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

**Ans:-**

**Positional-only parameters(/)**

**There is a new function parameter syntax “/” which indicates that some function parameters must be specified positionally and can’t be used as keyword arguments. Assignment Expressions(:=)**

**This operator is used to assign and return a value in the same expression. This removes the need for initializing the variable upfront.**

**f-strings now support “=”**

**This string formatting mechanism is known as Literal String Interpolation or more commonly as F-strings (because of the leading f character preceding the string literal). The idea behind f-strings is to make string interpolation simpler. Python 3.8 allows the use of the above-discussed assignment operator and equal sign (=) inside the f-strings. reversed() works with a dictionary Unlike Python 3.7, now in Python 3.8, the built-in method “reversed()” can be used for accessing the elements in the reverse order of insertion. No parentheses for return and yield statements ‘yield’ and ‘return’ statements do not require parentheses to return multiple values. pow() function In the three-argument form of pow(), when the exponent is -1, it calculates the modular multiplicative inverse of the given value.**

**Syntax Warning If you miss a comma in your code such as a = [(1, 2) (3, 4)], instead of throwing TypeError, it displays an informative Syntax warning.**

1. **What is monkey patching in Python?**

**Ans:-**

**In python, monkey patching refers to run-time modifications of a class or module. In Python, we can actually change the behavior of code at run-time.**

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

**Ans:-**

**In python, when we use = operator, it doesnot create a copy of the object. It just creates the variable, which shares a reference to the same memory location. This is called as shallow copy. In deep copy, new object is a copy of old object and it is created at different memory address and any changes made to new object won't be reflected in old object.**

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

**ANS:-**

**The maximum possible length of an identifier is not defined in Python. It can be any number.**

1. **What is generator comprehension?**

**Ans:-**

**It is like a list comprehension but it returns an iteraor. We use () instead of [] to achieve this.**

**numbers = (i for i in range(5))**

**print(numbers)**

**print(next(numbers))**

**print(next(numbers))**

**print(next(numbers))**

**print(next(numbers))**

**print(next(numbers))**

**<generator object <genexpr> at 0x0000023D6233A970>**

**0**

**1**

**2**

**3**

**4**