

PYTHONDA FAYLLAR BILAN ISHLASH KO‘NIKMALARI

Xudayberdiyev Firdavs Qodir o‘g‘li

Sharof Rashidov nomidagi Samarqand davlat universiteti

Urgut filiali 2-kurs talabasi

Xursandov Hamidillo Fayzillo o‘g‘li

Sharof Rashidov nomidagi Samarqand davlat universiteti Urgut filiali Tabiiy fanlarni o‘qitish metodikasi kafedrasida assistenti

ANNOTATSIYA

Ushbu maqolada python dasturlash tilida fayllar bilan ishlash, fayllarda ma’lumotlarni ochish, o‘qish, o‘chirish va yopish, shuningdek fayllar bilan ishlashda qulayliklari va afzalliklari haqida so‘z boradi.

Kalit so‘zlar: *python dasturlash tili, fayllar bilan ishlash, open, xavfsizlik*

АННОТАЦИЯ

В данной статье рассказывается о работе с файлами на языке программирования Python, открытии, чтении, удалении и закрытии данных в файлах, а также об удобстве и преимуществах работы с файлами.

Ключевые слова: *язык программирования Python, работа с файлами, открытость, безопасность*

ABSTRACT

This article talks about working with files in the python programming language, opening, reading, deleting and closing data in files, as well as the convenience and benefits of working with files.

Key words: *python programming language, working with files, open, security*

KIRISH

So‘nggi yillarda Python dunyodagi eng mashhur dasturlash tillaridan biriga aylandi. U mashinani o‘rganishdan tortib veb-saytlarni ishlab chiqish va dasturiy ta’minotni sinovdan o‘tkazishgacha bo‘lgan hamma narsa uchun ishlatiladi. Python - bu umumiy maqsadli dasturlash tili bo‘lib, u dasturiy ta’minot va veb-ishlab chiqish, avtomatlashtirish va umumiy topshiriqlarni bajarish kabi turli vazifalar uchun ishlatilishi mumkinligini anglatadi. Ushbu ko‘p qirralilik yangi foydalanuvchilar uchun foydalanish qulayligi bilan birgalikda uni bugungi kunda eng ko‘p ishlatiladigan dasturlash tillaridan biriga aylandi.

Pythonda fayl operatsiyalari ma’lumotlarni fayllarga o‘qish va yozish uchun zarurdir va ular ma’lumotlarni manipulyatsiya qilish, tahlil qilish va saqlashda hal qiluvchi rol o‘ynaydi. Ushbu maqolada biz Pythonda fayl operatsiyalari asoslarini,

jumladan, fayllarni ochish, o‘qish, yozish va manipulyatsiya qilish, shuningdek, fayllar bilan samarali ishlashning ba’zi ilg‘or usullarini o‘rganamiz.

Fayl - bu shaxsiy kompyuterda saqlanadigan muhim ma’lumotlar elementi. Har bir fayl fayl nomi va kengaytmasi bilan tavsiflanishi mumkin. Pythonda fayllar bilan ishlash dasturlaringiz uchun ma’lumotlarni saqlashi va boshqarishi mumkin bo‘lgan har xil turdagi fayllarni yaratishga imkon beradi. Birinchidan, yaratilishi mumkin bo‘lgan har xil turdagi fayllar haqida so‘z yuritsak. Umumiy fayl turlariga matnli fayllar, CSV (Comma Separated Value) fayllari va JSON (JavaScript Object Notation) fayllari kiradi. Matn fayllari ma’lumotlarni ochiq matn sifatida saqlaydi, CSV va JSON esa tuzilgan ma’lumotlarni saqlash uchun mashhur formatlardir.

MUHOKAMA VA NATIJALAR

Faylni ochish vazifani bajaradigan har bir fayl turidagi asosiy qadamdir. Pythondagi fayllar o‘rnatilgan `open()` funksiyasi bilan ochilishi mumkin. Ideal holda, u ikkita string argumentni oladi:

- Biz ochmoqchi bo‘lgan fayl nomi va kengaytmani o‘z ichiga olgan fayl yo‘li qator sifatida uzatilishi kerak;

- Biz faylni ochmoqchi bo‘lgan rejim, satr sifatida uzatiladi.

Shunday qilib, faylni ochish sintaksisi quyidagicha ko‘rinadi:

`open(“”, “”)`

Pythonda faylni ochish rejimlari

Rejim	Tavsif	Misol
r	O‘qish rejimi. Faylni o‘qish uchun ochadi (standart rejim). Agar fayl mavjud bo‘lmasa, xato paydo bo‘ladi.	<code>file = open('example.txt', 'r')</code>
w	Yozish rejimi. Faylni yozish uchun ochadi. Agar fayl mavjud bo‘lsa, u kesiladi. Agar fayl mavjud bo‘lmasa, u yaratiladi.	<code>file = open('example.txt', 'w')</code>
a	Qo‘shish rejimi. Faylni yozish uchun ochadi, lekin mavjud ma’lumotlarni qayta yozish o‘rniga fayl oxiriga yangi ma’lumotlarni qo‘shadi. Agar fayl mavjud bo‘lmasa, u yaratiladi.	<code>file = open('example.txt', 'a')</code>
x	Eksklyuziv yaratish rejimi. Faylni yozish uchun ochadi, lekin u allaqachon mavjud bo‘lmasa. Agar fayl mavjud bo‘lsa, xato paydo bo‘ladi.	<code>file = open('example.txt', 'x')</code>
b	Ikkilik rejim. Faylni matn rejimi o‘rniga ikkilik rejimda ochadi.	<code>file = open('example.txt', 'rb')</code>

t	Matn rejimi (standart). Faylni ikkilik rejimda emas, matn rejimida ochadi.	<code>file = open('example.txt', 'rt')</code>
+	Yangilash rejimi. Faylni o‘qish va yozish uchun ochadi.	<code>file = open('example.txt', 'r+')</code>

Python dasturlash tilida ikki xil fayllar mavjud: matnli fayllar va ikkilik fayllar. Matn fayllari oddiy ASCII satrlari bo‘lib, ular matnli ma’lumotlarni inson tomonidan o‘qiladigan shaklda saqlaydi. Ikkilik fayllar o‘qib bo‘lmaydigan shakldagi ikkilik ma’lumotlarni o‘z ichiga oladi. Ular bajariladigan dasturlarni, tasvirlar yoki ovozli kliplar kabi multimedia obyektlarini saqlash uchun ishlatilishi mumkin.

Fayllarni ochish/yopish: Ikkilik fayldan ma’lumotlarni o‘qish/yozish uchun avval uni to‘g‘ri rejim argumenti (o‘qish, yozish yoki qo‘shish) bilan ochish kerak. Faylda amallar bajarilgandan so‘ng, uni `close()` funksiyasi bilan yopish kerak.

Ma’lumotlarni o‘qish/yozish: Ikkilik faylni o‘qish va yozishda siz qaysi rejimdan foydalanmoqchi ekanligingizni belgilashingiz kerak; bu faylni ochishda faqat o‘qish uchun ikkilik ruxsat uchun “rb” yoki faqat ikkilik ruxsatni yozish uchun “wb” konstantalaridan biriga o‘tish orqali amalga oshiriladi. Odatiy bo‘lib, ikkilik fayl bilan o‘qish/yozish operatsiyalari bayt oqimlari (ya’ni, bir vaqtning o‘zida 8 bit) yordamida amalga oshiriladi.

Ikkilik rejim: Shuni ta’kidlash kerakki, katta hajmdagi ma’lumotlar bilan ishlashda ikkilik rejimdan foydalanish eng yaxshi amaliyotdir, chunki u kichikroq xotira maydoni tufayli matn rejimiga nisbatan tezroq uzatish tezligini ta’minlaydi.

Pythonda fayllar bilan ishlash afzalliklari:

Ko‘p qirralilik: Pythonda fayllar bilan ishlash sizga fayllarni yaratish, o‘qish, yozish, qo‘shish, nomini o‘zgartirish va o‘chirish kabi keng ko‘lamli operatsiyalarni bajarishga imkon beradi.

Moslashuvchanlik: Pythonda fayllar bilan ishlash juda moslashuvchan, chunki u turli xil fayl turlari (masalan, matnli fayllar, ikkilik fayllar, CSV fayllar va boshqalar) bilan ishlash va fayllar ustida turli operatsiyalarni bajarish (masalan, o‘qish, yozish, qo‘shish, va boshqalar.) imkonini beradi.

Foydalanuvchi uchun qulay: Pythonda fayllar bilan ishlash fayllar bilan ishlash uchun qulay interfeysni ta’minlaydi, bu fayllarni yaratish, o‘qish va boshqarishni osonlashtiradi.

O‘zaro platformalar: Python fayllar bilan ishlash funksiyalarida fayllarni qayta ishlash turli platformalarda (masalan, Windows, Mac, Linux) ishlaydi, bu uzluksiz integratsiya va muvofiqlikni ta’minlaydi.

Pythonda fayllar bilan ishlashning kamchiliklari:

Xatoga moyil: Pythonda fayllar bilan ishlash xatolarga moyil bo‘lishi mumkin, ayniqsa kod ehtiyotkorlik bilan yozilmagan bo‘lsa yoki fayl tizimi bilan bog‘liq muammolar mavjud bo‘lsa (masalan, fayl ruxsatlari, fayllarni blokirovka qilish va h.k.).

Xavfsizlik: Pythonda fayllar bilan ishlash, ayniqsa, dastur tizimdagi nozik fayllarga kirish yoki o‘zgartirish uchun foydalanish mumkin bo‘lgan foydalanuvchi ma’lumotlarini qabul qilsa, xavfsizlikka xavf tug‘dirishi mumkin.

Murakkablik: Pythonda fayllar bilan ishlash, ayniqsa, yanada rivojlangan fayl formatlari yoki operatsiyalari bilan ishlashda murakkab bo‘lishi mumkin. Fayllar to‘g‘ri va xavfsiz ishlov berilishini ta’minlash uchun kodga diqqat bilan e’tibor qaratish lozim.

Ishlash: Pythonda fayllar bilan ishlash boshqa dasturlash tillariga qaraganda sekinroq bo‘lishi mumkin, ayniqsa katta fayllar bilan ishlashda yoki murakkab operatsiyalarni bajarishda.

Umuman olganda, Pythonda fayllar bilan ishlash keng doiradagi operatsiyalarni bajarish uchun ishlatilishi mumkin bo‘lgan kuchli va ko‘p qirrali vositadir. Biroq, dasturlarni yozishda Pythonda fayllar bilan ishlashning afzalliklari va kamchiliklarini diqqat bilan ko‘rib chiqish, kodning xavfsiz, ishonchli va yaxshi ishlashini ta’minlash kerak.

XULOSA

Xulosa qilib aytganda, Pythonda fayllar bilan ishlash ma’lumotlar fayllari bilan ishlash uchun juda foydali va ko‘p qirrali vositadir. Bugun muhokama qilgan qadamlarni umumlashtirib, siz turli xil fayl formatlari bilan osongina ishlashingiz va ma’lumotlaringizni aniq ehtiyojlaringizga moslashtirishni boshlashingiz mumkin. Raqamli dunyo qanchalik ko‘p ma’lumotlar yig‘ish va manipulyatsiyaga tayanishini hisobga olib, biz ushbu mahoratning ahamiyatini aytib o‘tdik.

Python dasturida fayllarga kirish va ularni boshqarish uchun ajoyib imkoniyatdir. Boshqa dasturlash tillaridan farqli o‘laroq, Python File Openda fayllar bilan ishlash va juda oddiy va yuqorida aytib o‘tilganidek, ko‘p vaqtni tejashga yordam beradi. Shu bilan birga, ularning vazifalarini yanada samarali qilish uchun fayllarni qayta ishlash bilan bog‘liq yana bir nechta operatsiya va usullarni o‘rganishga harakat qilish mumkin.

ADABIYOTLAR (REFERENCES)

1. JetBrains. Python Developers Survey 2018 Results. 2018. url: <https://www.jetbrains.com/research/python-developerssurvey-2018>

2. Hinsen K (2007). ScientificPython Manual.URL <http://dirac.cnrs-orleans.fr/ScientificPython/ScientificPythonManual/>
3. Wes McKinney. Python for data analysis. O'Reilly, Sebastopol, California, first edition, October 2012. Annotation: Book on data analysis with Python introducing thePandas library.
4. <https://www.analyticsvidhya.com/blog/2021/10/introduction-to-file-operations-in-python/>
5. <https://datatrained.com/post/file-handling-in-python/>