Functions in Python

Riya Jacob K
Assistant Professor on Contract
Dept of Computer Applications

FUNCTIONS

- A **function** is a **block of organized, reusable code** that is used to perform a **single,** related **action**.

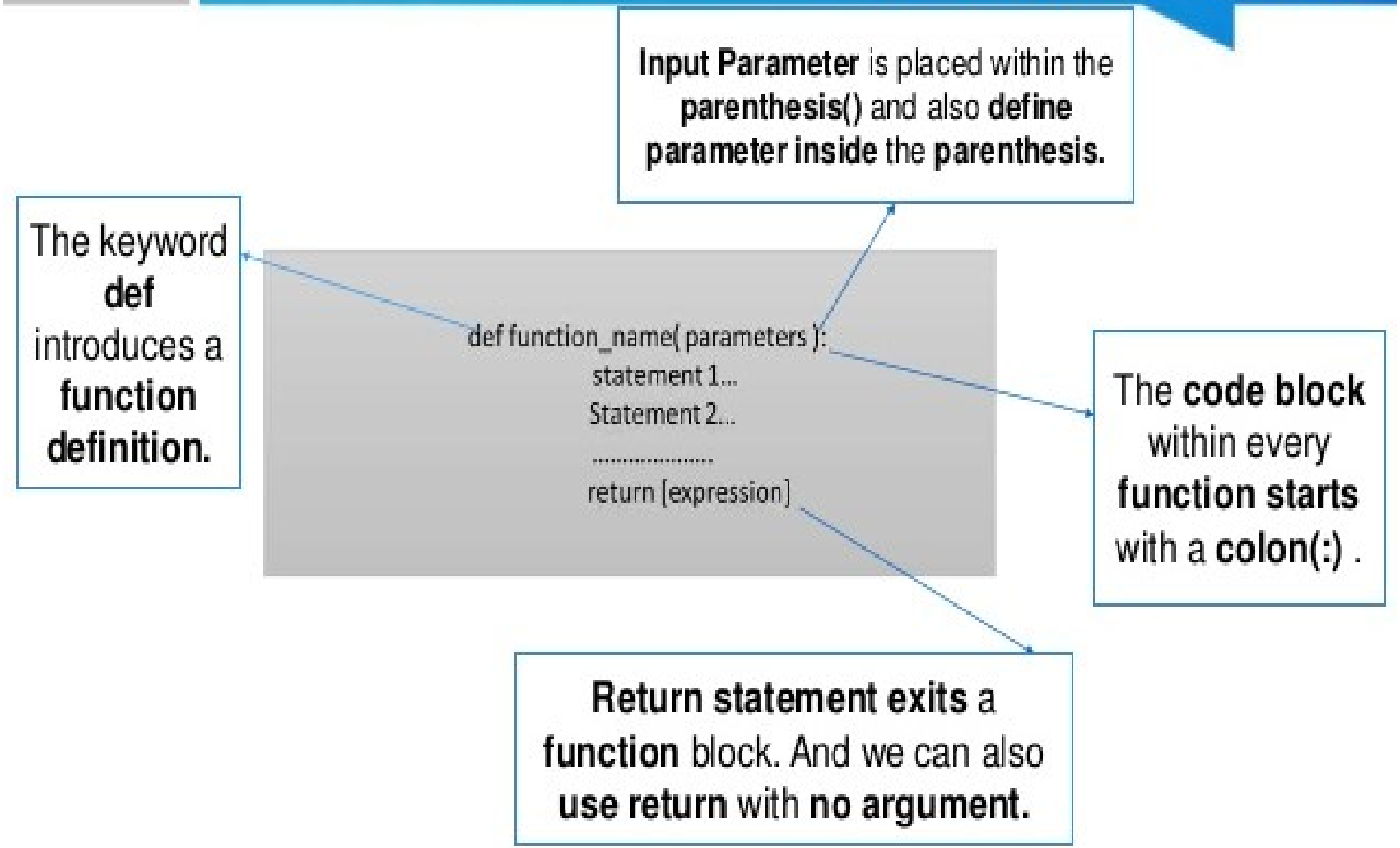
- **Functions** provide **better modularity** for the applications.

- **Functions** provide a **high degree of code reusing**.

RULES FOR DEFINING FUNCTION IN PYTHON

- 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**. We also define **parameters inside** these **parentheses.**
- 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**.

Function Syntax



Example

```
# Defining function print_str(str1)
''' This function prints the string being passed as an
argument '''
def print_str(str1):
    print(str1)
    return

# Calling user-defined function print_str(str1)
print_str("Calling the user defined function")
```

Function is printing the argument 'str1'

Function Call

PASSING ARGUMENTS TO FUNCTIONS

- In programming, there are two ways in which arguments can be passed to functions :-
  - **Pass by Value**:
    - Function creates a copy of the variable(Object in Python) passed to it as an argument.
    - The actual object is not affected.
    - Object is of **immutable type,**because immutable objects cannot be modified.
  - **Pass by Reference**:
    - The actual object is passed to the called function.
    - All the changes made to the object inside the function affect its original value.
    - Object is **mutable type**, as mutable objects can be changed, the passed objects are updated.