

Лабораторна робота № 13

Тема: Пакет NumPy. Основні принципи роботи.

Мета: Навчитися виконувати операції над масивами і матрицями, користуючись модулем NumPy (Numeric Python).

Література: <https://pythonworld.ru/numpy>

Зміст роботи:

Написати програми для рішення задач:

Завдання 1.

Задано 2 одновимірних масиви. Виконайте з ними всі арифметичні операції. Використовуючи метод конкатенації (об'єднання) масивів створіть новий масив з двох попередніх і знайдіть максимальний, мінімальний елемент, суму елементів та їх добуток. Результати всіх операцій вивести на екран, додаючи коментарі.

Завдання 2.

Задано одновимірний масив із 15 елементів. Сформувати новий масив в якому кожен елемент заданого масиву зменшити на середнє значення та відсортувати отриманий масив за зростанням. Результати всіх операцій вивести на екран.

Завдання 3.

Задано одновимірний масив з 20 елементів. Для ініціалізації використайте функцію `random()`. Перетворіть його у двовимірний. Кожен елемент масиву збільшити на 10. Результат виведіть у файл та на екран.

Завдання 4.

Задано двовимірний масив цілих чисел в діапазоні від -15 до 15. Створіть новий масив в якому всі числа менші 0 замінити на -1, більші 0 - на 1. Результати вивести на екран.

Завдання 5.

Застосуйте функції `sort()`, `min()`, `sum()`, `mean()` для двовимірного масиву розміром [4, 6]. Оцініть результат та опишіть його у звіті. Метод заповнення масиву даними виберіть на свій розсуд. Результати всіх операцій вивести на екран.

Методичні рекомендації

Головною особливістю NumPy є об'єкт *array* – масив, який подібний до списків Python. Однак з масивами можна виконувати числові операції з великими обсягами інформації в рази швидше і, головне, набагато ефективніше ніж із списками.

Приклади створення масивів:

```
import numpy as np
```

```
s=[ 1, 6, 8, 9, 10]
```

```
m=np.array(s)
```

```
print(m)
```

```
n=np.arange(0, 1, 0.2)
```

```
print(n)
```

```
import numpy as np

a = np.array([1, 2, 3, 4], float)
>>> a
array([ 1.,  2.,  3.,  4.])
>>> type(a)
<class 'numpy.ndarray'>
```

Для заповнення масивів випадковими числами використовуємо

np.random.rand(10) # одновимірний масив випадкових значень

```
array([0.06480376, 0.62333573, 0.0363841, 0.23728226, 0.58031617, 0.64754324,
0.87082152, 0.56534312, 0.40102058, 0.79272714])
```

np.random.rand(3, 4) # двовимірний масив випадкових значень

np.random.randint(10, size = 7) # одновимірний масив випадкових цілих чисел

```
array([0, 8, 1, 0, 0, 7, 3])
```

np.random.randint(10, size = (4, 4)) # двовимірний масив випадкових цілих чисел

Щоб відсортувати масиви скористаємось функцією **sort()**.

Для проходження по масиву використовуємо конструкції:

```
for x in np.nditer(b): # двовимірний масив
```

```
    print(x)
```

```
for y in b.flat: # двовимірний масив
```

```
    print(y)
```

```
for z in A: # одновимірний масив
```

```
    print(z)
```

```
for row in b:
```

```
    print('element', row) # друк по рядкам
```

Для роботи з двовимірним масивом доцільно скористатися вкладеними циклами.

Наприклад

```
import numpy as np
```

```
A = np.array([9, 1, 7, 6, 3, 9, 5, 2, 0])
```

```
b=A.reshape(3, 3)
```

```
for x in b:
```

```
    for y in x:
```

```
        print('element', y)
```

Щоб об'єднати масиви в один використовують функцію **numpy.concatenate**
`C=np.concatenate([a,d])`
Для роботи з файлами використовуємо функції **savetxt()** - запис у файл
loadtxt() - читання з файлу

Контрольні запитання.

1. Які недоліки і переваги імпортування модуля `numpy` з використання інструкцій `import numpy` і `from numpy import`.
2. Чому при ініціалізації об'єкту `array` кожен з елементів у списку повинен мати тип, вказаний другим аргументом при виклику функції?
3. Як працюють зрізи у багатовимірних масивах?
4. Як визначити кількість рядків і стовпців у матриці?
5. Базові операції над масивами
6. Які алгоритми сортування функція підтримує функція `sort`?
7. Який тип даних повертають оператори порівняння об'єктів типу `array`?
8. Як це використовується для фільтрування елементів масиву?
9. Назвіть функції для роботи з матрицями.