

Curriculum

1. Introduction to Python

Goal : Give brief idea of what Python is and touch on basics.

Objectives:

- Define Python
- Setup Python environment
- Write your first Python program
- Know why Python is popular
- Discuss flow control

Topics:

- Get an overview of Python
- List the Advantages/Disadvantages of Python
- Start Python
- Use the Interpreter
- Discuss Python Scripts on UNIX/Windows
- Variables, Keywords, Built-in Functions, Strings, Different literals, Math operators and expressions, Writing to the screen, String formatting, Command line parameters and Flow Control.
- Learn about Interpreted Languages
- Explore Pydoc
- Discuss Interpreter PATH
- Run a Python Script
- Explore Python Editors and IDEs Use

Hands On:

- Variables

- Literals

2. Sequences and File Operations

Goal : Learn different types of sequence structures, related operations and their usage. Also learn diverse ways of opening, reading, and writing to files.

Objectives:

- Define Reserved Keywords and Command Line Arguments
- Describe Flow Control and Sequences
- Define and Describe Dictionaries and Sets
- Describe how to Get User Input from Keyboard
- Practice Working with Files

Topics:

- Lists
- Indexing and Slicing
- Functions for all sequences
- Operators and keywords for sequences
- comprehensions
- Dictionaries and sets.
- Modes of opening a file
- File methods
- Tuples
- Iterating through a sequence
- Using enumerate()
- The xrange() function List
- Generator expressions
- Working with files
- File attributes

Hands On:

- List - properties, related operations
- Dictionary - properties, related operations, comparison with list
- Tuple - properties, related operations, comparison with list
- Set - properties, related operations, comparison with dictionary

3. Deep Dive – Functions, Sorting, Errors and Exception, Regular Expressions and Packages

Goal : Learn how to create generic python scripts, how to address errors/exceptions in code and finally how to extract/filter content using regex.

Objectives:

- Explain Functions and various forms of Function Arguments
- Define Modules
- Explain Standard Library
- Describe Zip Archives and Packaging

Topics:

- Functions
- Global variables
- Sorting
- Lambda Functions
- Sorting dictionaries
- Errors and Exception Handling
- The standard exception hierarchy using Modules
- Module search path
- Function Parameters
- Variable scope and Returning Values
- Alternate Keys
- Sorting collections of collections
- Sorting lists in place
- Handling multiple exceptions
- The Import statement
- Package installation ways Module Aliases and Regular Expressions

Hands On / Demo :

- Functions - syntax, arguments, keyword arguments, return values
- Sorting - sequences, dictionaries, limitations of sorting
- Packages and module - modules, import options, sys path
- Lambda - features, syntax, options, comparison with functions
- Errors and exceptions - types of issues, remediation

Object Oriented Programming in Python

Goal : Understand the Object-Oriented Programming world in Python and use of standard libraries.

Objectives:

- Implement Regular Expression and its Basic Functions
- Develop applications based on Object Oriented Programming and Methods
- Use Classes, Objects, and Attributes

Topics:

- The sys Module
- STUDIO
- Paths
- Walking directory trees
- Random Numbers
- Zipped Archives
- Defining Classes
- Instance methods
- Class methods and data
- Private methods and Inheritance
- Interpreter information
- Launching external programs
- Directories and filenames
- Math Function
- Dates and Times
- Introduction to Python Classes
- Initializes
- Properties
- Static methods

Hands On:

- Regular expressions - regex library, search/match object, findall, sub, compile
- OOPS paradigm - Inheritance, Polymorphism and Encapsulation in Python
- Classes - classes and objects, access modifiers, instance and class members

Debugging, Databases and ProjectSkeletons

Goal :Learn how to debug, how to use databases and how a project skeleton looks like in Python.

Objectives:

- Debug python scripts using pdb
- Classify Errors
- Create project Skeletons
- Perform CRUD operations on SQLite database
- Debug python scripts using IDE
- Develop Unit Tests
- Implement Database using SQLite

Topics:

- Debugging
- Using unit tests
- Required packages
- Project Directory
- Testing your set up
- Creating a database with SQLite 3
- Creating a database object.
- Dealing with errors
- Project Skeleton
- Creating the Skeleton
- Final Directory Structure
- Using the skeleton
- CRUD operations

Hands On:

- Debugging - debugging options, logging, troubleshooting
- Project skeleton - industry standard, configurations, sharable libraries
- Unit testing - TDD, unittest library, assertions, automated testing
- RDBMS - Python for RDBMS, PEP 49, CRUD operations on Sqlite

6. Introduction to Django WebFramework

Goal :In this module, you will be introduced to Django and learn how to create views and perform URL

Objectives :

- Explain Web Framework
- Create a basic Django app
- Use HTTP request and response objects
- Explain MVC pattern
- Create Views
- Use URLConf

Topics :

- Web development
- Features of Django
- MVC model
- Views
- Introduction to Django Web Framework
- Installing Django
- HTTP concepts
- URL Mapping

Hands On/Demo :

- Create a simple View using Django

7. Templates and Forms

Goal : In this module, you will learn how to create Templates and Forms in Django

Objectives :

- Explain the Django Template System
- Render Templates
- Process Form Data
- Load Template Files
- Create Forms
- Customize Form Field Validation

Topics :

- Django Template Language
- Creating Template Objects
- Utilities of Templates
- Tags, Variables and Filters

- Rendering Templates
- Form Handling
- Display
- Template Inheritance
- Form validation and Error Messages Form

Hands On/Demo :

- Create a Form that accepts personal data to from a user

8. Models and Dynamic Webpages

Goal :In this module, you will learn how to create Database Models and add Dynamic content to your webpages

Objectives :

- Define Database Models
- Populate a Database, CRUD
- jQuery and AJAX with Django to create Dynamic websites
- Use Model Fields
- Use QuerySets for data retrieval Use

Topics :

- Django Models
- Model Inheritance
- Primary keys and the Model
- Toggle Hidden Content
- Model Fields
- CRUD on DB
- Dynamic Webpages
- jQuery and AJAX integration

Hands On/Demo :

- Adding a Like button to a webpage

9. Serialization

Goal :In this module, you will learn how to serialize and deserialize data and create APIs

Objectives :

- Explain Data Serialization
- Use Serializers and Deserializers
- Use REST APIs
- Use Django's REST Framework
- Use Model Serializers

Topics :

- Serialization and Deserialization
- Serializer class
- REST APIs
- Django REST Framework
- Model Serializers

Hands On/Demo :

- Creating a REST API

10. Parsing XML and JSON with Python

Goal :In this module, you will learn how to parse data stored in XML & JSON formats using Python

Objectives :

- Explain XML and JSON file formats
- Parse data stored in both XML and JSON formats
- Explain XML-RPC
- Stores data in XML and JSON formats

Topics :

- XML-RPC
- XML, parsing object to XML and back

Hands On/Demo :

- Parse data stored in XML/JSON format to native Python type and vice-versa