1. Why are functions advantageous to have in your programs?

Answer:-

Functions offer several advantages that make them essential in programming:

* Reusability: Functions allow you to encapsulate a specific piece of code and give it a name. You can then call the function whenever you need that code's functionality, promoting code reusability. It reduces redundancy and makes the code more modular and easier to maintain.

* Modularity: Functions break down complex tasks into smaller, manageable parts. This promotes the idea of dividing the program into logical modules, making it easier to understand, test, and debug.

* Abstraction: Functions hide the implementation details from the rest of the program. When you call a function, you don't need to know how it works internally; you only need to know what it does and what inputs it requires. This simplifies the programming process and makes the code more readable.

* Readability: Using functions can improve the readability of your code. Instead of having long blocks of code in a single location, you can name the blocks of code (functions) based on their purpose. This makes it easier for other developers (or even yourself in the future) to understand the code's intentions.

* Maintainability: By breaking code into smaller functions, you make it easier to update or modify specific parts of the program without affecting other areas. This separation of concerns improves code maintainability.

* Code organization: Functions help you organize your code into logical components, which makes it easier to navigate and manage. This organization becomes especially crucial as projects grow in size and complexity.

2. When does the code in a function run: when it's specified or when it's called?

Answer:-

The code in a function executes when the function is called, not when the function is defined.

3. What statement creates a function?

Answer:-

The def statement defines, i.e. creates a function.

4. What is the difference between a function and a function call?

answer:-

A function consists of the def statement and the code in its def clause.

A function call is what moves the program execution into the function, and the function call evaluates to the function's return value.

5. How many global scopes are there in a Python program? How many local scopes?

Answer:-

There is one global scope, and a local scope is created whenever a function is called.

6. What happens to variables in a local scope when the function call returns?

Answer -

When a function returns, the local scope is destroyed, and all the variables in it are forgotten.

8. If a function does not have a return statement, what is the return value of a call to that function?

Answer -

If there is no return statement for a function, its return value is None.

9. How do you make a function variable refer to the global variable?

Answer -

A global statement will force a variable in a function to refer to the global variable.

10. What is the data type of None?

Answer -

The data type of None is NoneType.

11.What does the sentence import areallyourpetsnamederic do?

Answer -

the import keyword is used to include external modules or libraries in your code, allowing you to use their functionality. When you use the import statement, Python looks for a module with the specified name and tries to load it. If the module is found, Python executes the code within that module, and you gain access to its functions, classes, and variables.

However, in the sentence "import areallyourpetsnamederic," it seems like the module name is unusual and not a standard Python module. Therefore, unless you have created a custom module named "areallyourpetsnamederic.py," attempting to run this import statement would raise a ModuleNotFoundError.

12. If you had a bacon() feature in a spam module, what would you call it after importing spam?

Answer -

This function can be called with spam.bacon().

13. What can you do to save a programme from crashing if it encounters an error?

Answer-

To prevent a program from crashing when encountering an error, you can use error handling techniques to gracefully handle exceptions.

Python provides a built-in mechanism for this called "try-except" blocks. By using try-except blocks, you can catch exceptions that occur during the execution of your code and take appropriate actions to handle those exceptions.

14. What is the purpose of the try clause? What is the purpose of the except clause?

Answer-

The purpose of the try clause in Python is to enclose a block of code that might raise exceptions.

It allows you to identify potential error-prone sections of code and attempt to execute them while anticipating and handling any exceptions that may arise.

The purpose of the except clause in Python is to handle exceptions raised within the corresponding try block.

When an exception occurs in the try block, Python checks the except clauses one by one to find a matching exception type. If it finds a match, it executes the code inside that except block to handle the exception.