**Python Programing**

* What is Programing

Programing is the way for us to tell computers to what to do. Computer is a dumb machine and it does what we tell it to do. Hence, we learn programming and tell computers to do what we are very slow at doing.

* What is Python

Python is a dynamically typed, general purpose programming language that supports an object-oriented programming approach as well as a functional programming approach. Python is an interpreted and a high-level programming language.

* Features of Python

Python is simple and easy to understand. It is Interpreted and platform-independent which makes debugging very easy. Python is an open-source programming language. Python provides very big library support. Some of the popular libraries include NumPy, Tensorflow, Selenium, OpenCV, etc. It is possible to integrate other programming languages within python.

* Uses of Python

Python is used in Data Visualization to create plots and graphical representations. Python helps in Data Analytics to analyse and understand raw data for insights and trends. It is used in AI and Machine Learning to simulate human behaviour and to learn from past data without hard coding. It is used to create web applications. It can be used to handle databases. It is used in business and accounting to perform complex mathematical operations along with quantitative and qualitative analysis.

* Modules and pip in Python

Module is like a code library which can be used to borrow code written by somebody else in our python program. There are two types of modules in python

1. Built in Modules - These modules are ready to import and use and ships with the python interpreter. there is no need to install such modules explicitly
2. External Modules - These modules are imported from a third-party file or can be installed using a package manager like pip or conda. Since this code is written by someone else, we can install different versions of a same module with time.

* Python programming

1. Print statement: Print statement useful to when we want to print to something to console / output.

# print (“type something here”) =>values, end, sep, file, flush.

1. Comments: A comment is a part of the coding file that the programmer does not want to execute, rather the programmer uses it to either explain a block of code or to avoid the execution of a specific part of code while testing.

# Hash symbol is used for the comments. For multiple we can create the multiline string and leave with out assigning to a variable.

“”” Like this we can use the multi-line comments”””

1. Escape Sequence Characters: In programming languages, we have special meaning for the symbols. While we are using that be careful while using them. “\” Back slash will one among them it will change the meaning of next character.

# print (‘I’m a good person.’) 🡪 It is not valid since single quote in the middle of the string we ignore that special meaning using back slash. # print (‘I\’m a good person.’)

1. Variables and Data Types: Variable is like a container that holds data. Creating a variable is like creating a placeholder in memory and assigning it some value.

# a = 1

# b = True

# c = “Apple”

# d = None

Data type specifies the type of value a variable hold. This is required in programming to do various operations without causing an error.

# Numeric data => Integer, Float, Complex

# Text data => String

# Boolean data => True, False

# Sequence data => List, Tuple

# Mapped data => Dictionary

1. Arithmetic operators: Arithmetic operators which will help to do mathematical calculations.

# Addition => 5 + 5 = 10

# Subtraction => 6 – 3 = 3

# Multiplication => 5 \* 3 = 15

# Division => 6 / 3 = 2.0 This is normal division

# Floor division => 5 // 2 = 2 This is floor division which will int as the output.

# Exponent => 2 \*\* 3 = 8

# Modulo => 5 % 2 = 2 This will return reminder value as the output.

1. Type casting: The conversion of one data type into the other data type is known as type casting in python or type conversion in python. In python we have 2 types of type casting are there
2. Explicit typecasting - The conversion of one data type into another data type, done via developer or programmer's intervention or manually as per the requirement, is known as explicit type conversion.

# a = input (“Enter number: “) By default it is string we need to convert as the integer. # a = int(a)

1. Implicit type casting - Data types in Python do not have the same level i.e. ordering of data types is not the same in Python. Some of the data types have higher-order, and some have lower order. While performing any operations on variables with different data types in Python, one of the variable's data types will be changed to the higher data type. According to the level, one data type is converted into other by the Python interpreter itself (automatically). This is called, implicit typecasting in python.

# a = 10, b = 10.5 , c = a + b, Automatically c is float.

1. Input function: System required any information from the user to perform certain actions we will provide using input function.

# Input (“Type something…”)

1. Strings: Strings is a series characters which are arranged in the sequential manner. We can print the string inside the double quotes or single quotes.

# name = “apple”

# print (name) => apple

In some times we need to print the quotes in the strings in that scenario we can use either escape character or using alternative solutions.

# print (‘I \’m a good person’) => Back slash will remove special meaning for the single quote inside the string.

#print (‘He said, “I want to have apple everyday”’)

Multiline string: When we want print multiple lines of statements we will use multiline string.

Unicode = “Unicode separates the code points from the details of the encoding system. This permits a much wider range of characters up to four bytes. The Unicode character set incorporates the entirety of the ASCII character set as the first 127 characters. All ASCII characters have the same code points in both encodings”

# print (Unicode)

Accessing Characters of a String: We can access the string characters by using of index.

# name = “apple”, print (name [0]) – a

Looping through string: Since string will support for the indexing we can loop through string.

# for char in name:

Print(char)

String slicing: In side a string we want to access some series of characters we will do with help of slicing.

# name [1: 4: 2] 1- starting index, 4- final index, 2- step definition. In python it will always consider lower index and it won’t consider upper index.

# name [ 1:]

# name [: 5]

String operations: To perform certain actions we depend on string methods.

Name = “apple”

1. Length of string: To calculate the length of string.

# Print (len (name)) - 5

1. Upper: It will convert all characters in a string to upper case.

# name. upper () - APPLE

1. Lower: It will convert all characters in a string to lower case.

# name. lower () – apple, if any upper case will be there it will convert to the lower case.

1. Strip: Strip method will remove the extra white spaces in front and end of the string.

# name = “ apple “

# name. Strip (), print(name) – apple

1. Rstrip: This method will remove any tailing characters with specific arguments.

# name = “apple!!!!”

# name. rstrip (“!”) – print(name) – apple

1. Replace: Replace method will replace some characters with some other.

# name. replace (“a”, “b”) – bpple

1. Split: Split function will break the string into parts based on argument given.

# name. split(“l”) – [“app”, “e”]

1. Capitalise: Capitalise method will give first letter of string to the upper case.

# name. capitalise () – Apple

1. Center: Center method will align the string to the center based on the argument length.

# name. center (50) - “This are come with the space” apple

# name. center (50, “.”) - ……………. apple……………………...

1. Count: Count method will count the number of occurrences of character in a string.

# name. count (“l”) – 2

1. Endswith: If any string trailing characters are same as argument it will give Boolean value.

# name. endswith(“e”) – True

# name. endswith (“l”, 0, 2) – True, we can check for the sub strings also with the same manner.

1. Find: Find method will check for the first occurrence of the element. If element found will return index value else return

-1.

# name. find (“u”), -1 because element not found. Else it will give us the index.

1. Index: Index method will give us the index of the element other it will raise exception.

# name. index(“a”) – 0

1. Isalpha: This method will return the Boolean value if string only consist of A-Z, a – z, 0-9.

# name. isalpha () – True

1. Islower: This method returns Boolean value if all characters in a string are lower case.

# name. islower () – True

1. Isprintable: This method will return the Boolean value if all characters in a string are printable.

# name. isprintable () – True

1. Isspace: String will consist any space it will return Boolean value.

# name. isspace () – False

1. Istitle: Every word in a string first letter will be the upper case based on that it will return Boolean value

# name. istitle () – False

1. Isupper: Every character in a string are upper case it will return Boolean value.

# name. isupper () – False

1. Startswith: String start with the given argument it will return the Boolean value.

# name. startswith (“a”) – True

1. Swapcased: This method will return all lower case characters to upper case and upper case characters to lower case.

# name. swapcase () – APPLE

1. Conditional statements: Conditional statements are used to evaluate the expression. Based on expression boolean value program will take different paths while execution.
2. If – else condition: Generally, if the expression is true it will go to if block else expression will go to else block.

# If True:

Execute the code which is present in this block

Else:

Execute the code which is present in this block

1. If – elif - else condition: If want to check for the more expressions then we case elif block in between if and else block. We can use multiple elif block as per requirement.

# If True:

Execute the code which is present in this block

Elif True:

Execute the code which is present in this block

Else:

Execute the code which is present in this block

1. Nested if – else condition: If multiple expressions need to check inside the expression, we can use the nested if – else conditions.

# If True:

Execute the code which is present in this block

Elif True:

If True:

Execute the code which is present in this block

Elif True:

Execute the code which is present in this block

Else:

Execute the code which is present in this block

Else:

Execute the code which is present in this block

1. Match case:
2. For loops: Sometimes as a programmer I want to execute some lines of code or some functions multiple times we can use loops.

We can make it in few lines of code only and code redundancy is not accepted in programming.

1. Iterating over the string: Strings will support indexing by using of for loops we can iterate over the strings.

# name = “apple”

For i in name:

Print(i)

1. Iterating over the list: We want to traverse through each element in an array we will use for loop

# my\_list = [“apple”, “bat”, “cat”, “dog”]

For item in my\_list:

Print(item)

1. Using range function: We want to fetch item or numbers with in that range we can use function.

# for I in range (0, 6, 2):

Print(i) 0 - lower index, 6 – upper index, 2 step definition.

1. While Loop: While loop runs till condition become false. Some times it led to infinite loops as well it will create memory issues while using while loop, we need to careful with the condition.

# While True:

Loop running.

Else block with while loop: We can use else block with the while loop after loop executed successfully else block will execute.

# While True:

Loop running.

Else:

Print something.

1. Do – while loop: In python we don’t have any such concept like do while loop. How it will work means initially it will print value then check for the condition.

# I = 0:

While True:

Print (i)

If I > 0:

Break

1. Break, continue statements: These statements will use in loops if we want pass current iteration will use continue statement. If want to pass the whole loop will use break statement. There are very useful while working with the loops.
2. Functions: Functions are some of lines of code which we can call whenever it is required to perform that action. In large programs we need to perform some operations multiple times here functions play vital role. And any changes need to be made in that function instead of changing every where we can fix only one place it will give us proper result. There are 2 types of functions are there in python
3. In built functions: In python there are some in built functions are there we can use them as per user requirement it comes along with the python installation.

# max (), print (), input ().

1. User defined function: User defined functions are the function which are created by the user as per their requirement.

# def user\_defined\_function(arguments):

Pass

User\_defined\_function () => calling function

1. User defined function types: There are 4 types of functions are there we can classify the user defined functions.

Default arguments: We can provide the default arguments while creating function even while calling function if we are not provided argument also it considers from function.

Key word arguments: In key word argument while calling the function we can call argument with the parameter. Like key = value, we need to provide the all arguments.

Required arguments: In required arguments we need to provide each argument which were define in the function.

Variable length arguments: Like we need to provide more arguments we don’t know while defining function in that condition we can use the variable length arguments. While calling function we can provide as many arguments we required.

# def variable\_length\_argument(\*arguments):

Keyword variable length arguments: It is same like variable length arguments only in this case we will use the key words as well while calling function.

# def key\_word\_arguments(\*\*arguments):

Return statement: The return statement is used to return the value of the expression back to the calling function.

1. Python Lists: Python list are the data structures where we can store the multiple values in the single variable. And we can store different data types and it is ordered collection and mutable data structure. List will support for the indexing and slicing. While slicing it will create new memory address.

List indexing: List will support for the indexing we can access the variable element in the list using index and we can loop over the list and we can perform slicing operations on the list.

# List [2], Access the element using the index

# for I in list, Loop through the list.

List Comprehension: List comprehension is the way of define the list with some other list and do some modification.

# list = [i \* i for I in range (1, 10)]

List methods: When ever we want to perform some operations in the list will have some list methods.

# sort, this will sort elements in the list.

# reverse, this method will reverse the order of the list.

# index, this method will element index in the list.

# count, this method will give us count the particular element occurrence.

# copy, this method will do shallow copy of the list.

# append, adding element at the end of list.

# insert, inserting element at the given index.

# extend, this method will add other iterable to the list.

1. Tuples: Tuples are order collection data structures same like as the lists. There is small difference between the list and tuples, tuples are immutable due to which bit faster compare with the list.

Tuple methods: Since tuples are immutable in nature, we have only few methods

# index, it will return index value of element in the tuple

# count, it will give particular element occurrence in the tuple.

1. String formatting: Combining string with the integer will lead to Type error. When ever we are combining string with the integer before embedding, we need to convert inter to string then we can in add in string. There some ways to do string formatting

# Using format method

# using f strings

1. Doc Strings: In python we need to provide any information to user like what that function will do what uses of that function will do with the help of doc strings. We will define doc strings immediate after the function method or class.

# Print (method. \_\_doc\_\_), This print doc string to the console

1. Recursion: The function calling it self with small change in parameter until base condition is satisfied this one, we will call it as the recursion.

# Recursion help us to find factorial of number, Fibonacci sequence.

1. Set: Set is an un ordered, mutable data structure it won’t allow duplicates. Here we have some set methods we can discuss all those things.

# union: union method will give unique elements in both sets.

# update: it will update one set to the other set.

# intersection: common elements in the both the sets

# intersection update: it will set to common elements in sets

# symmetric difference: set without common elements

# symmetric difference update: update set without common elements

# difference: elements which are not there in other set

# difference update: update the set elements which are not in other set.

# isdisjoint: no common element it will return true else false

# is superset: all elements upper set will return true

# is subset: all elements in the lower set it will return true.

# add: adding element to the set

# update: update one set with the other set

# remove / discard: remove element from the set

# pop: remove element from the set

# del: deleting the set

# clear: clear all the elements in the set

1. Dictionary: Dictionary is order data structure where we can store the values in key value pairs. All keys in dictionary al unique. If we want to perform any operation over the dictionary with help of dictionary methods we can perform

# dict [‘key’], Return value from the dictionary.

# dict. Get (‘key’)

# dict. Keys (), it will return all keys.

# dict. Values (), it will return all values.

# dict. Items (), return both the key and value pairs.

#dict. Update (), update key, value to the dict

# dict. Clear (), clear all elements from the dict.

# dict. Pop(key), remove the element from the dict

# dict. Popitem (), remove last key, value pair from the dict.

# dict. Del (), Delete the key, value pair from the dict

1. Exception handling: Exception handling is the process of responding to unwanted or unexpected events when a computer program runs. Exception handling deals with these events to avoid the program or system crashing, and without this process, exceptions would disrupt the normal operation of a program.

Exception in python: Python has many built-in exceptions that are raised when your program encounters an error

When these exceptions occur, the Python interpreter stops the current process and passes it to the calling process until it is handled. If not handled, the program will crash.

Try – Catch block: try…. except blocks are used in python to handle errors and exceptions. The code in try block runs when there is no error. If the try block catches the error, then the except block is executed.

# Try:

Number = input (“Enter the number: “)

Except ValueError:

Print (“Please enter the proper number”)

Finally Block: Final block is the also part of the try except block finally we declare the final block which will useful to print out something at the end irrespective of which block is executed.

Custom Exception: We can raise custom exception using of the raise key word in program and we can create the custom exception as well it will inherit property from the exception class.

1. Shorthand conditional statement: We can write the conditional statement in single line this one we can call it as the shorthand if – else condition.

# “Even number” if num % 2 == 0 else “Odd number”

1. Enumerate function: Enumerate function will useful while working iterables and we need to know index of that iterable by using enumerate function.

# for I, item in enumerate (list, start = 1):

1. Virtual Environment: A virtual environment is a tool used to isolate specific Python environments on a single machine, allowing you to work on multiple projects with different dependencies and packages without conflicts.

# python -m venv “Environment name”, Create a new virtual environment/

# myenv\Scripts\activate.bat, Activating virtual environment.

# Deactivate, Deactivating virtual environment.

# pip freeze > requirements.txt, get requirement file from team member or another person using the environment.

# pip install -r requirements.txt, install the requirements.

1. Importing in python: Import is the process of using existing modules functions methods in the current script. There are several ways are there to import python packages.

# import math, importing with the package name.

# import math as m, we can call with the alias name as well.

# from math import \*, This will load all methods and variables.

# dir function, It will return all methods inside of that module.

1. If \_\_name\_\_ == “\_\_main\_\_” : This method will help us identifying the method is using directly or importing from the some other module we can able to identify. If we are not using that particular method while importing packages it will run the multiple output being loaded from the importing module.
2. OS Module: The os module in Python is a built-in library that provides functions for interacting with the operating system. It allows you to perform a wide variety of tasks, such as reading and writing files, interacting with the file system, and running system commands.
3. Local vs Global: Defining variable inside the function is called as the local variable and we can access that variable in side that particular function only. Define variable outside the function is know as the global variable and we can access this function anywhere from the program. The global keyword inside function will help us to change the global variable inside function.
4. File operations: We can perform various operation in the files such as the read, write, append operations. We can perform write and read, append, create, text, binary operations on the files. And we have with keyword using with we can perform operations on the files in these we no need to close the files.
5. Read, read lines, Write lines: In any text files we want to read the information line by line we can use read line method we can print all the information using of while loop we have read lines method as well, we can use that method as well. We have write lines method as well we can write something to to the file if we need to write anything to next line we can use the escape character.
6. Seek, tell, truncate: Seek will give us the information about the what is there after particular lines of code. Tell will give us current position while writing any information. Truncate will help us to display the information after trimming.
7. Lambada function: Lambada is an anonymous function which will take the no. of input and give us the single output.

# lambda arguments: expression

1. Map, filter, reduce: Map will help us to connecting function with the iterable and filter will remove some values in iterable based on the condition, reduce function will take min 2 inputs to any number of inputs and give us the result.
2. Is vs ==: These are the comparison operators which will help us to compare one value to other value. Is will compare the memory address of the variables and == will compare the values of the variables.

* Object – Oriented programming: Object oriented programming is the programming paradigm by using which we can use this approach as the template and work on things.

There are mainly two approaches are there in the programming which are

1. Functional programming
2. Objected oriented programming

Objected oriented programming in which we are handing the data with help of the classes and objects. Class is defined as the blue print or template by using class we can create as many objects we want object referred as the real word entity. In object-oriented programming variables referred as the attributes and functions referred as the methods.

Self-keyword: Self key word is used to represent the current instant of the class.

Constructor: Constructor is the special method which is used to initialise the objects. Once class will be called constructor will be invoked automatically.

There are two types of constructors will be there

Default constructor: This constructor won’t take any argument other than self keyword and it will be return what is there in side constructor.

Parametrised constructor: This constructor will take arguments along with the self key word and these arguments can be used inside of the class.

Decorators: Decorators are the functions which can change the run time behaviour of the existing function without changing the existing function code.