

Title : Python Basic Assignment_3

Name : Bhavikkumar Modi

E-mail : bmodi700@gmail.com

Date : 15.01.2022

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

- It saves code memory by reducing repetitive code.

Also, it marks program modular (easy to understand and use)

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

- When it's called

3. What statement creates a function?

- `def xyz()`

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

- function is a definition. Under function (i.e. `def abc(a,b)`) we need to put program which need to be executed.

- function call is use when need to use particular function (i.e. `abc(a=2,b=3)`). Used for invoking function.

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

- global scope : accessible to entire code

- local scope : accessible only inside function

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

- local scope variable get erased

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

- it returns values from function to variable from which function is called. Yes, it is possible to have return value in an expression

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

- None

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

- using `global` keyword

10. What is the data type of None?

- **NoneType**

11. What does the sentence `import re` do?

- **it imports library/package `re` after `import` keyword**

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

- **`spam.bacon()`**

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

- **exception handling**

```
try
except
```

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

- **it let code execute even if some runtime error occurs in intermediate line**