

# **ОСНОВИ СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ, НЕЙРОННИХ МЕРЕЖ ТА ГЛИБОКОГО НАВЧАННЯ**

## **Частина 6. ВИСОКОРІВНЕВА МОВА ПРОГРАМУВАННЯ PYTHON**

**Лекція 6.9. Базові бібліотечні модулі. Пакети  
для роботи зі штучним інтелектом.**

# Вбудовані модулі Python

Стандартний інтерпретатор Python має низку вбудованих модулів, що забезпечують взаємодію з оточенням.

## Системні:

- **sys** – забезпечує доступ до системно-залежних параметрів та функцій.
- **string** – загальні операції з рядками.
- **datetime** – операції обробки часу і дати.
- **calendar** – функції роботи з календарем.
- \*\*\*\*\*

## Математичні:

- **math (cmath)** – математичні функції (комплексні).
- **statistic** – функції математичної статистики.
- **random** – генерація випадкових чисел
- \*\*\*\*\*

# Вбудований модуль Sys

Модуль забезпечує доступ до змінних, що використовуються або підтримуються інтерпретатором, та до функцій, які сильно взаємодіють з інтерпретатором (> 100 функцій).

Типові функції:

- `sys.prefix` - директорія встановлення інтерпретатору python.
- `sys.path` - список шляхів пошуку модулів.
- `sys.setrecursionlimit` - встановити максимальну глибину рекурсії.
- `sys.__stdin__`, `sys.__stdout__`, - значення потоків вводу, виводу.
- `sys.version` - версія python.
- `sys.version_info` - кортеж з п'яти компонентів номеру версії.
- `sys.platform` - інформація про операційну систему.

# Вбудований модуль Math

Модуль забезпечує доступ до математичних функцій, визначених стандартом мови С.

Дійсні числа: **math**

Комплексні числа: **cmath**

Групи функцій:

- Степенні і логарифмічні функції.
- Тригонометричні функції.
- Гіперболічні функції.
- Спеціальні функції.
- Константи.
- Чисельне - теоретичні функції.

Посилання <https://docs.python.org/3.8/library/math.html>

# Пакети: SciPy : NumPy



**NumPy (нум пай) – n-вимірні масиви**

**SciPy (сай пай) – наукові обчислення**

**SymPy (сім пай) – символічні обчислення**

**Matplotlib - 2D графіка**

# NumPy

Відкрита бібліотека (пакет) розширення Python для підтримки великих багатовимірних масивів та матриць та виконання операцій з ними.

Модулі:

- `np.emath` – математичні функції,
- `np.random` – випадкові функції,
- `np.fft` – дискретне перетворення Фур'є,
- `np.linalg` – лінійна алгебра,
- `np.matlib` – матричні операції.

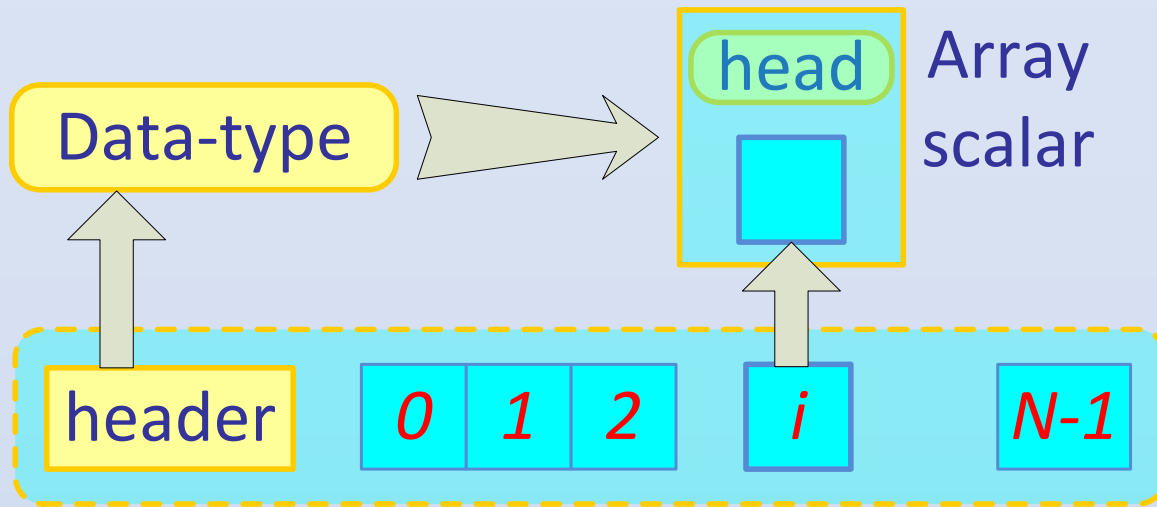
Функції:

- - сортування,
- - поліноми,
- - статистичні,
- - фінансові,
- - .....

Посилання <https://uk.wikipedia.org/wiki/NumPy>

Посилання <https://docs.scipy.org/doc/numpy/reference/index.html>

# NumPy



**ndarray** –  
колекція  
елементів  
одного типу.

**header** – вбудований об'єкт опису масиву.

Спосіб інтерпретації кожного елемента в масиві визначається окремим об'єктом – **data-type**.

Крім основних типів (`int`, `float`, ...), об'єкти типів даних також можуть представляти структури даних.

Вибраний елемент (за індексом) є об'єктом Python – `array_scalar`, вбудований у NumPy.

# NumPy. Типи елементів

## Базові array scalar типи:

- `int_` – цілочисловий тип,
- `float_` – тип з рухомою комою,
- `complex_` – комплексний тип,
- `bytes_` – байт тип,
- `unicode_` – символи юнікоду,
- `bool_` – логічний тип.

за замовчуванням `float_`

## Варіації array scalar типів:

<code>int8</code>	<code>uint8</code>	<code>float8</code>	<code>complex64</code>	<code>bool8</code>
<code>int16</code>	<code>uint16</code>	<code>float16</code>	<code>complex128</code>	
<code>int32</code>	<code>uint32</code>	<code>float32</code>	<code>complex192</code>	
<code>int64</code>	<code>uint64</code>	<code>float64</code>	<code>complex256</code>	
		<code>float128</code>		



# NumPy. Створення

## 5 базових механізмів створення масиву

- Внутрішній (arange, ones, ...).
- Перетворення з інших структур Python (list, ...).
- Зчитування з файлу .
- Формування з послідовності (strings, ...).
- З використанням спеціальних функцій (random,...).

## Внутрішній механізм:

- `array( object [,dtype, copy, order,...] )`
- `empty(shape[,dtype,order])` – пустий масив
- `ones(shape[,dtype, order])` – масив 1-ць
- `zeros(shape[,dtype, order])` – масив 0-ів
- `full(shape,fill_value[,dtype, order])` – масив заповнений значеннями `fill_value`

`shape` – кортеж, що визначає розмірність,

`dtype` – тип елементу,

`order` – порядок збереження.

# NumPy. Математичні операції

Функція	Повертає
<code>negative(x1, / [, ...])</code>	Поелементно $-X1$
<code>positive (x1, / [, ...])</code>	Поелементно $+X1$
<code>add(x1,x2, / [, ...])</code>	Поелементне $X1 + X2$
<code>subtract(x1,x2, / [, ...])</code>	Поелементне $X1 - X2$
<code>multiply(x1,x2, / [, ...])</code>	Поелементне $X1 * X2$
<code>divide(x1,x2, / [, ...])</code>	Поелементне $X1 / X2$
<code>remainder(x1,x2, / [, ...])</code>	Поелементне $X1 \% X2$
<code>mod(x1,x2, / [, ...])</code>	Поелементне $X1 \% X2$
*****	
<code>sign(x, / [, ...])</code>	Поелементне <b>знак</b> $X$
<code>power(x1,x2, / [, ...])</code>	Поелементне $X1 ** X2$

# NumPy. Математичні функції

Функція	Повертає
<code>exp(x, /[, ...])</code>	Поелементна експонента
<code>log(x, /[, ...])</code>	Поелементний логарифм
<code>sqrt(x, /[, ...])</code>	Поелементний корінь
<code>gcd(x1,x2, / [, ...])</code>	Поелементний НОД
ТИНОНОМЕТРИЧНІ	
<code>sin(x, /[, ...])</code>	Поелементне <i>sin()</i>
<code>asin(x, /[, ...])</code>	Поелементне <i>asin()</i>
ГИПЕРБОЛІЧНІ	
<code>sinh(x, /[, ...])</code>	Поелементне <i>sinh()</i>
<code>asinh(x, /[, ...])</code>	Поелементне <i>asinh()</i>
ПЕРЕТВОРЕННЯ	
<code>deg2rad(x, /[, ...])</code>	Поелементне <i>grad -&gt; rad</i>
<code>rad2deg(x, / [, ...])</code>	Поелементне <i>rad -&gt; grad</i>

# NumPy. Порівняння, логіка

Функція	Повертає
<code>greater(x1,x2, / [, ...])</code>	Поелементне <code>&gt;</code>
<code>greater_equal(x1,x2, / [, ...])</code>	Поелементне <code>&gt;=</code>
<code>less_equal(x1,x2, / [, ...])</code>	Поелементне <code>&lt;</code>
<code>not_equal(x1,x2, / [, ...])</code>	Поелементне <code>&lt;=</code>
<code>equal(x1,x2, / [, ...])</code>	Поелементне <code>==</code>
<code>greater(x1,x2, / [, ...])</code>	Поелементне <code>!=</code>
<code>logical_and(x1,x2, / [, ...])</code>	Поелементне <i>and</i>
<code>logical_or(x1,x2, / [, ...])</code>	Поелементне <i>or</i>
<code>logical_xor(x1,x2, / [, ...])</code>	Поелементне <i>xor</i>
<code>logical_not(x, / [, ...])</code>	Поелементне <i>not</i>

# NumPy. Модуль random

Функція	Повертає
<b>Integers (low [, high, size, dtype, endpoint])</b>	Випадкові цілі від <i>low</i> до <i>high</i> (виключно), або коли <i>endpoint=True</i> , до <i>high</i> (включно).
<b>random ([size, dtype, out])</b>	Випадкові з рухомою комою в інтервалі [0.0, 1.0).
<b>bytes(length)</b>	Випадкову послідовність байтів
<b>Більш 30 функцій розподілення :</b>	
<b>normal([loc, scale, size])</b>	Нормальне розподілення

# NumPy. Статистика ....

Функція	Повертає
<code>median(a[, axis, ...])</code>	Медіана масиву
<code>average(a[, axis, ...])</code>	Середня арифметична
<code>mean (a[, axis, ...])</code>	Мода масиву
<code>std (a[, axis, ...])</code>	Стандартне відхилення
<code>corcoef (x[, y, rowar, bias, ddof])</code>	Коефіцієнти кореляції
<code>correlate (x, v[, mode])</code>	Взаємна кореляція двох масивів
<code>histogram (a,[,bins,range, normed, ...])</code>	Гістограма розподілу значень в масиві

# NumPy. Сортування ....

Функція	Повертає
<code>sort(a[, axis, kind, order])</code>	Отсортовану копію <b>a</b>
*****	
<code>argmax(a[, axis, kind, order])</code>	Індекс максимального елемента <b>A</b>
<code>argmin(a[, axis, kind, order])</code>	Індекс мінімального елемента <b>A</b>
<code>where(condition, [x,y])</code>	Елемент <b>x</b> або <b>y</b> , враховуючи умову <b>condition</b>
<code>count_nonzero (a, [, axis])</code>	Кількість не нульових елементів масиву <b>a</b>

# NumPy. Модуль linalg

Функція	Повертає
<code>Dot(a,b[, out])</code>	Скалярне множення двох масивів <b>a</b> <b>b</b>
<code>matmul([x1,x2, /[, out, casting, order, ...])</code>	Множення матриць <b>x1</b> <b>x2</b>
<code>linalg.matrix_power(a,n)</code>	Степінь <b>n</b> матриці <b>a</b>
<code>linalg.norm(x[,ord,axis,keep])</code>	Норма масиву <b>x</b>
<code>linalg.solve(a,b)</code>	Рішення СЛАР $\mathbf{a} * \mathbf{x} = \mathbf{b}$

<https://docs.scipy.org/doc/numpy/reference/routines.linalg.html>



# SciPy

Відкрита бібліотека (пакет) високоякісних наукових та інженерних інструментів.

Модулі:

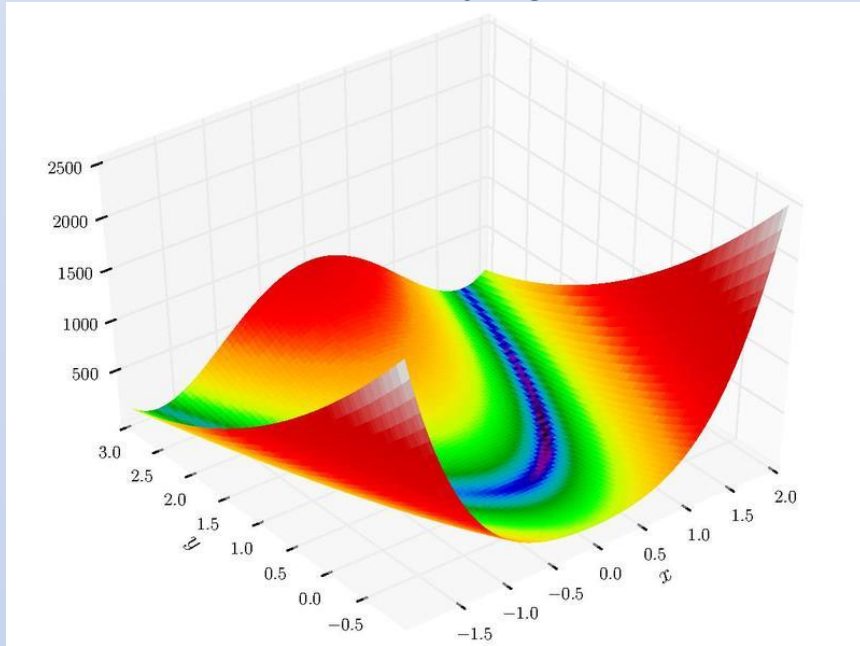
- `.constants` – фізичні константи,
- `.integrate` – інтегрування,
- `.optimize` – оптимізація,
- `.interpolate` – інтерполяція,
- `.fft` – перетворення Фурє,
- `.signal` – обробки сигналів,
- `.linalg` – лінійна алгебра,
- `.sparse` – розріджені матриці,
- `.spatial` – дерева, метрики,
- `.special` – спеціальні функції
- `.ndimage` – обробки зображень,
- `.stats` – статистичні функції.

# SciPy (приклад)

Функція `scipy.optimize.minimize()` – безумовна мінімізація функцій багатьох змінних.

Тестова функція Розенброка:

$$f(x) = \sum_{i=0}^{n-2} [100 * (x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$$



Мінімум = 0, коли  
всі  $x_i = 1$

# SymPy

Відкрита бібліотека для символічної математики. Повнофункціональна комп'ютерною алгебра (CAS).

Основні модулі:

- **Polynomials** – поліноми,
- **Solving equations** – рішення рівнянь,
- **Combinatorics** – комбінаторика,
- **Discrete math** – дискретна математика,
- **Matrix** – матричні операції,
- **Geometry** – геометричні обчислення,
- **Physics** – фізичні обчислення,
- **Statistics** – статистика,
- **Cryptography** – криптографія.

*... та багато іншого*

## Рекомендована ЛІТЕРАТУРА

- **Програмування числових методів мовою Python:** підруч. / А. В. Анісімов, А. Ю. Дорошенко, С. Д. Погорілий, Я. Ю. Дорогий ; за ред. А. В. Анісімова. – К. : Видавничо-поліграфічний центр "Київський університет", 2014. – 640 с.
- **Програмування числових методів мовою Python:** навч. посіб. / А. Ю. Дорошенко, С. Д. Погорілий, Я. Ю. Дорогий, Є. В. Глушко ; за ред. А. В. Анісімова. – К. : Видавничо-поліграфічний центр "Київський університет", 2013. – 463 с.
- **Основи програмування Python:** Підручник для студ. спеціальності 122 «Компютерні науки» / А.В.Яковенко; КПІ.- Київ: КПІ, 2018 . – 195 с.

# Посилання

- <https://scipy.org/>

## Контрольні запитання

- Визначте призначення і надайте призначення базових функцій модуля **sys**. Надайте приклади.
- Визначте призначення і надайте призначення базових функцій модуля **math**. Надайте приклади.
- Визначте призначення **numpy**. Поясніть принципи організації масивів в **numpy**.
- Поясніть основні механізми створення масивів в пакеті **numpy** . Надайте приклади.
- Надайте призначення модулів та функцій пакету **numpy**. Надайте приклади застосування.

**The END**

**Частина 6. Лекція 6.9.**