[COM4513-6513] Lab 0: Python Introduction

Instructor: Nikos Aletras

The goal of this lab session (**not assessed**) is to make you familiar with Python 3 (**NOT Python 2**, see this for a quick overview) and some libraries that will use throughout the course.

It is strongly recommended to use Ubuntu Linux for the labs (how to login to Ubuntu). On Ubuntu, use `/apps/anaconda/bin/python` to run python from the Anaconda distribution. You should also make sure that when using your own machine and operating system (e.g. MacOS, Linux, Windows) any Python package versions should be identical to the ones we use on the University PCs (check the Anaconda version) to avoid any issues in executing your code by the markers (that could result into losing marks). You could also work on Windows machines by using python from the Anaconda prompt but it is not recommended.

For developing in Python, you could use standard IDEs like PyCharm. You will also find useful IPython and Jupyter notebooks for quick prototyping.

Here is the plan:

- Go through this tutorial and this one ("1. Getting started with Python for science" section) on numpy, scipy and matplotlib libriaries, as you will need them for your assignments.
- Have a look at the spacy.io library. It offers functionality that will be useful for your assignments such as part-of-speech tagging, syntactic parsing and named entity recognition.
- Familiarize yourselves with PyTorch for NLP following this tutorial. Note that PyTorch is only available on Ubuntu, and you have to use it for at least one of the assignments.
- Version Control Systems help keep software engineers sane and science reproducible by keeping track of the code we write to implement our ideas. A good place to start is to go through this tutorial on Git and GitHub. If you decide to use Github for maintining your code for the labs, you should make sure that your repos are private.