

Pythonda fayllar bilan ishlash:

Biz o'tkan darsda yangi fayl yaratish hamda uni python indentifikatorida ochishni organgan edik. Dasturlar biz bergan buyruqlarni so'zsiz bajaradi shu sababdan ham bu sohaga qiziqadiganlar soni doimo ortib bormoqda. Pypi.org saytiga joylangan kutubxonalar haqida ko'p marotaba aytib o'tdik. Shu bilan birgalikda python standart qiymatlar bilan ishlashni ham o'tib kelmoqdamiz. Bugungi darsligimiz ham hular bilan o'tadi. Biz fayl yaratdik hamda unga qanday qilib malumot kiritishni organdik. Endilikda biz o'zimiz yaratgan faylni pythondan chiqmagan holda o'chirishni o'rganamiz. **Kod_1** Biz o'tkan darsda kodimiz turgan manzilga yangi fayl ochishni organgan edik. **Kod_1** ni ishga tushirdik va hatolikga duch keldik. Xatolikni tarjima qilsak bu nomli fayl topilmadi. Ya'ni o'chirishdan oldin biz yaratishimiz lozim ekan. **Kod_2** ni ishga tushiramiz hamda fayl yaratib bolingandan song. **Kod_3** ga yuzlanamiz. **Biz Kod_1** da xatolikga duch keldik. Bunday xatoliklar kormaslik uchun aksar dasturchilar **Kod_3** kabi ishlashadi. Sizga gam shunday ishlashni tavsiya qilib qolaman. Yani bu yerda yangi yaratgan faylimizni o'chirib yuboradi hamda Fayl muvafaqiyatli o'chirildi matnini konsolga chiqaradi. Aks holda Fayl topilmadi deya chiqadi. E'tibor bering biz OS kutubxonasining remove metodidan foydalanmoqdamiz.

Fayllar bilan ishlashni davom etamiz. Biz shu tartibda papkalarni ham yaratishimiz hamda ularni olib tashlashimiz mumkin.

Papka yaratish uchun os kutubxonasidan foydalansak ham boladi. **Kod_4** ga axamiyat bering bu yerda **makedirs** metodidan foydalangan holda yangi papka yaratmoqdamiz agarda kodni ikki marotaba ishga tushirsangiz kodda WinError chiqishi kuzatilinadi. Ya'ni papka allaqachon mavjud deb chiqadi. Yuqorida aytib o'tkanimizdek yaratgan papkamizni endi o'zhib yuborishimiz ham mumkin. **Kod_5** bu yerda berilgan nomdagi papkani olib tashlamoqdamiz. **Rmdir** bu metod papkalarni yoq qilishda ishlatilindi.

Massiv(array) nima?

Massiv, dasturlashda bir yoki bir nechta elementlarni birlikda saqlaydigan ma'lumot tuzilishidir. Ushbu tuzilishda, biror ma'lum bir turdagi elementlar o'zaro tartibda joylashadi va indekslar orqali murojaat qilinadi. Massiv elementlariga murojaat qilish uchun indekslardan foydalaniladi. Indeks 0 dan boshlanib, massiv elementlarining tartibini bildiradi. Misol uchun, ro'yhat massivining birinchi elementiga murojaat qilish uchun indeks 0 ni ishlatamiz. Massivlar, ma'lumotlarni bir qatordan ko'p bo'limlarga (jadvallarga) ham tuzish imkoniyatini beradi. Bu massivning bir nechta qatorlardan va ustunlardan iborat matritsa shaklidagi tuzilishni bildiradi. Shunday holda, massivni boshqarish va ma'lumotlarga murojaat qilishning ko'p yordam va imkoniyatlari mavjud. Massivlar dasturlashning asosiy konseptlaridan biridir va ko'p tillarda, asosan Python, C++, Java va boshqalarida ishlatiladi.

pip install numpy

NumPy, Python'da ilmi hisob-kitoblarda foydalaniladigan kuchli kutubxonadir. Bu kutubxona kop ishlatiluvchi kutubxonalar orasida top 10 talikdan biri hisoblanadi. NumPy, ko'p o'lchamli massiv ob'ektlarini va ushbu massivlar ustida ishlash uchun keng imkoniyatlar taqdim etadi. Bu kutubxonaning oxirgi versiyasi 1.24 hisoblanadi hamda Python 3.7 dan boshlab ishlaydi. Kutubxonani **pip install numpy** orqali kamyuteriingizga yuklab olishingiz mumkin. Kutubxonani kodingizga **import numpy** yoki **import numpy as np** orqali chaqirishingiz mumkin. Bu haqida toliq malumotni <https://numpy.org/> sayti orqali olishingiz mumkin. NumPy - massivlar bilan ishlash uchun ishlatiladigan Python kutubxonasi. Shuningdek, u chiziqli algebra, Furiye transformatsiyasi va matritsalar sohasida ishlash uchun funktsiyalarga ega. NumPy 2005 yilda **Travis Oliphant** tomonidan yaratilgan. Bu ochiq kodli loyiha bo'lib, undan erkin foydalanishingiz mumkin. **NumPy** raqamli Python degan ma'noni anglatadi.

Nima uchun NumPy ro'yxatlardan tezroq?

NumPy massivlari ro'yxatlardan farqli o'laroq xotirada doimiy ravishda bitta joyda saqlanadi, shuning uchun jarayonlar ularga kirish va ularni juda samarali boshqarishi mumkin. Bu xatti-harakat informatika fanida mos yozuvlar joylashuvi deb ataladi. Bu NumPy ro'yxatlarga qaraganda tezroq bo'lishining asosiy sababidir. Shuningdek, u eng so'nggi CPU arxitekturalari bilan ishlash uchun optimallashtirilgan. NumPy Python kutubxonasi bo'lib, qisman Python tilida yozilgan, lekin tez hisoblashni talab qiladigan qismlarning aksariyati C yoki C++ da yozilgan.

NumPy kod bazasi qayerda?

NumPy uchun manba kodi ushbu github omborida joylashgan <https://github.com/numpy/numpy>

github: ko'p odamlarga bir xil kod bazasida ishlash imkonini beradi. **Kod_6** NumPy veriyasini aniqlab oldik.

NumPy dan foydalanamiz:

NumPy kutubxonasidan foydalangan holda yangi soda turdagi massiv yaratamiz. **Kod_7**. array (ro'yxat) dastur tili kontekstida, bir yoki bir nechta elementlarni birlikda saqlaydigan bir ma'lumot tuzilishi. Bu, ma'lum bir turdagi elementlarni birdaniga saqlash uchun foydalaniladi.

NumPy da o'lchamlar sonini tekshiring?

NumPy massivlari **ndim** massivning nechta o'lchami borligini bildiruvchi butun sonni qaytaruvchi atributni taqdim etadi.

Misol

Massivlar qancha o'lchamga ega ekanligini tekshirmoqdamiz: **Kod_8**

Massiv elementlariga kirish

Massivni indekslash massiv elementiga kirish bilan bir xil. Siz massiv elementiga uning indeks raqamiga murojaat qilib kirishingiz mumkin.

NumPy massivlaridagi indekslar 0 dan boshlanadi, ya'ni birinchi element 0 indeksga ega, ikkinchisi esa 1 va hokazo. Indeksiga murojatni **Kod_10** da ko'rishingiz mumkin. Massivlarni bir biriga ularni qo'shgan holda javobni chop qilishingiz mumkin: **Kod_11** ga etibor qiling. Bu yerda biz 3-hamda 4 – indeksga berilgan qiymatlarni qo'shgan holda javobni chop qildik.

2 o'lchamli massivlar elementlariga kirish uchun biz elementning o'lchami va indeksini ifodalovchi vergul bilan ajratilgan butun sonlardan foydalanishimiz mumkin. Ikki o'lchovli massivlarni qatorlar va ustunlar bilan jadval kabi o'ylab ko'ring, bu erda o'lcham satrni va indeks ustunni ifodalaydi. **Kod_12**. Birinchi qator, ikkinchi ustundagi elementga kirish. **Kod_13**. Ikkinchi qator, beshinchi ustundagi elementga kirish.

3 o'lchamli massivlar elementlariga kirish uchun biz elementning o'lchamlari va indeksini ifodalovchi vergul bilan ajratilgan butun sonlardan foydalanishimiz mumkin. **Kod_14**