Week 1 — *Python starter: Hello world*

The goal of the present exercises is to discover a simple "Hello" program.

Exercise 1: *Installing Python environment*

- Installation is made by typing the command:
  ```
  sudo apt-get install python3
  ```

- Installation of Spyder python IDE is made with the command:
  ```
  sudo apt-get install spyder3
  ```

Exercise 2: *Exploiting the main arguments*

- Get the sources by updating your cloned copy of the SP4E repository
  ```
  git pull upstream master
  ```

- Open the file `hello.py` (Using Spyder, or your preferred editor)

- Launch the program, by clicking play or in a console with:
  ```
  python3 hello.py
  ```

- Comment each line of the source file.

- What is the nature (type) of the `sys.argv` variable (from `sys` module)?

- Modify the main function so that the message printed to screen should be "Hello N" with N being a parameter passed when launching the program.
  Spyder users: you have to go in the menu "Run⇒Configure" to specify the parameters to be passed to the program.

Exercise 3: *First loop*

- Modify the program so that the program first computes the series:

$$S_n = \sum_{k=1}^{n} k \tag{1}$$

  where n should be taken as an argument and the result should be printed aside of the "Hello"

- How many operations are necessary to perform this series computation?

- Considering the analytic prediction what is the overhead?

Exercise 4: *Modules*

- Split the obtained program in two files:
  1. main.py
  2. series.py

The file `series.py` should contain a function `computeSeries`

```
def computeSeries(Niterations):
```

The file `main.py` should contain the main function and the call to the series module.

Exercise 5: *Debugging*

- Linux Users: launch pdb on your program by

```
pdb hello.py arg1 arg2 arg3
```

- Spyder Users: open and launch the debugging stage using "Spyder". The commands provided below are to be typed in the IPython console tab.

- Set a breakpoint to the main function

```
(pdb) break main
```

- Run the program and pass the correct arguments Linux users:

```
(pdb) run
```

Spyder users: In the debug menu find the correct button

- Step over each instruction of your main function until the end of the program.

- Console users:

```
(pdb) next
...
(pdb) next
```

Spyder users: click the step button.

- re-run the program, but this time enter the function you created using the `step` command.

- Advance into the loop and print the content of your counter

```
(pdb) print i
```

- make a conditional breakpoint for when the counter is equal to 10

```
(pdb) break line, i == 10
(pdb) continue
```

where `line` should be the line number inside the loop.

- Change the value of the counter (during the execution of the program) back to zero

```
(pdb) i = 0
(pdb) print i
```

Exercise 6: *Pushing your modifications to your personal GIT repository*

Push your work to your own repository. See previous exercises if you don't remember.