

JSON

- JSON = Javascript Object Notation
- Standardisiertes Format zum Speichern von Daten/Informationen
- Findet in vielen Programmiersprachen Gebrauch
- Relativ einfach lesbar für das menschliche Auge
- Im Vergleich zu CSV:
 - JSON unterstützt Datentypen (z.B. bool, string etc.)
 - JSON bietet Verschachtelung
 - JSON Dateien sind üblicherweise größer
- Use Cases
 - Konfigurationsdateien
 - Datenbanken
 - APIs

Syntax von JSON-Dateien

- Ähnlich wie Python dict

```
{  
    "name": "Karim",  
    "age": 40,  
    "hasChildren": true,  
    "other": null  
}
```

- Minimale Unterschiede bei Datentypen

Datentypen von Python vs. JSON

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

Komplette Syntax: <https://www.json.org/json-en.html>

JSON in Python

- Python unterstützt JSON nativ:
`import json`
- Schreiben eines Python dict in json file (=serialization)
 with open(filename, "w") as file:
 json.dump(data, file)
- Lesen einer json file in Python (=deserialization)
 with open(filename, "r") as file:
 data = json.load(file)
- Nützlicher Parameter für .dump:
 - indent -> rückt die Daten in der json file ein (int-value)

Übung

- Siehe 07 – Übung.py
- Zusatz:
 - Finde in folgender Liste eine interessante öffentliche API:
 - <https://github.com/public-apis/public-apis>
 - Greife auf die API zu und lies die Daten ein
 - Überlege dir Fragestellungen bzw. Use Cases zu den Daten