



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

Google

amazon



cisco



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

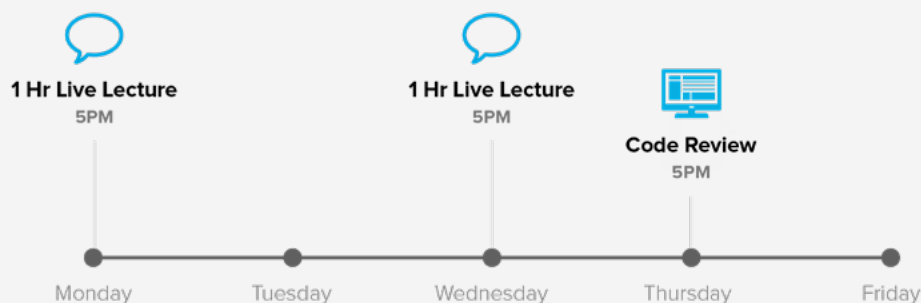


Microsoft
Virtual
Academy
★★★★★
COURSE RATING

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

Activities subject to change based on campus and curriculum



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

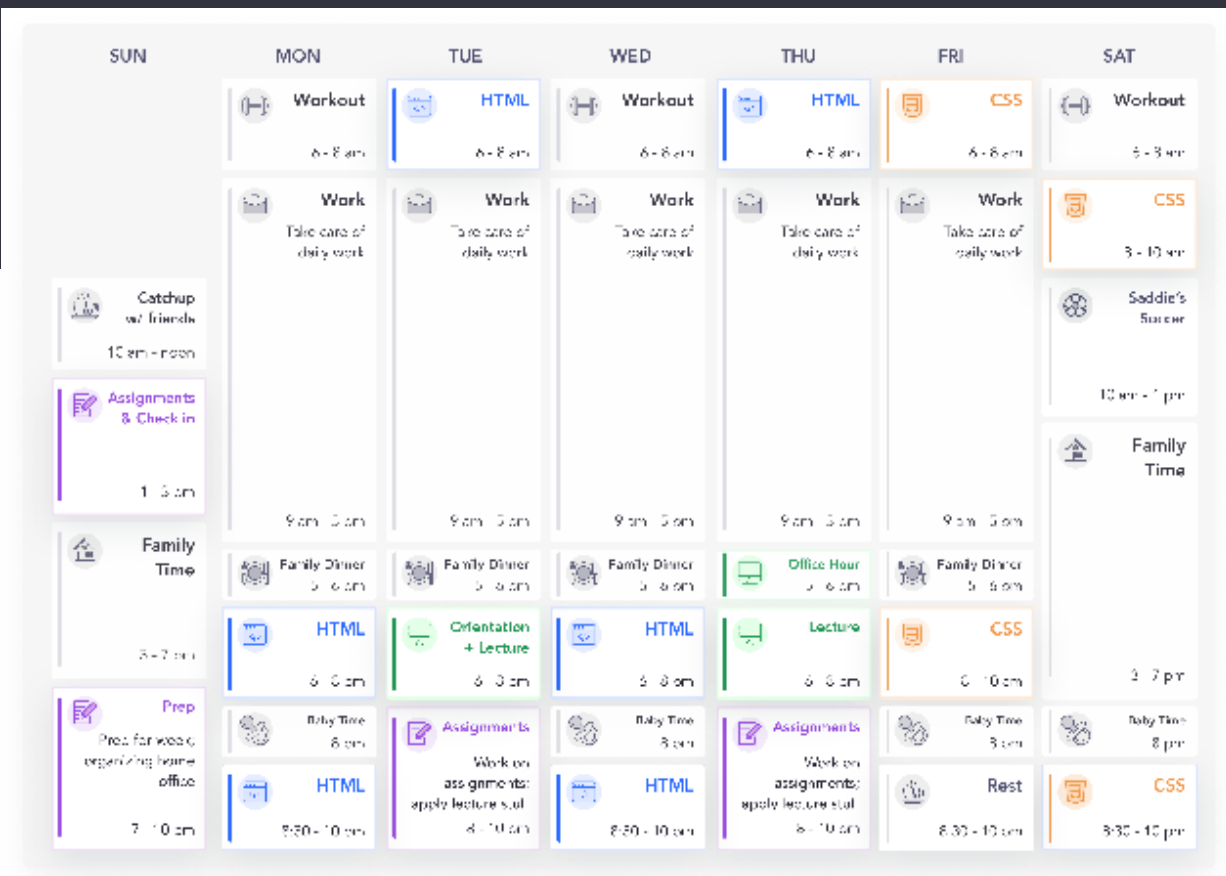


TA Support
Mon-Fri: 11:00am - 8:00pm
Sat: 8:00am - 6:00pm
Sun: 8:00am - 2:00pm

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.

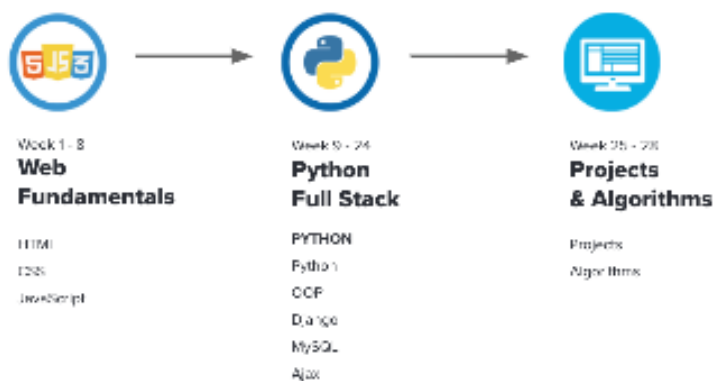
List out responsibilities and see who can help

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



Unlike the Accelerated program, you do not have a choice of stack.

You also do not have the option to add any additional stacks at this time.

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



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



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

How to Enroll

1

Explore

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