

Engineering programming

Python basics

Thibault Thétier - thibault.thetier@vub.be

Tom Godden - tgodden@vub.be

1ste semester 2021

WHAT IS THE COURSE ABOUT?

Manipulating, processing, cleaning and crunching data in Python

- ▶ Introduction to Python
- ▶ NumPy & Scipy: scientific computing
- ▶ Matplotlib: graphs & data visualization
- ▶ Pandas: high-performance data manipulation

WHY PYTHON?

- ▶ Good for fast prototyping : Reduce programming time to test or implement something quickly
- ▶ Python has little syntax overhead, designed to be easy to learn and use
- ▶ Interpreted language, means that there is no compilation. Code can be executed as soon as it is written
- ▶ Makes mathematical computations easy : NumPy, SciPy, Matplotlib, etc.

DISADVANTAGES

- ▶ Low Speed
- ▶ Inefficient memory consumption
- ▶ Prone to cause runtime error

ANACONDA, SPYDER AND JUPYTER

- ▶ Anaconda:
 - ▶ distribution for scientific computing: data science, machine learning, large-scale data processing, etc.
 - ▶ comes with over 250 packages automatically installed
- ▶ Spyder
 - ▶ Integrated Development Environment (IDE) focused on scientific work and data science
- ▶ Jupyter
 - ▶ Create and share documents that contain live code, equations, visualizations and text

HELLO WORLD!

Hello world in C#:

```
1 static void Main(string[] args)
2 {
3     Console.WriteLine("Hello World!");
4 }
```

Hello world in Python:

```
1 print("Hello world!")
```

What you can see:

- ▶ You do not need a main function to write your code in
- ▶ No end of line character

VARIABLES

Python is dynamically typed: you don't declare the type of the variables.

Type	Example
Numbers	128, 3.14, 4+5j
Strings	'Thibault', "Tom's"
Lists	[1, "string", 2.45]
Tuples	(1, "string", 2.45)

Strings, Lists and Tuples are sequences.

It is also good to initialise variables (with 0 or without a value).

PLAYING WITH THE CONSOLE

Try the following commands in the console

▶ `1+2*3`

▶ `(1+2)*3`

▶ `2**4`

▶ `-7/2`

▶ `7//2`

▶ `-7.3//2.1`

▶ `12%5`

▶ `9.1%2`

▶ `1 < 2`

▶ `3 == 5`

▶ `3 != 5`

▶ `0 and 1`

▶ `0 or 1`

▶ `a = 3`

▶ `type(a)`

▶ `a = 7/3`

▶ `type(a)`

▶ `type(a) == int`

▶ `type(a) is int`

▶ `type(a) is not int`

STRINGS

Strings in python are surrounded by either single quotation marks, or double quotation marks.

Try:

```
▶ S = ""  
▶ S = 'Hello'  
▶ S[1]  
▶ S[2]="w"  
▶ "o" in S  
▶ len(S)  
▶ S + "World"
```

```
▶ S[:2]  
▶ print(S)  
▶ print("the year is:  
    "+str(35))  
▶ print("the year is:  
    {}".format(2020))
```

You can display a string literal with the print() function.

LISTS

Collection of any objects, **ordered** and **indexed**.

```
L = []
```

```
L = [10, 20, 30, 40, 50, 60]
```

```
L = list(range(4))
```

Try:

▶ `L[1]`

▶ `L[2]=23`

▶ `L.append(123)`

▶ `L.insert(1,-10)`

▶ `30 in L`

▶ `del L[4]`

▶ `L.remove(10)`

▶ `len(L)`

▶ `L.sort()`

▶ `L[:2]`