The Introduction to Python for Data Scientists course is designed to teach IT professionals how to code in Python.

The course begins by examining why to adopt Python, where it excels, and how it compares to other programming languages. It then transitions into a nuts-and-bolts examination of key language features, concepts, and functionalities and how to effectively use them to create Python applications.

Objectives

After this course, you will be able to:

- Install and configure a development environment to support Python
- Create a basic stand-alone Python application
- Perform basic text-processing functionality using Python
- Solve real world problems with Python

Prerequisites

In order to succeed in this course, you will need:

- Basic familiarity with at least one programming language
- Experience writing simple code to solve problems
- Knowledge of some fundamental programming concepts

Ineligibility Criteria

This course is **not** suitable for you if:

- You have never attempted to code in any programming language
- You have significant experience programming in Python

Length

3 days

Outline

What is Python?

- Why use Python?
- Compare Python to other programming languages

Python Development

- Download and install Python and Jupyter
- Introduction to Jupyter Notebooks

Getting Started with Python

- Variables/Typing
- Basic types: int, float, string
- Built-in Python functions
- Python arithmetic
- Strings
- Lists and tuples
- Dictionaries

Control flow

- Code Blocks/Colons
- Boolean logic
- Control structures
    - if/elif/else
    - Loops – for, while, range operator, the in operator
- Functions
    - Arguments
    - Positional arguments
    - Default arguments
    - *args and **kwargs

Programming Pythonically

- Iterables/sequences
- Slicing operator
- join() / split()
- sort() vs. sorted()
- enumerate()
- zip()
- Comprehensions
- Sets

File I/O

- Opening files in Python
- Modifying files in Python

Modules

- Modules and libraries
- The Python standard library
- Command line parsing: argparse
- System management: os, sys, subprocess
  File I/O and file management: shutil, tempfile, glob

Exception Handling with Python

- What is an exception?

- Error types: index, name, type, syntax, value, etc
- LBYL vs. EAFP
- try/except, else, finally

Regular Expressions

- Regular expression syntax and the re module
- Search and replace
- Compiling regular expression patterns