

PYTHON AZ ADATOK TENGERÉBEN

2020.02.27. HWSW MEETUP

RABATIN GÁBOR — DATA SCIENTIST

ADAT

Adatbázisok

- SQL lekérdezések, ORM

pyodbc, SQLAlchemy



ADAT

Adatbázisok

- SQL lekérdezések, ORM

pyodbc, SQLAlchemy

API

- HTTP kérések
- Szolgáltatásokhoz külső lib-ek

requests, custom libs

ADAT

Adatbázisok

- SQL lekérdezések, ORM

pyodbc, SQLAlchemy

API

- HTTP kérések
- Szolgáltatásokhoz külső lib-ek

requests, custom libs

ADAT

Scraping

- Adott weboldal(ak)ról kell begyűjteni
- Jogi következmények!

BeautifulSoup, scrapy

Adatbázisok

- SQL lekérdezések, ORM

pyodbc, SQLAlchemy

Scraping

- Adott weboldal(ak)ról kell begyűjteni
- Jogi következmények!

BeautifulSoup, scrapy

API

- HTTP kérések
- Szolgáltatásokhoz külső lib-ek

requests, custom libs

Fájlok

- CSV, JSON, Excel
- Szöveg, hang, kép

Pandas, SciPy, NumPy

ADAT

ADATELŐKÉSZÍTÉS

ADATELŐKÉSZÍTÉS

- Nem megfelelő formátum
- Hibás, hiányos adat kezelése



MODELLEK ÉPÍTÉSE

MODELLEK ÉPÍTÉSE

- Összefüggések feltárása
- Előrejelzés
- Klaszterezés, anomáliadetekció



EREDMÉNYEK KÖZLÉSE

EREDMÉNYEK KÖZLÉSE

- Adatvizualizáció
 - Magyarázó ábrák
 - Adat megértése

matplotlib

 plotly

 bokeh

EREDMÉNYEK KÖZLÉSE

- Adatvizualizáció
 - Magyarázó ábrák
 - Adat megértése
- Webes felület
 - API
 - Weboldal

matplotlib

plotly

bokkeh

django



Flask



PYTHON ELŐNYEI

PYTHON ELŐNYEI

- Könnyű prototipizálás, gyors validáció
- Gyorsan tanulható
- Könnyen olvasható
- Open-source
- Jó közösség

SZERETHETŐ PYTHON

SZERETHETŐ PYTHON



Albérlet adatok letöltése

SZERETHETŐ PYTHON

1

Albérlet adatok letöltése

2

Adatok szűrése, megfelelő
formátumra hozása

SZERETHETŐ PYTHON

1

Albérlet adatok letöltése

2

Adatok szűrése, megfelelő
formátumra hozása

3

Adatvizualizáció

ZEN OF PYTHON

```
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```



KÖSZÖNÖM A FIGYELMETEKET

RABATIN.GABOR@DMLAB.HU – [LINKEDIN.COM/IN/GABORR/](https://www.linkedin.com/in/gaborr/)