**Python Functions**

In this article, you'll learn about functions, what a function is, the syntax, components, and types of functions. Also, you'll learn to create a function in Python.

What is a function in Python?

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 our program grows larger and larger, functions make it more organized and manageable.

Furthermore, it avoids repetition and makes the code reusable.

Syntax of Function

def function\_name(parameters):

"""docstring"""

statement(s)

def greet(name):

"""

This function greets to

the person passed in as

a parameter

"""

print("Hello, " + name + ". Good morning!")

Above shown is a function definition that consists of the following components.

1. Keyword def that marks the start of the function header.

2. A function name to uniquely identify the function. Function naming follows the same rules of writing identifiers in Python.

3. Parameters (arguments) through which we pass values to a function. They are optional.

4. A colon (:) to mark the end of the function header.

5. Optional documentation string (docstring) to describe what the function does.

6. One or more valid python statements that make up the function body. Statements must have the same indentation level (usually 4 spaces).

7. An optional return statement to return a value from the function.

Example of a function

def greet(name):

"""

This function greets to

the person passed in as

a parameter

"""

print("Hello, " + name + ". Good morning!")

greet('Paul')

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. Unless you can remember what you had for dinner last week, always document your code.

In the above example, we have a docstring immediately below the function header. 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.

For example:

Try running the following into the Python shell to see the output.

>>> print(greet.\_\_doc\_\_)

This function greets to

the person passed in as

a parameter

The return statement

The return statement is used to exit a function and go back to the place from where it was called.

Syntax of return

return [expression\_list]

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.

For example:

>>> print(greet("May"))

Hello, May. Good morning!

None

Python Program to Make a Simple Calculator

In this example you will learn to create a simple calculator that can add, subtract, multiply or divide depending upon the input from the user.

## Example: Simple Calculator by Using Functions

# Program make a simple calculator

# This function adds two numbers

def add(x, y):

return x + y

# This function subtracts two numbers

def subtract(x, y):

return x - y

# This function multiplies two numbers

def multiply(x, y):

return x \* y

# This function divides two numbers

def divide(x, y):

return x / y

print("Select operation.")

print("1.Add")

print("2.Subtract")

print("3.Multiply")

print("4.Divide")

while True:

# take input from the user

choice = input("Enter choice(1/2/3/4): ")

# check if choice is one of the four options

if choice in ('1', '2', '3', '4'):

num1 = float(input("Enter first number: "))

num2 = float(input("Enter second number: "))

if choice == '1':

print(num1, "+", num2, "=", add(num1, num2))

elif choice == '2':

print(num1, "-", num2, "=", subtract(num1, num2))

elif choice == '3':

print(num1, "\*", num2, "=", multiply(num1, num2))

elif choice == '4':

print(num1, "/", num2, "=", divide(num1, num2))

# check if user wants another calculation

# break the while loop if answer is no

next\_calculation = input("Let's do next calculation? (yes/no): ")

if next\_calculation == "no":

break

else:

print("Invalid Input")

Output

Select operation.

1.Add

2.Subtract

3.Multiply

4.Divide

Enter choice(1/2/3/4): 3

Enter first number: 15

Enter second number: 14

15.0 \* 14.0 = 210.0

Let's do next calculation? (yes/no): no