

02_In-Class_Python_Defining_a_Function

ITF-010-Python+02-DefiningFunctions
Training Clarusway
Pear Deck - November 17, 2021 at 8:35PM

Part 1 - Summary

Use this space to summarize your thoughts on the lesson

Part 2 - Responses

Slide 1



Use this space to take notes:

Slide 2

Table of Contents



- ▶ Main Principles of 'Defining'
- ▶ Execution of a Function

CLARUSWAY®
WAY TO REFINEMENT YOURSELF



Use this space to take notes:

Slide 3



1 Main Principles of 'Defining'

CLARUSWAY®
WAY TO REFINEMENT YOURSELF

Use this space to take notes:

Slide 4

Your Response

How was the pre-class content?

Students, drag the icon!

Pear Deck

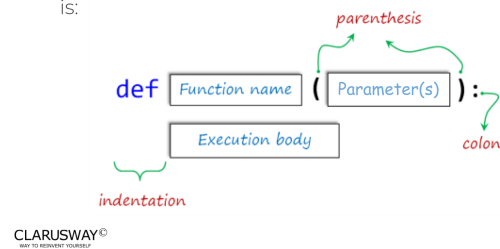
Pear Deck Interactive Slide
Do not refresh this slide

Use this space to take notes:

Slide 5

► Main Principles of 'Defining' (review) ►

- The basic **formula syntax** of user-defined function is:

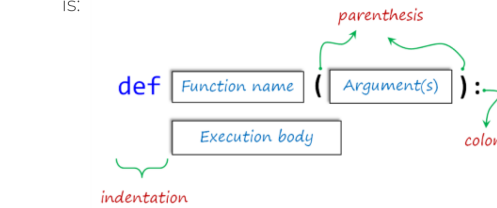


Use this space to take notes:

Slide 6

► Main Principles of 'Defining' (review) »

- The basic **formula syntax** of user-defined function is:



CLARUSWAY®
WAY TO REINVENT YOURSELF

6

Use this space to take notes:

Slide 7

► Main Principles of 'Defining' (review) »

- Defining a simple function 📌

```
1 def first_function(argument_1, argument_2) :  
2     print(argument_1**2 + argument_2**2)
```

$$\text{argument_1}^2 + \text{argument_2}^2$$

CLARUSWAY®
WAY TO REINVENT YOURSELF

7

Use this space to take notes:

Slide 8

► Main Principles of 'Defining' (review) »

- Let's call and use `first_function`.

```
1 first_function(2, 3) # here, the values (2 and 3) are
   allocated to the arguments
```

Use this space to take notes:

Slide 9

► Main Principles of 'Defining' (review) »

- Let's call and use `first_function`.

```
1 first_function(2, 3) # here, the values (2 and 3) are
   allocated to the arguments
```

```
1 13
```

Use this space to take notes:

Slide 10

Your Response

Answer 1:
15 -2.5 amazingamazingamazing

► Main Principles of 'Defining' (review) »

- Let's define the multiplying function `multiply(a, b)`.

```
1 def multiply(a, b) :  
2     print(a * b)  
3  
4 multiply(3, 5)  
5 multiply(-1, 2.5)  
6 multiply('amazing ', 3) # it's really amazing, right?
```

What is the output? Try to figure out in your mind...



CLARUSWAY
LEARN BY YOURSELF, write your response!

Peer Deck Interactive Slide

Use this space to take notes:

Slide 11

► Main Principles of 'Defining' »

- Let's define the multiplying function `multiply(a, b)`.

```
1 def multiply(a, b) :  
2     print(a * b)  
3  
4 multiply(3, 5)  
5 multiply(-1, 2.5)  
6 multiply('amazing ', 3) # it's really amazing, right?
```

```
1 15  
2 -2.5  
3 amazing amazing amazing
```

CLARUSWAY®
WAY TO REINVENT YOURSELF



Use this space to take notes:

Slide 12

► Main Principles of 'Defining' (review) »

- Let's give an example by leaving the parentheses empty.

```
1 def motto() :  
2     print("Don't hesitate to reinvent yourself!")  
3  
4 motto() # it takes no argument
```

What is the output? Try to figure out in your mind...

CLARUSWAY®
WAY TO REINVENT YOURSELF

12

Use this space to take notes:

Slide 13

► Main Principles of 'Defining' (review) »

- Let's give an example by leaving the parentheses empty.

```
1 def motto() :  
2     print("Don't hesitate to reinvent yourself!")  
3  
4 motto() # it takes no argument
```

```
1 Don't hesitate to reinvent yourself!
```

CLARUSWAY®
WAY TO REINVENT YOURSELF

13

Use this space to take notes:

Slide 14

► Main Principles of 'Defining' »

► Task :

- Define a function named `add` to sum two numbers and print the result.

Use this space to take notes:

Slide 15

► Main Principles of 'Defining' (review) »

► The code can be like :

```
1 def add(a, b):  
2     print(a + b)  
3  
4 add(-3, 5)  
5
```

Output

```
2
```

Use this space to take notes:

Slide 16

► Main Principles of 'Defining' »

► Task :

- Define a function named `calculator` to calculate four math operations with two numbers and print the result.
- Warn user in case of wrong entry : `"Enter valid arguments"`

```
1  
2 calculator(88, 22, "+")  
3
```

Output

```
110
```

CLARUSWAY®
WAY TO REINVENT YOURSELF

16

Use this space to take notes:

Slide 17

► Main Principles of 'Defining' »

► The code might be like :

```
1 def calculator(x, y, opr):  
2     if opr == "+":  
3         print(x + y)  
4     elif opr == "-":  
5         print(x - y)  
6     elif opr == "*":  
7         print(x * y)  
8     elif opr == "/":  
9         print(x / y)  
10    else:  
11        print("enter valid arguments!")
```

CLARUSWAY®
WAY TO REINVENT YOURSELF

17

Use this space to take notes:

Slide 18



2 Execution of a Function

CLARUSWAY®
DATA SCIENCE FOUNDATION

Use this space to take notes:

Slide 19

► Execution of a Function (review) »

- The result of a function :

- `print`
- `return`



```
def multiply_1(a, b) :  
    print(a * b) # it prints something  
multiply_1(10, 5)
```

CLARUSWAY®
DATA SCIENCE FOUNDATION



Use this space to take notes:

Slide 20

► Execution of a Function (review) »

- The result of a function :

- `print`
- `return`

```
def multiply_1(a, b) :  
    print(a * b) # it prints something  
multiply_1(10, 5)
```

50

Use this space to take notes:

Slide 21

► Execution of a Function (review) »

- The result of a function :

- `print`
- `return`

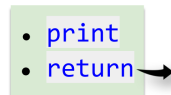
```
def multiply_2(a, b) :  
    return(a * b) # returns any numeric  
data type value  
print(multiply_2(10, 5))
```

Use this space to take notes:

Slide 22

► Execution of a Function (review) »

- The result of a function :



```
def multiply_2(a, b) :  
    return(a * b) # returns any numeric  
                  data type value  
print(multiply_2(10, 5))  
50
```

Use this space to take notes:

Slide 23

► Execution of a Function (review) »

- Compare the usage options :

```
1 print(type(multiply_1(10, 5)))  
2 print(type(multiply_2(10, 5)))
```

Use this space to take notes:

Slide 24

► Execution of a Function (review) »

- The outputs are :

```
1 print(type(multiply_1(10, 5)))  
2 print(type(multiply_2(10, 5)))
```

```
1 50  
2 <class 'NoneType'>  
3 <class 'int'>
```

CLARUSWAY®
WAY TO REINVENT YOURSELF

24

Use this space to take notes:

Slide 25

► Main Principles of 'Defining' »

► Task :

- Define a function named `calculator` to calculate four math operations with two numbers and `return` the result.

```
1  
2 print(calculator(-12, 2, "+"))  
3
```

Output

```
-10
```

CLARUSWAY®
WAY TO REINVENT YOURSELF

25

Use this space to take notes:

Slide 26

► Main Principles of 'Defining' »

► The code might be like :

```
1 def calculator(x, y, o):
2     if o == "+":
3         return(x + y)
4     elif o == "-":
5         return(x - y)
6     elif o == "*":
7         return(x * y)
8     elif o == "/":
9         return(x / y)
10    else : return ("enter valid arguments!")
11
```

CLARUSWAY®
WAY TO REINVENT YOURSELF

28

Use this space to take notes:

Slide 27

Your Response

► Main Principles of 'Defining' »

► Task :

- Define a function named `absolute_value` to calculate and return absolute value of the entered number.
- You can add docstring for an explanation.

```
1 print(absolute_value(3.3))
2 print(absolute_value(-4))
3
```

Output

```
3.3
4
```



Now, write your response!

Peer Deck Interactive Slide

Use this space to take notes:

Slide 28

► Main Principles of 'Defining'

► The code might be like :

```
1 def absolute_value(num):
2     """This function returns the absolute
3     value of the entered number"""
4
5     if num >= 0:
6         return num
7     else:
8         return -num
9
10 print(absolute_value.__doc__)
11
```

By the way, we can display the docstring of this function

Output

```
This function returns the absolute
value of the entered number
```

Use this space to take notes:

Slide 29

THANKS!

End of the Lesson
(Defining a Function)

next Lesson

The Matter of Arguments

click above

CLARUSWAY®
WAY TO REPRESENT YOURSELF

Link(s) on this slide:

- <https://lms.clarusway.com/course/view.php?id=21§ion=1>

Use this space to take notes: