Python Basics IV

Saif Ali

Department of Electrical Engineering

Jamia Millia Islamia

CONDITIONAL STATEMENTS, ITERATIVE STATEMENTS, FUNCTIONS

https://shorturl.at/rELP5

Conditional Statements

Change the flow or behaviour of a program based on some **condition**.

Go to the market.

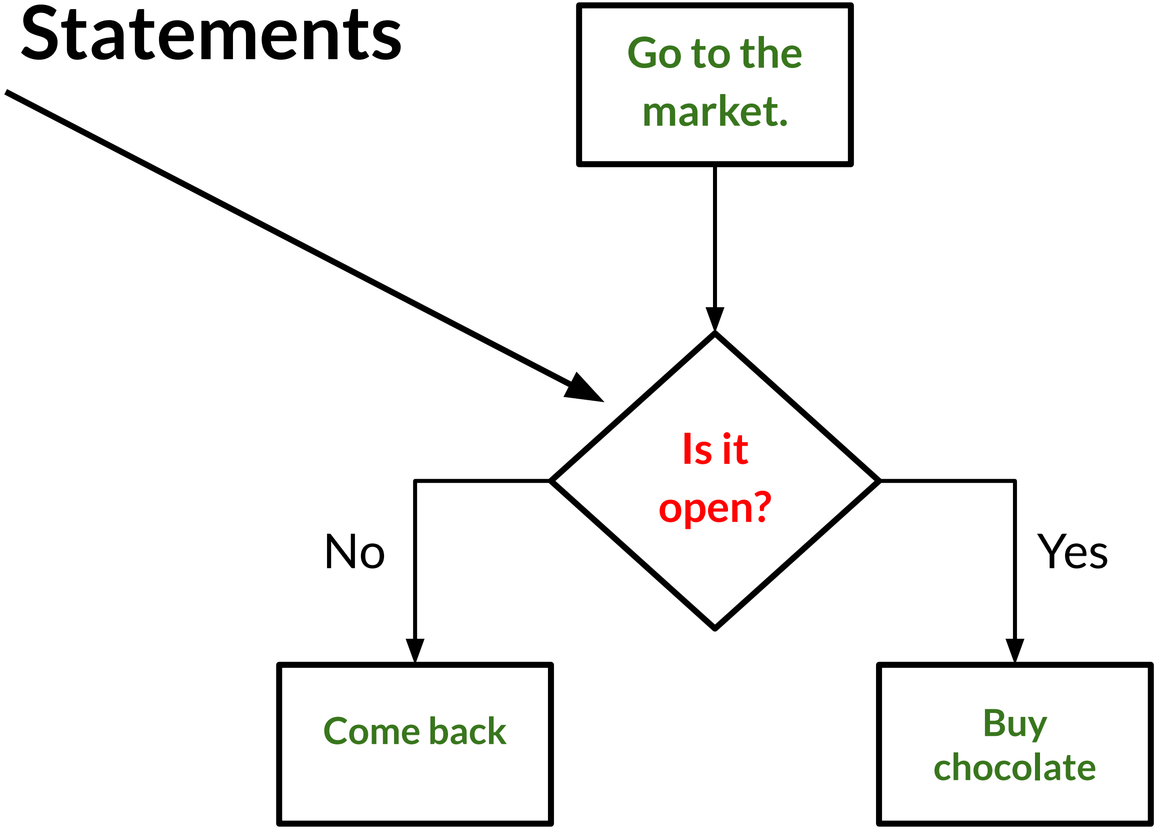
if it is open, buy chocolate, else come back

if, else - keywords

it is open - condition

Go to the market, buy chocolate, come back - actions

Conditional Statements



Go to the market.

Is it open?

No

Yes

Come back

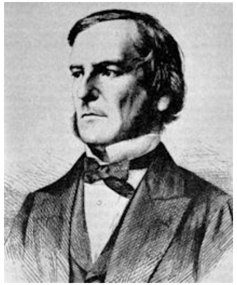
Buy chocolate

Boolean values

A **Boolean value** is either true or false.

In Python, the two Boolean values are **True** and **False** (the capitalization must be exactly as shown), and the Python type is **bool**.

```
flag = True
type(flag)

bool
```



It is named after the British mathematician, George Boole, who first formulated Boolean algebra — some rules for reasoning about and combining these values. This is the basis of all modern computer logic.

Boolean Expressions

A **Boolean expression** is an expression that evaluates to produce a result which is a Boolean value.

For example, the operator == tests if two values are equal. It produces (or yields) a Boolean value:

```
5 == (3 + 2)   # Is five equal 5 to the result of 3 + 2?
True
```

Logical Operators

Six operators that test various kinds of conditions that can exist between two variables.

```
x == y               # Produce True if ... x is equal to y
x != y               # ... x is not equal to y
x > y                # ... x is greater than y
x < y                # ... x is less than y
x >= y               # ... x is greater than or equal to y
x <= y               # ... x is less than or equal to y
```

Conditional Statements

In order to write useful programs, we almost always need the ability to check conditions and change the behavior of the program accordingly.

Conditional statements give us this ability. The simplest form is the **if statement**:

```
marketIsOpen = True
if marketIsOpen == True:
    print("Buy chocolate! :)")
else:
    print("Come back :(")
```

Quick Quiz

What is a “statement” in Python?

Iterative Statements or Loops

Do something **repeatedly** a **certain number of times**.

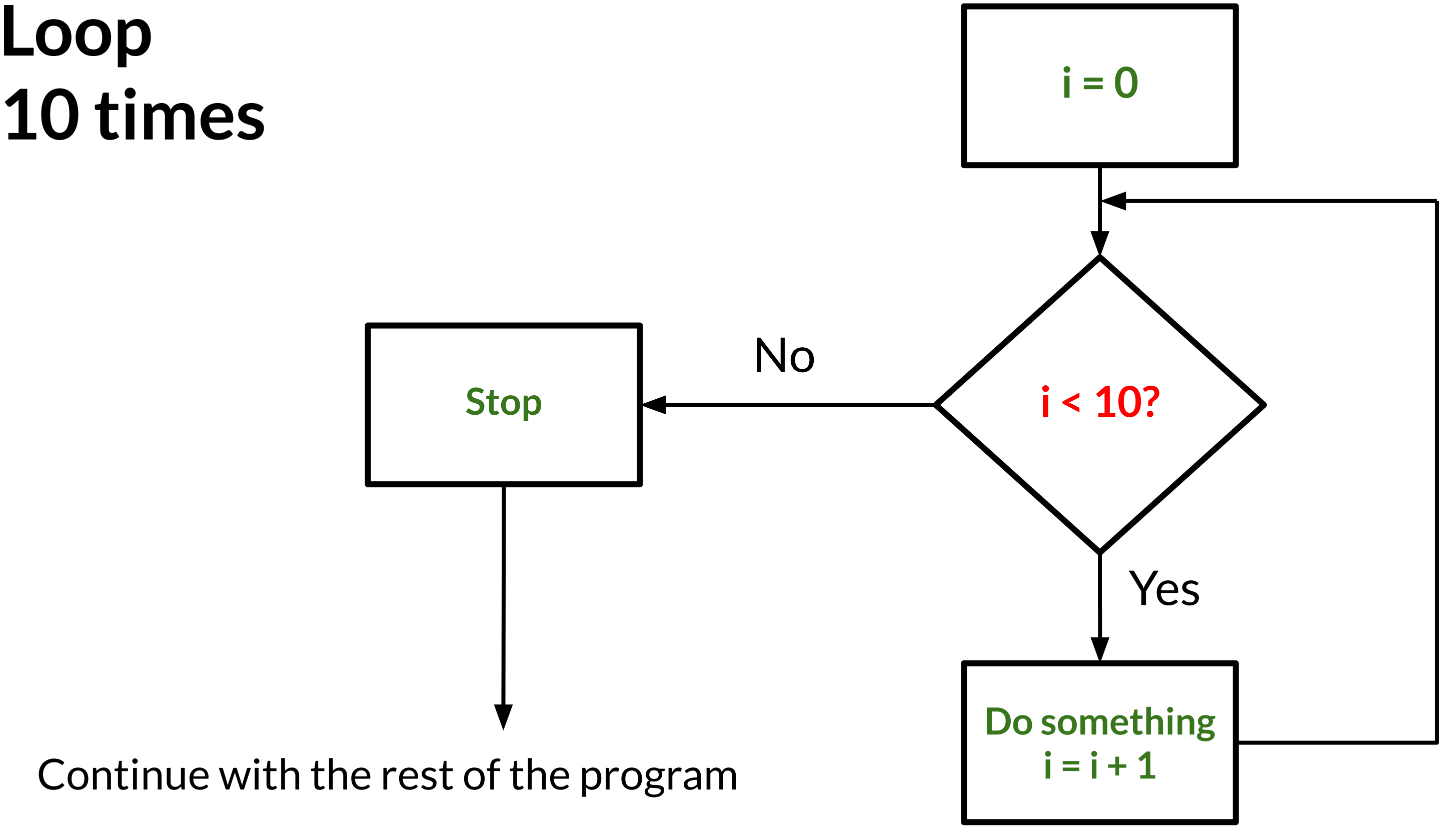
Do something **repeatedly while some condition is met**.

Do something **repeatedly until some condition is met**.

The Infinite Shampooing Loop

1. Lather
2. Rinse
3. Repeat

Loop
10 times



i = 0

i < 10?

No

Stop

Yes

Do something
i = i + 1

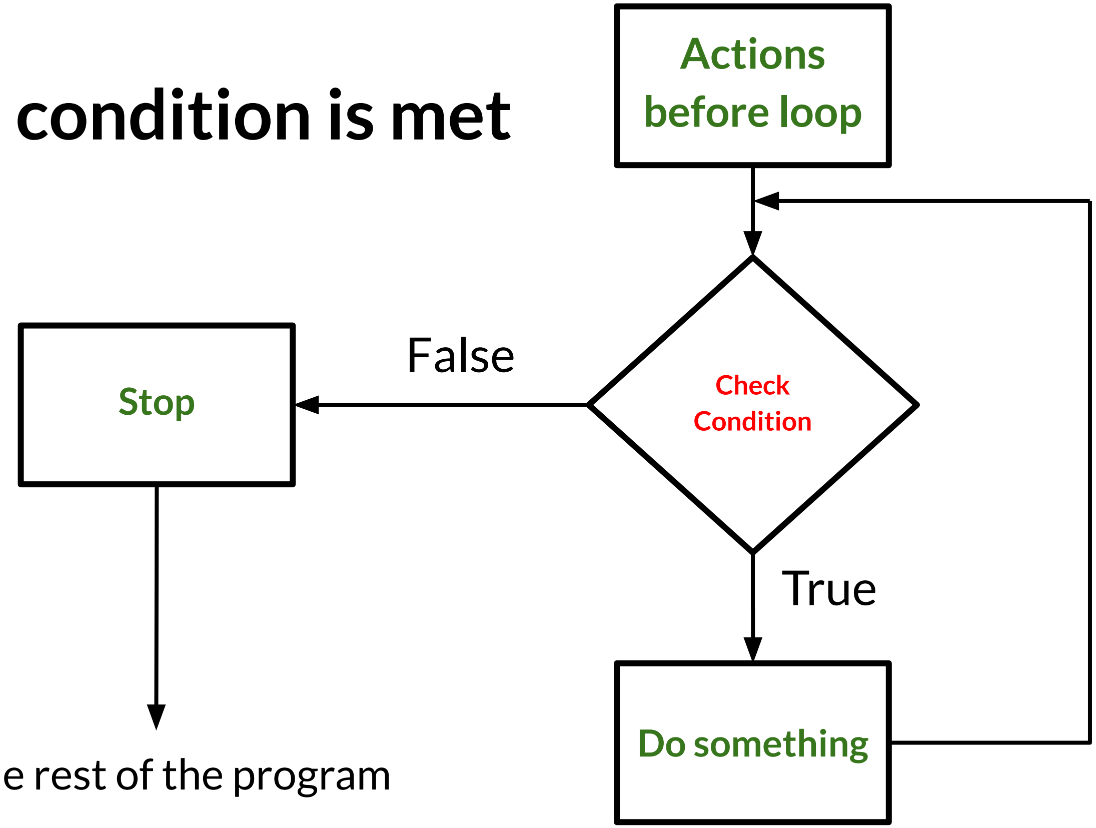
Continue with the rest of the program

The for loop

```
for <this many times>
    do something
```

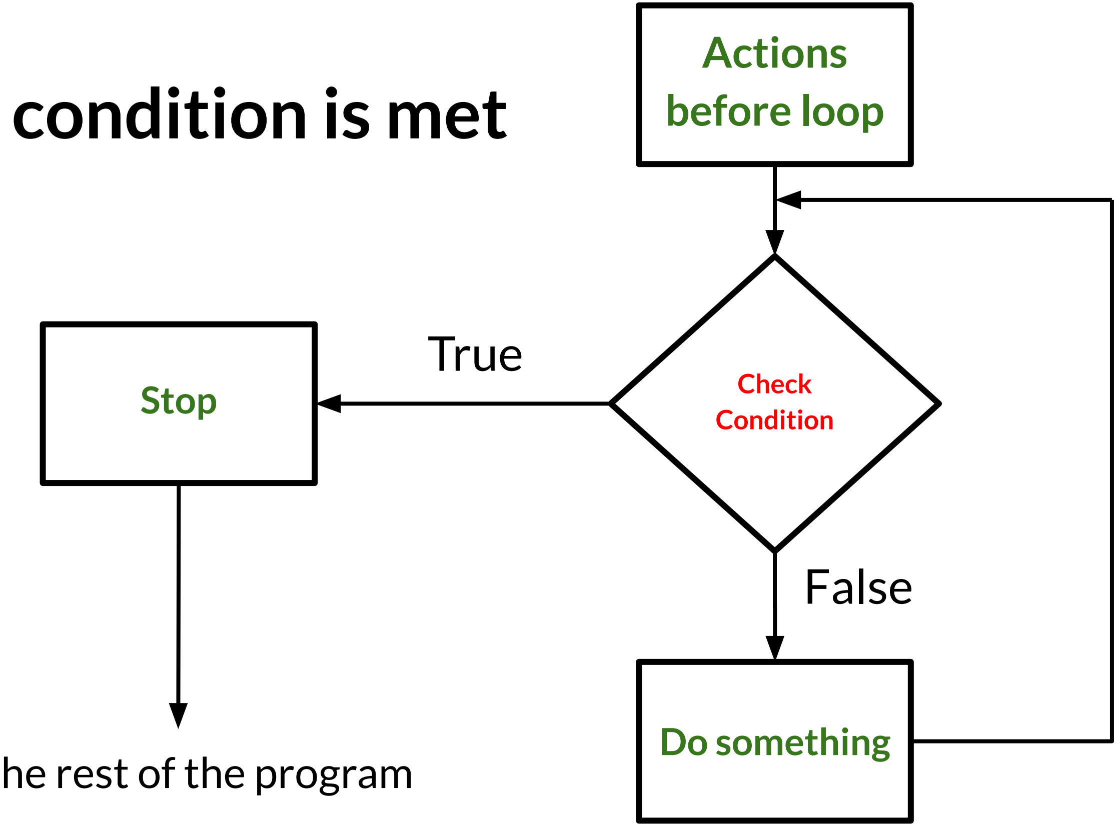
```python
for i in range(1,11):
    print(i)
```

Loop
while some condition is met



Continue with the rest of the program

Loop
until some condition is met



Actions before loop

Check Condition

True

Stop

False

Do something

Continue with the rest of the program

The while loop

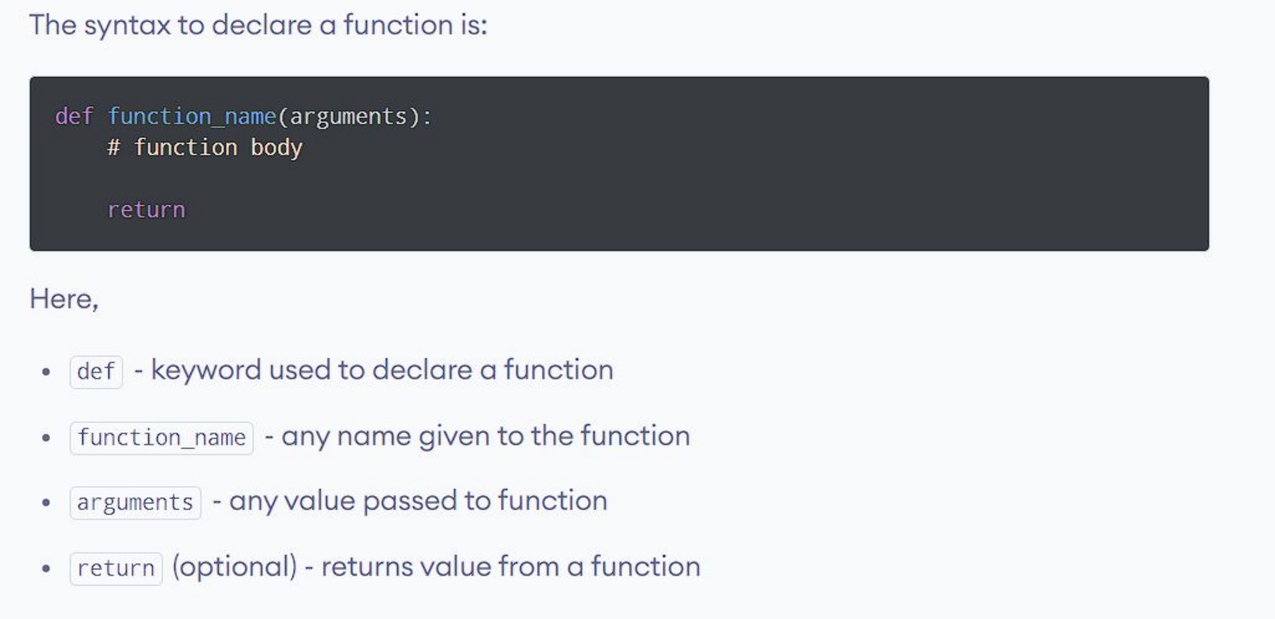
```
while <condition>
    do something
```

```
count = 0
while count < 5:
    print("Do something!")
    count += 1
```

Functions

In Python, a function is a **named sequence of statements** that belong together. Their primary purpose is to help us organize programs into chunks that match how we think about the problem.

Function Declaration



The syntax to declare a function is:

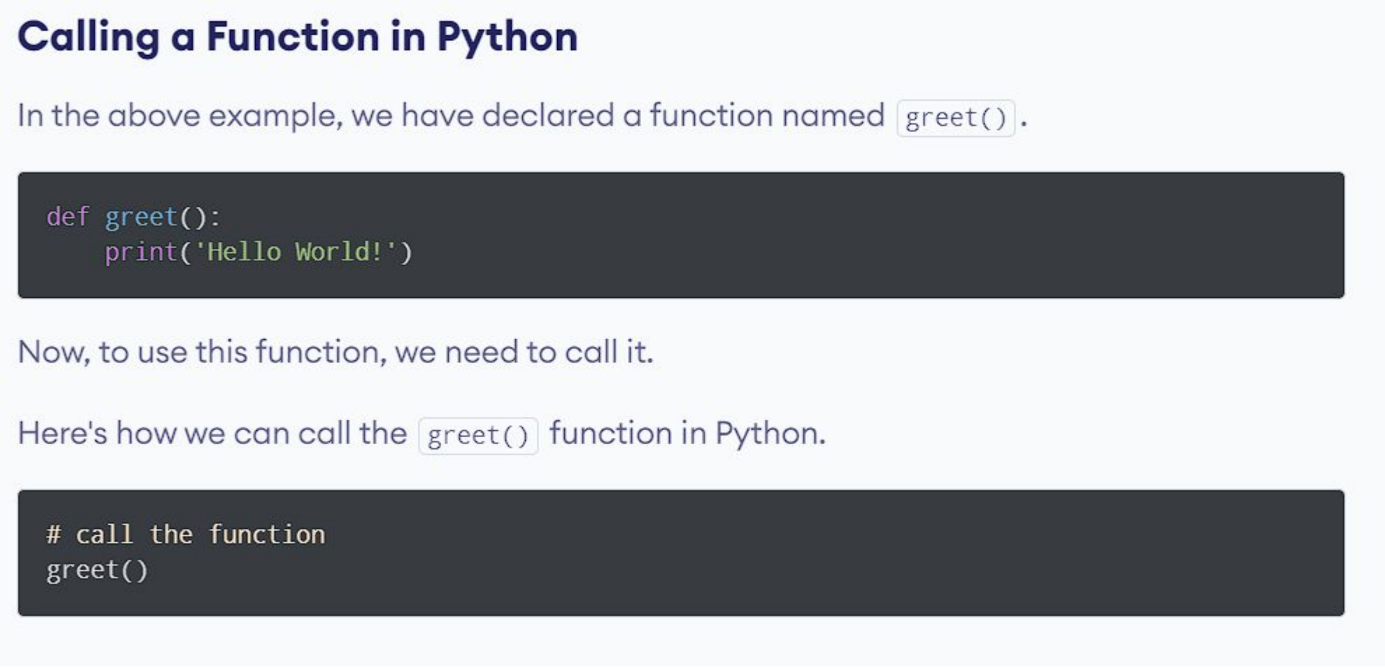
```
def function_name(arguments):
    # function body

    return
```

Here,

- `def` - keyword used to declare a function
- `function_name` - any name given to the function
- `arguments` - any value passed to function
- `return` (optional) - returns value from a function

Source:
https://www.programiz.com/python-programming/function

Calling a function



Calling a Function in Python

In the above example, we have declared a function named `greet()`.

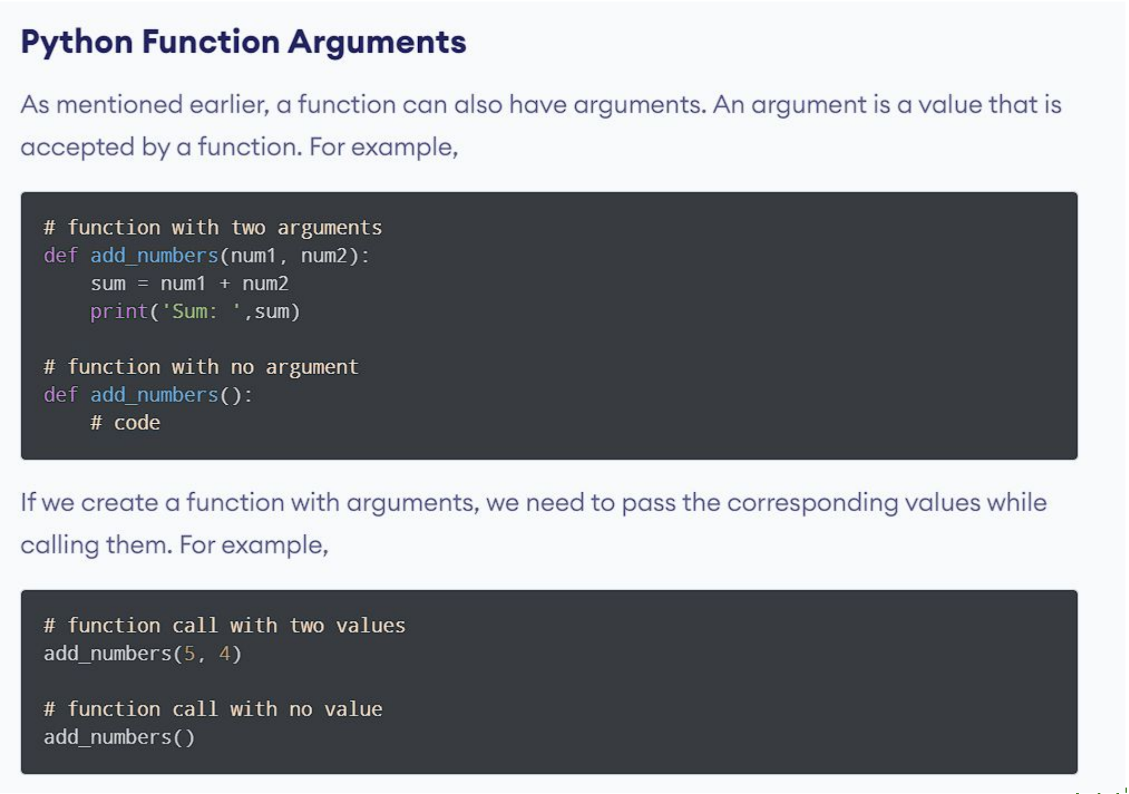
```
def greet():
    print('Hello World!')
```

Now, to use this function, we need to call it.

Here's how we can call the `greet()` function in Python.

```
# call the function
greet()
```

Function Arguments



Python Function Arguments

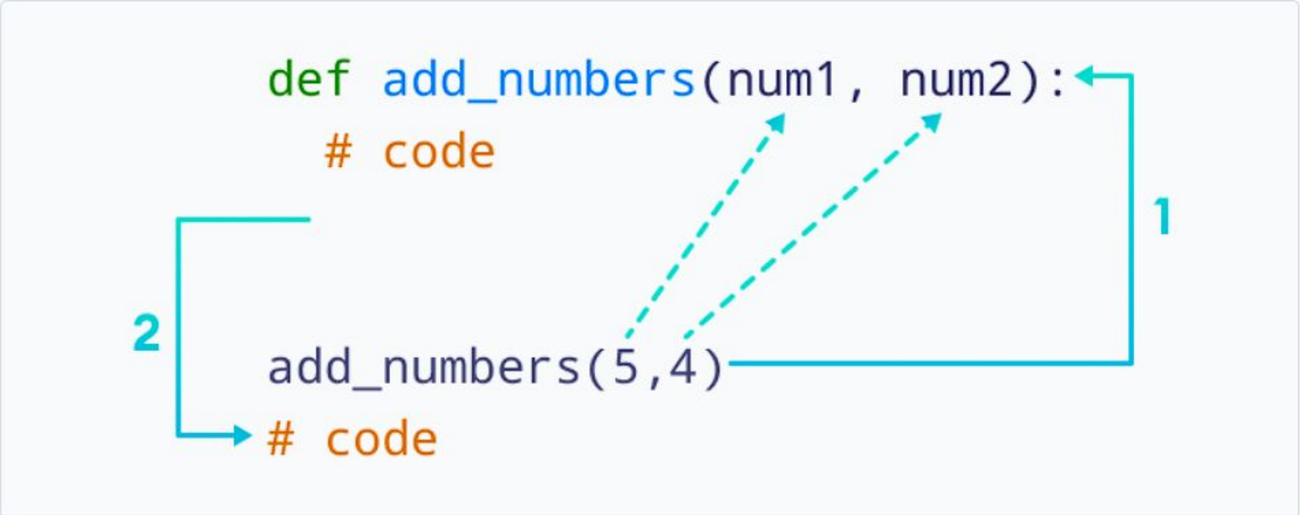
As mentioned earlier, a function can also have arguments. An argument is a value that is accepted by a function. For example,

```
# function with two arguments
def add_numbers(num1, num2):
    sum = num1 + num2
    print('Sum: ',sum)

# function with no argument
def add_numbers():
    # code
```

If we create a function with arguments, we need to pass the corresponding values while calling them. For example,

```
# function call with two values
add_numbers(5, 4)

# function call with no value
add_numbers()
```

Function Flow of Execution



```
def add_numbers(num1, num2):
  # code


add_numbers(5,4)
# code
```

Python Function with Arguments