

DATA.ML.100 Introduction to Pattern Recognition and Machine Learning
TAU Computing Sciences
Exercise 0 *Setting up Python*

Be prepared for the exercise sessions (watch the demo lecture). You may ask TAs to help if you cannot make your program to work, but don't expect them to show you how to start from the scratch.

1. **Python environment setup** (10 points)

- (a) Install Python to your computer.

The most common way for ML practitioners is to install Python via the *Anaconda* tool. Python libraries develop fast and therefore conflicts between the libraries often occur. Anaconda allows you to define different library combinations for different projects and therefore help to avoid conflicts.

Go to <https://www.anaconda.com> and look for the correct Anaconda installer for your OS (Linux, Windows, iOS). If you're a IT/CS/EE student we recommend you to learn Linux and thus the following instructions are for Linux users. For other OSs you need to google how things work in your system.

By default Anaconda is started automatically in login, but I want to switch that off by the following spell in terminal:

```
$ conda config --set auto_activate_base false
```

For my DATA.ML.100 course projects I make own Anaconda environment:

```
$ conda create --name dataml100  
$ conda activate dataml100
```

Now inside the course Anaconda environment I install the main Python packages and libraries

```
(dataml100)$ conda install python # Install Python
```

- (b) Install *integrated development environment* (IDE)

You write and run your programs using IDE. There are multiple choices available and the following are the most popular within our students: 1) PyCharm, 2) VSCode, 3) Spyder. Install your favourite IDE and use it to write programs.

- (c) Write sorting algorithm

Download *mysort_skeleton.py* code from the course page and write your sorting algorithm.

```
import sys  
  
# Read command line arguments and convert to a list of integers  
arr = sys.argv[1].split(',')  
my_numbers = [None]*len(arr)  
for idx, arr_val in enumerate(arr):  
    my_numbers[idx] = int(arr_val)  
  
# Print  
print(f'Before_sorting_{my_numbers}')  
# My sorting (e.g. bubble sort)  
# ADD HERE YOUR CODE  
  
# Print  
print(f'After_sorting_{my_numbers}')
```

The output should be (code run on terminal):

```
(dataml100) $ python mysort.py 5,5,2,10,11,0,-4  
Before sorting [5, 5, 2, 10, 11, 0, -4]  
After sorting [-4, 0, 2, 5, 5, 10, 11]
```