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

Ans.

Functions are very useful for programs as they serve very important roles in various aspects of programming :

1.Modularity : Functions help in making the codes break down into small bits and parts (into modules) to enhance the management of the codes written. Modularity helps to make the code more understandable and easy to run,test and debug.

2.Readability: Functions makes the code more readable and easier to understand. We can give more understandable names to the functions so that in future the code can be used without any confusion and with proper clarity.

3.Testing , Debugging – Functions make the codes easy to test and debug if any error occurs. Individual functions are easy to test and correct any errors occurring.

4.Reusability : Functions makes the code more reusable in various parts of the code without writing the same code each and every time in different places. Reusability of functions prevents redundancy and repetition.

5.Namespacing: This feature enhances the naming convention of the variables and prevents the clash of the variable names even if they are repeated in same code. They prevent the conflict between variables without disturbing their scope even if the name are same for two or more functions.

6.Encapsulation: Functions help in encapsulating the various workings of the functions which helps in accomplishment of the task without revealing the details of the structure of the codes written. Prevents the abstract workings and the backend operations of the function coming to forefront and performing the desired deed as intended. They help to encapsulate the complex code flow making the code more readable and organised and making the structure of the codes more organised.

7.Parameterisation: Functions accept parameters as defined by the creator which enhances the code customisation as per their own choice and making them more flexible to be used without disturbing the source code.

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

Ans. A function runs when a function is called or invoked and not when the function is specified. A function is defined once and called many times for the reusability purpose. A function is defined and its code is specified at the time of it’s creation and the function runs only when the function is called and invoked and then only the code inside the function gets executed and the code inside the function runs and performs the task.

3. What statement creates a function?

Ans. The def keyword is used to mark the beginning of the creation of the function.

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

Ans. A function is merely a piece of code which starts with a def keyword and has parameters created if applicable and a body in which the desired execution blocks of code is written which coes to comes to work if executed and a return statement which returns a value after execution of the body. They are parts and blocks of code to serve a desired task.

A function call is the way of calling or invoking a function when desired to perform the task intended. When a function is called or invoked the code statement in the the body of the function gets executed and performs the task desired.

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

Ans. There is typically one global scope as the global variable is defined beforehand creation of a function and infinite local scopes which depend upon the number of function calls made. Variables defined within a function are local to the function and scopes of local variables are created as many times the function is called.

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

Ans. Variables in local scope are called for as many times as he function is being called or invoked.

7. What is the concept of a return value? Is it possible to have a return value in an expression?

Ans. A return value is defined after the body of the function is defined and it returns a output value after the body of a function gets executed successfully. A return value produces results after a function code gets executed.

Yes it is possible to define return with an expression and the value it produces can also be used in another expression .

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

Ans. If a function does not have a return statement then the python automatically returns a None output by default.

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

Ans. If we want to declare a global variable as a function variable then we need to use the keyword global and then write the name of the global variable after that.

10. What is the data type of None?

Ans. The data type of None is NoneType . It represents the absence of a value or a null value.

11. What does the sentence import areallyourpetsnamederic do?

Ans. the sentence import areallyourpetsnamederic calls the module named areallyourpetsnamederic from the .py file and executes the codes maybe functions etc. written inside the file and if the .py file is absent or ot found then it returns an ImportError output.

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

Ans.

Import spam

spam.bacon()

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

Ans. We always use try-except block of codes where the main program code is written inside try blocks and except block is used to handle any type of exceptions, errors which would arise due to any type of situations which henceforth will prevent any type of crashing of the program.

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

Ans. the main program code is written inside try blocks and except block is used to handle any type of exceptions, errors which would arise due to any type of situations which henceforth will prevent any type of crashing of the program.