

## Core-Python

### **Python Overview:**

What is Python? The Birth of Python, History of Python, Features of Python, Versions of Python, Applications of Python. Implementation of Python.

Python distributions: Cpython, Jython, IronPython, pypy.

### **The Python Environment:**

Installation of Python, Python Documentation, Getting Help, How to develop python applications/projects, Python Editors and IDE's, Basic Syntax, Running a Python Script, Python Scripts on UNIX/Windows.

### **Getting Started:**

Keywords, Data Types, Variables, assign values to variables, multiple assignments, Space Indentation, Quotes, Comments, Print(), Type(), Id( ) Functions, input(), raw\_input() functions, How to read the data from keyboard, Type conversions, Type conversions ,Number systems, Mutable and Immutable objects.

### **String Handling:**

What is String? Single-quoted, Triple-quoted , String indexing, String Slicing, Working with Functions, Working with String Methods.

### **Operators:**

What is Expression, What is Operator?

Types of operators:

- ✓ Arithmetic Operators
- ✓ Relational Operators
- ✓ Logical Operators
- ✓ Assignment Operators
- ✓ Short Hand Assignment Operators
- ✓ Walrus Assignment Operators
- ✓ Bitwise Operators
- ✓ Membership Operators

- ✓ Identity Operators
- ✓ Operator Precedence.  
Difference between 'is' operator and '== ' operator

## **Flow Control:**

About Flow Control Statements, Elements of Control Flow Statements, Types of Control Flow Statements.

## **Conditional Statements:**

- ✓ simple if
- ✓ if else
- ✓ if elif else
- ✓ elif ladder
- ✓ nested if

## **Looping Statements:**

- ✓ while loop
- ✓ infinite while loop
- ✓ while else,
- ✓ for loop,
- ✓ for else
- ✓ nested loops

## **Jumping statements:**

- ✓ Break statement
- ✓ Continue statement
- ✓ Pass statement

**Python Collections:**

What is Collection Object? Types of Collection Objects:

- ✓ Sequence Collection Objects
- ✓ Non Sequence Collection Objects

**List Collection:**

What is list, creating list, accessing/deleting/updating list elements, indexing, slicing and matrix, working with built-in list functions and methods. list comprehension.

**Tuple Collection:**

What is tuple, creating tuple, accessing/deleting/updating tuple elements, indexing, slicing and matrix, working with built-in tuple functions and methods.

**Set Collection:**

What is Set, creating Set, working with built-in set functions and method and set comprehension, mathematical set operations:

**Frozen set Collection:**

What is Frozen set, creating frozen set, working with built-in frozen set Functions and methods

**Dict Collection (Mapping):**

What is dictionary, creating a dictionary, accessing/deleting/updating? Dictionary elements, working with built-in dictionary functions and Methods, dictionary comprehension.

**Functions:**

What is Function, advantages of functions, Types of Functions?

- ✓ Built-in Functions
- ✓ User-Defined Functions

What is Built-in Function?

print( ), type( ), id( ), len( ), min( ), max( ), sum( ), sorted( ), reversed( ), range( ), xrange( ), abs( ), all( ), any( ), format( ), enumerate( ), map( ), filter( ), reduce( ), round( ), zip( ), ..., etc.

What is User-Defined Function, how to create a user-defined function And how to call user-defined functions.

What is Parameter, What is Argument and types of arguments?

- ✓ Normal Arguments
  - Non-Default Arguments
  - Default Arguments
  - Non-Keyword Arguments
  - Keyword Arguments
- ✓ Arbitrary Arguments
- ✓ Kw args

What is Return statement, how to handle the return statement.

What are variable, types of variables?

- ✓ Local Variables
- ✓ Global Variables
- ✓ Non-Local Variables

What is name space, scope and scope of the variables in python.

Pass the collection as a parameter to the function and pass the function as a parameter to the function, call by value, call by reference, function overloading.

What is Function Recursion?

What is Anonymous function (Lambda Functions), Built-in higher order functions like filter ( ), map ( ), reduce ( ).

## **ADVANCED PYTHON**

### **OOPS CONCEPTS:**

About OO programming, Benefits of OOP's concepts. The OOP's concepts are:

- ✓ Encapsulation
- ✓ Inheritance
- ✓ Polymorphism
- ✓ Data Abstraction

What is a Class, Defining a Class, what is Object, Creating a object, what is reference variable, Class methods and data, static variables and non-static variables, local variables, Static methods, Instance methods, nested classes, nested methods, Constructors, Garbage collection, Destructors, Built-in attributes of a class, add and remove the attributes of a class from outside of that class, what is Has-a relationship or Association, aggregation, composition, What is Is-a relationship or Inheritance, Types of inheritances, what is MRO, Polymorphism (over loading & over riding), Data hiding, access modifiers, Dunder methods. Setters and getters, what is abstract class and abstract method.

**Modules:**

What is a module?, Creating user defined module, Importing a module

In python:

- ✓ normal import
- ✓ from import
- ✓ from import with \*

Renaming a module, module search path, reloading a Module, Dir. function,

Working with Standard modules (Built-ins, Math, Calendar, Random,

Date time and time, Os and sys, String...),

The hidden concept of if `__name__ == '__main__'` condition.

**Packages:**

What is a package?, Creating user defined package, Importing a package in python:

- ✓ normal import
- ✓ from import
- ✓ from import with \*

**Multi-Threading:**

what is multi-tasking, types of multi-tasking, what is thread, what is multi-threading, Defining a Thread, starting a Thread, Thread Life Cycle, What is Scheduling, suspend Thread by using `sleep()`, Threads synchronization.

Synchronization primitives are :

- ✓ Semaphore,
- ✓ Locks,
- ✓ Events,
- ✓ Condition Variables.

What is GIL?

**Errors and Exception Handling:**

What is Error?, types of Errors

- ✓ Syntax Errors,
- ✓ Runtime Errors,

What is Exception?, Types of Exceptions:

- ✓ Built-in Exceptions
- ✓ User-defined Exceptions

What is Exception Handling, how to Handling the Exceptions:

- ✓ try block/clause
- ✓ except block/clause
- ✓ Finally block/clause.

Need of Exception handling single try block with multiple Except blocks, Nested try blocks, Handling Multiple Exceptions, What is user defined exception, how to create user-defined exceptions, how To Raising the user defined exceptions and how to handle the user defined Exceptions. What is Assertion, how to implement the assertions in python.

## **File Handling:**

History of file concept, what is file, types of file formats, mode of the files, order of the file handling, Opening a file, Closing a file, Writing data to files, Reading a data from files, tell( ), Seek( ), read( ), readlines( ), read(n), write( ), writelines( ), close( ), readline( ), functions.

## **Serialization and de-serialization:**

What is Serialization, how to implement Serialization in python, what is de-serialization, how to implement Serialization and de-serialization in python.

- ✓ pickle module
- ✓ marshal module

## **Database Access:**

Basics of database (What is Data, What is Information, What is DBMS, Types of Data bases), Connections, Executing SQL and queries,

Basic SQL commands are:

- ✓ DDL(Data Definition Languages)
- ✓ DML(Data Manipulation Languages)
- ✓ DCL(Data Control Languages)
- ✓ TCL(Transaction Control Languages)
- ✓ DRL (Data Retrieval Languages).

Working with sqlite3 database.

Working with MySQL database.

Working with mongoDB

## **Command Line Arguments:**

What is Command Line argument, How to implement Command Line Arguments in python.

## **Regular Expressions:**

What is regular expression?, Wild card characters, Forming regular expressions, special characters, Character classes, Quantifiers, Greedy matches, Grouping, Match, Search functions, matching/searching, findall function, finditer function

Sub function, splitting a string, Replacing text, Flags. Like ignore the cases, multi-lines.

**Advanced concepts in python:**

- ✓ Iterators
- ✓ Generators
- ✓ Closure's
- ✓ Decorators
- ✓ Working with JSON files
- ✓ Working with CSV files
- ✓ Data classes
- ✓ Property decorator
- ✓ How to run python code internally?
- ✓ How to print assembly code for our python script?
- ✓ Python memory management
- ✓ Type hinting and typing module
- ✓ Type checking with mypy module
- ✓ What is monkey patching

**Basics of Django:**

Introduction about Web-applications:

- ✓ What is web application
- ✓ Architecture of web application
- ✓ What are the Requirements to Design a web application

Introduction about Framework:

- ✓ What is Framework
- ✓ What are the advantages of Framework
- ✓ List of python web related Frameworks

Introduction about Django:

- ✓ What is Django
- ✓ History of Django
- ✓ What are the Features of Django
- ✓ Introduction to MVT Design pattern
- ✓ Architecture of Django

Django Environment setup:

Working with Django framework what are the pre-requisites we are needed.

- ✓ Python
- ✓ IDE's
- ✓ Virtual Environment
- ✓ Django

Structure of the project:

- ✓ How to create a project
- ✓ Describe project structure
- ✓ How to create a application for project
- ✓ Describe application structure
- ✓ How to Run the project

**Data Science:**

- ✓ what is data science
- ✓ what is data analysis
- ✓ what is data reporting
- ✓ how to analyzing the data by using NumPy and Pandas packages
- ✓ how to generating the reports by using Matplotlib package

**GUI/Desktop applications:**

- ✓ what is GUI
- ✓ how to developing GUI applications in Python
- ✓ what are the modules are available in python
- ✓ what is tkinter/Tkinter
- ✓ how to create widgets like
- ✓ Button
- ✓ Checkbox
- ✓ Radio button
- ✓ Entry Field
- ✓ Scroll bar
- ✓ Menu bar
- ✓ List box

**Debugging:**

- ✓ what is bug
- ✓ what is debugging
- ✓ how to debugging in python by using pdb module and GUI

**Testing:**

- ✓ what is testing
- ✓ what is the difference between testing and debugging
- ✓ what is unit testing
- ✓ what is integration testing
- ✓ how to perform unoit testing by using unittest and Pytest
- ✓ what is AAA model



- ✓ what is manual testing
- ✓ what is automation testing
- ✓ what is Selenium
- ✓ how test an application through python

**Networking:**

- ✓ what is Socket
- ✓ what are the networking protocols
- ✓ what are Networking modules/internet modules
- ✓ how to provide the communication between client and server
- ✓ how to sending an email

**Web scrapping:**

- ✓ what is web scrapping
- ✓ how to working with BeautifulSoup4

**Pillow package:**

- ✓ Load an Image
- ✓ Displaying an Image
- ✓ Blur the Image
- ✓ Cropping an Image
- ✓ Rotating an Image

**Boto module for AWS:**

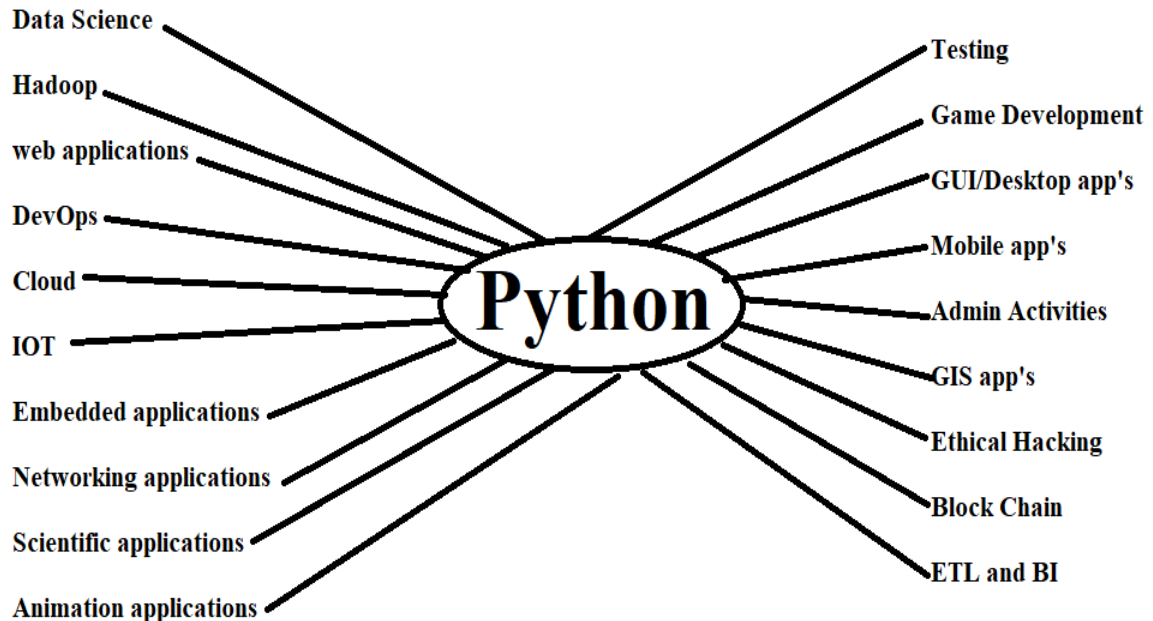
- ✓ What is Boto ?
- ✓ Creating a bucket
- ✓ Print the buckets in s3
- ✓ Print the list of buckets
- ✓ Filtering the buckets
- ✓ Uploading a file into the bucket
- ✓ Downloading files from s3 bucket

- ❖ Interview preparation tests
- ❖ Resume preparation
- ❖ Real-time project explanation.
  - Beginner Level
  - Intermediate Level
  - Advanced Level

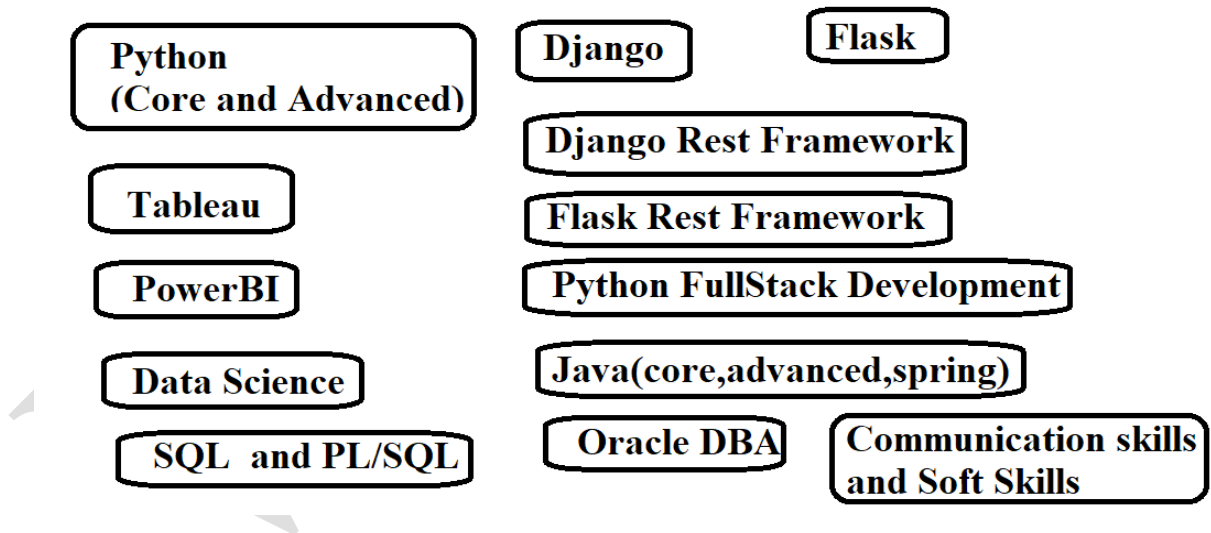
# VAGDEVI TECHNOLOGIES

**PYTHON Content**

**9133757367,9515901571**



## Courses



## VAGDEVI TECHNOLOGIES

Flat no. 210, Annapurna Block, Adithya Enclave, Ameerpet, Hyderabad  
www.Vagdevitechnologies.com Phone: 9133757367,9515901571  
Email: Vagdevitechnologies@gmail.com