



Master Course Packet

Onsite & Online
Full Time & Part-Time
Web Development, Data Science,
Cybersecurity, and UI/UX Design

8000+
grads to date

\$72,325
avg alumni salary*

Over 8000 alumni hired by tech companies worldwide



UBER

LinkedIn

*As of Feb 2018 alumni data



Onsite Bootcamp

14 Week Immersive Bootcamp
3 Full Stack Curriculum

8000+
grads to date

\$72,325
avg alumni salary*

Over 8000 alumni, hired by tech companies worldwide



UBER

LinkedIn

*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

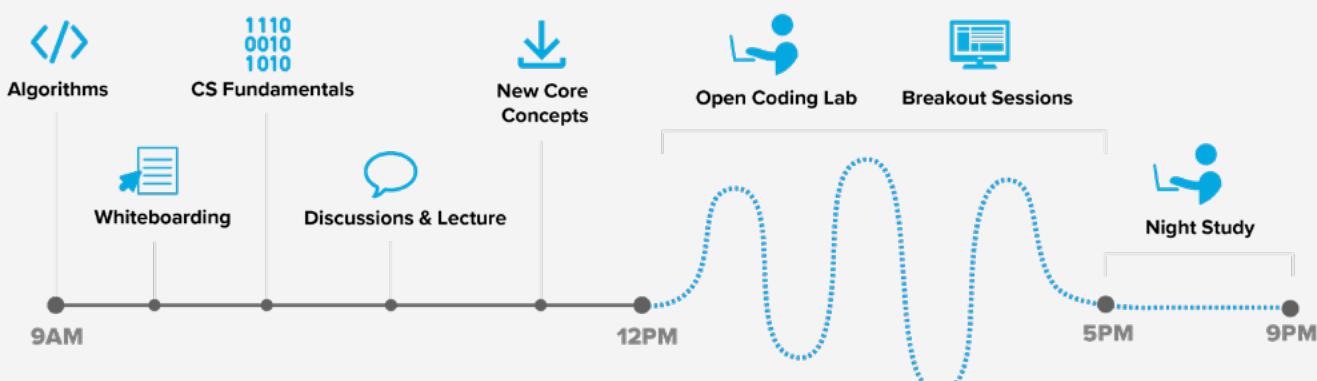
You'll start coding from day one on campus. Dive into a fast, project-based learning environment that fosters collaboration, not competition.



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.

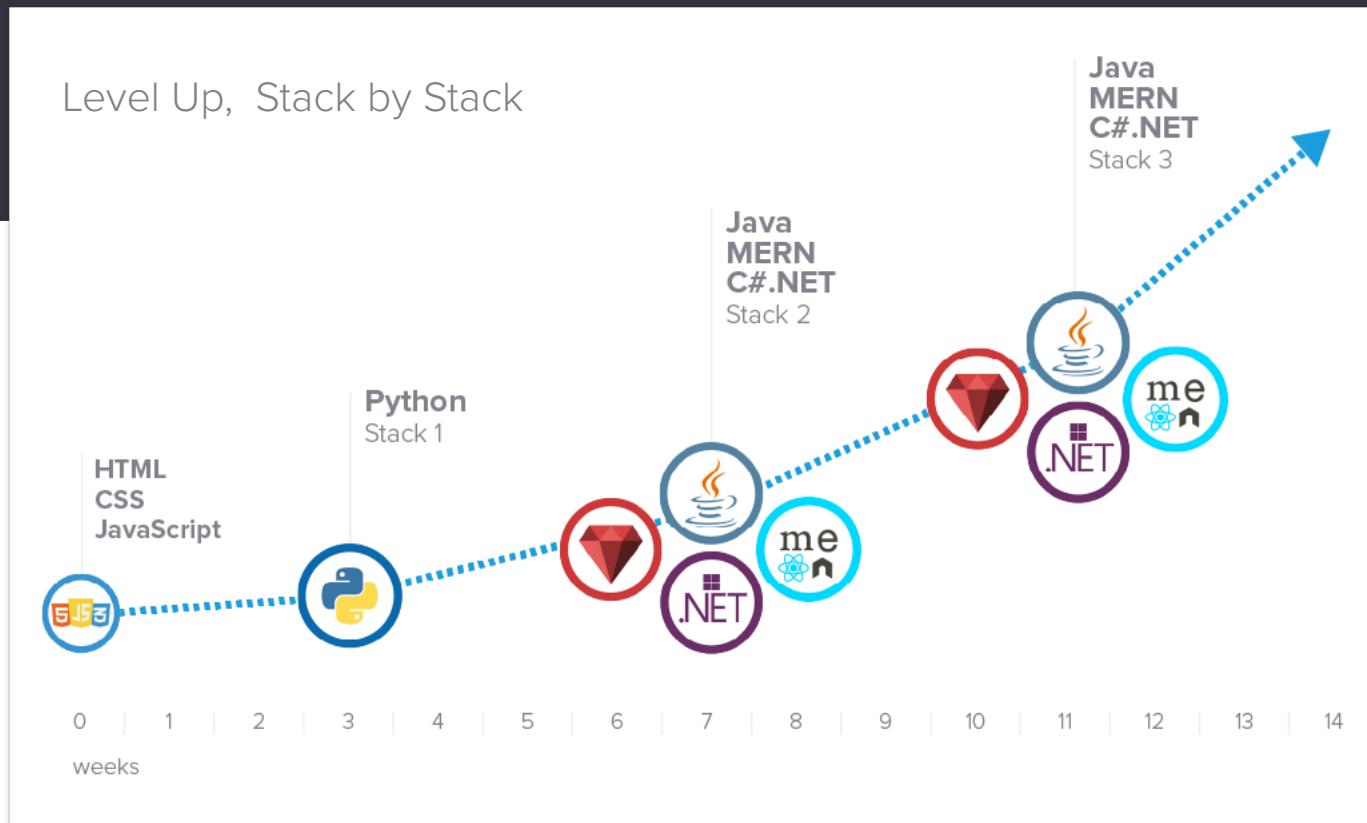
A Typical Day at the Dojo



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.



Web Fund.

- Terminal
- Git/GitHub
- HTML5
- CSS3
- Javascript
- jQuery



Python

- Python 3
- OOP
- Flask
- MySQL
- Ajax



C#.NET

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



JAVA

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



MERN

- Javascript ES6
- MongoDb
- Express.js
- React
- Node.js
- Socket.io



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

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



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 Modules & Packages in Python
- Creating Multiple Objects
- Updating Methods with 'Super'

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

Deployment

- Amazon Web Services (EC2)
- Linux
- PostgreSQL



Java

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 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

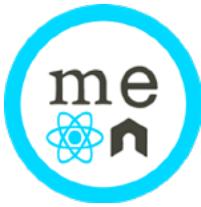
Spring Security

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

Deployment

- Amazon Web Services (EC2)
- Linux
- PostgreSQL

*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



C# .NET

Full Stack Development

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



Online Bootcamp

Full-Time Online
3 Full Stack Curriculum

8000+
grads to date

\$72,325
avg alumni salary*

Over 8000 alumni, hired by tech companies worldwide



UBER

LinkedIn

*As of Feb 2018 alumni data

Online Full-Time

No matter where you are in the world, your career as a software developer starts on your first day.

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.



Hands-on, Structured Teaching

Dive into an immersive online learning environment filled with live mentorship, instruction, and collaboration with real instructors and classmates.

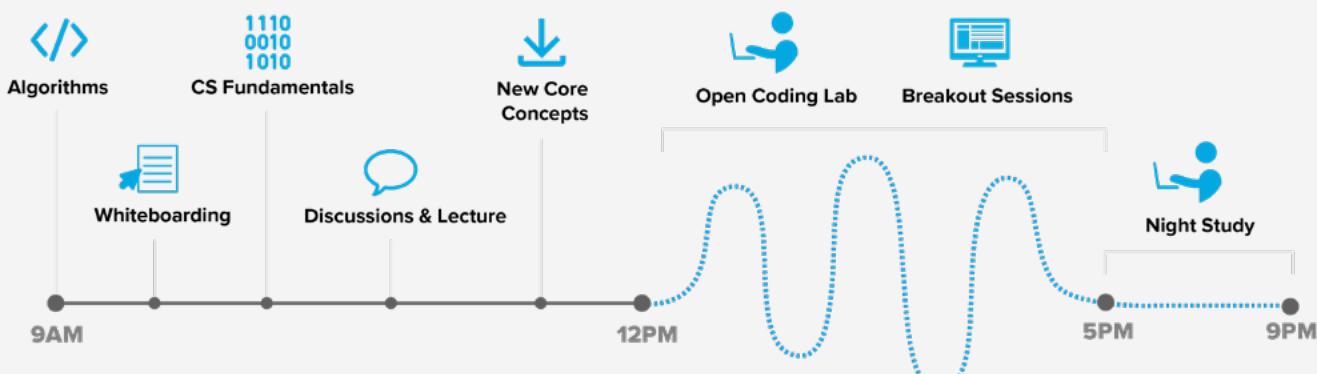
All from the comfort of your own home.



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.

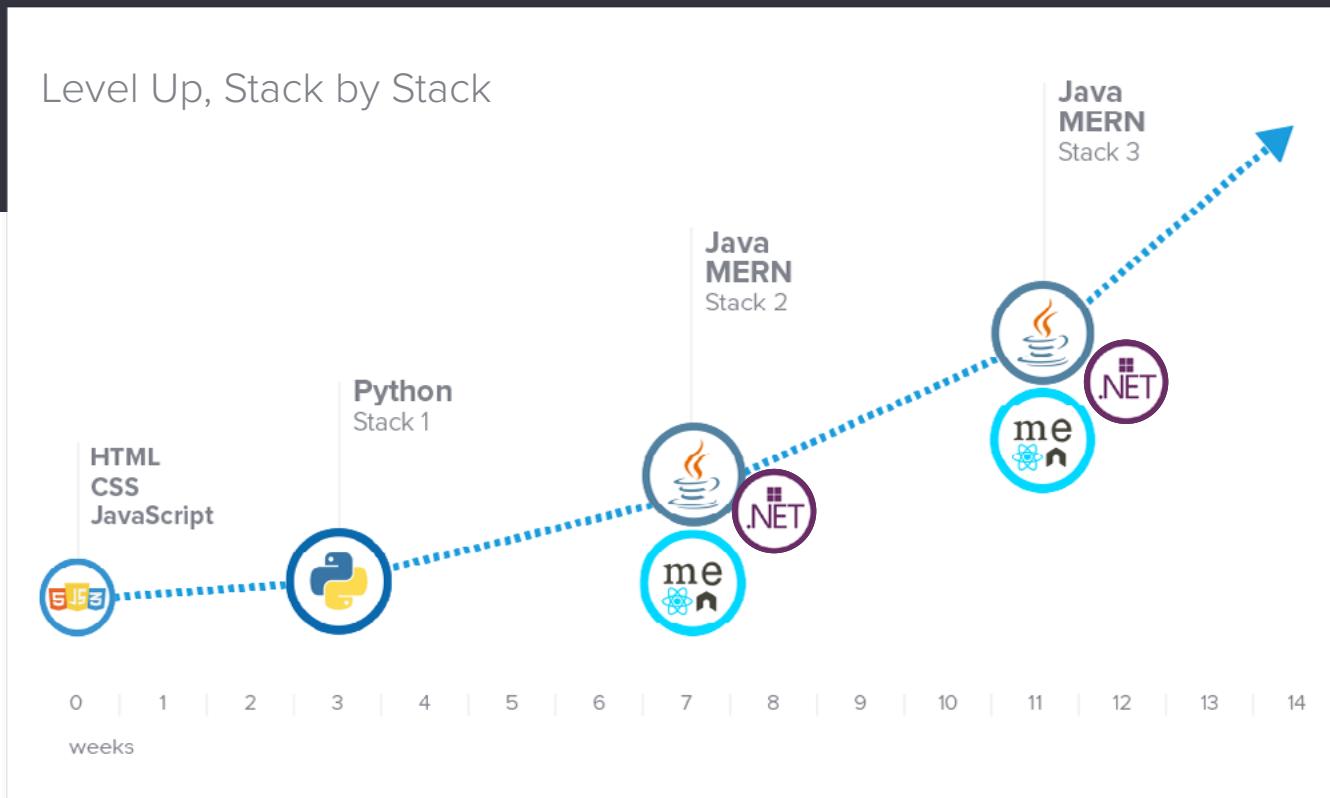
A Typical Day in the Online Bootcamp



Activities subject to change based on campus and curriculum

3 Full Stacks Online

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.



Web Fund.

- Terminal
- Git/GitHub
- HTML5
- CSS3
- Javascript
- jQuery



Python

- Python 3
- OOP
- Flask
- MySQL
- Ajax



Java

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



MERN

- Javascript ES6
- MongoDb
- Express.js
- React
- Node.js
- Socket.io



C#.NET

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



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

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 Modules & Packages in Python
- Creating Multiple Objects
- Updating Methods with 'Super'

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

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 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

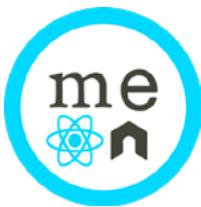
Spring Security

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

Deployment

- Amazon Web Services (EC2)
- Linux
- PostgreSQL

*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



C# .NET

Full Stack Development

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



Part-Time Online

Accelerated and Flex Pacing
2-4 Hours / Week in Lecture
10-30 Hours / Week in Self-Study

10-30 Hrs

per week

3 Stacks

to choose from

16 to 28 Wks

flexible schedule

Over 8,000 alumni, hired by tech companies worldwide



UBER

LinkedIn

*As of Feb 2018 alumni data

Online Part-Time

In 16 to 28 weeks, you can transition to a career in development without quitting your day job.

This program is a flexible alternative that provides full, online access to our Python curriculum -- complete with live support and collaboration with instructors and classmates.



Two Options to Fit Your Schedule

ACCELERATED

16 weeks

25 hrs/wk



Complete web fundamentals, then choose from the following stacks:



FLEX

28 weeks

14 hrs/wk



Complete web fundamentals, then start Python



ONLY Python is available through Flex at this time.

ACCELERATED

Learn to build applications in the top programming stacks of 2020. Pick between Python, MERN, or Java as your stack, or choose to extend the program and learn multiple languages.

Awards & Recognition

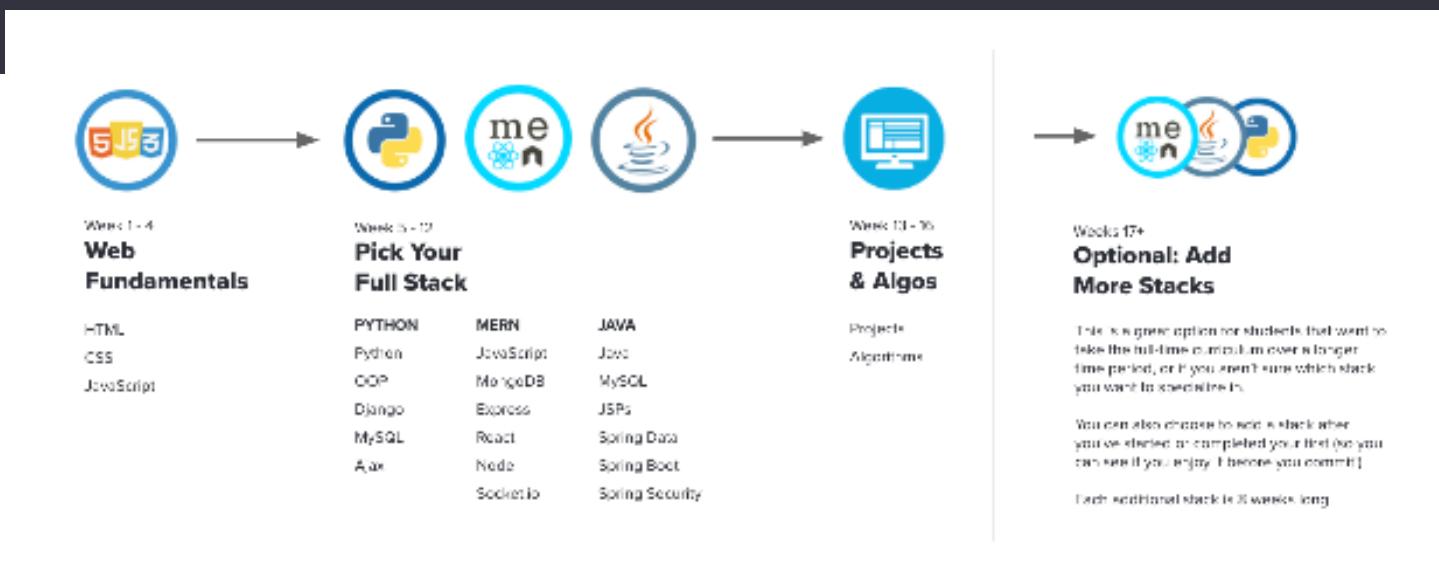


Innovation
Challenge
Winner

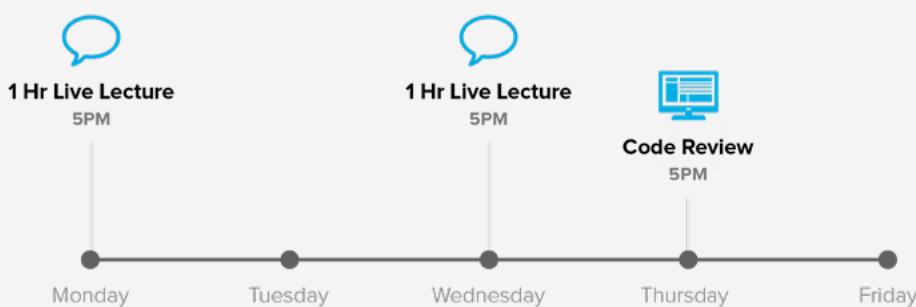


Microsoft
Virtual
Academy
★★★★★
COURSE RATING

Your Progression Plan



A Typical Week in the Part-Time Program



Lectures are delivered either on Mon/Wed or Tues/Thurs
Flex program only has one lecture per week



Self Study

20-30 hours/wk in Accelerated
10-15 hours/wk in Flex



30 min. Code Review

available for assignment feedback and help Monday-Friday as instructors' schedule allows



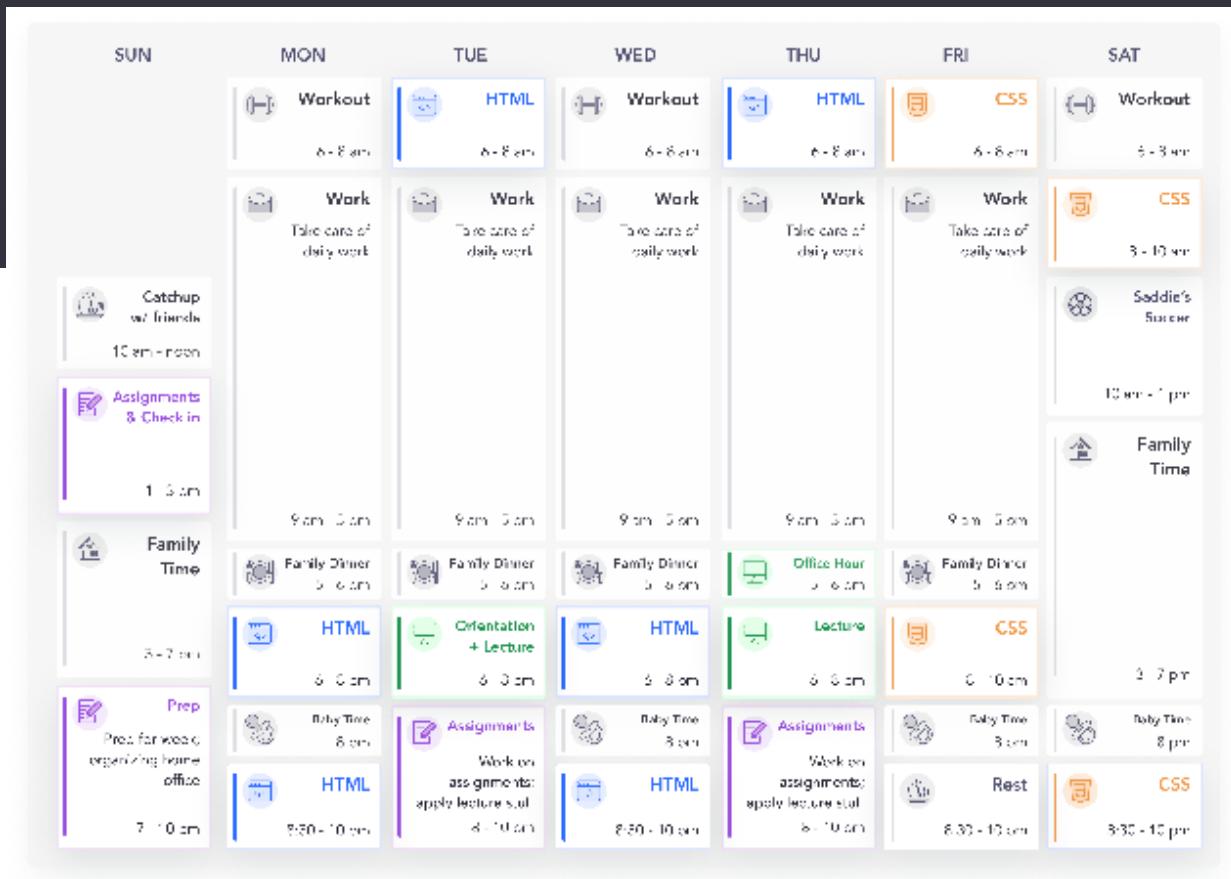
TA Support

Mon-Fri: 11:00am - 8:00pm
Sat: 8:00am - 6:00pm
Sun: 8:00am - 2:00pm

All times in PST

Time Management

Here's what a typical week might look like for someone who continues to work full-time as well as participate in family activities while in the Accelerated program.



Pro Tips from Student Success

Overestimate the time you need for self-study

The Part-Time Online program expects you to dedicate at least 20 hours per week in the learning platform working through content. So, for the first few weeks, allocate 24 hrs for that work. It is easier to scale back than scale up.

Create a calendar and stick with it!

It sounds simple, but a calendar can be shared with family and friends to help you stay accountable and to get insight into when you're going to be heads down. It also gives you a reality check into how much time you actually spend.

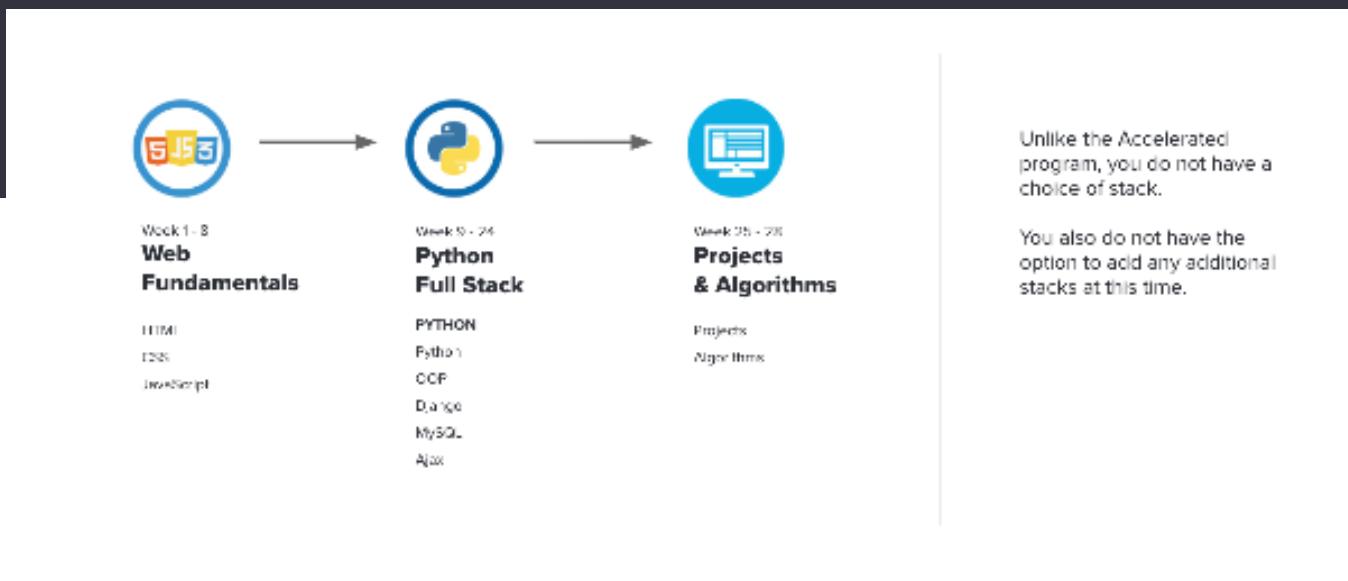
List out responsibilities and see who can help

Create a list of your household and family responsibilities. See if you can offload any tasks or get additional help from housemates, friends, and family. If you'll be working during this time, do the same exercise with coworkers.

FLEX

The same Python curriculum, over a longer amount of time, so you can manage the rest of your commitments more easily.

Your Progression Plan



**Whether you choose Accelerated or Flex,
we are here to support you.**



Hands-on, Structured Teaching

Dive into an immersive online learning environment filled with live mentorship, instruction, and collaboration with real instructors and classmates.

All from the comfort of your own home.



Anyone Can Learn to Code

Anyone can learn to code, but the path to becoming a developer isn't easy. Students typically dedicate 20-30 hours a week to self-study in the accelerated program, and 10-15 hours in Flex.



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

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 Modules & Packages in Python
- Creating Multiple Objects
- Updating Methods with 'Super'

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

Deployment

- Amazon Web Services (EC2)
- Linux
- PostgreSQL



Java

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 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

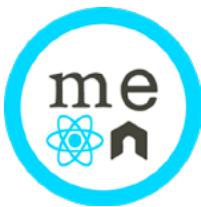
Spring Security

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

Deployment

- Amazon Web Services (EC2)
- Linux
- PostgreSQL

*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



Data Science & Machine Learning in Python

Learn Data Science Online in 16 Weeks

Part-Time

class commitment

Career Focus

built into curriculum

Learn by Doing

real projects, real data

Over 8,000 alumni, hired by tech companies worldwide



*As of Feb 2018 alumni data

Overview

Take a deep dive into the fundamentals of data science and machine learning in Python over 16 weeks. You'll gain a comprehensive understanding of the entire data science process from end-to-end, including data prep, data analysis and visualization, as well as how to apply machine learning algorithms to various situations or tasks.

You'll walk away with a project portfolio showcasing your data science acumen as well as an understanding of one of the fastest growing job sectors out there.



Designed for the Real World



Learn By Doing

A practical, accelerated curriculum designed for you to fix real-work problems by building real Data Science projects and solutions. You'll tackle over 100 interview-style questions so that you're fully prepared for the job search.



Core Concepts, Real Data-Sets

In 16 weeks, you'll learn the principle concepts and technologies behind modern Data Science, and work on real data-sets and problems to put your learning into practice.



Hands-On Training

Learn modern Data Science through hands-on assignments, projects, and mentorship from your instructor. Lectures are always live. You also have access to TAs.



End-to-End, Extensive Curriculum

We'll cover the full Data Science process and the technologies to do the job, from data prep with Python libraries, to data modeling in Scikit-Learn, to visualization and presentation.



Data Science Curriculum

Python & Machine Learning

WEEK 1

Python for Data Science

Learn the Python fundamentals needed for data science.

WEEK 2

Manipulating and Understanding Data

Learn how to load, clean, and manipulate data using the Python library Pandas. Additionally, you will learn the strengths and weaknesses of using Python to manipulate data.

WEEK 3

Foundations of Data Modeling

Build visualizations to not only understand your data, but also how to communicate results to stakeholders.

WEEK 4

Statistical Inference

Learn how to use Python to implement key statistical techniques and understand statistics better by experimenting with Python on real-world datasets. This week concludes with a project to showcase your knowledge.

WEEK 5

Intro to Machine Learning

What is machine learning and why should you use the Python library Scikit-Learn for Machine Learning. Topics include types of machine learning, how to format your data to be acceptable for an algorithm, and how to train an algorithm.

WEEK 6

Decision Trees & Random Forests

Learn about tree-based machine learning algorithms, how to tune them to maximize their performance, and the strengths and weaknesses of each algorithm. Additional topics include feature selection for machine learning, and comparing machine learning algorithms.

WEEK 7

Logistic Regression and Regularization

Learn about the logistic regression algorithm and get a visual understanding of how the algorithm works. Additional topics include: logistic regression for multiclass classification, L1 and L2 regularization, and hyperparameter tuning the algorithms learned so far.

WEEK 8

Clustering Algorithms

You'll learn about a host of clustering algorithms, how to tune them, and the strengths and weaknesses of each.



Data Science Curriculum

Python & Machine Learning

WEEK 9

Dimensionality Reduction

What is dimensionality reduction. How to use it for data visualization, speed up machine learning algorithms, and understand your data better. Algorithms covered include Principal Component Analysis (PCA).

WEEK 10

Gradient Boosting Machines

Learn what gradient boosting algorithms are, why they are so performant, and how to get started with Kaggle competitions.

WEEK 11

Using SQL with Python

Working with databases is an essential part of being a data analyst, data scientist, and data engineer. This unit will cover how SQL and Python work together.

WEEK 12

Intro to Deep Learning

Learn about why deep learning has transformed industries, various deep learning frameworks, and when to use deep learning techniques. Topics include recurrent neural networks (RNN) and Convolutional Neural Networks (CNN).

WEEK 13

Database Architecture

Become familiar with entity relationship diagrams (ERD) and learn the advantages of using a relational database. Learn intermediate SQL queries to access and aggregate information.

WEEK 14

Intro to ETL

Develop an understanding of the process of extracting, transforming, and loading data.

WEEK 15

Introduction to Statistics

Learn tools for statistical analysis including measures of central tendency, variance and standard deviation and comparing means.

WEEK 16

Model Assumptions

Explore model assumptions and how to test for them. Apply this knowledge to choose the appropriate model for a data set.



Part-Time Online Cybersecurity Bootcamp

24 Weeks to a Cyber Career

Part-Time
class commitment

Career Services
Included

Learn by Doing
50-75% Lab Work

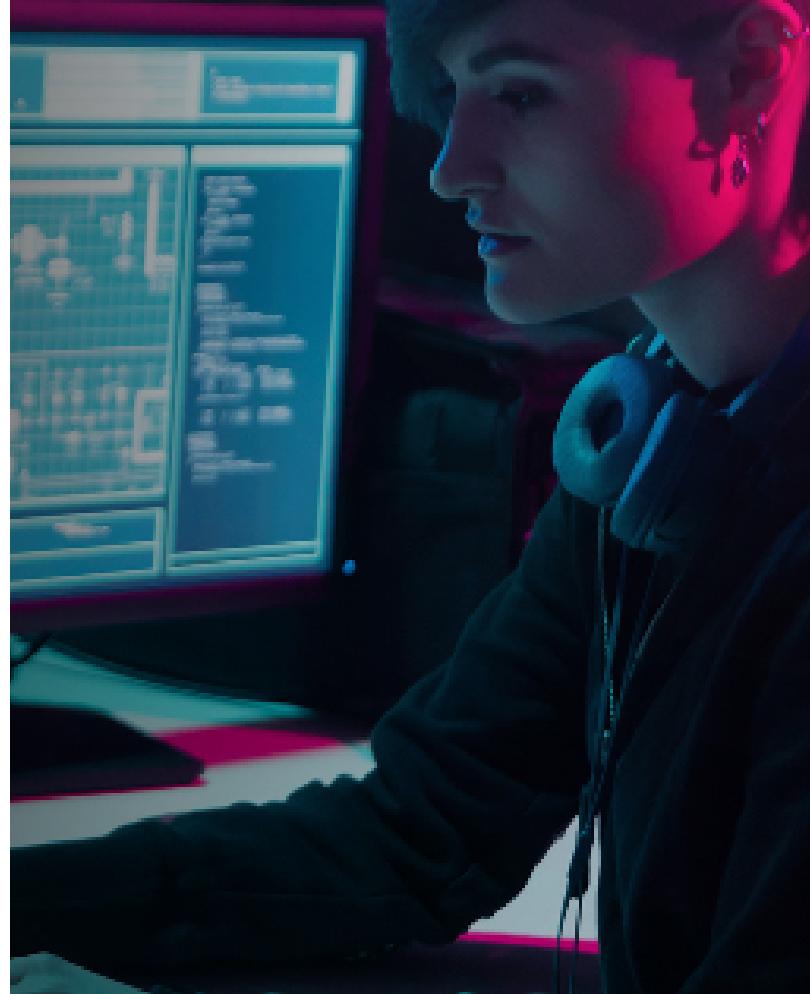
Over 6,000 alumni, hired by tech companies worldwide



*As of 2020 alumni data

Overview

The importance of cybersecurity today cannot be overstated. As our reliance on technology grows, there's a corresponding need to secure and defend networks and data against leaks, theft, and attacks. That's good news for cybersecurity specialists - the U.S. Bureau of Labor Statistics projects cybersecurity jobs will grow 31% through 2029. In short, there's job security in cybersecurity.



What You'll Get



Top Industry Certifications

Learn skills applicable to certifications such as the Network+, Linux+, Server+, Cloud+, and certified Ethical Hacker (CEH),, and receive vouchers for CompTIA Security+ and CySA+.



Learn By Doing

Gain hands-on experience with a host of popular tools such as Wireshark, Kali Linux, Metasploit, and more within a sandbox environment.



Cyber-Specific Career Services

Receive personalized career support from a dedicated cybersecurity career services manager, and keep your career service access for life.



End-to-End, Extensive Curriculum

Cover the latest real-world deployment of cybersecurity management practices, including defensive and offensive tactics, NIST Cybersecurity Framework, and event & incident management

A Professional-Grade Curriculum

From CompTIA Security+ to CySA+ certifications and beyond, our Cybersecurity program teaches students critical skills to assist in the identification, assessment, reporting, and mitigation of technology and information security risks.

This professional-grade program provides information, strategies, and tactics to identify and manage information system vulnerabilities, create effective defenses and preventative measures, and deploy countermeasures against attackers.

After completing Coding Dojo's Cybersecurity program, students are mission-ready to identify, assess, report, and mitigate technology and information security risks.

Vouchers Included!



Your Progression Plan



Week-By-Week Curriculum

Curriculum is subject to change

WEEK 1

Fundamentals of Hardware

Learn about hardware and operating system fundamentals to build your baseline.

Labs:

- Exploring the Lab Environment
- Scanning and Identifying Network Nodes
- Intercepting and Interpreting Network Traffic with Packet Sniffing
- Analyzing the Results of a Credentialated Vulnerability Scan

WEEK 2

Kali Linux Introduction

Build out your Kali Linux machine while also learning about networking and data security.

Labs:

- Managing the Life Cycle of a Certificate
- Managing Certificates with OpenSSL
- Auditing Passwords with a Password Cracking Utility
- Managing Centralized Authentication

WEEK 3

Networking & Security

Learn about network configurations and data security.

Labs:

- Installing, Using, and Blocking a Malware-based Backdoor
- Performing Network Reconnaissance and Vulnerability Scanning

WEEK 4

Cybersecurity Concepts

Enter into the cybersecurity realm by thinking about vulnerabilities and risks.

Labs:

- Managing Access Controls in Windows Server
- Configuring a System for Auditing Policies
- Managing Access Controls in Linux

WEEK 5

Virtual Machines

Get into more Windows administration tools.

Labs:

- Configuring Identity and Access Management Controls
- Implementing a Secure Network Design
- Configuring a Firewall

WEEK 6

Exploitation

Use Metasploit to run exploit code.

Labs:

- Configuring an Intrusion Detection System
- Implementing Secure Network Addressing Services
- Implementing a Virtual Private Network
- Implementing a Secure SSH Server

WEEK 7

Building a Defensible Network

Learn the basics of different network architecture and design principles.

Labs:

- Implementing Endpoint Protection
- Securing the Network Infrastructure
- Identifying Application Attack Indicators

WEEK 8

Identifying Risks to your Network

Take the necessary steps to identify the risks to your network.

Labs:

- Identifying a Browser Attack
- Implementing PowerShell Security
- Identifying Malicious Code

Week-By-Week Curriculum

Curriculum is subject to change

WEEK 9

Scanning the Network

Begin scanning a network periphery to find vulnerabilities.

Labs:

- Identifying Application Attacks
- Managing Data Sources for Incident Response
- Configuring Mitigation Controls

WEEK 10

Cloud Computing

Learn about the Cloud and its implications for risks and vulnerabilities, install Kali Linux with Cloud Computing.

Labs:

- Acquiring Digital Forensics Evidence
- Backing Up and Restoring Data in Windows and Linux
- Managing Incident Response, Mitigation and Recovery

WEEK 11

Certification Review

Tie off remaining concepts as you prepare for the CompTIA Security+ exam. Take the belt exam as your practice exam.

WEEK 12

CompTIA Security+ Certification

A crucial point in your learning timeline: schedule, review, and complete your Security+ exam.



WEEK 13

Threat Intelligence

Utilize threat intelligence and classification to support organizational security.

Labs:

- Investigating Threat Data and Intelligence Sources
- Developing a Network Threat Model
- Analyzing Output from Network Security Monitoring Tools
- Discovering the Lab Environment

WEEK 14

Intelligence & Indicator Management

Implement controls to mitigate attacks and software vulnerabilities.

Labs:

- Analyzing Output from Security Appliance Logs
- Analyzing Output from Endpoint Security Monitoring Tools

WEEK 15

Security Best Practices

Learn best practices of software, hardware, and infrastructure assurance.

Labs:

- Analyzing Email Headers
- Configuring SIEM Agents and Collectors
- Analyzing, Filtering, and Searching Event Log and syslog Output

WEEK 16

Cloud Access & Techniques

Revisit cloud technologies and how to protect your cloud-based solutions.

Labs:

- Collecting and Validating Digital Evidence
- Analyzing Network-related IoCs

Week-By-Week Curriculum

Curriculum is subject to change

WEEK 17

Attacking & Protecting

Learn about DoS attacks and mitigation as well as deploying honeypot solutions to intercept hackers.

Labs:

- Analyzing Host and Application IoCs
- Following an Incident Response Process

WEEK 18

Impact Analysis & SIEM

Implement configuration changes to existing controls to improve security based on data analysis you make part of everyday security monitoring activities.

Labs:

- Following an Incident Response Process
- Observing IoCs during a Security Incident

WEEK 19

Proactive Threat Hunting

Become proactive in your approach to cybersecurity by seeking threats.

Labs:

- Reviewing Risk Management Processes
- Analyzing Output from Topology and Host Enumeration Tools
- Testing Credential Security
- Configuring Vulnerability Scanning and Analyzing Outputs

WEEK 20

Security Monitoring

Deploy Security Content Automation Protocol (SCAP)

Labs:

- Assessing Vulnerability Scan Outputs
- Assessing the Impact of Regulation on Vulnerability Management
- Performing Account and Permissions Audits
- Configuring Network Segmentation and Security

WEEK 21

Incident Response

Apply incident response procedures

Labs:

- Configuring and Analyzing Share Permissions
- Assessing the Impact of Web Application Vulnerabilities
- Analyzing Output from Web Application Assessment Tools

WEEK 22

Digital Forensics

Collect and analyze indicators of compromise.

Labs:

- Analyzing Output from Cloud Infrastructure Assessment Tools
- Assessing the Impact of Threats to Cloud Infrastructure

WEEK 23

Certification Review

Tie off remaining concepts as you prepare for the CompTIA CySA+ exam. Take your second belt exam as prep.

WEEK 24

CompTIA CySA+ Certification

The final step in your bootcamp journey - your CySA+ certification!



Online Part-Time UI/UX Design



Become a UI/UX Pro in 24 weeks

Part-Time
class commitment

Career Services
Included

Learn by Doing
50-75% Design Work

Over 8,000 alumni, hired by tech companies worldwide

Google

amazon

f

IBM

Microsoft

in

Uber



Overview

Our UI/UX Design course is a flexible alternative for people trying to change careers or get ahead in their current roles. It features real-time support from instructors, our industry-tested learning platform, hands-on professional assignments and much more. Ideal for students who cannot commit to a full-time course, this course is designed to skill you up quickly to achieve a career in UX/UI design.

What You'll Get



Real Client Project

You will solve real UI/UX problems by working with real clients by honing your design presentation skills and how to effectively communicate your design process to your business stakeholders.



Learn By Doing

Gain hands-on experience with a host of popular design tools such as Figma, Figjam, Miro, Zoom, GDrive, Gdocs.



UI/UX Specific Career Services

Receive personalized career support from a dedicated UI/UX career services manager, and keep your career service access for life.



End-to-End, Extensive Curriculum

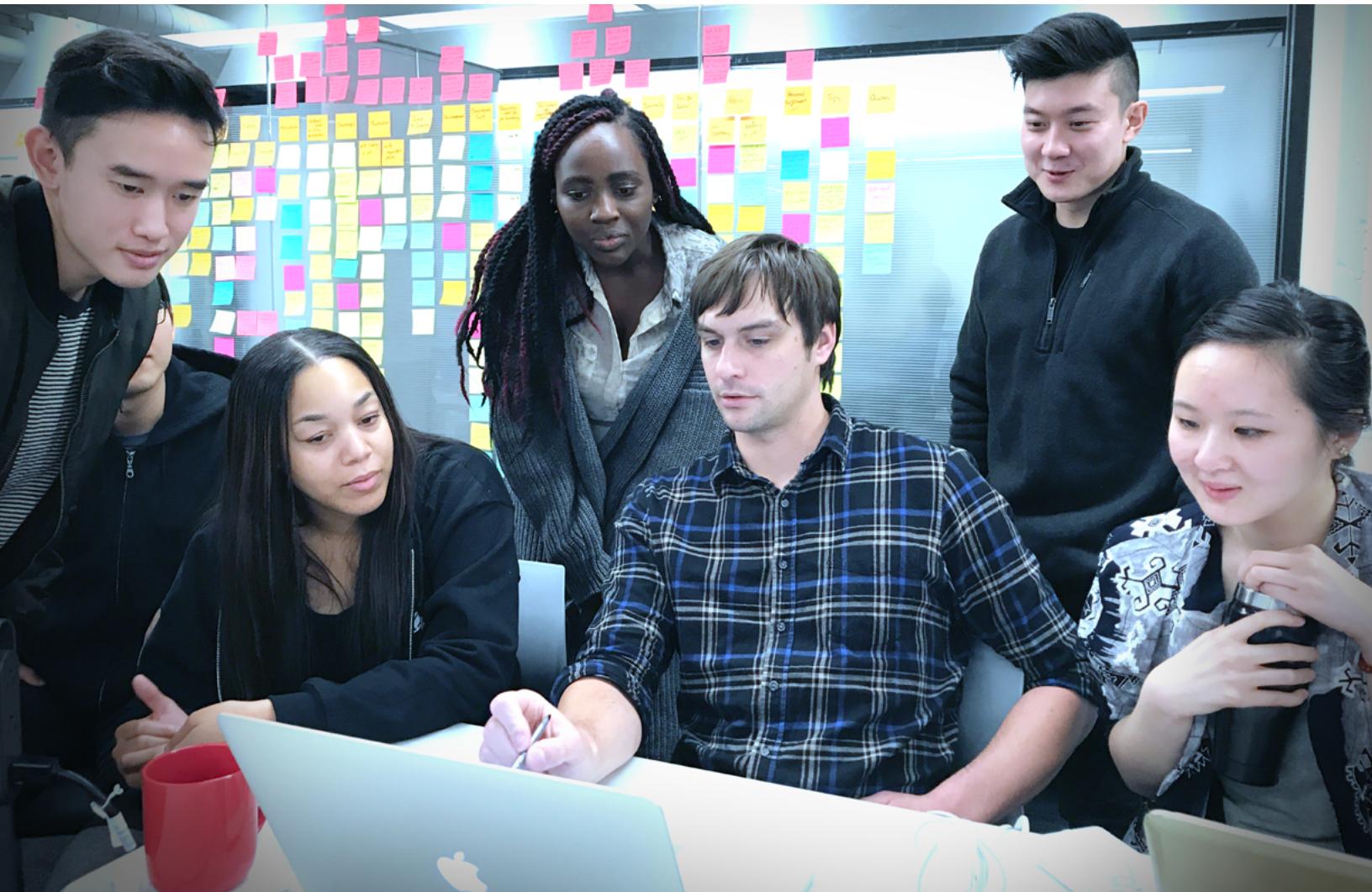
This program is divided into 3 phases. Phase 1 is where you'll learn the fundamentals of UX research, UX design, and UI design. The training wheels come off in phase 2 and 3 when you get to work on a real client project.

A Professional-Grade Curriculum

The program has been designed by industry professionals with feedback from real UI/UX designers and product designers in the field to deliver a curriculum that will give you the relevant skills necessary to be job ready upon completion.

This professional-grade program provides you with the information and knowledge necessary to design functional and beautiful digital products.

After completing Coding Dojo's UI/UX program, you'll be ready to research, concept, design, develop, and test your own digital products.



Progress through the Part-Time UI/UX Program

The program has been designed by industry professionals with feedback from real UI/UX designers and product designers in the field to deliver a curriculum that will give you the relevant skills necessary to be job ready upon completion.



Week-By-Week Curriculum

Curriculum is subject to change

WEEK 1

Project Kick-off, Discovery and Strategy

Learn about user-centered design and design thinking.

Activities:

- Welcome to Coding Dojo's UI/UX Design Program
- User-Centered Design and Design Thinking
- Creative brief, teams, and work expectations
- Feedback and Design Critique

WEEK 2

Research Planning & Implementation

Learn to problem solve through user research.

Activities:

- Understanding Research
- Research Methodology and Techniques
- Domain Research and Competitive Analysis
- Site Audit

WEEK 3

Conducting User Interviews

Learn the art of conducting user interviews and gathering insights to build empathy.

Activities:

- User Research
- Guide to User Interviews
- Sourcing Users for Research & Testing

WEEK 4

Research Data Synthesis

Practice analyzing and synthesizing the research you gathered.

Activities:

- Affinity Mapping and Research Synthesis
- Why and How We Develop Personas
- Problem Statements and Design Principles

WEEK 5

Concepting & Prototyping

Learn how to ideate and concept problem solving design solutions.

Activities:

- Ideating & Concepting with 685 Sketching
- Task Flows
- Concepts and Feature Validation
- Formative Testing vs. Summative Testing

WEEK 6

Testing, Iterating & Converging

Converge your designs after multiple rounds of testing and iterating.

Activities:

- Testing Methodologies
- Conducting usability tests
- Feature validation & converging Prototypes

WEEK 7

Communicating Your Designs

Deliver your presentations and communicate your designs effectively.

Activities:

- Creating Effective Design Presentations
- UX Design Handoff & Annotations
- Working with stakeholders

WEEK 8

Developing Your User Interface

Develop your visual design style and UI process.

Activities:

- Getting started with UI Design
- Visual Design & Best Practices
- Creating Design Mockups
- Fonts, Color in Design
- Working with Grids (Responsive Design)



Week-By-Week Curriculum

Curriculum is subject to change

WEEK 9

Translating Designs to High Fidelity

Develop your visual eye by producing beautiful high fidelity interfaces.

Activities:

- Responsive Web Design
- High Fidelity Web Designs - Round 1
- High Fidelity Web Designs - Round 2
- High Fidelity Web Prototypes

WEEK 10

High Fidelity Prototyping and Testing

Conduct usability testing of your high fidelity prototypes.

Activities:

- Usability & Desirability Testing High Fidelity Designs
- Mockups to HF Screens
- Creating HF Prototypes in Figma
- Testing Visual Designs

WEEK 11

Handoff & Design Systems

Learn how to properly handoff your design deliverables by building a design system.

Activities:

- Usability & Desirability Testing High Fidelity Designs
- Mockups to HF Screens
- Creating HF Prototypes in Figma
- Testing Visual Designs

WEEK 12

Working w/ Clients & Project Kickoff

Work with a real client on a design project that you can showcase in your portfolio.

Activities:

- Client Project process
- Communication and cadence
- Expected deliverables and scope

WEEK 13

Project Specifications, Scope and Strategy

Utilize the skills you've gained to define your project specifications and scope.

Activities:

- Getting Started with Research
- Domain & Competitive Analysis
- Client Project presentation

WEEK 14

User & Stakeholder Interviews

Conduct research with your users and client stakeholders.

Activities:

- Evaluating Users for Research & Testing
- Subject matter expert (SME) interviews
- User/SME Interview script
- Initial Synthesis and Takeaways

WEEK 15

Research Synthesis & Insights

Synthesize and analyze your research findings.

Activities:

- Synthesizing your Research Findings
- Affinity Mapping
- Creating Personas
- Problem Statement & Design Principles
- Journey Mapping

WEEK 16

Ideation & Concepting

Ideate and concept through multiple different design solutions for your client.

Activities:

- Task Flows
- Low Fidelity Concepts
- Information Architecture
- Site Map



Week-By-Week Curriculum

Curriculum is subject to change

WEEK 17

Wireframing and Prototyping

Learn how to implement mobile and web design patterns.

Activities:

- Featuring Prioritisation & Converged Design
- Mobile/Web Design Patterns
- Prototyping in Figma
- Testing your converged Prototype

WEEK 18

Testing & Communicating Insights

Present your tested UX prototype to your client.

Activities:

- Usability testing
- Creating Effective Design Presentations
- UX Design Handoff & Annotations

WEEK 19

Developing Visual Styles

Further enhance your visual design skills.

Activities:

- UI Mobile/Web Visual Design
- UI Patterns for Mobile/Web
- Visual Competitive Analysis
- Moodboards & Style Tiles

WEEK 20

High Fidelity Designs

Further develop your high fidelity design.

Activities:

- High Fidelity Screen development
- High Fidelity Prototypes
- Usability & Desirability Testing HF Designs
- High Fidelity User Testing - Research & Plan Script

WEEK 21

Final Presentations & Handoff

Communicate your visual designs effectively to your client.

Activities:

- Visual Design: Presentations
- UI Presentation Deck
- Creating a Design System

WEEK 22

Career Kickoff & Developing Your Brand

Develop your portfolio and your own personal brand.

Activities:

- Values Report review
- Personal Statement & LinkedIn/Social Media
- Case Study 1&2 Drafts
- Portfolio research + platform review

WEEK 23

Case Studies & Portfolios

Write your case studies and design your portfolio.

Activities:

- Visual design development
- Formatting assets
- Case Study Draft review 1&2
- Resume review
- Personal branding/logo

WEEK 24

Getting Job Ready

Practice your interviewing skills and get job ready.

Activities:

- MVP - Minimum Viable Portfolio
- Interview Presentations
- Final Case Studies
- Job Readiness review

How to Enroll

Explore

1

Schedule a Q&A call with

Admissions to get quick answers about the bootcamp or join the next open house.

2

Apply

Ready to join? Submit your application and pick your start date to join.

3

Complete your Interview

Schedule an interview with admissions. The interview is non-technical - no technical experience is required.

4

Deposit to Enroll

If accepted, submit your deposit to save your seat and gain access to bootcamp prep materials for your start date.

Apply Now

Financing Options

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

TALK TO US



Pay in Full

Save on tuition by paying in full upon enrollment



Installments

Spread payments over the course with standard and custom installment plans



3rd Party Financing

Finance bootcamp with a third party loan from a variety of lenders