Introduction to Python: Creating Scalable, Robust, Interactive Code

Modality: On Demand

Duration: 20 Hours

About the course:

Now that you have taken our Introduction to Python: Absolute Beginner and fundamentals course, it is time to move on to this one in order to further build your knowledge about python. This will allow you to learn how to create codes that are not only interactive but are vigorous and scalable.

You will learn the significance of introducing Python Libraries in your code. The in depth understanding of things like formatting, objects and operators, you will be able to create Python 9.py files that will have you working in the filing system. This course will teach you how to make codes with the use of robust error handling methods and how to package your python files for the purpose of running in other programs or from the terminal, and help you with the Docstring documentation.

Once this course is over, you will able to create long-lasting codes that not only can withstand errors but also are solid with documentation features. This course will equip you with skills that allow you to make code using robust data structures.

This course will also give you practical experience and you will be able to use sample code in Jupyter Notebooks on Azure.

Aims:

- Creating a code that can handle errors, try/except included
- Use of Python Libraries which includes math, os, datetime
- Being able to make precise and powerful statements
- Being able to document the code structure
- Python (.py) file configuration for the purpose of reusing in code and in terminal
- Tuple and dictionary data structures

Targeted Audience:

- System Analyst
- Programmers

Prerequisites:

You must have taken two of these courses previously

- Introduction to Python: Absolute Beginner Course
- Introduction to Python: Fundamentals Course

Course Outline:

Using Python Modules

- Import and use Math
- Working with Date and Time
- Date and Time Arithmetic
- File System
- Practice
- Required Code

powerful statements

- Compound Conditionals
- Advanced Loop Structures
- Containment Identity Precedence
- Powerful Output
- Practice
- Required Code

Methods for robust code

- Error Handling
- Files
- Tuples
- Dictionaries
- Practice
- Required Code

Durable Functions

- Command Line Scripts
- Variable Scope
- Docstring Function Documents
- Generate Pydoc Documents
- Practice
- Required Code

Final

- Required Code?