Leiden University
Medical Center

Python Programming

Introduction

Mihai Lefter

Outline

About the course

- Aimed at PhD students, Postdocs, researchers, analysts, ...
- Focus on:
  - Basic understanding of Python.
  - Programming as a tool to do your research.
  - Slightly biased on bioinformatics.

Hands on!

Programming is fun!

- You only learn programming by doing it.
- Lecture format:
  - Blended teaching + exercising.
- Have your laptop open during the lessons.
- Repeat the code from the slides, play around with it.
- Do the session exercises.
- There will be a few assignments.

Teachers

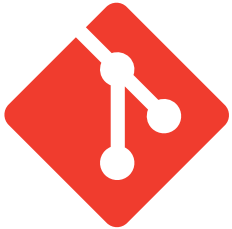
- Sander Bollen
  a.h.b.bollen@lumc.nl
- Jonathan Vis
  j.k.vis@lumc.nl
- Mark Santcroos
  m.a.santcroos@lumc.nl
- Guy Allard
  w.g.allard@lumc.nl
- Mihai Lefter
  m.lefter@lumc.nl

Program

	Tuesday 27/11	Wednesday 28/11	Thursday 29/11	Friday 30/11
9:00 - 10:00	Welcome, Introduction to Python	Assignments review	Assignments review	Assignments review
10:00 - 11:00	Data types	String methods, errors, and exceptions	Object-oriented programming	Data visualisation with Matplotlib
11:00 - 12:00	Flow control	Standard library, reading, and writing files	Jupyter Notebook	Data visualisation with Bokeh
12:00 - 13:00	Lunch break		Data mangling with pandas	Biopython
13:00 - 14:00	Practical session		Lunch break	
14:00 - 17:00				

Software requirements

- Anaconda:
  - Python 3.7.
  - Comes with all that's required:
    - Python interpreter.
    - Jupyter Notebook.
    - Libraries: NumPy, Panda, matplotlib, Bokeh, Biopython, ...
  - Installation instructions.
- Git:
  - Installation instructions.



Assignments

- We make use of GitHub Classroom.
  - GitHub account required.
  - Receive link with assignment repository.
- Own forked repository to work on:
  - Clone it.
  - Code it.
  - Push it.
- Direct file upload to repository is also possible.

Getting help

- Ask a teacher (we will be around in the afternoon).
- If it's private, mail one of the teachers.

History

- Created early 90's by Guido van Rossem at CWI.
  - Name: Monty Python.
- General purpose, high-level programming language.
- Design is driven by code readability.

CWI Centrum Wiskunde & Informatica

Features

- Interpreted, no separate compilation step needed.
- Imperative and object-oriented programming.
  - And some functional programming.
- Dynamic type system.
- Automatic memory management.

We'll come back to most of this.

Why Python?

- Readable and low barrier to entry.
- Rich scientific libraries.
- Many other libraries available.
- Widely used with a large community.

Python 2 versus Python 3

- Python 2.7 is the last Python 2.
- Python 3 is backwards incompatible.
- Some libraries don't support it yet.
- Some Python 3 features are backported in Python 2.7.
- Python 2.7 will not be maintained past 2020.

We'll use Python 3.7 for this course.

Running Python code

Two main ways of writing and executing Python code:

- Interactively:
  - Statement by statement, directly in the interpreter.
- Non-interactively:
  - By editing in a file and running the code afterwards.

The standard Python interpreter

Start it by typing **python** on the command line:

terminal

```
$ python
Python 3.7 (default, Nov 12 2018, 13:43:14)
[GCC 5.4.0 20160609] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

- It shows an interpreter prompt.
- You can give it Python code to interpret.

The IPython interpreter

Similar to the standard Python interpreter, but with:

- syntax highlighting;
- tab completion;
- cross-session history;
- etc.

Start it by typing **ipython** on the command line:

terminal

```
$ ipython
Python 3.7 (default, Nov 12 2018, 13:43:14)
Type 'copyright', 'credits' or 'license' for more information
IPython 6.1.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]:
```

Integers

IPython

```
In [1]:  17
Out[1]:  17

In [2]:  (17 + 4) * 2
Out[2]:  42

In [3]:  2 ** 3
Out[3]:  8
```

Floating point numbers

IPython

```
In [4]: 3.2 * 18 - 2.1
Out[4]: 55.5

In [5]: 36 / 5
Out[5]: 7.2
```

Floating point numbers

Scientific notation:

IPython

```
In [6]:  1.3e20 + 2
Out[6]:  1.3e+20

In [7]:  1.3 * 10**20
Out[7]:  1.3e+20
```

Variables

- We can use names to reference values (variables).
- No need to declare them first or define the type.

IPython

```
In [8]:  a = 1.3e20
In [9]:  b = 2
In [10]: a
Out[10]: 1.3e20

In [11]: c = a + 1.5e19 * b
In [12]: c * 2
Out[12]: 3.2e+20
```

Every value has a type

- View it using `type`.

IPython

```
In [13]: type(27)
Out[13]: int

In [14]: type(3 * 2)
Out[14]: int

In [15]: type(3 / 2)
Out[15]: float

In [16]: type(a)
Out[16]: float
```

Some operations are defined on more than one type

- Possibly with different meanings.

IPython

```
In [17]: type(3 * 2.0)
Out[17]: float

In [18]: drinks = 'beer' * 5 + 'whiskey'
In [19]: drinks
Out[19]: 'beerbeerbeerbeerbeerwhiskey'

In [20]: type(drinks)
Out[20]: str
```

Dynamic typing

- At runtime variables can be assigned values of different types.

IPython

```
In [21]: a
Out[21]: 1.3e+20

In [22]: type(a)
Out[22]: float

In [23]: a = 'spezi'
In [24]: type(a)
Out[24]: str
```

Strongly typed

- Operations on values with incompatible types are forbidden.

IPython

```
In [25]: 'beer' + 34

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-17-ec918fbfdf41> in <module>()
----> 1 'beer' + 34

TypeError: Can't convert 'int' object to str implicitly
```

Hands on!

1. We've seen that `b = 2` is legal.
   a. What about `2 = b`?
   b. How about `a = b = 1`?
2. In math notation you can multiply x and y like this: xy. What happens if you try that in Python?
3. How many seconds are there in 42 minutes and 42 seconds?
4. How many miles are there in 16 kilometers? (1 mile = 1.61 km)
5. Let's assume that you run a 42 km race in 4 hours 42 minutes and 42 seconds.
   a. What is your average pace (time per mile in minutes and seconds)?
   b. What is your average speed in miles per hour?
6. Use string operations to reference `'tra la la la'` in a variable named `song`.
7. If an article costs 249 Euros including the 19% Value Added Tax (VAT), what is the actual VAT amount in Euros for the corresponding article?

Leiden University Medical Center

Acknowledgements

Martijn Vermaat
Jeroen Laros
Jonathan Vis