Introduction to Programming Homework 1

Due Wednesday September 28 by 14h00

You will turn in your homework via e-mail [andrew.yarmola@uni.lu](). For this homework, you will work in IPython. I will provide the instructions for sending me your work down below.

- Download and install Anaconda with Python 3.5 from [continuum.io/downloads]().
- Download the SciPy Lecture Notes PDF found at [scipy-lectures.org]().
- After installation, open Terminal (Linux, macOS) or Command Prompt (Windows) and launch IPython by typing `ipython` at the prompt and then pressing ENTER.
    - Note : at the **end** of your work you will submit your command history using the command `%history -g -f homework_1.py`.
- If you have problems or questions, don't hesitate to email me.

Exercise 1

- a. Create a variable called `desc` with the string value `'The sum of 2 plus 2 is equal to'`.
- b. Check the type of `desc`.
- c. Call the command `print(desc, 2+2)`. Notice that you are giving the `print` function two input variables. In the context of programming, input variables are usually called *arguments*.
- d. Create a variable called `print_arg_exp` where you explain in words what `print` would do if you gave it three arguments instead of two. For example, a *wrong* answer would be : `print_arg_exp = 'print would crash if given three arguments'`.

Reading

Read Sections 1.1 - 2.2.3 inclusive in the SciPy Lecture Notes. Feel free to follow along with their examples and experiment! There are a few things here we did not talk about in lecture, such as strings, dictionaries, and tuples. We will start with these topics next class in detail.

Exercise 2

Write a one line command to output the list of values modulo 373 of the numbers 1 to 999 inclusive. Hint : use list comprehension.

Exercise 3

Start by creating the lists :

```
days = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
months = ['January', 'February', 'March', 'April', 'May',
'June','July', 'August', 'September', 'October', 'November', 'December'
]
```

Write a one line command to output the list of pairs (lists in this case)

```
[['January', 31], ['February', 28], ['March', 31],
 ['April', 30], ['May', 31], ['June', 30], ['July', 31],
 ['August', 31], ['September', 30], ['October', 31],
 ['November', 30], ['December', 31]]
```

We will learn how to properly *fold* (or intertwine) lists together later.

Exercise 4

Use the `sum()` to approximate π using the Bailey–Borwein–Plouffe formula

$$\pi = \sum_{k=0}^{\infty} \left[\frac{1}{16^k} \left(\frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6} \right) \right]$$

Use as many summands as you feel appropriate.

Exercise 5

When using list comprehension, you can also filter a list using the `if` keyword. The `if` keyword must be followed by an expression that returns a Boolean value. For example

```
[ x for x in range(1, 100) if x > 50 and x % 3 == 1 ]
```

For `x in range(1, 100)`, x is added to the list if `x > 50 and x % 3 == 1` is `True`. This should remind you of normal set notation

$$\{x \mid x \in \{1, 2, \ldots, 99\}, \ x > 5, \text{ and } x = 1 \bmod 3\}$$

- a. Find all *even* numbers between 1000 and 3000 inclusive which are divisible by 7 but are not a multiples of 3.
- b. Use python to verify that

$$\{x \mid x \in \{1, 2, \ldots, 10000\}, \text{ and } x^2 = 2 \bmod 5\} = \emptyset$$

Submitting your work

- At the IPython prompt, type

  ```
  %history -g -f homework_1.py
  ```

  and hit ENTER. This will record your command history to a file called `homework_1.py`. The characters `-g` and `-f` are called flags. The flag `-g` specifies that we want *all* IPython history and the flag `-f` specifies that the next string is a the name of the file into which the command history will be saved.
- Unless you are familiar with terminal, you may not know exactly *where* IPython saved the file `homework_1.py`. To look this up, type

  ```
  %pwd
  ```

  and hit ENTER. This will *print (your) working directory* (pwd for short). You can find your `homework_1.py` file here.
- If you would like, you can use a *basic* text editor such as TextEdit on a Mac or Notepad on Windows to edit your `homework_1.py` file. You can also use `spyder` (included with Anaconda) or get the popular text editor Atom (atom.io). Once you are done editing, email to (andrew.yarmola@uni.lu).