

## PYTHON DASTURLASH MUHITIDA FAYLLAR BILAN ISHLASH.

*Kodirkulov Bunyod Narzulloevich**Sirdaryo viloyati Shirin shahar**Shirin energetika kolleji informatika fani o'qituvchisi*

**Annotatsiya:** Ushbu maqolada Python dasturlash tilida fayllar bilan ishlashning nazariy va amaliy jihatlari yoritilgan. Fayllarni ochish, o'qish, yozish va yopish kabi asosiy operatsiyalar hamda JSON va CSV formatlaridagi fayllarni boshqarish usullari ko'rib chiqilgan. Misollar orqali Python'da matnli, strukturalangan va statistik ma'lumotlar bilan ishlashning asosiy qoidalari tushuntirilgan. Shuningdek, maqolada Python dasturlarida fayllar bilan ishlash jarayonini samarali tashkil etish uchun tavsiyalar va amaliy dasturlar keltirilgan. Ushbu maqola Python tilini o'rganayotgan dasturchilar va fayllar bilan ishlash bo'yicha amaliy tajriba orttirishni maqsad qilgan mutaxassislar uchun mo'ljallangan.

**Kalit so'zlar:** with open, JSON, CSV, txt, statistik, fayl

**Kirish:**

Fayllar bilan ishlash dasturlashning muhim qismlaridan biri bo'lib, dasturlar o'z ma'lumotlarini saqlash, o'qish va tahlil qilish uchun fayllardan foydalanadi. Python dasturlash tili fayllar bilan ishlash uchun oddiy va kuchli imkoniyatlarni taqdim etadi. Bu maqolada Python'da fayllar bilan ishlashning asosiy usullari, jumladan fayllarni o'qish, yozish, o'zgartirish va yopish amallari, shuningdek, turli formatdagi fayllarni boshqarish bo'yicha misollar keltiriladi.

**Python'da fayllar bilan ishlash asoslari**

Python dasturlash tilida fayllar bilan ishlash uchun quyidagi bosqichlar amalga oshiriladi:

1. **Faylni ochish:** Faylni foydalanishga tayyorlash.
2. **Faylda amallarni bajarish:** O'qish, yozish yoki o'zgartirish.
3. **Faylni yopish:** Resurslarni tejash uchun ochilgan faylni yopish.

Faylni ochish uchun Python **open()** funksiyasidan foydalanadi:

```
file = open('file_name.txt', mode)
```

Bu yerda:

- **file\_name.txt** – ochilayotgan fayl nomi.
- **mode** – fayl bilan qanday amal bajarilishini belgilaydigan parametr:
  - 'r' – o'qish uchun ochish.
  - 'w' – yozish uchun ochish (mavjud bo'lsa, faylni tozalaydi).
  - 'a' – fayl oxiridan yozishni qo'shish.
  - 'b' – bayt rejimida ishlash.

### **Oddiy misol: Faylni o‘qish va yopish**

```
# Faylni ochish
file = open('example.txt', 'r')
```

```
# Ma'lumotni o‘qish
content = file.read()
print(content)
```

```
# Faylni yopish
file.close()
```

### **Fayllarni boshqarish usullari**

#### **1. Faylni o‘qish**

Python fayldagi ma'lumotlarni bir necha usul bilan o‘qishga imkon beradi:

- **Barcha ma'lumotni o‘qish:**

- with open('example.txt', 'r') as file:
- content = file.read()
- print(content)

- **Har bir satrni o‘qish:**

- with open('example.txt', 'r') as file:
- for line in file:
- print(line.strip())

- **Ma'lumotlarni ro'yxat shaklida olish:**

- with open('example.txt', 'r') as file:
- lines = file.readlines()
- print(lines)

#### **2. Faylga yozish**

Faylga yozish uchun 'w' yoki 'a' rejimi ishlatiladi:

- **Yangi fayl yaratish va yozish:**

- with open('output.txt', 'w') as file:
- file.write("Bu yangi fayl dir.\n")
- file.write("Python juda qulay dasturlash tili!")

- **Mavjud faylga yozishni qo'shish:**

- with open('output.txt', 'a') as file:
- file.write("\nBu satr fayl oxiriga qo'shildi.")

#### **3. JSON fayllar bilan ishlash**

JSON (JavaScript Object Notation) – ma'lumotlarni almashish uchun qulay format. Python'da JSON bilan ishlash uchun json moduli mavjud:

```
import json
```

```
# Ma'lumotni faylga yozish
data = {'name': 'Ali', 'age': 25, 'city': 'Toshkent'}
with open('data.json', 'w') as json_file:
    json.dump(data, json_file)
```

```
# JSON faylni o'qish
with open('data.json', 'r') as json_file:
    loaded_data = json.load(json_file)
    print(loaded_data)
```

#### 4. CSV fayllar bilan ishlash

CSV (Comma-Separated Values) formatidagi fayllar ma'lumotlar bilan ishlashda keng qo'llaniladi. Python csv modulini taqdim etadi:

```
import csv
```

```
# CSV faylni o'qish
with open('data.csv', 'r') as csv_file:
    reader = csv.reader(csv_file)
    for row in reader:
        print(row)
```

```
# CSV faylga yozish
with open('output.csv', 'w', newline='') as csv_file:
    writer = csv.writer(csv_file)
    writer.writerow(['Name', 'Age', 'City'])
    writer.writerow(['Ali', 25, 'Toshkent'])
```

#### Amaliy dastur: Matnli fayldan statistik ma'lumotlarni olish

Quyida matnli fayldan necha so'z va harf borligini aniqlash bo'yicha dastur keltiriladi:

```
# Matnli fayldan statistik ma'lumot olish
with open('example.txt', 'r') as file:
    content = file.read()
    words = content.split()
    num_words = len(words)
    num_chars = len(content)
```

```
print(f"So'zlar soni: {num_words}")
print(f"Harflar soni: {num_chars}")
```

### **Xulosa**

Python dasturlash muhitida fayllar bilan ishlash oddiy va samarali amalga oshiriladi. Bu dasturlar uchun zarur bo'lgan ma'lumotlarni saqlash, ularga kirish va ularni qayta ishlash imkonini beradi. JSON va CSV kabi formatlarni boshqarish Python'ning qo'shimcha modullari yordamida juda qulay. Ushbu maqolada berilgan misollar esa dasturchilar uchun asosiy bilimlarni mustahkamlashga xizmat qiladi.

### **Foydalanilgan adabiyotlar:**

1. Anvar Narzullayev Pythonda dasturlash asoslari
2. Mark Lutz, Learning Python. O'Reilly Media, 2020.
3. [www.mohirdev.uz](http://www.mohirdev.uz)