WELCOME EVERYONE!

Prayer

REVIEW

1. What are Python conditions?

2. What are Python loops?

Pretest

Instruction: Time Duration: 10 minutes Direction: Choose the correct answer. Write the letter of your answer on a clean ¼ sheet of paper.

1. What is a Python function?
   a. A loop in Python.
   b. A way to define a block of reusable code.
   c. A data type in Python.
   d. A comment in Python.
2. How do you define a Python function?
   a. By using the "for" keyword.
   b. By using the "if" keyword.
   c. By using the "def" keyword.
   d. By using the "return" keyword.
3. What is a parameter in a Python function?
   a. A value returned by the function.
   b. A variable used inside the function.
   c. A function's name.
   d. A loop that repeats the function.
4. Which keyword is used to call a function in Python?
   a. "call"
   b. "execute"
   c. "run"
   d. The function's name is followed by parentheses.
5. What is the purpose of the "return" statement in a function?
   a. To define the function.
   b. To call the function.
   c. To repeat the function.
   d. To send a value back as the result of the function.

Pretest

Answers

b) A way to define a block of reusable code.

c) By using the "def" keyword.

b) A variable used inside the function.

d) The function's name is followed by parentheses.

d) To send a value back as the result of the function.

PYTHON FUNCTION

Learning Objectives

01 Recognize python functions;

02 Demonstrate programs using python functions; and

03 Construct python programs using functions.

Introduction

Python Functions is a block of statements that return the specific task. The idea is to put some commonly or repeatedly done tasks together and make a function so that instead of writing the same code again and again for different inputs, we can do the function calls to reuse the code contained in it repeatedly. Increased code reusability and readability are some of the benefits of using functions.

There are two types of functions in Python which are built-in library functions and user-defined functions.

Python Functions

Built-in functions

Python's standard library includes several built-in functions. Some of Python's built-in functions are **print(), len(), sum(),** etc. These functions are always available, as they are loaded into the computer's memory as soon as you start the Python interpreter.

Built-in Function

Built-in Function

print()

is used to display or output text or other data to the console or standard output.

```
print("Hello, World!")  # Display a simple text message
x = 10
print("The value of x is:", x)  # Display the value of a variable
```

len()

is used to determine the length or the number of items in a sequence or collection.

```
text = "Hello, World!"
length = len(text)  # Returns 13
```

sum()

is used to calculate the sum of all the elements in an iterable (such as a list, tuple, or set) or from values in an iterable and a starting value.

```
numbers = [1, 2, 3, 4, 5]
total = sum(numbers)  # Returns 15 (1 + 2 + 3 + 4 + 5)
```

User-defined Functions

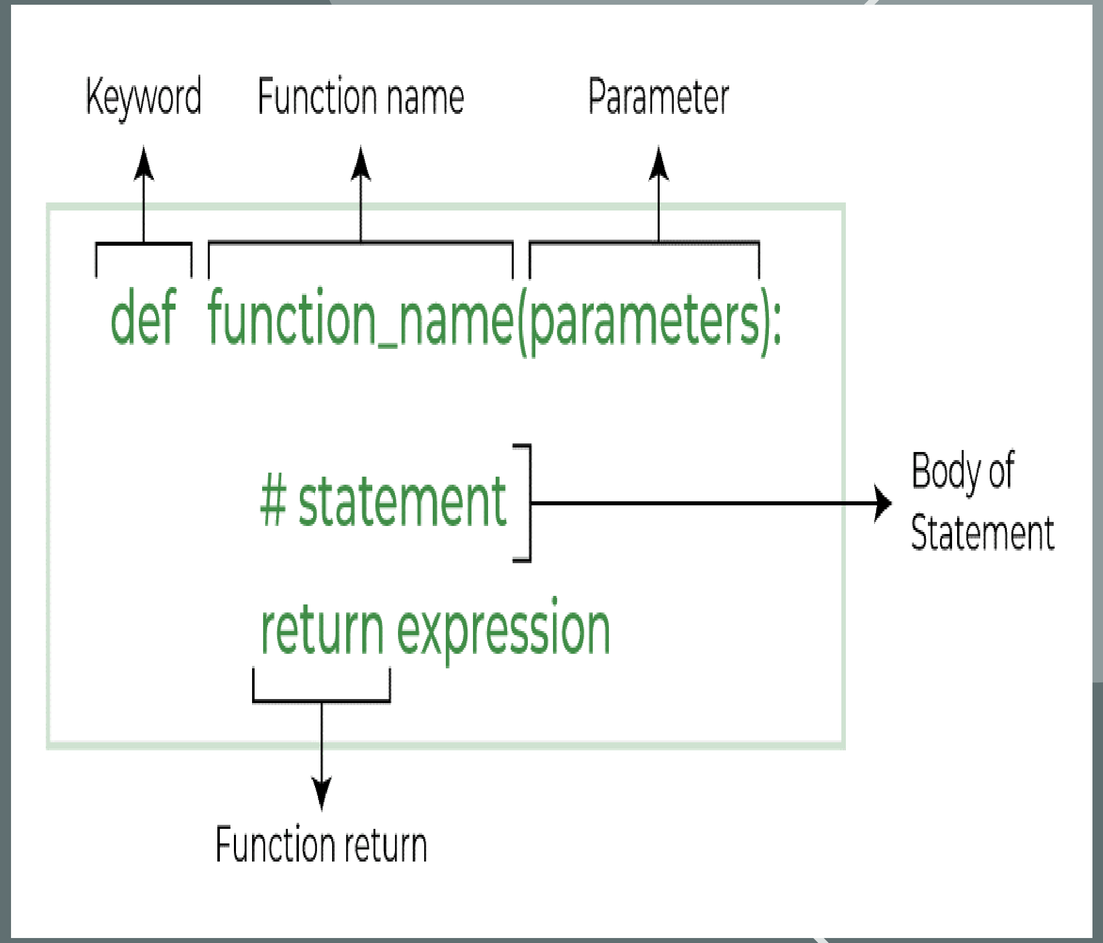
User-defined Functions

def **-** keyword used to declare a function.

function_name **-** any name given to the function.

arguments **-** any value passed to the function.

return (optional) **-** returns value from a function.



Rules for Python Variables

A variable name must start with a letter or the underscore character

A variable name cannot start with a number

A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)

Variable names are case-sensitive (age, Age and AGE are three different variables)

Function

Are used to organize and divide specific task in a program that will only run when it is called.

Creating a Function

We can create a user-defined function in Python, using the def keyword. We can add any type of functionalities and properties to it as we require.

Syntax:

```
def function_name():
        #Code
```

Example:

```
def greet():
        print(“Hello World!”)
```

Calling a Function

We can create a user-defined function in Python, using the def keyword. We can add any type of functionalities and properties to it as we require.

Syntax:

```
def function_name():
        #Code
```

Example:

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

greet()
```

Creating a Function Return type

Return Value is a special statement you can use inside a function or method to send the function's result back to the caller.

Syntax:

```
def function_name():
        #Code
```

Example:

```
def find_square(num):
        return num * num

square = find_square(3)

print(square)
```

Let’s try an example

Create a program that will make the user input a number, the program should SQUARE the number then print it.

EXAMPLE

Input: 5

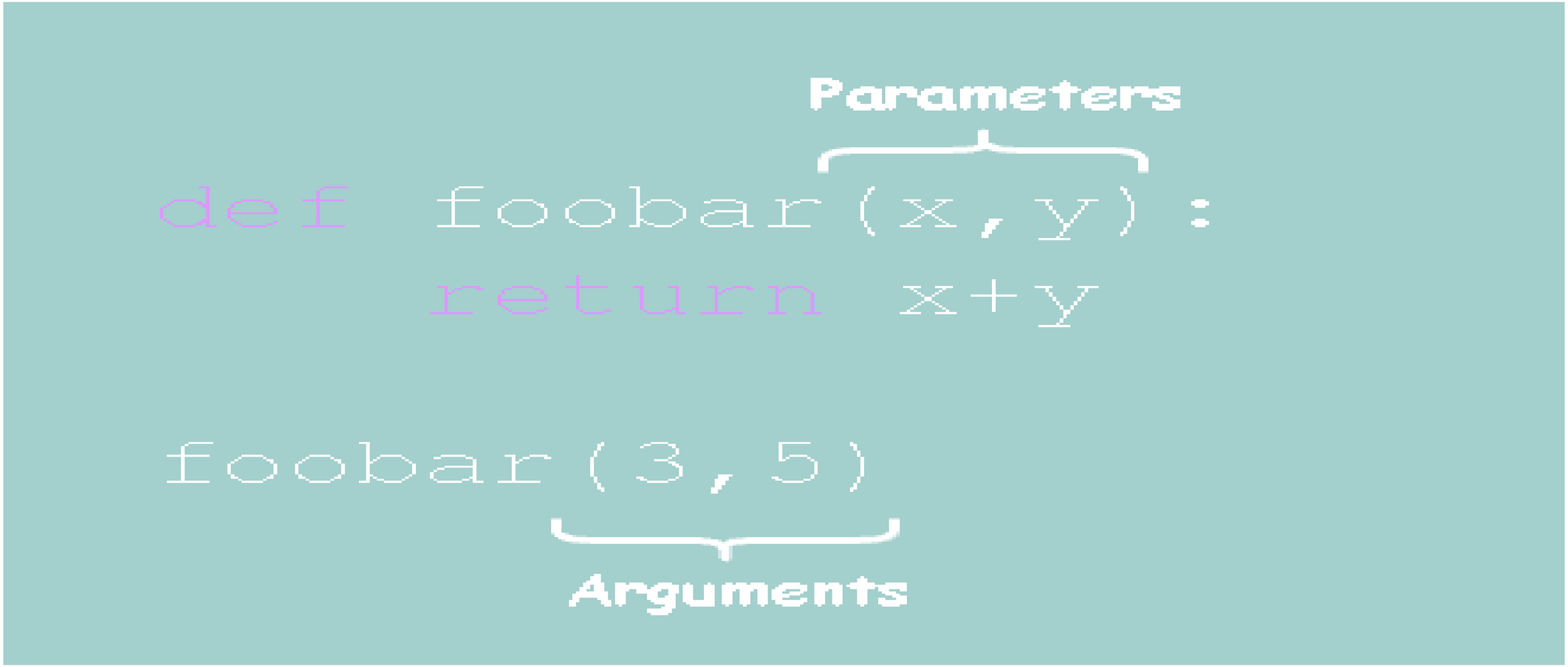
Output: 25

What are the parameters and arguments?

A **parameter** is a variable in a function definition. It is a placeholder and hence does not have a concrete value.

An **argument** is a value passed during function invocation.

What are the parameters and arguments?



Creating a Function Argument

A function can have any number of arguments separated by a comma.

Syntax:

```
def function_name(y,x):
    #Code
```

Example:

```
def add_numbers(num1, num2):
    return num1 + num2
```

Calling a Function Argument and Providing Arguments

A function can have any number of arguments separated by a comma.

Syntax:

```
def function_name(y,x):
    #Code
```

Example:

```
def add_numbers(num1, num2):
    return num1 + num2

add_numbers(5,4)
```

What are default Arguments?

A default argument is a parameter that assumes a default value if a value is not provided in the function call for that argument. The following example illustrates Default arguments.

Syntax:

```
def function_name(y, x=50):
        #Code
```

Example:

```
def function_name(y, x=50):
        print("y: ", y)
        print("x: ", x)

funtion_name(10)
```

Try to analyze and give your answer

```
def my_function(country = "Norway"):
        print("I am from " + country)

my_function("Sweden")
my_function("India")
my_function()
my_function("Brazil")
```

Post Test

Instruction: Time Duration: 10 minutes Direction: Choose the correct answer. Write the letter of your answer on a clean ¼ sheet of paper.

1. Functions that we can create on our own based on our requirements.
    a. User-defined Function
    b. User-declared Function
    c. Defined-user Function
    d. Declared-user Function
2. It is a variable in a function definition. It is a placeholder and hence does not have a concrete value.
    a. Argument
    b. Function
    c. Parameter
    d. Placeholder
3. It is the actual value or expression provided to a function when it's called.
    a. Argument
    b. Function
    c. Parameter
    d. Placeholder

Post Test

Instruction: Time Duration: 10 minutes Direction: Choose the correct answer. Write the letter of your answer on a clean ¼ sheet of paper.

4. A function can have any number of arguments separated by a _____.
    a. colon
    b. semicolon
    c. comma
    d. parenthesis
5. After creating a function in Python, we can call it by using the name of the function followed by parenthesis containing _____ of that particular function.
    a. argument
    b. name
    c. parameters
    d. variable

Post Test

Instruction: Time Duration: 10 minutes Direction: Choose the correct answer. Write the letter of your answer on a clean ¼ sheet of paper.

II. Identification: Provide what is being asked.

6-7. Give 2 examples of built-in functions.

8-10. Give 3 rules in naming a function.

Pretest

Answers

1. A
2. C
3. A
4. C
5. C

6-7. print(), len(), sum()

8-10. A variable name must start with a letter or the underscore character, cannot start with a number, can only contain alpha-numeric characters and underscores (A-z, 0-9, and _), or are case-sensitive (age, Age, and AGE are three different variables)

ASSIGNMENT

Advance Assignment

Directions: Conduct preliminary research on the following:

- Python Modules
- Python Packages

Thank you!

References

PranathiBadugu (2023, May 04). Python Functions; Geeksforgeeks.
https://www.geeksforgeeks.org/python-functions/

Mustafeez, A.N. (2023). Parameter vs. argument; Educative.io.
https://www.educative.io/answers/parameter-vs-argument

Tutorialspoint (2023). Python – Functions: Tutorialspoint
https://www.tutorialspoint.com/python/python_functions.htm