Python with Django training

COURSE BROCHURE & SYLLABUS

Why Apponix for Python

- 100% student satisfaction
- Experienced and expert trainers in Python and its trending technological changes.
- Provides individual laptop access with high speed Wi-Fi network
- Facilitated lab and learner-friendly infrastructure.
- Air-conditioned classrooms.
- Coaching is designed so as to meet the upcoming opportunities and technological needs.
- Prepares the students to meet impending interviews and tests with a professional profile and immense confidence.

Python with Django course objectives

- Understand the core construction of Python
- Realise the status of Python as the best scripting language
- Explain the structure and applications of python
- Learn the objective types of Python like Lists, Tuples, Strings, Dictionaries etc.
- Knowing how Python as the best object-oriented programming language
- Understanding the fundamental concepts like Flow control and conditions, File handling, OOPs and Python modules.
- Learn how to handle exception and error
- Understand and create maps and websites in Django
- Handle Django models, REST framework, AJAX and DjangojQuey for creating websites and its other applications.
- Identify Django template system
- Learn the required processes to function class inheritance that helps in reusability
- Indexing and slicing of data in python.

Course Syllabus

Python Course Syllabus

SECTION - I

1. Overview

- Why do we need Python?
- Program structure

2. Environment Setup

- Python Installation
- Execution Types
- What is an interpreter?
- Interpreters vs Compilers

- Using the Python Interpreter
- Interactive Mode
- Running python files
- Working with Python shell
- Integrated Development Environments (IDES)
- Interactive Mode Programming
- Script Mode Programming

3. Basic Concepts

- Basic Operators
- Types of Operator
- Python Arithmetic Operators
- Python Comparison Operators
- Python Assignment Operators
- Python Bitwise Operators
- Python Logical Operators
- Python Membership Operators (in, not in)
- Python Identity Operators (is, is not)
- Python Operators Precedence

4. Basic Concepts

- **Data Types**
- Variables
- Assigning Values to Variables
- Multiple Assignment
- Python Numbers
- Python Strings
- Accessing Values in Strings
- String Special Operators
- String Formatting Operator
- Triple Quotes
- Built-in String Operations
- **Python Lists**
- Accessing Values in Lists
- Updating Lists
- Delete List Elements
- Basic List Operations
- Indexing, Slicing, and Matrixes
- Built-in List Functions & Methods

- **Python Tuples**
- Accessing Values in Tuples
- Updating Tuples
- Delete Tuple Elements
- Basic Tuples Operations
- Indexing, Slicing, and Matrixes
- No Enclosing Delimiters
- Built-in Tuple Functions
- **Python Dictionary**
- Accessing Values in Dictionary
- Updating Dictionary
- Delete Dictionary Elements
- Properties of Dictionary Keys
- Built-in Dictionary Functions & Methods

5. Basic Operators in Python

- Types of Operator
- Python Arithmetic Operators
- Python Comparison Operators
- Python Assignment Operators
- Python Bitwise Operators
- Python Logical Operators
- Python Membership Operators (in, not in)
- Python Identity Operators (is, is not)
- Python Operators Precedence

6. Loops and Decision Making

- if statements
- ..else statements
- nested if statements
- while loop
- for loop
- nested loops
- Loop Control Statements
- 1) break statement
- 2) continue statement
- 3) pass statement

SECTION - II

1. Functions

- Defining a Function
- Syntax
- Calling a Function
- Pass by reference vs value
- Function Arguments
- Required arguments
- Keyword arguments
- Default arguments
- Variable-length arguments
- The return Statement
- Scope of Variables
- Global vs. Local variables

2. Python Modules and Packages

- Framework vs Packages
- Folium Introduction
- Why are modules used?
- Creating modules
- The import Statement
- The from...import Statement
- The from...import * Statement
- Locating Modules
- The PYTHONPATH Variable
- Namespaces and Scoping
- The dir( ) Function
- The globals() and locals() Functions
- The reload() Function
- Packages in Python

3. Basic OOPs Concept

- Creating class in Python
- Documented String
- Private Identifier
- Constructor
- Inheritance

- Polymorphism

4. Decorator, Iterator and Generator

5. Anonymous Function

- Lambda
- Map
- Filter
- Reduce

SECTION -III

1. File Manipulation

- Opening Text File
- Working with a File on Python
- The open function
- File modes
- The file object attributes
- close() method
- write() method
- read() method
- Files: Input
- Files: Output
- Reading files
- Renaming & deleting files
- Writing into a file
- remove() method

2. Python GUI

- Basic Operations using Tkinter
- Buttons and Textbox
- Menu Bar
- Message Box and Radio Button
- Checkbox and Event Creating
- Creating Application in GUI

3. SQL and Python

- Overview of SQLite
- Integrating Python with SQLite

4. NoSQL and Python

- Overview of NoSQL
- Integrating Python with NoSQL

5. Project Demonstration Tkinter with SQL

- **Project Demonstration Tkinter with SQL**

6. Other Concepts

- Errors and Exception Handling
- Standard exceptions
- Assertions in Python
- The assert Statement
- What is Exception?
- Handling an exception
- Syntax
- The except Clause with No Exceptions
- The except Clause with Multiple Exceptions
- The try-finally Clause
- Argument of an Exception
- Example with Tkinter Application
- Regular Expression
- Powerful Utilities
- Multithreading
- List Comprehensive
- Shallow and Deep Copy
- Unit testing

7. Advanced Concept -- Overviews

- Networking Overview
- Sending and Receiving Email by Python
- Basics of Pandas and Numpy
- How to use Anaconda
- How to create dashboard
- Overview of Django

Django Course Syllabus

SECTION - I

1. Introduction to Django

- What Is a Web Framework?
- The MVC Design Pattern
- Django's History

2. Installation of Django

- Installing Python
- Installing Django
- Setting Up a Database
- Starting a Project.
- The Development Server
- Django Commands Overview

3. The Basics of Dynamic Web Pages

- Your First View: Dynamic Content
- Mapping URLs to Views
- How Django Processes a Request
- URL configurations and Loose Coupling
- 404 Errors
- Your Second View: Dynamic URLs
- A Word About Pretty URLs
- Wildcard URL patterns
- Django's Pretty Error Pages

4. The Django Template System

- Template System Basics
- Using the Template System
- Creating Template Objects
- Rendering a Template
- Multiple Contexts, Same Template
- Context Variable Lookup
- Playing with Context Objects
- Basic Template Tags and Filters Tags
- Philosophies and Limitations

- Using Templates in Views
- Template Loading
- render_to_response()
- The locals() Trick
- Subdirectories in get_template()
- The include Template Tag.
- Template Inheritance

5. Interacting with a Database: Models

- The “Dumb” Way to Do Database Queries in Views
- The MTV Development Pattern
- Configuring the Database
- Your First App
- Defining Models in Python
- Your First Model
- Installing the Model
- Basic Data Access
- Adding Model String Representations
- Inserting and Updating Data
- Selecting Objects
- Filtering Data
- Retrieving Single Objects
- Ordering Data
- Chaining Lookups
- Slicing Data
- Deleting Objects
- Making Changes to a Database Schema
- Adding Fields
- Removing Fields
- Removing Many to Many Fields
- Removing Models

6. The Django Administration Site

- Activating the Admin Interface
- Using the Admin Interface
- Users, Groups and Permissions
- Customizing the Admin Interface
- Customizing the Admin Interface’s Look and Feel
- Customizing the Admin Index Page

- When and Why to Use the Admin Interface

SECTION - II

7. Form Processing

- Search
- The “Perfect Form”
- Creating a Feedback Form
- Processing the Submission
- Custom Validation Rules
- A Custom Look and Feel
- Creating Forms from Models

8. Advanced Views and URL configurations

- URL configuration Tricks.
- Streamlining Function Imports
- Using Multiple View Prefixes
- Special-Casing URLs in Debug Mode
- Using Named Groups
- Understanding the Matching/Grouping Algorithm
- Passing Extra Options to View Functions
- Using Default View Arguments
- Special-Casing Views
- Capturing Text in URLs
- Determining What the URL configuration Searches Against
- Including Other URL configurations
- How Captured Parameters Work with include()
- How Extra URL configurations Options Work with include()
- Section II : Django Sub Framework

9. Generic Views

- Using Generic Views
- Generic Views of Objects
- Extending Generic Views
- Making “Friendly” Template Contexts
- Adding Extra Context
- Viewing Subsets of Objects
- Complex Filtering with Wrapper Functions
- Performing Extra Work

10. Extending the Template Engine

- Template Language Review
- Request Context and Context Processors
- django.core.context_processors.auth
- django.core.context_processors.debug
- django.core.context_processors.i18n
- django.core.context_processors.request
- Guidelines for Writing Your Own Context Processors
- Inside Template Loading
- Extending the Template System
- Creating a Template Library
- Writing Custom Template Filters
- Writing Custom Template Tags
- Shortcut for Simple Tags
- Inclusion Tags
- Writing Custom Template Loaders
- Using the Built-in Template Reference
- Configuring the Template System in Standalone Mode

11. Generating Non-HTML Content

- The Basics: Views and MIME Types
- Producing CSV
- Generating PDFs
- Installing Report Lab
- Writing Your View
- Complex PDFs
- Other Possibilities
- The Syndication Feed Framework
- Initialization
- A Simple Feed
- A More Complex Feed
- Specifying the Type of Feed
- Enclosures
- Language
- URLs
- Publishing Atom and RSS Feeds in Tandem
- The Sitemap Framework
- Installation
- Initialization

- Sitemap Classes
- Shortcuts
- Creating a Sitemap Index.
- Pinging Google

12.Sessions, Users and Registration

- Cookies
- Getting and Setting Cookies
- The Mixed Blessing of Cookies
- Django’s Session Framework
- Enabling Sessions
- Using Sessions in Views
- Setting Test Cookies
- Using Sessions Outside of Views
- When Sessions Are Saved
- Browser-Length Sessions vs Persistent Sessions
- Other Session Settings
- Users and Authentication
- Enabling Authentication Support
- Using Users
- Logging In and Out
- Limiting Access to Logged-in Users
- Limiting Access to Users Who Pass a Test
- Managing Users, Permissions and Groups
- Using Authentication Data in Templates
- The Other Bits: Permissions, Groups, Messages and Profiles
- Permissions
- Groups
- Messages
- Profiles

Trainer Profile

- 15 years of experience with Object Oriented Programming in web domain.
- Advanced knowledge in python language and its features.
- 1500+ students were trained with 100% satisfaction.
- Well-informed in latest technologies and web application development
- Expert in python frameworks like Django, Flask etc. and excellent delivering skills
- Aware of Object Relational Mapper (ORM) libraries
- Certified trainer on PMR
- Experienced in ethical hacking, coding practice, Power DB and APP etc.

Instructor Experience

- Performance applications of low-latency and high-availability are designed and implemented.
- Writes competent codes that are reusable and testable.
- For the improvement of responsiveness and complete performance, back-end components are developed.
- Integrates databases, blob stores, key-value stores and other data storage solutions.
- Improves the functionality of prevailing systems
- Implementation of data protection solutions and robust security.
- User-facing elements are integrated into applications.