Coding in Action Lab I
Python-01

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_compare`

Create a program that takes two numeric arguments and prints their relation.

Turn-in directory:	`ex00/`
Files to turn in:	`ft_compare.py`
Allowed functions:	`print, float`

Examples:

```
42~ > python3 ft_compare.py 25 17
25.0 is greater than 17.0
42~ >

42~ > python3 ft_compare.py 25 25.0
25.0 is equal to 25.0
42~ >

42~ > python3 ft_str_print.py -25 -3.4
-25.0 is less than -3.4
42~ >
```

Exercise 01: `ft_abs`

Create a program that asks the user to insert a valid mathematical expression and prints the absolute value of the result.

Turn-in directory:	`ex01/`
Files to turn in:	`ft_abs.py`
Allowed functions:	`input, print, eval`

Example:

```
42~ > python3 ft_abs.py
Insert an expression: ((10*(-9))/15)-((17*12)/6)-2
The result is: 42.0
42~ >
```

Exercise 02: ft_max

Create a program that takes three numeric arguments and prints their maximum.

Turn-in directory:	ex02/
Files to turn in:	ft_max.py
Allowed functions:	print, len, float

Examples:

```
42~ > python3 ft_max.py 12 -3 7
The max is: 12.0
42~ >

42~ > python3 ft_max.py 17 12
Error! Usage: python3 ft_max.py <x> <y> <z>
42~ >
```

Exercise 03: `ft_min`

Create a program that takes three numeric arguments and prints their minimum, making use of a function `my_min` having three optional parameters.

Turn-in directory:	`ex03/`
Files to turn in:	`ft_min.py`
Allowed functions:	`print, len`

Examples:

```
42~ > python3 ft_min.py 12 3.5 2
The min is: 2.0
42~ >

42~ > python3 ft_min.py 12 3.5
Error! Usage: python3 ft_min.py <x> <y> <z>
42~ >

42~ > python3 -c ‘from ft_min import my_min; print(my_min())’
0
42~ >
```

Exercise 04: `ft_summorial`

Create a program that takes as argument an integer $n \geq 0$ and prints the sum of all positive integers less than or equal to n.

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

Examples:

```
42~ > python3 ft_summorial.py 5
The sum is: 15
42~ >

42~ > python3 ft_summorial.py
Error! Usage: python3 ft_summorial.py <n>
42~ >

42~ > python3 ft_summorial.py -2
Error! n must be >=0
42~ >
```

Exercise 05: ft_sum_list

Create a program that iteratively reads integers until the user enters 0, to finally print the sum of all inserted integers. The program must make use of a function `sum_list` whose unique parameter is a list and that returns the sum of all elements in the list.

Turn-in directory:	`ex05/`
Files to turn in:	`ft_sum_list.py`
Allowed functions:	`input, print, len, int`

Examples:

```
42~ > python ft_sum_list.py
Insert integer: 4
Insert integer: 2
Insert integer: 0
The sum is: 6
42~ >

42~ > python ft_sum_list.py
Insert integer: 0
The sum is: 0
42~ >

42~ > python -c ‘from ft_sum_list import sum_list; print(sum_list([-1,2,5]))’
6
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.