

PYTHONDA FAYLLAR BILAN ISHLASH

Pozilova Ra'no Zuxriddin qizi
Abdurahimova Mohichehra Avazjon qizi
Andijon Davlat Pedagogika Insituti
Aniq fanlar fakulteti Matematika va Informatika yo'nalishi
301-guruh talabalari
+998939732412
+998911515370

Annotatsiya: Ushbu maqolada katta hajmdagi ma'lumotlarni fayldan yuklab olish va dastur yakunida kerakli ma'lumotlarni va dastur natijasini faylga saqlashni o'rganamiz. Fayllar bilan ishlash dastur foydalanuvchilariga ham dasturga o'zlari istagan ma'lumotlarni yuklash imkoniyatini beradi.

Kalit so'zlar: Python, yuklab olish, fayllarni saqlash, fayllarga yozish, fayllarni o'qish, open, read.

Fayllar bilan ishlashning birinchi qadami bu fayldagi ma'lumotlarni kompyuter xotirasiga ko'chirish. Buning bir necha usuli bor, quyida ular bilan tanishamiz.

Faylni to'liqligicha o'qish.

Fayli to'lqi o'qish uchun quyidagi kodni yozamiz:

```
with open('pi.txt') as fayl:
```

```
pi = fayl.read()
```

Kodni tahlil qilamiz:

- Birinchi qatorda open() funksiyasi yordamida faylni ochayapmiz. Bunda funksiyaga argument sifatida fayl nomini berayapmiz. Bu yerda biz ochayotgan fayl va dasturimiz bir papkada bo'lishi muhim.
- open() funksiyasi faylni obyekt sifatida qaytaradi, as operatori yordamida esa biz obyektimizga fayl deb nom berayapmiz.
- Ikkinchi qatorda .read() metodi yordamida fayl obyektining tarkibidan bizga kerakli matnni olib, yangi, PI o'zgaruvchisiga yuklayabmiz.
- with operatorining vazifasi biz fayl bilan ishlab bo'lganimizdan so'ng faylni yopish. Yuqoridagi misolda, 2-qatordan so'ng Python zudlik bilan faylni yopadi. Yuqorida ko'rgan usulimiz fayl bilan ishlashning eng xavfsiz usuli. Aslida biz fayllarni to'g'ridan-to'g'ri fayl=open('pi.txt') yordamida ochishimiz, fayl bilan ishlab bo'lgandan so'ng esa fayl.close() komandasi yordamida faylni yopishimiz ham mumkin edi:

```
fayl = open('pi.txt')
```

```
PI = fayl.read()
```

print(pi)
fayl.close()

Lekin, bu usul xavfli hisoblanadi va tavsiya qilinmaydi. Gap shundaki, open() funksiyasi yordamida faylni ochganimizdan keyin, toki close() metodini chaqirgunga qadar faylimiz ochiq holatda turadi. Agar, faylni vaqtida yopmasak, yoki fayl yopilmasidan avval dasturimiz to'xtab qolsa fayl tarkibiga ziyon yetishi, ma'lumotlar yo'qotilishi mumkin. Misol uchun, boshqa dasturlarda ham (masalan Microsoft Word) faylni yopmasdan oldin kompyuteringiz o'chib qolsa, yoki dastur behosdan yopilib ketsa faylingizga ziyon yetkani kabi.

PAPKA ICHIDAGI FAYLLARNI OCHISH

Agar siz ochayotgan fayl dasturimiz bilan bir papkada emas, shu papka ichidagi papkada joylashgan bo'lsa, fayl nomidan avval papka nomi yoziladi: with open('data/pi.txt') as fayl:

```
pi = fayl.read()
```

Agar papkalar bir necha qavat bo'lsa, fayl nomini va ungacha bo'lgan papkalarni alohida yozib olgan afzal: faylnomi = 'data/math/numbers/pi.txt' with open(faylnomi) as fayl:

```
pi = fayl.read()
```

FAYLNI QATORMA-QATOR O'QISH

Ba'zida faylni to'liqligicha emas, qatorma-qator o'qish talab qilinishi mumkin. Masalan, faylda talabalarning ismi yoki kundalik ob-havo ma'lumotlari saqlanganda va hokazo. Bunday hollarda for tsiklidan foydalanamiz

```
filename = 'data/talabalar.txt'
```

```
with open(filename) as file:
```

```
for line in file:
```

```
print(line)
```

FAYLGA YOZISH

Ma'lumotlarni saqlashning eng qulay usuli bu faylga yozish. Dasturimiz bajarilishdan to'xtaganidan so'ng, xotiradagi ma'lumotlar o'chib ketishi mumkin, lekin faylga yozilgan ma'lumotlar saqlanib turaveradi. Fayllarni kelajakda qaytdan xotiraga yuklab, dasturimizni to'htagan joyidan davom etishimiz mumkin.

Yuqorida biz faylni ochishda open() funksiyasidan foydalandik, va yagona argument sifatida fayl nomini berdik. Bunda fayl faqatgina o'qish uchun ochiladi, unga yozib bo'lmaydi. Faylga ma'lumot yozish uchun open() funksiyasiga murojat qilishda fayl nomidan tashqari yana bir argument beramiz. Ikkinchi argument faylni aynan nima maqsadda ochishimizni bildiradi. Argumentlar quyidagilardan iborat bo'lishi mumkin:

Argument Qo'llanilishi Mazmuni

'w'	open('file.txt','w')	Faylni yozish uchun ochish. Fayl mavjud bo'lmasa yangi fayl yaratiladi. Fayl mavjud bo'lsa tarkibi o'chib ketadi
'r'	open('file.txt','r')	Faylni faqat o'qish uchun ochish (yozib bo'lmaydi)
'w+'	open('file.txt','w+')	Faylni o'qish va yozish uchun ochish. Fayl mavjud bo'lmasa yangi fayl yaratiladi. Fayl mavjud bo'lsa tarkibi o'chib ketadi.
'r+'	open('file.txt','r+')	Faylni o'qish va yozish uchun ochish.
'a'	open('file.txt','a')	Faylga ma'lumot qo'shish uchun ochish. Fayl mavjud bo'lmasa yangi fayl yaratiladi.
'a+'	open('file.txt','a+')	Faylga ma'lumot qo'shish va o'qish uchun yozish. Fayl mavjud bo'lmasa yangi fayl yaratiladi.

YANGI FAYLGA YOZISH

Yangi faylga ma'lumot yozish uchun open() funksiyasini chaqirishda 'w' (write) argumentidan foydalanamiz. Ochilgan faylga ma'lumot qo'shish uchun esa .write() metodini chaqiramiz.

```
faylnomi = 'ustozlar.txt'# ochilayotgan (yaratilayotgan) fayl nomi  
with open(faylnomi,'w') as fayl:  
fayl.write('anvar narzullaev') # faylga yozilayotgan ma'lumot
```

FAYLGA MA'LUMOT QO'SHISH

Agar mavjud faylga ma'lumot qo'shish talab qilinsa, open() funksiyasiga murojat qilishda 'a' (append) argumentidan foydalanamiz. Bunda yangi qo'shilgan ma'lumotlar faylning oxiriga qo'shiladi.

```
with open(faylnomi,'a') as fayl:  
fayl.write('Alijon Valiyev\n')  
fayl.write('2000')
```

O'ZGARUVCHILARNI FAYLDA SAQLASH

Yuqorida biz ma'lumotlarni matn ko'rinishida saqlashni ko'rdik. Agar dastur davomida turli o'zgaruvchilarni faylda saqlash talab qilinsa pickle modulidan foydalanamiz. Pickle ma'lumotlarni biz qanday ko'rinishda bersak, shunday ko'rinishda faylga yozadi. Yuqoridagi usuldan farqli ravishda, pickle yordamida yozilgan fayllarning tarkibini Pythondan tashqarida ko'rib bo'lmaydi.

PICKLE FAYLDAN O'QISH

Pickle fayldan o'qish uchun open() funksiyasini 'rb' (read binary) argumenti bilan chaqiramiz. O'zgaruvchilarni bitta faylga yozganimizda, har bir o'zgaruvchi alohida qatordan yoziladi. Fayldan o'qishda ham har bir qatorni alohida o'qishimiz kerak bo'ladi:

```
with open('info','rb') as file:
```

```
talaba1 = pickle.load(file)
```

```
talaba2 = pickle.load(file)
```

Foydalanilgan adabiyotlar.

1. Abbosbek Ibrogimov: “Python asoslari”.
2. Anvar Narzullayev: “Pythonda dasturlash asoslari”.
3. Abdurahmon Rashidov: “Python”
4. Ayurov Ravshan Hamdamovich: “Python dasturlash asoslari” 2020-yil.