



# Bevezetés a Python programozási nyelvbe

Szathmáry László

Debreceni Egyetem  
Informatikai Kar

9. Gyakorlat (2012. nov. 29.)

- weboldalak letöltése
- JSON szerializáció



# Weboldal letöltése

Weboldal letöltése. A letöltött HTML forrás sztringként lesz visszaadva:

```
3 import urllib2
4
5 def main():
6     response = urllib2.urlopen('http://python.org/')
7     html = response.read()
8     print html
```

Mivel ez gyakori művelet, ezért ezt 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)
```

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

A letöltött objektum a fájlrendszerben jelenik meg:

```
3 import urllib
4
5 url = 'http://wallpapers.leovacity.be/images/Forrest_wallpaper.jpg'
6
7 def main():
8     urllib.urlretrieve(url, '/tmp/forrest.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

# Feladat

Dívány szavazatok testreszabása

Link: <http://www.inf.unideb.hu/~szathml/pmwiki/index.php?n=Py.20121026a>

# Szorgalmi

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

- urllib (<http://docs.python.org/2/library/urllib.html>)
- urllib2 (<http://docs.python.org/2/library/urllib2.html>) [[HOWTO](#)]

Viszont ezeknek a használata eléggé körülményes tud lenni :(

Ezért érdemes megismerkedni a **requests** modullal ([link](#)). Ez nem része a standard library-nek, külön kell installálni. Sokkal egyszerűbb a fenti két modulnál...

Az előző link alján részletes User Guide található.

[Előadás a modul szerzőjétől.](#)

Telepítése:

```
sudo apt-get install python-pip  
sudo pip install pip -U  
sudo pip install requests -U
```

Ezt csak 1x kell lefuttatni  
(ha még nincs fent a pip  
csomagkezelő a gépünkön).

Ezt frissíti a pip  
csomagkezelőt a legfrissebb  
verzióra.

requests installálása / frissítése

# JSON szerializáció

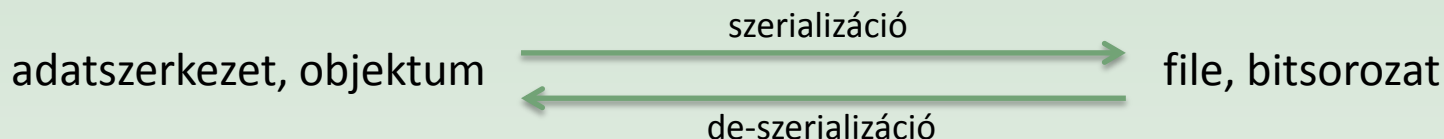
- <http://json.org/>
- <http://en.wikipedia.org/wiki/Json>
- <http://www.w3schools.com/json/>



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

## Szerializáció

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 alternatívájaként

- 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 Syntax Rules

JSON syntax is a subset of the JavaScript object notation syntax.

- Data is in name/value pairs
- Data is separated by comma
- Curly brackets holds objects
- Square brackets holds arrays

## JSON Values

JSON values can be:

- A number (integer or floating point)
- A string (in double quotes)
- A Boolean (true or false)
- An array (in square brackets)
- An object (in curly brackets)
- null

(forrás)

# JSON file betöltése

input file (person.json)

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

pretty print  
(adatszerkezetek olvashatóbb kiírása)

json modul  
(json fájlok írása / olvasása)

```
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)
```

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

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

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 karaktersorozatból), 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: <http://www.inf.unideb.hu/~szathml/pmwiki/index.php?n=Py.20120920g>

Reddit-ről tájképek letöltése

Link: <http://www.inf.unideb.hu/~szathml/pmwiki/index.php?n=Py.20121126a>

# Adatszerkezet kiírása **file-ba** JSON formátumban

(lásd [write\\_person.py](#))

```
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()
```

Python szótár

`json.dump(data_structure, file_handler)`

`print json.dumps(person)`

az adatszerkezetből JSON sztringet  
készít, s azt adja vissza

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

csinos kiírás

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

csinos kiírás,  
kulcsok szerint rendezve