

# Python warm-up

4TTV313U - Introduction to simulation of dynamic process

University of Bordeaux - 2022

## 1 Some small basic programs

### Loops and functions

1. Write a program that prints the square value of the 10 first integers and the sum and mean of both series (integers and their square values).
2. Write the factorial function `facto(n)` using a loop structure. [`facto(5)=120`]

### Python interactions

Write the secret number game program:

A player enters an integer number and the second player has to guess it. The program indicates if the player's value is too high or too low and if he finds it, how many trials were necessary.

## 2 About lists

In the last session, we built a list of squared values. There are different ways to do this.

```
list_1=list(range(0,9))

#Using a loop
list_2=[]
for l in list_1:
    list_2.append(l**2)

#Using map
list_3=list(map(lambda x: x**2,list_1))

#Using list comprehension syntax
list_4=[i**2 for i in list_1]
```

### Functions: map, filter, reduce

**map** applies a function on each element

```
list_m=list(map(lambda x: x/2,range(10)))
```

**filter** only returns elements verifying certain conditions

```
list_f=list(filter(lambda x: x % 2==0,range(10)))
```

## List comprehension

What do those lines do?

```
b = [x/2.0 for x in [i**2 for i in range(10)]]
c = [(x, x+10) for x in range(10) if x%2==0]
d = [x**y for x, y in c]
```

## Warning!

Try the following lines. What is going on? Why?

```
x = [0,10,20,1,5]
y = x
x [4]=30
print(y)
```

```
x = [0,10,20,1,5]
y = x[:]
x [4] = 30
print(y)
```

## Exercises

Give a list of all numbers below 1000 that are divisible by 3 (use `filter`) and double them (`map`). Do the same with a list comprehension syntax.

## 3 I/O for text files with pandas

To manipulate text files in CSV format, we will use the `pandas` library.

We will use the interactive Python 3 shell to inspect the file “`NAO_year.txt`”. This file contains the evolution of the North Atlantic Oscillation index.

First, type `ipython3` on the terminal. You can now execute the followings commands to inspect the file :

```
import pandas as pd
from matplotlib import pyplot as plt

# We open the file
data = pd.read_csv("NAO_year.txt")
data
data.keys()

# We plot the values. Do you think the result is correct ? Why ?
data.plot(y='Value')
plt.show()

# We sort the table by date
data = data.sort_values(by=['Year', 'Month'], ascending=True)
data = data.reset_index(drop=True)

# We save the sorted data in a new file.
data.to_csv('sorted_NAO_year.txt', index=False, encoding='utf-8')

# We plot the values of the sorted data.
data.plot(y='Value')
plt.show()
```

```
# We display the histogram of each column
data.hist()
plt.show()

# We display the correlation matrix between each columns
data.corr()

# We convert some columns into a list
val = data['Value'].to_list()
year = data['Year'].to_list()
# We convert the values into a numpy list (we will see numpy list in the next td)
numpy_list_of_values = data['Value'].to_numpy()
```

## Exercise

We will have a look at the evolution of the world population since 1950. There are different estimates for these data but not all of them are complete. For now, we will focus on the United States Census which provides data for every year since 1950.

Use the data from `World_population.csv` to write a new file which will contain only the year and results from the census, i.e. the values in the columns `Year` and `census` (filename: `World_population_simplified.csv`). The population of 2020 has been estimated around 7.693 billions. Add this information to your file.

This document was adapted by Adrien Boussicault from an initial work of Coralie Picoche (2020). This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International Licence.

