- Functional Components
- useState, useEffect, useReducer
- context API

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

*Optional topics
### C# Fundamentals

**Intro to C#**
- .NET Core Console Applications
- Variables, Types, Type Casting, & Functions
- Control Structures
- Debugging .NET Core 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
- Generics

**Data Structures**
- Singly Linked Lists
- Doubly Linked Lists
- Tries

### ASP.NET Core

- Dependency Injection with ASP Services
- MVC Architecture
- Razor View Engine
- View Modeling
- Extension Methods
- Custom User Authentication/Authorization

### Object Relational Mapping (ORM)

**Working with ORMs**
- LINQ
- Dapper
- Entity Framework Core

### Identity Framework Core

- User Authentication/Authorization
- Identity Roles
- Third Party OAuth

### Deployment

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

*Optional topics*
How to Enroll

The goal of admissions is to make sure you’re ready for the challenges of the program, and that Coding Dojo is the right fit for you.

We look for motivation, how well you work with others, and perseverance.

1. Submit Application
2. Schedule Interview
3. Admissions Decision
4. Deposit to Enroll

Financing

Standard
- 2 Payments
- 50% of tuition due week 1
- 50% of tuition due midway

Monthly Plan
- As low as $250/month
- 30/60 month terms available

Financing provided by: [SKILLSfund](#)