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

Assignment Expressions (Walrus Operator): Python 3.8 introduced the "walrus operator" (:=), which allows assignment expressions within expressions. It enables you to assign values to variables as part of a larger expression, providing a more concise and expressive way to write code.

Positional-only Parameters

f-strings = Debugging

The "math" Module Improvements:

The "statistics" Module Improvements:

The "pathlib" Module Improvements:

Performance Improvements:

Other Enhancements:

1. What is monkey patching in Python?

Monkey patching refers to the practice of modifying or extending the behavior of existing code at runtime, typically by adding, replacing, or modifying methods or attributes of classes or objects. It allows you to change the behavior of a class or object without directly modifying its source code.

In Python, monkey patching is possible because of the language's dynamic nature. You can dynamically modify classes, objects, modules, or even built-in functions and methods

1. What is the difference between a shallow copy and deep copy?  
   The difference between a shallow copy and a deep copy lies in how the copying process handles objects that are nested or referenced within the original object

Shallow Copy

Deep Copy

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

In Python, the maximum possible length of an identifier is not explicitly defined. However, according to the Python language reference, an identifier can be any combination of letters, digits, and underscores (\_), with the restriction that it must start with a letter or an underscore. There is no specific limit on the length of an identifier imposed by the Python language itself.

However, it is important to note that although there is no formal maximum length, it is generally recommended to keep identifiers concise and meaningful for code readability and maintainability. Using excessively long identifiers may make the code harder to read and understand. Additionally, some tools and coding conventions may have their own guidelines or limitations on identifier length

1. What is generator comprehension?

Generator comprehension, also known as generator expression, is a concise way to create a generator object in Python. It is similar to list comprehension but returns a generator instead of a list. Generator comprehension allows you to generate values on the fly without creating a complete sequence in memory, which can be more memory-efficient and faster for large datasets.

The syntax for generator comprehension is similar to list comprehension, but instead of using square brackets [], it uses parentheses ()