

Introduction

- Introduction to Programming Languages
- Types of Programming Languages
- Interpreter based Languages vs Compiler based Languages
- What is Python?
- History of Python
- Languages used to develop Python
- Different languages vs. Python
- Realtime use-cases of Python
- Features and versions of Python
- Distributions of Python

Python Environment

- Installation of Python on WINDOWS, UNIX and MAC
- Path Settings for Python
- Python Documentation
- Getting Help
- Interactive Mode
- Working with Python Command Line Shell
- Batch Mode
- Working with Editors, IDE's and Notebooks
- Basic Syntax
- Running Python Scripts on Windows
- Running Python Scripts on LINUX
- Working with Python Cloud
- Executing python Scripts on Android
- Executing python Scripts on IOS
- Python Indentation
- Comments and Quotations
- Python Identifiers and Keywords

Variables and Naming Conventions

- Assigning values to variables in different ways
- Print(), type() and id() built functions
- Reading data from user using input()/raw_input()....

Number System

- Binary number system
- Octal number system
- Decimal number system
- Hexa-decimal number system

Python data types

- Static vs Dynamic data types
- Types of Data Types
- int, float
- complex, bool
- str, range, list, set
- tuple, dict
- None, frozenset
- bytes, bytearray

Type conversions and eval()

Introduction to Data Structures

String data Structure

- What is a string and Different ways to create a string
- String indexing and string slicing
- String concatenation and string multiplication
- String unpacking
- Splitting the data in different parts as per user
- capitalize() and title() and split()
- del, count(), find(), swapcase()
- reverse(),replace() and sort()
- String immutable

List Data Structure

- Different ways to create a list object
- Creating and working with homogeneous lists
- Creating an working with heterogeneous lists
- List indexing and list slicing, list reversing
- List concatenation and list multiplication
-

- Generating list by using range function
- List unpacking and list mutable
- Creating nested lists and indexing nested lists
- Python range() and xrange() functions
- Python insert, append and extend
- Remove, pop and clear
- Python list ascending and descending
- Converting given string data structure into list
- Converting given list data structure into string
- Creating list from user values
- Nested lists, nested lists sorting

Tuple Data Structure

- creating a tuple Object in different ways
- creating and working with homogeneous tuple
- creating and working with heterogeneous tuple
- tuple indexing and tuple slicing
- tuple concatenation and tuple multiplication
- tuple unpacking and tuple immutable
- all, any, len and sort
- del keyword
- python tuple ascending and descending
- creating and working with nested tuples
- Conversions:
 - o converting given string data structure into tuple
 - o converting given list data structure into tuple
 - o converting given tuple data structure into string
 - o converting given tuple data structure into list

- advantages of tuple over list data structure
- difference between list and tuple
- mutable vs. immutable
- Nested tuples

Set Data Structure

- Creating and working with set data structure in different ways
- Normal sets and frozen sets
- Set mutable and unpacking set data structure
- Creating and working with sets with homogeneous elements
- Creating and working with sets with heterogeneous elements
- Creating empty sets and modifying the empty sets
- Why sets not support indexing and slicing
- Add, remove and discard the elements to set data structure
- Issubset, issuperset and isdisjoint
- Union, intersection and difference
- Intersection_update and difference_update
- Symmetric_difference and symmetric_difference_update
- Conversions:
 - o Converting given string data structure into set
 - o Converting given list data structure into set
 - o Converting given tuple data structure into set
 - o Converting given set data structure into string
 - o Converting given set data structure into list
 - o Converting given set data structure into tuple

Dictionary Data Structure

- Creating and working with dictionary data structure in different ways
- Creating empty dictionary and working with empty dictionary
- Working with key and value pairs
- Dictionary mutable and unpacking dictionary
- Adding and deleting key and value pairs to the existing data structure
- Difference between pop and popitem operations
- Extracting only keys from the existing data structure
- Extracting only values from the existing data structure
- Clear and pop methods
- Del keyword and pop method
- Creating a dictionary from existing another data structure like tuple

Operators

- Arithmetic operators (+, -, *, /, %, //, **)
- Logical or Boolean operators (and, or, not)
- Assignment operators (=, +=, -=, /=, *=, %=, //=, **=, >>=, <<=, &=, |=, ^=)
- Comparison operators (>, <, >=, <=, ==, !=)
- Bitwise operators (~, <<, >>, &, ^, |)
- Identity operators (is, is not)
- Membership operators (in, not in)
- unary (+) and unary (-)
- Operator precedence

Python Functions and Arguments

- Defining functions and working with functions
- Using def keyword for functions
- Called functions and function definition and calling functions
- Formal arguments and actual arguments, return values

- Working with named arguments and keyword arguments
- Default arguments and positional arguments
- Working with default arguments and normal arguments
- *args and **kwargs arguments
- Argument unpacking
- Variable length arguments
- Using data structures to function definitions
- Nested functions, global, nonlocal keywords
- Dir() and Format() functions
- Enumerate function

Lambda Functions

- Creating functions by using lambda keyword
- Difference between def and lambda functions
- Working with filter functions
- Working with map functions
- Working with reduce functions
- Documentation strings
- Function Annotations
- Intermezzo : coding style

Control Statements

- Simple If statement
- If else statement
- Elif statement
- Nested if statement
- Membership test for string
- Membership test for tuple
- Membership test for list
- Membership test for set
- Membership test for dictionary
- conditional operator behaviour using if-else

Loopings

- For loop
- While loop

- Pass, continue and break statements
- Iterating over list, tuple, set and dictionary
- range() function
- while vs. else
- for vs. else
- infinite while and for loop creation
- pattern printing programming

Advanced Concepts on Data Structures

- List comprehension
- Dictionary comprehension
- Nested data structures

Modules

- What is module and purpose of modules
- Different types of modules
- Different ways to import modules
- Standard modules and user modules
- From `Ö` import *
- Creating own modules
- Using modules in other modules
- Working with some standard modules
- MATH, DATETIME, CALENDAR, SYS, OS Modules

Packages

- Introduction to Packages
- `__init__.py` file
- Defining Packages
- Importing from Packages

- Defining sub Packages
- Importing from sub Packages
- Differences between 2.X and 3.X packages

Garbage Collections

- Introduction to Garbage Collection
- Referenced versus UnReferenced Objects
- Object Reference Count
- Garbage Collector

- Objects Garbage Collection
- Destructors
- del statement
- Collections Garbage Collection

Advanced Python

File Handling

- What is a file
- Types of filesystems
- File vs. directory
- Creating a file in a directory
- Open the file in the python
- Different ways to open the file in Python
- Writing to the file
- Appending the data to the existing file
- Modes of operations
- Seek and tell methods
- Readline and readlines
- Working with words and characters in the file
- Real-time scenarios on files
- pickling/unpickling

OOPS Concepts

- Procedural vs. OOPs
- Difference between C/C++ Vs. Java Vs. Python
- Class and object
- Class(static) variables and instance variables
- Constructor
- Data hiding
- Method overloading and overriding
- Abstraction, accessing hidden properties of a class
- Inheritance
- Polymorphism
- Encapsulation
- Inner classes
- build in properties of a class
- Sorting custom objects

Exception Handling in Python

- What is an syntax error, Runtime error, exception,
- Exceptions handling Objective and exception hierarchy
- Try and except block
- Handling multiple exceptions using multiple excepts
- Handling multiple exceptions using single except
- Working with default except
- Handling exceptions with else and finally blocks
- Using assert for handling exceptions
- Return statement w.r.t try, except, finally blocks
- Creating custom exceptions

Logging in python

- What is logging and purpose of logging
- Creating a log file
- Storing runtime events in log file
- Different modes to store the data in log file
- DEBUG, INFO, WARNING, ERROR, CRITICAL

Iterators, generators and decorators, closures

- Working with yield keyword
- Difference between yield and return
- Decorating a function with another function
- PIP installation

Regular Expressions

- Basics of regular expressions
- Findall function
- Search function
- Match methods
 - o Group
 - o Groups

- Matching and searching
- Compile and sub functions
- Mobile numbers verifications
- Email ids verifications
- Web scrapping

Command Line Arguments

- Reading command line arguments
- Using command line arguments

Introduction to DBMS

- Installation of MySQL Database
- Creating Databases in MySQL
- Creating Users, assigning Privileges in MySQL
- Installation of MySQL Python Modules
- Establishing connection with MySQL
- Closing MySQL Database Connections
- Execution of Insert, Update, Delete and Select Queries

Working with Database Connection

- Connecting to database from Python application
- Creating connection to the database from Python application
- Creating database and tables from Python applications to the database
- Fetching data and updating data in the entities.
- Using cursor to execute SQL command in Python application
- Using Fetchall and Fetchone methods

MultiThreading

- Introduction
- Threading module
- Defining a Thread
- Starting a Thread
- Threads Synchronization
- Multithreaded Priority Queue
- Old Version Threading

GUI Programming

- Introduction
- Components and Events
- An Example GUI
- The root Component
- Adding Button
- Entry widget
- Text widget
- Radio button
- Check button
- List boxes
- Menus etc.

Network Programming

- Introduction
- Protocols
- Sockets
- Connect to Server
- Sending data
- Receiving data
- Handling Connections
- Email Sending Application
- SMS sending application

Our top priority is to strengthen your logical thinking and prepare you to tackle programming challenges across various platforms.

Every topic will be presented with multiple examples.

Learn how to analyse a problem to reach the best possible solution.

Topics listed are not in the order they will be covered.
