



CodeInActionLab

Python-02

Summary: This document is the subject for the Python-02 module of the CodeInActionLab @ aba2020.

Version: 1.2

Contents

I	Instructions	2
II	Foreword	3
III	Exercise 00 : ft_ext	4
IV	Exercise 01 : ft_ext_pos	5
V	Exercise 02 : ft_palindrome	6
VI	Exercise 03 : ft_sum_all	7
VII	Exercise 04 : ft_ascending	9
VIII	Exercise 05 : ft_trim	10
IX	Exercise 06 : ft_tuple	11
X	Exercise 07 : bzero	12
XI	Exercise 08 : dictionary	13
XII	Exercise 09 : dictionary	14
XIII	Exercise 10 : alp_sort	15
XIV	Submission and peer-evaluation	16

Chapter I

Instructions

- Only this page will serve as reference: do not trust rumors.
- Watch out! This document could potentially change up before submission.
- Make sure you have the appropriate permissions on your files and directories.
- You have to follow the submission procedures for all your exercises.
- Your exercises will be checked and graded by your fellow classmates.
- On top of that, your exercises will be checked and graded by a program called Moulinette.
- Moulinette is very meticulous and strict in its evaluation of your work. It is entirely automated and there is no way to negotiate with it. So if you want to avoid bad surprises, be as thorough as possible.
- Moulinette is not very open-minded.
- These exercises are carefully laid out by order of difficulty - from the easiest to the hardest. We **will not** take into account a successfully completed harder exercise if an easier one is not perfectly functional.
- Using a forbidden function is considered cheating. Cheaters get -42, and this grade is non-negotiable.
- If your program doesn't compile, you'll get 0.
- You cannot leave any additional file in your directory than those specified in the subject.
- Got a question? Ask your peer on the right. Otherwise, try your peer on the left.
- Your reference guide is called `Google / man / the Internet /`
- Examine the examples thoroughly. They could very well call for details that are not explicitly mentioned in the subject...
- By Odin, by Thor ! Use your brain !!!

Chapter II

Foreword

Here is a discuss extract from the Silicon Valley serie:

- I mean, why not just use Vim over Emacs? (CHUCKLES)
- I do use Vim over Emac.
- Oh, God, help us! Okay, uh you know what? I just don't think this is going to work. I'm so sorry. Uh, I mean like, what, we're going to bring kids into this world with that over their heads? That's not really fair to them, don't you think?
- Kids? We haven't even slept together.
- And guess what, it's never going to happen now, because there is no way I'm going to be with someone who uses spaces over tabs.
- Richard! (PRESS SPACE BAR MANY TIMES)
- Wow. Okay. Goodbye.
- One tab saves you eight spaces! - (DOOR SLAMS) - (BANGING)

. . .


(RICHARD MOANS)

- Oh, my God! Richard, what happened?
- I just tried to go down the stairs eight steps at a time. I'm okay, though.
- See you around, Richard.
- Just making a point.

Hopefully, you are not forced to use emacs and your space bar to complete the following exercices.

Chapter III

Exercise 00 : ft_ext

	Exercise 00
Turn-in directory : <i>ex00/</i>	
Files to turn in : ft_ext.py	
Allowed functions : exit , len , print , range	

- Write a function that takes as argument a list of integers and returns the minimum and maximum values of the list.



Remember, you have to handle errors!

Here's how it should be prototyped:

```
def ft_ext(l):
```


```
42~ > python3 ft_ext.py 17 12 19 42 1337 1 5
Max value is 1337 and min value is 1
42~ >
```

```
42~ > python3 ft_ext.py 17
Error! You must enter at least two numbers
42~ >
```

```
42~ > python3 ft_ext.py 17 12 19 a 42 1337 1 5
Error! 'a' is not a number
42~ >
```

Chapter IV

Exercise 01 : ft_ext_pos

	Exercise 01
ft_ext_pos	
Turn-in directory : <i>ex01/</i>	
Files to turn in : ft_ext_pos.py	
Allowed functions : exit, len, print, range	

- Write a function that takes as argument a list of integers and returns the minimum and maximum values of the list.

Here's how it should be prototyped:

```
def ft_ext_pos(l):
```


```
42~ > python3 ft_ext_pos.py 17 12 19 42 1337 1 5
Max value is 1337 and it position is 5
Min value is 1 and it position is 6
42~ >
```

```
42~ > python3 ft_ext_pos.py 17
Error! You must enter at least two numbers
42~ >
```

```
42~ > python3 ft_ext_pos.py 17 12 19 a 42 1337 1 5
Error! 'a' is not a number
42~ >
```

Chapter V

Exercise 02 : ft_palindrome

	Exercise 02
ft_palindrome	
Turn-in directory : <i>ex02/</i>	
Files to turn in : <code>ft_palindrome.py</code>	
Allowed functions : <code>len</code> , <code>print</code>	

- Write a program that takes as argument a string and checks if the string is palindrome, returning either **True** or **False**. You have to implement a function that prints the result

Here's how it should be prototyped:

```
def ft_palindrome(l):
```

```
42~ > python3 ft_palindrome.py sus
"sus" is palindrome
42~ >
```


```
42~ > python3 ft_palindrome.py gigi
"gigi" is not palindrome
42~ >
```

```
42~ > python3 ft_palindrome.py Are we not drawn onward to new era
Error! Insert just a string!
42~ >
```

```
42~ > python3 ft_palindrome.py "Are we not drawn onward to new era"
"Are we not drawn onward to new era" is palindrome
42~ >
```

Chapter VI

Exercise 03 : ft_sum_all

	Exercise 03
	ft_sum_all
	Turn-in directory : <i>ex03/</i>
	Files to turn in : ft_sum_all.py
	Allowed functions : input, len, print

- Write a function that takes as argument a list of lists of integers and adds up the elements from all of the nested list. Insert 0 to interrupt the integer list, insert -1 at the start of the list to start the sum.
- Error management:
 - If the number is negative print "Error! Insert just positive numbers! and the sum message
 - If the input is not a number and not a minus, print "Error! Insert just numbers!"


Here's how it should be prototyped:

```
def ft_sum_all():
```

```
42~ > python3 ft_sum_all.py
Insert the 1 numbers:
> 17
> 12
> 0
Insert the 2 numbers:
> 42
> 21
> 0
Insert the 3 numbers:
> 26
> 0
Insert the 4 numbers:
> 19
> 0
Insert the 5 numbers:
> -1
The sum of all numbers is 137
42~ >
```

Chapter VII

Exercise 04 : ft_ascending

	Exercise 04
ft_ascending	
Turn-in directory : <i>ex04/</i>	
Files to turn in : ft_ascending.py	
Allowed functions : eval, input, len, print, range, sorted	

- Write a function that takes as argument a list of integers and returns **True** if the list is sorted in ascending order and **False** otherwise. If the result is False, write a function that prints the list in ascending order.
 - If arguments are **less than 2**, print "**Error! You must enter at least one number**"
 - If one of arguments is not a number, print "**Error! 'x' is not a number**"

Here's how it should be prototyped:


```
def ft_ascending(l):
```

```
42~ > python3 ft_ascending.py 12 17 19 21 42 1337
True
42~ >
```

```
42~ > python3 ft_ascending.py 12 1 17 14 19 21 33 42 1337 16
12 1 17 14 19 21 33 42 1337 16 is not in ascending order.
The right order is: 1 12 14 16 17 19 21 33 42 1337
42~ >
```

Chapter VIII

Exercise 05 : ft_trim

	Exercise 05
	ft_trim
	Turn-in directory : <i>ex05/</i>
	Files to turn in : ft_trim.py
	Allowed functions : eval, len, print, range

- Write a program that takes as argument a list. Modifies it by removing the first and last elements and returns nothing. The modification to the list must be visible from the caller.
 - If arguments are **less than 3**, print **"Error! You must enter at least two numbers"**
 - If one of arguments is not a number, print **"Error! 'x' is not a number"**


Here's how it should be prototyped:

```
def ft_trim(l):
```

```
42~ > python3 ft_trim.py 1 12 17 14 19 21 33 42 1337 16
The updated list is: 12 17 14 19 21 33 42 1337
42~ >
```

Chapter IX

Exercise 06 : ft_tuple

	Exercise 06
ft_tuple	
Turn-in directory : <i>ex06/</i>	
Files to turn in : ft_tuple.py	
Allowed functions : exit, len, print, range	

- Write a program that takes as argument an integer **N** and returns a tuple **T2** identical to **T1** except for the element of index **N** to which it must **change sign**. If the element of the index **N** **does not exist**, the function must **print T1**.
 - If arguments are less than 1, print **"Error! You must enter at least one number"**
 - If **N** is greater than T1 length, print **"Error! Out of bound exception!"**

Here's how it should be prototyped:


```
def ft_tuple(l):  
  
T1 = (17, 21, 42, 37, 5, 84)
```

```
42~ > python3 ft_tuple.py 3  
T2 = (17, 21, -42, 37, 5, 84)  
42~ >
```

```
42~ > python3 ft_tuple.py 8  
Error! Out of bound exception!  
T1 = (17, 21, 42, 37, 5, 84)  
42~ >
```

Chapter X

Exercise 07 : bzero

	Exercise 07
bzero	
Turn-in directory : <i>ex07/</i>	
Files to turn in : bzero.py	
Allowed functions : eval, len, print, range	

- Write a function that takes as argument a list of integer and an integer **N** which must be set to **0**. If the element of index **N** does not exist, the return must be equal to the list
 - If arguments are less than **2**, print **"Error! You must enter at least two numbers"**
 - If **N** is greater than list length, print **"Error! Out of bound exception!"**

Here's how it should be prototyped:

```
def bzero(l):
```


```
42~ > python3 bzero.py 3 14 21 38 42 57 26
Updated list: 14 21 0 42 57 26
42~ >
```

```
42~ > python3 bzero.py 10 14 21 38 42 57 26
Error! The list is: 14 21 38 42 57 26
42~ >
```

```
42~ > python3 bzero.py
Error! Usage: python3 bzero.py <N | List>
42~ >
```

Chapter XI

Exercise 08 : dictionary

	Exercise 08
dictionary	
Turn-in directory : <i>ex08/</i>	
Files to turn in : <code>dictionary.py</code>	
Allowed functions : <code>all</code>	

- Write a program that lets the user define a (small) "**English->Italian**" dictionary by inserting an English word followed by its italian translations. The dictionary must be saved in a dict data structure. The process of defining the dictionary end when the user inserts the string '**done!**'.


Here's how it should be prototyped:

```
def translate(n):
```

```
42~ > python3 dictionary.py
> Hello
Ciao
> School
Scuola
> Hand
Mano
> done!
42~ >
```

Chapter XII

Exercise 09 : dictionary

	Exercise 09
dictionary	
Turn-in directory : <i>ex09/</i>	
Files to turn in : dictionary.py	
Allowed functions : all	

- Write a program that uses the previous function to build an **"English->Italian"** dictionary, build the inverse **"Italian->English"** dictionary and lets the user insert words (in either English o Italian) printing the corresponding translation. If the inserted word is not in any of the two dictionaries, an error message must be printed. The function terminate when the user inserts **'done!'**.


Here's how it should be prototyped:

```
def translate(n):
```

```
42~ > python3 ft_tuple.py 7
> Hello
Ciao
> School
Scuola
> Mano
Hand
> Testa
Head
> done!
42~ >
```

Chapter XIII

Exercise 10 : alp_sort

	Exercise 10
alp_sort	
Turn-in directory : <i>ex10/</i>	
Files to turn in : <code>alp_sort.py</code>	
Allowed functions : <code>print</code> , <code>sorted</code>	

- Write a function that takes as argument a string and prints, in alphabetic order, the letters of the string with their number of occurrences. **The function should not be case sensitive.**

Here's how it should be prototyped:

```
def alp_sort(s):
```

```
42~ > python3 ft_palindrome.py My name is Francesco and i study at 42Roma
Error! Insert just a string!
42~ >
```

```
42~ > python3 alp_sort.py "My name is Francesco and i study at 42Roma"
42Roma
Francesco
My
and
at
i
is
name
study
42~ >
```


Chapter XIV

Submission and peer-evaluation

Turn in your assignment in your `Git` repository as usual. Only the work inside your repository will be evaluated during the defense. Don't hesitate to double check the names of your files to ensure they are correct.



You need to return only the files requested by the subject of this project.