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

* Monkey patching in Python refers to the practice of dynamically modifying or extending existing classes or modules at runtime, typically to alter their behaviour or add new functionality. This is done by directly modifying the code of classes or modules, often without modifying their source code.

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

1. Shallow Copy:

- A shallow copy creates a new object, but it only copies references to the elements in the original object.

- If the original object contains nested objects (e.g., lists within a list), a shallow copy creates new references to the nested objects, but it does not recursively duplicate the nested objects themselves.

- Changes made to nested objects in a shallow copy will be reflected in the original object, as they share the same nested objects.

2. Deep Copy:

- A deep copy creates a new object and recursively duplicates all elements within the original object, including any nested objects.

- It creates entirely independent copies of all elements, so changes made to nested objects in a deep copy will not affect the original object.

- Deep copies are often used when you need a fully independent copy of an object and its contents.

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

* In Python, the maximum possible length of an identifier is not explicitly defined. However, it is generally recommended to keep identifiers reasonably short and meaningful for code readability. There is no strict maximum limit, but extremely long identifiers may not be practical and can make the code less readable.

1. **What is generator comprehension?**

* A generator comprehension is a concise way to create a generator in Python using a syntax similar to list comprehensions. It allows you to generate values on-the-fly without creating a full list in memory. Generator comprehensions use parentheses `()` instead of square brackets `[]` like list comprehensions.

Example of a generator comprehension:

```python

generator = (x \* 2 for x in range(10))

```