



# Programming 1

Laszlo SZATHMARY

University of Debrecen

Faculty of Informatics

## Lab #1

- introduction
- string data type

(last update: 2022-02-06 [yyyy-mm-dd])

2021-2022, 2nd semester



# About the course

**Title of the Course:** Programming 1

**Course ID:** INBGA0212 (for Business Informatics BSc students)

**Classes/week:** 2 + 0 + 2

**Credits:** 6

**Status:** Obligatory

**Assessment:** Practical mark

**Pre-requisite:** INBGA0105 (Introduction to Programming)

**Home Page of the Instructor:**

<https://arato.inf.unideb.hu/szathmary.laszlo/pmwiki/index.php?n=En.En>

**Time and classroom of the lecture:** *Monday* 14h-16h, IK-201

# Requirements

- Attendance (max. 3 missed labs)
- Achieving at least 60% of the total possible points. You can collect points with 3 mid-term tests (2 practical + 1 theoretical). Each test has 33.3% weight.
- Solving at least 50% of the homeworks.

Practical tests: on a computer. You get some exercises that you need to solve (write programs).

Theoretical test: in eLearning. Quiz-like test.

**Grade scale:**

% of points	Practical mark
[0 - 59]%	1
[60 - 69]%	2
[70 - 79]%	3
[80 - 89]%	4
[90 - 100]%	5

fail

pass



# Requirements

If you fail, there will be an extra mid-term test (last chance), where you can try to improve your mark. However, in this case the best mark can be maximum 3.

This extra test will be a practical test on computer.

# Bibliography

- *Guido van Rossum: Python Tutorial*  
(<https://docs.python.org/3/download.html>, also in PDF), 2022
- *Wesley J. Chun: Core Python Programming* (2nd Edition), 2006
- *Allen B. Downey: Think Python* (How to Think Like a Computer Scientist)  
<http://www.greenteapress.com/thinkpython/>, O'Reilly, 2012
- *Doug Hellmann: The Python Standard Library by Example*  
(Developer's Library), 2011  
[online version: **Python Module of the Week** (<https://pymotw.com/3/>)]

# Bibliography (cont.)

## Python 3

- *Mark Pilgrim: Dive Into Python 3* (<http://www.diveintopython3.net/>), 2009
- *Michael Driscoll: Python 101*, Leanpub, 2014 (beginner)
- *Michael Driscoll: Python 201*, Leanpub, 2016 (intermediate)

## Expert

- *Luciano Ramalho: Fluent Python*, O'Reilly, 2015

# Introduction



- Python is a general purpose, very high level programming language.
- Primary design goal: readability.
- Interpreted language, the code can be executed immediately.
- Multiparadigm (imperative, object-oriented, functional).
- The first version was released in 1991 and it was named after the Monty Python group.
- It was designed by Guido van Rossum, a Dutch researcher/programmer (born in 1956). 2005-2012: Google; 2013-2019: Dropbox. At the end of 2019 he retired, but at the end of 2020 he came back and joined Microsoft.
- What languages influenced Python: ABC, ALGOL 68, C, C++, Dylan, Haskell, Icon, Java, Lisp, Modula-3, Perl.
- What languages were influenced by Python: Boo, Cobra, D, Falcon, Groovy, JavaScript, Ruby, Go.

# Introduction



**Guido van Rossum**   
@gvanrossum



I decided that retirement was boring and have joined the Developer Division at Microsoft. To do what? Too many options to say! But it'll make using Python better for sure (and not just on Windows :-). There's lots of open source here. Watch this space.

6:00 PM · Nov 12, 2020 · Twitter Web App

**5.3K** Retweets   **2.2K** Quote Tweets   **38.4K** Likes



# Introduction

- Dynamic types and automatic garbage collection.
- Platform independent (Unix/Linux, Windows, Mac OS, etc.)
- Python has a large standard library („batteries included”), and there are thousands of freely available 3rd party modules<sup>1</sup>.
- The interpreter and the standard library are open source.
- Easy to learn, fun to use. It has a simple syntax. The source code can be read easily.
- It has efficient high-level data structures that help the work of programmers. Object-orientation is done simply and effectively.

<sup>1</sup> see <https://pypi.org> (on February 6, 2022 there were **355,098** packages available ; on January 14, 2021 there were **283,496** packages available )

# Introduction

- Ideal language for scripts and for rapid application development (RAD).
- Supports rapid prototyping.
- Similar programming languages: Perl, Ruby.
- A perfect choice for small scripts (ex. 10-20 lines), but it's also suitable for large projects with several thousands of lines of code! It has modules and packages, which allow us to keep the source code clear.
- For a long time, there were two branches that existed in parallel: Python 2 and Python 3 (Python 3 was forked in Dec. 2008). Python 2.7 is not maintained since January 1, 2020. If you start a new project, use Python 3.
- Here, in the labs we will use Python 3. The latest version is Python 3.10 . Since Python 2 (2.7) is still widely used, the key differences will be mentioned. Recommended version: Python 3.8+

# Links

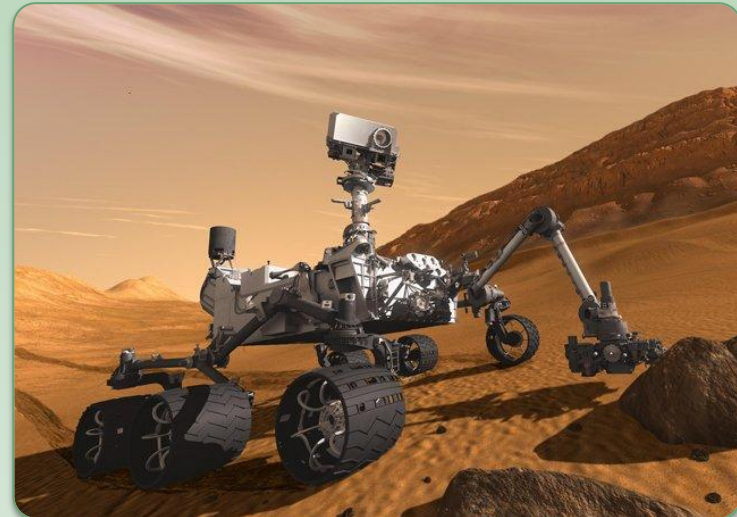
- Python HQ: <https://www.python.org/>
- Python documentation: <https://docs.python.org/>
- The Python Standard Library: <https://docs.python.org/library/>
- Python FAQ: <https://docs.python.org/faq/general.html>
- PEP 8 -- Style Guide for Python Code:  
<https://www.python.org/dev/peps/pep-0008/>
- <https://www.reddit.com/r/learnpython>
- <https://www.reddit.com/r/python>
- <https://stackoverflow.com/questions/tagged/python>

# Where is it used?

- Python success stories: <https://www.python.org/about/success/>

- **Scientific**

- [Biology](#)
- [Bioinformatics](#)
- [Computational Chemistry](#)
- [Data Visualization](#)
- [Drug Discovery](#)
- [GIS and Mapping](#)
- [Scientific Programming](#)
- [Simulation](#)
- [Weather](#)



Mars Curiosity (August 6, 2012)

Software: 2.5 million C lines.

Log files were tested with Python scripts.





















- Google (C, C++, Java, Python, Go, Dart, etc.)  
„Python where we can,  
C++ where we must”  
([link](#))

# How popular?



TIOBE index (<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>)



Jan 2022	Jan 2021	Change	Programming Language	Ratings	Change
1	3	▲	 Python	13.58%	+1.86%
2	1	▼	 C	12.44%	-4.94%
3	2	▼	 Java	10.66%	-1.30%
4	4		 C++	8.29%	+0.73%
5	5		 C#	5.68%	+1.73%
6	6		 Visual Basic	4.74%	+0.90%
7	7		 JavaScript	2.09%	-0.11%
8	11	▲	 Assembly language	1.85%	+0.21%
9	12	▲	 SQL	1.80%	+0.19%
10	13	▲	 Swift	1.41%	-0.02%
11	8	▼	 PHP	1.40%	-0.60%
12	9	▼	 R	1.25%	-0.65%
13	14	▲	 Go	1.04%	-0.37%
14	19	▲▲	 Delphi/Object Pascal	0.99%	+0.20%
15	20	▲▲	 Classic Visual Basic	0.98%	+0.19%
16	16		 MATLAB	0.96%	-0.19%
17	10	▼▼	 Groovy	0.94%	-0.90%
18	15	▼	 Ruby	0.88%	-0.43%
19	30	▲▲	 Fortran	0.77%	+0.31%
20	17	▼	 Perl	0.71%	-0.31%

# Literature

amazon Deliver to Hungary

All python programming

All Today's Deals Customer Service Registry Gift Cards Sell

1-48 of over 5,000 results for "python programming"

**Kindle Unlimited**  
☐ Kindle Unlimited Eligible

**Department**  
 Books  
 Programming Languages  
 Computer Programming  
 Computer Science  
 Databases & Big Data  
 Web Development & Design  
 Programming  
 Reference  
 Kindle Store  
 Computers & Technology

**Customer Reviews**  
 ★★★★★ & Up  
 ★★★★★ & Up  
 ★★★★★ & Up  
 ★★★★★ & Up

**Book Series**  
☐ Learning Python

**Deals & Discounts**  
 All Discounts  
 Today's Deals

**Prime Reading**  
☐ Prime Reading Eligible

**New Releases**  
 Last 30 days  
 Last 90 days  
 Coming Soon

**RESULTS**

**Best Seller**

**PYTHON CRASH COURSE**  
 2ND EDITION  
 A HANDS-ON, PROJECT-BASED INTRODUCTION TO PROGRAMMING  
 by Eric Matthes  
 OVER 1,000,000 COPIES SOLD

Python Crash Course, 2nd Edition: A Hands-On, Project-Based Introduction to Programming  
 by Eric Matthes  
 ★★★★★ ~ 5,782  
**Paperback**  
 \$21.49 ~~\$39.95~~  
 Ships to Hungary  
 More Buying Choices  
 \$16.45 (76 used & new offers)  
 Ages: 12 years and up  
 Other formats: [Kindle](#), [Spiral-bound](#)

**Best Seller**

**Learning Python**  
 Powerful Object-Oriented Programming  
 5th Edition  
 by Mark Lutz  
 ★★★★★ ~ 1,453  
**Paperback**  
 Other format: [Kindle](#)

(last checked on February 6, 2022)



# Conferences

PyCon US is the largest Python conference (<https://us.pycon.org/>).

## PyCon US 2021

- presentations : <https://pyvideo.org/events/pycon-us-2021.html> (85 videos)
- [https://www.youtube.com/playlist?list=PL2Uw4\\_HvXqvYk1Y5P8kryoyd83L\\_0Uk5K](https://www.youtube.com/playlist?list=PL2Uw4_HvXqvYk1Y5P8kryoyd83L_0Uk5K)
- there are several *tutorials* among the presentations

## PyCon US 2020

- presentations : <https://pyvideo.org/events/pycon-us-2020.html> (90 videos)
- <https://www.youtube.com/c/PyCon2020/videos>
- there are several *tutorials* among the presentations

<https://pyvideo.org/> collects the videos of Python conferences.

# Quiz

What's the name of the creator of Python?

- ☐ Larry Wall
- ☐ Yukihiro Matsumoto
- ☐ Guido van Rossum
- ☐ Rasmus Lerdorf

**Homework:** Look after the others. Who are they?



## Using the interpreter:

```
[22:28:56] ~ $ python3
Python 3.9.6 (default, Jun 30 2021, 10:22:16)
[GCC 11.1.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> █
```

## Writing a script:

```
1 #!/usr/bin/env python3
2
3 print("Hello, World!")
```

In Python 2, if you use accented characters, then you **must** add this line.

In Python 3, under Linux, this encoding is the default, thus this line can be omitted.

## Using special characters:

```
1 #!/usr/bin/env python3
2 # coding: utf-8
3
4 def main():
5     # using special characters
6     print("Jyväskylä")
```

```
4 >>> a = 6
5 >>> a
6 6
7 >>> a = "hello"
8 >>> len(a)
9 5
10 >>> a
11 'hello'
12 >>> A
13 Traceback (most recent call last):
14   File "<stdin>", line 1, in <module>
15 NameError: name 'A' is not defined
16 >>> "hello " + "world"
17 'hello world'
18 >>> "hello " + 6
19 Traceback (most recent call last):
20   File "<stdin>", line 1, in <module>
21 TypeError: cannot concatenate 'str' and 'int' objects
22 >>> "hello " + str(6)
23 'hello 6'
```

no need to declare variables

def

```
1 #!/usr/bin/env python3
2
3
4 def main():
5     print("Hello, World!")
6
7
8 main()
```

colon

indenting

no semicolon

**Style:** leave 2 empty lines before and after a function.

```
1 #!/usr/bin/env python3
2
3
4 def main():
5     print("Hello, World!")
6
7
8 if __name__ == "__main__":
9     main()
```

} Executed directly  
or  
called as a module?

Print the command-line arguments:

From now on, we will omit  
the first line:

```
3 import sys
4
5
6 def main():
7     print(sys.argv)
8
9
10 if __name__ == "__main__":
11     main()
```

`#!/usr/bin/env python3`

**Then:** provide a name as an argument (ex. `./hello.py Bob`),  
and greet the person („Hello Bob!”).

```
4 import sys
5
6 def hello(name):
7     if name == "Batman" or name == "Robin":
8         print("Batman or Robin")
9     else:
10        print(NoSuchFunction())
11
12 def main():
13     hello(sys.argv[1])
14
15 if __name__ == "__main__":
16     main()
```

no parenthesis after the if

We only get an error  
if code execution  
gets here!

One more reason to do unit tests for larger programs.  
Every branch must be tested!

## A general template for Python 3 scripts

```
1 #!/usr/bin/env python3
2 # encoding: utf-8
3
4
5 def main():
6     print('Py3')
7
8 #####
9
10 if __name__ == "__main__":
11     main()
```

**Tip:** save this file under the name `basic.py`, then if you want to write a new Python script, just make a copy of this file.

You can find this template here: <http://bit.ly/307H2PS>

```
>>> print "hello"
File "<input>", line 1
    print "hello"
          ^
SyntaxError: Missing parentheses in call to 'print'
>>>
>>> print("hello")
hello
>>>
>>> 7 / 2
3.5
>>>
>>> 7 // 2
3
>>>
```

most important  
changes  
in Python 3



```
>>> print "hello"
hello
>>>
>>> 7 / 2
3
>>>
>>> 7 // 2
3
>>>
```

# Strings

```
4 >>> s = "Hello"
5 >>> s
6 'Hello'
7 >>> s = 'Hello'
8 >>> s
9 'Hello'
10 >>> s = "isn't"
11 >>> s
12 "isn't"
13 >>> s = 'he said: "go home"'
14 >>> s
15 'he said: "go home"'
16 >>> s = "he said: \"go home\""
17 >>> s
18 'he said: "go home"'
19 >>> s = 'batman'
20 >>> len(s)
21 6
22 >>> s[0]
23 'b'
24 >>> s[0] = 'B'
25 Traceback (most recent call last):
26   File "<stdin>", line 1, in <module>
27   TypeError: 'str' object does not support item assignment
28 >>> s
29 'batman'
30 >>> s + '!'
31 'batman!'
32 >>> s = 'Joker'
33 >>> s.lower()
34 'joker'
35 >>> s.upper()
36 'JOKER'
37 >>> s.find('k')
38 2
39 >>> s.find('a')
40 -1
41 >>> s[20]
42 Traceback (most recent call last):
43   File "<stdin>", line 1, in <module>
44   IndexError: string index out of range
```

## String methods:

<http://docs.python.org/library/stdtypes.html#string-methods>

<https://goo.gl/uBQPYA>

strings are *immutable* objects  
(read-only)

**Homework:** select a string method and write a simple program that demonstrates the usage of this method.



## Some frequently used string methods

`s.lower()`, `s.upper()`

returns a lowercase, uppercase version of the string

`s.strip()`

removes the whitespace characters from both ends of the string

`s.isalpha()` / `s.isdigit()` / `s.isspace()`...

verifies if all characters of the string belong to the given character class

`s.startswith('other')`, `s.endswith('other')`

verifies if the string starts / ends with the other string

`s.find('other')`

Does the string include the other? If yes, return the index of the first character's occurrence. If not, return -1.

`s.replace('old', 'new')`

in the string replace all occurrences of 'old' with 'new'

`s.split('delim')`

Splits a string by a delimiter. Returns a list. See later.

`s.join(list)`

Opposite of *split*. Concatenates a list of strings by a delimiter. See later.

# Python is another tool



Consider Python as a new tool on your toolbelt.

Analyze the problem and choose the most appropriate tool.

# Eastern wisdom

*"I hear and I forget. I see and I remember. I do and I understand."*

Confucius



That is: practice [1], practice [2] and practice [3][4][5]...

[1] <https://arato.inf.unideb.hu/szathmary.laszlo/pmwiki/index.php?n=En.PyExercises>

[2] <http://www.pythonchallenge.com/>

[3] <http://projecteuler.net/> (mainly mathematical exercises)

[4] <https://www.hackerrank.com>

[5] <https://adventofcode.com/>

# Tips and tricks

Some useful aliases under Linux:

```
# ~/.bashrc
alias p2='python2'
alias p3='python3'
alias p='python3'

alias ..='cd ..'
alias ...='cd ..; cd ..'
# can be continued...
```

Don't type unnecessarily...



# Homework

1. [20121001a] string method