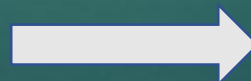


Python tanfolyam

8. alkalom

Gyakorló feladat – a megoldás

- Készíts programot, mely a Super Mario-ból is ismert két oldalú lépcsőt írja ki egy fájlba.
 - A felhasználótól a program indulásakor kérjük be, hogy hány lépcsőfokot szeretne, és azt is, hogy mi legyen a kimeneti fájl neve.
 - Az eredmény fájlt mindig írjuk felül.
 - A lépcsőfokok a # karakterrel legyenek ábrázolva.
 - Például, ha a 4 fokos lépcsőt kell készíteni, és a kimeneti fájl neve out.txt, akkor az out.txt tartalma:



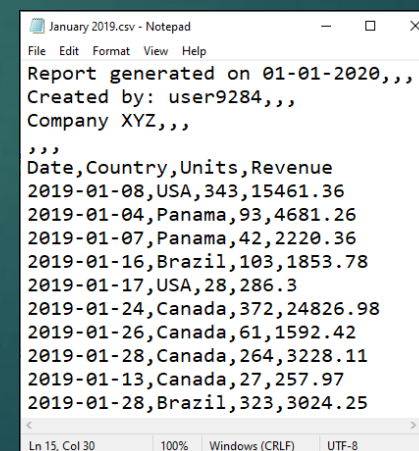
1	#	#
2	##	##
3	###	###
4	####	####

Struktúrált szöveges fájlok

- Az egyszerű szöveges fájloknak nincs semmilyen kötött formátumuk, a programozó dönt a feldolgozás módjáról.
 - Ez értelemszerűen megteremti az “ahány ház annyi szokás” problémát
- Szükség van valamilyen struktúrára, amit az ipar szabványosnak fogad el.

- XML
- JSON
- CSV
- YAML

YAML	XML	JSON
<pre>apis: - name: login port: 8080 - name: profile port: 8090</pre>	<pre><apis> <api> <name>login</name> <port>8080</port> </api> <api> <name>profile</name> <port>8090</port> </api> </apis></pre>	<pre>{ "apis": [{ "name": "login", "port": 8080 }, { "name": "profile", "port": 8090 }] }</pre>



```
January 2019.csv - Notepad
File Edit Format View Help
Report generated on 01-01-2020,,,
Created by: user9284,,,
Company XYZ,,,
,,,
Date,Country,Units,Revenue
2019-01-08,USA,343,15461.36
2019-01-04,Panama,93,4681.26
2019-01-07,Panama,42,2220.36
2019-01-16,Brazil,103,1853.78
2019-01-17,USA,28,286.3
2019-01-24,Canada,372,24826.98
2019-01-26,Canada,61,1592.42
2019-01-28,Canada,264,3228.11
2019-01-13,Canada,27,257.97
2019-01-28,Brazil,323,3024.25
```

Melyik struktúrát használjuk?

- Az évek során az XML (Extensible Markup Language) alapú fájlokról (kommunikációról) áttértünk a JSON (JavaScript Object Notation) alapúra.
 - Általában adatot írunk le vele
- A YAML nagyon elterjedt a konfigurációs fájlok világában.
- A CSV alkalmas arra, hogy egyszerű táblázatos formátumú (flat) adatot tároljunk vele.
 - Az Excel egyik alapvető adatimport/export formátuma.

JSON

- JSON (JavaScript Object Notation) egy szabványosított formátum.
 - <https://www.rfc-editor.org/rfc/rfc8259> (RFC - Request For Comments)
- Mivel a JavaScript nyelv natívan tudja kezelni (be van építve a nyelvbe) ezt a formátumot, a modern frontendek (weboldalak) ezt használják kommunikációra a backend (szerver oldali kód) megoldásokkal.
- Azok a backend programozási nyelvek kerültek előtérbe, melyekben nagyon könnyű a JSON formátum kezelése.
 - NodeJS (JavaScript alapú nyelv)
 - Python

JSON formátum

- Alapvetően kulcs-érték párok, ahol az érték lehet
 - Egyszerű típus: Szöveg, Szám, Logikai érték, null
 - Objektum: kulcs-érték párok.
 - Tömb: Egyszerű típusú elemek, Objektumok vagy tömböket tartalmazhat.
- Egy objektum egy szintjén belül a kulcsok egyediek.

```
1 {  
2   "string": "Hi",  
3   "number": 2.5,  
4   "boolean": true,  
5   "null": null,  
6   "object": { "name": "Kyle", "age": 24 },  
7   "array": ["Hello", 5, false, null, { "key": "value", "number": 6 }],  
8   "arrayOfObjects": [  
9     { "name": "Jerry", "age": 28 },  
10    { "name": "Sally", "age": 26 }  
11  ]  
12 }  
13
```

Hogyan működik mindez Pythonban?

- A dictionary (dict) adatszerkezet lehetőséget ad arra, hogy a JSON formátumhoz nagyon hasonló adatszerkezetet kezeljünk.

- Létrehozása

```
my_dict_1 = dict() # egy üres dict-et hoz létre
my_dict_2 = {} # egy üres dict-et hoz létre
my_dict_3 = dict(name="Kovács József", age=30)
my_dict_4 = {"name": "Kovács József", "age":30)
#Mind my_dict_3 és my_dict_4 ugyanolyan dict lesz.
#A memóriában természetesen mind a 4 dict külön helyen lesz tárolva.
```

Dictionary kezelése

```
my_dict = {}

my_dict['name'] = "Kovács József" # a dict-hez felveszünk egy kulcsot egy értékkel

if 'name' in my_dict: #Ha a name kulcs benne van a dict-ben, akkor ...
    pass

v = my_dict.pop("name", None) #Eltávolítja a dict-ből az adott kulcsot.

v2 = my_dict['name']
#v2 értéke ugyanoda fog mutatni, mint ahova a my_dict['name'] mutat.
#Ha még sincs benne, akkor az hiba!

v3= {"x": [1,2,3,4, {"a": "Hello"}, [1,2,3]], "y": "Szia", z: {"b": False} }
```


JSON és dict()

- Könnyen belátható, hogy a JSON és a dict() nagyon hasonlítanak egymáshoz.
- Egy JSON formátumú szöveg egyszerűen konvertálható dict()-be
 - Ezáltal könnyen fel lehet dolgozni Python-ban
 - Szövegből → Adatszerkezet: De-szerializáció (deserialization)
- Egy dict()-ben lévő adatot egyszerűen konvertálhatunk JSON formátumú szöveggé
 - Ezáltal könnyen lementhető vagy átküldhető a hálózaton
 - Adatszerkezetből → Szöveg: Szerializáció (Serialization)

JSON és dict() konvertálások

A json package-t kell hozzá használni

```
import json
```

Ha az adat már egy változóban van:

```
v3= {"x": [1,2,3,4, {"a": "Hello"}, [1,2,3]], "y":"Szia", z: {"b": False} }  
x1 = json.dumps(v3) # A v3 dict-et JSON stringgé konvertálja  
v4 = json.loads(x1)  
  
# Az x1 stringet JSON-ként próbálja értelmezni, és betölti v4-be a dict-et.
```

Ha az adat fájlban van, vagy fájlba kell kiírni:

```
#Fájl megfelelő megnyitása után:  
json.dump(v3, json_file) #Fájlba írja a v3 dict tartalmát  
data = json.load(json_file)  
  
#A data változóba dict-ként bekerül a JSON fájl tartalma
```

Gyakorló- és házi feladatok

- Készítsünk programot, egy input.txt fájlban lévő szavakat kezdőbetű szerint megszámol, és az eredményt egy dict()-ben kiírja a standard outputra.
 - A bemeneti fájlban soronként egy szó legyen.
 - Példa kimenet: {a:3, g:2, z: 1}
- Fejlesszük tovább a programot, hogy a végeredményt egy output.json nevű fájlba írja ki, JSON formátumban
- Fejlesszük tovább a programot, hogy a kimenet ne csak a darabszámot tartalmazza, hanem a szavakat is.
 - A kimeneti formátum a következő legyen
 - {"a":{"darab":3, "szavak": ["alma", "aki", "az"]}, "z": {"darab":1, "szavak": ["zászló"]}}

Gyakorló- és házi feladatok (2)

- Fejlesszük tovább úgy a programot, hogy a kimeneti fájlban a szavak ábécé sorrendben legyenek a szavak tömbben.
- Írjunk programot, ami az előző feladatban lévő output.json fájlt képes beolvasni. Írjuk ki a standard outputra, hogy összesen hány szót találtunk az eredeti input.txt-ben.
- Fejlesszük tovább a programot úgy, hogy írja ki a szavakat is, ábécé sorrendben.