Module 3. Lesson 2. Functions. Continued

Confirmation of qualifications

To get started on the working tasks, demonstrate **your knowledge level** .

Prove that you are ready for the brainstorm!

Confirmation of qualifications

What is a function ?
What standard functions do you know?

Confirmation of qualifications

A function is

a named set of program commands that can be called from another part of the program.

Name	*Arguments*	*Result of work*
`print()`	Any number of numbers, strings, logical values	`None` — a special value keyword (we will say that such a function “does not return anything”)
`input()`	One string or nothing	Row
`int()`	One number or string	Integer number
`len()`	One string	Integer number

Confirmation of qualifications

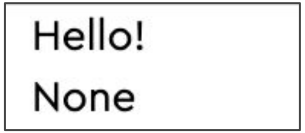
Do functions always return a value?

Confirmation of qualifications

Yes, but sometimes the result of a function’s operation is not defined.

```
print(print('Hello!'))
```

→

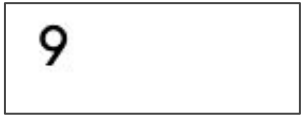


```
Hello!
None
```

print() will print the result of the operation of print(), which is not defined.

```
print(len('qwerty123'))
```

→



```
9
```

print() will print the result of the operation of len(), which is an integer.

In this case, programmers say that a function does not return a value.

How do we define our own function?

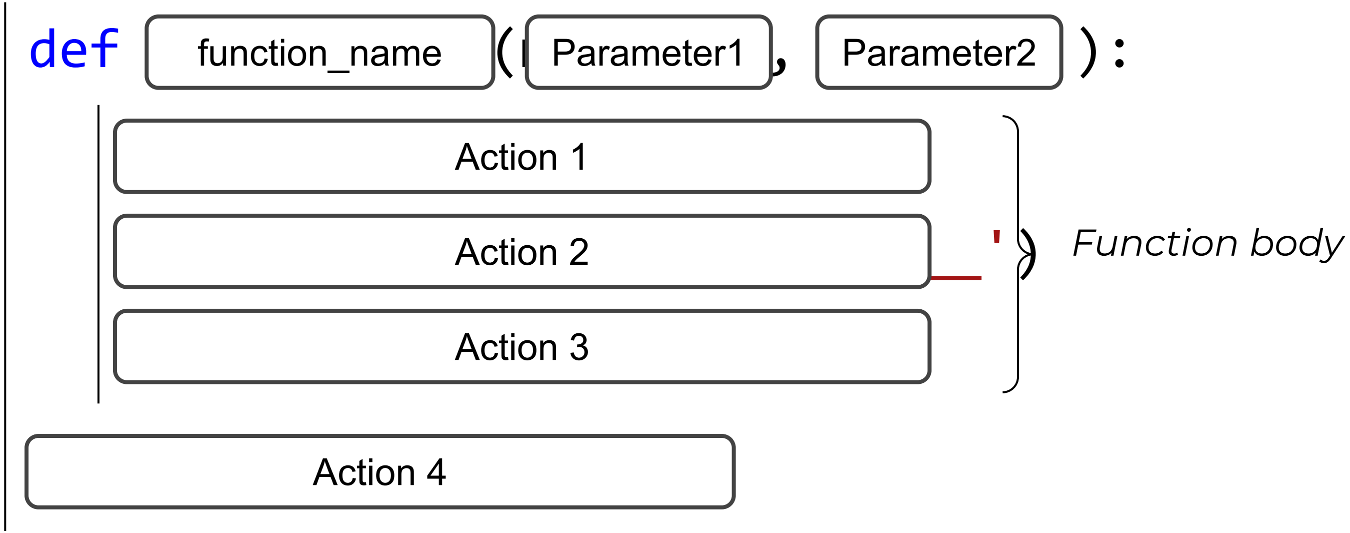
Confirmation of qualifications

Function declaration

The process of creating a function is called a **function declaration**.

To declare a function, you need to:

- □ specify the `def` operator;
- □ write the function name, list the parameters, and put a colon;
- □ describe in a programmatic manner how the function works.



def function_name (Parameter1 , Parameter2):

Action 1

Action 2

Action 3

Function body

Action 4

Confirmation of qualifications

How do we use a previously defined function in a program?

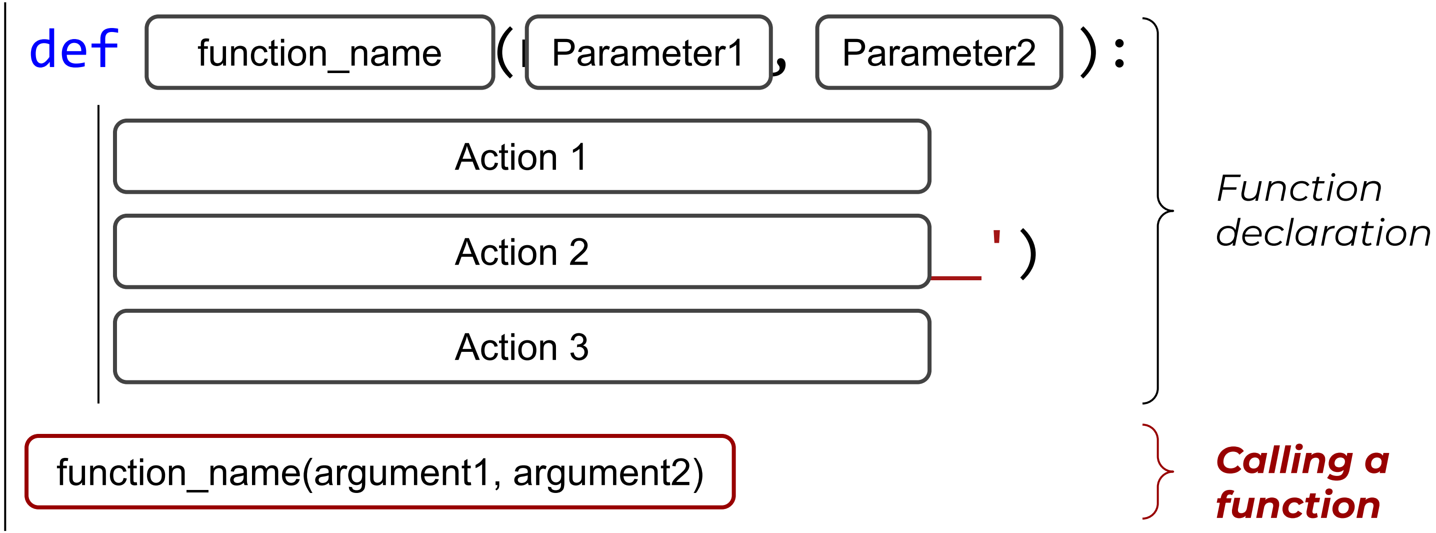
Confirmation of qualifications

Calling your own function

Pay attention!

When declaring a function, you list some variables called parameters. The values of those are assigned later when the function is called.

When calling a function, we pass arguments into it, i.e. concrete values (even if those are hidden behind variables).



```
def function_name ( Parameter1 , Parameter2 ):
    Action 1
    Action 2 ' )
    Action 3
function_name(argument1, argument2)
```

Function declaration

Calling a function

Confirmation of qualifications

What is local scope?

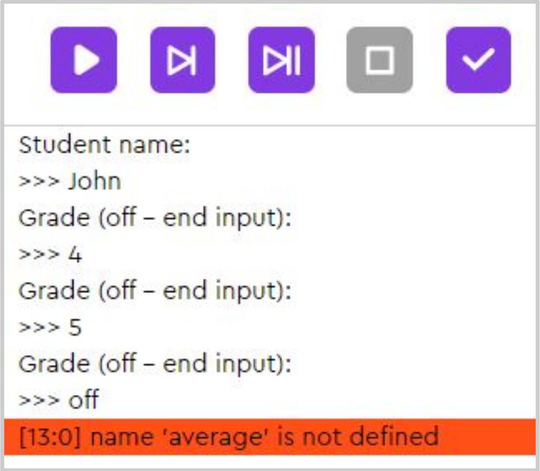
How does it affect the operation of the function and main program?

Confirmation of qualifications

Local scope

The Python interpreter links variables with where they are used. Programmer-defined **functions are standalone program blocks with their own variables**.

```
def average_grade(name):
    grade = input('Grade (off - end input):')
    summa = 0
    total = 0
    while grade != 'off':
        summa += int(grade)
        total += 1
        grade = input('Grade (off - end input):')
    average = summa/total


name = input('Student name:')
average_grade(name)
print(average)
```



```
Student name:
>>> John
Grade (off – end input):
>>> 4
Grade (off – end input):
>>> 5
Grade (off – end input):
>>> off
[13:0] name 'average' is not defined
```

The average variable has only been defined within the function. It has not been introduced in the main part of the program, so its value is unknown!

Confirmation of qualifications

How do we return a variable's value from our own function?

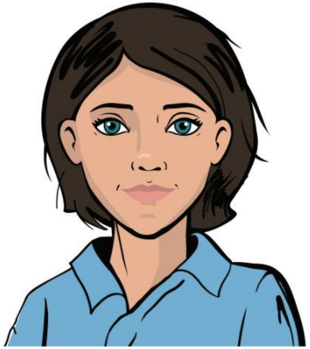
Confirmation of qualifications

Return operator

To get (return) a value from a function, we need to use the **return** operator. This value can be assigned to a variable in the main part of the program.

```
def average_grade(name):
    grade = input('Grade (off - end input):')
    summa = 0
    total = 0
    while grade != 'off':
        summa += int(grade)
        total += 1
        grade = input('Grade (off - end input):')
    average = summa/total
    return average

name = input('Student name:')
average = average_grade(name)
print(average)
```

The renewed `average_grade()` function takes 1 argument (name) and returns the value of average.



Confirmation of qualifications

Qualifications confirmed!

Great, you are ready to brainstorm and complete your work task!

Confirmation of qualifications

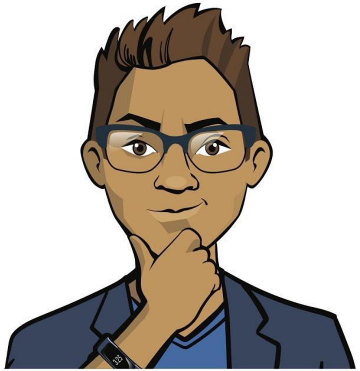
Brainstorm:

Nested functions

Important notice

We will analyze some tasks which require defining our own functions and calling one function from within another for the optimal solution.

Before that, we will formulate an important recommendation.



Brainstorm

One function — one purpose

Instead of declaring **one cumbersome function** , it is better to program **several compact ones** .

```
def get_result():
```

- ❑ **Reading** the data from the keyboard.
- ❑ **Calculation** based on the data.
- ❑ **Analyzing** the result.
- ❑ **The output** of the information for the user.

Call `get_result()`

Let's have a look at an arbitrary function. As it is now, this is just the main part of the program rewritten as a function.

How can we optimize the code?

Brainstorm

One function — one purpose

Instead of declaring **one cumbersome function** , it is better to program **several compact ones** .

```
def get_result():
```

- ❑ **Reading** the data from the keyboard.
- ❑ **Calculation** based on the data.

Let’s develop the `get_data()` function

- ❑ **Analyzing** the result.
- ❑ **The output** of the information for the user.

Let’s develop the `print_result()` function

Call `get_result()`

Brainstorm

One function — one purpose

Instead of declaring **one cumbersome function** , it is better to program **several compact ones** .

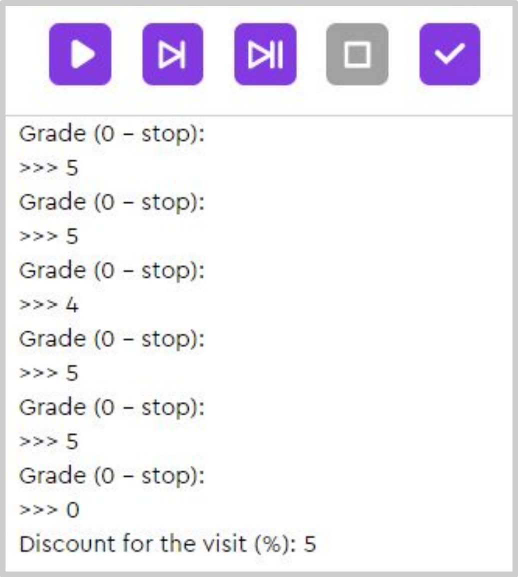
```
def get_data():
    Reading the data and performing the
    necessary calculations
    return parameter

def print_result(parameter):
    Analyzing the result and printing the output

result = get_data()

print_result(result)
```

The same code can be optimized in different ways.
It all depends on the developer’s skills and experience.

Brainstorm

Let’s have a look at a similar task

Task. The Center organizes an annual theater visit for its students. The students can get a discount on the price depending on the number of A’s they get on their final tests. Write a program that asks the user to input a student’s grades and then calculates the number of A’s. If the number of A’s is from 1 to 3, then the discount is 3%. If it is from 4 to 5, then the discount is 5%. If it is more than 5, then the discount is 10%.

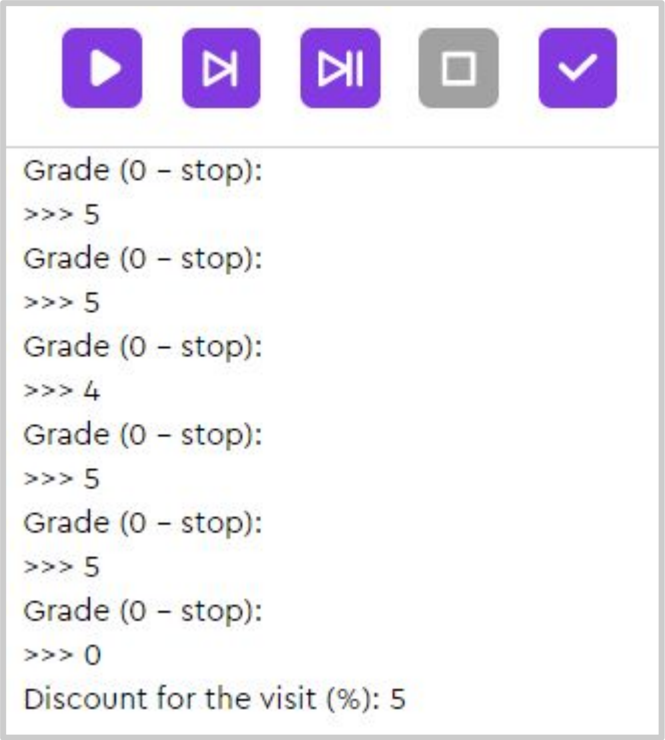


```
Grade (0 – stop):
>>> 5
Grade (0 – stop):
>>> 5
Grade (0 – stop):
>>> 4
Grade (0 – stop):
>>> 5
Grade (0 – stop):
>>> 5
Grade (0 – stop):
>>> 0
Discount for the visit (%): 5
```

What is the optimal solution for this task?
Counting A’s might be useful in other programs, too!

Brainstorm

```python
def amount_five():
    grade = int(input('Grade (0 - stop):'))
    amount_five = 0
    while grade != 0:
        if grade == 5:
            amount_five += 1
        grade = int(input('Grade (0 - stop):'))
    return amount_five

def set_discount():
    amount = amount_five()
    if amount >= 1 and amount <= 3:
        return 3
    elif amount >= 4 and amount <= 5:
        return 5
    elif amount > 5:
        return 10
    else:
        return 0

print('Discount for the visit (%):', set_discount())
```

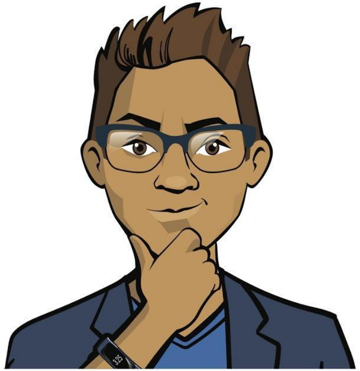


```
Grade (0 – stop):
>>> 5
Grade (0 – stop):
>>> 5
Grade (0 – stop):
>>> 4
Grade (0 – stop):
>>> 5
Grade (0 – stop):
>>> 5
Grade (0 – stop):
>>> 0
Discount for the visit (%): 5
```

Brainstorm

Before we continue:

1. **What will the program print** if we ask it to calculate the discount for students with the following grades: 3, 4, 5, 5, 4?
2. **Return** to the previous slide and **show** all the cases when one function is called from another (Python's built-in functions also count).
3. **Which of the functions** return values explicitly? Indicate them.



Brainstorm

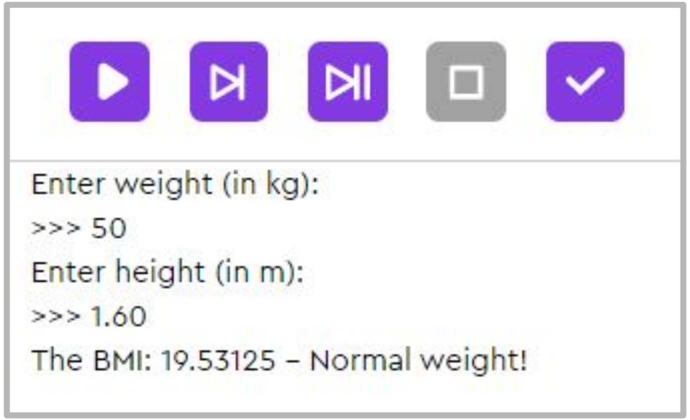
Let’s go over a task

Task. At Success, the Body Mass Index (CDC parameter description) is used to assess teachers' health risks. Write a program to calculate the BMI of an adult. The program should query for weight and height, read and print the BMI, and interpret the results, which organizations find quick and cheap to implement.

Recommendations. The calculation and printing parts must be written as standalone functions.

$$\text{BMI} = \frac{\text{WEIGHT}}{\text{HEIGHT} \times \text{HEIGHT}}$$

Less than or equal to 18.5	Underweight
18.5 to 25 inclusive	Normal weight
Greater than 25	Overweight



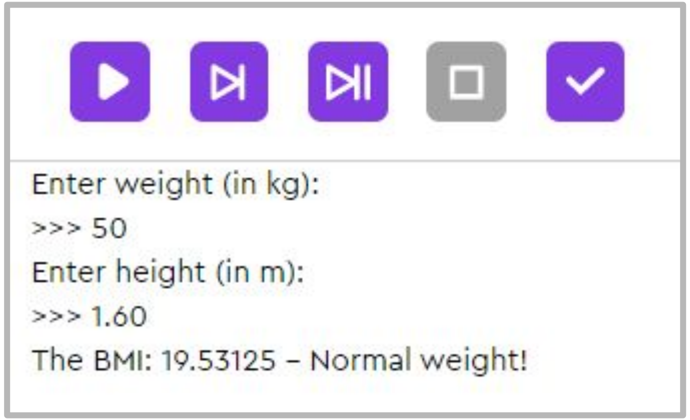
```
Enter weight (in kg):
>>> 50
Enter height (in m):
>>> 1.60
The BMI: 19.53125 – Normal weight!
```

What is the optimal solution for this task?

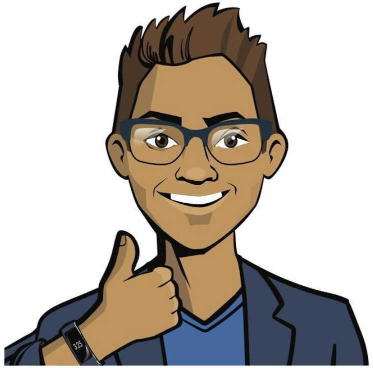
Brainstorm

Sample solution

```
def calc_bmi(weight, height): #weight in kg, height in m
    index = weight / (height * height)
    return index

def check_bmi(weight, height):
    index = calc_bmi(weight, height)
    if index <= 18.5:
        print('The BMI:', index, '- Underweight!')
    elif index > 18.5 and index <= 25:
        print('The BMI:', index, '- Normal weight!')
    else:
        print('The BMI:', index, '- Overweight')

weight = float(input('Enter weight (in kg):'))
height = float(input('Enter height (in m):'))
check_bmi(weight, height)
```



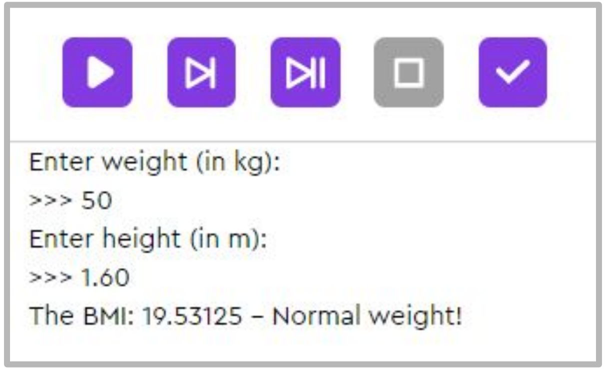
```
Enter weight (in kg):
>>> 50
Enter height (in m):
>>> 1.60
The BMI: 19.53125 – Normal weight!
```



Brainstorm

Sample solution

```
def calc_bmi(weight, height): #weight in kg, height in m
    index = weight / (height * height)
    return index


def check_bmi(weight, height):
    index = calc_bmi(weight, height)
    if index <= 18.5:
        print('The BMI:', index, '- Underweight!')
    elif index > 18.5 and index <= 25:
        print('The BMI:', index, '- Normal weight!')
    else:
        print('The BMI:', index, '- Overweight')


weight = float(input('Enter weight (in kg):'))
height = float(input('Enter height (in m):'))
check_bmi(weight, height)
```

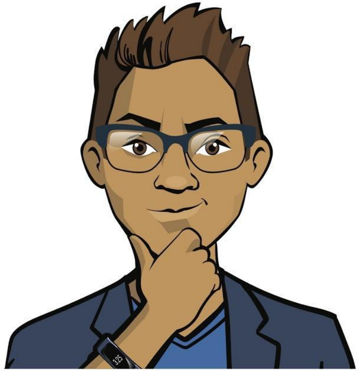


```
Enter weight (in kg):
>>> 50
Enter height (in m):
>>> 1.60
The BMI: 19.53125 – Normal weight!
```

Note that in this case, decimal fractions are explicitly converted (cast) into numeric values using the **float()** function.

Before we continue:

1. **What will the program print** if we enter the following data: 92 kg and 1.72 m?
2. The teachers have now asked us to **make the check_bmi() function more compact** and move the result printout to the main part of the program. How can we do this?



Brainstorm