Exercises Functions

Make the exercises on http://codingbat.com/python

- Create an account (use your Thomas More email address)
- Log in
- Click on prefs at the top right
- Fill in your last name and first name at "Memo" (in that order!)
- Fill in the email address of your Python teacher at "Share To" and click on "Share".
- Start working at the exercises. Make sure **you are always logged** in when practicing on codingbat!

For the following exercises, you **first** write **the mandatory function(s)** before programming the main program.

When an application requests information, the user will fill in the correct information. We therefore assume that the user does not enter incorrect information or unknown possibilities.

Exercise 1

Write a function that returns the number of degrees Fahrenheit (Tf) when the temperature in degrees Celsius (Tc) is given as a parameter. Use this conversion formula between Tc and Tf :

$$T_{\mathrm{F}} = T_{\mathrm{C}} \cdot \frac{9}{5} + 32$$

Then use this function in the program below.

```
Degrees Celsius: 38.2
38.2 degrees Celsius = 100.76 degrees Fahrenheit
```

Note that you did this exercise in chapter 1 but without using a function.

Exercise 2

Write a function to convert an amount in Euro into Dollar. The function has 2 parameters: the amount in Euro and the exchange rate of the day. The function returns the amount in Dollar.

Then use this function in the program below.

```
Current dollar rate (€ -> $): 1.2327
Your amount in Euro: 965
€ 965.0 = $ 1189.5555
```

Note that you did this exercise in chapter 1 but without using a function.

Exercise 3

For this exercise you first write the function **print_square**(). This function has 2 parameters: a number and a character.

- The number stands for the number of characters per line and the number of lines to be printed.
- The character determines the character to print.

For example: number= 5, character = '*': the function prints 5 lines with 5 asterisks per line.

```
*****
*****
*****
*****
*****
```

Write a program in which the character to be used and the desired number are entered first. Then use the function to press the square.

After the square has been printed, ask again which character you want to form lines with and the number of characters per line.

The program will keep running as long as the user does not press enter at the question 'Give the character with which the lines are formed (enter = stop):'. When the user presses enter, the question about the number of characters will of course no longer be asked.

```
Which character must be used to form the lines (enter = stop): $
Number of characters per line = number of lines = 3
$$$
$$$
$$$
Which character must be used to form the lines (enter = stop): %
Number of characters per line = number of lines = 2
%%
%%
Which character must be used to form the lines (enter = stop): :
Number of characters per line = number of lines = 4
::::
::::
::::
::::
Which character must be used to form the lines (enter = stop):
```

Exercise 4

a) Write a function **generate_list** that fills a list with a number of random integers between lower and upper limit. The number of generated numbers is given as a parameter as well as the lower and upper limit between which the generated numbers must fall. The generated list will be returned.

For example:

- **generate_list** (5,1,3) returns a list of 5 numbers between 1 to 3 (inclusive)
- **generate_list** (10,5,7) returns a list of 10 numbers from 5 to 7 (inclusive)

b) Write a function **filter** that receives a list as a parameter and filters the same list so that only unique elements in the list are returned.

For example: the list [1,2,3,3,4,5] is filtered into the following list of unique values [1, 2, 3, 4, 5].

c) Use both functions in the program below.

```
How many dice do you want to roll? 8
You threw:  [3, 5, 1, 4, 3, 5, 6, 1]
Your unique rolls were:  [3, 5, 1, 4, 6]
```

You can use the same functions to check, for example, how many times you have to roll 5 dice to have poker. Poker means that the 5 dice have exactly the same number of pips.

```
You threw:  [1, 1, 6, 2, 4]
You threw:  [6, 5, 6, 4, 5]
You threw:  [5, 1, 3, 3, 5]
You threw:  [6, 4, 4, 6, 6]
You threw:  [3, 3, 4, 2, 3]
You threw:  [2, 5, 5, 2, 3]
You threw:  [2, 5, 3, 1, 1]
You threw:  [3, 3, 3, 3, 3]
You threw poker after 651 times.
```

Exercise 5

Write a function that counts the number of upper and lower case letters in a string. The numbers will be returned as a list.

For example, when the function receives the string 'Life of Brian' as a parameter, the function returns [2, 9] where 2 is the number of uppercase letters and 9 is the number of lowercase letters.

Then use this function in the program below.

```
Your string: Life of Brian
Number of capitals: 2
Number of lowercase letters: 9
```

You can use the same function to check whether a new password meets the requirements. For example, suppose you set a password to contain at least 2 uppercase letters and at least 3 lowercase letters. Then write a program that continues to ask the user for a password until the input meets the requirements.

```
Set your password (at least 2 uppercase and 3 lowercase letters): myName
Set your password (at least 2 uppercase and 3 lowercase letters): MYNAMe
Set your password (at least 2 uppercase and 3 lowercase letters): MyName
```