

# Course Name: Practical Python Programing

Pre-requisites: *Basic Computer knowledge. Understanding of how computer works. Some hands-on programing knowledge will be very helpful.*

Length: 12 Weeks (60 hours).

## Course Outline:

### Week 1: Python Language Basics

- a) Why Python? The landscape. Of Application/System programing languages.
- b) Setup the development environment.
- c) "Hello World!" of Python.
- d) Comments And Pound Characters.
- e) Numbers And Math.
- f) Variables And Names.
- g) More Variables And Printing.
- h) Strings And Text
- i) More Printing
- j) What Was That?

### Week 2: Python Programing - Interactive and I/O

- a) Asking Questions.
- b) Prompting People.
- c) Parameters, Unpacking, Variables.
- d) Prompting And Passing.
- e) Reading Files.
- f) Reading And Writing Files.
- g) More Files.
- h) Names, Variables, Code, Functions.
- i) Functions And Variables.
- j) Functions And Files.
- k) Functions Can Return Something.
- l) Take home test.

### Week 3: Python Programing – Logic, List and more

- a) Memorizing Logic
- b) Boolean Practice
- c) What If
- d) Else And If
- e) Making Decisions
- f) Loops And Lists
- g) While Loops

- h) Accessing Elements Of Lists
- i) Branches and Functions
- j) Designing and Debugging
- k) Symbol Review
- l) Doing Things To Lists
- m) Dictionaries, Oh Lovely Dictionaries

## **Week 4: Python Programing – Modules, Classes and Objects**

- a) Modules, Classes, And Objects.
- b) Learning To Speak Object Oriented.
- c) Is-A, Has-A, Objects, and Classes.
- d) Inheritance Vs. Composition.
- e) A Project Skeleton.
- f) Automated Testing.
- g) Advanced User Input.
- h) A Practical Example.
- i) Another Practical Example.

## **Week 5: House Keeping Work and Django Framework**

- a) Python 3 vs. Python 2.7.
- b) Common Python Library and Package Management.
- c) Introduction to PyCharm Python Development Tools.
- d) Web Development Introduction.
- e) Introduction to Python based Web Development Framework.
- f) Introduction to Django.
- g) What we want to accomplish om the next 4 weeks?
- h) Installation libraries for the project.
- i) Virtual Environment
- j) Django Basics
  - a) Creating your Django Project
  - b) Creating a Django Application
  - c) Creating a View
  - d) Mapping URLs
  - e) Basic Workflows
    - a) Creating a new Django Project
    - b) Creating a new Django application

## **Week 6: Adding Visual to the Site and Presenting the Data**

- a) Templates and Static Media
  - 1. Using Templates
    - a) Configuring the Templates Directory
    - b) Dynamic Paths
    - c) Adding a Template
  - 2. Serving Static Media

1. Configuring the Static Media Directory
3. Static Media Files and Templates
4. The Static Media Server
5. Basic Workflow
- b) Models and Databases
  1. Project Requirements in detail
  2. Telling Django About Your Database
  3. Creating Models
  4. Creating and Migrating the Database
    - a) Setup Database and Create Superuser
    - b) Creating / Updating Models / Tables
  5. Django Models and the Django Shell
  6. Configuring the Admin Interface
  7. Creating a Population Script
  8. Basic Workflows
    - a) Setting up your Database
    - b) Adding a Model

## **Week 7: Connecting the Data to Website**

- a) Models, Templates and Views
  1. Basic Workflow: Data Driven Pages
  2. Showing Categories on Rango's Homepage
    - a) Importing Required Models
    - b) Modifying the Index View
    - c) Modifying the Index Template
  3. Creating a Details Page
    - a) URL Design and Mapping
    - b) Update Category Table with Slug Field
    - c) Category Page Workflow
    - d) Category View
    - e) Category Template
    - f) Parameterised URL Mapping
    - g) Modifying the Index Template
    - h) Demo.
- b) Fun with Forms
  1. Basic Workflow
  2. Page and Category Forms
    - a) Creating ModelForm Classes
    - b) Creating an Add Category View
    - c) Creating the Add Category Template
    - d) Mapping the Add Category View
    - e) Modifying the Index Page View
    - f) Demo
    - g) Cleaner Forms

## Week 8: Users and Templates

- a) User Authentication
  - 1. Setting up Authentication
  - 2. The User Model
  - 3. Additional User Attributes
  - 4. Creating a User Registration View and Template
    - a) Creating the UserForm and UserProfileForm
    - b) Creating the register() View
    - c) Creating the Registration Template
    - d) The register() View URL Mapping
    - e) Linking Together
    - f) Demo
  - 5. Adding Login Functionality
    - a) Creating the login() View
    - b) Creating a Login Template
    - c) Mapping the Login View to a URL
    - d) Linking Together
    - e) Demo
  - 6. Restricting Access
    - a) Restricting Access with a Decorator
  - 7. Logging Out
- b) Working with Templates
  - 1. Reoccurring HTML and The Base Template
  - 2. Template Blocks
    - a) Abstracting Further
  - 3. Template Inheritance
  - 4. Referring to URLs in Templates
  - 5. Exercises

## Week 9: Sessions and More Users

- a) Cookies and Sessions
  - 1. Cookies, Cookies Everywhere!
  - 2. Sessions and the Stateless Protocol
  - 3. Setting up Sessions in Django
  - 4. A Cookie Tasting Session
    - a) Testing Cookie Functionality
  - 5. Client Side Cookies: A Site Counter Example
  - 6. Session Data
  - 7. Browser-Length and Persistent Sessions
  - 8. Clearing the Sessions Database
  - 9. Basic Considerations and Workflow
- b) User Authentication with Django-Registration-Redux
  - 1. Setting up Django Registration Redux
  - 2. Functionality and URL mapping
  - 3. Setting up the Templates
    - a) Login Template

- b) Registration Template
- c) Registration Complete Template
- d) Logout Template
- e) Try out the Registration Process
- f) Refactoring your project
- g) Modifying the Registration Flow

## **Week 10: Make the Site Pretty**

- a) Bootstrapping the Site
  - 1. The New Base Template
  - 2. Quick Style Change
    - a) The Index Page
  - 3. The Login Page
    - a) Other Form-based Templates
  - 4. The Registration Template
  - 5. Using Django-Bootstrap-Toolkit
  - 6. The End Result
- b) Template Tags
  - 1. Providing Categories on Every Page
  - 2. Using Template Tags
  - 3. Parameterised Template Tags

## **Week 11: Adding the Internet Search to site and More**

- a) Adding External Search Functionality
  - 1. The Bing Search API
    - a) Registering for a Bing API Key
  - 2. Adding Search Functionality
  - 3. Storing your API KEY safely
  - 4. Exercises
  - 5. Putting Search into the Site
    - a) Adding a Search Template
    - b) Adding the View
- b) Site Exercises
  - 1. Track Page Click Throughs
    - a) Hint
  - 2. Searching Within a Category Page
  - 3. Create and View Profiles
- c) JQuery and Django
  - 1. Including JQuery in your Django Project/Application
  - 2. Stylistic Note
  - 3. Example Popup Box on Click
  - 4. Selectors
  - 5. DOM Manipulation Example
- a) AJAX in Django with JQuery
  - 1. AJAX based Functionality
  - 2. Add a “Like Button”

- a) Workflow
- b) Updating Category Template
- c) Create a Like Category View
- d) Making the AJAX request
- 3. Adding Inline Category Suggestions
  - a) Workflow
  - b) Parameterise the Get Category List Function
  - c) Create a Suggest Category View
  - d) Map View to URL
  - e) Update Base Template
  - f) Add AJAX to Request Suggestions

## **Week 12: Testing and Deployment**

- a) Automated Testing
  - 1. Running Tests
  - 2. Testing the models in Rango
  - 3. Testing Views
  - 4. Testing the Rendered Page
  - 5. Coverage Testing
  - 6. Exercises
- b) Deploying Your Project
  - 1. Creating a PythonAnywhere Account
  - 2. The PythonAnywhere Web Interface
  - 3. Setting up your Web Application
    - a) Configuring the WSGI Script and its virtual environment
    - b) Assigning Static Paths
    - c) Bing API Key
    - d) Turning off DEBUG Mode
  - 4. Log Files
- a) Q&As