



Szkriptnyelvek

Szathmáry László
Debreceni Egyetem
Informatikai Kar

1. Gyakorlat

- bevezető
- a sztring adattípus

(utolsó módosítás: 2022. aug. 19.)

2022-2023, 1. félév



A tantárgyról

A tantárgy neve: Szkriptnyelvek

A tantárgy kódja: INBPM9942L / INBMM9945L

A tárgy PTI / MI BSc-s hallgatók számára lett meghirdetve.

Sáv, kredit: kötelezően választható, 3 kredit

Előfeltétel: Programozás 1

A gyakorlatvezető honlapja: <https://arato.inf.unideb.hu/szathmary.laszlo>

A gyakorlat ideje és helye:

- kedd 8.00-10.00, ???

Követelmények

A **gyakorlati jegy** megszerzésének egyik feltétele a **rendszeres részvétel** a gyakorlatokon. A félév során legfeljebb 3 hiányzás megengedett. Aki ezt túllépi, az automatikusan elégtelent kap. **Késés:** egy pár perces késést még tolerálok, de egy nagyobb késést már fél hiányzásnak fogok venni. Vagyis két nagyobb késés egy hiányzásnak lesz elszámolva.

A szorgalmi időszakban **2 zárthelyi dolgozat** lesz. Az első **papíron**, a második **számítógépen**. A két ZH átlaga alapján megajánlok egy jegyet.

Ha a megajánlott jegy egész érték, akkor **nincs** javítási lehetőség. Ha valaki két jegy között áll (pl. 3/4, 4/5), annak lesz javítási (rontási?) lehetősége a vizsgaidőszak első hetében. Ekkor **legfeljebb egy fél jegyet** lehet javítani/rontani. A hallgatónak néhány konkrét programozási feladatot kell helyben megoldania, illetve ismernie kell a Python programozási nyelvvel kapcsolatos fogalmakat. Akinek mindkét zárthelyi dolgozata elégtelen lett, annak nincs lehetősége javításra.

Követelmények

Házi feladatok

Minden héten számos házi feladatot kapnak. A leadási határidő után kiválasztok ezek közül néhányat, s ezeket nézem meg tüzetesebben. A házi feladatoknak legalább a 80%-át meg kell oldani. Ebbe a 80%-ba csak az általam kiválasztott feladatokat számítom be.

Leadási határidő: a gyakorlat előtti nap éjféli.

Ha valaki a **házi feladatok** 80%-ánál kevesebbet old meg, akkor a két zárthelyi dolgozatra kapott jegyek átlagánál gyengébb jegyet fogok megajánlani.

Ha valaki a **házi feladatok** 50%-ánál kevesebbet old meg, akkor a félévi teljesítményére 1-est adok. Ekkor nincs javítási lehetőség.

A házi feladatok megoldásaiban csak az addig tanultakat lehet felhasználni!

Követelmények

Certiport tanúsítvány

- <https://www.inf.unideb.hu/certiport>
- <https://arato.inf.unideb.hu/szathmary.laszlo/pmwiki/index.php?n=Acad.Certificates>

Ha valaki sikeresen teljesíti a tárgyat (vagyis legalább 2-est kap), s a félév vége előtt szerez egy Python tanúsítványt, akkor hajlandó vagyok az illetőnek **két jeggyel jobb** osztályzatot adni.



Ajánlott irodalom

- *Guido van Rossum*: **Python Tutorial**
(<https://docs.python.org/3/download.html>, PDF-ben is, 150 oldal), 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 változat: **Python Module of the Week** (<https://pymotw.com/3/>)]
- *Gérard Swinnen*: **Tanuljunk meg programozni Python nyelven**
(2005, Python 2.2)
(online letölthető: <http://mek.oszk.hu/08400/08435/>)
- *Rashi Gupta*: **Mindentudó Python**
(2003, Python 2.2)

Ajánlott irodalom (folyt.)

Python 3

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

Expert

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

Bevezető



- A Python egy általános célú, nagyon magas szintű programozási nyelv.
- Fő tervezési szempont: olvashatóság.
- Interpreteres nyelv, a megírt program azonnal futtatható.
- Multiparadigmás (imperatív, objektum-orientált, funkcionális).
- Az első változat 1991-ben jelent meg, nevét a Monty Python csoportról kapta.
- Tervezője Guido van Rossum holland kutató / programozó (1956-ban született). 2005-2012: Google; 2013-2019: Dropbox. 2019 végén nyugdíjba vonult, de 2020 végén visszatért és csatlakozott a Microsoft-hoz.
- Mely nyelvek voltak rá hatással: ABC, ALGOL 68, C, C++, Dylan, Haskell, Icon, Java, Lisp, Modula-3, Perl.
- Mely nyelvekre volt hatással: Boo, Cobra, D, Falcon, Groovy, JavaScript, Ruby, Go.

Bevezető



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

Bevezető

- Dinamikus típusokat és automatikus memóriakezelést használ.
- Platformfüggetlen (Unix/Linux, Windows, Mac OS, stb.)
- A Pythonnak igen kiterjedt és széles körű standard könyvtára van („batteries included”), amit még kiegészítenek az egyéb (mások által megírt) publikus modulok („3rd party modules”)¹.
- Az interpreter és a standard könyvtár teljesen nyílt forrású.
- Könnyen tanulható, egyszerű a szintaxisa. A megírt kód könnyen olvasható.
- A programozói munkát hatékony magas szintű adatszerkezetek segítik. Egyszerűen, ugyanakkor nagyon hatásosan valósítja meg az objektumorientált programozást.

¹ Lásd <https://pypi.org> (2022. 08. 19-én **395 420** csomag volt elérhető ; 2022. 02. 03-án **354 369** csomag volt elérhető).

Bevezető

- Ideális nyelv szkriptek írásához, illetve gyors alkalmazásfejlesztéshez („rapid application development”).
- Gyors prototípusfejlesztést tesz lehetővé („rapid prototyping”).
- Hasonló programozási nyelvek: Perl, Ruby.
- Tökéletes választás kisebb (pl. 10-20 soros) szkriptekhez, de NEM CSAK erre jó! Nagy méretű, több ezer soros programokat is lehet benne írni úgy, hogy a program áttekinthető marad (modulok, csomagok).
- Sokáig két ág létezett egymással párhuzamosan: Python 2 és Python 3. A Python 3 fork 2008. dec.-ben jött létre. A Python 2.7-hez 2020. január 1. óta már nem jön ki újabb frissítés!
- A gyakorlaton a Python 3-as verzióját fogjuk használni. A jelenlegi legfrissebb verzió a Python 3.10-es. Mivel nagyon sok helyen még mindig a Python 2-t használják (2.7), ezért ki fogunk térni a legfontosabb eltérésekre. Javasolt verzió: Python 3.8+.

Linkek

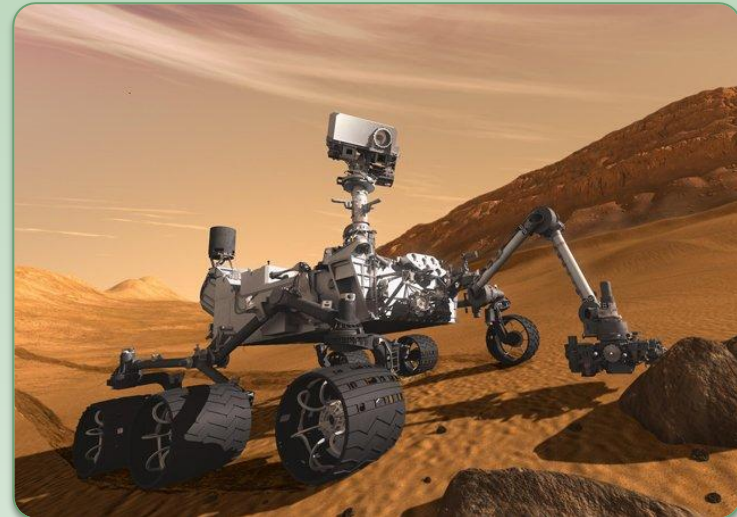
- Python HQ: <https://www.python.org/>
- Python dokumentáció: <https://docs.python.org/>
- A Python Standard Library: <https://docs.python.org/library/>
- Python FAQ: <https://docs.python.org/faq/general.html>
- PEP 8 -- Style Guide for Python Code:
<https://www.python.org/dev/peps/pep-0008/>
- <https://www.reddit.com/r/learnpython>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

Hol használják?

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

- **Scientific**

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



Mars Curiosity (2012. aug. 6.)


Szoftver: 2,5 millió C programsor.

















A logfile-ok tesztelését Python szkriptekkel végezték.

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

Mennyire népszerű?

TIOBE index (<https://www.tiobe.com/tiobe-index/>)



Aug 2022	Aug 2021	Change	Programming Language	Ratings	Change
1	2	▲	 Python	15.42%	+3.56%
2	1	▼	 C	14.59%	+2.03%
3	3		 Java	12.40%	+1.96%
4	4		 C++	10.17%	+2.81%
5	5		 C#	5.59%	+0.45%
6	6		 Visual Basic	4.99%	+0.33%
7	7		 JavaScript	2.33%	-0.61%
8	9	▲	 Assembly language	2.17%	+0.14%
9	10	▲	 SQL	1.70%	+0.23%
10	8	▼	 PHP	1.39%	-0.80%
11	16	▲▲	 Swift	1.27%	+0.30%
12	12		 Classic Visual Basic	1.27%	+0.04%
13	22	▲▲	 Delphi/Object Pascal	1.22%	+0.60%
14	23	▲▲	 Objective-C	1.22%	+0.61%
15	18	▲	 Go	0.98%	+0.08%
16	14	▼	 R	0.92%	-0.13%
17	17		 MATLAB	0.90%	-0.08%
18	15	▼	 Ruby	0.82%	-0.18%
19	13	▼▼	 Fortran	0.81%	-0.32%
20	20		 Perl	0.72%	-0.06%

Szakirodalom

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
5th Edition
by Mark Lutz
★★★★☆ ~ 1,453
Paperback
Other format: Kindle

(lekérdezés dátuma: 2022. aug. 19.)

Konferenciák

A legnagyobb konferencia a PyCon US (<https://us.pycon.org/>).

PyCon US 2022

- 112 db videó
- https://www.youtube.com/playlist?list=PL2Uw4_HvXqvYeXy8ab7iRHjA-9HiYhRQI
- az előadások között tutorialok is vannak

PyCon US 2021

- előadások: <https://pyvideo.org/events/pycon-us-2021.html> (85 videó)
- https://www.youtube.com/playlist?list=PL2Uw4_HvXqvYk1Y5P8kryoyd83L_0Uk5K
- az előadások között tutorialok is vannak

A <https://pyvideo.org/> oldalon további konferenciák videóanyagai is fent vannak.

Kvíz

Hogy hívják a nyelv alkotóját?

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

HF: Nézzünk utána, hogy kik a többiek.

Interpreter használata:

```
[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.
>>> █
```

Szkript írása:

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

Ékezetek használata:

```
1 #!/usr/bin/env python3
2 # coding: utf-8
3
4 def main():
5     # megjegyzés ékezettel
6     print("Helló!")
```

Python 2-ben ékezetek esetén
ezt ki **kellett** írni.

Python 3-ban ez az alapértelmezés,
de nyugodtan kiírhatjuk.

```
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'
```

változót nem kell külön deklarálni

def

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

kettőspont

indentálás

nincs pontosvessző

Stílus: egy függvény előtt és után hagyjunk ki 2 üres sort.

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

} Direkt módon futtatjuk
vagy modulként?

Írassuk ki a parancssori argumentumokat:

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

A továbbiakban nem írom ki külön a
#!/usr/bin/env python3
sort...

Majd: adjunk meg egy nevet argumentumként (pl. `./hello.py Bob`),
s üdvözljük az illetőt („Hello Bob!”).

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

if után nem kell zárójel

Csak akkor derül ki a hiba, ha idekerül a vezérlés!

Ezért (is) lényegesek a unit tesztek komolyabb rendszerek esetén.
Minden ágot le kell velük tesztelni.

Egy általános minta Python 3 szkriptekhez

```
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()
```

Tipp: mentjük el ezt a file-t `alap.py` néven, majd ha egy új Python programot akarunk írni, akkor egyszerűen csak készítsünk erről egy másolatot.

A minta letölthető innen: <https://goo.gl/IELOYy>

```
>>> 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
>>>
```

← függvény

← matematikai osztás!

← az egész osztás operátora
(mindig ezt jelenti)

A Python 3
leglényegesebb
változásai



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

← utasítás

← egész osztás, mint C-ben

← az egész osztás operátora
(mindig ezt jelenti)

Sztringek



```
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
```

Sztring metódusok:

<https://docs.python.org/3/library/stdtypes.html#string-methods>

<https://goo.gl/uBQPYA>

strings are *immutable* objects
(read-only)

HF: válasszunk ki egy sztring metódust s írjunk egy kis szkriptet ami bemutatja a használatát.

Néhány gyakori sztring metódus

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

a sztring kisbetűs, nagybetűs verziójával tér vissza

`s.strip()`

a whitespace karaktereket levágja a sztring elejéről és végéről

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

megnézi, hogy a sztring vmennyi karaktere az adott karakterosztályba tartozik-e

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

megnézi, hogy a sztring a másik sztringgel kezdődik-e / végződik-e

`s.find('other')`

A sztringben szerepel-e a másik sztring (nem reguláris kifejezésként adjuk meg).

Ha igen, akkor az első előfordulás első karakterének indexével tér vissza.

Ha nem, akkor -1 a visszatérési érték.

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

a sztringben az 'old' vmennyi előfordulását 'new'-ra cseréli

`s.split('delim')`

A sztringet az adott szeparátor mentén részsstringek listájára bontja. A szeparátor nem reguláris kifejezés. Példa: `'aaa,bbb,ccc'.split(',')` -> `['aaa', 'bbb', 'ccc']`. Ha csak `s.split()` -et írunk, akkor a whitespace karakterek mentén bontja fel a sztringet.

`s.join(list)`

A `split()` ellentéte. Egy lista elemeit kapcsolja össze egy adott szeparátorral (ez lesz az s sztring). Példa: `'---'.join(['aaa', 'bbb', 'ccc'])` -> `aaa---bbb---ccc`.

A Python egy újabb eszköz



A Pythonra úgy tekintünk, mint egy újabb eszközre az eszköztárunkban.
Az adott feladathoz a legmegfelelőbb eszközt használjuk.

Keleti bölcsesség

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

Confucius



Vagyis: gyakorolni [1], gyakorolni [2] és gyakorolni [3][4][5]...

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

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

[3] <http://projecteuler.net/> (főleg matematikai jellegű feladatok)

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

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

Még egy tipp: a Prog. 2-n kapott feladatokat gyakorlásképpen oldják meg Pythonban is. Fordítva is lehet: az itt kapott feladatokat oldják meg Java / C# nyelven is.

Tippek és trükkök rovat

Ajánlott rövidítések:

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

alias ..='cd ..'
alias ...='cd ../..'
# lehet folytatni...
```

Ne gépeljünk feleslegesen...

Feladatok

1. [20121001a] sztring metódus