

# Лабораторна робота №1

## Базове знайомство з бібліотекою NumPy

**Мета роботи:** Ознайомитись з основними можливостями роботи з масивами бібліотеки NumPy, визначити статистичні характеристики.

### Короткі теоретичні відомості

Python можна завантажити на <https://www.python.org/downloads/>

Потрібно переконатись в наявності або додатково встановити бібліотеки NumPy, SciPy, matplotlib, seaborn, pandas, scikit-learn.

Можна скористатись дистрибутивом Anaconda, який містить усі бібліотеки, потрібні для роботи з даними: <https://www.anaconda.com/products/individual>

NumPy є базовою бібліотекою для представлення та обчислення даних в Python. Особливо цікавими є багатовимірні масиви ndarrays та похідні від них.

Бібліотеку NumPy зазвичай імпортують як np:

```
import numpy as np
```

Приклад створення масиву:

```
a = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
```

Типи даних в масивах можуть бути булевий тип (bool), цілий (int), цілий без знаку (uint), тип з плаваючою точкою (float), комплексний. При створенні масиву можна задавати тип даних, наприклад:

```
a=np.array([1, 2, 3, 4], dtype='float')
```

Автоматичне створення масивів:

1. Створення масиву, елементи якого починаються з 1, закінчуються 15 з кроком 2

```
a = np.arange(1,16,2)
```

2. Масив з дев'яти цілочислених одиниць:

```
np.ones(9, dtype=int)
```

3. Масив двовимірний, кожний рядок якого містить чотири нулі, тип даних з плаваючою точкою

```
np.zeros ((2, 4), dtype=float)
```

4. Масив з п'яти значень, розміщених рівномірно між 0 та 1.

```
np.linspace(0, 1, 5)
```

5. Масив 4x3 з рівномірно розподілених випадкових чисел від 0 до 1.

```
np.random.random((4, 3))
```

6. Масив 4x4 випадкових цілих чисел від 0 до 100.

```
np.random.randint(0, 100, (4, 4))
```

7. Порожній масив з 5 елементів. Насправді, його елементами буде випадковий вміст комірок пам'яті

```
np.empty(5)
```

Всі масиви мають наступні атрибути: `ndim` (розмірність), `shape` (розмір кожного виміру), `size` (загальний розмір масиву), `dtype` (тип елементів), `nbytes` (розмір масиву в байтах), `itemsize` (розмір елементу в байтах):

```
a = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
```

```
a.shape
```

Індексація елементів

Доступ до елементів за індексом

```
a[1], a[-2], a[0, -1]
```

Виділення підмасиву:

```
a[початок:кінець:крок]
```

За замовчанням початок=0, крок=1. Якщо вказано від'ємний крок, то елементи починаються з кінця і закінчуються початком.

Для багатовимірних масивів: `a[початок:кінець:крок, початок:кінець:крок]`

Підмасиви `NumPy` не є копіями даних масиву, а їх представленням. Тобто якщо змінити дані в підмасиві, то вони зміняться й у початковому масиві. Щоб створити саме копію, потрібно використати метод `copy()`.

Арифметичні операції з масивами

Додавання (+, `np.add`), віднімання (-, `np.subtract`), множення (\*, `np.multiply`), ділення (/, `np.divide`), піднесення до ступеню (\*\*, `np.power`), ділення за модулем (% , `np.mod`), зміна знаку на протилежний (-, `np.negative`).

```
a = np.ones(4)
```

```
a+3
```

```
a*2
```

Операція `reduce` багаторазово застосовує задану операцію до елементів масиву, поки не залишиться один результат.

```
a = np.arange(0, 6)
```

```
np.add.reduce(a)
```

accumulate працює аналогічно, але зберігає проміжні результати:

```
np.add.accumulate(a)
```

outer видає результат застосування операції до всіх пар елементів

```
a = np.arange(1, 10)
```

```
np.multiply.outer(a, a)
```

Виведення простих статистичних характеристик:

Сума значень `np.sum`, мінімальне та максимальне значення `np.min`, `np.max`, середнє значення `np.mean`, середньоквадратичне відхилення `np.std`, дисперсія `np.var`, медіана `np.median`, персентилі `np.percentile`.

Для читання `.csv` файлу можна скористатись функціями бібліотеки `pandas`

```
import pandas as pd
```

```
data = pd.read_csv('iris.csv')
```

```
petal = np.array(data['petal_length'])
```

### **Завдання до лабораторної роботи**

Створити програму, яка:

1. Генерує випадкові і не випадкові масиви різними способами, зазначеними в теоретичних відомостях.
2. Демонструє звернення до елементів масиву за допомогою індексів, в тому числі від'ємних; виділення підмасивів як одновимірних, так і багатовимірних масивів.
3. Демонструє основні арифметичні операції над масивами, а також роботу методів `reduce`, `accumulate`, `outer`.
4. Вираховує статистичні характеристики, а саме, мінімальне і максимальне значення, вибірккові середнє, дисперсію, середньоквадратичне відхилення, медіану та 25 та 75 персентилі, величини ширина пелюстки (`petal_width`) з набору даних щодо квіток ірису (`iris.csv`).

Оформити звіт. Звіт повинен містити:

- титульний лист;
- код програми;
- результати виконання коду;

Продемонструвати роботу програми та відповісти на питання стосовно теоретичних відомостей та роботи програми.