% Fundamentals of programming in Biomedicine
% Dr. Enrique M. Muro (muro at uni-mainz.de)
% March 2022

Fundamentals of programming in biomedical science

Bioinformatics and Computational Biology and hence computers and programming are part of the knowledge (and CV) of modern Biomedicine.

If programming in Biology started as a side-tool in crystallography (evolution, sequencing, etc), it ended up with ubiquity all over Biology. Independently of your area of expertise, basic programming skills are needed to access, process and analyze any kind of Biological data: clinical, from pharmaceutics, virology, genetics, structures, protein interactions, diseases and so on.
The fundamentals of programming in Biomedicine will be introduced with Python 3. The lectures will consist in explanations and continuous programming: an active engaging from the students is expected during the classes. No previous programming experience is expected.

For the examination grades (1/8 of the total module), the students will be evaluated during the day of teaching and the grades provided with the whole module grades. Again, it is expected active learning and commitment during the day: online interaction with the class, online presentations of solutions to problems, interactions, discussions, etc. Please check that you have a computer, a proper internet connection and browser where you can assist the classes and code; also, microphone and camera (working properly). If you do not have any of these, please contact the module responsible with urgency.

Your computer needs some simple software in advance. Please, set it up during the week. In the case you have problems send me an **email before the 18.2.2022**. Also, write me some lines, in the case you have some programming experience, letting me know about it.

Note that **the day of the teaching we can not solve any computer configuration problem**, everything needs to be prepared in advance.

For more reading on the relevance of programming in Biology see:

> - *All Biology Is Computational Biology. Markowetz, F. (2017). PLOS Biol. 15, e2002050. doi:10.1371/journal.pbio.2002050*

> - *The digital toolbox. (2014). Nature. Sep 4;513(7516):6. doi: 10.1038/513006b*

Computer set up:

Usually Python 3 is already installed by default in many computers. So, check it.

1. Test that "Python 3" is installed

To test if Python 3 is installed in your computer, simply run from the terminal (or Windows Console):

```
python3 --version
```

```
Python 3.8.10
```

3.8.10 in my case. It is possible that you have only Python 2 (for instance 2.7.18) installed in your computer; this will be of no use. You will need to have Python 3 installed.

Next (if Python 3 is installed), open and interactive session. Run in the terminal (or Windows Console):

```
python3
```

You will see the next (Python 3 interactive session) adapted to your Operative System:

```
Python 3.8.10 (default, Nov 26 2021, 20:14:08)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Now, you can write and interactively execute Python code. Finally, make a quickly check and exetute:
print("Hello World")

```
Python 3.8.10 (default, Nov 26 2021, 20:14:08)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello World")
Hello World
>>>
```

If you see the **Hello World** message, Python 3 is installed: just skip the next section (2. Install "Python 3"). Otherwise, follow it.

2. Install "Python 3"

We will use only Python 3 (not Python 2). Then, install Python 3 from https://www.python.org:

https://www.python.org/downloads/

Download the last *"stable release"* for your Operative System (3.*.*):

- Windows

```
https://www.python.org/downloads/windows/
```

- MacOs

```
https://www.python.org/downloads/macos/
```

- Linux: Python 3 comes preinstalled on most Linux distributions. Let us know in advance if you have any difficulty to set it up.

2.1. Test that "Python 3" is already installed

Follow **1. Test that "Python 3" is installed**

3. Install a text editor

Our programs need be written in *"plain text"* not in *"rich text"* and if you have any kind of orthographic corrector needs to be deactivated. Therefore, it is not possible to write programs with "MS Word" and you just need to have a simple plain text editor. Whatever you use, check that you can manage it before the day of the class; check that you can open files from any directory, modify them, save, etc. Simple editors already set up are:

- Windows (notepad).

- Linux (gedit. it has already color highlighting for python code). Otherwise install one for your distribution. Let me know if you have any difficulty.

- MacOs (TextEdit): If you use TextEdit, be sure that "**Plain Text**" is selected from the Format menu and **no orthographic corrector** is activated. In preferences->Menu you do not need to have any option selected. See for instance:

```
https://www.youtube.com/watch?v=L_PQ7PMByGk
```

The previous editors will do the job, but it is very helpful to have color highlighting for our code. Standard editors are:

- Windows (Notepad++): One of the most famous and popular editors. Download it from:

```
https://notepad-plus-plus.org/
```

- MacOs (Atom): Dowload it from (https://atom.io/).

Again, whatever text editor you want to use for programming, just be sure that you
can create new files in any directory, open, modify, save them, and so on.

4. Create and check your first program.

4.1 Create a working folder where you are going to write all the code related to these lectures:

programmingBiomedicineYourName
For instance, if you are Peter Schmidt: programmingBiomedicinePschmidt

4.2 With the "text editor" (just installed)

Open a new file an save it in the programmingBiomedicineYourName directory with the name: "hello_word.py".
The file extension (.py) indicates the editor that you are editing a python program and it will highlight your python code. Write print("Hello "World") and save.

Finally, in the terminal or Windows Console (**Note***) and in the programmingBiomedicineYourName folder, run the next:

```
python3 hello_world.py
```

and you should see the "Hello World" message. Then, you are ready for the classes.

*Note: Windows Console, in the case you never used it, please, proceed with the next steps:

- watch a Windows Console Tutorial (only the first)

  https://www.youtube.com/watch?v=MBBWVgE0ewk

- Using **cd**, go to the directory (folder) where you created your hello_world.py file (programmingBiomedicinePschmidt, for Peter Schmidt)

- and execute the next:

```
python3 hello_world.py
```