



Szkriptyelvek

Szathmáry László
Debreceni Egyetem
Informatikai Kar

10. Gyakorlat

- [weboldalak letöltése](#)
- [JSON szerializáció](#)

(utolsó módosítás: 2025. aug. 14.)

2025-2026, 1. félév



Weboldalak letöltése

```
4 import urllib.request
5
6 def main():
7     response = urllib.request.urlopen("http://python.org")
8     raw = response.read()
9     print(type(raw))      # bytes ←
10    # print(raw)
11
12    html = raw.decode("utf-8")
13    print(type(html))    # str ←
14    # print(html)
```

Mivel ez gyakori művelet, ezért ezt gyorsan tegyük is bele a `pygyak.py` modulunkba `get_page()` néven. Használata a következőképpen nézzen ki:

```
3 from pygyak import get_page
4
5 url = 'http://python.org'
6
7 def main():
8     print(get_page(url)) ←
```

Sztringet adjon vissza!

URL letöltése (ez lehet HTML, kép, stb.).

A letöltött objektum a fájlrendszerben jelenik meg:

```
4 import urllib.request
5
6 url = "http://www.nhdfl.org/uploads/NHB%20photos/HB0P2.jpg"
7
8 def main():
9     urllib.request.urlretrieve(url, "/tmp/forest.jpg")
```

Vagy használhatunk a letöltésre egy külső programot is, pl. a wget -et:

```
3 import os
4
5 url = 'http://wallpapers.leovacity.be/images/Forrest_wallpaper.jpg'
6 to = '/tmp/forrest.jpg'
7
8 def main():
9     cmd = 'wget {url} -O {output}'.format(url=url, output=to)
10    print(cmd)
11    os.system(cmd)
```



külső program hívása külön processzben

A `requests` modul

A Python standard library a következő modulokat tartalmazza URL-ek letöltésére:

- `urllib` (<https://docs.python.org/3/library/urllib.html>), gyűjtő-modul
- `urllib.request` (<https://docs.python.org/3/library/urllib.request.html>)
- ...

Viszont ezeknek a használata elégé körülményes tud lenni :(

Ezért érdemes megismerkedni a `requests` modullal ([link](#)). Ez nem része a standard library-nek, külön kell installálni. Sokkal egyszerűbb a standard könyvtár moduljainál...

Az előző link alján részletes User Guide található.

[Előadás a modul szerzőjétől.](#)

Ezt csak 1x kell lefuttatni
(ha még nincs fent a pip3
csomagkezelő a géünkön).

Telepítése:

```
sudo apt install python3-pip
```

```
pip3 install requests --user -U
```

requests telepítése / frissítése

A `requests` modul használata

```
import requests

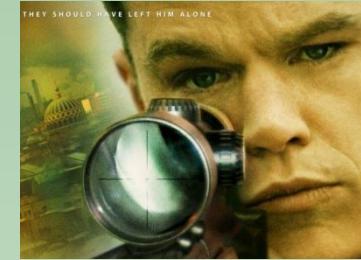
url = "http://index.hu"
r = requests.get(url)
print(r.text)
```

Feladat

Dívány szavazatok testreszabása

Link: <https://arato.inf.unideb.hu/szathmary.laszlo/pmwiki/index.php?n=Py3.20121026a>

JSON sserializáció

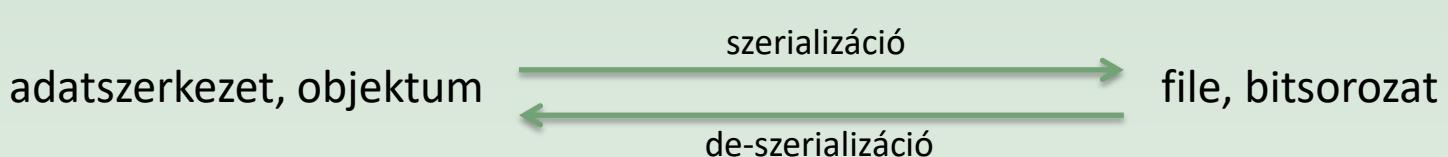


- <http://json.org/>
- <http://en.wikipedia.org/wiki/Json>

A **json** modul segítségével memóriában lévő Python objektumokat tudunk JSON formátumba sserializálni.

Sserializáció

Egy adatszerkezet vagy objektum konvertálása olyan formátumba, amelyet le lehet tárolni pl. egy fájlban (vagy hálózaton keresztül át lehet küldeni), majd később ugyanabban, vagy egy másik számítási környezetben „újra lehet éleszteni”.



Formátumok: XML, JSON, YAML, stb.

A JSON sok esetben használható az XML alternatívjaként

- szintén szöveg-alapú, az ember számára is jól olvasható
- szintén hierarchikus felépítésű
- szintén nagyszerűen használható alkalmazások közti kommunikációra
- egyszerűbb mint az XML
- nem annyira bőbeszédű, rövidebb
- gyorsabban lehet írni/olvasni

XML:

```
<Person>
  <FirstName>Homer</FirstName>
  <LastName>Simpson</LastName>
  <Relatives>
    <Relative>Grandpa</Relative>
    <Relative>Marge</Relative>
    <Relative>The Boy</Relative>
    <Relative>Lisa</Relative>
  </Relatives>
</Person>
```

JSON:

```
{
  "firstName": "Homer",
  "lastName": "Simpson",
  "relatives": [ "Grandpa", "Marge", "The Boy", "Lisa" ]
}
```

XML:

```
<persons>
  <person>
    <name>Ford Prefect</name>
    <gender>male</gender>
  </person>
  <person>
    <name>Arthur Dent</name>
    <gender>male</gender>
  </person>
  <person>
    <name>Tricia McMillan</name>
    <gender>female</gender>
  </person>
</persons>
```

JSON:

```
[
  {
    "name": "Ford Prefect",
    "gender": "male"
  },
  {
    "name": "Arthur Dent",
    "gender": "male"
  },
  {
    "name": "Tricia McMillan",
    "gender": "female"
  }
]
```

XML:

```
<settings>
  <path>/home/luke/Dropbox/Public</path>
  <user_id>123456</user_id>
  <auto_sync>True</auto_sync>
</settings>
```

(forrás: [Dropbox Publicus](#))

JSON:

```
{
  "path": "/home/luke/Dropbox/Public",
  "user_id": 123456,
  "auto_sync": true
}
```

JSON szintaktikai szabályok

- az adatok kulcs/érték párok
- az adatok vesszővel vannak elválasztva
- a kapcsos zárójelek objektumot jelölnek
- a szöglletes zárójelek tömböt jelölnek

JSON értékek

Egy JSON érték a következő lehet:

- szám (egész vagy lebegőpontos)
- sztring (idézőjelek között)
- boolean (true vagy false)
- tömb (szöglletes zárójelek között)
- objektum (kapcsos zárójelek között)
- null

JSON file betöltése

input file ([person.json](#))

pretty print
(adatszerkezetek olvashatóbb kiíratása)

```

3 from pprint import pprint
4 import json
5
6
7 def read_file():
8     f = open('person.json', 'r')
9     d = json.load(f) ←
10    f.close()
11
12    print(type(d))
13    pprint(d, indent=4)

```

visszaadott érték: Python adatszerkezet
(itt a példában: dictionary)

```

1 {
2     "last": "Doe",
3     "first": "John",
4     "age": 39,
5     "sex": "M",
6     "registered": true,
7     "salary": 70000
8 }

```

json modul
(json fájlok írása / olvasása)

```
{
    'age': 39,
    'first': 'John',
    'last': 'Doe',
    'registered': True,
    'salary': 70000,
    'sex': 'M'
}
```

kimenet (szótár)

JSON sztring betöltése

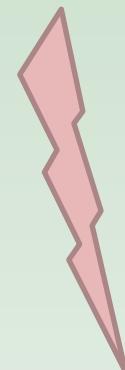
```
7 def read_string():
8     s = """
9         "path": "/home/luke/Dropbox/Public",
10        "user_id": 123456,
11        "auto_sync": true
12    """
13
14    d = json.loads(s) ←
15
16    print(d)
```

(lásd [settings.py](#))

loads
(**s** mint sztring)

Lényeg

A `json` modul vesz egy sztringet (akár egy fájlból, akár egy karakterszorozatból), s abból **felépít egy Python objektumot** (pl. listát, szótárat). Vagyis a sztringet **nem nekem kell parse-olni**, hanem a modul ezt elvégzi helyettünk!



Feladatok

Saját IP címem lekérdezése

Link: <https://arato.inf.unideb.hu/szathmary.laszlo/pmwiki/index.php?n=Py3.20120920g>

Reddit-ről tájképek letöltése

Link: <https://arato.inf.unideb.hu/szathmary.laszlo/pmwiki/index.php?n=Py3.20121126a>

Adatszerkezet kiírása file-ba JSON formátumban

```

3 import json
4
5 def write_file():
6     person = {
7         'last': 'Doe',
8         'first': 'John',
9         'age': 39,
10        'sex': 'M',
11        'registered': True,
12        'salary': 70000,
13    }
14
15    f = open('employee.json', 'w')
16    json.dump(person, f)
17    f.close()

```

(lásd [write_person.py](#))

Python szótár

json.dump(data_structure, file_handler)

print(json.dumps(person))

az adatszerkezből JSON sztringet készít, s azt adja vissza

json.dump(person, f, indent=4)

csinos kiíratás

json.dump(person, f,
 indent=4, sort_keys=True)

csinos kiíratás,
kulcsok szerint rendezve



házi feladat



Feladatok

1. [\[20120816b\] get_alap.py](#)
2. [\[20120815k\] szökés Alcatrazból](#)