



CODE SCHOOL

AN ACCREDITED AIM INSTITUTE PROGRAM

Python Specialization

Class Description: This 80-hour class starts introducing computer languages, programming concepts, and critical thinking. Python is most commonly used in the world of Data Analysis. Upon completion of this course, individuals should have enough knowledge to help maintain certain data analysis applications.

Class Goals: The goal of this class is to introduce growing tech minds to the basics of programming and best practices. To do this we will use python, a language used by programmers, data scientists, and software engineers. Python is an advanced computer language but is good for beginners because it is said to be easy to learn and easy to write and understand syntax.

Class Objectives:

- Participants will come out of this module with foundational knowledge equivalent to that needed of a data analysis application administrator.
- Participants will build a foundation of knowledge on which to further grow programming language knowledge

Needed Materials:

- Laptop with the following minimum specs:
 - 3 years old or newer and have 2 GB of RAM (8GB or more recommended) as well as 2.5 GB available hard-disk space for installation; additional free space required during installation (cannot install on a volume that uses a case-sensitive file system or on removable flash storage devices).
- Access to portal.aimcodeschool.org
- Pens
- Paper

Grading:

- Grading will be on a pass-fail basis and determined on participation, attendance and completion of assigned course work.
- All assignment deadlines are final, no extensions will be granted.
- If a participant misses one day, it is up to him/her to reach out to the instructors for makeup assignments.
- Pass: A student must successfully complete at least 70% of the class objectives, tasks, and knowledge requirements
- Fail: a student who does not successfully complete at least 70% of the class objectives, tasks, and knowledge requirements

Assignments:

- This class is designed to give participants ample opportunity to complete all assignments in class. However, should a participant be unable to complete the assigned course work in class, it will be up to them to complete at home.

Topics Covered in This Course

1. Introduction to Backend Programming
2. Process of a python application
 - a. Web Development
 - b. Desktop Application
 - c. Software development
 - d. Education
 - e. Database access
 - f. Network programming
 - g. Game development
 - h. 3d graphics
3. Operating system / Cross platform
 - a. Windows
 - b. Mac
 - c. Linux/ Unix
 - d. Raspberry Pi...etc
4. Introduction to Python Language
 - a. Object oriented programming

-
- i. Structure
 - ii. Indentation importance
 - b. Variables
 - c. Functions
 - i. System functions
 - ii. User functions
 - d. Data Types
 - i. Formatting strings
 - e. Operators
 - 5. Strings in Depth
 - a. Multiline
 - b. Accessing string elements
 - c. Slicing
 - d. Negative indexing
 - e. String Methods
 - f. User inputs
 - 6. Lists / Arrays
 - a. Accessing / Negative Indexing
 - b. Index ranges
 - c. Changing List Items
 - d. Add / Removing from List
 - 7. Conditionals
 - a. Decision Making
 - b. Comparison Operator refresher
 - c. If statements
 - d. Elif statements
 - e. Else statements
 - f. Logical operator refresher
 - g. Membership operator refresher
 - h. Nested If statements
 - 8. Python Modules
 - a. Getpass Module
 - 9. Compound Exercises
-

10. Python While loops

- a. While loop
- b. Assignment operator refresher
- c. Break statement
- d. Escaping Characters
- e. Carriage Return
- f. PIP
 - i. Installation
- g. Continue statement
- h. Nested While loops

11. Input Validation / Error Handling

- a. Try except statement
- b. Raising exceptions
- c. System exceptions
- d. Finally statement
- e. Error handling requirements refresh

12. Python For Loops

- a. Break Statement
- b. Continue Statement
- c. range() function
- d. Nested Loops

13. Python Game Design

- a. Part one
- b. Part two
- c. Part Three

14. Python Challenges

15. Python Dictionaries

- a. Accessing Dictionary Items
- b. Adding/ removing items
- c. Looping through a dictionary
- d. copy(), dict() functions
- e. Nested dictionaries

16. Python Functions

-
- a. Calling
 - b. Parameters vs arguments
 - c. Default parameter value
 - d. Return vs print()
 - e. Arbitrary arguments (*args)
 - f. Keyword Arguments
 - g. Arbitrary Keyword Arguments(**kwargs)

17. Classes/ OOP

- a. Properties
- b. Behaviors
- c. Objects
- d. Instance Variables
- e. __init__() function
- f. Streamlines and reusable code
- g. Creating Methods

18. Finish Game Design Code

19. Final Project