Python Technical Interview Questions

Python Technical Interview Questions

Python Inbuilt Functions

1. Data Type Conversion Functions:

   - `int()`, `float()`, `str()`: Convert between data types.

   - `list()`, `tuple()`, `set()`, `dict()`: Convert to respective collection types.

2. Mathematical Functions:

   - `abs()`: Returns the absolute value.

   - `min()`, `max()`: Return the minimum or maximum of an iterable or multiple arguments.

   - `sum()`: Sums up elements of an iterable.

   - `round()`: Rounds a number to a given precision.

3. Iterable and Sequence Functions:

   - `len()`: Returns the length of an iterable.

   - `enumerate()`: Returns an enumerate object, adding a counter to an iterable.

   - `zip()`: Aggregates elements from multiple iterables.

   - `sorted()`: Returns a sorted list of the specified iterable.

   - `reversed()`: Returns a reversed iterator.

   - `map()`: Applies a function to every item in an iterable.

   - `filter()`: Filters elements of an iterable based on a function.

4. Object and Type-Related Functions:

   - `type()`: Returns the type of an object.

   - `isinstance()`: Checks if an object is an instance of a specific class or type.

   - `id()`: Returns the unique identifier of an object.

- `dir()`: Tries to return a list of valid attributes of an object.
- `getattr()`, `setattr()`, `hasattr()`: Get, set, or check for an attribute in an object.

5. Input and Output Functions:

- `print()`: Outputs to the console.
- `input()`: Reads a string from standard input.
- `open()`: Opens a file and returns a corresponding file object.

6. Memory Management Functions:

- `del`: Deletes objects.
- `id()`: Returns the unique identifier of an object, which is its memory address.

7. Functional Programming Tools:

- `lambda`: Creates a small anonymous function.
- `map()`: Applies a function to every item of an iterable.
- `filter()`: Constructs an iterator from elements of an iterable for which a function returns true.
- `reduce()`: Applies a function of two arguments cumulatively to the items of an iterable (from the `functools` module).

8. Error Handling Functions:

- `try`, `except`, `finally`, `raise`: Used for exception handling.

9. Help and Documentation:

- `help()`: Invokes the built-in help system.
- `dir()`: Lists all the attributes and methods of an object.

10. Miscellaneous Functions:

- `range()`: Generates a sequence of numbers.
- `all()`, `any()`: Return True if all or any of the elements of an iterable are true.
- `bin()`, `hex()`, `oct()`: Convert an integer to binary, hexadecimal, or octal string.

Python Technical Interview Questions

Sample Python Technical Interview Questions

1. Explain the difference between Python 2 and Python 3.
2. What is the purpose of the `__init__` method in Python?
3. How does Python handle memory management?
4. What are Python decorators, and how do they work?
5. How would you implement a stack in Python?
6. Explain the difference between deep copy and shallow copy.
7. What is a lambda function? Provide an example.
8. How does Python's garbage collection work?
9. Explain the Global Interpreter Lock (GIL).
10. How do you handle exceptions in Python?
11. What are list comprehensions? Give an example.
12. Describe the use of `with` statement in Python.
13. What is the difference between `@staticmethod` and `@classmethod`?
14. How do you convert a string to a number in Python?
15. What are *args and **kwargs in Python functions?
16. How does Python's list slicing work?
17. How do you create a virtual environment in Python?
18. What is the difference between `is` and `==` in Python?
19. How can you optimize Python code for performance?
20. What are Python generators, and how do they work?