Functions in Python

BY ROHIT S. AGRAWAL

What is a function?

- In Python, a function is a group of related statements that performs a specific task.
- Functions help break our program into smaller and modular chunks.
- As the program grows larger and larger, functions make it more organized and manageable.
- It avoids repetition and makes the code reusable.
- def function_name(parameters):
- """docstring"""
- statement(s)

How to call a function

- Once we have defined a function, we can call it from another function, program, or even the Python prompt.
- To call a function we simply type the function name with appropriate parameters.
- In python, the function definition should always be present before the function call.
- Try to create a function Greetings and call it.

Docstrings

- The first string after the function header is called the docstring and is short for documentation string.
- It is briefly used to explain what a function does.
- Although optional, documentation is a good programming practice.
- Create a function with the docstring.
- We generally use triple quotes so that docstring can extend up to multiple lines.
- This string is available to us as the '__doc__' attribute of the function.
- print(greeting.__doc__)

What is pass statement?

In Python programming, the ‘pass’ statement is a null statement.

The difference between a comment and a pass statement in Python is that while the interpreter ignores a comment entirely, pass is not ignored.

Suppose we have a loop or a function that is not implemented yet, but we want to implement it in the future.

They cannot have an empty body. The interpreter would give an error.

So, we use the ‘pass’ statement to construct a body that does nothing.

The return statement

- The return statement is used to exit a function and go back to the place from where it was called.
- This statement can contain an expression that gets evaluated and the value is returned.
- If there is no expression in the statement or the return statement itself is not present inside a function, then the function will return the ‘None’ object.
- Create a function that prints a some statement.
- Call the function
- Call the function in the print statement

Scope and Lifetime of variables

- Scope of a variable is the portion of a program where the variable is recognized.
- Parameters and variables defined inside a function are not visible from outside the function. Hence, they have a local scope.
- The lifetime of a variable is the period throughout which the variable exists in the memory.
- The lifetime of variables inside a function is as long as the function executes.
- They are destroyed once we return from the function.
- A function does not remember the value of a variable from its previous calls.
- Create a function and try to change the value of variable inside function.

Function Arguments

- In Python, you can define a function that takes variable number of arguments.
- Create a function with 2 arguments and call it.
- Try to call the function with only 1 argument.
- Up until now, functions had a fixed number of arguments.
- In Python, there are other ways to define a function that can take variable number of arguments.

Default Arguments

- Function arguments can have default values in Python.
- We can provide a default value to an argument by using the assignment operator (=).

Keyword Arguments

- When we call a function with some values, these values get assigned to the arguments according to their position.
- Python allows functions to be called using keyword arguments.
- When we call functions in this way, the order (position) of the arguments can be changed.

Arbitrary Arguments

- Sometimes, we do not know in advance the number of arguments that will be passed into a function.
- Python allows us to handle this kind of situation through function calls with an arbitrary number of arguments.
- In the function definition, we use an asterisk (*) before the parameter name to denote this kind of argument.