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

🡪Python 3.8 introduced several new features and enhancements. Here are some of the key additions:

1. Assignment Expressions (The Walrus Operator): This feature allows assignment expressions within expressions. It is denoted by the ":=" operator, and it assigns a value to a variable while also returning that value as the result of the expression.

2. Positional-Only Parameters: Python 3.8 introduced a syntax to define function parameters that can only be passed positionally and not as keyword arguments. This allows developers to define more restrictive function signatures.

3. The "math.prod()" Function: The math module now includes a "prod()" function that calculates the product of all the elements in an iterable. It is a shorthand for writing a for loop to calculate the product manually.

4. The "f-strings" Debug Mode: Python 3.8 added a new "f-strings" debugging feature that allows you to prefix an f-string with 'f' to evaluate the expression inside the f-string at runtime and display the result. This can be helpful in debugging and understanding the output of f-strings.

5. The "statistics.Mode()" Function: The statistics module now includes a "mode()" function that calculates the mode (most frequently occurring value) in a given dataset.

6. Performance Improvements: Python 3.8 introduced several performance optimizations, including a faster call protocol, optimized built-in types, and more efficient handling of built-in function calls.

7. Reversible Dictionaries: Python 3.8 introduced the "reversed()" function for dictionaries, allowing you to reverse the key-value pairs of a dictionary.

8. Syntax Warnings: Python 3.8 introduced new syntax warnings to provide better feedback when using syntax that will be deprecated in future versions of Python.

1. What is monkey patching in Python?

🡪In Python, the term monkey patch refers to making dynamic (or run-time) modifications to a class or module. In Python, we can actually change the behaviour of code at run-time.

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

🡪When an object is copied using copy(), it is called shallow copy as changes made in copied object will also make corresponding changes in original object, because both the objects will be referencing same address location.

When an object is copied using deepcopy(), it is called deep copy as changes made in copied object will not make corresponding changes in original object, because both the objects will not be referencing same address location.

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

🡪In Python, the highest possible length of an identifier is 79 characters. Python is a high-level programming language.

Python, particularly when combined with identifiers, is case-sensitive.

When writing or using identifiers in Python, it has a maximum of 79 characters.

Unlikely, Python gives the identifiers unlimited length.

However, the layout of PEP-8 prevents the user from breaking the rules and includes a 79-character limit.

1. What is generator comprehension?

🡪 A generator comprehension is a single-line specification for defining a generator in Python.

It is absolutely essential to learn this syntax in order to write simple and readable code.

Generator comprehension uses round bracket unlike square bracket in list comprehension.

The generator yields one item at a time and generates item only when in demand. Whereas, in a list comprehension, Python reserves memory for the whole list. Thus we can say that the generator expressions are memory efficient than the lists.