Frequently Asked Python Interview Questions

1. What is Python?

Python is a high-level, interpreted programming language known for its simplicity and readability. It supports multiple programming paradigms and is widely used in various domains such as web development, data science, and automation.

2. Explain the difference between Python 2 and Python 3.

Python 2 and Python 3 are two major versions of the Python programming language. Python 3 introduced several syntax changes and improvements over Python 2 to enhance code readability and address certain design flaws.

3. What are the advantages of using Python for Data Science?

Python is widely used in Data Science due to its simplicity, readability, and the availability of powerful libraries and tools specifically designed for data analysis and machine learning. Some advantages include:

- **Libraries:** Python offers efficient data manipulation and visualization with libraries like NumPy, pandas, and matplotlib.
- **Machine Learning**: Libraries like scikit-learn, TensorFlow, and PyTorch provide robust tools for machine learning.
- **Community Support**: Python's large, active community contributes new libraries, shares knowledge, and offers support.
- **Integration**: Python easily integrates with other languages and tools, fitting well into various data science workflows.
- **Flexibility**: Python allows data scientists to handle tasks from data cleaning to model deployment within one environment.
- **Scalability:** Python works with datasets of any size, making it suitable for projects of all scales.
- **Ease of Learning:** Python's simple syntax is beginner-friendly, yet powerful for experienced data scientists.

4. What is PEP 8?

PEP 8 is the Python Enhancement Proposal that provides guidelines for writing clean, readable Python code. It covers topics such as naming conventions, indentation, and code layout to promote consistency across Python projects.

5. List down the differences between List, Tuple, Set and Dictionary in Python.

(on next page)

Parameters	List	Tuple	Set	Dictionary
Definition	A list is an ordered, mutable collection of elements.	A tuple is an ordered, immutable collection of elements.	A set is an unordered collection of unique elements.	A dictionary is an unordered collection of key-value pairs.
Syntax	Syntax includes square brackets [,] with ‘,’ separated data.	Syntax includes curved brackets (,) with ‘,’ separated data.	Syntax includes curly brackets { , } with ‘,’ separated data.	Syntax includes curly brackets { , } with ‘,’ separated key-value data.
Creation	A list can be created using the list() function or simple assignment to [].	Tuple can be created using the tuple() function.	A set dictionary can be created using the set() function.	A dictionary can be created using the dict() function.
Empty Data Structure	An empty list can be created by l = [].	An empty tuple can be created by t = ().	An empty set can be created by s = set().	An empty dictionary can be created by {}.
Order	It is an ordered collection of data.	It is also an ordered collection of data.	It is an unordered collection of data.	Ordered collection in Python version 3.7, unordered in Python Version=3.6.
Duplicate Data	Duplicate data entry is allowed in a List.	Duplicate data entry is allowed in a Tuple.	All elements are unique in a Set.	Keys are unique, but two different keys CAN have the same value.

Indexing	Has integer based indexing that starts from ‘0’.	Also has integer based indexing that starts from ‘0’.	Does NOT have an index based mechanism.	Has a Key based indexing i.e. keys identify the value.
Addition	New items can be added using the append() method.	Being immutable, new data cannot be added to it.	The add() method adds an element to a set.	update() method updates specific key-value pair.
Deletion	Pop() method allows deleting an element.	Being immutable, no data can be popped/deleted.	Elements can be randomly deleted using pop().	pop(key) removes specified key along with its value.
Sorting	sort() method sorts the elements.	Immutable, so sorting method is not applicable.	Unordered, so sorting is not advised.	Keys are sorted by using the sorted() method.
Reversing	reverse() method reverses the list.	Immutable, so reverse method is not applicable.	Unordered, so reverse is not advised.	No integer-based indexing, so no reversal.

6. What is the purpose of the 'pass' statement in Python?

The 'pass' statement in Python is a null operation that does nothing when executed. It is often used as a placeholder in situations where code is required syntactically but no action is needed.

7. How do you create a function in Python?

In Python, a function is defined using the 'def' keyword followed by the function name and parameters. The function body is indented and contains the code to be executed when the function is called.

8. What is the difference between mutable and immutable data types in Python?

Mutable data types in Python can be modified after creation, while immutable data types cannot be changed. Examples of mutable types include lists and dictionaries, while examples of immutable types are tuples and strings.

9. Explain the concept of indentation in Python.

Indentation in Python is used to define the structure of the code. It is crucial for indicating blocks of code within functions, loops, and conditional statements. Consistent indentation is a key aspect of Python syntax.

10. What is the purpose of the 'import' statement in Python?

The 'import' statement in Python is used to bring external modules or packages into the current script, allowing access to their functions and variables. It enables code reuse and modularity in Python programs.

11. How do you handle exceptions in Python?

Exceptions in Python are managed using 'try', 'except', and 'finally' blocks. Code that may raise an exception is placed within the 'try' block, and specific error handling logic is defined in the 'except' block.

12. What is the difference between 'append' and 'extend' methods in Python lists?

The 'append' method in Python lists adds a single element to the end of the list, while the 'extend' method appends multiple elements (such as another list) to the existing list. 'append' modifies the list in place, while 'extend' combines lists.

13. How do you create a dictionary in Python?

A dictionary in Python is created using curly braces {} and key-value pairs separated by colons. Keys must be unique and immutable, while values can be of any data type. Dictionaries provide efficient data retrieval based on keys.

14. What is the purpose of the 'range' function in Python?

The 'range' function in Python generates a sequence of numbers within a specified range. It is commonly used in loops to iterate over a sequence of numbers or to create lists of numbers based on a start, stop, and step value.

15. How do you create a for loop in Python?

A for loop in Python iterates over a sequence of elements such as a list, tuple, or string. It is defined using the 'for' keyword followed by a variable that represents each element in the sequence. The loop body is indented and executed for each iteration.

16. What is the difference between 'break' and 'continue' statements in Python loops?

The 'break' statement in Python terminates the current loop and exits to the next statement outside the loop, while the 'continue' statement skips the remaining code in the current iteration and proceeds to the next iteration of the loop.

17. How do you create a while loop in Python?

A while loop in Python executes a block of code repeatedly as long as a specified condition is true. It is defined using the 'while' keyword followed by the condition to be evaluated. The loop body is indented and continues to execute until the condition becomes false.

18. What is the purpose of the 'return' statement in functions?

The 'return' statement in Python functions is used to exit the function and return a value to the caller. It can also be used to return multiple values as a tuple. Functions without a return statement implicitly return 'None'.

19. How do you create a string in Python?

Strings in Python are created by enclosing text within single ('') or double ("") quotes. They can also be created using triple quotes (''' ''' or """ """) for multi-line strings. Strings are immutable sequences of characters in Python.

20. What is the purpose of the 'len' function in Python?

The 'len' function in Python is used to determine the length of a sequence such as a string, list, tuple, or dictionary. It returns the number of elements in the sequence, allowing for dynamic sizing and indexing operations.

21. Explain the difference between '==' and 'is' in Python.

The `==` operator compares the values of two objects, while the `is` operator compares the identity (memory location) of two objects. Two objects with the same value may not necessarily be the same object in memory.

22. What is a lambda function in Python?

A lambda function in Python is an anonymous function that can take any number of arguments but can only have one expression. It is defined using the `lambda` keyword followed by the arguments and a colon `:`.

23. What is the purpose of the 'with' statement in Python?

The `with` statement in Python is used to provide a convenient syntax for working with objects that need setup and teardown, such as file handling or locking. It ensures that the object is properly initialized and cleaned up, even if an exception occurs.

24. Explain the concept of list comprehension in Python.

List comprehension in Python provides a concise way to create lists. It allows you to generate a new list based on an existing iterable (such as a list, tuple, or string) using a single line of code. List comprehension can include conditional statements and loops.

25. What is the difference between 'sorted' and 'sort' methods in Python?

The `sorted` function in Python returns a new sorted list from an iterable, while the `sort` method sorts the elements of a list in-place. The `sorted` function can be used with any iterable, while `sort` is only applicable to lists.

26. How do you create a set in Python?

Sets in Python are created using curly braces `{}` or the `set()` function. Sets are unordered collections of unique elements. Duplicate elements are automatically removed when a set is created.

27. What is the purpose of the 'map' function in Python?

The `map` function in Python applies a given function to each item of an iterable (such as a list, tuple, or string) and returns a map object. It is commonly used to perform an operation on each element of an iterable.

28. What is the difference between 'join' and 'split' methods in Python strings?

The `join` method in Python concatenates the elements of an iterable (such as a list, tuple, or set) into a single string, using the string on which the method is called as the separator. The `split` method splits a string into a list of substrings, using a specified separator (whitespace by default).

29. How do you create a class in Python?
In Python, a class is defined using the `class` keyword followed by the class name. The class body contains the attributes (variables) and methods (functions) that define the behavior of the class. Classes can also inherit from other classes using the `()` syntax after the class name.

30. How do you create a module in Python?

In Python, a module is a file containing Python definitions and statements. To create a module, you simply need to write Python code in a file with a `.py` extension. Modules can be imported into other Python scripts using the `import` statement, allowing for code reuse and organization.

31. What is the purpose of the 'try-except' block in Python?

The `try-except` block in Python is used for exception handling. The `try` block contains the code that might raise an exception, while the `except` block specifies the type of exception to catch and the corresponding error handling logic. This allows for graceful error handling and prevents the program from crashing.

32. Explain the concept of duck typing in Python.

Duck typing in Python is a style of dynamic typing in which an object's methods and properties determine the valid semantics, rather than its inheritance from a particular class. If an object walks like a duck and quacks like a duck, it is considered a duck, regardless of its actual type.

33. What is the difference between 'append' and 'insert' methods in Python lists?

The `append` method in Python lists adds an element to the end of the list, while the `insert` method inserts an element at a specified index. `append` modifies the list in-place, while `insert` shifts the existing elements to make room for the new element.

34. Explain the Global Interpreter Lock (GIL) in Python.

The Global Interpreter Lock (GIL) in Python is a mutex that protects access to Python objects, preventing multiple native threads from executing Python bytecodes simultaneously. This means that only one thread can execute Python bytecode at a time, limiting the parallelism of multi-threaded Python programs.

35. What are metaclasses in Python?

Metaclasses in Python are the "classes of classes." They define the behavior of classes, allowing customization of class creation, modification, and instantiation. Metaclasses are rarely used but provide a powerful way to control class behavior and enforce specific patterns in class definitions.

36. Describe the difference between 'staticmethod' and 'classmethod' in Python.

In Python, the `staticmethod` decorator defines a method that does not receive the instance or class as the first argument. It behaves like a regular function but is associated with the class for organizational purposes. On the other hand, the `classmethod` decorator defines a method that takes the class as the first argument, allowing access to class-level attributes and methods.

37. What is the purpose of the 'init' method in Python classes?

The `__init__` method in Python classes is a special method used for initializing new objects. It is called when a new instance of the class is created, allowing the class to initialize its attributes and set up the object's initial state. This method is commonly used to assign initial values to object properties.

38. Explain the concept of generators in Python.

Generators in Python are functions that enable the creation of iterators. They use the `yield` keyword to return data one at a time, allowing for efficient memory usage and lazy evaluation. Generators are useful for generating large sequences of data without storing them in memory all at once.

39. What is the difference between 'args' and 'kwargs' in Python functions?

In Python functions, `*args` and `**kwargs` allow functions to accept a variable number of positional and keyword arguments, respectively. `*args` collects additional positional arguments into a tuple, while `**kwargs` collects additional keyword arguments into a dictionary. This flexibility enables functions to handle varying numbers of arguments.

40. Explain the concept of abstract base classes in Python.

Abstract Base Classes (ABCs) in Python are classes that define abstract methods that must be implemented by subclasses. They provide a way to define a common interface for a group of related classes, ensuring that subclasses adhere to a specific structure and behavior.

41. What is the difference between 'set' and 'frozenset' in Python?

In Python, a `set` is a mutable collection of unique elements, while a `frozenset` is an immutable collection of unique elements. Sets are unordered and can be modified, while frozensets are hashable and cannot be changed after creation. Frozensets are useful for creating hashable collections.

42. What is the difference between 'map' and 'filter' functions in Python?

In Python, the `map` function applies a given function to each item in an iterable and returns an iterator of the results. On the other hand, the `filter` function applies a given function to each item in an iterable and returns an iterator containing only the items for which the function returns `True`.