Introduction to Python Programming

Instructor

Brandon Krakowsky, Lecturer at the School of Engineering and Research & Education Director at the Wharton School

Course Description

This course provides an introduction to programming and the Python language. Students are introduced to core programming concepts like data structures, conditionals, loops, variables, and functions. This course includes an overview of the various tools available for writing and running Python, and gets students coding quickly. It also provides hands-on coding exercises using commonly used data structures, writing custom functions, and reading and writing to files. This course may be more robust than some other introductory python courses, as it delves deeper into certain essential programming topics.

Course Learning Objectives

- Identify core aspects of programming and features of the Python language
- Understand and apply core programming concepts like data structures, conditionals, loops, variables, and functions
- Use different tools for writing and running Python code
- Design and write fully-functional Python programs using commonly used data structures, custom functions, and reading and writing to files

Intended Audience

This course is intended for students and professionals who have minimal or no prior programming exposure. It's for motivated learners who have experience with rigorous coursework, and are looking to gain a competitive edge in advancing their career.

Course Prerequisites

- High school or college math.
- Minimal prior programming exposure may be helpful but not needed (e.g. Computational Thinking for Problem Solving).

Penn Engineering
ONLINE LEARNING

Course Outline

Module 1: Course Introduction, Intro to Programming and The Python Language, Variables, Conditionals, Jupyter Notebook, and IDLE

- Learning Objectives
  - Identify core aspects of programming and features of the Python language
  - Use different tools for writing and running Python code
  - Understand and apply core programming concepts like conditionals and variables
  - Write code to process user input and do basic error checking
- Topics
  - Introduction to Programming
  - Introduction to Python
  - Configuring Python and Tools
  - The Python Language
  - Downloading & Installing IDLE (Python's Integrated Development & Learning Environment)
  - Python Scripts
  - Variables
  - Flow Control: Conditionals
  - Catching Errors

Module 2: Intro to Lists, Loops, and Functions

- Learning Objectives
  - Understand and apply core programming concepts like data structures, loops, and functions
  - Create a list to collect information as a sequence
  - Define custom functions with proper documentation
  - Write code to manipulate text
  - Develop a fully-functional Python program with functions to analyze numbers
- Topics
  - Data Structures: Introduction to Lists
  - Flow Control: Loops
  - 'for' Loops
  - 'while' Loops
  - Functions
  - Modular Programming

Module 3: More with Lists, Strings, Tuples, Sets, and PyCharm

- Learning Objectives
  - Use PyCharm, an industry standard IDE for writing and running Python code
  - Discover more operations with lists
  - Understand and apply data structures including tuples and sets for storing and manipulating information
  - Write more advanced code to dissect text
- Topics
  - Introduction to PyCharm
  - Data Structures: More About Lists
  - Strings
  - Data Structures: Tuples
  - Data Structures: Sets

Module 4: Dictionaries and Files

- Learning Objectives
  - Write code to read and write to files
  - Understand and apply dictionaries to manage data
  - Create fully-functional dynamic Python programs using data structures, user-defined functions, and file I/O
  - Think analytically about complex problems
- Topics
  - Data Structures: Dictionaries
  - Files

Course Assessment

This course will use a variety of assessments. Ungraded code-along videos allow students to practice along with the instructor, and self-assess their ability to apply the concepts and skills they learned, before attempting the graded assessments. Graded assessments include:

- Quizzes to check your knowledge in each module
- Programming assignments to test your level of understanding

To earn a certificate in this course, learners must earn a passing score on all assessments:

- Homework Assignments: 60% or above
- Quizzes: 75% or above

Recommended Resources

- *Python Crash Course*, by Eric Matthes:
  https://nostarch.com/pythoncrashcourse2e
  (You should also be able to find it on the O'Reilly site through the UPenn library)
- *Think Python*, by Allen B. Downey:
  https://greenteapress.com/wp/think-python/
- *Automate the Boring Stuff with Python*, by Al Sweigart:
  https://automatetheboringstuff.com/
- *Python in Easy Steps*, by Mike McGrath:
  https://www.amazon.com/Python-easy-steps-Covers-3-7/dp/1840788127/

Effort

We expect this course will take you 5-7 hrs per week to complete, for a total of 4 weeks.

Communication and Support

You can communicate with course staff and other students through the discussion forums. Please reach out to us through the discussion forum with any questions about the course content. Please allow at least 48 hours to receive a response from a TA or course staff.

Note: All communication on the discussion forum must follow the Coursera Honor Code . Never post code or solutions to assignments on the discussion forum. If you are having difficulty with code or solutions, a TA may provide an email address to send it in for private assistance.