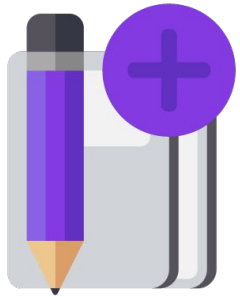
Brainstorm:

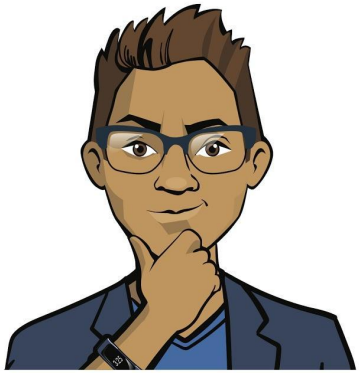
Creating functions



Working with functions

During your training, we already discussed the definition of a function. Do you remember it?

Give examples of some functions you already know.



Brainstorm

A function is

an algorithm that is composed in a programming language and has a unique name.

Some functions you already know:

Function	*Purpose*
`print()`	Printing the arguments given within the brackets
`input()`	Reading data
`int()`	Conversion to the integer type
`len()`	Determining the length of a string

Brainstorm

A function is

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

Formal definition and structure of a function:

```
value = function_name(argument1, argument2, ...)
```

Functions can explicitly return **the result of their operation** into the program.

The unique function **name**.

Data passed into the function as input.

Brainstorm

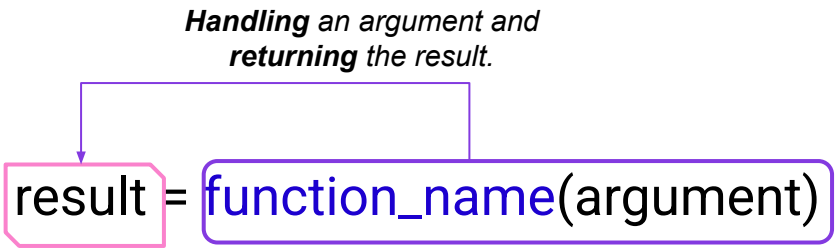
Working with functions

We have already discussed that functions can explicitly **return the result** of their operation — a **value**. For comparison:

This function does not explicitly return a value:

```
function_name(argument)
```

This function does return a value:



Brainstorm

Working with functions

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

```
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.

Brainstorm

Working with functions

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

In the first half of the working day, we are going to create functions that do not return a value, and in the second half, those that return a value.

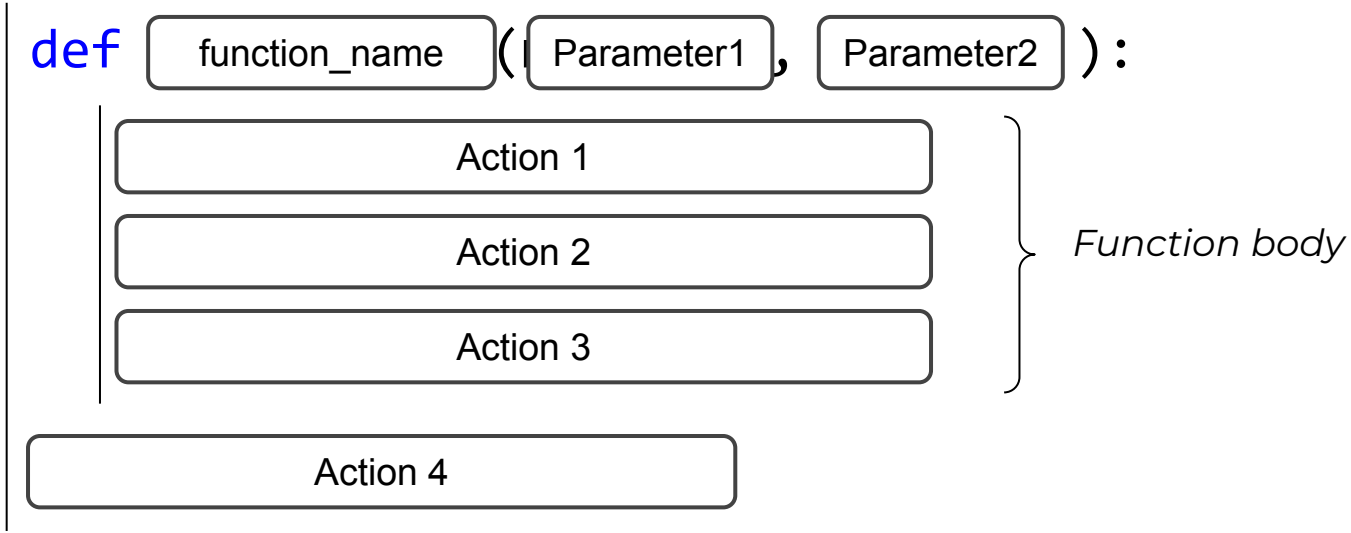
Brainstorm

Creating our own functions

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.



Brainstorm

Creating our own functions

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 these are hidden behind variables).

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

function_name(argument1, argument2)
```

Function declaration

Calling a function

Brainstorm

Let’s recall and solve the initial task

Let’s write a program that prints out Center-branded labels for the students' notebooks. These labels must contain the Center’s name and the “Name”, “Course”, and “Group” fields. The program should also ask for the number of labels to print, and print that number of labels as shown in the picture.

```
Label printing machine
Number of students:
>>> 2
THE SUCCESS CENTER
Name: ____
Course: ____
Group: ______
THE SUCCESS CENTER
Name: ____
Course: ____
Group: ______
Done! Take your labels.
```

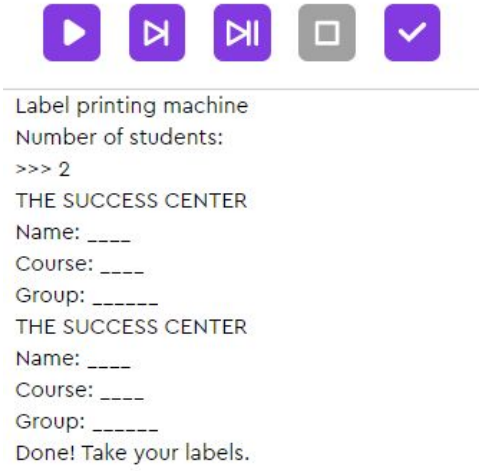
How do we “pack” label printing into a function?

Brainstorm

Let’s recall and solve the initial task

Write a program that prints out Center-branded labels for the students' notebooks. These labels must contain the Center’s name and the “Name”, “Course”, and “Group” fields. The program should also ask for the number of labels to print, and print that number of labels as shown in the picture.

```
def print_label():
        print('THE SUCCESS CENTER')
        print('Name: ____')
        print('Course: ____')
        print('Group: ______')


print('Label printing machine')
amount = int(input('Number of students:'))
for i in range (amount):
      print_label()
print('Done! Take your labels.')
```



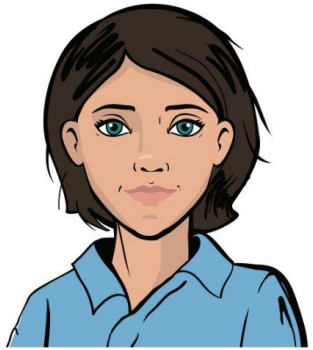
Brainstorm

Let’s recall and solve the initial task

Write a program that prints out Center-branded labels for the students' notebooks. These labels must contain the Center’s name and the “Name”, “Course”, and “Group” fields. The program should also ask for the number of labels to print, and print that number of labels as shown in the picture.

```
def print_label():
        print('THE SUCCESS CENTER')
        print('Name: ____')
        print('Course: ____')
        print('Group: ______')


print('Label printing machine')
amount = int(input('Number of students:'))
for i in range (amount):
      print_label()
print('Done! Take your labels.')
```

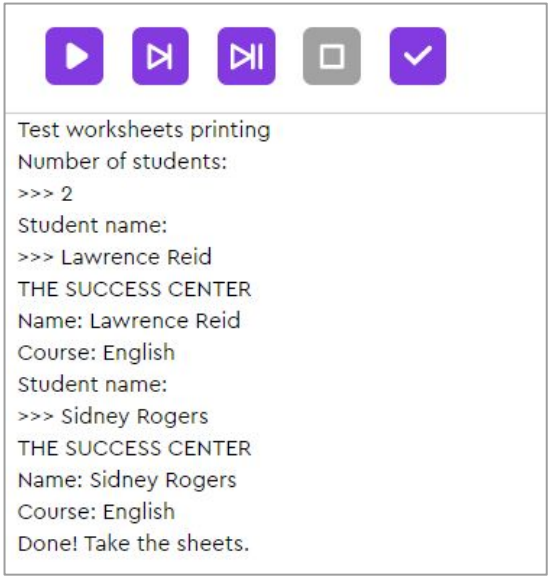
The `print_label()` function we have written does not take any arguments or return values.



Brainstorm

Let’s go over a task

Task. Write a program that prints out personalized labels for English lesson test sheets. It asks the user for the number of students and their names.



How do we solve this task using functions?

Brainstorm

Let’s go over a task

Task. Write a program that prints out personalized labels for English lesson test sheets. It asks the user for the number of students and their names.

A **solution** that changes print_label() — now it prints a personalized label.

```
def print_label(name):
    print('THE SUCCESS CENTER')
    print('Name:', name)
    print('Course: English')

print('Test worksheets printing')
amount = int(input('Number of students:'))
for i in range(amount):
    name = input('Student name:')
    print_label(name)
print('Done! Take the sheets.')
```

```
Test worksheets printing
Number of students:
>>> 2
Student name:
>>> Lawrence Reid
THE SUCCESS CENTER
Name: Lawrence Reid
Course: English
Student name:
>>> Sidney Rogers
THE SUCCESS CENTER
Name: Sidney Rogers
Course: English
Done! Take the sheets.
```

Brainstorm

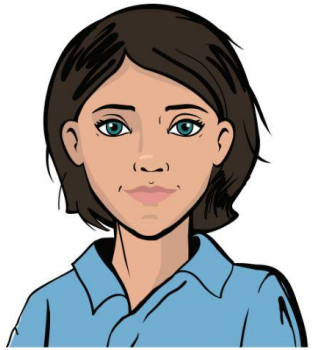
Let’s go over a task

Task. Write a program that prints out personalized labels for English lesson test sheets. It asks the user for the number of students and their names.

A **solution** that changes print_label() — now it prints a personalized label.

```
def print_label(name):
    print('THE SUCCESS CENTER')
    print('Name:', name)
    print('Course: English')

print('Test worksheets printing')
amount = int(input('Number of students:'))
for i in range(amount):
    name = input('Student name:')
    print_label(name)
print('Done! Take the sheets.')
```

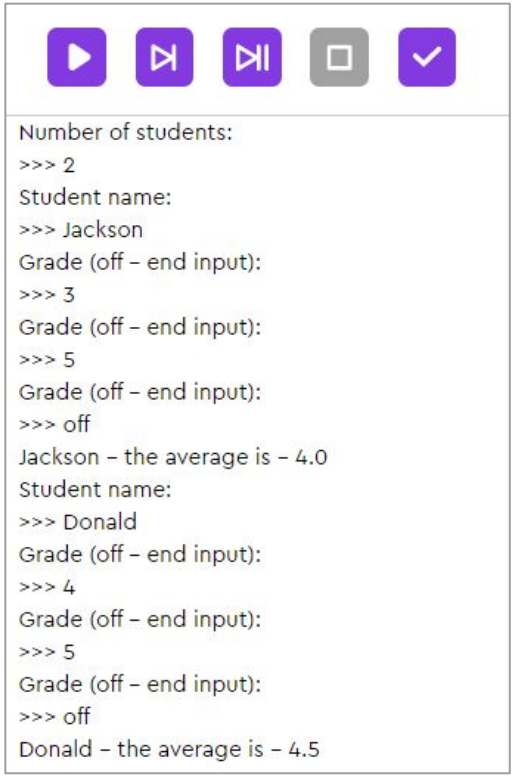
The renewed `print_label()` function takes
1 argument (name) and does not return any values.



Brainstorm

Let’s go over one more task

Task. Write a program calculating a student’s average. The program must ask for a student’s name and their grades and then calculate and print out their average in the following format: <Name> — the average is — <Grade>.



How do we solve this task using functions?
How about creating average_grade()?

Brainstorm

Let’s go over one more task

Task. Write a program calculating a student’s average. The program must ask for a student’s name and their grades and then calculate and print out their average in the following format: <Name> — the average is — <Grade>.

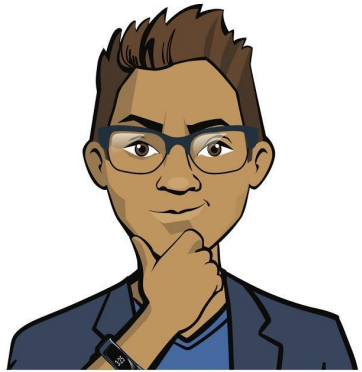
```
def average_grade(name):
    grade = input('Grade (off - end input):')
    summ = 0
    total = 0
    while grade != 'off':
        summ += int(grade)
        total += 1
        grade = input('Grade (off - end input):')
    print(name, '- the average is -', summ/total)

amount = int(input('Number of students:'))
for i in range(amount):
    name = input('Student name:')
    average_grade(name)
```

```
Number of students:
>>> 2
Student name:
>>> Jackson
Grade (off – end input):
>>> 3
Grade (off – end input):
>>> 5
Grade (off – end input):
>>> off
Jackson – the average is – 4.0
Student name:
>>> Donald
Grade (off – end input):
>>> 4
Grade (off – end input):
>>> 5
Grade (off – end input):
>>> off
Donald – the average is – 4.5
```

Brainstorm

Before we continue:

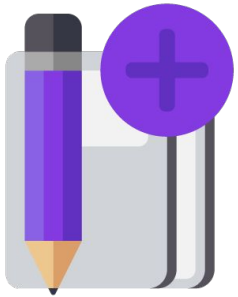
1. Does the print_grade() function **explicitly return** a value? If so, what is it?
2. **What will the program print** if we ask it for the average of a student called John with the following grades: 4, 3, 4, 5?
3. Return to the previous slide and show the print_grade() function's **arguments** and its **parameters**.



Brainstorm

Brainstorm:

Returning a value



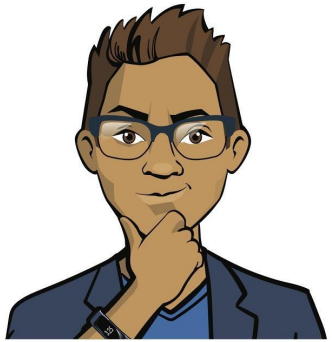
Let’s return to the previous task

The Center’s teachers are asking us to make it so that the average they get from the function can be used in the main part of the program. They may use it to check conditions for contest entry, for example.

What do we need to do to achieve this?

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


name = input('Student name:')
print_grade(name)
```

For now, let’s look at a calculation for a single student average.



Brainstorm

Let’s return to the previous task

The Center’s teachers are asking us to make it so that the average they get from the function can be used in the main part of the program. They may use it to check conditions for contest entry, for example.

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


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

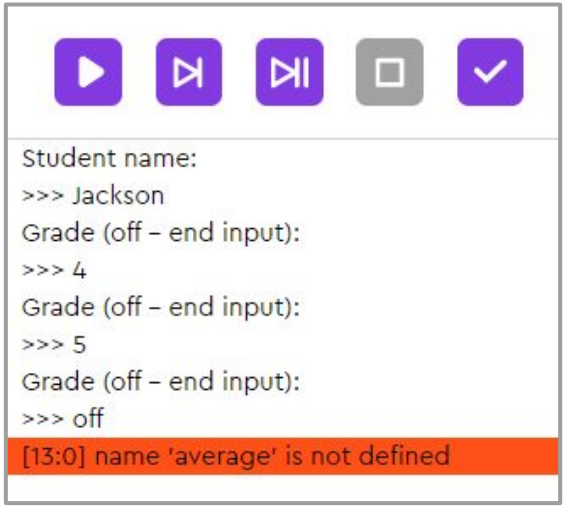
Emily suggested that we start by isolating and printing out the average. Will such a program work?

Brainstorm

Let’s return to the previous task

The Center’s teachers are asking us to make it so that the average they get from the function can be used in the main part of the program. They may use it to check conditions for contest entry, for example.

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


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



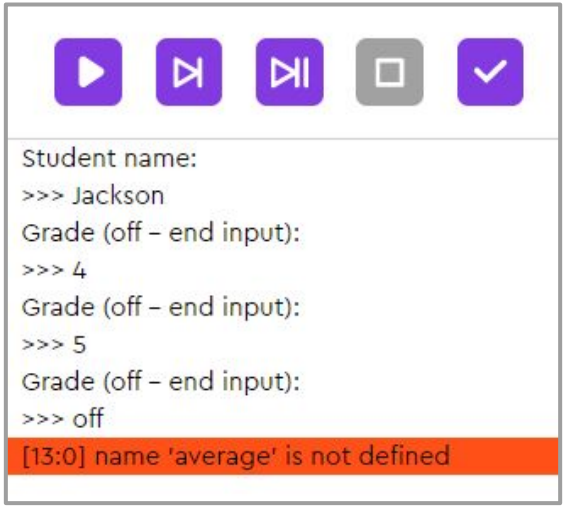
In this case, the interpreter will display an error!

Brainstorm

Local scope

The Python interpreter links variables to 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):')
    summ = 0
    total = 0
    while grade != 'off':
        summ += int(grade)
        total += 1
        grade = input('Grade (off - end input):')
    average = summ/total


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



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!

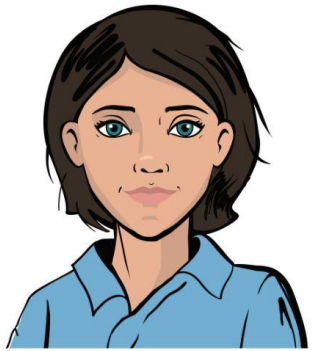
Brainstorm

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):')
    summ = 0
    total = 0
    while grade != 'off':
        summ += int(grade)
        total += 1
        grade = input('Grade (off - end input):')
    average = summ/total
    return average

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

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



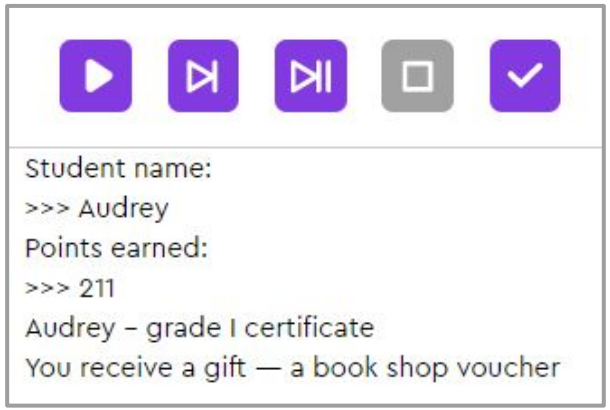
Brainstorm

Let’s go over a task

Task. The Center held a geography contest.

- The participants with 50 to 99 points are awarded grade III certificates.
- The participants with 100 to 199 points are awarded grade II certificates.
- The participants with 200 points or more are awarded grade I certificates.
- All the other participating students are awarded appreciation letters.

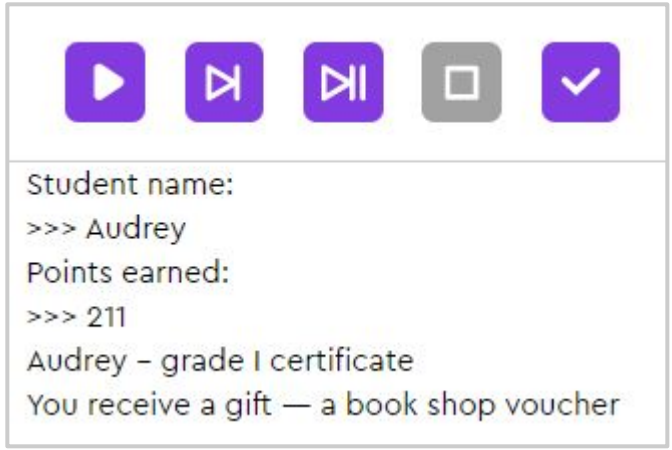
Write a function that asks for a student’s name and their points and returns the following string: Student <name> — <award>.



Brainstorm

Sample solution

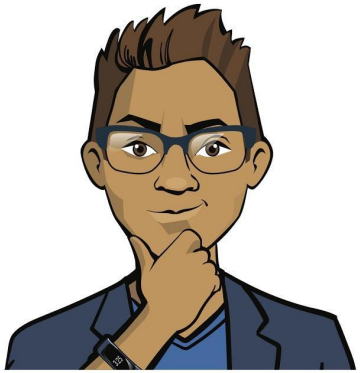
```
def get_result(score):
  if score >= 50 and score < 100:
      result = 'grade III certificate'
  elif score >= 100 and score < 200:
      result = 'grade II certificate'
  elif score >= 200:
      result = 'grade I certificate'
  else:
      result = 'appreciation letter'
  return result

name = input('Student name:')
score = int(input('Points earned:'))
result = get_result(score)
print(name, '-', result)
if result == 'grade I certificate':
   print('You receive a gift – a book shop voucher ')
```



Brainstorm

Before we continue:

1. **What will the program print** if we ask for the result of a student named Alice who has earned 190 points?
2. **In the program, show** the name of the function, its arguments, and the return value.
3. **What operator** allows us to get the result of a function's operation and transfer it to the main part of the program?



Brainstorm