XCODERS
IT Training & Placements

PYTHON DJANGO

TRAINING COURSE BROCHURE

💥 ***Why Choose XCoders?***

🛠 ***Hands-on projects*** *for* ***real-world experience***

🌟 ***Guaranteed Internships*** *with* ***stipends up to ₹12,500!***

💬 *(Terms and conditions apply)*

🎯 ***100% Placement Assistance*** *to help you secure your dream job!*

📄 ***5 to 10 Interview Calls*** *Tailored support to help you land your ideal job.*

Module 1 : Introduction to Pythons

- **What is Python? History and Evolution:** Introduction to Python’s history, purpose, and growth in the programming landscape.
- **Unique Features of Python:** Overview of Python’s strengths, including readability, simplicity, and flexibility.
- **Installing Python and Setting Up Environment:** Step-by-step guide on setting up Python and configuring environments for coding.
- **First Python Program:** Writing a basic program to understand syntax and structure.
- **Python Basics:** Understanding identifiers (variable names), keywords, and indentation to structure code.
- **Comments and Documentation:** Using comments for code readability and documenting code properly.
- **Command Line Arguments:** Passing arguments from the command line to a Python script.
- **Getting User Input:** How to take input from users for interactive applications.

Module 2 : Python Operators

- **Overview of Operators:** Introduction to the types of operators and their importance.
- **Arithmetic, Logical, and Comparison Operators:** Perform math, logic, and comparison operations.
- **Bitwise, Identity, and Membership Operators:** Bitwise operations, checking identities, and membership of values.
- **Operator Precedence and Usage with Strings:** Understanding order of operations and using operators with strings.

Module 3 : Python Data Types

- **Variables and Variable Types** (Strings, Numeric, Boolean): Defining variables and understanding different data types.
- **Lists and Operations on Lists:** Working with lists, adding, accessing, and modifying list elements.
- **Dictionaries and Their Usage:** Creating dictionaries for key-value pairs, ideal for storing related data.

Module 4 : Control Statements

- **Introduction to Control Statements:** Understanding the purpose of control flow in programming.
- **Pass Statements:** Placeholder statements for future code.
- **Conditional Statements** (If, Else, Elif): Executing code based on conditions.
- **Nested If Statements:** Using if statements within other if statements to handle complex conditions.

📁 **Course Project**

To-Do List Manager

A to-do list manager where users can add, remove, and mark tasks as completed. The application will manage tasks and provide basic CRUD (Create, Read, Update, Delete) operations.

Module 5 : Loops

- **For and While Loops:** Iterating over a sequence or condition.
- **Loop Else Statements:** Executing code after a loop completes.
- **Nested Loops:** Loops within loops, useful for multidimensional data.
- **range() Function, break and continue Statements:** Controlling loop execution and using range to set loop limits.

Module 6 : Functions, Modules, and Packages

- **User-Defined Functions and Arguments:** Creating reusable code blocks with parameters.
- **Importing and Working with Modules:** Importing Python’s built-in and custom modules.
- **Exploring Packages (Math, Random, etc.):** Using packages for specialized tasks like math calculations or random numbers.
- **Recursion and Variable-Length Arguments:** Using recursion for self-referencing functions and handling flexible arguments.

Module 7 : String Handling

- **String Operations, Slices, and Functions:** Manipulating strings, slicing parts of strings, and using string functions.

Module 8 : Python Regular Expressions

- **Basics of Regular Expressions:** Introduction to pattern matching.
- Matching, Searching, and Replacing Patterns: Finding and replacing patterns in text using regex functions.

Module 9 : Python Lists

- **Creating and Accessing Lists:** Creating lists, adding/removing elements, and list operations.
- **List Operators, Functions, and Methods:** Advanced list handling, including sorting, slicing, and list comprehension.

Module 10 : Python Dictionary

- **Working with Key-Value Pairs:** Storing and managing data using dictionaries.
- **Dictionary Functions and Methods:** Using built-in methods to manipulate dictionaries.

Module11: Python Tuple

- **Advantages of Tuples over Lists:** Immutable data structures for secure data storage.
- Tuple Operations and Methods: Basic operations and methods for handling tuples.

Module 12: Python Sets

- **Creating and Working with Sets:** Understanding sets as unique data collections.
- **Set Operations (Union, Intersection):** Using set operations for mathematical set manipulation.
- **Frozen Sets:** Immutable sets for unchangeable collections.

Module 13: Object-Oriented Programming in Python

- **Classes, Objects, and Attributes:** Core OOP concepts.
- **Instance and Class Variables, Methods:** Variables and methods within classes.
- **Special Methods (__init__, __str__, etc.):** Built-in methods for initialization and representation.
- **Private Attributes and The Self Reference:** Managing private data and referencing class instances.

Module 14: Inheritance

- **Forms of Inheritance:** Understanding single, multiple, and hierarchical inheritance.
- **Creating Derived Classes:** Extending base classes to create new functionalities.
- **Method Overriding and Superclass Initialization:** Modifying inherited methods and initializing parent classes.

Module 15: Exception Handling

- **Understanding Exceptions:** Identifying runtime errors and how to handle them.
- **Try, Except, Else, and Finally Clauses:** Structured error handling.
- **Raising and Defining Custom Exceptions:** Creating custom error types for specific situations.

Module 16: File I/O

- **Reading and Writing Text Files:** Working with files for data storage and retrieval.
- **Binary Files and Pickle Module:** Handling binary files and serializing data with Pickle.

Module 17: Functional Programming

- **Defining and Calling Functions:** Creating reusable code blocks.
- **Scope (Global and Local):** Understanding variable scope and usage.
- **Lambda Functions:** Anonymous functions for short-term use.

Course Project

Offline Management System

Develop a Student Management System that stores and manages student data such as names, IDs, grades, and courses.

Module18: Python Built-in Functions

- **Common Built-In Functions:** Overview of useful functions, e.g., abs(), len(), map(), reduce(), etc.

Module 19: Git and GitHub

- **Version Control Basics:** Introduction to Git and GitHub.
- **Setting Up and Using Git:** Installing Git and creating repositories.
- **Branching, Merging, and Workflow:** Managing changes in production-ready projects.

Module 20: Python Database Connectivity

- **Connecting to Databases:** Using MySQL and other databases with Python.
- **CRUD Operations:** Basic database interactions – Create, Read, Update, Delete.

Module 21: Django

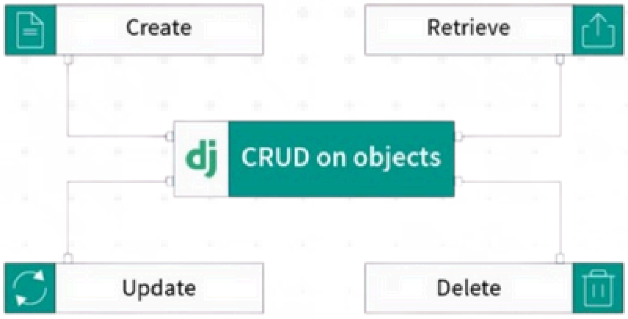
- **Django Setup and URL Mapping:** Installing Django and setting up URLs.
- **JQuery and Templates:** Using templates and JQuery for web interactivity.
- **Static Files and Models:** Serving static assets and working with data models.
- **Forms and Validation:** Building and validating forms for user input.
- **User Authentication and Registration:** Managing user accounts and authentication.
- **Deployment and Debugging:** Deploying Django applications to platforms like GitHub.

Course Project
CRUD Application

Basic app with Create, Read, Update, and Delete functionality.

Course Project
E-commerce Portal

Create an E-commerce Portal that allows users to browse products, add items to a shopping cart, and proceed to checkout.


Create
Retrieve
dj CRUD on objects
Update
Delete