Python and Django

Training Program (4*10 hours)

Pre-requisites

1. Participants need to have familiarity with any one programming language
2. (Optional) Participants should be conversant with some development tools - either command line or IDE

Lab requirements

1. Local installation of Anaconda distribution for Python (https://www.continuum.io/downloads)
2. Local installation of IDE of choice (Pycharm/Visual Studio Code/..)
3. Internet access on participant machines to install any other required packages and for accessing data-sets
4. Access to following sites: https://github.com/, https://pypi.org/, https://www.anaconda.com/

Agenda

Day 1

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

Day 2

Dictionaries *[1.5 hr]*

- Key-value pair pattern
- Defining dict's
- Accessing dict's
- Adding/Modifying elements
- Ways for Iteration
- Application areas for dictionaries

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()

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

Day 3

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

Working with file formats *[2 hr]*

- CSV format
- CSV reader and DictReader
- Iterating over rows

- JSON format
- Field data and formats
- JSON data parsed to dict's
- Modifying dict data
- Writing out dict data to JSON

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 4

Classes in python *[2 hr]*

- Object Oriented concepts
- __init__
- self
- private vs public convention
- magic functions/dunders
- object creation
- type of objects
- Operator overloading
- inheritance, multiple inheritance
- Static and class methods

Modules *[1 hrs]*

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

Day 5

Web Development Basics *[1.5 hrs]*

- Routing
- Static Files
- Rendering Templates
- Accessing Request Data
- Redirects and Errors
- About Responses
- Sessions
- Logging
- Hooking in WSGI Middlewares
- Extensions, middleware and plugins
- Deploying to a Web Server

Web development with Django *[2 hr]*

- Introduction
- Django-admin command manage.py
- Views and URLconfs
- Templates
- Models
- The Django Admin site
- A ‘Hello world’ demo
- Views and URLconfs

Day 6

Django Architecture *[1 hrs]*

- Sites and apps
- Shared configuration
- Minimal Django layout
- Middleware
- Built-in flexibility

Configuring a Project *[1 hrs]*

- Executing manage.py

- Starting the project
- Database setup
- The development server

Adding an Application *[1 hrs]*
- Generate the application files
- Defining models
- Related objects
- SQL Migration
- App configuration
- Accessing models

Basic Views *[1 hrs]*
- What is a view
- HttpResponse
- URL route configuration
- Shortcut: get_object_or_404()

Day 7

REST APIs *[1.5 hrs]*
- URL's
- HTTP verbs
    - GET: Retrieve data.
    - POST: Create new resources.
    - PUT/PATCH: Update existing resources.
    - DELETE: Remove resources.
- Resources
- Statelessness
- JSON/XML transport

Templates *[2 hrs]*
- The need for templates
- Templating language
- Variable substitution blocks {{ }}
- Code execution blocks {% %}
- Supported python constructs in templating
- Rendering
- Safety and escaping

Day 8

ORM and Models *[2 hrs]*

- Object Relational Mapping Concept
- Equivalance between classes and DB tables
- Defining ORM models in Django
- Making migrations and Migrating
- Checking generated SQL
- Specifying Constraint Relationships
- Auto-increment id's
- Nullable

Querying the Models *[2 hrs]*

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

Day 9

Serialization *[1 hrs]*

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

Class-based views *[2 hrs]*

- django.views.View base class
- .get(), .post(), etc methods
- Adding to urls.py: .as_view() method
- ContextMixin
- TemplateResponseMixin
- TemplateView
- ListView and DetailsView

Advanced features/Way forward with Django *[1 hr]*

- Django Admin Site configuration
- Django Rest Framework (DRF)
- Authentication with Auth Middleware
- Contrib.messages
- Signals
- Swagger API's

Day 10

Data Ecosystem in python *[2.5 hrs]*

- Numpy
- Pandas
- Pandas basic operations for data manipulation
- Visualization with matplotlib & seaborn
- Jupyter Notebooks
- Machine Learning Basics
- Demo of Binary classification Deep Learning

Python development best practices *[1 hr]*

- Naming conventions
- Type annotating
- Documentation and docgen
- Linting and quality checks
- Formatting

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