# 26. JSON va CSV bilan ishlash




Dasturlar ma'lumot almashadi: sayt serverdan, Excel jadvaldan, bir dastur boshqasidan. Buning uchun ikki eng mashhur matn formati bor: **JSON** (ichma-ich tuzilmalar — API, sozlamalar) va **CSV** (jadvallar — Excel, hisobotlar). Python ikkalasi uchun ham tayyor modulga ega.

## 1. JSON nima?

JSON (JavaScript Object Notation) — Python lug'ati va ro'yxatiga juda o'xshash matn: kalitlar qo'shtirnoqda, qiymatlar — son, satr, `true/false`, `null`, ro'yxat yoki obyekt:

| Python | JSON |
|---|---|
| dict | object { } |
| list, tuple | array [ ] |
| str | string "..." |
| int, float | number |
| True / False | true / false |
| None | null |

## 2. dumps va loads — satr bilan

```
import json
talaba = {"ism": "Malika", "yosh": 15, "fanlar": ["matematika", "informatika"]
```

```
matn = json.dumps(talaba, ensure_ascii=False)
print(matn)
print(type(matn).__name__)
qayta = json.loads(matn)
print(qayta["fanlar"][1], qayta["faol"])
```

*Natija:*

```
{"ism": "Malika", "yosh": 15, "fanlar": ["matematika", "informatika"], "faol":
str
informatika True
```

`ensure_ascii=False` — o'zbekcha harflar `\u…` kodlarga aylanmasin; `indent=2` — chiroyli (o'qilishi oson) ko'rinish:

```
import json
print(json.dumps({"shahar": "Toshkent", "aholi": 3000000}, ensure_ascii=False,
print(json.dumps({"soz": "o'zbek"}))
```

*Natija:*

```
{
  "shahar": "Toshkent",
  "aholi": 3000000
}
{"soz": "o'zbek"}
```

# 3. dump va load — fayl bilan

```
import json
sozlamalar = {"til": "uz", "mavzu": "qorong'i", "shrift": 14}
with open("sozlamalar.json", "w", encoding="utf-8") as f:
    json.dump(sozlamalar, f, ensure_ascii=False, indent=2)

with open("sozlamalar.json", encoding="utf-8") as f:
    s = json.load(f)
print(s["mavzu"], s["shrift"] + 2)
```

*Natija:*

```
qorong'i 16
```

```
import json
json.loads("{'ism': 'Ali'}")
```

*Natija:*

```
json.decoder.JSONDecodeError: Expecting property name enclosed in double quote
```

JSON da satrlar faqat **qo'shtirnoqda** yoziladi — bittalik tirnoq xato (JSONDecodeError) beradi.

# 4. Ichma-ich JSON bilan ishlash

```
import json
javob = '{"sinf": "9-A", "talabalar": [{"ism": "Ali", "ball": 78}, {"ism": "Ma
d = json.loads(javob)
ballar = [t["ball"] for t in d["talabalar"]]
print(d["sinf"], "o'rtacha:", sum(ballar) / len(ballar))
eng = max(d["talabalar"], key=lambda t: t["ball"])
print("Eng yaxshi:", eng["ism"])
```

*Natija:*

```
9-A o'rtacha: 87.0
Eng yaxshi: Malika
```

# 5. CSV nima?

CSV (Comma-Separated Values) — har qator bir yozuv, ustunlar vergul bilan ajratilgan oddiy matn. Excel uni ocha oladi va saqlay oladi:

```
with open("ballar.csv", "w", encoding="utf-8") as f:
    f.write("ism,sinf,ball\nAli,9,78\nMalika,9,95\nSardor,10,88\n")
print(open("ballar.csv", encoding="utf-8").read())
```

*Natija:*

```
ism,sinf,ball
Ali,9,78
Malika,9,95
Sardor,10,88
```

# 6. csv.reader va csv.writer

Qo'lda `split(",")` qilish xavfli: qiymat ichida vergul bo'lsa ("Toshkent, Chilonzor") hammasi buziladi. `csv` moduli bunday holatlarni (qo'shtirnoq bilan) to'g'ri ishlaydi:

```
import csv
with open("manzil.csv", "w", newline="", encoding="utf-8") as f:
    w = csv.writer(f)
    w.writerow(["ism", "manzil"])
    w.writerow(["Ali", "Toshkent, Chilonzor"])
    w.writerows([["Vali", "Nukus"], ["Gani", "Xiva, Ichan qal'a"]])

print(open("manzil.csv", encoding="utf-8").read())
with open("manzil.csv", newline="", encoding="utf-8") as f:
    for qator in csv.reader(f):
        print(qator)
```

*Natija:*

```
ism,manzil
Ali,"Toshkent, Chilonzor"
Vali,Nukus
Gani,"Xiva, Ichan qal'a"

['ism', 'manzil']
```

```
['Ali', 'Toshkent, Chilonzor']
['Vali', 'Nukus']
['Gani', "Xiva, Ichan qal'a"]
```

> **Eslatma**
>
> CSV faylni yozishda `open(..., newline="")` beriladi — aks holda Windows'da qatorlar orasida bo'sh qator paydo bo'ladi.

# 7. DictReader va DictWriter

Ustunlarga nom bilan murojaat qilish qulayroq — har qator lug'at bo'lib keladi:

```
import csv
with open("ballar.csv", "w", newline="", encoding="utf-8") as f:
    w = csv.DictWriter(f, fieldnames=["ism", "sinf", "ball"])
    w.writeheader()
    w.writerows([{"ism": "Ali", "sinf": 9, "ball": 78}, {"ism": "Malika", "sin

with open("ballar.csv", newline="", encoding="utf-8") as f:
    qatorlar = list(csv.DictReader(f))
print(qatorlar[0])
print("O'rtacha:", sum(int(q["ball"]) for q in qatorlar) / len(qatorlar))
```

*Natija:*

```
{'ism': 'Ali', 'sinf': '9', 'ball': '78'}
O'rtacha: 87.0
```

Diqqat: CSV dan o'qilgan barcha qiymatlar — **satr**. Hisoblashdan oldin `int()`/`float()`.

# 8. Amaliy misol: CSV → JSON hisobot

```
import csv, json
from collections import defaultdict
with open("savdo.csv", "w", newline="", encoding="utf-8") as f:
    f.write("sana,mahsulot,summa\n2026-11-01,non,12000\n2026-11-01,sut,24000\n

jami = defaultdict(int)
with open("savdo.csv", newline="", encoding="utf-8") as f:
    for q in csv.DictReader(f):
        jami[q["mahsulot"]] += int(q["summa"])

hisobot = {"jami": sum(jami.values()), "mahsulotlar": dict(sorted(jami.items()
print(json.dumps(hisobot, ensure_ascii=False, indent=2))
```

*Natija:*

```
{
  "jami": 69000,
  "mahsulotlar": {
    "choy": 25000,
    "sut": 24000,
```

```
    "non": 20000
  }
}
```

# 9. Xulosa

- JSON: `json.dumps/loads` (satr), `json.dump/load` (fayl); `ensure_ascii=False`, `indent=2`.
- JSON da qo'shtirnoq, `true/false/null`; Python da `True/False/None`.
- CSV: `csv.reader/writer`, `DictReader/DictWriter`; `newline=""`; o'qilgan qiymatlar — satr.