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

Ans- Python 3.8 was released on October 14, 2019, and it came with several new features and optimizations. Some of the major features introduced in Python 3.8 are:

1. Assignment Expressions: Also known as the "walrus operator" (:=), this new operator allows you to assign a value to a variable as part of an expression. For example, you can now write if (n := len(my\_list)) > 10: instead of n = len(my\_list) followed by if n > 10:.
2. Positional-only Parameters: Functions can now specify parameters that can only be passed positionally, i.e., not as keyword arguments. This allows for more flexible function signatures and can help clarify the intended usage of a function.
3. f-strings support '=' for self-documenting expressions: This feature allows you to use the '=' character in f-strings to indicate an expression's result in the formatted string.
4. Improved Typing Support: Python 3.8 includes several improvements to the typing module, including the addition of TypedDict, Final, and Literal types.
5. Performance improvements: Python 3.8 includes several optimizations that improve the interpreter's performance, including improved hash algorithms, optimized dictionary lookups, and faster variable annotations.
6. Debug Mode: This feature allows you to run Python in debug mode, which enables various debugging features and checks.
7. Pickle Protocol 5: Python 3.8 introduced a new version of the pickle protocol (protocol 5), which improves performance and supports new datatypes.

These are some of the major features introduced in Python 3.8, among others.

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

Ans- Monkey patching in Python refers to the technique of dynamically modifying code at runtime, usually by replacing or adding new functions or methods to an existing module or class. Monkey patching can be useful in situations where you need to modify the behavior of third-party code, or when you want to add new functionality to an existing object without subclassing or modifying its source code.

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

Ans- In Python, a shallow copy and a deep copy are two different ways to create a copy of an object. A shallow copy creates a new object which stores a reference to the original object's memory location. In other words, it creates a new object with the same reference to the internal data as the original object. Changes to the internal data of the original object will be reflected in the shallow copy, and vice versa.

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

Ans- In Python, the maximum length of an identifier is not explicitly defined. However, Python's official style guide, PEP 8, recommends limiting all lines of code to a maximum of 79 characters, including identifiers. This is considered a best practice for improving code readability.

Meanwhile there is no hard limit on the length of an identifier in Python, it's generally best to use descriptive but concise names that are no longer than necessary to convey the intended meaning of the identifier.

1. **What is generator comprehension?**

Ans- Generator comprehension is a compact way of creating a generator object in Python. It is similar to list comprehension, but instead of creating a list, it creates a generator that can be used to lazily generate a sequence of values. The syntax for generator comprehension is similar to that of list comprehension, but with parentheses instead of square brackets.

Example: gen = (x\*\*2 for x in range(5))