PYTHON INTERVIEW QUESTION

Q1. What is init keyword?

A1. In Python, the __init__ method is a special method that is automatically called when an object of a class is created. It is also known as the constructor method. The name __init__ stands for "initialize," and its purpose is to initialize the attributes of an object.

The __init__ method is defined within a class and takes at least one argument, self, which refers to the instance of the object being created. Additional arguments can be included to initialize other attributes of the object.

Q2. What is self-keyword?

A2. In Python, self is a conventional name for the first parameter of a method in a class. It refers to the instance of the object on which the method is being called. The purpose of using self is to access the instance's attributes and methods within the class.

Q3. What is lambda function?

A3. Lambda function is a small anonymous function that is defined using the **`lambda`** keyword. It is also known as an anonymous function because it doesn't have a name like a regular function defined using the **`def`** keyword.

Q4. Difference between lambda and normal function?

A4. Lambda functions are anonymous, meaning they don't have a specific name assigned to them. They are often used when a small function is needed for a short period and doesn't require a formal function definition. Normal functions have a name and can be referenced by that name throughout the program.

Q5. What are generators?

A5. Generators are a type of iterable, which can be thought of as functions that produce a sequence of values over time. They are defined using a special kind of function called a

generator function, which uses the **`yield`** keyword instead of **`return`** to produce a series of values

Q6. Is Python compiled or interpreted language? What does it mean?

A6. Python source code is written in plain text files with the **`.py`** extension. When you execute a Python program, the source code is first compiled into an intermediate form called bytecode. This compilation process is performed by the Python interpreter, which is a program that reads and executes Python code.

- Compilation: The Python interpreter compiles the source code into bytecode. This step includes syntax checking and generating the intermediate representation.
- Interpretation: The bytecode is executed by the Python Virtual Machine, which translates it into machine instructions on the fly. The PVM handles memory management, variable bindings, and other runtime tasks.

Q7. What is the difference between list and tuples in Python?

A7. In Python, lists and tuples are both used to store collections of items. Lists are mutable, which means you can modify their elements after creation. You can add, remove, or change items to a list. Tuples, on the other hand, are immutable, meaning their elements cannot be modified once they are assigned. Once a tuple is created, you cannot add, remove, or modify its elements.

Q8. What is the difference between list and set in Python?

A8. In Python, lists and sets are both used to store collections of items.

Lists allow duplicate elements, meaning you can have multiple occurrences of the same value in a list. Sets, on the other hand, do not allow duplicate elements. If you try to add a duplicate value to a set, it will be ignored.

Q9. When to use dictionary?

A9. Dictionaries are used when you need to store and retrieve data in key-value pairs. They are highly useful in various scenarios where you want to associate values with unique keys.

Q10. What are decorators?

A10. Decorators are a powerful feature that allows you to modify the behavior of functions or classes without directly changing their source code. Decorators provide a way to wrap or decorate a function or class with additional functionality.

In essence, a decorator is a function that takes another function as input, extends or modifies its behavior, and returns a new function or callable object. This allows you to add functionality to functions or classes transparently, without altering their original implementation.

Q11. What are Iterators?

A11. An iterator is an object that implements the iterator protocol, which consists of the **`__iter__()`** and **`__next__()`** methods. Iterators are used to traverse through elements of a collection or sequence, allowing you to access items one by one without the need to load the entire collection into memory.

Q12. What is slicing?

A12. Slicing refers to the technique of extracting a portion (subsequence) of a sequence object such as a list, tuple, or string. It allows you to retrieve a contiguous section of elements from the original sequence based on specified start, stop, and step parameters.

Q13. What is mutable and immutable?

A13. Mutable and immutable are terms used to describe the ability of an object to be modified after it is created.

- Immutable objects are those whose state (values) cannot be changed after they are created.
- Mutable objects are those that can be modified after they are created.