PYTHON DASTURLASH TILIDA FAYLLAR VA ISTISNOLAR BILAN ISHLASH.

Qulliyev Javohirbek G'anijon o'g'li
Buxoro davlat pedagogika instituti "Aniq fanlar" kafedrasi o'qituvchisi
Muzaffarova Dilshoda Mehriddin qizi
Buxoro davlat pedagogika instituti "Aniq va tabiiy fanlar" fakulteti talabasi

Annotatsiya: *Python dasturlash tili ma'lum oddiy yoki chiziqli dasturlarni yozish uchun emas balki bu tilda turli murakkablikdagi loyihalarni ishlab chiqish mumkin . Bu til dasturchilarga yangi va yangi yo'nalishlarga kirish imkonini bermoqda. Python quyidagi sohalarda qo'llaniladi: Kiber xavfsizlikga oid masalalaeni hal qilishda, Web va Internet dasturlash, kompyuter o'yinlarini yaratish, ma'lumotlar bazasi bilan ishlash (DB), computer vision, suniy intelekt, juda tez rivojlanayotgan buyumlar interneti (IoT) texnologiyasi va hokazo. Ushbu maqolada biz python dasturlash tilida fayllar bilan ishlashni ko'rib chiqamiz hamda fayllardan ma'lumotlarni o'qishni va turli rejimlarda ishlashni ko'rib chiqamiz.*

Kalit so'zlar: *replace, open, file, with, reverse, close.*

РАБОТА С ФАЙЛАМИ И ИСКЛЮЧЕНИЯМИ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON.

Куллиев Жавохирбек Ганижон угли
Бухарский государственный педагогический институт Преподаватель кафедры «Точные науки»
Музаффарова Дилшода Мехриддиновна
Бухарский государственный педагогический институт Студентка факультета «Точные и естественные науки».

Аннотация: *Язык программирования Python не предназначен для написания каких-то простых или линейных программ, но на этом языке можно разрабатывать проекты различной сложности. Этот язык позволяет программистам выходить на все новые и новые направления. Python используется в следующих областях: решение проблем кибербезопасности, веб- и интернет-программирование, создание компьютерных игр, управление базами данных (БД), компьютерное зрение, искусственный интеллект и быстро развивающаяся технология Интернета вещей (IoT). и так далее. В этой статье мы рассмотрим работу с файлами на языке программирования Python, а также рассмотрим чтение данных из файлов и работу в разных режимах.*

Ключевые слова: *replace, open, file, with, reverse, close.*

WORKING WITH FILES AND EXCEPTIONS IN THE PYTHON PROGRAMMING LANGUAGE.

Qulliyev Javohirbek G'anijon o'g'li
Bukhara State Pedagogical Institute Teacher of the "Exact Sciences" department
Muzaffarova Dilshoda Mehriddin qizi
Bukhara State Pedagogical Institute Student of the Faculty of "Exact and Natural Sciences".

Abstract: *Python programming language is not for writing certain simple or linear programs, but it is possible to develop projects of various complexity in this language. This language allows programmers to enter new and new directions. Python is used in the following areas: Cybersecurity problem solving, Web and Internet programming, computer game creation, database management (DB), computer vision, artificial intelligence, and the rapidly developing Internet of Things (IoT) technology. and so on. In this article, we'll look at working with files in the python programming language, and we'll look at reading data from files and working in different modes.*

Keywords: *replace, open, file, with, reverse, close.*

KIRISH

Ushbu maqolada katta 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. 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.

Faylni to'liqligacha o'qish. Boshlanishiga bizga fayl kerak. Keling, yangi son.txt faylini yaratamiz, va ichiga quyidagi matnni joylaymiz:

85461875154842545659564l son.txt son.txt faylini yuklab oling bir qatordan iborat faylimiz 85461875154842545659564l sonining qiymatini saqlaydi (30 xona aniqlik bilan).

Fayli to'lqi o'qish uchun quyidagi kodni yozamiz:with open('son.txt') as fayl:

son = 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, son 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('son.txt') yordamida ochishimiz, fayl bilan ishlab bo'lgandan so‘ng esa fayl.close() komandasi yordamida faylni yopishimiz ham mumkin edi:

fayl = open('pi.txt')SON= fayl.read()print(SON)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 yetgani 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.

.replace() metodi matn tarkibidagi biror harf yoki belgini boshqa harf yoki belgi bilan almashtirish uchun ishlatiladi.

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/son.txt') as fayl:

son = fayl.read()

Agar papkalar bir necha qavat bo‘lsa, fayl nomini va ungacha bo‘lgan papkalarni alohida yozib olgan afzal:

faylnomi = 'data/programm/numbers/son.txt'with open(faylnomi) as fayl:

son= fayl.read()

Windowsda papkalar orasida "\" belgisi ishlatilsada, Pythonda "/" belgisini ishlataveramiz. Agar \ belgisini ishlatishni istasangiz, bu belgini 2 marta yozing: C:\\python\\darslar\\data

Faylni qatorma-qator o‘qish. Ba‘zida faylni to‘liqligicha emas, qatorma-qator o‘qish talab qilinishi mumkin. Masalan, faylda talabalrning ismi yoki kundalik ob-havo ma‘lumotlari saqlangdanda va hokazo. Bunday hollarda for siklidan foydalanamiz:

filename = 'data/ismlar.txt' with open(filename) as file: for line in file:

print(line) Natija:

Salim sayidov

Nazir alimov

Nargiza allayeva

Qatorlarni ro‘yxat ko‘rinishida saqlab olish uchun, .readlines() metodidan foydalanamiz.

with open(filename) as file: ismlar = file.readlines()print(ismlar)

Natija: ['Salim sayidov\n', 'Nazir olimov\n', 'Nargiza allayeva\n']

E'tibor bering, har bir talaba ismidan so‘ng qator tashlab belgisi (\n) tushib qolgan. Biz bu belgilarni .rstrip() metodi yordamida olib tashlashimiz mumkin:

talabalar = [ism.rstrip() for talaba in talabalar] print(ismlar)

Natija: ['Salim sayidov', 'Nazir olimov', 'Nargiza allayeva']

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‘tagan 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 = 'talabalar.txt'# ochilayotgan (yaratilayotgan) fayl nomi

```
with open(faylnomi,'w') as fayl:
fayl.write(Muzaffarova Dilshoda) # faylga yozilayotgan ma'lumot
```

Diqqat!!! open() funksiyasini 'w' argumenti bilan chaqirganimizda ehtiyot bo‘lishimiz kerak, sababi agar bunday fayl mavjud bo'lsa, uning ichidagi barcha ma'lumotlar o'chib ketadi.

Faylga yozayotgan ma'lumotlarimiz matn ko'rinishida bo'lishi kerak. Aks holda dasturimiz xato beradi.

```
faylnomi = 'yangi_fayl.txt'
ism = Alijon Sattarov
tyil = 2004
with open(faylnomi,'w') as fayl:
fayl.write(ism)
fayl.write(tyil)
```

Natija: TypeError: write() argument must be str, not int

Xatoning oldini olish uchun sonlarni avval str() funksiyasi yordamida matnga keltirib olamiz.

```
faylnomi = 'yangi_fayl.txt'
ism = Alijon Sattarov
tyil = 2004
with open(faylnomi,'w') as fayl:
fayl.write(ism)
fayl.write(str(tyil))
```

Fayllar matn formatida yoziladi, va biz ularni istalgan matn muharriri yordamida ochib ko‘rishimiz mumkin.

Faylda saqlangan ma’lumotlar. Faylga bir nechta ma'lumot yozganimizda, ma’lumotlar alohida qatordan emas, bir qatorda bir-biriga qo‘shib qo‘shib yoziladi.

Buning oldini olishimiz uchun matn so‘ngida \n belgisini qo‘shib ketishimiz kerak boladi:

```
faylnomi = 'yangi_fayl.txt'
ism = Alijon Sattarov
tyil = 2004
with open(faylnomi,'w') as fayl
fayl.write(ism+'\n')
fayl.write(str(tyil)+'\n')
```

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('1992')
```

Agar biz ochayotgan fayl mavjud bo‘lmasa, Python yangi fayl yaratadi.

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 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.

Xulosa. Umuman olganda python dasturlash tilida faylllar bilan ishlash boshqa dasturlash tillariga qaraganda osonroq va soddaroq hisoblanadi hamda tilning foydalanish imkoniyatlarini oshiradi. Ushbu maqola orqali biz faqat qisman ma’lumotlarni keltirb o‘tdik. Agarda fayllar haqida kengroq tushuncha olishni xohlasangiz quyidagi manbalarga murojaat qilishingiz mumkin.

FOYDALANILGAN ADABIYOTLAR:

1. www.python.org.
2. www.w3schools.com.
3. www.geeksforgeeks.org.