

Pythan Advanced Programming (Review of Python)

Interactive Mode: Interactive Mode, as the name suggests, allows us to interact with OS.

Script Mode: In script mode, we type Python program in a file and then use interpreter to execute the content of the file.

Number: Number data type stores Numerical Values.

Sequence: A sequence is an ordered collection of items, indexed by positive integers.

Arithmetic operators: +, -, *, /, %, ** and //.

Relational operators: <, <=, >, >=, != or <> and ==.

Logical operators: or, and, and not

Assignment Operator: =, +=, -=, *=, /=, %=, **= and //=

Functions in Python: A function is named sequence of statement(s) that performs a computation.

Module: A module is a file containing Python definitions (i.e. functions) and statements.
Standard library

of Python is extended as module(s) to a Programmer.

String: In python, consecutive sequence of characters is known as a string. An individual character in a string is accessed using a subscript (index).

List: Like a string, list is a sequence of values. List can be of any type.

Dictionaries: A dictionary is like a list, but more in general. In a list, index value is an integer, while in a dictionary index value can be any other data type and are called keys.

Tuples: A tuple is a sequence of values, which can be of any type and they are indexed by

integer.

(Concept of Object Oriented Programming)

Object: clearly defines an entity in terms of its properties and behaviour.

Class: a blueprint of an object.

Encapsulation: combining of data and the functions associated with that data in a single unit

Data Hiding: the mechanism of hiding the data of a class from the outside world

Abstraction: providing only essential information to the outside world and hiding their background details

Inheritance: forming a new class (derived class) from an existing class (called the base class).

Polymorphism: ability to use an operator or function in various forms.

Static Binding: the linking of function call to the function definition is done during compilation of the program.

Dynamic Binding: the linking of function call to the function definition is done during the execution of the program.

(Classes in Python)

Namespace: A mapping from names to objects. Examples of namespaces are built-in names, global names in a module and local names in function invocation

Scope: A region of Python program where a namespace is directly accessible.

In Python a name, storing any type of data, can refer to only one thing at a time.

The scope of a variable is its enclosing function, class or file.

The names always belong to the namespace where they are bound.

Names declared with global keyword have to be referred at the file level.

LEGB rule: when a name is encountered during the execution of the program, it searches for that name in the following order:

L. Local - It first makes a local search i.e. in current def statement.

E. Enclosing functions - It searches in all enclosing functions, from inner to outer.

G. Global (module) - It searches for global modules or for names declared global

B. Built-in (Python) - Finally it checks for any built in functions in Python.

Class definitions should be given before it is referenced.

`__init__` is a special method used to initialize the members of a class.

`self` is the first argument that is passed to the methods of a class.

A class object can be used in two ways - Instantiation and Attribute reference Class attributes belong to the class and will be shared by all instances Instance attributes belong to a particular instance of a class only.

The attributes - data and methods can be added to the class dynamically.

`getattr(obj, name,[default])`: is used to access the attribute of the object

`hasattr(obj,name)`: is used to check if an attribute exists or not

`setattr(obj,name,value)`: is used to set an attribute with a value.

`delattr(obj,name)` : is used to delete an attribute

`__dict__` : gives the dictionary containing class namespace

`__doc__`: returns the docstring of a class

`__name__`: it gives the class name

`__module__`: specifies the module name in which the class is defined

`__bases__`: it gives a tuple containing base classes

`__del__`: is invoked when the module is being deleted

`__str__`: returns the string representation of the objects Private variables can only be accessed from inside the objects.

Name Mangling: A name is prefixed with two leading underscores and no more than one trailing underscore.

Static Method: is a method that does not obey the usual convention in which self, an instance of the class, is the first argument to the method.

Python uses two strategies for memory allocation- Reference counting and Automatic garbage collection.

Reference Counting: works by counting the number of times an object is referenced by other objects in the system. When an object's reference count reaches zero, Python collects it automatically.

Automatic Garbage Collection: Python schedules garbage collection based upon a threshold of object allocations and object de-allocations. When the number of allocations minus the number of deallocations are greater than the threshold number, the garbage collector is run and the unused block of memory is reclaimed.

(Inheritance)

Inheritance: In object oriented programming, inheritance is a mechanism in which a new class is derived from an already defined class. The derived class is known as a subclass or a child class. The pre-existing class is known as base class or a parent class or a super class.

Single Inheritance: In single inheritance a subclass is derived from a single base class.

Multilevel Inheritance: In multilevel inheritance, the derived class becomes the base of another class.

Multiple Inheritance: In this type of inheritance, the derived class inherits from one or more base classes.

Hierarchical Inheritance: In this type of inheritance, the base class is inherited by more

than one class.

Hybrid Inheritance: This inheritance is a combination of multiple, hierarchical and multilevel inheritance.

Overriding Methods: The feature of overriding methods enables the programmer to provide specific implementation to a method in the subclass which is already implemented in the superclass.

Abstract Methods: An abstract method is a method declared in a parent class, but not implemented in it. The implementation of such a method can be given in the derived class.