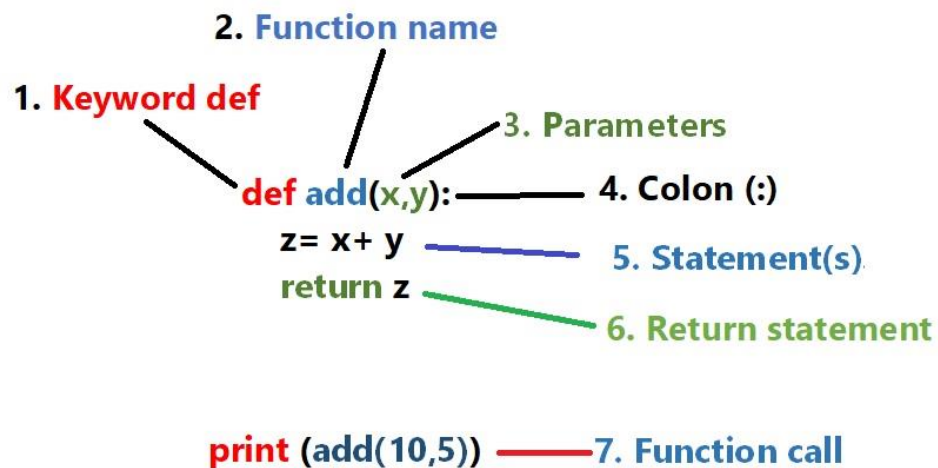


Components of Python Functions

A block of statements that defines a function in Python consists of the following parts:



1. Keyword def

Function definition starts with the keyword 'def' that defines the function. It tells the beginning of the function header.

2. Function name

The function_name represents the name of a function. Every function must be a unique user-defined name or an identifier and given to it in the function header statement.

It is an excellent practice to name your function according to the work it performs. In the above example, the function name is msg, and add.

3. Parameters

The formal parameters (arguments) placed inside the parentheses are optional. A parameter is a piece of data or information that we use inside the function to perform a specific task.

Generally, parameters are a list of local variables that will be assigned with values when the function gets called. They are used to passing values to functions.

They can be zero, or more parameters, separated by commas inside the parentheses. In the above example, there are two parameters in the function definition.

4. Colon (:)

A colon indicates the end of function headers.

5. Statement(s):

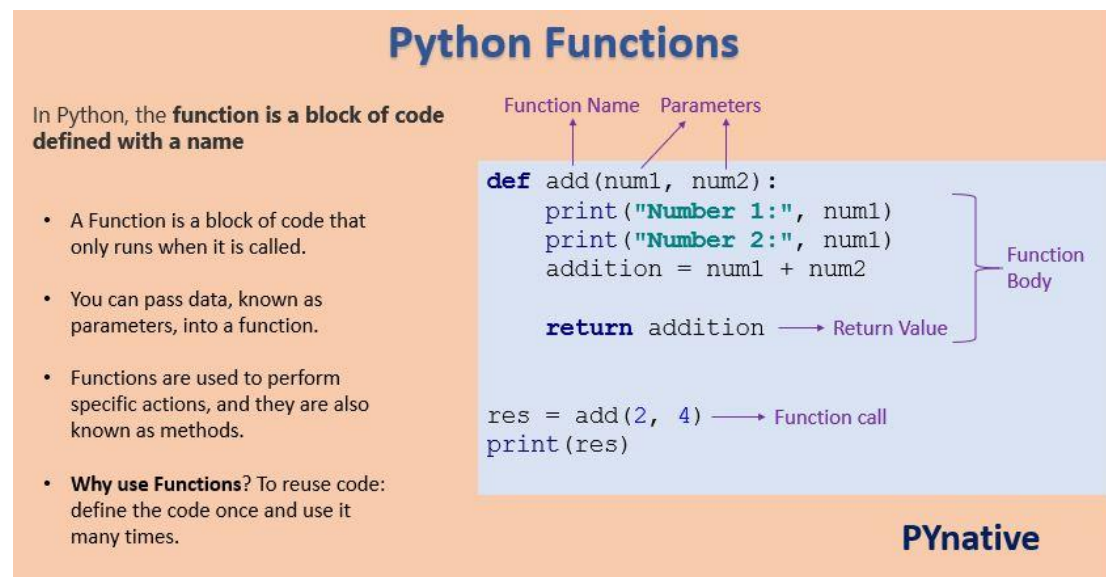
The body of function consists of one or more valid statements. Each statement must be intended with the same indentation to form a block.

Once the function is invoked, the formal parameters inside the parentheses become arguments.

6. Return statement

A return statement ends the function call and returns a value from a function back to the calling code. It is optional. If we do not define return statement inside the function's body, the function returns the object 'None'.

We will learn about the return statement in the further tutorial in more detail.



Python Functions

In Python, the **function is a block of code defined with a name**

- A Function is a block of code that only runs when it is called.
- You can pass data, known as parameters, into a function.
- Functions are used to perform specific actions, and they are also known as methods.
- **Why use Functions?** To reuse code: define the code once and use it many times.

```
def add(num1, num2):  
    print("Number 1:", num1)  
    print("Number 2:", num1)  
    addition = num1 + num2  
  
    return addition
```

res = add(2, 4)
print(res)

Labels in the diagram: Function Name (points to 'def'), Parameters (points to 'num1, num2'), Return Value (points to 'addition'), Function Body (bracketed around the function code), Function call (points to 'add(2, 4)').

PYnative

7. Function call