Introduction to Python

Python Beginners: Level 1

Nature of the course: Theory + Practical

Total hours per day: 2 hours

Course duration: 3 weeks

Course Summary

The DTC - Python 1 course is targeted for beginners who want to learn how to think and write meaningful pieces of codes or read codes written by someone else in Python. This course teaches how to map literary description of a problem (requirement) to an application/library coded in Python. This is a core basic level course that is essential for anyone who has no prior programming experience but wishes to be a professional Android engineer in future.

Completion Criteria

After fulfilling all of the following criteria, the student will be deemed to have finished the Module:

1. Has attended 90% of all classes held
2. Has received an average grade of 80% on all assignments
3. Has received an average of 60% in assessments
4. The tutor believes the student has grasped all of the concepts and is ready to go on to the second module.

Required Text Books

1. Python Crash Course: A Hands-On, Project-Based Introduction to Programming
2. Python Programming: An Introduction to Computer Science

Prerequisites

- Basic knowledge about programming, bits/bytes, procedures, classes, computer architecture, etc. If you just have a theoretical knowledge that is perfectly okay but you should have strong convictions on what programming is, and what you hope to achieve from this class.
- Willing and eager to spend at least 10-20 hours (varying from student-to-student) per week outside of the training class to read/write codes in Python (self-study and practice).
- There is no prior educational level requirement for this course. Anyone from 10+2 student to someone who is doing their PHD in Genetic Engineering is welcome to take this course.
- If you are only interested in theory and have no interest/patience in spending at least 10 hours every week throughout the duration of the course, then this course might not be for you.
- If you have absolutely no idea about programming or do not see yourself doing programming in the next six -odd months, then this class may not be for you!

Course Details

WEEK 1

OVERVIEW OF PYTHON LANGUAGE

- Introduction
- H/w and s/w requirements
- Installation of python
- Using interpreter

CORE DATA STRUCTURES

- String, variables
- Tuples
- List
- Dictionary
- Operation on data structures
- Slicing

CONSTANT, VARIABLES AND DATA TYPES

- Primitives and non-primitives variables

WEEK 2

DECISION AND BRANCHING

- IF, ELSE, SWITCH, BREAK, CONTINUE

LOOPING

- FOR, WHILE, DO-WHILE

FUNCTIONS

- Building modules
- Functions
- Function types
- Lambdas
- Map/filter
- Comprehension
- For, while, do-while

WEEK 3

EXCEPTION HANDLING

- Introduction
- Handling exceptions
- Raising exception

- Catching exceptions
- Chaining exceptions

OOPS

- Introduction to class/objects
- Writing a class
- Inheritance
- Polymorphism
- Encapsulation
- Operator overloading
- Working with database

LABS

Lab assignments will focus on the practice and mastery of contents covered in the lectures; and introduce critical and fundamental problem-solving techniques to the students.

Intermediate Python: Level 2

Nature of the course: Theory + Practical

Total hours per day: 2 hours

Course duration: 4 weeks

Course Summary

The DTC – Python – Level 2 course is designed for students who have some prior hands-on programming experience with the Python programming language at a beginning level. This course is ideal for people who have previously programmed in another programming language (e.g., Java, Obj-C, PHP, C, C++, etc.) and wish to learn Python. This course is designed for high school and university students who want to do Python coursework, including those who have worked in the media industry since graduation or are working as a professional freelance PHP developer.

Completion Criteria

After fulfilling all of the following criteria, the student will be deemed to have finished the Module:

1. Has attended 90% of all classes held
2. Has received an average grade of 80% on all assignments
3. Has received an average of 60% in assessments
4. The tutor believes the student has grasped all of the concepts and is ready to go on to the second module.

Required Text Books

1. Python Crash Course: A Hands-On, Project-Based Introduction to Programming
2. Python Programming: An Introduction to Computer Science

Prerequisites

- Successfully complete the entrance test with score of at least 40% (for trainees directly applying to this level).
- Successfully complete the DWIT Training - Python – Level 1 course (not applicable to trainees directly applying to this level).
- Successfully complete the interview.
- Willing and eager to spend at least 10-20 hours (varying from student-to-student) per week outside of the training class to read/write codes in Python (self-study and practice).

Course Details

WEEK 1

WEB APPLICATION BASICS

- How the web works
- Overview of Django
- Django philosophies

- What we are going to build

GETTING STARTED

- Introduction of projects and apps
- Creating a basic simple web application
- Introduction to Django admin and how to use it
- How to use applications

WEEK 2

VIEWS AND TEMPLATES

- Introduction to views and templates
- How views and templates work
- Introduction to bootstrap
- Bootstrap for basic ui design
- Introduction to Django forms and how to use it

WEEK 3

MODELS

- Introduction to models
- How models work
- Using databases
- Making queries
- Database relationships
- Sessions

WEEK 4

RESTFUL API

- Introduction to rest api
- Rest api application
- Creating rest api for application
- Using rest api in ui

EXTRAS

- CONCLUSION AND DISCUSSIONS ON THE SUBJECT

LABS

Lab assignments will focus on the practice and mastery of contents covered in the lectures; and introduce critical and fundamental problem-solving techniques to the students.

Advanced Python: Level 3

Nature of the course: Theory + Practical

Total hours per day: 2 hours

Course duration: 4 weeks

Course Summary

This course expands on the DTC – Python – Level 2 foundation and offers advanced subjects to equip learners for a career as an Android software engineer.

Completion Criteria

After fulfilling all of the following criteria, the student will be deemed to have finished the Module:

1. Has attended 90% of all classes held
2. Has received an average grade of 80% on all assignments
3. Has received an average of 60% in assessments
4. The tutor believes the student has grasped all of the concepts and is ready to go on to the second module.

Required Text Books

1. Fluent Python: Clear, Concise, and Effective Programming
2. Programming Python: Powerful Object-Oriented Programming

Prerequisites

- Successfully completed the DWIT Training - Python – Level 2 or obtained at least 40% score on the entrance exam.
- The latter case applies for new students that are directly attempting this training.
- Successfully complete the interview.
- Willing and eager to spend at least 10-20 hours (varying from student-to-student) per week outside of the training class to read/write codes in Python (self-study and practice).
- Please note that this is a lab intensive course where the students will be expected to work on lab exercises for approximately half the duration of the session.

Course Details

WEEK 1

INTRODUCTION TO PYTHON FOR DATA SCIENCE

- Python Basics
- Different data types

PYTHON LISTS

- Store many different data points under a single name
- Create, subset and manipulate
- Lists in all sorts of ways

FUNCTIONS AND PACKAGES

- Importing Python packages
- Calling functions
- Numpy
- Write superfast code with Numerical Python
- Create different types of visualizations A package to efficiently store
- Calculations with huge amounts of data

WEEK 2

MATPLOTLIB

- Create different types of visualizations
- Learn how to build Complex and customized plots based on real data.

CONTROL FLOW AND PANDAS

- Write conditional constructs to tweak the execution of scripts
- The Pandas Data Frame
- The key data structure for Data Science in Python

PYTHON FOR DATA SCIENCE (ADVANCE COURSE)

- Getting Started with Data Science
- Data Science: Generating value from Data
- The Data Science Process
- Week Assignment

WEEK 3

BACKGROUND IN PYTHON AND UNIX

- Key Data Structures
- Week Assignment

JUPYTER NOTEBOOKS AND NUMPY

- Jupyter Notebooks
- Numpy Advance tutorial practice
- Satellite Image Application in numpy
- Week Assignment

PANDAS

- Working with pantaa
- Week Assignment

WEEK 4

DATA VISUALIZATION

- Introduction to Data Visualization
- Case Studies
- Matplotlib and other Libraries
- Week Assignment

INTRODUCTION TO MACHINE LEARNING

- Regression Classification
- Clustering
- Analysis
- Week Assignment

WORKING WITH TEXT AND DATABASES

- Working with Databases
- Natural Language Processing
- Working with Text
- Week Assignment

FINAL PROJECT

LABS

Lab assignments will focus on the practice and mastery of contents covered in the lectures; and introduce critical and fundamental problem-solving techniques to the students.

Learning Outcomes

- Learn how to set up Python and develop a simple application
- Declare and perform operations on simple data types, including strings, numbers, and dates
- Declare and perform operations on data structures, including lists, ranges, tuples, dictionaries, and sets
- Write conditional statements and loops
- Define and use functions, classes, and modules
- Learn how to design object-oriented programs with Python classes
- Learn how to use class inheritance in Python for reusability
- Learn how to use exception handling in Python applications for error handling.