Submitted to: Sir Muhammad Ayub

Submitted By: Syed Muhammad Ali

ID: 14258

Dept & Semester: BS(SE) – 5th

Subject: Data Science

Q1. a. Why Functions are used discuss in detail?

ANSWER

Function

A function is a set of statements that take inputs, do some specific computation and produces output. Python provides built-in functions like print(), etc. but we can also create your own functions. These functions are called user-defined functions.

Why are Function used

A user use functions in programming to bundle a set of instructions that he want to use repeatedly or that, because of their complexity, are better self-contained in a sub-program and called when needed. That means that a function is a piece of code written to carry out a specified task.

To carry out that specific task, the function might or might not need multiple inputs. When the task is carried out, the function can or cannot return one or more values.

Functions are of Three types:

- **Built-in Function**

  Any function that is provided as part of a high-level language and can be executed by a simple reference with specification of arguments.

  For Example: help() to ask for help, min() to get minimum value, print() to print.
- **User Defined Function**

  In Python, a user-defined function's declaration begins with the keyword def and followed by the function name. The function may take arguments(s) as input within the opening and closing parentheses, just after the function name followed by a colon

  For Example: def add_numbers(x,y):
- **Anonymous Function**

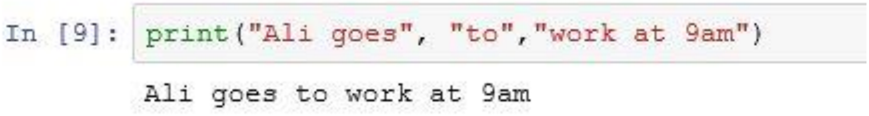
  An anonymous function is a function that is defined without a name. While normal functions are defined using the def keyword in Python, anonymous functions are defined using the lambda keyword. Hence, anonymous functions are also called lambda functions.

  A lambda function is a small anonymous function. A lambda function can take any number of arguments, but can only have one expression

  For Example: x = lambda a : a + 10

b. How arguments are used in function , write a simple program in Python?

ANSWER



```
In [9]: print("Ali goes", "to","work at 9am")

        Ali goes to work at 9am
```

In this we have defined a function printing three arguments

Q2. a. Why .upper(),.lower(),capitalize() and .swapcase() function are used ?

ANSWER

Upper

The Python upper() method is used to convert all letters in a string to uppercase. The Python isupper() returns true if all of the characters in a string are uppercase, and false if they aren't

Syntax: string.upper()

Lower

In Python, lower() is a built-in method used for string handling. The lower() methods returns the lowercased string from the given string. It converts all uppercase characters to lowercase. If no uppercase characters exist, it returns the original string

Syntax: string.lower()

Capitalize

In Python, the capitalize() method converts the first character of a string to capital (uppercase) letter. If the string has its first character as capital, then it returns the original string

Syntax: string_name

Swapcase

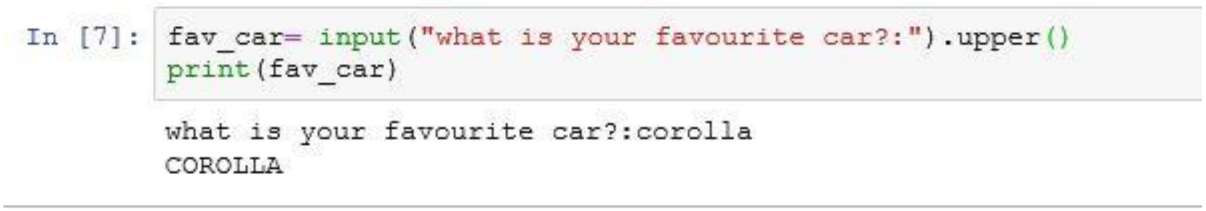
The swapcase() method returns a string where all the upper case letters are lower case and vice versa.

Syntax: string.swapcase()

b. Write a program in which the discussed functions are used.

Note : Q2 part a functions.

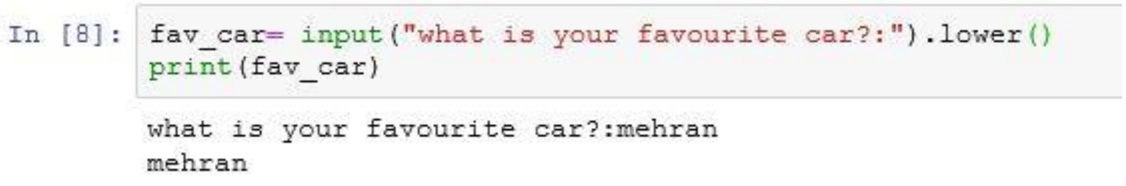
ANSWER



```
In [7]: fav_car= input("what is your favourite car?:").upper()
        print(fav_car)

        what is your favourite car?:corolla
        COROLLA
```

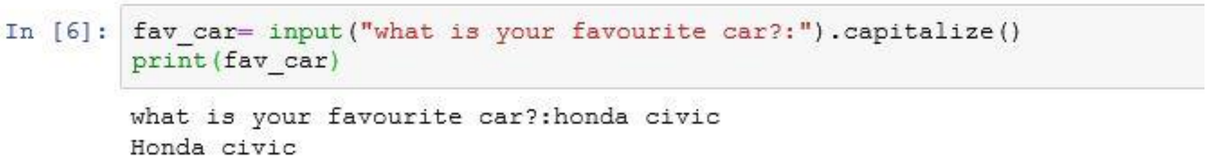
In this we learn how to use the function of upper case by using .upper()



```
In [8]: fav_car= input("what is your favourite car?:").lower()
        print(fav_car)

        what is your favourite car?:mehran
        mehran
```

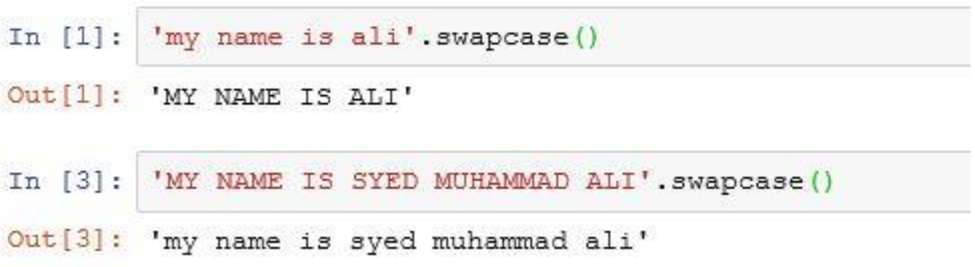
By doing this we learn how to use lower case in a function by using .lower()



```
In [6]: fav_car= input("what is your favourite car?:").capitalize()
        print(fav_car)

        what is your favourite car?:honda civic
        Honda civic
```

By putting this function we now know how to use capitalize case by using .capitalize()



```
In [1]: 'my name is ali'.swapcase()

Out[1]: 'MY NAME IS ALI'

In [3]: 'MY NAME IS SYED MUHAMMAD ALI'.swapcase()

Out[3]: 'my name is syed muhammad ali'
```

In this function we know how to use swapcase in a function by using .swapcase()

Q3. a. What are the rules for defining the function?

ANSWER

Function

A function is a block of organized, reusable code that is used to perform a single, related action. Functions provide better modularity for your application and a high degree of code reusing.

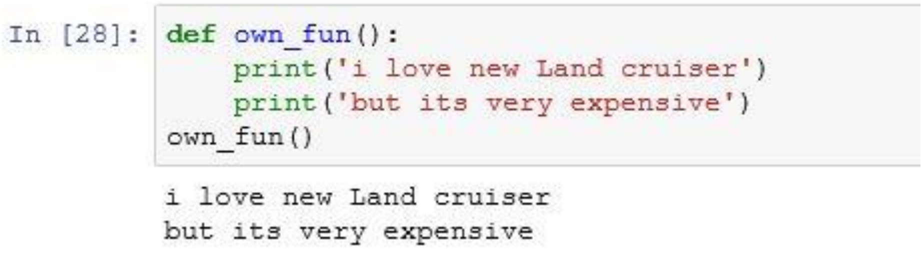
Python gives you many built-in functions like print(), etc. but you can also create your own functions. These functions are called user-defined functions.

Rules of Function

- Input values can be any English word. Output values are letters from the English alphabet
- Input values can be any rational number. Output values can be any rational number.
- Input values can be any whole number. Output values can be any whole number.
- Input values can be any whole number between 1 and 365. Output values can be any month of the year.

b. Write a suitable program of our defined function in Python?

ANSWER



```
In [28]: def own_fun():
             print('i love new Land cruiser')
             print('but its very expensive')
         own_fun()

         i love new Land cruiser
         but its very expensive
```

In this program we learn how to use defined function of our own by the def . Through this we can use any of our own function easily.

Q4. a. What are the rules for defining the function and Parameter passing to the function?

ANSWER

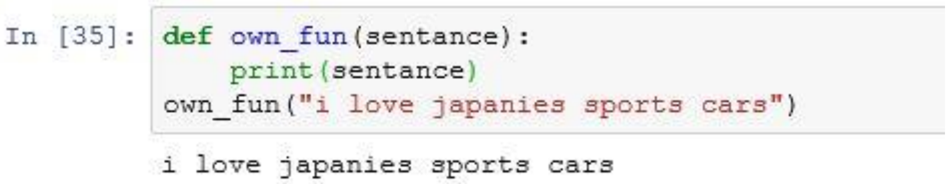
Arguments are passed by value; that is, when a function is called, the parameter receives a copy of the argument's value, not its address. This rule applies to all scalar values, structures, and unions passed as arguments. Modifying a parameter does not modify the corresponding argument passed by the function call.

Rules

- Function blocks begin with the keyword def followed by the function name and parentheses (( )).
- Any input parameters or arguments should be placed within these parentheses. You can also define parameters inside these parentheses.
- The first statement of a function can be an optional statement - the documentation string of the function or docstring.
- The code block within every function starts with a colon (:) and is indented.
- The statement return [expression] exits a function, optionally passing back an expression to the caller. A return statement with no arguments is the same as return None.

b. Write a suitable program of our defined function by parameter passing in Python?

ANSWER



```
def own_fun(sentance):
    print(sentance)
own_fun("i love japanies sports cars")
```

```
i love japanies sports cars
```

The parameter passing function is defined in this program. Putting our own any kind of function name and adding the parameter.

Q5. a. What are return values to a Function discuss in detail?

ANSWER

Return values to a Function

A return is a value that a function returns to the calling script or function when it completes its task. A return value can be any one of the four variable types: handle, integer, object, or string. The type of value your function returns depends largely on the task it performs.

We use the Function Returns edit combo box in the General page of the New Script dialog to tell JAWS the type of value the function returns. We also type the description of the return in the Return description edit box. Adding a description for the return, helps us and anyone else use your function to determine exactly what the value should be used for within the calling script or function.

When we create a new function that returns a string value, the Script Manager places the following function beginning line into our script file:

```
String Function MyFunction ()
```

The "string" key word that precedes the "function" key word tells us that the MyFunction function returns a string value to the calling script or user-defined function.

Python return statement

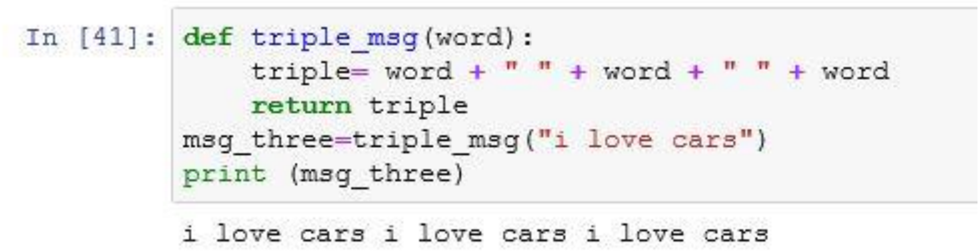
A return statement is used to end the execution of the function call and “returns” the result (value of the expression following the return keyword) to the caller. The statements after the return statements are not executed. If the return statement is without any expression, then the special value None is returned.

Syntax:

```
def fun():
statements
.
.
return[expression]
```

b. Write a suitable program of a Function with returning value?

ANSWER



```
In [41]: def triple_msg(word):
             triple= word + " " + word + " " + word
             return triple
         msg_three=triple_msg("i love cars")
         print (msg_three)
```

```
i love cars i love cars i love cars
```

In this program we studied the returning value. In this case we used three word message and creating a retuning value also. Through this the three times message has been passed as we can see in the above program.