Coding in Action Lab I
Python-00

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_str_print`

Create a program that asks the user to insert a string and prints that string.

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

Example:

```
42~ > python3 ft_str_print.py
Insert your string: Hello
Your string is: Hello
42~ >
```

Exercise 01: `ft_sum`

Create a program that asks the user to insert two integers and prints their sum.

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

Example:

```
42~ > python3 ft_sum.py
Insert your first integer: 17
Insert your second integer: 25
Result: 42
42~ >
```

Exercise 02: `ft_sum_time`

Create a program that asks the user to insert a time of the day, in the form of a number of hours, minutes and seconds, and prints the total time in seconds.

Turn-in directory:	`ex02/`
Files to turn in:	`ft_sum_time.py`
Allowed functions:	`input, print`

Example:

```
42~ > python3 ft_sum_time.py
Insert hours: 1
Insert minutes: 35
Insert seconds: 4
Total seconds: 5704
42~ >
```

Exercise 03: `ft_print`

Create a program that asks the user to insert a string s and an integer n, and prints the characters at index n and $-n$ of s.

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

Examples:

```
42~ > python3 ft_print.py
Insert a string: My name is Marvin
Insert an integer: 5
m a
42~ >

42~ > python3 ft_print.py
Insert a string: I live in Rome
Insert an integer: 42
Error: index out of range
42~ >
```

Exercise 04: `ft_print (v2)`

Create a program that asks the user to insert a string s and an integer n, and prints the substring of s composed of all characters from index n to index $-n$, both included.

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

Examples:

```
42~ > python3 ft_print.py
Insert a string: My name is Marvin
Insert an integer: 5
me is Ma
42~ >

42~ > python3 ft_print.py
Insert a string: I live in Rome
Insert an integer: 42
Error: index out of range
42~ >
```

Exercise 05: `ft_space_print`

Create a program that asks the user to insert a string s and prints s so that the last character of s is on column 20.

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

Examples:

```
42~ > python3 ft_space_print.py
Insert a string: My name is Marvin
   My name is Marvin
42~ >

42~ > python3 ft_space_print.py
Insert a string: I live in wonderful Rome
ve in wonderful Rome
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.