COURSE OUTLINE

PY-INP

PYTHON INTRODUCTION

DURATION	LEVEL	TECHNOLOGY	DELIVERY METHOD	TRAINING CREDITS
4 Days	Introduction	Python	Instructor Led	NA

INTRODUCTION

In this Python training course, students learn to program in Python. The course is aimed at students new to the language who may or may not have experience with other programming languages.

This Python course is taught using Python 3; however, differences between Python 2 and Python 3 are noted.

AUDIENCE PROFILE

This course is intended for students who would like to learn Python programming and may or may not have programming experience.

PREREQUISITES

Before attending this course, students must have:

- Some programming experience (recommended but not compulsory)

COURSE OBJECTIVES

- Learn how Python works and what it is good for
- Understand Python's place in the world of programming languages
- Learn to work with and manipulate strings in Python
- Learn to perform math operations with Python
- Learn to work with Python sequences: lists, arrays, dictionaries, and sets
- Learn to collect user input and output results
- Learn flow control processing in Python
- Learn to write to and read from files using Python
- Learn to write functions in Python
- Learn to handle exceptions in Python
- Learn to work with dates and times in Python

COURSE CONTENT

Module 1: Python Basics

This module explains how to get started using Python.

Lessons

- Getting Familiar with the Terminal
- Running Python
- Running a Python File
- Literals
- Variables
- Constants and Deleting Variables
- Writing a Python Module
- print () Function
- Collecting User Input
- Reading from and Writing to Files
- Exercise: Working with Files

Lab 1: Click here to enter text.

- Exercise: Hello, world!
- Exercise: Exploring Types
- Exercise: A Simple Python Script
- Exercise: Hello, You!
- Exercise: Working with Files

After completing this module, students will be able to:

- work with Python, use variables, output data, collect user input, and write simple Python functions and modules.

Module 2: Functions and Modules

You have seen some of Python’s built-in functions. In this lesson, you will learn to write your own.

Lessons

- Defining Functions
- Variable Scope
- Global Variables
- Function Parameters
- Returning Values
- Returning Values
- Importing Modules
- Methods vs. Functions

Lab 1: Exercises in this lesson

- Exercise: A Function with Parameters
- Exercise: Parameters with Default Values

After completing this module, students will be able to:

- Define functions with or without parameters
- Understand variable scope and how to import modules.

Module 3: Math

Python includes some built-in math functions and some additional built-in libraries that provide extended math (and related) functionality. In this lesson, we'll cover the built-in functions and the math and random libraries.

Lessons

- Arithmetic Operators
- Assignment Operators
- Precedence of Operations
- Built-in Math Functions
- The math Module
- The random Module

Lab 1: Exercises in this lesson

- Exercise: Floor and Modulus
- Exercise: How Many Pizzas Do We Need?
- Exercise: Dice Rolling

After completing this module, students will be able to:

- Do basic math in Python use the math and random modules for extended math functionality.

Module 4: Python Strings

This module explains how to work with Python Strings.

Lessons

- Quotation Marks and Special Characters
- String Indexing
- Slicing Strings
- Concatenation and Repetition
- Combining Concatenation and Repetition
- Python Strings are Immutable
- Common String Methods
- String Formatting
- Formatted String Literals (f-strings) (introduced in Python 3.6)
- Built-in String Functions

Lab 1: Exercises in this lesson

- Exercise: Indexing Strings
- Exercise: Slicing Strings
- Exercise: Repetition
- Exercise: Playing with Formatting
- Exercise: Outputting Tab-delimited Text

After completing this module, students will be able to:

- Manipulate and format strings.

Module 5: Iterables: Sequences, Dictionaries, and Sets

Iterables are objects that can return their members one at a time. The iterables we will cover in this lesson are lists, tuples ranges, dictionaries, and sets.

Lessons

- Definitions
- Sequences
- Lists
- Sequences and Random
- Tuples
- Ranges
- Converting Sequences to Lists
- Indexing
- Slicing
- min (), max(), and sum()
- Converting between Sequences and Strings
- Unpacking Sequences
- Dictionaries
- The len() Function
- Sets
- *args and **kwargs

Lab 1: Exercises in this lesson

- Exercise: Remove and Return Random Element
- Exercise: Simple Rock, Paper, Scissors Game
- Exercise: Creating a Dictionary from User Input

After completing this module, students will be able to:

- Understand lists, tuples, ranges, dictionaries, and sets.
- Understand the *args and **kwargs parameters.

Module 6: Virtual Environments

A virtual environment provides a self-contained directory tree with its own Python installation and additional packages necessary for the project(s) being done in that environment. As such, scripts can be run in a virtual environment that have dependencies that are different from those in other development projects that may be running in the standard environment or in separate virtual environments.

Lessons

- Packages with pip

Lab 1: Exercises in this lesson

- Exercise: Creating, Activating, Deactivating, and Deleting a Virtual Environment
- Exercise: Working with a Virtual Environment

After completing this module, students will be able to:

- Create and use virtual environments
- Install packages with pip.

Module 7: Flow Control

This module explains how to change the flow by using conditional statements and loops.

Lessons

- Conditional Statements
- Compound Conditions
- The is and is not Operators
- all () and any() and the Ternary Operator
- In Between
- Loops in Python
- break and continue
- Looping through Lines in a File
- The else Clause in Loops
- The enumerate () Function
- Generators
- List Comprehensions

Lab 1: Exercises in this lesson

- Exercise: All True and Any True
- Exercise: Word Guessing Game
- Exercise: for...else

After completing this module, students will be able to:

- Write if-elif-else conditions and to loop through sequences.
- Understand the enumerate () function, generators, and list comprehensions.

Module 8: Exception Handling

This module explains how to anticipate and handle exceptions gracefully.

Lessons

- Exception Basics
- Generic Exceptions
- The else and finally Clauses
- Using Exceptions for Flow Control
- Raising Your Own Exceptions

Lab 1: Installing and Configuring Windows 7

- Exercise: Raising Exceptions
- Exercise: Running Sum

After completing this module, students will be able to:

- Handle Python exceptions.

Module 9: Python Dates and Times

This module explains how to use Python's built-in modules to work with dates and times.

Lessons

- Understanding Time
- The time Module
- Time Structures
- Times as Strings
- Time and Formatted Strings
- Pausing Execution with time.sleep()
- The datetime Module
- datetime.datetime Objects
- datetime.timedelta Objects

Lab 1: Exercises in this lesson

- Exercise: What Colour Pants Should I Wear?
- Exercise: Report on Departure Times

After completing this module, students will be able to:

- Work with the time and datetime modules

Module 10: File Processing

This module explains how to process files.

Lessons

- Opening Files
- Writing to Files
- The os Module
- os.walk()
- The os.path Module
- A Better Way to Open Files

Lab 1: Exercises in this lesson

- Exercise: Finding Text in a File
- Exercise: Writing to Files
- Exercise: List Creator
- Exercise: Comparing Lists

After completing this module, students will be able to:

- Work with files and directories on the operating system

Module 11: PEP8 and Pylint

This module explains how to use the official Python style guide.

Lessons

- PEP8
- Pylint

After completing this module, students will be able to:

- Understand the PEP8 coding standards and how to use Pylint to analyze your code

ASSOCIATED CERTIFICATIONS & EXAM

None