1. What are the new features added in Python 3.8 version?
2. What is monkey patching in Python?
3. What is the difference between a shallow copy and deep copy?
4. What is the maximum possible length of an identifier?
5. What is generator comprehension?

Answer:

1. Python 3.8 introduced several new features including:

* The Walrus Operator (:=)
* Positional-only parameters in functions
* f-strings now support = to provide default values
* The typing module has new features like TypedDict, Final, and Literal
* Debug Mode (python -X dev) provides better diagnostics for debugging
* An update to the Python documentation
* Performance improvements, such as faster calls to built-in functions, faster file system operations, and faster runtime startup time.

1. Monkey patching in Python refers to the practice of modifying the behavior of a module, class, or function at runtime by modifying its attributes or methods. It involves changing the code of an existing object, often done by redefining a method or attribute of a class, to add or change functionality.
2. In Python, a shallow copy creates a new object, but it only copies the reference to the original object's memory location, not the object itself. As a result, changes made to the original object are reflected in the shallow copy. A deep copy creates a new object and copies the original object and all its child objects, so changes made to the original object do not affect the deep copy.
3. The maximum length of an identifier in Python is implementation-dependent. In CPython, the maximum length of an identifier is 255 Unicode code points.
4. A generator comprehension is a compact way to create a generator object, which produces a sequence of values on the fly. It has a similar syntax to list comprehensions, but it uses parentheses instead of square brackets. For example, the following code creates a generator object that generates the squares of the first 10 integers:

squares = (x\*\*2 for x in range(1, 11))