Exercise 0-1: Python Tech Stack

The objective is to prepare a convenient Python development environment, needed for some part of our laboratory practice.

Development Environment

Our development environment consists of following components:

Anaconda

We use **Anaconda** as a main development environment.
Anaconda includes

- high-performance distribution of **Python**, as well as interactive **iPython** interpreter
- over 100 of the most popular Python, R and Scala **packages** developed for projects in data science

Anaconda comes with a suite of graphical interface tools called **Anaconda Navigator.**
Anaconda Navigator starts from the window of Anaconda application launcher.

Conda

Conda is the

- Anaconda environment **management system**
- installs and manages the packages installed in Anaconda
- also takes care of the environment management and the **dependencies** for different languages, available in Anaconda - Python, R, Ruby, Lua, Scala, Java, JavaScript, C/ C++, FORTRAN

Conda is itself an open source package enabled to

- find and install the necessary external packages for a specific type of a project
- can create **virtual environments** for each individual project

Conda runs in a terminal mode control by **CLI**.
You can download or read about Conda at https://conda.io/docs/index.html.

Jupyter

Jupyter is a helper application

- special kind of editor
- browser based
- interactive

Jupyter files are called **notebooks.** A notebook can contain both *live code* and *document text* in the same file.
Jupyter includes Python code interpreter with immediate result.

Python

Python is a powerful interpreted programming language

- open source from https://www.python.org/
- with easy to learn and elegant syntax
- runs on most known platforms

It combines

- dynamic typing
- efficient high-level data structures

- effective approach to object-oriented programming

Python programs are rapidly developed scripts.
Python interpreter is distributed with an extensive **standard library** of classes for various implementations.

There also exist huge number of free
- third party Python modules
- program examples
- tools
- additional documentation

Python interpreter can be extended with new functions and data types implemented in C or C++.

Python Popular Libraries

`Pip` - an installer and packaging system for Python

`IPython` - interactive Python

`NumPy` - numerical Python

`Scipy` - math and scientific computing

`Pandas` - high-performance data analysis

`Scikit-learn` - a popular and powerful machine learning library

`Scrappy` - web crawling framework

`NLTK` - natural language toolkit

`Pattern` - a web mining library

`OpenCV` - a computer vision library

`Matplotlib` - visualization library

`Seaborn` - statistical visualization

Install Anaconda

Download Anaconda https://www.anaconda.com/
- select your OS
- alternatively, take Miniconda, if you do not have enough space

Install Anaconda following the wizard.

Test the Installation

Open your **Terminal**

Type

```
python
```

it should show you Python version and metadata

Type

```
import numpy
```

If nothing happens, it is a sign that the package is already imported by Anaconda.

To exit python type

```
exit()
```

Try the installation with some other packages from the list above.

Test conda

Type

```
conda –V
```

to check if and which version is there

Type

```
conda --help
```

to see which commands it uses

Type

```
conda list
```

to see which packages it manages

Test Jupyter

Type

```
jupyter notebook
```

It opens in your default browser Alternatively, it runs from Anaconda Navigator
Jupyter starts the Jupyter server

Update the Installation

Type

```
conda update conda
```

Type

```
conda update anaconda
```

Try the internal Python package manager pip Type

```
pip list
```

See what pip also knows about the installed packages

To update specific package, e.g. scikit-learn library, type

```
conda update scikit-learn
```

In addition to the tools provided by Anaconda's common data science environment, we will use some specialized frameworks, which we will install later.