**PYTHON**

**Data Types**

1. Integers (int): Whole numbers without decimal points

e.g., 5, -10, 1000.

2. Floats (float): Numbers that have decimal points.

e.g., 3.14, 2.71828, 0.5.

3. Strings (str): Ordered sequences of characters enclosed in single.

double, or triple quotes, e.g., 'hello', "world", '''Python'''.

4. Booleans (bool): Represents truth values True or False.

5. Lists (list): Ordered, mutable collections of elements enclosed in square brackets.

e.g., [1, 2, 3], ['apple', 'banana', 'orange'].

6. Tuples (tuple): Ordered, immutable collections of elements enclosed in parentheses.

e.g., (1, 2, 3), ('apple', 'banana', 'orange').

7. Dictionaries (dict): Unordered collections of key-value pairs enclosed in curly braces

e.g., {'name': 'Alice', 'age': 25}.

8. Sets (set): Unordered collections of unique elements enclosed in curly braces.

e.g., {1, 2, 3}, { 'apple', 'banana', 'orange'}.

**Functions**

A function is a block of code that performs a specific task. There are two types of functions,

· Standard library functions

Ex: print ()

· User defined functions

Ex: sum1()

An argument is a value accepted by function.

· Argument with default values

· Keyword argument

· Arbitrary argument

· Arbitrary keyword argument

**Lambda function:**Lambda function is a special type of function without function name.

**Higher order function**: A function that operate with another function or it contains other functions as a parameter, returns a function as output.

Ex: Filter (), Map (), Reduce()

**Recursive function**: A function which calls itself is known as recursive function.

**Scope**

A variable scope specifies the region where we can access a variable. Based on scope, we have

· Global

· Local

· Nonlocal

**Conditional statement**

* **if\_statement:**The if statement is used to check a condition and execute a block of code if the condition is true.
* **If\_else statement:**The else statement is used to execute a block of code if the if condition is false.
* **If-elif-else statement**:The elif statement allows you to check multiple conditions.
* **While:** The while loop executes a block of code as long as the specified condition is true.
* **For:** The for loop is used to iterate over a sequence (such as lists, tuples, strings, etc.) or any iterable object.

**Object oriented programming**

python is an object oriented language like any other general purpose programming language

The concept of OOP in python focuses on creating "objects" and making reusable codes.

**Class:**

* a class is a collection of objects or we can say as user defined datatypes
* a class also holds the methods and attributes of the object.

**Object:**

* an object(instance) is known as instantiation of class
* an object is a group of interrelated variables and methods
* when a class is defined only the description of the object is defined therefore no memory or storage is allocated at that point

**Methods:**

* methods are function defined inside the body of a class
* the method is a function that is associated with an object
* in python, a method is not unique to class instances.any object type can have methods
* methods define the behavior of an object

**Inheritance:**

* inheritance is creating a new class that uses all the properties and behavior of another class
* the new class formed is called derived class(or child class) and the existing class whose properties are inherited is known as a base class(or parent class)
* it provides reusability of code

**Polymorphism:**

* poly means many and morphs means shape
* polymorphism means using a familiar interface for multiple forms(data types). in single terms we understand that a task can be performed in various ways
* + operator is used for integer data types for arithmetic addition operations, whereas for string data types, + operator is used to perform concatenation
* this is one of the examples in python

**Encapsulation:**

* using oop in python,we can restrict access to methods and variables
* Python doesn't have any private keywords unlike java. this prevents data from direct modification which is called encapsulation
* in python we denote private attributes using underscore as the prefix ie, single underscore or double underscore

**Abstraction:**

* data abstraction and encapsulation are often used as synonyms
* abstraction is used to hide internal details and show only functionalities
* eg:you know how to turn on/off light but you don't know what is happening inside
* an abstract class cannot be instantiated which means you cannot create objects for the class.it can only be used for inheriting the functionalities