

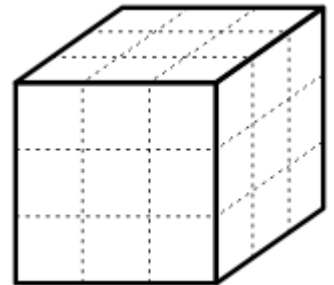
# АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРИЗОВАНИХ ІНФОРМАЦІЙНО-ВИМІРЮВАЛЬНИХ СИСТЕМ



# Лекція 13

## 2D та 3D масиви в Python

1. Вступ до масивів в Python
2. 2D масиви в Python
3. 3D масиви в Python



# 1. Вступ до масивів в Python

Масиви можна розглядати як списки в Python. Масив - це тип структури даних, який зберігає дані у суміжній пам'яті.

Дані, що зберігаються, мають той самий тип даних. Оскільки змінні можуть зберігати одне значення одночасно, масив може зберігати декілька значень одночасно. Для обробки даних масиву ми повинні зрозуміти, що таке індекс, що таке елемент, як обчислити довжину масиву, як отримати елемент у масиві, як додати або видалити елемент у масиві, як циклічно масив тощо.

- **Індекс:** це число, яке представляє значення масиву, і завжди починається з 0.
- **елемент:** це значення в масиві.
- **len ():** загальна кількість елементів у масиві.
- **append () :** це спосіб додати елемент до масиву.
- **remove ():** метод видалення елемента з масиву.

## 2. 2D масиви в Python

Масив називається двовимірним, якщо для задання положення елемента необхідно вказати значення двох індексів.

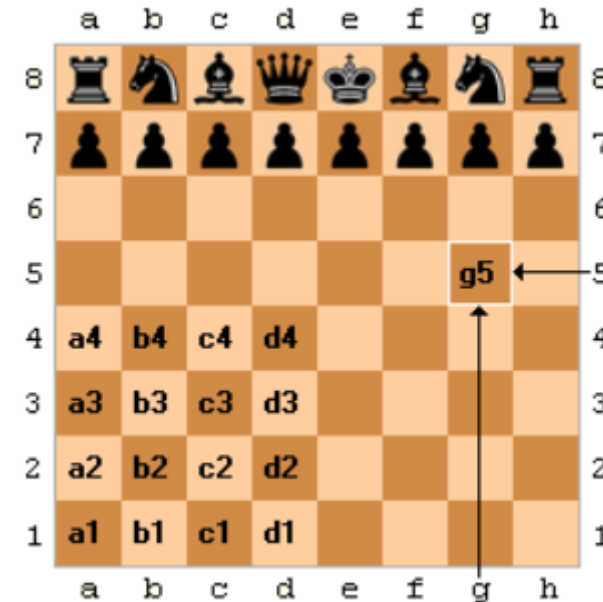
Двовимірні масиви ще називають *матрицями*.

У C++ *двовимірний масив* – це список *одновимірних масивів*.

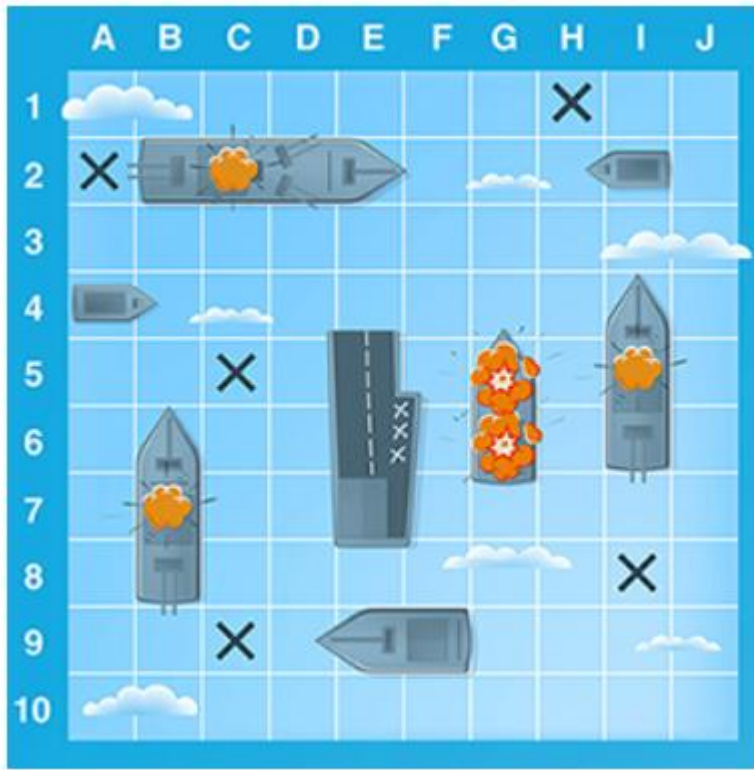
У реальному житті теж є приклади задання місця знаходження об'єкта за допомогою 2 координат.

На шаховій дошці положення фігури задається літерою (від *a* до *h*) та цифрою (від 1 до 8).

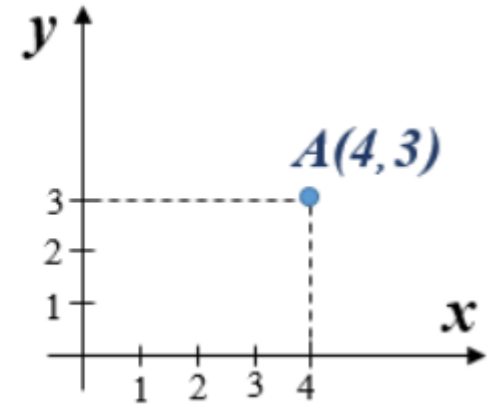
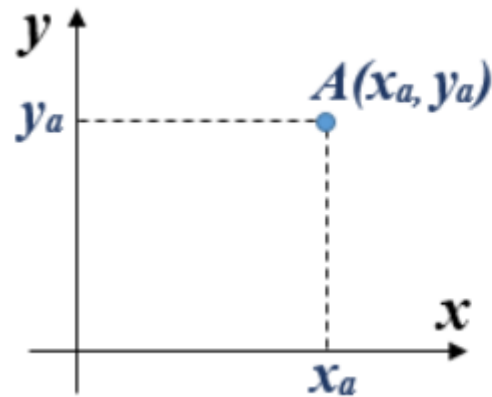
$a (0,1,2...10) \Rightarrow 1,2,3...10$



У грі “Морський бій” вказують місце розташування корабля або його частини літерою та числом.



Розміщення точки на площині у декартовій системі координат вказують 2 числа – абсциса та ордината.



## Загальний формат опису двовимірного масиву

тип ім'я\_масиву [кількість\_рядків] [кількість\_стовпців];

У вищенаведеному описі:

⊗ *тип* – це тип елементів масиву (він ще називається базовим типом);

⊗ *ім'я\_масиву* – ім'я змінної, за яким здійснюється доступ до елементів масиву.

### Приклад

Опис масиву з іменем *A*, що містить **3** рядки по **5** цілих чисел у кожному.

```
int A[3][5];
```



У С++ нумерація комірок починається з нуля. Всі клітинки об'єднані спільним іменем **A**.

A	0	1	2	3	4
0					
1					
2					

У масиві розмірністю **3x5** (3 рядки, 5 стовпців) рядки будуть нумеруватися від **0** до **2**, стовпці – від **0** до **4**.

Рядка з номером 3 та стовпця з номером 5 у даній матриці не буде.

## Розмір та ініціалізація двовимірного масиву

Після запису `int A[3][5];` у пам'яті комп'ютера виділяється  $3 \cdot 5 = 15$  комірок цілого типу *int*.

Якщо одна комірка типу *int* займає 2 байти, то всього буде виділено  $3 \cdot 5 \cdot 2 = 30$  байт пам'яті ( $3 \cdot 5 = 15$  клітинок по 2 байти кожна).

Двовимірні масиви можуть бути ініціалізовані під час опису.

# Приклад ініціалізації масиву $M$ цілих чисел розміром $3 \times 4$

```
int M[3][4] =  
{  
    {4, 16, -1, 31},  
    {-8, 12, 0, 17},  
    {34, 20, 8, -9}  
};
```

Зазвичай елементи масиву вводять із клавіатури або зчитують із файлів.

Питання введення та виведення елементів двовимірного масиву буде розглянуто пізніше.

# Доступ до елементів двовимірного масиву

Для доступу до елементів двовимірного масиву потрібно вказати два індекси:

- ⊗ перший задає номер рядка;
- ⊗ другий – номер стовпця комірки таблиці.

Щоб звернутись до деякого елементу масиву, потрібно після імені масиву в квадратних дужках вказати номер рядка та номер стовпця клітинки, в якій розміщується елемент.

Щоб не плутати, який індекс (номер рядка чи стовпця) писати першим, а який вказувати другим, можна скористатись наступної аналогією.

	0	1	2	3	4
0				12	
1					
2		-9			

У багатьох університетах номер аудиторії складається з номера поверху та номера кімнати на цьому поверсі. Наприклад, аудиторія № 513 – це кімната на 5-ому поверсі з номером 13. Ми спочатку повинні піднятись на потрібний поверх, а потім шукати необхідну кімнати в межах даного поверху.

Аналогічно клітинка матриці задається спочатку номером рядка (поверху), а потім номером клітинки (кімнати на цьому поверсі).

A	0	1	2	3	4
0					
1					
2					

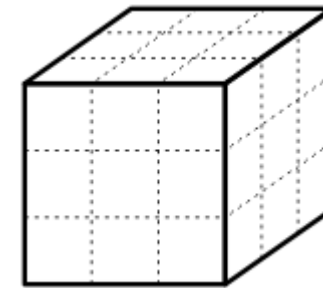
← A[номер рядка][номер стовпця]



← [номер поверху][номер кімнати]

# 3D масиви в Python

Тривимірний масив у Python - це структура даних, яка дозволяє організовувати дані у тривимірний блок, а не просто у двовимірну таблицю. Зазвичай ми працюємо з двовимірними масивами, які мають два індекси (рядок та стовпець), але у тривимірних масивах у нас є три індекси, які вказують на розташування у просторі. Ось детальний приклад тривимірного масиву в Python:



**Приклад:**

**# Використання вбудованих списків списків**

```
three_d_array = [[[0 for _ in range(3)] for _ in range(3)] for _ in range(3)]
```

**# Присвоєння значення деяким елементам тривимірного масиву**

```
three_d_array[0][0][0] = 1
```

```
three_d_array[1][1][1] = 42
```

**# Вивід значень елементів**

```
print("Значення елемента з індексами [0][0][0]:", three_d_array[0][0][0])
```

```
print("Значення елемента з індексами [1][1][1]:", three_d_array[1][1][1])
```

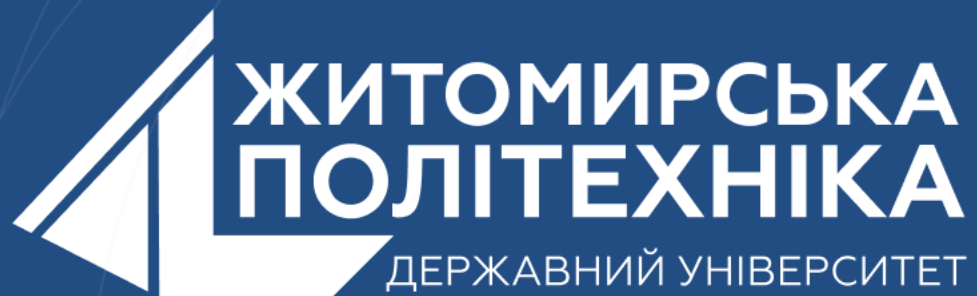


В цьому прикладі 'three\_d\_array' це тривимірний масив розміром 3x3x3. Ми використовуємо генератори списків для створення тривимірного масиву, заповненого нулями. Потім ми присвоюємо значення декільком елементам цього масиву, звертаючись до них за допомогою трьох індексів.

Також 'three\_d\_array[0][0][0]' має значення 1, оскільки його так задали. А 'three\_d\_array[1][1][1]' має значення 42.

Використовуючи тривимірні масиви, ви можете ефективно представляти дані, які мають тривимірну структуру, таку як об'єкти у тривимірному просторі.

   @ZTUEDUUA



- Розвиваємо лідерів
- Створюємо інновації
- Змінюємо світ на краще

