

# Full Stack with Python Django Training

## Django Training Overview

Django is an open-source web application framework which was written in Python Language. Django was mainly developed to solve the complex database-driven websites and to create web application framework with Django functionality.

## Objectives of the Course

- To gain enough knowledge on Python Language
- Usage of Django template system
- Defining Database Models in Django
- To create applications to solve problems and their need

## Pre-requisites

- Basic Programming knowledge and HTML is enough

## Who should do the course

- Full Stack Developers
- Web Developers
- Technical Project Managers
- Aspirants those who are willing to learn Django

## Dajngo Course Content

### Introduction to WEB

- What is Web?
- Web Features?
- W3C and W3C Members
- Introduction to What WG

### Core HTML

- Introduction
- Parts in HTML Document
- Version Information
- Head Section
- Meta Information
- Favicons
- Body Section
- HTML FORMS
- Anchors, Images

## **Advance HTML5**

- Introduction
- HTML5 HISTORY
- Why HTML5?
- New Features and Groups
- Structure of HTML5 Document
- Power of HTML5 and Features
- Semantics and Block Level Elements
- HTML5 Forms
- HTML5 Multimedia

## **Core CSS**

- Introduction
- CSS Basics
- CSS Introduction
- CSS Syntax
- CSS Versions
- CSS Id & Class
- CSS Styling
- Styling Backgrounds
- Styling Text
- Styling Fonts
- CSS Borders

## **Core JavaScript**

- What is Script? Types of Scripts?
- Introduction to JavaScript
- Comments and Types of Comments
- Popup Boxes
- Variables & Operators
- JavaScript Functions and Events
- Conditional Statements
- Looping Control Statement

## **Core Python**

- Introduction to Python
- What is Python?
- History of Python
- Python Versions
- Features of Python
- How to Install Python
- Install Python with Diff IDEs
- Creating Your First Python Program
- Printing to the Screen
- Reading Keyboard Input
- Using Command Prompt and GUI or IDE

## **Different Modes in Python**

- Execute the Script
- Interactive Mode
- Script Mode
- Python Comments
- Working with Python in Unix/Linux/Windows/Mac/Android
- Python New IDEs
- PyCharm IDE
- How to Work on PyCharm
- PyCharm Components
- Debugging process in PyCharm
- SublimeText IDE
- What is PIP?

## **Variables in Python**

- What is Variable?
- Variables in Python
- Constants in Python
- Standard Data Types
- Operators and Operands
- Swap variables
- Type Conversion
- String Handling

## **Python Conditional Statements**

- How to use "if condition" in conditional structures
- if statement (One-Way Decisions)
- if .. else statement (Two-way Decisions)
- How to use "else condition"
- if ..elif .. else statement (Multi-way)
- When "else condition" does not work
- How to use "elif" condition
- How to execute conditional statement with minimal code
- Nested IF Statement

## **Python LOOPS**

- How to use "While Loop"
- How to use "For Loop"
- How to use For Loop for set of other things besides numbers
- Break statements in For Loop
- Continue statement in For Loop
- Enumerate function for For Loop

## **Python Lists**

- Lists are mutable
- Getting to Lists

- List indices
- Traversing a list
- List operations
- List slices
- List methods
- Map, filter and reduce

## **Python TUPLE**

- Advantages of Tuple over List
- Packing and Unpacking
- Comparing tuples
- Creating nested tuple
- Using tuples as keys in dictionaries
- Deleting Tuples
- Slicing of Tuple
- Tuple Membership Test

## **Python Sets**

- How to create a set?
- Iteration Over Sets
- Python Set Methods
- Python Set Operations
- Union of sets
- Built-in Functions with Set
- Python Frozenset

## **Python Dictionary**

- How to create a dictionary?
- Python Hashing?
- Python Dictionary Methods
- Copying dictionary
- Updating Dictionary
- Delete Keys from the dictionary
- Dictionary items() Method
- Sorting the Dictionary
- Python Dictionary in-built Functions

## **Python Functions**

- What is a Function?
- How to define and call a function in Python
- Types of Functions
- Significance of Indentation (Space) in Python
- How Function Return Value?
- Types of Arguments in Functions
- Default Arguments
- Non-Default Arguments
- Keyword Arguments

- Non-keyword Arguments
- Arbitrary Arguments
- Rules to define a function in Python
- Various Forms of Function Arguments
- Scope and Lifetime of variables
- Anonymous Functions/Lambda functions
- map(), filter(), reduce() functions
- What is a Docstring?

## **Advanced Python**

- Python Exception Handling
- Python Errors
- Common RunTime Errors in PYTHON
- Abnormal termination
- Chain of importance Of Exception
- Exception Handling
- Try ... Except
- Try ..Except .. else
- Try ... finally

## **Python Class and Objects**

- Introduction to OOPs Programming
- Object Oriented Programming System
- OOPS Principles
- Basic concept of Object and Classes
- Access Modifiers
- How to define Python classes
- Self-variable in python
- What is Inheritance? Types of Inheritance?
- How Inheritance works?

## **Python Regular Expressions**

- What is Regular Expression?
- Regular Expression Syntax
- Understanding Regular Expressions
- Regular Expression Patterns
- Literal characters

## **Django Web Framework**

- What is a Framework
- Introduction to Django
- Django – Design Philosophies
- History of Django
- Why django and Features
- Environment setup
- Web Server

## **MVC Pattern**

- MVC Architecture vs MVT Architecture
- Django MVC – MVT Pattern

## **Getting Started with Django**

- Creating the first Project
- Integrating the Project to sublime text
- The Project Structure
- Running the server
- Solving the issues and Migrations
- Database Setup
- Setting Up Your Project

## **Create an Application**

- What Django Follows
- Structure of django framework
- Model Layer
- What are models
- Model fields
- Querysets

## **Django – Admin Interface**

- Starting the Admin Interface
- Migrations

## **Views Layer**

- Simple View
- Basic view(displaying hello world)
- Functional views, class based views

## **Django – URL Mapping**

- Organizing Your URLs
- Role of urls in djanggo
- Working urls
- Forms
- Sending Parameters to Views
- Templates layer
- The Render Function

## **Django Template Language (DTL)**

- Role of template layer in django
- Filters,Tags, Tag if, Tag for, Block and Extend Tags

- Comment Tag, Usage of templates
- Extending base template

## **Django – Models**

- Creating a Model
- Manipulating Data (CRUD)
- Linking Models
- Django – Page Redirection

## **Django – Sending E-mails**

- Sending a Simple E-mail
- Sending Multiple Mails with send\_mass\_mail
- Sending HTML E-mail
- Sending HTML E-mail with Attachments

## **Django – Form Processing**

- Using Form in a View
- Usage of forms
- Crud operations using forms
- Crispy forms in Django

## **Django – File Uploading**

- Uploading an Image
- Django – Apache Setup

## **Django – Cookies Handling**

- Django – Sessions
- Django – Comments

## **Django Admin**

- Creating Super User
- Using admin in Django
- Adding models to admin
- Adding model objects using admin
- Displaying in cmd using querysets

## **Static files**

- Loading css files into templates
- Loading js files into templates
- Uploading image using models
- User authentication

## **System Design**

- System Architecture
- System Architecture:
- Scalability
- Availability and Reliability
- Performance Optimization

## **SQL Server**

- Fundamentals (Database, Tables Rows & Columns)
- Primary Key, Foreign Key, Referential Integrity And Constraints
- SQL DDL Commands
- Store Procedure And Their Importance
- SQL Joins
- SQL Sub Query and Co- Related Queries

## **Sample Projects and Websites**

- News website (BLOGs Forums)
- Drawing a Graph using DB values