5. **Modules and Libraries**:

   - Python has a vast standard library, providing numerous pre-built modules and functions for common tasks.

   - External libraries can be installed using package managers like pip.

6. **File Handling**:

   - Python can read from and write to files easily using built-in functions.

7. **Exception Handling**:

   - Errors can be handled using try, except, else, and finally blocks.

8. **Object-Oriented Programming (OOP)**:

   - Python supports OOP principles, including classes, objects, inheritance, and polymorphism.

9. **List Comprehensions**:

   - A concise way to create lists based on existing lists or other iterable objects.

10. **Lambda Functions**:

    - Anonymous functions defined using the lambda keyword.

11. **Packages and Modules**:

    - Python code can be organized into modules and packages, promoting code reusability.

12. **Virtual Environments**:

    - Allows isolating Python environments to manage project dependencies effectively.

13. **Input and Output**:

    - Input can be taken from the user using "input()" function, and output can be displayed using "print()" function.

14. **Documentation**:

    - Python code is often documented using docstrings to provide useful information for developers.