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

**Ans:**

1. The walrus operator - Assignment expressions have come to Python with the “walrus” operator :=. This will enable you to assign values to a variable as part of an expression. e.g. while line:=f.readline():.

2. Positional-only arguments - A special marker, /, can now be used when defining a method’s arguments to specify that the functional only accepts positional arguments on the left of the marker.

3. f-strings now support “=” - print(f"foo={foo} bar={bar}")

4. reversed() now works with dict - list(reversed(my\_dict))

5. Simplified iterable unpacking for return and yield -

def baz():

rest = (4, 5, 6)

return 1, 2, 3, \*rest

6. New syntax warnings - The Python interpreter now throws a SyntaxWarning in some cases when a comma is missed before tuple or list.

1. What is monkey patching in Python?

**Ans:**

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.

# monk.py

class A:

def func(self):

print ("func() is being called")

import monk

def monkey\_f(self):

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

monk.A.func = monkey\_f

obj = monk.A()

obj.func()

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

**Ans:** Shallow copy creates a copy with the same reference of both of the objects means any change in one will make change in other, while deep copy create a copy with different reference means change in one will not make change in other.

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

**Ans:** 79 characters.

1. What is generator comprehension?

**Ans:**

Generator comprehension is same as list comprehension, difference is here we use paranthesis rather that square brackets. Below are example of both.

List comprehension - list\_comprehension = [i for i in range(11) if i % 2 == 0]

Generator comprehension - generator\_expression = (i for i in range(11) if i % 2 == 0)