

## Course Details

Python Basic to Advanced

**Duration:** 40 Hours

## Prerequisites

Participants need to have familiarity with any one programming language

(Optional) Participants should be conversant with some development tools - either command line or IDE

## Lab Setup

Local installation of Anaconda distribution for Python (<https://www.continuum.io/downloads>)

Internet access on participant machines to install any other required packages and for accessing data-sets

## Course Outline

### Day 1

#### Python Installation [1 hr]

Official python distribution

Anaconda

Package management

Conda

Poetry

#### Python Introduction & Basics [1.5 hrs]

The Python interpreter

Working with command-line/IDE

Python Data Types

Built in operators, functions and methods

Data and type introspection basics: type(), dir()

Syntax

- Blocks and indentation

Concepts

- Scope, lifetime
- Garbage collection

Exercises to try above concepts

#### Lists, tuples, sets [2 hr]

Defining lists

Indexes and slices

Accessing operations

Modifying operations

for(each) loops; comparison to traditional for loop

Immutability and tuples

Defining sets

Checking membership using in

#### Flow control [1.5 hrs]

For(each) loop

Absence of traditional for (i=0, i<n, i++) loop

Basic examples with foreach loop

Emulating traditional for loops using range(n)

While loops

Basic if conditions

Combining logical conditions with 'and', 'or', 'not'

Nested if's, if and for

Multi-level elif's

Match and pattern matching

### **Functions [1.5 hrs]**

def keyword

Functions without args

Functions with fixed num of args

Functions with variable number of args and default values

Returning more than one values

Keyword based args

## **Day 2**

### **File I/O [1 hrs]**

Open function

File objects and supported methods

Reading with for loop

Explicit reading with read(), readline(), readlines()

outfile.write()

Flushing output file handles

Exercises: cat, tail, head, tac, wc -l

### **String manipulation [1.5 hr]**

Str and string types

Operators and methods

Strings as immutable

Pattern Matching

Basics of Regular Expressions

're' module

Match objects

Submatches

.findall()

.subs()

### **Dictionaries [2 hr]**

Key-value pair pattern

Defining dict's

Accessing dict's

Adding/Modifying elements

Ways for Iteration

Application areas for dictionaries

## Modules [2 hrs]

What are modules?  
Pre-installed modules  
Installing new modules  
Python repository  
Pip  
Easy\_install  
Standard module library  
Sys module  
Os module

## Day 3

### Logging [2 hr]

logging module  
Creating a Logger  
Handler  
Formatter  
Filter  
Logging Levels: INFO, WARN, DEBUG, ERROR, CRITICAL, FATAL  
Custom Logging levels  
Os logging  
Multiprocess logging and synchronization

### Classes in python [1.5 hr]

\_\_init\_\_  
self  
private vs public convention  
magic functions/dunders  
object creation  
type of objects  
Operator overloading  
inheritance, multiple inheritance  
Static and class methods

### Relational Database Interaction and ORMs [2 hr]

CRUD operations  
SQL  
Python DB API 2.0  
ORM concept  
SQLAlchemy  
Declaring Models  
Querying  
Relationships  
Inserting, Deleting, Modifying

### Working with file formats [1 hr]

JSON format

Field data and formats

JSON data parsed to dict's

Modifying dict data

Writing out dict data to JSON

## Day 4

### Data Ecosystem in Python [0.5 hrs]

Scipy

Numpy

Pandas

Matplotlib

### Pandas Basics [1 hrs]

from\_csv/json/excel methods

DataFrames

Series

Inherited operations from numpy arrays

Selection and filtering

### Pandas grouping and restructuring [2 hrs]

value\_counts()

group\_by() and aggregation functions

sort\_values() and sort\_index()

pivoting/unstacking

Merging dataframes

Appending

## Web Servers/Frameworks

### Basics [1.5 hrs]

Routing

Static Files

Rendering Templates

Accessing Request Data

Redirects and Errors

About Responses

Sessions

Message Flashing

Logging

Hooking in WSGI Middlewares

Using Flask Extensions

Deploying to a Web Server

### Flask [1.5 hrs]

Implementing a simple route

Decorator syntax

Requests and responses

Jsonify

URL patterns  
Templates and rendering  
Testing webapi's with requests, postman

Day 5

### Django Basics [1 hr]

Basic Architecture  
MVC concept  
Project and Applications  
manage.py  
Running and debugging web server  
Auto reload  
Minimal Hello World app

### Basic Views [1 hrs]

Basic Views  
Routes  
HttpResponse  
get\_object\_or\_404()

### ORM and Models [1 hrs]

Object Relational Mapping  
Defining ORM models in Django  
Specifying Constraint Relationships  
Auto-increment id's  
Nullable

### Querying the Models [1.5 hrs]

Query Sets  
Field lookups  
Chaining filters  
Slicing Query Sets  
Related fields  
Q objects  
F objects

### Serialization [1 hrs]

Serialization and Deserialization  
JSONifying objects with JsonResponse  
Serializer classes  
Model Serializers  
REST APIs

### Class-based views [1 hrs]

django.views.View base class  
.get(), .post(), etc methods  
Adding to urls.py: .as\_view() method

ContextMixin  
TemplateResponseMixin  
TemplateView  
ListView and DetailsView

**Django REST Framework [2 hrs]**

Viewsets  
Routers  
Serializers  
Premission classes  
Browsable API's  
Renderers

**Summary, wrap-up, Q&A [.5 hrs]**