## Észrevételek az első feladathoz

A Python beépített SimpleHTTPRequestHandler osztálya hasznos lehet gyors teszteléshez vagy fájlok ideiglenes kiszolgálásához, de komolyabb webes alkalmazásokhoz számos korlátja miatt nem ajánlott. Ezen korlátok miatt hasznos lehet inkább keretrendszereket használni, mint a **Django** vagy a **Flask**:

**Biztonság**

A SimpleHTTPRequestHandler csak minimális biztonsági ellenőrzéseket végez, nem véd például könyvtárbejárási támadások ellen, és nem támogatja a HTTPS-t. Ezzel szemben a **Django** és a **Flask** beépített vagy könnyen integrálható biztonsági funkciókat kínálnak, mint például CSRF-védelem, HTTPS támogatás, felhasználókezelés és jogosultságok.

**Teljesítmény és skálázhatóság**

A beépített handler nem képes több egyidejű kapcsolat kezelésére, azaz egyszerre csak egy kapcsolat van, és semennyire sem optimalizált nagy forgalomra. A Django és a Flask viszont könnyen integrálhatók nagy teljesítményű szerverekkel (pl. Gunicorn, uWSGI, Nginx), így alkalmasak éles környezetben való futtatásra is. Django esetén elmondható, hogy már az alapbeállítás is (kis sqlite adatbázisok) könnyen kiszolgálnak kis és közepes project-eket, nagyjából napi 100 ezer hívásig.

**Funkcionalitás**

A SimpleHTTPRequestHandler csak GET és HEAD kéréseket kezel, míg a Django teljes körű HTTP támogatást nyújtanak (POST, PUT, DELETE stb.), valamint lehetőséget adnak adatbázis-kezelésre, sablonmotor használatára, URL routingra és REST API-k építésére.

Másképpen csak kiszolgál statikus file-okat.

Note:

A HEAD request (HTTP HEAD kérés) olyan, mint egy GET kérés, de a szerver csak a válasz fejléceit küldi vissza, a tartalmat (body-t) nem.

**Többszálúság és párhuzamos feldolgozás**

Alapértelmezetten a handler csak egyetlen kérést tud kezelni egyszerre. A modern keretrendszerek viszont támogatják a párhuzamos feldolgozást, ami elengedhetetlen a mai webes alkalmazásoknál.

**Naplózás és hibakezelés**

A beépített megoldás minimális naplózást kínál, míg a Django és a Flask részletes, testreszabható naplózási és hibakezelési lehetőségeket biztosítanak, ami segíti a fejlesztést és a hibakeresést.