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

ANS:

### 1. PEP 572 (Assignment Expressions) – The walrus operator (:=)

You will get to see a new type of operator which is being known as the walrus operator (:=).

This allows you to assign variables inside an expression. The major benefit of this is to save you some lines of code and you can write even cleaner and compact code in Python.

**Example of Walrus Operator in Python**

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

if( (size:=len(num)) < 10 ):

print(f”Length of list is small, size={size}”)

**Example 2**

This can be useful while writing loops, something like this.

Item = getItem()

while Item:

do\_something(item)

Item = getItem()

Can be written as:

while Item:= getItem():

do\_something(Item)

**Example 3**

Simplifying list comprehensions when filtering a value

[y for x in data if (y := f(x)) is not None]

### 2. PEP 570 (Positional only arguments)

There is a new function parameter syntax (/) to highlight that some of the functions must be stated positionally and not by keyword arguments.

We also have an operator (\*) that indicates that the arguments must be keyword only.

This can be a little confusing but after seeing the code this gets easier to understand, so let’s see them in action.

def func(a, b, c, d):

print(a,b,c,d)

We can call this function however we want with keyword arguments or positions.

func(d=2, a=3, b=2, c=6) #Valid - prints 3 2 6 2

func(1,3,4,5) #Valid - prints 1 2 4 5

Now consider the following definition:

def func( a,b,/,c,d,\*,e,f ):

print(a,b,c,d,e,f )

This will impose the way we can call this function:

* a and b arguments are positional only.
* c and d arguments can be positional as well as keyword.
* e and f arguments are keyword only.

func(1,2, 3,4, e=5, f=6 ) #Valid - prints 1 2 3 4 5 6

func(1,2, c=3,d=4, e=5, f=6 ) #Valid - prints 1 2 3 4 5 6

However, these calls will be invalid

func(a=1,b=2, 3,4, e=5, f=6 ) #Invalid - Error : a and b arguments are positional only

func(1,2, c=3,d=4, 5, 6 ) #Invalid - Error : e and f are keyword only arguments.

### 3. PEP 590 ( Vectorcall)

This release has made some improvements in the vector call which is a fast calling protocol for CPython.

A new C API is introduced to optimize the calls of objects.

This feature was already used in CPython but with the new C API, “fastcall” convention can be used by a user-defined extension class.

### 4. PEP 574 ( Pickle Protocol 5 with out-of-band data)

Pickle is useful to transfer big amounts of data between Python processors to take full advantage of multicore processors.

It’s important to maximize the transfer speed by optimizing memory copies. Pickle protocol 5 now supports out-of-band data buffers and extra metadata is required.

* **PickleBuffer** type for **\_\_reduce\_ex\_\_** returns out-of-band buffers.
* **buffer\_callback** parameter while pickling handles out-of-band data buffers.
* **buffers** parameter while unpickling shows out-of-band data buffers.

2. What is monkey patching in Python?

ANS:

 modifying or updating a piece of code or class or any module at the runtime. In simple words, we can change the behavior or working of a class/ module at the runtime without changing the whole python code.

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

ANS:

In Shallow copy, a copy of the original object is stored and only the reference address is finally copied. In Deep copy, the copy of the original object and the repetitive copies both are stored.

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

ANS:

An identifier can have a maximum length of 79 characters in Python. Python is one of the most popular programming languages. Guido van Rossum created it, and it was released in 1991. The language is often used in web development (server-side), software development, mathematics, and system scripting.

5. What is generator comprehension?

ANS:

A generator comprehension is a single-line specification for defining a generator in Python. It is absolutely essential to learn this syntax in order to write simple and readable code. Note: Generator comprehensions are not the only method for defining generators in Python.