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

Answer:

Below are some of the new advancements in Python 3.8.

🡪 Walrus operator (assignment expressions) is used to assign and return a value in the same expression. This eliminates the need to initialize variables beforehand. The big advantage of this is that it saves few lines of code.

🡪 We can use positional-only parameters (/) to define function parameters that can only be passed by position, not by name.

🡪 The yield and return statements do not require parentheses to return multiple values.

🡪 The reversed() works with dictionaries. We can access elements in reverse insertion order using the built-in reversed() method.

🡪 Changed dictionary comprehensions to compute the key first and then the value.

🡪 importlib\_metadata is a new library addition to Python's standard utilities module that provides an API for accessing the metadata of installed packages.

🡪 f-string now supports the equality operator (=) to simplify string interpolation. In Python 3.8, we can use the above assignment operators and the equal sign (=) inside f-strings. It is also uses for for self-documenting expressions and debugging.

🡪 The 3-argument form of pow() computes the modulo multiplicative reciprocal of the specified value if the exponent is -1.

🡪 csv.DictReader now returns a dictionary instance instead of collections.OrderedDict.

🡪 Better diagnostics and error messages, including more detailed error messages for syntax mistakes, are now provided by the Python interpreter. If we don't have commas in our code. For example, a = [(10, 12) (30, 45)] will give us an informative syntax warning instead of a TypeError.

🡪 The typings module has been updated to include several new types and features such as: Literal types and TypedDict.

🡪 New modules and enhancements to the standard library include a statistics module and new asyncio module features.

🡪 Faster f-strings, enhanced breakpoint() debugging, and enhanced multiprocessing module performance are additional enhancements.

1. What is monkey patching in Python?

Answer:

Monkey patching in Python refers to changes made to a class or module during runtime. The behaviour of code can actually be altered at run-time in Python. It entails altering or introducing new functionality to a module, class, or object that already exists without actually altering its original source code.

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

Answer:

🡪 shallow copy: when an object is copied using copy(), it's called shallow copy as changes made in copied object will also make corresponding changes in original object, because both the objects will be representing same address position.

🡪 deep copy: when an object is copied using deepcopy(), it's called deep copy as changes made in copied object won't make corresponding changes in original object, because both the objects won't be representing same address position.

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

Answer:

The maximum length of the identifier is not defined (unlimited length), but practical limits are set by free memory and the processing time required to process longer names. However, according to PEP-8 (Style Guide for Python Code) lines should be limited to 79 characters.

1. What is generator comprehension?

Answer:

A generator comprehension is a one-line specification for defining a generator in Python. It's like a list comprehension, but it returns an iterate. To achieve this we use parentheses '()' instead of square bracket '[]'.

Example:

num = (i for i in range(10))

print(num)

print(next(num))

print(next(num))