O’ZBEKISTON RESPUBLIKASI MUDOFAA VAZIRLIGI AXBOROT-KOMMUNIKATSIYA TEXNOLOGIYALARI VA ALOQA HARBIY INSTITUTI



Kiberxavfsizlik fakulteti

Axborot texnalogiyalari va dasturiy injneringi kafidrasi

AT-142-21-o’quv guruhi

2-Mustaqilish: Pythonda fayllar bilan ishlash

Bajardi: Otajonov S.S

Qabulqildi:Sapayev Sh. R

Toshkent 2024-yil

Mavzu: Pythonda fayllar bilan ishlash

Ushbu bo'limda 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.

**FAYLDAN O'QISH**

Kompyuterimizda aksar ma'lumotlar fayl ko'rinishida saqlanadi. Bu xoh matn bo'lsin, xoh jadval, xoh rasm, xoh video. Fayllarda turli ma'lumotlar saqlanishi mumkin, ob-havo ma'lumotlari, yillik hisobotlar, mijozlarning telefon raqamlari, talabalarning baholari va hokazo.

Ko'pgina holatlarda dastur davomida katta ma'lumotlarni aynan fayllardan o'qib olish talab qilinadi. Ayniqsa, tahliliy dasturlarda fayl ko'rinishida saqlangan, katta hajmdagi jadvallar bilan ishlash tabiiy. Lekin fayllar bilan ishlash boshqa holatlarda ham ko'p asqotadi, misol uchun oddiy matnni html ko'rinishga o'tkazishni avtomatlashtiruvchi dastur yozishda.

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

Uch qatordan iborat faylimiz π*π* sonining qiymatini saqlaydi (30 xona aniqlik bilan).

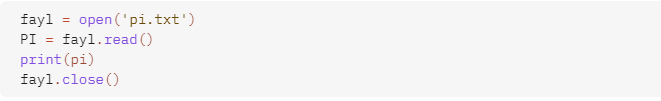
Fayli to'lqi o'qish uchun quyidagi kodni yozamiz:



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.

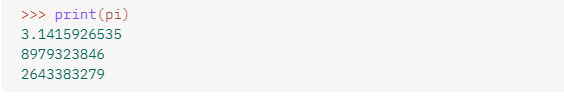
Yuqorida ko'rgan usulimiz fayl bilan ishlashning eng xavfsiz usuli. Aslida biz fayllarni to'g'ridan-to'g'ri fayl=open('pi.txt') yordamida ochishimiz, fayl bilan ishlab bo'lgandan so'ng esa fayl.close() komandasi yordamida faylni yopishimiz ham mumkin edi:



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.

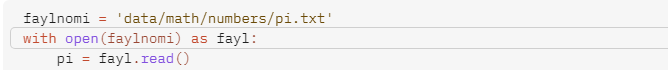
Shuning uchun open() funksiyasiga with orqali murojat qilganimizda, faylimiz with blokining oxirigacha ochiq turadi, va with tugashi bilan, fayl ham yopiladi. Demak fayl ustidagi amallarni biz with bloki ichida bajarib olishimiz kerak.

Keling endi pi ning qiymatini konsilga chiqaramiz:



Agar siz ochayotgan fayl dasturimiz bilan bir papkada emas, shu papka ichidagi papkada joylashgan boʻlsa, fayl nomidan avval papka nomi yoziladi:



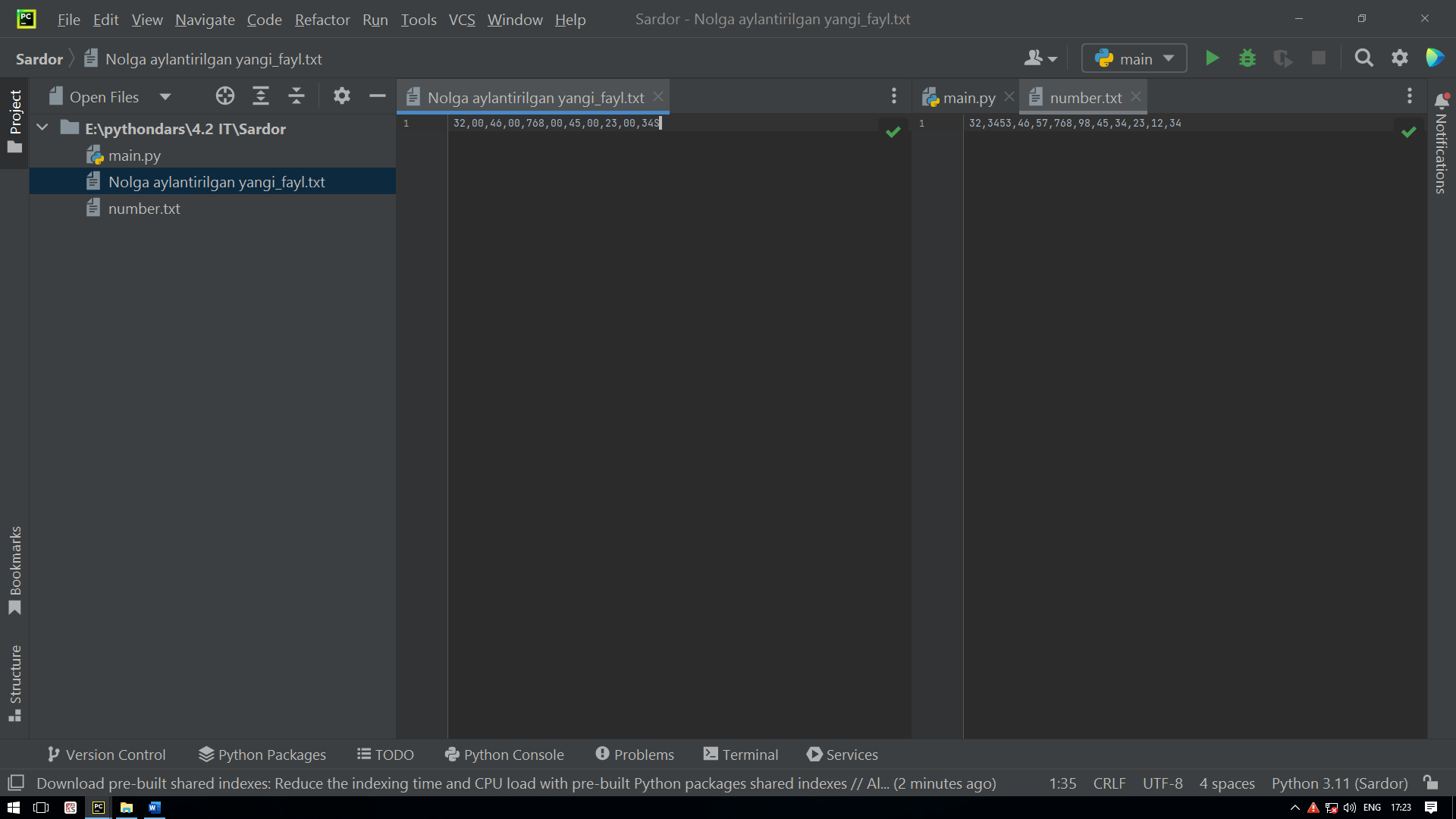
Agar papkalar bir necha qavat boʻlsa, fayl nomini va ungacha boʻlgan papkalarni alohida yozib olgan afzal: 

2-topshiriq

Fayl berilgan fayl ichidagi elementlarning juft numerdagilarini ikkita nolga aylantiring va yangi faylga yozing

with open('number.txt', 'r') as file:  
 numbers = file.read().strip().split(',')  
  
for i in range(len(numbers)):  
 if i % 2 == 1:  
 numbers[i] = '00'  
  
with open('Nolga aylantirilgan yangi\_fayl.txt', 'w') as file:  
 file.write(','.join(numbers))  
  
print("juft raqamlar 0 ga almashtirildi")

2-topshiriq natijasi



Xulosa

Python'da fayllar bilan ishlash juda sodda va qulay. Faylni ochish, o'qish, yozish, yopish va boshqa ko'plab operatsiyalarni amalga oshirish uchun yuqoridagi metodlardan foydalanishingiz mumkin. Faylni o'qish va yozish operatsiyalarini with blokida amalga oshirish, resurslarni boshqarish va xatoliklardan qochish uchun eng yaxshi amaliyot hisoblanadi.