Contents

1 Session 5

In this async session we will learn about functions and then we will have some exercises to do in order to put in practice all stuff we have learnt so far.

There is a forum open in Campus, if you feel stuck with anything and would like to get some help, please post there. I will check in several times during the day to answer questions that may pop. Please, also consider that I will prioritize questions in the forum before questions you send directly to my email.

Finally, please send over to my email your solutions to the exercises using a subject line like follows:

```
Solutions exercise 5 - <name_and_last_name>
```

Where **name_and_last_name** is your... well... name and last name :) If I had to send it i'd do it with a subject line like this:

```
Solutions exercise 5 - Pepe García
```

2 Functions

Functions are groups of Python statements that have a name and can be executed on demand. They will behave as black boxes to which we will be able to pass data in the form of **parameters** and that will give us output in the form of **return values**.

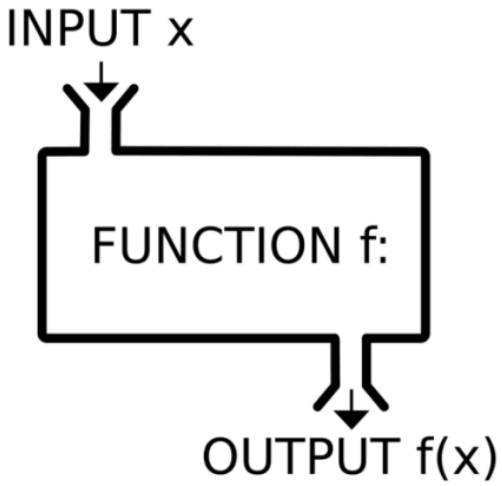

Figure 1: function

2.1 Calling functions

The syntax for calling functions is the following:

```
function_name(parameter1, parameter2, parameterN)
```

When naming functions we will need to apply the same naming rules as for variables.

We have already seen some functions, such as **print()**, **type()**, **str()**, etc. Notice that we have used all these functions by passing only one argument, but there are others to which we can pass more than one argument.

2.2 Declaring functions

We can declare our own functions using the **def** keyword with the following syntax:

```
def function_name(parameter1, parameter2):
    #<body>
```

When creating a function we need to indent the body to tell Python what piece of code we want to include inside the function, as we did with **if** statements.

2.3 Returning values from functions

Functions in Python can return values after doing all the operations they perform.

2.4 Function Parameters

Parameters are values that are injected to the function body when we call it

2.5 Exercises

2.5.1 exercise 1

Create a function `weekly_commute_time` that asks the user their daily commute time and returns their weekly time spent commuting.

2.5.2 exercise 2

What do the following expressions return?

- `True or 11 > 34`
- `False and (1 == 1)`
- `(77 // 11) > 6 and False`

2.5.3 exercise 3

Create a function `area_triangle` that takes the base and height of a triangle and returns its area

2.5.4 exercise 4

Create function `area_triangle_rectangle` that takes the base, height, and the kind of shape and calculates its area. It should work for both triangles and rectangles.

2.5.5 exercise 5

Create a function `im_in_love` that takes a weekday number (from monday to friday), and returns how that weekday is (according to The Cure!):

```
I don't care if Monday's blue
Tuesday's grey and Wednesday too
Thursday I don't care about you
It's Friday, I'm in love
```