Python

8. gyakorlat

Strohmayer Ádám

Agenda

- Fájlkezelés
  - Jupyter
  - FastAPI

Fájlkezelés

Fájl megnyitása: open(fájlnév, mód, kódolás)

többféle fájlkezelési mód!

olvasás (r), írás (w), hozzáírás (a), bináris műveletek (b)

Alapértelmezetten **le kell zárni a megnyitott fájlstreamet!**

```
try:
    fstream = open("test.txt", "r", encoding="utf-8")
    text = fstream.readline()
    print(text)
    fstream.close()
except FileNotFoundError as e:
    print("Nem található az adott fájl!", e)
```

```
- python test.py
szia anyu
```

Bármilyen alacsony szintű stream kezelésére jó (pl. socket)!

Fájlkezelés with-tel

With automatikusan lefuttatja a close-t a fájlstreamre!

Miért kell? – pl. több folyamat is hozzáférhessen a fájlhoz!

Dunderei: __enter__, __exit__

```
with open("test.txt", "r") as f:
    #f-ként hivatkozzunk a streamre!
    print(f.readline())
    #readline automatikusan EOF-ig vagy
newlineig olvas!
```

```
- python test.py
jĂşjcika
```

Az encoding meghatározása jó gyakorlat!

readlines() összes sort beolvassa egy változóba!

Fájlolvasás ciklussal

Readlines() listában fogja eltárolni az összes sort!

soronkénti beolvasás: **assignment expression**nel

```
with open("test.txt", "r") as f:
    unique_lines = set()
    while (line := f.readline()):
        print(line, end="")
        unique_lines.add(line)
    print(unique_lines)
```

```
- python test.py
j
u
u
j
c
i
k
a
 {'u\n', 'j\n', 'k\n', 'i\n', 'a', 'c\n'}
```

f.read() egy változóba olvassa az egész fájlt

(ha nincs paraméter megadva!)

Fájlírás

Fájlírás szintúgy az open metódussal lehetséges!
"w" mód: ha nincs ilyen fájl, létrehozza/felülírja az azonos nevűt
"a" mód: ha van ilyen fájl, hozzáfűzi a fájl végéhez a bemenetet

```
with open("test.txt", "a+", encoding="utf-8") as f:
    f.write(" reggie")
    f.seek(0) #fájl elejére visszaugrás!
    print(f.read())

with open("test.txt", "w+", encoding="utf-8") as f:
    f.write("hello")
    f.seek(0)
    print(f.read())
```

```
- python test.py
szia reggie
hello
```

JSON írása, olvasása

Import json – alap Python könyvtár része!

Beolvasás: json.load(fájlstream)

Kiíratás:

json.dump(adat, fájlstream, indent=4) – indent forma miatt!

```python
import json

with open("song.json", "w") as f:
    data_dict = {"title": "i", "release_date": "2014-09-23"}
    json.dump(data_dict, f, indent=4)
    #milyen adatot, milyen fájlba, milyen JSON formázással

with open("song.json", "r") as f:
    data = json.load(f)
    print(data)
```

```
- python test.py
{'title': 'i', 'release_date': '2014-09-23'}
```

További fájltípusok beolvasása az előadásdiákon!

Stream problémák típusai:

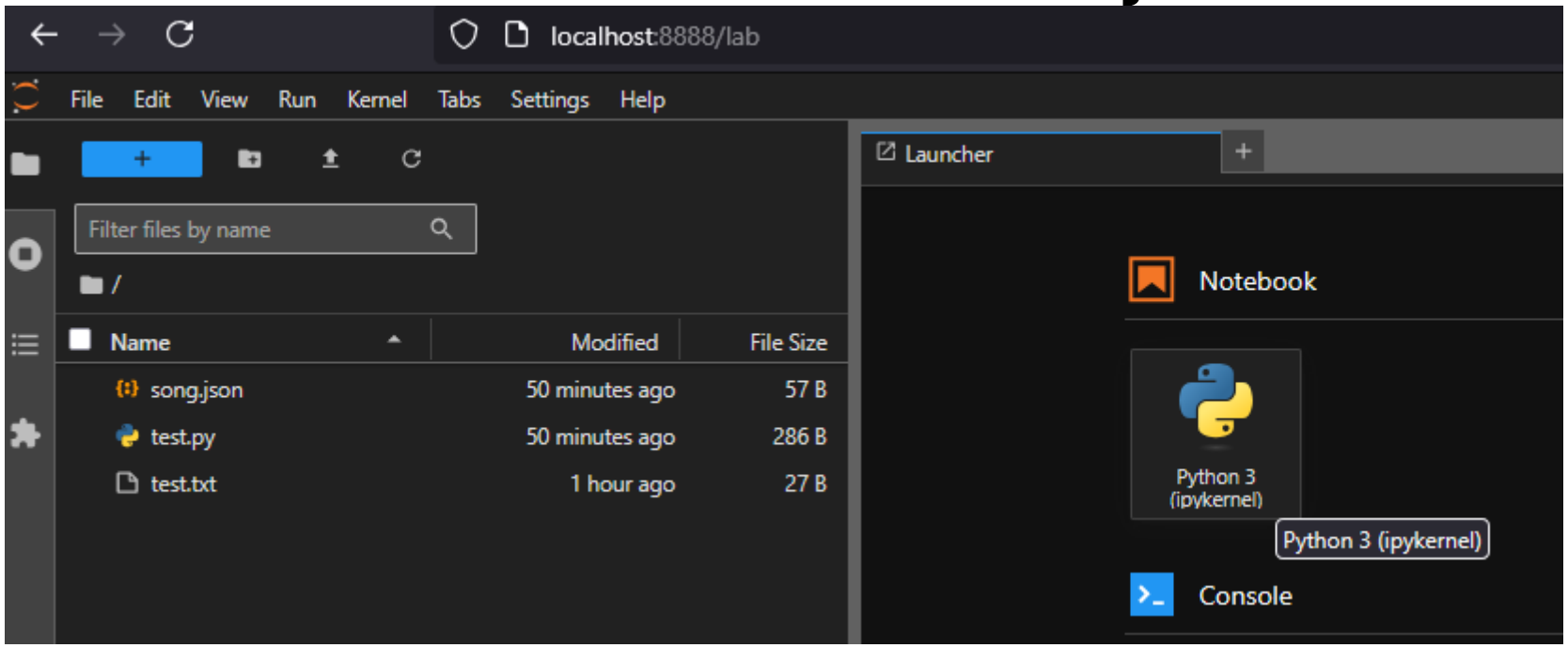
```
from os import strerror

try:
    s = open("c:/users/user/Desktop/file.txt","r")
    ###
    s.close()
except Exception as e:
    if e.errno == errno.ENOENT:
        print("A fájl nem létezik.")
    elif e.errno ==errno.EMFILE:
        print("Túl sok fájlt nyitott meg.")
```

Hibaüzenetek:

errno.EACCES → Permission denied
errno.EBADF → Bad file number
errno.EEXIST → File exists
errno.EFBIG → File too large
errno.EISDIR → Is a directory
errno.EMFILE → Too many open files
errno.ENOENT → No such file or directory
errno.ENOSPC → No space on device

Jupyter

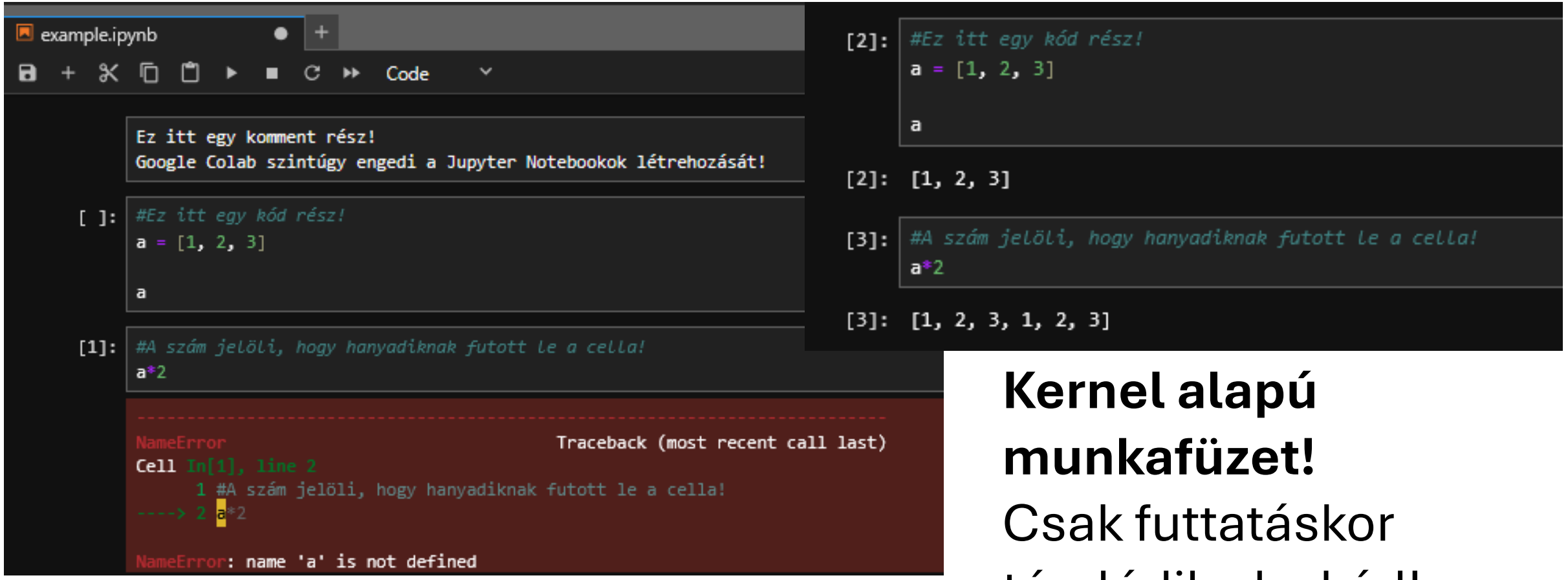
Interaktív webes környezet kódfuttatáshoz!
Notebook blokkokban futtathatja le a kódot!



pip install jupyterlab

Előhívása: jupyter lab (parancssorból)

Jupyter Notebook



Egyszerű kódpróbákat tesz lehetővé!

Kernel alapú munkafüzet!

Csak futtatáskor tárolódik el a kód!

FastAPI

Aszinkronitást biztosító webes keretrendszer!
Elsősorban API-k létrehozására használt, de alap webalkalmazásokat is lehet vele építeni.

Fő elve, hogy az erőforrásokat egységes interfészen keresztül lehessen elérni! (itt: HTTP metódusokkal)
A megfelelő műveletet egy erőforráson a megfelelő HTTP metódus jelzi! (GET, POST, PUT, DELETE)

Restful API fejlesztésére használják!

Virtuális környezetek

venv – Python alap része!

Izolált fejlesztési környezetet tesz lehetővé!

Használandó fájlokat NE tegyük a .venv-be!

python –m venv .venv

-m : modulként futtassuk a venv modult a jelenleg használt Python interpreterünkből

venv : maga a modul

.venv : virtuális környezet neve (. miatt rejtett általában)

Megnyitása : .venv\Scripts\Activate.ps1 vagy activate.bat

PowerShellből jogosultsághoz kötött, CMD-ből futtatható!

Egyszerű sütik beállítása

FastAPI lehetővé teszi a sütik beállítását – **itt set_cookie responseban kiosztja, kérésekben mindig át fogjuk adni!**

Sütik a böngészőben vannak eltárolva **(F12)**

```python
from fastapi import FastAPI, HTTPException, Response, Cookie

app = FastAPI()

@app.get('/set_cookie')
def set_cookie(val: str, response: Response) -> dict:
    response.set_cookie(key="suti", value=val, httponly=True)
    return {'Message': 'suti megváltoztatva: ' + val}


@app.get('/authtest')
def authtest(Cookie1: str | None = Cookie(default=None)) -> dict:
    if Cookie1 != 'ADMiN':
        raise HTTPException(status_code=401, detail='Unauthorized')
    return {'Message': 'Welcome home!'}
```

Gyakorlatban több validáció van, több mindenre oda kell figyelni (backend)...

Köszönöm a figyelmet!