CSV & JSON

Python 3 · kompakte Grundlagen mit direkt nutzbaren Beispielen

1 | CSV oder JSON?

Format	Geeignet
CSV	flache Tabellen
JSON	verschachtelte Strukturen
CSV	Excel/Tabellenimport
JSON	Konfiguration/API
beide	Text, nicht Binär

2 | CSV lesen

```
import csv
with open("werte.csv", newline="", encoding="utf-8") as f:
    reader = csv.reader(f, delimiter=";")
    for zeile in reader:
        print(zeile)
```

3 | DictReader

- Erste Zeile dient als Spaltenname.
- Jede Zeile wird Dictionary.

```
import csv
with open("werte.csv", newline="", encoding="utf-8") as f:
    for zeile in csv.DictReader(f, delimiter=";"):
        print(zeile["temperatur"])
```

4 | CSV schreiben

```
import csv
with open("out.csv", "w", newline="", encoding="utf-8") as f:
    writer = csv.writer(f, delimiter=";")
    writer.writerow(["zeit", "wert"])
    writer.writerow(["10:00", 21.5])
```

5 | CSV-Datentypen

- CSV liefert zunächst Strings.
- Zahlen gezielt umwandeln.
- Dezimaltrennzeichen/fehlende Werte definieren.

```
text = zeile["temperatur"]
wert = float(text.replace(",", "."))
```

6 | JSON lesen

```
import json
with open("config.json", encoding="utf-8") as f:
    config = json.load(f)
print(config["drucker"]["name"])
```

CSV & JSON

Python 3 · kompakte Grundlagen mit direkt nutzbaren Beispielen

7 | JSON schreiben

```
import json
daten = {"name": "N4", "online": True}
with open("config.json", "w", encoding="utf-8") as f:
    json.dump(daten, f, ensure_ascii=False, indent=2)
```

8 | JSON-Typen

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

9 | Text statt Datei

```
import json
text = '{"name": "PLA", "temp": 205}'
daten = json.loads(text)
neu = json.dumps(daten, ensure_ascii=False)
```

10 | Validierung

- Erwartete Spalten/Schlüssel prüfen.
- Fehlende Werte eindeutig behandeln.
- Datentypen und Wertebereiche kontrollieren.
- Ungültige Zeilen mit Nummer protokollieren.
- Originaldatei nicht vorschnell überschreiben.

11 | Typische Fehler

Problem	Prüfen
Umlaute falsch	UTF-8/ensure_ascii
leere CSV-Zeilen	newline=''
KeyError	Spaltenname/Schlüssel
JSONDecodeError	Syntax/Komma
Zahl bleibt Text	int/float

12 | Mini-Übung: Profil

Aufgabe

Speichere ein PLA-Profil als formatiertes JSON.

Musterlösung

```
import json
profil = {"material": "PLA", "duese": 205, "bett": 55}
with open("pla.json", "w", encoding="utf-8") as f:
    json.dump(profil, f, ensure_ascii=False, indent=2)
```