1. What are the new features added in Python 3.8 version?

In the version 3.8 below features has been added as below:

Valrus operator: that is := which can save some line of code as below

line = f.readline()

while line:

... # process line

line = f.readline()

Can now be written in short like this:

while line := f.readline():

... # process line

Also in 3.8 we can define the positional argument such as by defining a ‘/’ in a function definition.

Def fun1(x,y,c,/):

Pass

Now in 3.8 f string can support below kind of a statements:

print(f"foo={foo} bar={bar}")

## Now a built in function reversed() now works with dictionary as well.

Also now python interpreter can throw a syntax error warning at time of coding.

1. What is monkey patching in Python?

In Python, the term monkey patch refers to dynamic (or run-time) modifications of a class or module. In Python, we can actually change the behavior of code at run-time.

Example:

class A:

     def func(self):

          print ("func() is being called")

import monk

def monkey\_f(self):

     print ("monkey\_f() is being called")

# replacing address of "func" with "monkey\_f"

monk.A.func = monkey\_f

obj = monk.A()

# calling function "func" whose address got replaced

# with function "monkey\_f()"

obj.func()

1. What is the difference between a shallow copy and deep copy?

With the shallow and deep copy we are creating a new object itself or clone object not a object references.

As given below by importing the copy module:

import copy

# initializing list 1

li1 = [1, 2, [3,5], 4]

# using copy for shallow copy

li2 = copy.copy(li1)

# using deepcopy for deepcopy

li3 = copy.deepcopy(li1)

In the above code, the copy() returns a shallow copy of list and deepcopy() return a deep copy of list.

In case of deep copy, a copy of object is copied in other object. It means that **any changes** made to a copy of object **do not reflect** in the original object. In python, this is implemented using “**deepcopy()**” function.

In case of shallow copy, a reference of object is copied in other object. It means that **any changes** made to a copy of object **do reflect** in the original object. In python, this is implemented using “**copy()**” function

1. What is the maximum possible length of an identifier?

79 chars.

1. What is generator comprehension?

A generator comprehension is a single-line specification for defining a generator in Python

We can define the generator comprehension as below:

gen2 = (i for i in [1,2,3,4])

next(gen2)