Find and read 20 interview questions for Functions.

FUNCTIONS QUESTIONS:

1.What is the use of the "self" keyword in Python?

Ans: In Python, "self" refers to the instance of a class that a method calls. It is typically used within a method to refer to instance variables or contact other instance methods. When a method calls on an instance of a class, the self keyword accesses the instance's attributes and methods.

2. What is a lambda function in Python?

In Python, a lambda function is a small, anonymous function that can have any number of arguments but can only have one expression. Lambda functions are a shorthand for creating simple functions that are only needed once. They are made using the lambda keyword, followed by the function's arguments and a colon, and then the expression for evaluation.

3.How do you create a function in Python?

def keyword is used to create a function in Python, followed by the name of the function and any parameters the function will accept in parentheses. The code inside the function should be indented to show that it is a part of it.

4.What is the difference between "map" and "filter" functions in Python?

"map" and "filter" are built-in functions operating on iterable objects. The main difference between the two functions is that "map" applies a given function to each item in an iterable and returns an iterator with the results. In contrast, "filter" applies a given function to each item in an iterable and returns an iterator with only the things that meet the given condition.

5. What is the 'main' function in Python? How do you invoke it?

In the world of programming languages, the 'main' function is considered as an entry point for the execution for a program. But in Python, this is known that the interpreter serially interprets the file line-by-line. This means that Python does not provide the 'main()' function explicitly. But this doesn't mean that it a cannot simulate the execution of 'main'. It can do this by defining the user-defined 'main()' function and using the python file's '\_\_name\_\_' property. This '\_\_name\_\_' variable is a particular built-in variable that points to the current module's name.

6.What is a Function in Python programming?

A function is an object which represents a block of code and is a reusable entity. It brings modularity to a program and a higher degree of code reusability. Python has given us many built in functions such as print() and provides the ability to create user-defined functions.

7.What is the Return keyword used for in Python?

The purpose of a function is to receive the inputs and return some output. The return is a Python statement which can use in a function for sending a value back to its caller.

8.Is it mandatory for a Python function to Return a value?

It is not necessary for a function to return any value. However, if needed, we can use None as a return value.

9.What is \*args and \*\*kwargs?

\*args is used when the programmer is not sure about how many arguments are going to be passed to a function, or if the programmer is expecting a list or a tuple as argument to the function.\*\*kwargs is used when a dictionary(keyword arguments) is expected as an argument to the function.

10.What are decorators?

In Python, decorators serve as essential functions that enable the addition of functionality to an already existing function without alerting its structure.These decorators are denoted by the @decorator\_name syntax in Python and are invoked in a bottom-up manner.

11. What are the two types of functions in Python?

There are two types of functions in Python: built-in functions and user-defined functions. Built-in functions are functions that are already defined in the Python language, such as the print() function. User-defined functions are functions that are created by the user, and they can be created to do anything that the user wants them to do.

12. Do Python functions have return values? If yes, then how many can they have?

Yes, Python functions can have return values. They can have a single return value, or they can have multiple return values.

13. Why does Python support both positional and keyword arguments to its functions?

Python supports both positional and keyword arguments in order to give developers more flexibility when designing their functions. Positional arguments are those that are passed in by position, without explicitly specifying the parameter name. Keyword arguments are those that are passed in by explicitly specifying the parameter name. Python allows for both types of arguments so that developers can choose the approach that makes the most sense for their particular function.

14. Is there any way to define static methods in Python? If yes, then how?

Static methods are defined in Python by using the @staticmethod decorator. This decorator can be applied to any method, and will cause the method to be treated as a static method, even if it is not defined as such.

15. Is it possible to pass a variable number of arguments to a function in Python? If yes, then how?

Yes, it is possible to pass a variable number of arguments to a function in Python. This can be done using the \*args and \*\*kwargs parameters. \*args allows for a variable number of non-keyworded arguments to be passed to a function, while \*\*kwargs allows for a variable number of keyworded arguments to be passed.

16. Can you explain the difference between named parameters and keyword parameters in Python?

In Python, named parameters are those that are assigned a name in the function definition, while keyword parameters are those that are assigned a value when the function is called. For example, in the following function definition, the parameter x is a named parameter, while the parameter y is a keyword parameter:

17. What’s the difference between positional parameters and default parameters in Python?

Positional parameters are parameters that are required to be provided in order for the function to run. Default parameters are parameters that are not required to be provided, but have a default value that will be used if no other value is provided.

18. What does the Python filter function do, and how is it used?

Python filter() function filters elements from an iterable (e.g., list, tuple) based on a specified condition (predicate). It returns an iterator containing only elements that satisfy the condition defined by the given function.

Syntax:

filter(function, iterable)

19. How do you access the value returned by a function invoked in another function?

You can access the value returned by a function invoked in another function by using the return statement. For example, if you have a function that calculates the sum of two numbers, you can access the value returned by the function by using the return statement.

20. What is a pass in Python?

Pass means performing no operation or in other words, it is a placeholder in the compound statement, where there should be a blank left and nothing has to be written there.