



PYTHONDA FAYLLAR VA ISTISNOLAR BILAN ISHLASH

PYTHONDA FAYLLAR BILAN ISHLASH

Fayllar bilan ishlash pythonda muhim qismlardan biri. Ayniqsa, web dasturlar bilan ishlashda. Pythonda fayllarni hosil qilish, o'qish, yangilash va o'chirish imkoniyati mavjud.

Fayllarni ochish **open()** funksiyasi bilan amalga oshadi. Bunda ushbu funksiya 2 ta parameter qabul qiladi: **Fayl nomi** va **rejimi**. Rejim deganda faylni qay maqsadda ochish nazarda tutiladi. Bu rejimlar quyidagilar:

- ❖ **"r"** – **Read** – faylni o'qish uchun ochish. Agar fayl mavjud bo'lmasa, xatolik yuz beradi.
- ❖ **"a"** – **Append** – faylga qo'shimcha qo'shish uchun ochish. Agar fayl mavjud bo'lmasayangi fayl ochadi.
- ❖ **"w"** – **Write** – faylga yozish uchun ochish. Agar fayl mavjud bo'lmasa, yangi fayl ochadi.
- ❖ **"x"** – **Create** – yangi fayl hosil qilish. Agar bunday fayl mavjud bo'lsa xatolik yuzberadi.

Bundan tashqari qo'shimcha 2 ta rejim bor. Ular yuqoridagilar bilan qo'llaniladi:

- ❖ **"t"** – **Text** – matn turi ya'ni fayl matndan iborat bo'ladi.
- ❖ **"b"** – **Binary** – binar rejim (ikkilik). Masalan rasmlarni binary rejimda joylashtirish.

Fayl ochish

Fayl ochish quyidagicha bo'ladi. Agar bizda qandaydir **.txt** fayl bo'lsa, uni o'qish rejimida ochish bunday qilamiz:

```
f = open("fayl_nomi.txt")
```

E'tibor berdingizmi, bu yerda hech qanday rejim ko'rsatilmadi. Chunki **"r"** va **"t"** rejimlari avtomatik hisoblanadi. Ya'ni hech qanday rejim ko'rsatilmasa, ular ishga tushadi. Faylimiz o'qilish uchun ochiladi va u matn ko'rinishida bo'ladi. Bularni aniq ko'rsatgan holda ham ochish mumkin. Bu quyidagicha bo'ladi:

```
f = open("fayl_nomi.txt", "rt")
```

Fayllarni o'qish

Avvalgi darsda nima maqsadda ochishimizga qarab, turli rejimlarini borligini ko'rib chiqdik. Hozir biz faylni ochib o'qish uchun ochib ko'ramiz.

Avval **.txt** kengaytmali biror faylga 4 – 5 qatorli matn kiritamiz, uni python faylimiz joylashgan papkaga bitr nom bilan saqlaymiz. Uni **open()** funksiyasi bilan ochamiz va **read()** funksiyasi bilan o'qiyamiz:

```
f = open("fayl_nomi.txt", "r")  
  
print(f.read())
```

Agar fayl boshqa bir papkada joylashgan bo'lsa o'sha faylga yo'llanmani ko'rsatish kerak:

```
f = open("D:\fayllarim\fayl_nomi.txt", "r")  
  
print(f.read())
```

Faqat ma'lum qismni o'qish

read() funksiyasi fayldagi butun matnni o'qiydi. Ammo bizga uning faqatgina ma'lum bir qismi kerak bo'lsa, uni belgilab ko'rsatishimiz kerak. Quyidagi misolimizdagi kod matnning dastlabki 10 ta harf yoki belgisini ekranga chiqaradi:

```
f = open("fayl_nomi.txt", "r")  
  
print(f.read(10))
```

Qatorlarni o'qish

Matnni qatorma qator o'qish ham mumkin. **readline()** funksiyasi aynan shuning uchun mo'ljallangan. Uni bir marta ishlatsak birinchi qator o'qiladi. Yana ishlatsak ikkinchisi va hokazo o'qiladi. Quyidagi kodimiz ishga tushsa, birinchi va ikkinchi qatorlarni o'qiydi.

```
f = open("fayl_nomi.txt", "r")  
  
print(f.readline())  
print(f.readline())
```

Faylni yopish

Fayl bilan ishlab bo'lgach albatta uni yopish kerak. Buni **close()** funksiyasi bilan amalga oshiramiz. Yuqoridagi kodimizda faylni ochib dastlabki ikkita qatorni o'qigan edik. Endi o'sha faylni yopamiz.

```
f = open("fayl_nomi.txt", "r")

print(f.readline())
print(f.readline())

f.close()
```

Faylga yozish

Avvalgi darslarda fayllarni nima maqsadda ochishimizga qarab turli rejimlar borligini ko'rib chiqdik. Hozir biz faylni ochib o'qish uchun faylni ochib ko'ramiz.

Avval **.txt** kengaytmali biror faylga 4-5 qatorli papka kiritamiz. Uni python faylimiz joylashgan papkaga biror nom bilan saqlaymiz. Uni **open()** funksiyasi bilan ochamiz va **read()** funksiyasi bilan o'qiyamiz:

```
f = open("fayl_nomi.txt", "a")
f.write("Matnga qo'shimcha qo'shdik.")
f.close()

# Endi faylni o'qiyamiz
f = open("fayl_nomi.txt", "r")
print(f.read())
f.close()
```

Agar faylni **"w"** rejimida ochib unga ma'lumot kiritsak, o'sha fayldagi avvalgi ma'lumotlar o'chib ketadi. Uning o'rniga biz kiritgan ma'lumot qoladi:

```
f = open("fayl_nomi.txt", "w")
f.write("Matnga qo'shimcha qo'shdik.")
f.close()

# Endi faylni o'qiyamiz
f = open("fayl_nomi.txt", "r")
print(f.read())
f.close()
```

Yangi fayl ochish

Yangi fayl ochish uchun ham **open()** funksiyasini ishlatamiz. Uni **"x"** rejimida ochish kerak. Agar bunday fayl allaqachon mavjud bo'lsa dasturda xatolik yuz beradi.

“a” va “w” rejimlari aslida yozish uchun ishlatilsada biz ochmoqchi bo’lgan fayl mavjud bo’lmasa ular avtomatik tarzda shu nomli yangi fayl ochadi.

my_file nomli yangi fayl hosil qilish quyidagicha bo’ladi:

```
f = open("my_file.txt", "x")
```

Faylni o’chirish

Faylni o’chirish uchun os moduliga murojaat qilamiz va undagi **os.remove()** funksiyasidan foydalanamiz. Masalan, biror faylimiz bor. Uni nomini bilamiz. Uni o’chirish quyidagicha bo’ladi:

```
import os  
  
os.remove("fayl_nomi.txt")
```

Fayl mavjudligini tekshirish

Fayl mavjudligini tekshirib, agar u mavjud bo’lsa o’chirish quyidagicha bo’ladi:

```
import os  
  
if os.path.exists("fayl_nomi.txt"):   
    os.remove("fayl_nomi.txt")  
else:  
    print("Bunday fayl mavjud emas")
```

Papkani o’chirish

Agar biror bir papkaning o’zini o’chirmoqchi bo’lsak **os.rmdir()** funksiyasini ishlatamiz. Ammo biz faqat bo’sh papkalarni o’chirishimiz mumkin. Masalan, bizda **dasturlar** degan papka bor va ubo’m bo’sh. Uni o’chirish uchun mana bunday qilish kerak:

```
import os  
  
os.rmdir("dasturlar")
```

PYTHONDA ISTISNOLAR BILAN ISHLASH

Agar kodimizda xatolik yuz bersa yoki istisno holatlar bo'lib qolsa Python bizga xatolik haqida xabar beradi. Bunday istisno holatlar bilan ishlash uchun **try** va **except** blokidagi amal bajariladi.

Masalan, biz **x** degan o'zgaruvchini ekranga chiqarmoqchi bo'lamiz. Lekin unday o'zgaruvchining o'zi yo'q bo'lsa xatolik yuz beradi va dastur ishlamaydi. Shuning uchun biz shunday qilamizki, agar **x** o'zgaruvchi mavjud bo'lmasa, bu haqida xabar berilsin:

```
try:
    print(x)
except:
    print("x o'zgaruvchi mavjud emas")
```

x o'zgaruvchi mavjud emas

except kalit so'zini ishlatganimizda Python istalgan turdagi istisno holat uchun amal qiladi. Ammo biz qaysi turdagi istisno holatini tekshirishni o'zimiz belgilashimiz mumkin. Bunday holatlarning turlari ko'p, biz esa asosiylarini sanab o'tamiz:

- ❖ **NameError** – murojaat qilinayotgan obyekt topilmasa, ishga tushadi.
- ❖ **ValueError** – o'zgaruvchining qiymati unga mos bo'lmagan turda bo'lsa ishga tushadi.
Masalan, biror harfli qiymatni son deb qabul qilmoqchi bo'lsak, shunday bo'ladi.
- ❖ **TypeError** - o'zaro nomutanosib qiymatlar bilan amallar bajarilsa ishga tushadi. Masalan harfga son qo'shmoqchi bo'lganimizda.
- ❖ **ZeroDivisionError** – Istalgan sonni nolga bo'lish holati bo'lganda ishga tushadi.

Hozir biror sonni nolga bo'lishni tekshiramiz.

Buni shunchaki except bilan ham yoki **ZeroDivisionError** bilan ta'kidlab ham tekshirish mumkin. Natija bir xil bo'ladi:

```
try:
    5/0
except:
    print("nolga bo'lish mumkin emas")

try:
    7/0
except ZeroDivisionError:
    print("nolga bo'lish mumkin meas")
```

```
nolga bo'lish mumkin emas
nolga bo'lish mumkin meas
```

else

else kalit so'zi hech qanday xatolik yuz bermaganda bajariladigan amalni ko'rsatish uchun ishlatiladi:

```
try:
    print("Salom")
except:
    print("Dasturda xatolik bor")
else:
    print("Hech qanday xatolik yo'q")
```

```
Salom
Hech qanday xatolik yo'q
```

finally

finally bloki ichida ko'rsatilgan amal xatolik bo'lishi yoki bo'lmasligidan qat'iy nazar bajariladi.

```
try:
    print(x)
except:
    print("x mavjud emas")
else:
    print("Hech qanday xatolik yo'q")
finally:
    print("Tekshiruv tugadi")
```

```
x mavjud emas
Tekshiruv tugadi
```

Istisno holatini hosil qilish

Dasturchi sifatida o'zimiz ham istisno holatini tuzishimiz mumkin. Buning uchun **raise** kalit so'zini ishlatamiz. Masalan, biror son agar noldan kichik bo'lsa dasturimiz xatolik haqida xabar berishi kerak bo'lsa:

```
x = -1

if x < 0:
    raise Exception("Manfiy son aniqlandi")
```

raise kalit so'zi bilan qanday turdagi istisno holati bo'lishini ham o'zimiz belgilashimiz mumkin. Hozir **TypeError** istisnoli holatini tuzamiz. Bunda agar kiritilgan qiymati butun sonli o'zgaruvchi bo'lmasa xatolik haqida xabar berilsin:


```
x = "abc"

if type(x) is not int:
    raise TypeError("Qiymat butun son bo'lishi kerak")
```

USER INPUT

Biz dasturimizda foydalanuvchidan biror ma'lumot kiritishni so'rashimiz mumkin. User input anashunday xizmat bo'lib, u **input()** funksiyasi yordamida ishlaydi. **Python 2.7** versiyasida bu funksiya **raw_input()** bo'lgan.

```
Ismingizni kiriting: Abbosbek
Sizning ismingiz: Abbosbek
print('Sizning ismingiz: ', name)
```

Foydalanuvchi nafaqat so'z balki son kiritishi ham mumkin va shu son ustida amal bajarishga egabo'lamiz. Bunda endi kiritilayotgan ma'lumotni son deb qabul qilishimiz buyurishimiz kerak. Hozir kiritilgan sonning kvadratini chiqaruvchi dastur tuzamiz.

```
x = int(input("Son kiriting: "))

kv_x = x*x
print(kv_x)
```

```
Son kiriting: 7
49
```