Newton School of Technology

8

Functions

by Gladden Rumao

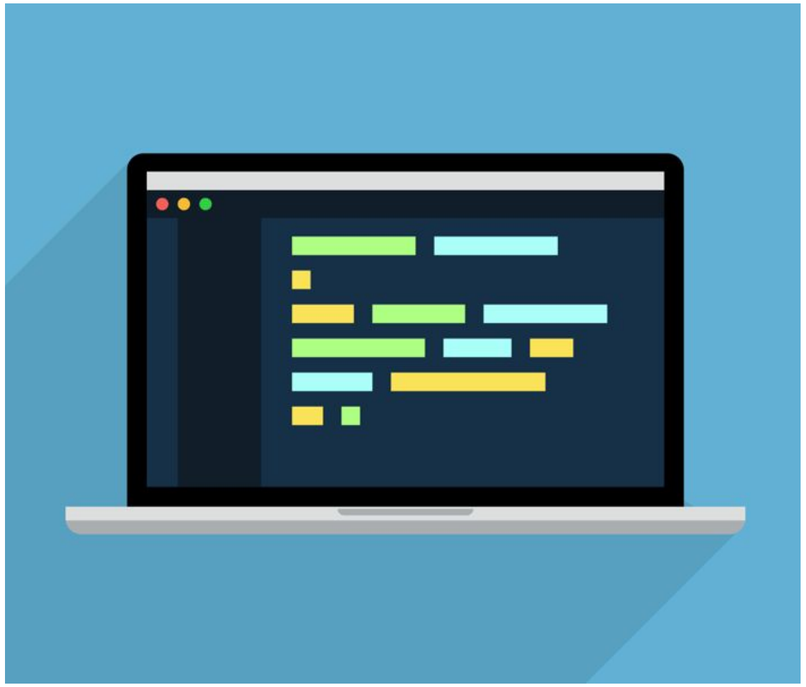
CSA101 : Problem Solving with Programming

Newton School of Technology

Recap of Previous Lecture!

Quick Recap :

- Nested if else
- Logical Operators
- AND
- OR
- NOT

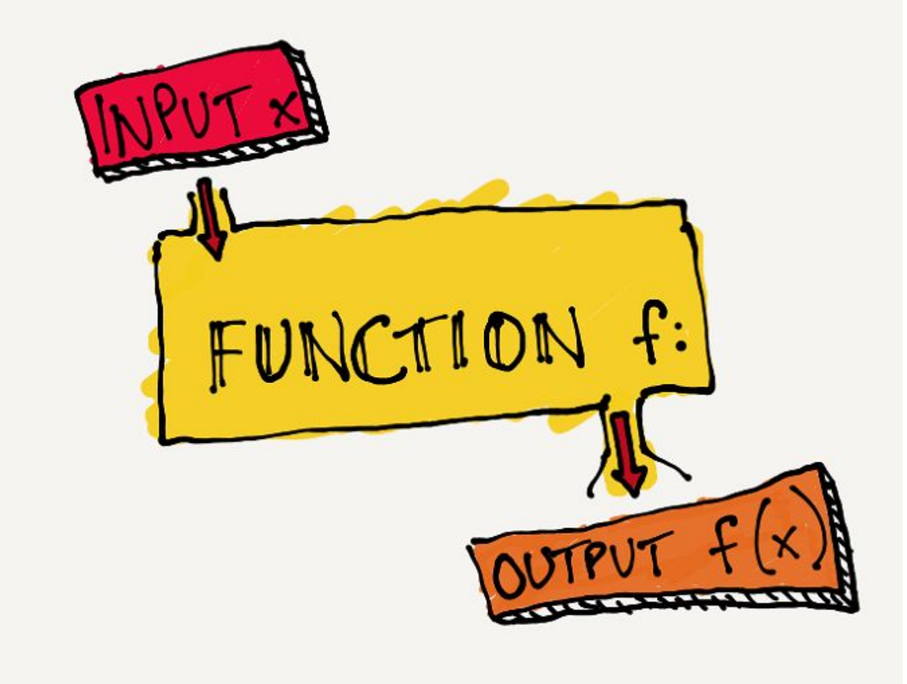


Newton School of Technology

Functions

What is a function?

A function is a block of **organized**, **reusable** code that is used to perform a **single specific** action. Functions provide a better **modularity** for your application and a high degree of **code reusability**.



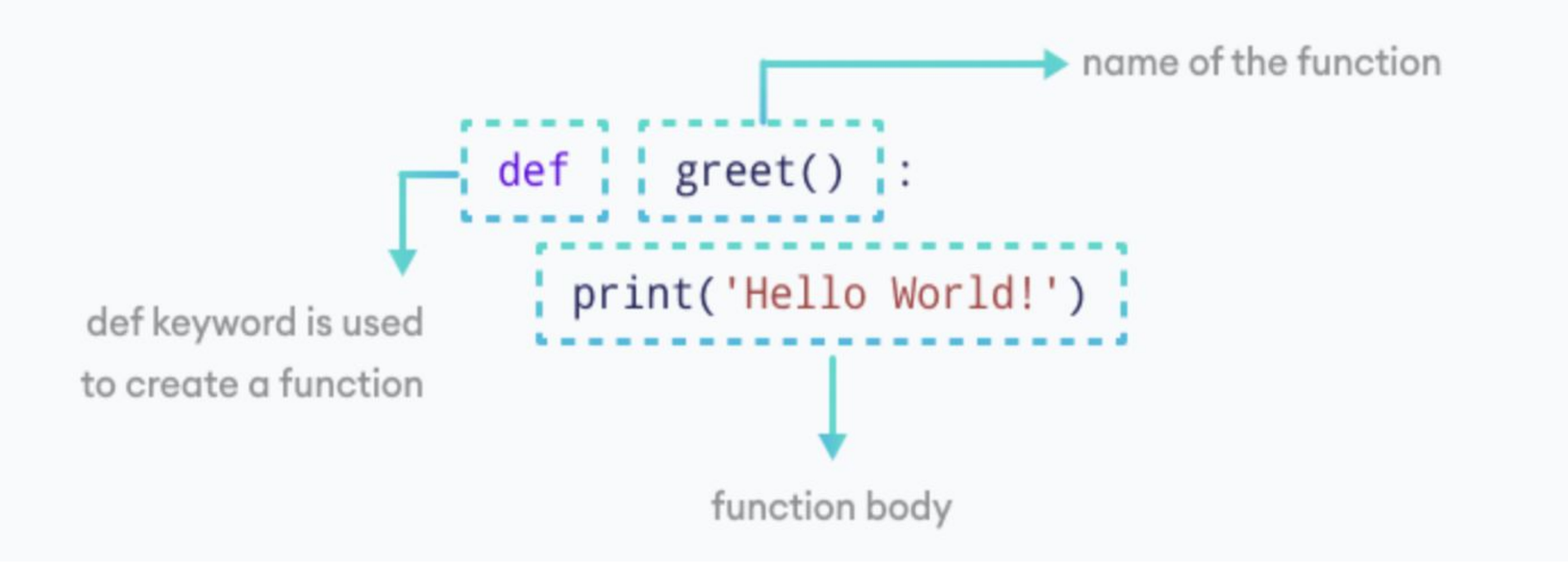
Purpose of using function :

1. Code Reusability
2. Improved readability
3. Modularity
4. Abstraction
5. Helps in Testing and debugging

Overall makes life easy for you.

Defining a Function (Without Parameters):

```
def function_name():
    #body of the function
    print("Hey! I am body of function.")
```



Newton School of Technology

Function to add two numbers

Defining a Function (With Parameters):

```
#Function to add two numbers
def add_two_nums(a, b):
    c = a + b
    print(c)
```

Calling a function:

Just call the name.

```
greet()

add_two_nums(10, 20)
```

Newton School of Technology

In-Class Question!

Spot the difference :

```python
#Function to add two numbers
def add_two_nums(a, b):
    c = a + b
    print(c)
```

```python
#Function to add two numbers
def add_two_nums(a, b):
    c = a + b
    return c
```

Return statement:

1. The **return** statement in Python is used to **exit** a function.

2. It also sends a value **back** to the **function caller**.

3. If a function **does not** have a return statement, it returns **None** by default.

Guess the output:

```
#Function to add two numbers
def add_two_nums(a, b):
    c = a + b
    return c

d = add_two_nums(10, 20)
print(d)
```

Guess the output:

```python
def add_and_subtract_two_nums(a, b):
    c = a + b
    return c
    d = b - a
    return d

print(add_and_subtract_two_nums(10, 20))
```

5 star Question

What will be the output?

```python
def add_and_subtract_two_nums(a, b):
    c = a + b
    return c
    d = b - a
    return d

print(add_and_subtract_two_nums(10, 20))
```

5 star Question

Once the return statement is executed in a function it exits and rest of the code is not executed.

Newton School of Technology

Function Arguments!

What are arguments ?

Types of Function Arguments:

1. **Positional Arguments.**
2. **Default Arguments.**
3. **Keyword Arguments.**

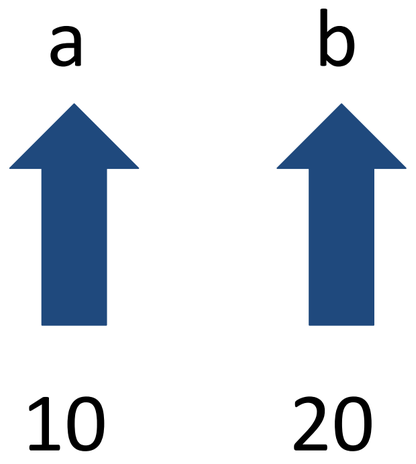
Positional Arguments:

When you call a function, you provide values for its parameters in the **exact order** they are defined. These values are then assigned to the corresponding parameters in the **same order**.

Positional Arguments: Example

```python
#Function to add two numbers
def add_two_nums(a, b):
    c = a + b
    print(c)

add_two_nums(10, 20)
```



Default Arguments: Example

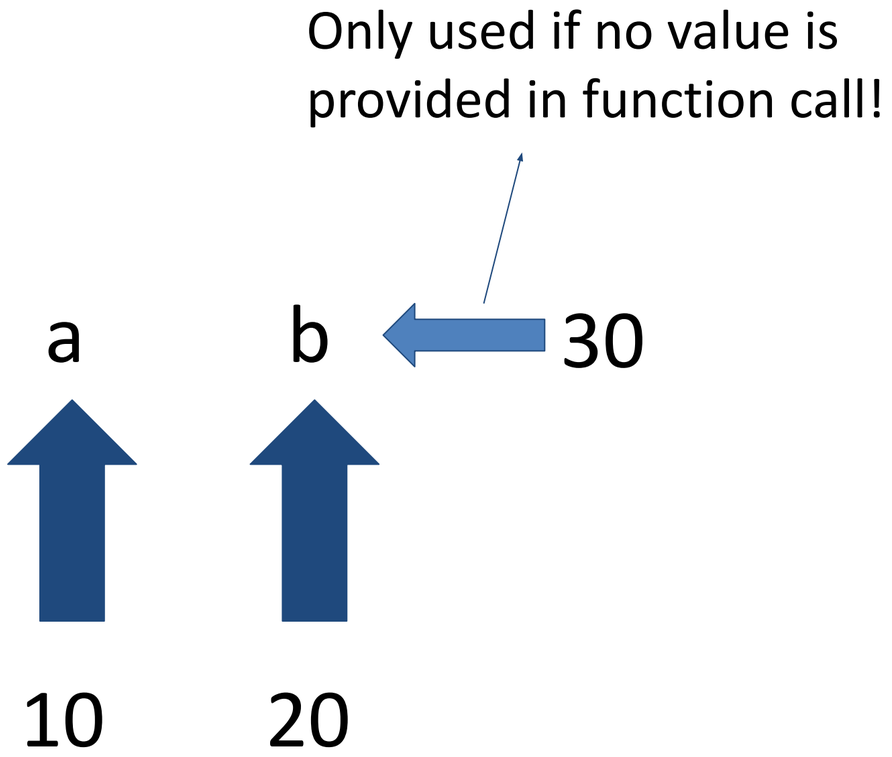
If an argument is not provided when the function is called, the default value specified in the function definition is used.

Note: Default arguments must come after any positional arguments in the function definition.

Default Arguments: Example

```python
#Function to add two numbers
def add_two_nums(a, b=30):
    c = a + b
    print(c)

add_two_nums(10, 20)
add_two_nums(10)
```

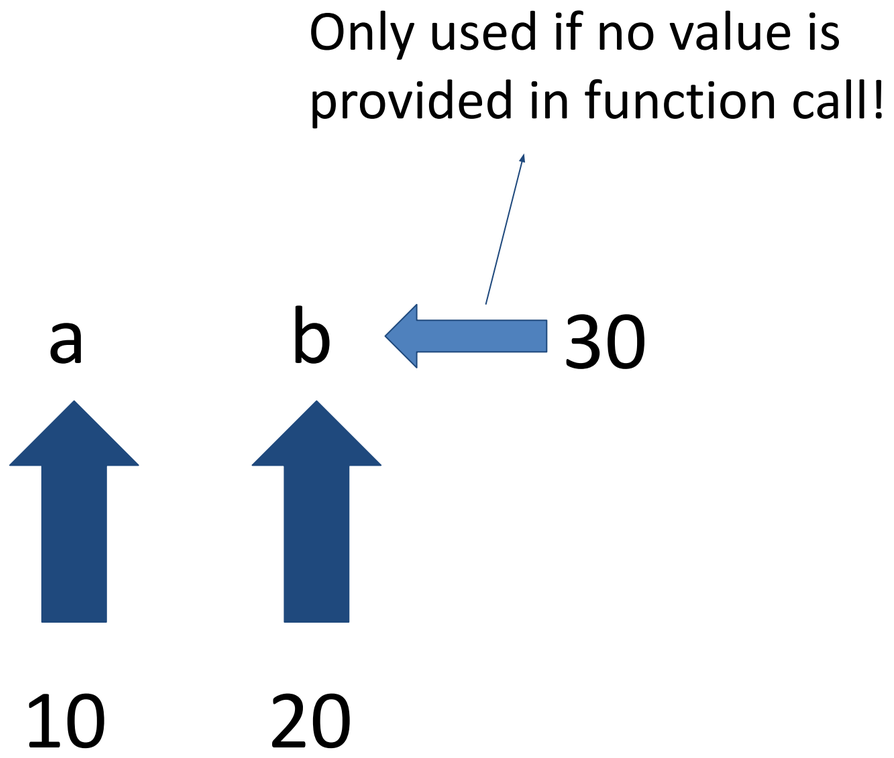


Default Arguments: Example

```python
#Function to add two numbers
def add_two_nums(a, b=30):
    c = a + b
    print(c)

add_two_nums(10, 20)
add_two_nums(10)
```

Output:
30
40



Keyword Arguments:

Keyword arguments in Python allow you to call a function by specifying the **names of the parameters**.

This makes the function call more **explicit** and can improve **readability**.

Note: Keyword arguments must come after any positional arguments in the function definition.

Keyword Arguments: Examples

```
#Function to add two numbers
def add_two_nums(a, b):
    c = a + b
    print(c)

add_two_nums(b=40, a=20)
```

```
def add_two_nums(a, b=20):
    c = a + b
    print(c)

print(add_two_nums(b=10, a=10))
```

Keyword Arguments: Examples

```python
#Function to add two numbers
def add_two_nums(a, b):
    c = a + b
    print(c)

add_two_nums(b=40, a=20)
```

Output:
60

```python
def add_two_nums(a, b=20):
    c = a + b
    print(c)

print(add_two_nums(b=10, a=10))
```

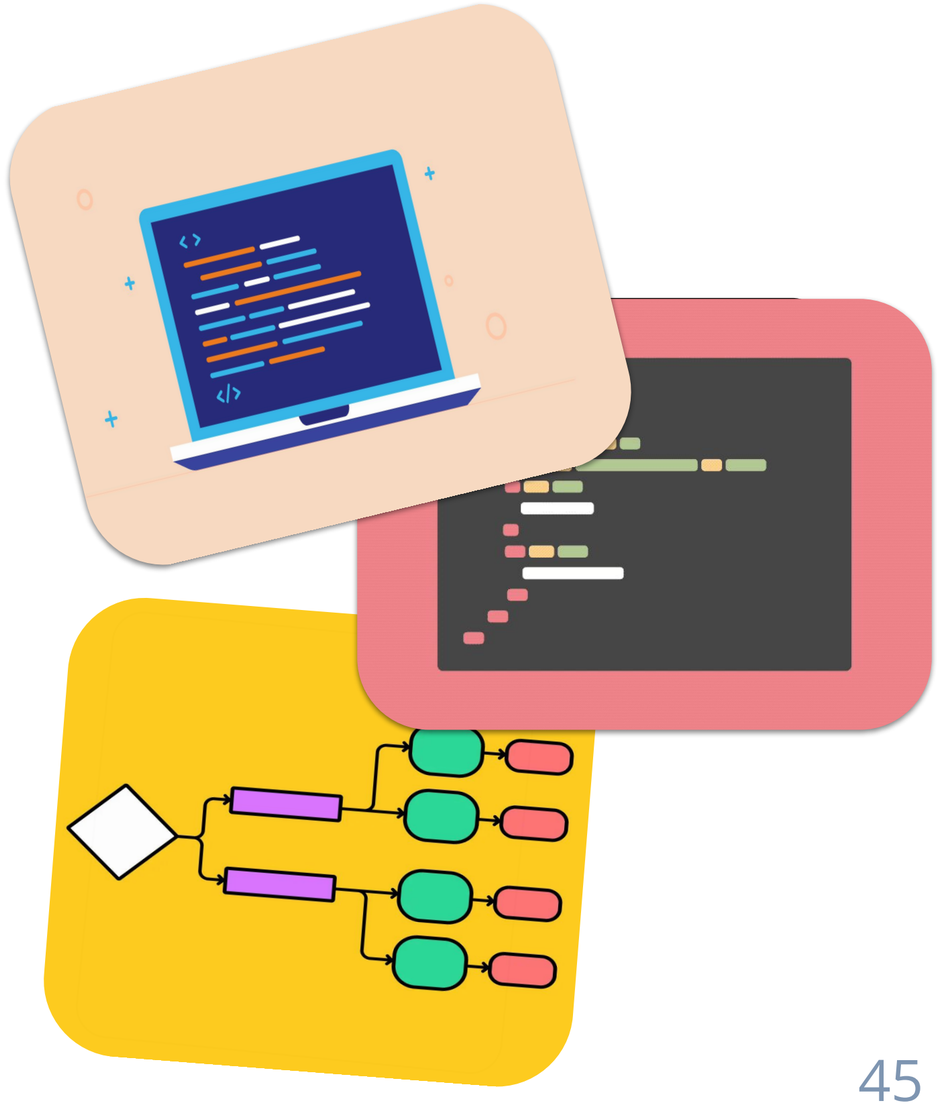
Output:
20

Newton School of Technology

In-Class Questions!

Summary

- **Functions**:
  - Definition and purpose
  - Defining, calling, arguments and return statement
  - Type of arguments



Thank You!