

Coding in Action Lab I

Python-02

1 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 suggest you to solve them in the given order.
- Using a forbidden function is considered cheating. Cheaters get -42, and this grade is non-negotiable.
- If your program contains a syntax error, you'll get 0.
- You cannot leave any additional file in your directory other 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 Web /
- Examine the examples thoroughly. Your code must *exactly* reproduce the examples, for the specified test cases. Beware that the examples could very well call for details that are not explicitly mentioned in the problem statement.
- By Odin, by Thor ! Use your brain !!!

Exercise 00: ft_ext_pos

Create a program that takes as argument a list of integers and prints the minimum and maximum values of the list, together with their position.

Turn-in directory:	ex00/
Files to turn in:	ft_ext_pos.py
Allowed functions:	print, len, int

Examples:

```
42~ > python ft_ext_pos.py 2 3 1 5 -3 -1 4
The min is -3 and its position is 4
The max is 5 and its position is 3
42~ >
```

```
42~ > python ft_ext_pos.py
Error! Usage: python3 ft_ext_pos.py <x1> ... <xn>
42~ >
```

Exercise 01: ft_palindrome

Create a program that takes as argument a string and checks if the string is palindrome, making use of a function that returns `True` or `False`.

Turn-in directory:	ex01/
Files to turn in:	ft_palindrome.py
Allowed functions:	print, len

Examples:

```
42~ > python ft_palindrome.py kayak
The string kayak is palindrome
42~ >
```

```
42~ > python ft_palindrome.py exam
The string exam is not palindrome
42~ >
```

```
42~ > python ft_palindrome.py are we not drawn onward to new era
Error! Insert just 1 string!
42~ >
```

```
42~ > python ft_palindrome.py "are we not drawn onward to new era"
The string are we not drawn onward to new era is palindrome
42~ >
```

```
42~ > python -c 'from ft_palindrome import is_palindrome; print(is_palindrome("kayak"))'
True
42~ >
```

Exercise 02: ft_matrix

Create a program that takes as arguments two integers n and m , asks the user to insert the elements of a $n \times m$ real matrix A , prints A , and finally prints the sum of the elements of A over the rows and over the columns.

Turn-in directory:	ex02/
Files to turn in:	ft_matrix.py
Allowed functions:	print, range, int, float, input

Examples:

```
42~ > python ft_matrix.py 2 3
Insert the element in position (0, 0): 1
Insert the element in position (0, 1): 2
Insert the element in position (0, 2): 3
Insert the element in position (1, 0): 4
Insert the element in position (1, 1): 5
Insert the element in position (1, 2): 6
The inserted matrix is:
[1.0, 2.0, 3.0]
[4.0, 5.0, 6.0]
The sum over rows is:
[6.0, 15.0]
The sum over columns is:
[5.0, 7.0, 9.0]
42~ >

42~ > python ft_matrix.py 2
Error! Usage: python3 ft_matrix.py <n> <m>
42~ >
```

Exercise 03: ft_sorted

Create a program that takes as argument a sequence of integers and checks whether the sequence is sorted in descending order. If this is not the case, it must print the correctly sorted list.

Turn-in directory:	ex03/
Files to turn in:	ft_sorted.py
Allowed functions:	print, int, len, sorted

Examples:

```
42~ > python ft_sorted.py 5 4 2 1
The inserted sequence is sorted!
42~ >
```

```
42~ > python ft_sorted.py 5 2 4 0
The inserted sequence is not sorted!
The correct order is [5, 4, 2, 0]
42~ >
```

```
42~ > python ft_sorted.py 5
Error! You must insert at least 2 numbers
42~ >
```

Exercise 04: ft_trim

Create a program that takes as argument a list and prints it without the first and last element, by making use of a function `trim` that takes the list as a parameter and returns nothing.

Turn-in directory:	ex04/
Files to turn in:	ft_trim.py
Allowed functions:	print, len

Examples:

```
42~ > python ft_trim.py a b c d
The new list is: ['b', 'c']
42~ >
```

```
42~ > python ft_trim.py a b
The new list is: []
42~ >
```

```
42~ > python ft_trim.py a
Error! You must insert at least 2 values
42~ >
```

```
42~ > python -c 'from ft_trim import trim; print(trim([1,2,3,4]))'
None
42~ >
```

Exercise 05: ft_char_count

Create a program that takes as argument a string and prints, in alphabetical order, the characters of the string with their number of occurrences. The program must case insensitive.

Turn-in directory:	ex05/
Files to turn in:	ft_char_count.py
Allowed functions:	print, len, dict

Examples:

```
42~ > python ft_char_count.py programming
Char count:
a = 1
g = 2
i = 1
m = 2
n = 1
o = 1
p = 1
r = 2
42~ >

42~ > python ft_char_count.py "this is a test"
Char count:
  = 3
a = 1
e = 1
h = 1
i = 2
s = 3
t = 3
42~ >

42~ > python ft_char_count.py
Error! No string given
42~ >
```

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. Do not hesitate to double check the names of your files to ensure they are correct.