

Python Training Overview

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. Python has been one of the premier, flexible, and powerful open-source language that is easy to learn, easy to use, and has powerful libraries for data manipulation and analysis

Objectives of the Course

- To understand the concepts and constructs of Python
- To create own Python programs, know the machine learning algorithms in Python and work on a real-time project running on Python

Python Course Content

Core Python

Introduction to Python

- What is Python?
- Why Python?
- Who Uses Python?
- Characteristics of Python
- History of Python
- What is PSF?
- Python Versions
- How to Download and Install Python
- Install Python with Diff IDEs
- Features and Limitations of Python
- Python Applications
- Creating Your First Python Program
- Printing to the Screen
- Reading Keyboard Input
- Using Command Prompt and GUI or IDE
- Python Distributions

Variables in Python

- What is Variable?

- Variables and Constants in Python
- Variable, Variable names and Value
- Mnemonic Variable Names
- Values and Types
- What Does "Type" Mean?
- Multiple Assignment
- Python different numerical types
- Standard Data Types
- Operators and Operands
- Order of Operations
- Swap variables
- Python Mathematics
- Type Conversion
- Mutable Versus Immutable Objects

String Handling

- What is string?
- String operations and indices
- Basic String Operations
- String Functions, Methods
- Delete a string
- String Multiplication and concatenation
- Python Keywords, Identifiers and Literals
- String Formatting Operator
- Structuring with indentation in Python
- Built-in String Methods
- Define Data Structure?
- Data Structures in PYTHON

Python Operators and Operands

- Arithmetic, Relational Operators and Comparison Operators
- Python Assignment Operators
- Short hand Assignment Operators
- Logical Operators or Bitwise Operators
- Membership Operators
- Identity Operators
- Operator precedence
- Evaluating Expressions

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 Statemen

Python LOOPS

- How to use "While Loop" and "For Loop"
- How to use For Loop for set of other things besides numbers
- Break statements, Continue statement, Enumerate function for For Loop
- Practical Example
- How to use for loop to repeat the same statement over and again
- Break, continue statements

Learning Python Strings

- Accessing Values in Strings
- Various String Operators
- Some more examples
- Python String replace() Method
- Changing upper and lower case strings
- Using "join" function for the string
- Reversing String
- Split Strings

Sequence or Collections in PYTHON

- Strings
- Unicode Strings
- Lists
- Tuples
- buffers
- xrange

Python Lists

- Lists are mutable
- Getting to Lists
- List indices

- Traversing a list
- List operations, slices and methods
- Map, filter and reduce
- Deleting elements
- Lists and strings

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
- Built-in functions with Tuple
- Dotted Charts

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
- Dictionary len() Method
- Variable Types
- Python List cmp() Method
- Dictionary Str(dict)

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 and Non-Default Arguments
- Keyword Argument and Non-keyword Arguments
- Arbitrary Arguments
- Rules to define a function in Python
- Various Forms of Function Arguments
- Scope and Lifetime of variables
- Nested Functions
- Call By Value, Call by Reference
- Anonymous Functions/Lambda functions
- Passing functions to function
- map(), filter(), reduce() functions
- What is a Docstring?

File Handling

- What is a data, Information File?
- File Objects
- File Different Modes and Object Attributes
- How to create a Text File and Append Data to a File and Read a File
- Closing a file
- Read, read line, read lines, write, write lines...!!
- Renaming and Deleting Files

Python Modules

- What is a Module?
- Types of Modules
- The import Statement
- The from...import Statement
- ..import * Statement
- Underscores in Python
- The dir() Function
- Creating User defined Modules
- Command line Arguments
- Python Module Search Path

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
- Argument of an Exception
- Python Custom Exceptions
- Ignore Errors
- Assertions
- Using Assertions Effectively

Packages in Python

- What is a Package?
- Introduction to Packages?
- py file
- Importing module from a package
- Creating a Package
- Creating Sub Package
- Importing from Sub-Packages
- Popular Python Packages

Advanced Python

Python Class and Objects

- Introduction to OOPs Programming
- Object Oriented Programming System
- OOPS Principles
- Define Classes
- Creating Objects
- Class variables and Instance Variables Constructors
- Basic concept of Object and Classes
- Access Modifiers
- How to define Python classes
- Python Namespace
- Self-variable in python

- Garbage Collection
- What is Inheritance? Types of Inheritance?
- How Inheritance works?
- Python Multiple Inheritance
- Overloading and Over Riding
- Polymorphism
- Abstraction
- Encapsulation
- Built-In Class Attributes

Python Regular Expressions

- What is Regular Expression?
- Regular Expression Syntax
- Understanding Regular Expressions
- Regular Expression Patterns
- Literal characters
- Repetition Cases
- Example of w+ and ^ Expression
- Example of \s expression in re.split function
- Using regular expression methods
- Using re.match()
- Finding Pattern in Text (re.search())
- Using re.findall for text
- Python Flags
- Methods of Regular Expressions

File Handling

- Working with CSV files and Dictionary
- Working with JSON files

Multi-Threading

- What is Multi-Threading
- Threading Module
- Defining a Thread
- Thread Synchronization

Django Web Framework

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

Getting Started with Django

- Creating the first Project
- Integrating the Project to PYCHARM & Other IDEs
- The Project Structure
- Running the in the server
- Setting Up Your Project
- Create and configure Django apps
- Django App Structure
- Configuring App in Project

URLs and Views

- What is URL?
- Define URL patterns
- What is View
- Configure URLs
- Developing different views

URL dispatcher

- Django URL Mapping

- Configuring URLconf's
- Django URL Functions
- Path()
- Re_path()
- URL Patterns App Level and Project Level
- Include()

Django Templates

- Define Django Template
- Django MVT Pattern
- Django Template Configuration
- render() function
- Django Template Language
- Template Tags or Template Variables
- String interpolation
- What is a context?
- Define ContextProcessor?

Working with Static Files

- include Static Files inside Template
- Configuring static files
- Creating Project with static resources
- HTML, CSS, Images

DJANGO MODEL

- Define Database?
- Define DBMS, RDBMS?
- What is Model?
- Database Configuration

- How to Check Django Database Connection
- Configuration of MySQL Database
- SQLite
- Defining Django Models
- Django Model Fields
- Field Options
- What is a Migration?
- migrate Command
- Creation of Super User
- Register Model Inside Admin Interface
- Difference between makemigrations and migrate
- Define QuerySet

Relationships in Django Models

- What is ORM?
- One-To-One Relationships
- Many-To-One Relationships
- Many-To-Many Relationships

Django Forms or Model Forms

- What is Model Form?
- Advantages of Django Forms
- Creating Django Forms
- CSRF
- Creating a Login form
- Creating Registration Form

Django Form Validation

- What is Validation?

- is_valid()
- Clean Methods
- Creating Project
- What is HTTP?
- Basic Features
- http request methods
- HTTP – URL Encoding

Django's Inbuilt Core Validators

- Validators
- Custom Validators
- Validation of Total Form using Clean Method
- Validators
- RegexValidator
- Validating EmailID
- What is BOT?
- How to prevent Requests from BOT

Model Based Forms

- How to develop MBF
- Exclude-List
- Include-Tuple
- Creating SuperUser

Advanced Templates

- Template Inheritance
- Template Filters
- Template tags for relative URLs
- Block-endblock

- extends
- Advantages of Template Inheritance
- Why Template Filters?
- What is Template Filter?
- How to Create Customized Template Filters?

Session Management in Django

- Cookies
- What is Cookie?
- Why Cookie?
- Types of Cookies
- Advantages, Limitations
- `set_test_cookie()`:
- `test_cookie_worked()`
- `delete_test_cookie()`
- `set_cookie()`
- `get()`
- Django Session Framework
- `request.session['key']`
- `request.session.get_expiry_date()`

Authentication & Authorization

- Define Authentication
- Define Authorization
- Enabling authentication
- Working with Admin

Class Based Views (CBV)

- Function Based Views

- Class based Views
- Generic Class-Based Views
- Simple Generic Views
- Template View
- List Views

Django File Upload

- What is upload
- Creating media_root & url
- How to upload files
- Upload images
- Upload Audio Files
- Upload Video Files

Django CRUD Operations

- What is CRUD?
- CREATE
- READ
- UPDATE
- DELETE
- CRUD with Project

Django Middleware

- Define Middleware
- Builtin Middleware

- Customizing Middleware
- Middleware Methods

How to Send Email in a Django

- What is mail system
- Mail Requirements
- send_mail()
- Configuring Mail Settings
- Sending Email
- Other Email Functions

Outputting CSV with Django

- What is CSV?
- CSV Using Models
- Creating a Live CSV Project

Outputting PDF with Django

- What is PDF?
- Creating a Live PDF Project

Django Crispy Forms

- What is Crispy Form?
- Installing Crispy form
- Template Packs
- Login Crispy form
- Registration Crispy forms
- Crispy with Bootstrap

GIT & Github

- What is git?
- Define Version Control System
- Git Installation
- Git commands
- Define Github
- Pulling and Pushing
- Live Environment

Bitbucket

- What is Bitbucket?
- Web based Version Control System
- Commits
- Branches
- Pull requests
- Pipelines

Deploying Django Apps & Heroku

- What is deployment?
- Creating a Project
- Deploying Django App at product level
- Install heroku CLI
- Virtual Environment
- Introduction to HEROKU
- Deploy at HEROKU

Introduction to Web Services

- What is Web Service?
- Types of WebServices
- SOAP based WebServices?

- Features and Limitations?

Introduction to REST API (Restful Services)

- What is REST?
- What is API?
- What is JSON?
- HTTP Status Codes?
- HTTP Methods for RESTful Services
- POST
- PUT
- GET
- DELETE

Basics of Data Science, Machine learning will be covered (NUMPY, PANDAS, MATPLOTLIB)