

# Hands on Workshop on Python



**"Life's better without braces and Semicolons"**

(Bruce Eckel)

Python is among the top five programming languages worldwide. More and more programmers are migrating to Python because of its combination of simple syntax and power. The language is easy to learn and remember. You can get more things done in fewer lines of code. Moreover, you can make use of hundreds of useful libraries to solve problems quickly.

This hands-on workshop is structured around carefully planned exercises. Concepts will be introduced and then exercises will be handed out to participants. Code that is written will be immediately reviewed by the trainer and discussed with the class. The best way to learn is by doing, by making mistakes. This workshop enables participants to start coding from the outset. Slides and presentations are kept to a minimum.

## **Skills Imparted**

Python overview. Language syntax. Writing pythonic code. Hands-on learning and problem solving. Using Jupyter, IDLE or PyCharm editor. Looking up the help on core libraries.

### **Chapter 1: An Introduction to Python**

### **Chapter 2: Beginning Python Basics**

- 2.1. The print statement
- 2.2. Comments
- 2.3. Python Data Structures & Data Types
- 2.4. String Operations in Python
- 2.5. Simple Input & Output
- 2.6. Simple Output Formatting

### **Chapter 3: Python Program Flow**

- 3.1. Indentation
- 3.2. The If statement and its' related statement
- 3.3. An example with if and it's related statement

- 3.4. The while loop
- 3.5. The for loop
- 3.6. The range statement
- 3.7. Break & Continue
- 3.8. Assert
- 3.9. Examples for looping

## **Chapter 4: Functions & Modules**

- 4.1. Create your own functions
- 4.2. Functions Parameters
- 4.3. Variable Arguments
- 4.4. Scope of a Function
- 4.5. Function Documentation/Docstrings
- 4.6. Lambda Functions & map
- 4.7. An Exercise with functions
- 4.8. Create a Module
- 4.9. Standard Modules

## **Chapter 5: Exceptions**

- 5.1. Errors
- 5.2. Exception Handling with try
- 5.3. Handling Multiple Exceptions
- 5.4. Writing your own Exceptions

## **Chapter 6: File Handling**

- 6.1. File Handling Modes
- 6.2. Reading Files
- 6.3. Writing & Appending to Files
- 6.4. Handling File Exceptions
- 6.5. The with statement

## **Chapter 7: Classes In Python**

- 7.1. New Style Classes
- 7.2. Creating Classes
- 7.3. Instance Methods
- 7.4. Inheritance
- 7.5. Polymorphism
- 7.6. Exception Classes & Custom Exceptions

## **Chapter 8: Regular Expressions**

- 8.1 Simple Character Matches
- 8.2 Special Characters
- 8.3 Character Classes
- 8.4 Quantifiers
- 8.5 The Dot Character
- 8.6 Greedy Matches
- 8.7 Grouping
- 8.8 Matching at Beginning or End
- 8.9 Match Objects
- 8.10 Substituting
- 8.11 Splitting a String
- 8.12 Compiling Regular Expressions

### 8.13 Flags

## **Chapter 9: Data Structures**

### 9.1 List Comprehensions

### 9.2 Nested List Comprehensions

### 9.3 Dictionary Comprehensions

### 9.4 Functions

### 9.5 Default Parameters

### 9.6 Variable Arguments

### 9.7 Specialized Sorts

### 9.8 Iterators

### 9.9 Generators

### 9.10 The Functions any and all

### 9.11 The with Statement

### 9.12 Data Compression

## **Chapter 10: Writing GUIs in Python**

### 10.1 Introduction

### 10.2 Components and Events

### 10.3 An Example GUI

### 10.4 The root Component

### 10.5 Adding a Button

### 10.6 Entry Widgets

### 10.7 Text Widgets

### 10.8 Checkbuttons

### 10.9 Radiobuttons

### 10.10 Listboxes

### 10.11 Frames

### 10.12 Menus

### 10.13 Binding Events to Widgets

## **Chapter 11: Network Programming**

### Introduction

### 11.1 A Daytime Server

### 11.2 Clients and Servers

### 11.3 The Client Program

### 11.4 The Server Program

### 11.5 Recap

### 11.6 An Evaluation Client and Server

### 11.7 The Server Portion

### 11.8 A Threaded Server

## **Chapter 12 : Python MySQL Database Access**

### Introduction

### 12.1 Installation

### 12.2 DB Connection

### 12.3 Creating DB Table

### 12.4 INSERT, READ, UPDATE, DELETE operations

### 12.5 COMMIT & ROLLBACK operation

### 12.6 Handling Errors

# **Advanced Python on Raspberry Pi**

## **Chapter 1: An Introduction to datetime in Python**

- 1.1. date – Manipulate just date (Month, day, year)
- 1.2. time – Time independent of the day (Hour, minute, second, microsecond)
- 1.3. datetime – Combination of time and date (Month, day, year, hour, second, microsecond)
- 1.4. timedelta— A duration of time used for manipulating dates
- 1.5. tzinfo— An abstract class for dealing with time zones

## **Chapter 2: Introduction to Multithreading in Python**

- 2.1. Kernel threads
- 2.2. User-space Threads or user threads
- 2.3. thread in python
- 2.4 threading in python

## **Chapter 3: Introduction to Multiprocess in Python**

- 3.1. GIL in Python
- 3.2. Process class in python
- 3.3. Synchronization between processes
- 3.4. Using a pool of workers

## **Chapter 4: HTTP Server in Python**

- 4.1. HTTP server
- 4.2. The TCP socket address
- 4.3 Create a simple HTTP file
- 4.4. Create an HTTP web server

## **Chapter 5: Exceptions**

- 5.1. Exception handling with try and Handling Multiple Exceptions
- 5.2. try , except ,finally and default python handlers

## **Chapter 6: Introduction to Raspberry PI**

- Raspberry GPIO's
- Interfacing Sensors to RPi
- I2C and SPI communication on RPi
- Basics of Image Processing
- Introduction to OpenCV
- Interfacing Camera to RPi
- Color detection in RPi

- Sign Detection in RPi

## **Chapter 7: Accessing API -1**

- Gmail(send,receive and much more things in gmail)
- Youtube(upload,download and finding trending videos in Youtube)
- Openweather ( Extracting weather condition and forecast based on lat and lon based on API key)

## **Chapter 8: Accessing the API-2**

- **Introduction to IoT and protocols MQTT and HTTP in python API's**
- **Protocols MQTT and HTTP in python**
- Thingsboard API to push sensor data
- Adafruit API to publish and subscribe sensor

## **Chapter 9: Flask in Python**

- Running Python server in Local machine
- Accessing HTML and CSS files in flask (/)
- Flask with SQL data base

## **Chapter 10: Deep learning and Introduction to TensorFlow**

- Face detection and Face recognition using Haar cascade Algorithm
- Object detection using Tensor Flow API
- People count in Python
- Text detection in Images
- Text to Speech conversion in Python
- More examples in Python