## What is Python?

1. Python is an interpreted, object-oriented, high-level programming language and hence it is a programmer friendly language.
2. Python's simple, easy to learn and easy to read its just like English language.
3. Python supports modules and packages, which encourages program modularity and code reuse.

(Interpreted:- An interpreted language is a programming language that means it’s source code converted into byte code then executed by PVM )

## What is Limitations?

1. Performance wise not up to the mark because it is interpreted language .
2. Not using for mobile Application and Game Development.
3. Python is interpreted language and is slow as compared to c/c++ and java.
4. Memory allocation bcoz of dynamically typed.

## What are the benefits of Python?

1. Presence of third-party modules.

2. Extensive support libraries(NumPy for numerical calculations, Pandas for data analytics etc)

3. Open source and community development

4. Versatile, Easy to read, learn and write

5. User-friendly data structures

6. High-level language

7. Dynamically typed language(No need to mention data type based on the value assigned, it takes data type)

8. Object-oriented language

9. Portable and Interactive

10. Ideal for prototypes – provide more functionality with less coding

11. Highly Efficient(Python’s clean object-oriented design provides enhanced process control, and the language is equipped with excellent text processing and integration capabilities, as well as its own unit testing framework, which makes it more efficient.)

12. (IoT)Internet of Things Opportunities

13. Interpreted Language

14. Portable across Operating systems

## What is PEP8 and why it is Important?

[(**https://www.python.org/dev/peps/pep-0008/**]((https:/www.python.org/dev/peps/pep-0008/))

The PEP is an abbreviation form of **Python Enhancement Proposal**.

**PEP 8 enhances the readability of the**[**Python**](https://www.javatpoint.com/python-tutorial)**code.**

Writing code with proper logic is a key factor of programming, but many other important factors can affect the code's quality. The developer's coding style makes the code much reliable, and every developer should keep in mind that Python strictly follows the way of order and format of the string.

1. Code Layout
2. String Quotes
3. WhiteSpace in Expression and Statement
4. Trailing Commas
5. Comments
6. Naming Convention
7. Programming Recommendation

## What is the Scope of Python?

A variable is only available from inside the region it is created. This is called scope.

1. A variable created inside a function belongs to the local scope of that function, and can only be used inside that function.
2. As explained in the example above, the variable x is not available outside the function, but it is available for any function inside the function.
3. A variable created in the main body of the Python code is a global variable and belongs to the global scope. Global variables are available from within any scope, global and local.

## What are the Packages and Modules in Python?

**Modules:-**

A group of functions, variables and classes saved to file, which is nothing but modules.

Every python (.py) file is module.

**nPackages:-**

Package is nothing but folder or directory which represents collection of Python modules. It is an encapsulation mechanism to group related modules into a single string.

Any folder or directory contains \_init\_.py file, is consider as python package. This file can be empty.

Advantages:-

1. We can improve naming conflicts.
2. We can identify our components uniquely.
3. It improves modularity of the application.

## What are the Global/Public, Protected and Private attribute in Python?

By default every attribute is public. We can access from anywhere either within the class or from outside the class.

Eg.:🡪 name=”Ritesh”

Protected attribute can be accessed within the class anywhere but from outside of the class only in child class. We can specify an attribute as protected by prefixing with \_symbol.

E.g:🡪

\_variablename=value

\_name=”Ritesh”

Private attribute can accessed only within class i.e. from outside of the class we cannot access. We can declare a variable as private explicitly by prefixing with 2 underscore symbol.

E.g:🡪 \_\_variablename=value

## What is the doc string in Python?

Documentation String represent the description of the class. Within the class doc string is always optional . We can get doc string by using following two ways.

1 . print(classname.\_\_doc\_\_) 2. Help(classname)

**Syntax:-**

**class Student:**

**“””This is Student class with required Data”””**

**print(Student.\_\_doc\_\_)**

**help(Student)**

## How Memory Managed in Python?

Python is used a dynamic memory allocation which is managed by the Heap data structure. Memory Heap holds the objects and the other data structures that will be used in programming.

## What is Pickling and Unpickling?

The process of writing state of object to the files is called pickling and the process of reading state of an object from the file is called as Unpickling.

We can implement Pickling and Unpickling by using pickle module of Python.

Pnickle module **dump()** function to perform pickling.

E.g.:🡪 **pickle.dump(Object,file)**

Pickle module contains **load()** function to perform unpickling.

E.g. **:🡪 obj=pickle.load(file)**

## What is PythonPath in Python?

PythonPath is an environment variable which you can set to add additional directories where python will look for modules and packages

## What is “Help” and “DIR” function?

help():🡪 return the object of the module

If no argument is given “help()”, the interactive help system starts on the interpreter consol.

If argument is given then it shows its documentation string along with modules, keyword and attribute.

>>> def myfunc():

... '''This is myfunc'''

... x = 2

... return x

...

>>> help(myfunc)

Help on function myfunc in module \_\_main\_\_:

myfunc()

This is myfunc

dir():🡪 return all properties and method of the specified object, without values.

Without arguments, it returns the list of names in the current local scope.

With an argument , attempt to return a list valid attributes for the object.

>>> dir()

['\_\_annotations\_\_', '\_\_builtins\_\_', '\_\_doc\_\_', '\_\_loader\_\_', '\_\_name\_\_', '\_\_package\_\_', '\_\_spec\_\_', 'myfunc']

>>> dir(myfunc)

['\_\_annotations\_\_', '\_\_call\_\_', '\_\_class\_\_', '\_\_closure\_\_', '\_\_code\_\_', '\_\_defaults\_\_', '\_\_delattr\_\_', '\_\_dict\_\_', '\_\_dir\_\_', '\_\_doc\_\_', '\_\_eq\_\_', '\_\_format\_\_', '\_\_ge\_\_', '\_\_get\_\_', '\_\_getattribute\_\_', '\_\_globals\_\_', '\_\_gt\_\_', '\_\_hash\_\_', '\_\_init\_\_', '\_\_init\_subclass\_\_', '\_\_kwdefaults\_\_', '\_\_le\_\_', '\_\_lt\_\_', '\_\_module\_\_', '\_\_name\_\_', '\_\_ne\_\_', '\_\_new\_\_', '\_\_qualname\_\_', '\_\_reduce\_\_', '\_\_reduce\_ex\_\_', '\_\_repr\_\_', '\_\_setattr\_\_', '\_\_sizeof\_\_', '\_\_str\_\_', '\_\_subclasshook\_\_']

## What is difference between “.py” and “.pyc ” files?

Py files contain the source code of the program, whereas .pyc files contains bytecode of the program.

pyc files are created by the Python interpreter when a . py file **is imported**.

pyc files are placed in the same directory as the . py file. In Python 3.2, the compiled files are placed in a **\_\_pycache\_\_ subdirectory**, and are named differently depending on which Python interpreter created them.

## Why Python is Dynamically Typed Language?

In Python we are not required to declare type of variable. Whenever we are assigning the value, based on value, type will be allocated automatically. Hence Python is considered as dynamically type Language.

## How Python is Interpreted?

We are not require to compile Python programs explicitly. Internally Python Interpreter will take care that compilation. (the source code of a python program is converted into byte code that is then executed by PVM ).

## What is the difference between the Break, Pass and Continue keyword?

Break, Pass and Continue are the Transfer statement.

**Break**:🡪 We can use a break statement inside loops to break loop execution based on some condition.

Eg:-

for I in range(10):

if i==7:

print(“process is enough… break the statement”)

break

print(i)

o/p:🡪 0 1 2 3 4 5 6

**Continue**:🡪 We can use continue statement to skip current iteration and continue next iteration.

Eg:-

Cart=[10,20,30,40,50,60,70,25,21,36]

for item in cart:

if item>=50:

print(“We cannot process this item:”,item)

continue

print(item)

o/p:- 10 20 30 40

We cannot process this item : 50

We cannot process this item : 60

We cannot process this item : 70

25,21,36

**Pass:🡪**  Pass is keyword in Python. In our program systematically if block is required which won’t do anything then we can define that empty block with pass keyword.

Eg:🡪

For I in range(100):

If i%9==0:

Print(i)

Else:

Pass

o/p:🡪 9 18 27………..99

## What is the difference between Shallow copy and Deep Copy?

This are the assignment statement in Python do not copy objects, they create bindings between a target and an object.

The difference between shallow and deep copying is only relevant for compound objects.

🡪A shallow copy constructs a new compound object and then insert reference into the objects found in the original.(changes reflect in the original list)

l2=copy.copy(l1)

🡪 A Deep copy construct a new compound object and then, recursively, inserts copies into it of the objects found in the original. (changes not reflect in the original list)

L1=Copy.deepcopy(l1)

Eg:🡪

import copy

l1=[1,2,3,[4,5],6]

#l2=copy.deepcopy(l1)

l2=copy.copy(l1)

print(l1)

print(l2)

l2[3][0]=7

print("Change in list l2",l2)

print(l1)

## What is List and Tuple? Internal implementation of List?

If we want to represent a group of individual objects as a single entity where insertion order is preserved and duplicate are allowed then we go for List.

Tuple is same as List except that it is immutable. i.e. once we create Tuple object, we cannot perform any changes in that.

|  |  |
| --- | --- |
| List | Tuple |
| List is a Group of Comma separeated Values within Square Brackets and Square Brackets are mandatory. Eg: i = [10, 20, 30, 40] | Tuple is a Group of Comma separeated Values within Parenthesis and Parenthesis are optional. Eg: t = (10, 20, 30, 40) t = 10, 20, 30, 40 |
| List Objects are Mutable i.e. once we creates List Object we can perform any changes in that Object. Eg: i[1] = 70 | Tuple Objeccts are Immutable i.e. once we creates Tuple Object we cannot change its content. ValueError: tuple object does⎝t[1] = 70 not support item assignment |
| If the Content is not fixed and keep on changing then we should go for List | If the content is fixed and never changes then we should go for Tuple. |
| List Objects can not used as Keys for Dictionries because Keys should be Hashable and Immutable | Tuple Objects can be used as Keys for Dictionries because Keys should be Hashable and Immutable. |
| [ ] | { } |

## What is meant by Decorators, Generator and Iterators?

**Decorators:🡪** Decorators is a function which can take a function as argument and extends its functionality and returns modified function with extended functionality.

Generators:🡪 Generators is a function which is responsible to generate a sequence of values. We can write generator functions just like ordinary functions, but it uses yield keyword to return values.

**Iterators:🡪** Iteratorin python is an object that is used to iterator over iterable objects like list, tuple, dict and sets. The iterator objects is initialized used using the iter() method. It uses the next() method for iteration.

Eg:🡪

List=(“a”,”b”,”c”)

L1=iter(List)

Print(next(L1))

Print(next(L2))

Print(next(L3))

## What is meant by Slicing?

Slice means piece. [] operator is called slice operator, which can be used to retrieve parts of the string.

In python String follows the zero based index. The index can be either be +ve or –ve.

## What is \*args and \*\*kwargs??

args:🡪

Sometimes we can pass the variable number argument to our function, such type of argument are called as variable length argument. We can declare a variable argument with symbol “ \* ” as follows.

Def f1(\*num):

kwargs:🡪 keyword variable length argument

we can call this function by passing any number of keyword argument. Internally these keyword argument will be stored inside a dictionary.

def display(\*\*kwargs):

for k,v in kwargs.items():

print(k,"=",v)

display(n1=10,n2=20,n3=30)

display(rno=100,name="Durga",marks=70,subject="Java")

Output

n1 = 10

n2 = 20

n3 = 30

rno = 100

name = Durga

marks = 70

subject = Java

## Find the max accurance of given list? Or most repeated Number from List?

L=[1,3,2,56,76,3,4,5,34,5,4,6,8,67,8,4]

print(max(L))

max\_occurance = max(L, key = L.count)

print("Maxminum Occure Number is",max\_occurance)

print("Number of repetation",L.count(max\_occurance))

## What is NameSpaces in Python?

Name space is a collection of currently define symbolic names along with information about the object that each name references. In Python there are three types of namespace.

1. Built-In 2. Global 3. Local

Built\_In:🡪 The Built-In namespace contains the name of all built-in object present in the python.

Eg:-- True, False, id, input etc….

Global:🡪 The Global Namespace contains any names defined at the program or module level. The global namespace is created when the program starts and exists until the program is terminated by the python interpreter.

Local:🡪 A local namespace is defined for class, a function, a loop, or any block of code. The names defined in a block of code or function are local to it.