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

Python 3.8 introduced several new features and improvements. Some of the notable features added in Python 3.8 include:

* Assignment Expressions (Walrus Operator): The introduction of the := operator allows assignment expressions within other expressions, enabling more concise and readable code.
* Positional-Only Parameters: Python 3.8 added support for defining positional-only parameters in function signatures. These parameters can only be passed by position and not by keyword, providing more control over the function's parameter usage.
* f-strings support = for self-documenting expressions: f-strings now support the = specifier, which provides a simple way to include the expression's value along with the expression itself, improving code readability and debugging.
* The math.prod() function: The math module now includes a prod() function that calculates the product of all elements in an iterable. This provides a convenient way to compute products without resorting to manual loops.
* The typing.TypedDict class: Python 3.8 introduced the TypedDict class in the typing module, which allows the definition of dictionaries with specified keys and value types. This helps in providing type hints and enforcing type checks for dictionaries.
* The multiprocessing.shared\_memory module: Python 3.8 added the shared\_memory module to the multiprocessing package, enabling the creation and sharing of shared memory segments between processes.

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 objects or classes. It allows you to modify the behavior of code without directly changing its source code.

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

* Shallow copy: Creates a new object that references the original elements. Changes to shared elements affect both the original and copied objects.
* Deep copy: Creates a new object with completely independent copies of all elements. Changes to the copied object do not affect the original or any other copies.

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

In Python, the maximum possible length of an identifier is not explicitly defined by the language specification. However, the practical limit is typically determined by the system's memory constraints and the maximum length of strings supported by the underlying operating system.

1. What is generator comprehension?

Generator comprehension, also known as generator expression, is a concise way to create generator objects in Python. It is similar to list comprehension but generates values on-the-fly instead of creating a complete list in memory.