

# Python

7. gyakorlat

Strohmayer Ádám

# Agenda

- Flask
- FastAPI
- Django

# Flask

## **Minimális webfejlesztési keretrendszer!**

Minimalizmusa miatt flexibilis – sok döntést hagy a fejlesztőnek.

## **Jinja2 sablonmotorral HTML oldalakat lehet vele renderelni!**

Flaskben a `render_template` metódussal átadhatóak változók a HTML oldalnak!

HTML oldalon belül `{{data}}` formában hivatkozhatók változók, `{% ... %}` formában vezérlési szerkezetek is megadhatóak!

**Jinja2 működése:** <https://jinja.palletsprojects.com/en/stable/>

# Flask példa

```
from flask import Flask, render_template

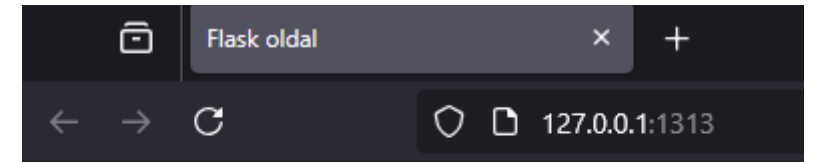
app = Flask("flaskpelda")
#futtatás: flask run, vagy python flask_test.py
#fontos a helyes mappaszerkezet!

@app.get("/") #alap oldalra (index.html) vonatkozó kód
def root():
    flask_title = "Flask oldal"
    lst = [i for i in range(5)]
    #ezeket a változókat fogja átadni a HTML oldal számára!
    return render_template("index.html",
                           page_name = flask_title,
                           nums = lst
                           ) #a kulccsal lehet hivatkozni ott rájuk!

if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0', port=1313)
```

# Flask példa – Jinja2 template

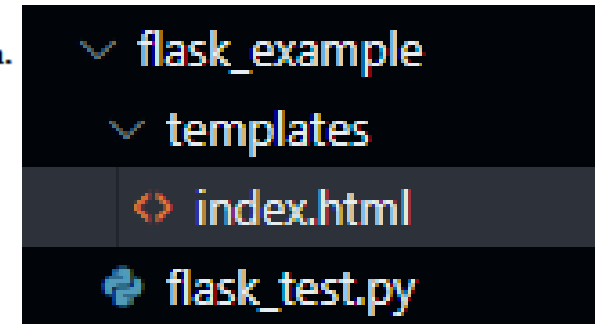
```
<!DOCTYPE html>
<html lang="en">
<head>
    ...
    <title>{{page_name}}</title>
</head>
<body>
    {% if (nums | length) % 2 == 0 %}
        <h1>A számok száma páros.</h1>
    {% else %}
        <h1>A számok száma páratlan.</h1>
    {% endif %}
    <ul>
        {% for num in nums %}
            <li>{{ num }}</li>
        {% endfor %}
    </ul>
    <b>Összesen: {{(nums | length)}} darab szám.</b>
</body>
</html>
```



**A számok száma páratlan.**

- 0
- 1
- 2
- 3
- 4

Összesen: 5 darab szám.



**Fontos a helyes mappaszerkezet!**

# 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!**

# Starlette

- **ASGI alapú webes keretrendszer (uvicorn alapon)**
- Protokollokkal kommunikál a háttérben (WebSocket, HTTP)
- FastAPI alapját képező keretrendszer (+Pydantic)



<https://www.starlette.dev>

# Django

**Magas szintű, full-stack Python webfejlesztési keretrendszer!**  
Nagyobb webalkalmazások készítésére használatos.

**Összetettebb** – kell ismerni hozzá a szerver- és kliensoldali webfejlesztési technikákat.

**Beépített biztonsági, ORM, jogosultságkezelő megoldásokkal!**

**Bővebben Djangoról:** <https://www.djangoproject.com/>



# Mi történt eddig?

- Bevezettük a Flask keretrendszert.
- Bevezettük a FastAPI keretrendszert.
- Beszéltünk a FastAPI keretrendszer működéséről.
- Megemlítettük a Django fontosabb tulajdonságait.

Köszönöm a figyelmet!