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

**ANSWER:** Some of the new features added in Python 3.8 are:

* Assignment expressions (also known as the walrus operator): Allows assigning a value to a variable within an expression. This feature makes code shorter and easier to read.
* Positional-only parameters: Allows specifying parameters that can only be passed by position and not by keyword, making it easier to write and use functions.
* f-strings support = to format string literals. They are faster and easier to read than other string formatting methods.
* A new syntax for typing named tuples: Allows defining named tuples with type annotations in a more concise and readable way.
* The 'yield from' expression has been deprecated: It has been replaced with the 'yield' expression, which simplifies the code and reduces confusion.
* New and improved modules: Python 3.8 includes several new and improved modules, including the asyncio module for asynchronous programming, the math.isqrt() function for computing the integer square root, and the statistics.fmean() function for computing the arithmetic mean of floating-point numbers.

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

**ANSWER:** Monkey patching is a technique in Python where you can modify the behavior of a module, class, method, or function at runtime by changing or replacing its attributes or methods. It involves dynamically modifying the code of an object or module in order to change its behavior or add new features.

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

**ANSWER:** A shallow copy creates a new object that is a copy of the original object, but the new object only contains references to the same memory locations as the original object. if you modify one of the mutable objects in the original object or the shallow copy, both will be affected. In Python, you can create a shallow copy using the copy() method or the slice notation.

On the other hand, a deep copy creates a new object that is a copy of the original object, including all nested objects. This means that if the original object contains mutable objects, a deep copy will also create new copies of those mutable objects. In Python, you can create a deep copy using the deepcopy() method from the copy module.

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

**ANSWER:** According to the Python documentation, there is no specified limit on the length of an identifier in Python. However, it is recommended to keep identifier names reasonably short and descriptive to enhance code readability and maintainability.

1. **What is generator comprehension?**

**ANSWER:** A generator comprehension is a concise way to create a generator object in Python. It is similar to list comprehensions, but instead of creating a list, it creates a generator object that produces values on-the-fly as they are needed. Generator comprehensions are useful when you want to create a sequence of values, but don't need to store the entire sequence in memory at once. Because generator comprehensions create generator objects, they are also more memory-efficient than list comprehensions when dealing with large data sets.