

Python programming language

Week 12. Python File input/output

Alisher Ismailov

Lecturer

Department of Information technology
Andijan Branch of Tashkent Financial Institute

Email: alisherismailov534@gmail.com

Python dasturlash tili

12-Mavzu. Python dasturlash tilida fayllar bilan ishlash

Ma'ruzachi: Alisher Ismoilov







Axborot texnologiyalari kafedrası

Toshkent moliya instituti Andijon filiali

Elektron pochta: alisherismailov534@gmail.com

12-Mavzu. Python dasturlash tilida fayllar bilan ishlash

Reja:

-  Fayl bilan ishlash
-  Serverda faylni ochish
-  Fayllarni yopish
-  Yangi fayl yaratish
-  Faylni o'chirish
-  Butun Papkani o'chirish

Python dasturlash tilida fayllar bilan ishlash

Fayl bilan ishlash har qanday dasturlash tilining muhim qismi hisoblanadi.

Python dasturlash tilida fayllarni yaratish, o'qish, yangilash va o'chirish uchun bir nechta funksiyalar mavjud.

Python dasturlash tilida fayllar bilan ishlash

Python dasturlash tilida fayllar bilan ishlashning asosiy funksiyasi **open()** funksiyasidir.

open() funksiyasi ikkita parametрни oladi; **fayl nomi** va rejimi (***tartibot***).

Faylni ochishning to'rt xil usuli (rejimi, (***tartibot***)) mavjud:

"r" - o'qish - Faylni o'qish uchun ochadi, agar fayl mavjud bo'lmasa, xato ko'rsatadi

"a" - Qo'shish - faylni qo'shish uchun ochadi, agar u mavjud bo'lmasa, uni yaratadi

"w" - Write – faylga ma'lumot yozish uchun ochadi, agar u mavjud bo'lmasa, uni yaratadi

"x" - Yaratish - ko'rsatilgan faylni yaratadi, agar fayl mavjud bo'lsa, xato ko'rsatadi

Python dasturlash tilida fayllar bilan ishlash

- **Serverda faylni ochish**
- Faraz qilaylik, bizda Python dasturi bilan bir xil papkada joylashgan quyidagi fayl bor:
- *Demo.txt (fayl nomi)(o'zgartir)*
- *Salom! Demo.text fayliga xush kelibsiz.*

Ushbu sinov uchun yaratildi.

Python dasturlash tilida fayllar bilan ishlash

Faylni ochish uchun `open()` funksiyasidan foydalanish mumkin.

`open()` funksiyasi fayl mazmunini o'qish uchun `read()` funksiyasiga ega fayl obyektini qaytaradi:

```
In [154]: f = open("Demo.txt", "r")  
          print(f.read())
```

```
Salom, Bu fayl Python dasturlash tili  
uchun yaratildi
```

Agar fayl boshqa joyda joylashgan bo'lsa, fayl manzilini ko'rsatishingiz kerak bo'ladi.

Python dasturlash tilida fayllar bilan ishlash

Faylning faqat qismlarini o`qish

`read()` funksiyasi butun matnni qaytaradi, lekin siz qaytarmoqchi bo`lgan belgilarni ham belgilashingiz mumkin:

```
In [40]: f = open("Demo.txt", "r")  
         print(f.read(5))
```

```
Salom
```

Python dasturlash tilida fayllar bilan ishlash

- **Qatorlarni o‘qish**
- **readline()** funksiyasi yordamida bitta qatorni o‘qishingiz mumkin:

```
In [42]: f = open("Demo.txt", "r")  
print(f.readline())
```

```
Salom! Demo.text fayliga xush kelibsiz.
```

Python dasturlash tilida fayllar bilan ishlash

- `readline()` funksiyasi ikki marta chaqirib, ikkita qatorni o‘qishingiz mumkin:

```
In [158]: f = open("Demo.txt", "r")  
          print(f.readline())  
          print(f.readline())
```

```
Salom, Bu fayl Python dasturlash tili  
uchun yaratildi
```

Python dasturlash tilida fayllar bilan ishlash

- Fayl qatorlarini sikl orqali butun fayl ma'lumotlarini o'qishingiz mumkin:

```
In [159]: f = open("Demo.txt", "r")
```

```
for x in f:  
    print(x)
```

```
Salom, Bu fayl Python dasturlash tili  
uchun yaratildi
```

Python dasturlash tilida fayllar bilan ishlash

- **Fayllarni yopish**

- Ishni tugatgandan soʻng, faylni har doim yopish yaxshi amaliyot hisoblanadi. Buning uchun **close()** funksiyasidan foydalanishingiz mumkin:

Python dasturlash tilida fayllar bilan ishlash

- *Eslatma: Siz har doim fayllaringizni yopishingiz kerak, ba'zi hollarda buferlash tufayli faylga kiritilgan o'zgarishlar faylni yopmaguningizcha ko'rinmasligi mumkin.*

```
In [46]: f = open("Demo.txt", "r")
print(f.readline())
f.close()
```

```
Salom! Demo.text fayliga xush kelibsiz.
```

Python dasturlash tilida fayllar bilan ishlash

- **Mavjud faylga ma'lumot yozish**
- Mavjud faylga ma'lumot yozish uchun **open()** funksiyasiga parametr qo'shishingiz kerak:
- "a" - Qo'shish - faylning oxiriga qo'shiladi
- "w" - Write - har qanday mavjud tarkibni ustidan qayta yozadi

Python dasturlash tilida fayllar bilan ishlash

```
f = open("Demo.txt", "a")
f.write("Bu qator dastur orqali kiritildi")
f.close()

#open and read the file after the appending:
f = open("Demo.txt", "r")
print(f.read())
```

```
Salom! Demo.txt fayliga xush kelibsiz.
Ushbu sinov uchun yaratildi.
Now the file has more content!Bu qator dastur orqali kiritildi
```

Python dasturlash tilida fayllar bilan ishlash

- **"Demo.txt" faylini ochish va tarkibini qayta yozish:**

```
f = open("Demo.txt", "w")  
f.write("fayl ma'lumotlari butunlay o`chirilib, ushbu qator kiritildi")  
f.close()
```

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

```
fayl ma'lumotlari butunlay o`chirilib, ushbu qator kiritildi
```

Python dasturlash tilida fayllar bilan ishlash

- **Yangi fayl yaratish**

- Python dasturlash tilida yangi fayl yaratish uchun **open()** funksiyasidagi quyidagi parametrlardan biridan foydalanishingiz mumkin:

- "x" - Yaratish - fayl yaratadi, agar fayl mavjud bo'lsa, xatoni qaytaradi
- "a" - Qo'shish - ko'rsatilgan fayl mavjud bo'lmasa fayl yaratadi
- "w" - Write - ko'rsatilgan fayl mavjud bo'lmasa fayl yaratadi

Python dasturlash tilida fayllar bilan ishlash

```
In [55]: f = open("Demo3.txt", "x")
          f.write("dastur orqali fayl yaratildi")
          f.close()

          f = open("Demo3.txt", "r")
          print(f.read())
```

```
dastur orqali fayl yaratildi
```

Python dasturlash tilida fayllar bilan ishlash

- **Faylni o‘chirish**
- Faylni o‘chirish uchun siz **OS modulini import** qilishingiz va uning **os.remove()** funksiyasini ishga tushirishingiz kerak:

Python dasturlash tilida fayllar bilan ishlash

```
In [59]: import os
         if os.path.exists("Demo.txt"):
             os.remove("Demo.txt")
         else:
             print("Fayl o`chirilgan ")
```

```
Fayl o`chirilgan
```

Python dasturlash tilida fayllar bilan ishlash

- **Butun Papkani o‘chirish**
- Butun papkani o‘chirish uchun **os.rmdir()** funksiyasidan foydalanishingiz mumkin:

```
In [60]: import os  
os.rmdir("mening_papkam")
```

Python dasturlash tilida fayllar bilan ishlash

- **Fayl yaratishga oid qisqa dastur yaratib ko`ramiz.**
- **Quyida dastur qadamlari va kodlari keltirilgan**

Python dasturlash tilida fayllar bilan ishlash

- **Login faylni yaratish**

- `f = open("login.txt", "w")`
- `f.write("python")`
- `f.close()`

- `loginfayl = open("login.txt", "r")`
- `login = loginfayl.readline()`

Python dasturlash tilida fayllar bilan ishlash

- **Parol faylni yaratish**

- `p = open("parol.txt", "w")`
- `p.write("python2023")`
- `p.close()`

- `parolfayl = open("parol.txt", "r")`
- `parol = parolfayl.readline()`

Python dasturlash tilida fayllar bilan ishlash

- **Login va Parolni ekranga chop etamiz**

- `print(login)`
- `print(parol)`

Python dasturlash tilida fayllar bilan ishlash

- **Foydalanuvchidan login va parol kiritishni so`raymiz**
- lg = input("Loginni kiriting: ")
- pr = input("Parolni kiriting: ")

Python dasturlash tilida fayllar bilan ishlash

- **Kiritish nomli funksiya yaratamiz**

- `def Kiritish(kitob):`
- `k = open("Kitoblar.txt", "a")`
- `k.write(kitob)`
- `k.close`

Python dasturlash tilida fayllar bilan ishlash

- **Uy sahifasini yaratamiz va uy sahifasida menu ham yaratamiz**

- if lg == login and pr == parol:
- print("Hush kelibsiz")
-
- menu = input("""Agar kitob nomini kiritmoqchi bo`lsangiz 1 raqamini kiriting,
• agar kitob nomlarini ko`rmoqchi bo`lsangiz 2 raqamini kiriting""")
-
- if int(menu) == 1:
- kitobKiritish = input("Kitob nomini kiriting: ")
- Kiritish(kitobKiritish)
- elif int(menu) == 2:
- kitob = open("Kitoblar.txt", "r")
- print(kitob.read())
- else:
- print("Login yoki parol noto`g`ri")

Python dasturlash tilida fayllar bilan ishlash

- **Dasturni to`liq kodi**

```
Hush kelibsiz
Agar kitob nomini kiritmoqchi bo`lsangiz 1 raqamini kiriting,
                    agar kitob nomlarini ko`rmoqchi bo`lsangiz 2 raqamini kiriting
2
BoburnomaAvlodlar DovoniAlchemistShaytanat
```

In []:

Foydalanilgan adabiyotlar

1. Mastering Object-Oriented Python: Build powerful applications with reusable code using OOP design patterns and Python 3.7, 2nd Edition, Steven F. Lott, Packt Publishing (June 14, 2019)
2. Learning Python, 5th Edition Fifth Edition, Mark Lutz , O'Reilly Media, June 12, 2013
3. Python Programming for Beginners: The Ultimate Guide for Beginners to Learn Python Programming: Crash Course on Python Programming for Beginners, AMZ Publishing, Independently published (July 13, 2021)
4. <https://www.python.org/>
5. <https://www.w3schools.com/>
6. <https://www.codecademy.com/catalog/language/python>
7. <https://realpython.com/>
8. <https://www.anaconda.com/>

**E'tiboringiz
uchun rahmat!**