

2024.11.18.

Órai Feladat

Határidő:

55 perc

1. JSON mentése és betöltése fájlból

Készíts egy programot, amely egy szótárt ment JSON formátumban egy fájlba, majd betölti és kiírja a tartalmát.

Minta adat:

```
{"nev": "Anna", "kor": 25, "varos": "Budapest"}
```

2. JSON fájl módosítása

Tölts be egy JSON fájlt, adj hozzá egy új kulcs-érték párt ("ország": "Magyarország"), majd mentsd vissza.

3. Számok rendezése JSON fájlból

Olvass be egy JSON fájlt, amely egy listát tartalmaz számokkal, rendezd őket növekvő sorrendbe, majd írd vissza a fájlba.

Minta JSON:

```
[12, 5, 7, 3, 18]
```

4. Legnagyobb és legkisebb érték keresése JSON fájlból

Egy JSON fájlban egy lista van tárolva. Töltsd be a fájlt, és írd ki a lista legnagyobb és legkisebb elemét.

5. Összeg és átlag JSON adatokból

Egy JSON fájl tartalmazza egy osztály tanulóinak pontszámait egy listában. Számítsd ki a pontszámok összegét és átlagát.

Minta JSON:

```
[85, 90, 78, 92, 88]
```

6. Hibaellenőrzés JSON fájl olvasásánál

Írj egy programot, amely megpróbál betölteni egy JSON fájlt. Ha a fájl nem létezik, hozzon létre egy új fájlt alapértelmezett adatokkal.

Alapértelmezett adatok:

```
{"uzenet": "Ez egy alapértelmezett JSON fájl."}
```

7. JSON fájl validáció

Készíts egy programot, amely egy JSON fájl tartalmát betölti, és ellenőrzi, hogy érvényes-e a JSON szerkezet. Hibás szerkezet esetén adjon vissza hibaüzenetet.

8. Adatszűrés JSON fájlból

Egy JSON fájl egy listát tartalmaz szótárakkal. Írd ki a szótárakat, amelyekben a "kor" értéke nagyobb, mint 30.

Minta JSON:

```
[
  {"név": "Anna", "kor": 25, "város": "Budapest"},
  {"név": "Péter", "kor": 35, "város": "Debrecen"},
  {"név": "László", "kor": 40, "város": "Szeged"},
  {"név": "Júlia", "kor": 28, "város": "Pécs"}
]
```

9. Szövegfájl JSON formátumra alakítása

Egy szavak_50.txt fájl minden sora egy szót tartalmaz. Írj egy programot, amely ezeket a szavakat JSON formátumba alakítja, és menti egy fájlba.

10. Pontszámok kiértékelése JSON-ból Try-Except-tel

```
pontszamok.json -> [85, 90, 78, 92, 88, 70, 95]
hibas_pontszamok.json -> [85, 90, 78, "ez nem is szám", 92, 88, 70, 95]
```

Egy JSON fájl tartalmazza a pontszámokat egy listában. Írj egy programot, amely:

1. Ellenőrzi, hogy a fájl létezik-e.
 2. Ha nem létezik, hibakezeléssel létrehoz egy új fájlt.
 3. Ha a fájl létezik, kiírja a legmagasabb és legalacsonyabb pontszámot, valamint az átlagot.
-

11. Összetett feladat

Írj egy programot, amely egy JSON fájlba ment egy dolgozók listáját (nev, pozicio, fizetes kulcsokkal), majd:

1. Rendezze őket a fizetés alapján növekvő sorrendben.
2. Írja ki a legmagasabb és legalacsonyabb fizetést kapó dolgozót.
3. Számítsa ki az átlagfizetést.