

MAVZU YUZASIDAN SAVOLLAR



1. To'plam nima maqsadda ishlatiladi?
2. To'plamning ro'yxat va kortejdan farqli jihati nimada?
3. To'plam tarkibini o'zgartirish mumkinmi?
4. To'plamlar ustida qanday amallar bajariladi?
5. To'plam va ro'yxatning bir-biridan farqi nimada?

UYGA VAZIFA



1. Foydalanuvchi tomonidan kiritilgan sonlar qatorida qancha raqam ishtirok etganligini aniqlovchi dastur tuzing.

Kiruvchi ma'lumot

4 5 8 9 1 4 5 2

Chiquvchi ma'lumot

6

2. Foydalanuvchi tomonidan ikkita sonlar qatori kiritilgan. Nechta son ham birinchi, ham ikkinchi qatorda ishtirok etganligini aniqlovchi dastur tuzing.

Kiruvchi ma'lumot

9 5 8 4

5 9 3 4

Chiquvchi ma'lumot

2

38-dars. FAYLLAR BILAN ISHLASH

Ma'lumotlar fayllarda saqlanadi. Python dasturlash tili yordamida kompyuterdagi har xil fayl turlari bilan ishlash imkoniyati mavjud bo'lib, shartli ravishda ularni ikki turga bo'lish mumkin: *matn* va *binar fayllar*. Matn fayllari kengaytmasi cvs, txt, html va hokazolardan iborat, umuman olganda, matn shaklida ma'lumot saqlaydigan barcha fayllarlarni o'z ichiga oladi. Binar fayllarni esa tasvirli, audio, video, o'yin fayllari va boshqalar tashkil etadi. Fayl turiga qarab, fayl bilan ishlash Pythonda biroz farq qilishi mumkin.

Python kutubxonasining ochiq funksiyalari yordamida fayllar bilan ishlash mumkin. Fayllar bilan ishlaganda, amallar quyidagi ketma-ketlikda bajariladi:

- 1) open() metodi yordamida faylni ochish;
- 2) read() metodi yordamida faylni o'qish yoki write() metodi yordamida faylga yozish;
- 3) close() metodi yordamida faylni yopish.

Faylni ochish

Faylga ma'lumot yozish yoki uni tarkibini o'qish uchun fayl open() metodi yordamida ochiladi.

Sintaksisi:

open (file, mode)

`open()` funksiyasi Python tilida faylni ochish va uning tarkibini ekranda namoyish etish imkonini beradi. Ushbu funksiyaning birinchi parametri sifatida faylning qayerdaligini ko'rsatuvchi yo'l ko'rsatiladi, masalan, `D:/project/my.txt`. Ikkinchi parametr mode yordamida faylni ochish rejimi, ya'ni fayl ustida qanday ish bajarilishi ko'rsatiladi.

Rejim	Tavsifi
<i>r</i> (<i>read</i>)	Faylni o'qish uchun ochish. Agar fayl topilmasa, <code>FileNotFoundError</code> xatoligini beradi.
<i>w</i> (<i>write</i>)	Faylni qayta yozish uchun ochish. Fayl tarkibini o'chirib, bo'sh faylni ochadi. Agar fayl mavjud bo'lmasa, uni yaratadi.
<i>x</i> (<i>xwrite</i>)	Agar fayl mavjud bo'lsa, uni yozish uchun ochadi, aks holda ochmaydi.
<i>a</i> (<i>append</i>)	Faylning davomiga yozish uchun ochish. Agar fayl mavjud bo'lmasa, uni yaratadi.
<i>b</i> (<i>binary</i>)	Faylni binar (ikkilik) rejimda ochish. <i>w</i> va <i>r</i> kabi holatlar bilan birgalikda ishlatiladi. Masalan, <code>'rb'</code> , <code>'rt'</code>
<i>t</i> (<i>text</i>)	Faylni matnli holatda ochish.
+	Faylni o'qish va yozish uchun ochish.

Fayl bilan ishlash yakunlanganidan so'ng, uni `close()` metodi yordamida yopish kerak. Ushbu metod fayl bilan bog'liq barcha resurs ishlarini yakunlaydi.

Faylga yozish

Matnli faylni yozish uchun ochishda ikki xil rejim ishlatiladi: *a*(*append*) – fayl davomiga yozuv qo'shish, *w* (*write*) – fayl tarkibini o'chirib, qayta yozish.

Faylga yozish uchun `write()` metodidan foydalaniladi.

Sintaksisi:

`write(str)`

`str` parametri orqali satr kiritiladi. Agar boshqa turdagi ma'lumotlar kiritilishi lozim bo'lsa, ular satr turiga o'tkaziladi.

MISOL

<code>file=open('d:/project/my.txt', 'w')</code>	d:/project/ manzilidagi my.txt' fayli tarkibini o'chirib, ochadi. Agar mavjud bo'lmasa, yaratadi.
<code>text=file.write('Hello world!')</code>	<code>write</code> metodi yordamida faylga "Hello world!" satrini yozadi.
<code>file.close()</code>	Faylga yozish to'xtatilganini e'lon qilish uchun <code>close</code> metodi ishlatiladi.

MISOL

file=open('d:/project/my.txt', 'w')	d:/project/ manzilidagi my.txt' faylini davomiga matn yozish uchun ochadi. Agar mavjud bo'lmasa, yaratadi.
text=file.write('\nGood day!')	write metodi yordamida fayl tarkibidagi matn davomiga keyingi satrga o'tib, "Good day!" satrini yozadi.
file.close()	Faylga yozish to'xtatilganini e'lon qilish uchun close metodi ishlatiladi.

Fayldan o'qish

Fayldan ma'lumotlarni o'qish uchun r (read) rejimidan foydalaniladi.

Fayl tarkibidagi satrlarni o'qishda turli usullardan foydalaniladi:

read() – fayl tarkibidagi barcha ma'lumotlarni o'qiydi;

readline() – faylning faqat birinchi satrini o'qiydi;

readlines() – faylning barcha satrlarini ro'yxat elementi sifatida o'zlashtirib oladi.

Sintaksisi:

file_name.read()

file_name.readline()

file_name.readlines()

file_name o'rnida ochilgan fayl o'zlashtirilgan o'zgaruvchi nomi ko'rsatiladi.

Misol. read() metodi yordamida fayl tarkibini o'qish:

tt_file=open('d:/project/my.txt', 'r')	d:/project/ manzilidagi my.txt faylini ochadi.
text=tt_file.read()	read metodi yordamida fayl tarkibini o'qib, text o'zgaruvchisiga o'zlashtiradi.
print(text)	text o'zgaruvchisi qiymatini chiqaradi.
Hello world!	
Good day!	

Misol. for sikli yordamida fayl tarkibini o'qish:

tt_file=open('d:/project/my.txt', 'r')	d:/project/ manzilidagi my.txt faylini ochadi.
for text in tt_file:	for sikli yordamida tt_file o'zgaruvchisidagi har bir satr text o'zgaruvchisiga o'zlashtiradi
print(text, end="")	text o'zgaruvchisi qiymatini chiqaradi.
Hello world!	
Good day!	

Misol. `readline()` metodi yordamida fayl 1-satrini chiqarish:

<pre>tt_file=open('d:/project/my.txt', 'r') text1=tt_file.readline() print(text1) Hello world!</pre>	d:/project/ manzilidagi my.txt faylini ochadi.
	readline metodi yordamida fayl 1-satrini o'qib, text1 o'zgaruvchisiga o'zlashtiradi.
	Text1 o'zgaruvchisi qiymatini chiqaradi.

Misol. `readline()` metodi va `while` sikli yordamida fayl satrlarini chiqarish:

<pre>tt_file=open('d:/project/my.txt', 'r') text1=tt_file.readline() while text: print(text, end='') text=tt_file.readline() Hello world! Good day!</pre>	d:/project/ manzilidagi my.txt faylini ochadi.
	readline metodi yordamida fayl 1-satrini o'qib, text o'zgaruvchisiga o'zlashtiradi.
	text o'zgaruvchisi qiymati 0 ga teng bo'lgunga qadar sikl davom ettiriladi.
	text o'zgaruvchisi qiymatini chiqaradi.
	Fayldan keyingi satrni o'qib, yana text o'zgaruvchisiga o'zlashtiradi.

Misol. `readlines()` metodi yordamida fayl satrlarini chiqarish:

<pre>tt_file=open('d:/project/my.txt', 'r') text1=tt_file.readline() text1 = text [0] text2 = text [1] print(text1, end='') print(text2) Hello world! Good day!</pre>	d:/project/ manzilidagi my.txt faylini ochadi.
	readlines metodi yordamida fayl tarkibi ro'yxatga o'zlashtirib olinadi. Elementlari fayl satrlaridan tarkib topgan ro'yxat hosil qilinadi.
	Ro'yxatning 0 indeksidagi element qiymati text1 ga o'zlashtiriladi.
	Ro'yxatning 1 indeksidagi element qiymati text2 ga o'zlashtiriladi.
	text1 o'zgaruvchisi qiymati chiqariladi.
	text2 o'zgaruvchisi qiymati chiqariladi.

with ..as operatori

Fayl bilan ishlashda turli holatlarga duch kelish mumkin. Masalan, fayldan foydalanishga ruxsat berilmagan bo'lishi mumkin va h. k. Buyday holatlarda dasturning ma'lum bir qatorida xatolik yuz berishi va undan keyingi satrlardagi `close()` metodi orqali faylning yopilishi bajarilmasligi mumkin. Buning uchun Pythonda *with ..as* operatori mavjud bo'lib, bu operator har qanday holatda ham fayl yopilishini ta'minlaydi.

Sintaksisi:

with open(file, mode) as file_name

commands

with orqali ochiq fayl file_name o'zgaruvchi aniqlanadi hamda commands'da keltirilgan buyruqlar ketma-ketligi bajariladi. So'ng qanday holat yuzaga kelishidan qat'iy nazar, fayl avtomatik ravishda yopiladi.

```
with open('d:/project/my.txt', 'a')
as tt_file:
```

```
    tt_file.write('\nwith .. as')
```

```
    print('\nGood be', file = tt_file)
```

Hello world!

Good day!

with .. as

Good be

d:/project/ manzilidagi my.txt faylini davomiga yozish uchun ochadi va tt_file'ga o'zlashtiradi.

tt_file fayl davomini keyingi satriga 'with .. as' matnini write() metodi orqali yozadi.

tt_file fayl davomini keyingi satriga 'Good day' matnini print() orqali yozadi.

MAVZU YUZASIDAN SAVOLLAR



1. Fayllar nima maqsadda ishlatiladi?
2. Faylni ochish uchun qaysi buyruqdan foydalaniladi?
3. Faylning davomiga yozish imkoniyati mavjudmi, agar mavjud bo'lsa, qaysi rejim orqali amalga oshiriladi?
4. Faylni avtomatik tarzda yopish uchun qaysi operatoridan foydalaniladi?

39-dars. AMALIY MASHG'ULOT

Misol. E. Vohidovning "O'zbekim" qasidasi 4 misrasini faylga yozish dasturini tuzing.

Tarixingdir ming asrlar

Ichra pinhon, o'zbekim,

Senga tengdosh Pomir-u

Oqsoch Tiyonshon, o'zbekim.

```
file_name = 'd:/project/qasida.txt'
```

```
qasida = list()
```

```
for i in range(4):
```

```
    q_satri = input ('Qasida satrini
    kiriting ' + str(i+1) + ' - ')
```

Fayl nomi va manzilini file_name'ga o'zlashtiradi.

"qasida" nomli bo'sh ro'yxat yaratiladi.

4 misralik qasidani kiritish uchun sikllar soni 4 ga teng deb olinadi.

Qasidaning kiritilgan har bir satri o'qib olinadi va q_satri ga o'zlashtiriladi