Χρήση Προηγμένων Βιβλιοθηκών Python για Διαχείριση Δεδομένων

Αρχές Γλωσσών Προγραμματισμού και Μεταφραστών

Γ. Γαροφαλάκης, Σ. Σιούτας, Π. Χατζηδούκας

Data Persistence

- **JSON module**: Ανάγνωση και αποθήκευση δομημένων δεδομένων.
- **Openpyxl**: Χειρισμός αρχείων Excel.
- **Αρχεία .CSV** – ανάγνωση και εγγραφή
- **Pandas**: Δομημένη ανάλυση δεδομένων σε πίνακες.
- **NumPy**: Μαθηματικοί υπολογισμοί σε πίνακες.
- **Pickle** – **Shelve** Lilbraries

Data Persistence

JSON

- Μορφή δομημένων δεδομένων.
- JSON: JavaScript Object Notation.
- Ελαφριά μορφή ανταλλαγής δεδομένων.
- Πολύ διαδεδομένο σε Web εφαρμογές και APIs.
- Διαβάζεται και γράφεται εύκολα από ανθρώπους και μηχανές.

JSON

Το JSON αποτελεί ένα ανοικτό πρότυπο το οποίο χρησιμοποιεί κείμενο αναγνώσιμο από τον άνθρωπο με σκοπό τη μετάδοση πληροφοριακών αντικειμένων δεδομένων. Η συντομογραφία του JavaScript Object Notation είναι μια μορφή κειμένου που βασίζεται σε ένα υποσύνολο της γλώσσας προγραμματισμού JavaScript, αλλά είναι εντελώς ανεξάρτητη από την οποιαδήποτε γλώσσα προγραμματισμού. Χρησιμοποιείται ως μια εναλλακτική λύση της XML για τη μετάδοση δεδομένων μεταξύ του εξυπηρετητή και των εφαρμογών διαδικτύου.

Το JSON αποτελείται από δυο δομές. Η πρώτη δομή είναι μια συλλογή από ζεύγη ονομάτων /τιμών, στις περισσότερες γλώσσες προγραμματισμού αυτό μπορεί να γίνει αντιληπτό ως ένα αντικείμενο, μια δομή ένα λεξικό ή μια λίστα κλειδιών.

Η δεύτερη δομή είναι μια ταξινομημένη λίστα τιμών, στις περισσότερες γλώσσες προγραμματισμού αυτό μπορεί να γίνει αντιληπτό ως ένας πίνακας, μια λίστα ή μια ακολουθία.

JSON

- **Γιατί να χρησιμοποιήσουμε JSON;**
- Γρήγορη αποθήκευση δομημένων δεδομένων.
- Ιδανικό για ανταλλαγή πληροφορίας μεταξύ εφαρμογών.
- Υποστηρίζεται εγγενώς από πολλές γλώσσες προγραμματισμού.

Βασική Δομή JSON

- αποτελείται από:
  - **Αντικείμενα** (Object): σύνολο ζευγών "κλειδί:τιμή".
  - **Λίστες** (Array): συλλογές τιμών.
- Οι τιμές μπορεί να είναι:
  - String, αριθμοί, λογικές τιμές, null, αντικείμενα, λίστες.

Παράδειγμα JSON Αντικειμένου

```
{
  "όνομα": "Γιάννης",
  "ηλικία": 30,
  "ενεργός": true,
  "δεξιότητες": ["Python", "Data Analysis"]
}
```

Βιβλιοθήκη json

```
j_data = json.dumps(data) # data to json
data = json.loads(j_data) # json to  data
```

Python	JSON Equivalent
dict	object
list, tuple	array
str	string
int, float, int	number
True	true
False	false
None	null

Βιβλιοθήκη json - Μετατροπή λεξικού σε json

```
import json

person_dict = {'name’: John', 'age’: 18, 'children': None}

person_json = json.dumps(person_dict)

# Output: {"name": ”John", "age": 18, "children": null}
print(person_json)
```

Βιβλιοθήκη json – ανάγνωση από αρχείο

```
{
"name": "Bob", "languages": ["English", "Fench"]
}
```

```
import json

with open('path_to_file/person.json') as f:
        data = json.load(f)

# Output: {'name': 'Bob', 'languages': ['English', 'Fench']}

print(data)
```

Βιβλιοθήκη json – αποθήκευση σε αρχείο

```
import json

person_dict = {"name": "Bob", "languages": ["English", ”Greek"],
"married": True, "age": 35 }

with open('person.txt', 'w') as json_file:
        json.dump(person_dict, json_file)
```

```
new_data = {
    "όνομα": "Άννα",
    "ηλικία": 25
}

with open('new_data.json', 'w') as f:
    json.dump(new_data, f)
```

Data Persistence

Η openpyxl είναι μια δημοφιλής βιβλιοθήκη της Python.

- Επιτρέπει
  την **ανάγνωση**, **επεξεργασία** και **δημιουργία** αρχείων **Microsoft Excel** μορφής .xlsx.

Υποστηρίζει:

•Γραφή και ανάγνωση δεδομένων.

•Διαμόρφωση κελιών (formatting).

•Διαγράμματα (charts), τύπους (formulas), φίλτρα και άλλα.

OPENPYXL

- **Εγκατάσταση της Openpyxl :** pip install openpyxl

- **Άνοιγμα υπάρχοντος αρχείου Excel**

```
from openpyxl import load_workbook

wb = load_workbook('αρχείο.xlsx')
sheet = wb.active
```

- **Δημιουργία νέου αρχείου Excel**

```
from openpyxl import Workbook

wb = Workbook()
sheet = wb.active
sheet['A1'] = "Γεια σου κόσμε!"
wb.save('νέο_αρχείο.xlsx')
```

- **Ανάγνωση δεδομένων από κελιά**

```
print(sheet['A1'].value)
print(sheet.cell(row=1, column=2).value)
```

Περισσότερα εδώ:!!!
https://openpyxl.readthedocs.io/en/stable/

Αρχεία CSV 1/2

- Τα αρχεία CSV χρησιμοποιούνται για την αποθήκευση μεγάλου αριθμού μεταβλητών ή δεδομένων - το περιεχόμενο αποθηκεύεται σε απλό κείμενο.

- Το module CSV είναι μια ενσωματωμένη λειτουργία που επιτρέπει στην Python να αναλύει αυτούς τους τύπους αρχείων.

Αρχεία CSV 2/2

- Το κείμενο μέσα σε ένα αρχείο CSV παρατίθεται σε σειρές και κάθε ένα από αυτά έχει στήλες, όλες διαχωρισμένες με κόμματα.

- Κάθε γραμμή του αρχείου είναι μια σειρά ενώ τα κόμματα χρησιμοποιούνται για τον ορισμό και τη διαίρεση κελιών.

Ανάκτηση από αρχείο κειμένου

CSV

```
name,department,birthday month
John Smith,Accounting,November
Erica Meyers,IT,March
```

employee_birthday.txt

```
import csv

with open('employee_birthday.txt', mode='r') as csv_file:
        csv_reader = csv.DictReader(csv_file)
         line_count = 0
         for row in csv_reader:
                 if line_count == 0:
                         print(f'Column names are {", ".join(row)}’)
                         line_count += 1
                 print(f'\t{row["name"]} works in the {row["department"]}
department, and was born in {row["birthday month"]}.’)
                  line_count += 1
        print(f'Processed {line_count} lines.')
```

```
Column names are name, department, birthday month
John Smith works in the Accounting department, and was born in November. Erica Meyers works in the IT department, and was born in March.
Processed 3 lines.
```

Quoting CSV files

- ***csv.QUOTE_ALL*** - Quote everything, regardless of type.
- ***csv.QUOTE_MINIMAL*** - Quote fields with special characters
- ***csv.QUOTE_NONNUMERIC*** - Quote all fields that are not integers or floats
- ***csv.QUOTE_NONE*** - Do not quote anything on output

Εγγραφή σε αρχείο κειμένου

CSV

```
name,address,date joined
john smith,1132 Anywhere,Jan 4
 erica meyers,1234 Smith,March 2
```

'employee_file.csv

```
import csv

with open('employee_file.csv', mode='w') as employee_file:
employee_writer = csv.writer(employee_file, delimiter=',', quotechar='"',
quoting=csv.QUOTE_MINIMAL)

employee_writer.writerow(['John Smith', ‘Anywhere', ‘January'])
employee_writer.writerow(['Erica Meyers', ‘1234 Smith', 'March'])
```

Pandas

- Βιβλιοθήκη Python για διαχείριση δομημένων δεδομένων.
- Παρέχει Series και DataFrame.
- Καθαρισμός και ανάλυση δεδομένων.

Pandas - Δημιουργία Series

- import pandas as pd
- s = pd.Series([1, 3, 5, 7, 9])
- print(s)

// Μονοδιάστατη δομή δεδομένων με δείκτες.

Pandas - Δημιουργία DataFrame

- **DataFrame** είναι η βασική δομή δεδομένων στο Pandas.
- Παρόμοιο με πίνακα ή φύλλο εργασίας σε Excel, αλλά με πρόσθετες δυνατότητες ανάλυσης.
- **Δομή 2 διαστάσεων**: γραμμές και στήλες.
- **Στήλες**: Διάφοροι τύποι δεδομένων (αριθμοί, κείμενο, ημερομηνίες).
- **Γραμμές**: Παρουσιάζουν τα δεδομένα ως σειρές.

Pandas - Δημιουργία DataFrame

- import pandas as pd

- # Δημιουργία DataFrame από λεξικό
- data = {'Όνομα': ['Γιάννης', 'Άννα', 'Μαρία'], 'Ηλικία': [28, 22, 31], 'Πόλη': ['Αθήνα', 'Θεσσαλονίκη', 'Πάτρα']}
- df = pd.DataFrame(data)

- # Εμφάνιση του DataFrame
- print(df)

```
  Όνομα  Ηλικία        Πόλη
0  Γιάννης     28      Αθήνα
1    Άννα      22  Θεσσαλονίκη
2   Μαρία      31      Πάτρα
```

Pandas - Ανάγνωση CSV

- df = pd.read_csv(data.csv')
- print(df.head())

// Εισαγωγή δεδομένων από αρχείο CSV.

Pandas - Φιλτράρισμα δεδομένων

- new_df = df[df['Ηλικία'] > 25]
- print(new_df)

// Επιλογή δεδομένων βάσει κριτηρίου.

Pandas - Ομαδοποίηση (GroupBy)

- grouped = df.groupby('Κατηγορία').mean()
- print(grouped)

// Μέσοι όροι κατά κατηγορία.

NumPy

- Βιβλιοθήκη Python για αριθμητικούς υπολογισμούς.
- Παρέχει το ndarray για αποδοτική αποθήκευση δεδομένων.
- Υψηλή απόδοση

NumPy - Δημιουργία Πίνακα NumPy

- import numpy as np
- arr = np.array([1, 2, 3])
- print(arr)

μετατρέπει τη λίστα [1, 2, 3] σε έναν πίνακα NumPy.

[1 2 3]

// Μονοδιάστατος πίνακας.

NumPy - Ιδιότητες Πίνακα

- print(arr.shape)
- print(arr.dtype)

// Μέγεθος και τύπος στοιχείων.

NumPy - Μηδενικά και Τυχαίοι Αριθμοί

- np.zeros((2,3))

```
array([[0., 0., 0.],
       [0., 0., 0.]])
```

- np.random.rand(2,2)

```
array([[0.6353352 , 0.21235625],
       [0.74138876, 0.32301277]])
```

// Πίνακες ειδικών τιμών - όταν θέλουμε να δημιουργήσουμε δεδομένα για δοκιμές ή για μαθηματικούς υπολογισμούς, χωρίς να χρειάζεται να τα παρέχουμε χειροκίνητα

NumPy - Πράξεις Πινάκων

- a = np.array([1,2,3])
- b = np.array([4,5,6])
- print(a + b)

// Προσθήκη στοιχείων μεταξύ πινάκων.

NumPy - Στατιστικές Πράξεις

- print(np.mean(arr))
- print(np.std(arr))

// Υπολογισμός μέσου όρου και τυπικής απόκλισης.

Pickle library

- Μια βιβλιοθήκη της Python για την αποθήκευση και ανάγνωση αντικειμένων Python σε μορφή byte-stream.
- Επιτρέπει την **serialization** (μετατροπή) και **deserialization** (επαναφορά) αντικειμένων Python, κάνοντάς τα εύκολα μεταφερόμενα και αποθηκεύσιμα.
- **Κύρια Χρησιμότητα:**
- **Αποθήκευση** της κατάστασης ενός προγράμματος για μελλοντική χρήση.
- **Μεταφορά** αντικειμένων Python μεταξύ διαφορετικών εφαρμογών ή συστημάτων.
- Αποθήκευση **μοντέλων μηχανικής μάθησης** και άλλων σύνθετων δεδομένων.

Τύποι δεδομένων

- Booleans
- Integers
- Floats
- Complex numbers
- (normal and Unicode) Strings
- Tuples
- Lists
- Sets
- Dictionaries with picklable objects

Παράδειγμα pickle σε λίστα 1/2

```
import pickle

a = ['test value','test value 2','test value 3’]

file_Name = "testfile"
# open the file for writing
fileObject = open(file_Name,'wb')
```

Παράδειγμα pickle σε λίστα 2/2

```
# this writes the object a to the # file named 'testfile'
pickle.dump(a,fileObject)

# here we close the fileObject
 fileObject.close()
# we open the file for reading
fileObject = open(file_Name,'r’)
# load the object from the file into var b
b = pickle.load(fileObject)
```

Παράδειγμα pickle σε λεξικό

```
import pickle

emp = {1:"A",2:"B",3:"C",4:"D",5:"E"}
pickling_on = open("Emp.pickle","wb")
pickle.dump(emp, pickling_on)
pickling_on.close()
```

Παράδειγμα unpickle σε λεξικό

```
pickle_off = open("Emp.pickle","rb")

emp = pickle.load(pickle_off)

print(emp)
```

Πλεονεκτήματα- Μειονεκτήματα

+

- Βοηθά στην αποθήκευση περίπλοκων δεδομένων.
- Πολύ εύκολο στη χρήση, δεν απαιτεί πολλές γραμμές κώδικα

-

- Δυσκολία με μη python προγράμματα στο χειρισμό αντικειμένων Pickle
- Κίνδυνοι ασφαλείας από δεδομένα που προέρχονται από κακόβουλες πηγές.

Shelve library

- Μια βιβλιοθήκη της Python που επιτρέπει την αποθήκευση και ανάγνωση αντικειμένων Python σε αρχεία, με τον ίδιο τρόπο που δουλεύουμε με λεξικά.
- Χρησιμοποιεί μια βάση δεδομένων τύπου **key-value** (κλειδί-τιμή) για την αποθήκευση των δεδομένων.

- **Κύρια Χρησιμότητα:**
- Αποθήκευση αντικειμένων Python (όπως λιστών, λεξικών, κ.λπ.) σε αρχεία, επιτρέποντας εύκολη πρόσβαση και επαναφόρτωσή τους αργότερα.
- Ιδανικό για την αποθήκευση δεδομένων που μπορεί να είναι πολύ μεγάλα για να τα κρατήσουμε στη μνήμη.

Δημιουργία shelve

```
import shelve

with shelve.open('test_shelf.db') as
s: s['key1'] = {
        'int': 10,
        'float': 9.5,
        'string': 'Sample data',
}
```

Ανάγνωση δεδομένων

```
import shelve

with shelve.open('test_shelf.db') as s:
        existing = s['key1']

print(existing)
```

Εγγραφή δεδομένων

```
import shelve
with shelve.open('test_shelf.db') as s:
        print(s['key1'])
        s['key1']['new_value'] = 'test write'
with shelve.open('test_shelf.db', writeback=True)
as s:
        print(s['key1'])
```