



# Scripting Languages

Laszlo SZATHMARY  
University of Debrecen  
Faculty of Informatics

## Lab #1

- introduction
- string data type

(last update: 2024-02-19 [yyyy-mm-dd])

2023-2024, 2nd semester





# About the course

**Title of the Course:** Scripting Languages

**Course ID:** INBPA9942L

**Type of the course:** Differentiated knowledge topics

**Credits:** 3

**Pre-requisite:** Programming Languages 1

**Home Page of the Instructor:**

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



# Requirements

At the end of the semester you will get a **practical course mark**. For this, you will have to attend the labs. **Max. 3 absences** are tolerated. If you are absent more than 3 times, you will automatically fail the course. **Being late:** I can tolerate if you are a bit late. However, two longer delays will be considered as one absence!

There will be **two classroom tests**. The first one on paper (?), while the second one on computer. Your proposed mark will be the average of the marks you get on the tests.

If the proposed mark is an **integer**, then this is your **final mark** and you cannot improve it. If this mark is a **real number** (e.g. 3.5, 4.5, etc.), then **you can try to improve** (or worsen) your mark in the first week of the exam period. In this case, you can get a mark which is 0.5 better (or worse) than your proposed mark. I will give you one or two exercises that you'll have to solve on computer and I can also ask theoretical questions about the Python programming language.

If someone fails both classroom tests, (s)he cannot improve the final mark.



# Requirements

## Home assignments

You'll get several homeworks every week. At the beginning of a class, I'll pick one or two exercises that you'll have to show me. At least 75% of the homeworks must be solved. That is, at least 75% of the exercises must be solved that I select for presentation.

If you solve less than 75% of the homeworks, then your final mark will be worse than the average of your two tests!

If you solve less than 50% of the homeworks, then your final mark will be 1 (fail). In this case there is no possibility to improve your mark.

In your solutions, you can only use techniques that we've already learnt so far!

# Requirements

## Certiport exam

- <https://www.inf.unideb.hu/en/certiport>

If you complete the course successfully (i.e., your final mark is at least 2), and you get a Certiport certificate from Python before the end of the semester (exam period included), then I'm willing to add +1 to your final mark.





# Bibliography

- *Guido van Rossum*: **Python Tutorial** (<https://docs.python.org/3/download.html>, also in PDF), 2024
- *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 January 19, 2024 there were **507,699** 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 3.12 . Since Python 2 (2.7) is still used, the key differences will be mentioned. Recommended version: Python 3.8+



# Links

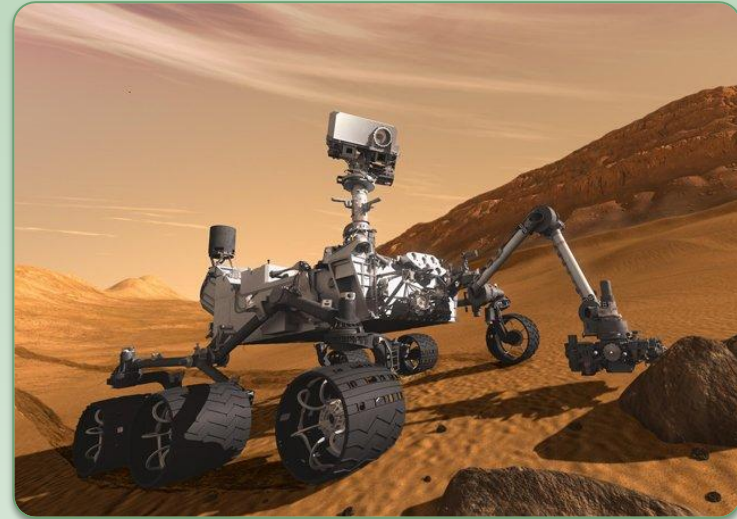
- Python HQ: <https://www.python.org/>
- Python documentation: <https://docs.python.org/>
- The Python Standard Library: <https://docs.python.org/library/>
- Python FAQ: <https://docs.python.org/faq/general.html>
- PEP 8 -- Style Guide for Python Code:  
<https://www.python.org/dev/peps/pep-0008/>
  
- <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 (<https://www.tiobe.com/tiobe-index/> )



Jan 2024	Jan 2023	Change	Programming Language	Ratings	Change
1	1		 Python	13.97%	-2.39%
2	2		 C	11.44%	-4.81%
3	3		 C++	9.96%	-2.95%
4	4		 Java	7.87%	-4.34%
5	5		 C#	7.16%	+1.43%
6	7	^	 JavaScript	2.77%	-0.11%
7	10	^	 PHP	1.79%	+0.40%
8	6	v	 Visual Basic	1.60%	-3.04%
9	8	v	 SQL	1.46%	-1.04%
10	20	^	 Scratch	1.44%	+0.86%
11	12	^	 Go	1.38%	+0.23%
12	27	^	 Fortran	1.09%	+0.64%
13	17	^	 Delphi/Object Pascal	1.09%	+0.36%
14	15	^	 MATLAB	0.97%	+0.06%
15	9	v	 Assembly language	0.92%	-0.68%
16	11	v	 Swift	0.89%	-0.31%
17	25	^	 Kotlin	0.85%	+0.37%
18	16	v	 Ruby	0.80%	+0.01%
19	18	v	 Rust	0.79%	+0.18%
20	31	^	 COBOL	0.78%	+0.45%

# Literature



amazon Deliver to Hungary

Books python programming

All Today's Deals Registry Customer Service Gift Cards Sell

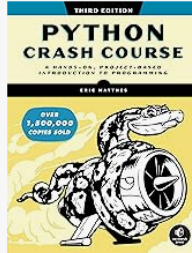
Books Advanced Search New Releases Best Sellers & More Amazon Book Clubs Children's Books Textbooks Textbook Rentals Best Books of the Month Best Books of 2023 Your Company Bookshelf

1-16 of over 10,000 results for "python programming"

Kindle Unlimited  
 Kindle Unlimited Eligible


Department  
Books  
Arts & Photography  
Biographies  
Business & Money  
Calendars  
Children's Books  
Christian Books & Bibles  
Comics & Graphic Novels  
Computers & Technology  
Cookbooks, Food & Wine  
Crafts, Hobbies & Home  
Education & Teaching  
Engineering & Transportation  
Health, Fitness & Dieting  
History  
Humor & Entertainment  
Law  
LGBTQ+ Books  
Literature & Fiction  
Medical Books  
Mystery, Thriller & Suspense  
Parenting & Relationships  
Politics & Social Sciences  
Reference  
Religion & Spirituality  
Romance  
Science & Math  
Science Fiction & Fantasy  
Self-Help

Results

 **Python Crash Course, 3rd Edition: A Hands-On, Project-Based Introduction to Programming**  
by Eric Matthes | Jan 10, 2023  
★★★★★ ~ 697  
**Paperback**  
\$31<sup>99</sup> List: \$49.99  
Delivery **Thu, Feb 8**  
Ships to Hungary  
More Buying Choices  
\$28.42 (57 used & new offers)

**Kindle**  
\$26<sup>99</sup> Print List Price: \$49.99  
Available Instantly

**Spiral-bound**  
\$50<sup>90</sup> List: \$56.99

 **Python Programming Language**  
by Berajah Jayne | May 1, 2019  
★★★★☆ ~ 2,091  
**Paperback**  
\$7<sup>95</sup>  
Delivery **Thu, Feb 8**  
Ships to Hungary  
More Buying Choices  
\$6.95 (10 used & new offers)

<https://www.amazon.com/s?k=python+programming>

(last checked on January 19, 2024)



# Conferences

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

## PyCon US 2023

- 142 videos
- [https://www.youtube.com/playlist?list=PL2Uw4\\_HvXqvY2zhJ9AMUa\\_Z6dtMGF3gtb](https://www.youtube.com/playlist?list=PL2Uw4_HvXqvY2zhJ9AMUa_Z6dtMGF3gtb)
- there are several *tutorials* among the presentations

## PyCon US 2022

- 112 videos
- [https://www.youtube.com/playlist?list=PL2Uw4\\_HvXqvYeXy8ab7iRHjA-9HiYhRQI](https://www.youtube.com/playlist?list=PL2Uw4_HvXqvYeXy8ab7iRHjA-9HiYhRQI)
- 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!")
```

## Using special characters:

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

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.

```
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

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

def

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: <https://bit.ly/3R0PN7G>

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

most important changes in Python 3



```
>>> print "hello" ← statement
hello
>>>
>>> 7 / 2 ← integer division (like in C)
3
>>>
>>> 7 // 2 ← integer division (always)
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/>

One more tip: Try to solve the exercises of the Programming 2 course in Python.  
And vice versa: try to solve the exercises that you get here in Java / C# / etc.



# Tips and tricks

Some useful aliases:

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

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

Don't type unnecessarily...



# Homework

1. [[20121001a](#)] string method