

**NUMPY KUTUBXONASIDA KO'P O'LCHOVLI MASSIVLAR BILAN
ISHLASH**

Tojimamatov Isroiljon Nurmatovich

*Farg'ona davlat universiteti o'qituvchisi
israiltojimatov@gmail.com,*

Komilova Zulxumor Xokimovna

Farg'ona davlat universiteti o'qituvchi

Kazimjonova Madinaxon Habibullo qizi

*Farg'ona davlat universiteti talabasi,
Madinaxonkozimjonova321@gmail.com*

Annotatsiya. Ushbuxon dasturlash maqolasida keng kutubxonaning NumPy kutubxonasining asosiy vositalari va uning yordamida ko'p o'lchovli massivlar bilan ishlash usullari ilmiy asosda tahlil qilish. Dasturiy ta'minot, sun'iy intellekt va katta ma'lumotlar tahlilida NumPy dasturiy ta'minoti, sifat strukturalari, ko'rib chiqish va samarali usullar. Maqola haqiqatga mos, xolis va ilmiy manbalarga xolis holda yozib bo'lib, dasturchilar, ilmiy izlanish olib tashlash va axborot texnologiyalari sohasi mutaxassislari uchun foydali bo'lishi mumkin.

Kalit so'zlar: NumPy, ko'p o'lchovli massiv, indekslash, axis, matematik amallar, statistik funktsiyalar, ma'lumotlar tahlili.

Аннотация. В данной статье по программированию представлен научный анализ основных инструментов обширной библиотеки NumPy и методов работы с многомерными массивами с ее использованием. Программное обеспечение NumPy, качественные структуры, обзор и эффективные методы в программном обеспечении, искусственном интеллекте и аналитике больших данных. Статья написана на основе фактов, беспристрастно и научно и может быть полезна программистам, исследователям и специалистам в области информационных технологий.

Ключевые слова: NumPy, многомерный массив, индексирование, ось, математические операции, статистические функции, анализ данных.

Annotation. This programming article provides a scientific analysis of the main tools of the NumPy library and methods for working with multidimensional arrays using it. NumPy software in software engineering, artificial intelligence and big data analysis, quality structures, reviews and effective methods. The article is written in a factual, unbiased and scientific way, and can be useful for programmers, researchers and information technology professionals.

Key words: *NumPy, multidimensional arrays, indexing, axis, mathematical operations, statistical functions, data analysis.*

Matritsalar deb ham ataladigan ko'p o'lchovli massivlar Pythonda kuchli ma'lumotlar tuzilmasi hisoblanadi. Ular sizga ma'lumotlarni bir nechta o'lchamlarda yoki o'qlarda saqlash va boshqarish imkonini beradi. Rasmlar, videolar va ilmiy ma'lumotlar kabi tuzilgan ma'lumotlarni ko'rsatish va qayta ishlash uchun matematika, statistika va informatika kabi sohalarda massivlardan foydalanasiz.

Pythonda siz NumPy, Pandas va TensorFlow kabi turli kutubxonalar yordamida ko'p o'lchovli massivlarni yaratishingiz mumkin. Ushbu maqolada biz Pythonda massivlar bilan ishlash uchun eng mashhur va keng qo'llaniladigan kutubxonalardan biri bo'lgan NumPy-ga e'tibor qaratamiz.

NumPy ko'p o'lchovli massivlarni samarali yaratish va boshqarish uchun foydalanishingiz mumkin bo'lgan kuchli N o'lchovli massiv ob'ektini taqdim etadi. Endi biz NumPy yordamida Pythonda ko'p o'lchovli massivlarni qanday yaratish va ular bilan ishlashga oid ba'zi misollarni ko'rib chiqamiz.

NumPy yordamida ko'p o'lchovli massivlarni qanday yaratish mumkin.

NumPy yordamida ko'p o'lchovli massiv yaratish uchun biz `np.array()` funktsiyadan foydalanishimiz va argument sifatida ichki qiymatlar ro'yxatiga o'tkazishimiz mumkin. Tashqi ro'yxat massivning satrlarini, ichki ro'yxatlar esa ustunlarni ifodalaydi.

NumPy yordamida 2 o'lchovli massivni qanday yaratishga misol:

```
import numpy as np
# 3 ta satr va 4 ta ustundan iborat 2 o'lchovli massiv yaratish
arr = np.array([[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]])
# Massivni chop eting
print(arr)
```

Chiqish:

```
→ [[ 1  2  3  4]
    [ 5  6  7  8]
    [ 9 10 11 12]]
```

Ushbu misolda biz birinchi navbatda NumPy kutubxonasini *import* qilamiz. Keyin funksiya yordamida 2 o'lchovli massiv yaratamiz `np.array()` va argument sifatida ro'yxatiga o'tamiz. Har bir ichki ro'yxat massivning qatorini, tashqi ro'yxat esa barcha qatorlarni o'z ichiga oladi. Nihoyat, `print()` yordamida massivni chop etamiz.

`np.zeros()` NumPy, shuningdek, `np.ones()`, va kabi ko'p o'lchovli massivlarni yaratish uchun boshqa funktsiyalarni ham taqdim etadi masalan, `np.random.rand()`. Siz ushbu funktsiyalardan tasodifiy qiymatlarga ega ma'lum shakl va o'lchamdagi massivlarni yaratish uchun foydalanishingiz mumkin.

NumPy yordamida ko'p o'lchovli massivlarga qanday kirish va o'zgartirish mumkin.

Ko'p o'lchovli massivni yaratganimizdan so'ng, indekslash va kesish yordamida uning elementlariga kirishimiz va o'zgartirishimiz mumkin. Biz satr va ustundagi $[i, j]$ elementga kirish uchun indeks belgisidan foydalanamiz.

NumPy yordamida 2 o'lchovli massiv elementlariga qanday kirish va ularni o'zgartirishga misol:

```
import numpy as np
# 3 ta satr va 4 ta ustundan iborat 2 o'lchovli massiv yarating
arr = np.array([[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]])
# 1-qator, 2-ustundagi elementga kirish
print(arr[1, 2]) # Chiqish: 7
# 0-qator, 3-ustundagi elementni o'zgartiring
arr[0, 3] = 20
# O'zgartirilgan massivni chop eting
print(arr)
```

Chiqish:

```
7
[[ 1  2  3 20]
 [ 5  6  7  8]
 [ 9 10 11 12]]
```

Ushbu misolda biz funktsiyadan foydalanib 2 o'lchovli massiv yaratamiz `np.array()`, so'ngra indekslash yordamida 1-qator, 2-ustundagi elementga kiramiz. Keyin yana indekslash yordamida 0-qator, 3-ustundagi elementni o'zgartiramiz. Nihoyat, funksiya yordamida o'zgartirilgan massivni chop etamiz `print()`.

Ko'p o'lchovli massivning bir nechta elementlariga bir vaqtning o'zida kirish va ularni o'zgartirish uchun biz kesishdan ham foydalanishimiz mumkin. Biz qatorlar orqali va ustunlardan iborat `arr[i:j, k:l]` pastki qatorga kirish uchun tilim belgisidan foydalanamiz $i, j-1, k, l-1$

NumPy yordamida 2 o'lchovli massiv elementlariga kirish va ularni o'zgartirish uchun kesishdan qanday foydalanishga misol:

```

import numpy as np
# 3 ta satr va 4 ta ustundan iborat 2 o'lchovli massiv yarating
arr = np.array([[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]])
# 0 dan 1 gacha satrlar va 1 dan 2 gacha ustunlarni o'z ichiga olgan pastki qatorga kirish
subarr = arr[0:2, 1:3]
# Pastki qatorni chop eting
print(subarr)
# Quyi qatorni 2 ga ko'paytirish orqali o'zgartiring
subarr *= 2
# O'zgartirilgan massivni chop eting
print(arr)

```

Chiqish:

```

⇒ [[2 3]
   [6 7]]
   [[ 1  4  6  4]
    [ 5 12 14  8]
    [ 9 10 11 12]]

```

Bu misolda biz `np.array()` funksiya yordamida 2 o'lchovli massiv yaratamiz, so'ngra 0 dan 1 gacha satrlar va 1 dan 2 gacha ustunlarni o'z ichiga olgan pastki massivga kirish uchun kesishdan foydalanamiz. Keyin pastki qatorni 2 ga ko'paytirish orqali o'zgartiramiz va `print()` funksiya yordamida o'zgartirilgan asl massivni chop etamiz.

Ko'p o'lchovli massivlarda amallarni qanday bajarish kerak.

NumPy ko'p o'lchovli massivlarda amallarni samarali bajarish uchun foydalanishingiz mumkin bo'lgan matematik va statistik funktsiyalarning keng doirasini taqdim etadi. Bu funktsiyalar turli shakl va o'lchamdagi massivlarda elementlar bo'yicha operatsiyalar, matritsalar va boshqa amallarni bajarishga yordam beradi.

NumPy yordamida 2 o'lchovli massivda ba'zi umumiy operatsiyalarni qanday bajarishga misol:

```

import numpy as np
# 3 ta satr va 4 ta ustundan iborat 2 o'lchovli massiv yarating
arr = np.array([[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]])
# Barcha elementlarning yig'indisini hisoblang
print(np.sum(arr)) # Chiqish: 78
# Har bir qatorning o'rtacha qiymatini hisoblang
print(np.mean(arr, axis=1)) # Chiqish: [ 2.5  6.5 10.5]
# Ikki matritsaning nuqta mahsulotini hisoblang
b = np.array([[2, 3], [4, 5], [6, 7], [8, 9]])
print(np.dot(arr, b)) # Chiqish: [[ 60  72]
                                #    [140 172]
                                #    [220 272]]

```

Chiqish:

```
⇒ 78  
[ 2.5  6.5 10.5]  
[[ 60  70]  
 [140 166]  
 [220 262]]
```

Bu misolda biz funksiya yordamida 2 o'lchovli massiv yaratamiz *np.array()*, so'ngra massivda amallarni bajarish uchun turli NumPy funksiyalaridan foydalanamiz.

Funktsiya yordamida birinchi navbatda barcha elementlarning yig'indisini hisoblaymiz *np.sum()*. Keyin funksiya yordamida har bir satrning o'rtacha qiymatini hisoblaymiz *np.mean()* va *axis=1* har bir satr bo'ylab o'rtachani hisoblash uchun parametrni belgilaymiz. Nihoyat, funksiya b yordamida 2 o'lchovli massiv va boshqa 2 o'lchovli massivning nuqta mahsulotini hisoblaymiz *np.dot()*.

Ko'p o'lchovli massivlar Pythonda kuchli va muhim ma'lumotlar strukturasidir. Ular bizga katta hajmdagi ma'lumotlarni samarali saqlash va boshqarish imkonini beradi. Ushbu maqolada biz Pythonda NumPy yordamida ko'p o'lchovli massivlarni yaratish va manipulyatsiya qilish asoslarini ko'rib chiqdik. Shuningdek, biz NumPy funksiyalaridan foydalangan holda ko'p o'lchovli massivlarda bajarishimiz mumkin bo'lgan ba'zi umumiy operatsiyalarni ko'rib chiqdik.

FOYDALANILGAN ADABIYOTLAR.

1. Tojimamatov, I., & Mirsiddiqova, M. (2025). BERILGANLAR BAZASIDA HAYOTIY SIKL. Модели и методы в современной науке, 4(6), 66-70.

2. Tojimamatov, I., & Siddiqova, G. (2025). TRANZAKSIYALARNI TAQSIMLANGAN TARZDA QAYTA ISHLASH MODELLARI. Современные подходы и новые исследования в современной науке, 4(6), 30-35.

3. Нурмаматович, Т. И., & Рахила, А. (2025). НА ОСНОВЕ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ ПОВЫШЕНИЕ УСТОЙЧИВОСТИ К ПОЛОМКАМ И АВАРИЯМ. YANGI O'ZBEKISTON, YANGI TADQIQOTLAR JURNALI, 2(8), 197-204.

4. Tojimamatov, I., & Ahmataliyeva, S. (2025). BERILGANLARNI MARKAZLASHGAN TARZDA BOSHQARISH TAMOYILLARI. Академические исследования в современной науке, 4(21), 59-64.

5. Tojimamatov, I., & Adxamova, C. (2025). AMALIY TIZIMLARDA BERILGANLAR BAZASINI BOSHQARISH TIZIMLARI O'RNINI. Академические исследования в современной науке, 4(21), 77-82.
6. Tojimamatov, I., & Fazliddinov, X. (2025). BERILGANLAR BAZASI ADMINISTRATORI VA UNING XUSUSIYATLARI. Академические исследования в современной науке, 4(21), 90-95.
7. Karimberdiyevich, O. M., Nurmamatovich, T. I., & Abdulaziz o'g'li, Y. M. (2024). BIG DATA SOHASIDAGI XALQARO LOYIHALAR. IZLANUVCHI, 1(1), 39-45.
8. Karimberdiyevich, O. M., Abdulaziz o'g'li, Y. M., & Zarifjon o'g'li, X. N. (2024). DATA MINING METODLARI VA BOSQICHLARI. YANGI O'ZBEKISTON, YANGI TADQIQOTLAR JURNALI, 1(4), 303-311.
9. Nurmamatovich, T. I. (2024). BERILGANLARNING TARMOQ MODELLARI: ODDIY VA MURAKKAB TARMOQ TUZILISHLARI.
10. <https://www.freecodecamp.org/news/multi-dimensional-arrays-in-python/>
11. https://www.w3schools.com/python/numpy/numpy_creating_arrays.asp
12. Array creation routines — NumPy v2.1 Manual