

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

Answer: There are many advantages of functions in programs

1. It help to follow the concept of DNR (do not repeat) your code
2. Reduce the size of file, because by creating one function we can re use as many time we want
3. Clean coding

Function provides efficiency. It is useful in a condition when we have to use the same operation multiple times, for example sum is function, if we don't have this function, then to sum numbers we have to do it manually every time.

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

Answer : Code in function only executed when we call the function

3. What statement creates a function?

Answer : "def function_name(parameter):—" is used to create a function

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

Answer : Function is a code that is written in a way to follow a particular instruction and return some values. While function call is a code where we pass the value or parameter.

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

Answer : Global scope defined as the variable that we have defined outside the function and that can be used anywhere inside or outside the function.

While Local scope refers to the variable defined inside the function these variables do not have any scope outside the function. Means if we created any variable inside the function we cannot use it outside that particular function. On the other hand, global variable can be used anywhere even in other function for example:

```
x = 10 # this is global variable

def sum():

    a = 10 # this is local variable

    b = 20 # local variable

    return a + b + x
```

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

Answer : Local variables scope is only limited to the function itself. If local variable is used in any operation inside function then it will perform that exact operation for example:

```
x = 10 # this is global variable

def sum():

    a = 10 # this is local variable

    b = 20 # local variable

    return a + b + x
```

Here a and b is a local variable and x is a global variable so if we call the function **sum()** then it will return the 40

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

Answer : Return is used inside a function used to return the final value after doing all the necessary code operations. So for example we want to define a function to add two numbers, so for that we will create a function to pass two parameters and then return a value as per the requirement.

Yes we can use return with expressions as well (if else)

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

Answer: If we create a function without return statement then there will be a blank or empty output of that function when we try to call it

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

Answer : We can use a global keyword before assigning the value to that variable inside function

10. What is the data type of None?

Answer: `NoneType`

11. What does the sentence `import a really your pets named eric` do?

Answer: This will import the library `a really your pets named eric` in the program file. We can use their function and method after importing

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

Answer: `spam.bacon()`

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

Answer : We can do the error handling by using try and except clause

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

Answer : We use try to handle errors like **ZeroDivisionError** and many more, sometimes due to these errors our programs get crashed so to avoid this we use try clause.

Except is used to get the value if we are having any such type of error in our program

For example:

```
try:
```

```
    1/0
```

```
except:
```

```
    print('cant perform this')
```

Output: cant perform this