



28TECH  
Become A Better Developer

# MẢNG HAI CHIỀU



Trong Python để sử dụng mảng 2 hay nhiều chiều ta sử dụng nested list. Ở phần mảng 2 chiều này các bạn sẽ học về các bài toán cơ bản trên mảng 2 chiều như bài toán về hàng, cột, các phép toán trên ma trận, ma trận xoay ốc... Để giải quyết các bài toán liên quan tới ma trận các bạn cũng sử dụng nested list trong Python.

# 1. Tạo mảng 2 chiều và nhập xuất:



Để khai báo mảng 2 chiều, ta cần chỉ rõ số lượng hàng, số lượng cột của ma trận.

## Khai báo mảng 2 chiều có n hàng và m cột

### EXAMPLE

```
n, m = 2, 3
a = [[0 for _ in range(m)] for _ in range(n)]
for i in range(n):
    a[i] = list(map(int, input().split()))
for i in range(n):
    for j in range(m):
        print(a[i][j], end = ' ')
    print()
```

#### INPUT

```
1 2 3
4 5 6
```

#### OUTPUT

```
1 2 3
4 5 6
```

## 2. Truy cập vào các phần tử trong mảng 2 chiều:



Để truy cập vào phần tử trong mảng, các bạn dùng chỉ số hàng và chỉ số cột. Chỉ số hàng và cột của mảng 2 chiều được đánh số từ 0 tương tự như mảng 1 chiều.

	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9

`a[0][2] = 3`

`a[2][1] = 8`

### 3. Các bài toán trên mảng 2 chiều:

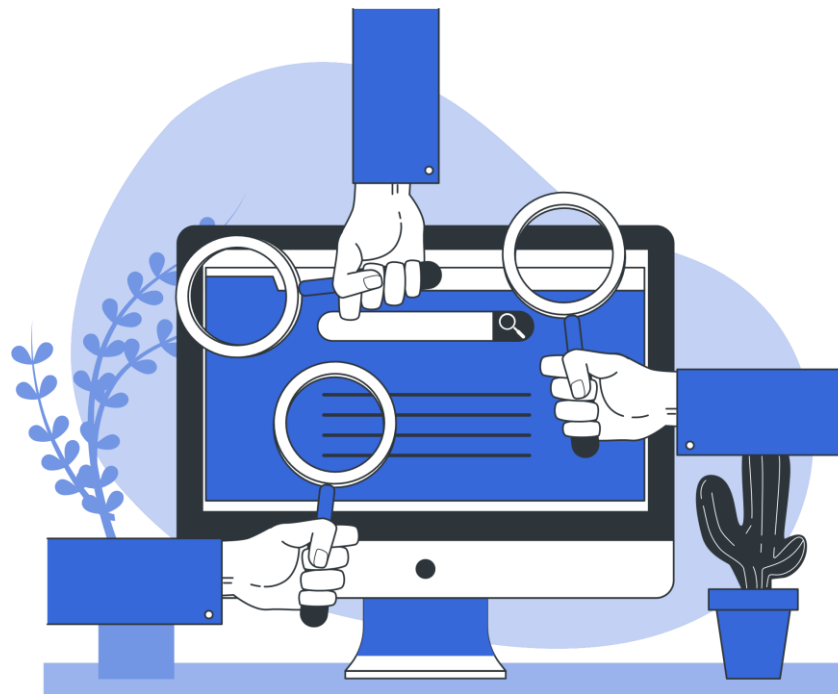
#### a. Tìm phần tử lớn nhất, nhỏ nhất trong mảng:

##### EXAMPLE

```
a = [[1, 2, 5], [3, 1, 0], [4, 1, 5]]  
min_val, max_val = 10**18, -10**18  
for row in a:  
    min_val = min(min_val, min(row))  
    max_val = max(max_val, max(row))  
print(min_val, max_val)
```

**OUTPUT**

0 5



### 3. Các bài toán trên mảng 2 chiều:

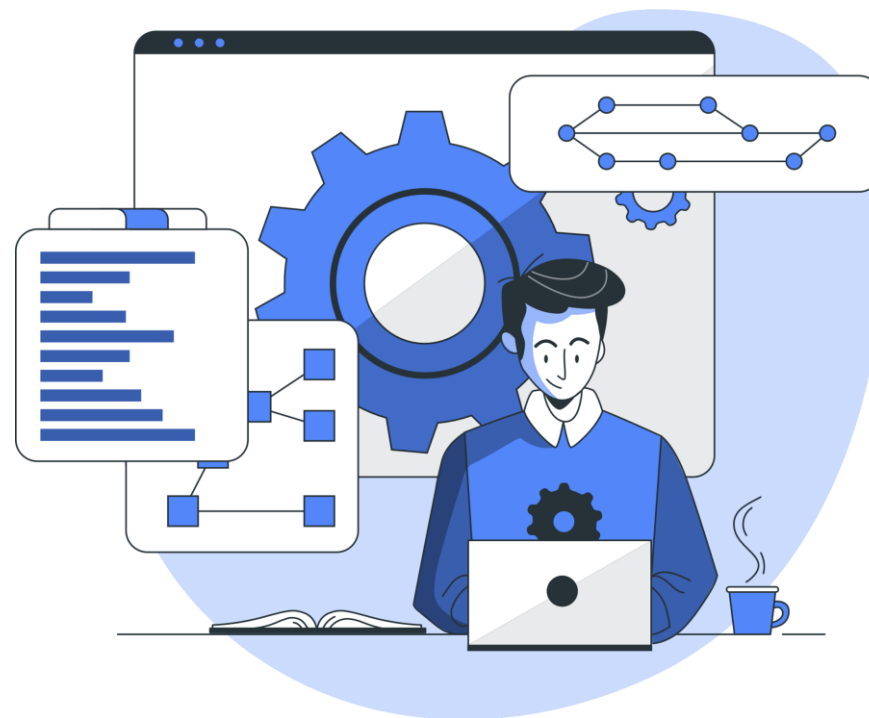
#### b. Flatten mảng 2 chiều thành mảng 1 chiều:

##### EXAMPLE

```
a = [[1, 2, 5], [3, 1, 0], [4, 1, 5]]  
b = [x for small_list in a for x in small_list]  
print(min(b))  
print(max(b))  
print(b)
```

##### OUTPUT

```
0  
5  
[1, 2, 5, 3, 1, 0, 4, 1, 5]
```



### 3. Các bài toán trên mảng 2 chiều:

#### c. Tính tổng từng hàng của ma trận (mảng 2 chiều):

##### EXAMPLE

```
a = [[1, 2, 5], [3, 1, 0], [4, 1, 5]]  
sum_row = [sum(row) for row in a]  
print(sum_row)
```

**OUTPUT**

8 4 10



## 4. Các phép toán trên ma trận:

### a. Tạo ma trận chuyển vị:



Ma trận chuyển vị của ma trận  $a$  cỡ  $n \times m$  là ma trận  $b$  có cỡ  $m \times n$ , trong đó các hàng của ma trận  $a$  trở thành các cột của ma trận  $b$ .

#### EXAMPLE

```
a = [[1, 2, 5],  
      [3, 1, 0],  
      [4, 1, 5]]  
a_t = [[row[i] for row in a] for i in range(len(a))]  
print(a_t)
```

#### OUTPUT

```
[[1, 3, 4],  
 [2, 1, 1],  
 [5, 0, 5]]
```



## 4. Các phép toán trên ma trận:

### b. Cộng trừ hai ma trận:



Trong đại số tuyến tính, **ma trận tương tự như một mảng 2 chiều** gồm n hàng và m cột. Để 2 ma trận có thể cộng hoặc trừ cho nhau thì chúng phải **có cùng số hàng và số cột**.

1	2	0
0	4	1

+

1	4	8
9	2	3

=

2	6	8
9	6	4

1	2	0
0	4	1

-

1	4	8
9	2	3

=

0	-2	-8
-9	2	-2

## 4. Các phép toán trên ma trận:

### b. Cộng trừ hai ma trận:

Cộng 2 ma trận cùng  
cỡ n hàng m cột

EXAMPLE

```
n, m = 3, 3
a = [[1, 2, 5], [3, 1, 0], [4, 1, 5]]
b = [[0, 1, 2], [1, 1, 1], [0, 1, 0]]
c = [[0 for _ in range(m)] for _ in range(n)]
for i in range(n):
    for j in range(m):
        c[i][j] = a[i][j] + b[i][j]
print(c)
```

OUTPUT

```
[[1, 3, 7], [4, 2, 1], [4, 2, 5]]
```

Trừ 2 ma trận cùng  
cỡ n hàng m cột

EXAMPLE

```
n, m = 3, 3
a = [[1, 2, 5], [3, 1, 0], [4, 1, 5]]
b = [[0, 1, 2], [1, 1, 1], [0, 1, 0]]
c = [[0 for _ in range(m)] for _ in range(n)]
for i in range(n):
    for j in range(m):
        c[i][j] = a[i][j] - b[i][j]
print(c)
```

OUTPUT

```
[[1, 1, 3], [2, 0, -1], [4, 0, 5]]
```

## 4. Các phép toán trên ma trận:

### c. Nhân hai ma trận:



Giả sử có 2 ma trận a cỡ  $n \times m$ , ma trận b cỡ  $p \times q$ , để ma trận a có thể **nhân** với ma trận b thì **số cột của ma trận a**, tức là **m phải bằng số hàng của ma trận b**, tức là p.

$$a[n][m] \times b[p][q] = c[n][q]$$



Khi đó **m = p** thì ma trận tích của a với b sẽ là **ma trận c có cỡ  $n \times q$** . Phần tử ở chỉ số (i, j) của ma trận tích c được tính bằng cách nhân **từng cặp phần tử ở hàng i của ma trận a với các phần tử ở cột j của ma trận b**.

## 4. Các phép toán trên ma trận:

### c. Nhân hai ma trận:

#### Chương trình mẫu

##### EXAMPLE

```
n, m, p = map(int, input().split())
a, b = [0] * n, [0] * m
for i in range(n):
    a[i] = list(map(int, input().split()))
for i in range(m):
    b[i] = list(map(int, input().split()))
c = [[0 for _ in range(n)] for _ in range(p)]
for i in range(n):
    for j in range(p):
        for k in range(m):
            c[i][j] += a[i][k] * b[k][j]
print(c)
```

## 5. Kỹ thuật duyệt các ô liền kề:

$i-1, j-1$	$i-1, j$	$i-1, j+1$
$i, j-1$	$i, j$	$i, j+1$
$i+1, j-1$	$i+1, j$	$i+1, j+1$

## 5. Kỹ thuật duyệt các ô liền kề:

Duyệt 4 ô chung cạnh với ô (i,j)

EXAMPLE

```
a = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
]
path = [[-1, 0], [0, -1], [0, 1], [1, 0]]
i, j = 1, 1 # 5
for x in path:
    i1, j1 = i + x[0], j + x[1]
    print(a[i1][j1], end = ' ')
```

OUTPUT

2 4 6 8

1	2	3
4	5	6
7	8	9

## 5. Kỹ thuật duyệt các ô liền kề:

Duyệt 8 ô chung đỉnh với ô (i,j)

EXAMPLE

```
a = [  
    [1, 2, 3],  
    [4, 5, 6],  
    [7, 8, 9]  
]  
path = [[-1, -1], [-1, 0], [-1, 1], [0, -1], [0, 1], [1, -1], [1, 0], [1, 1]]  
i, j = 1, 1 # 5  
for x in path:  
    i1, j1 = i + x[0], j + x[1]  
    print(a[i1][j1], end = ' ')
```

1	2	3
4	5	6
7	8	9

OUTPUT

1 2 3 4 6 7 8 9

## 5. Kỹ thuật duyệt các ô liền kề:

Duyệt 8 ô xung quanh nước đi của quân mã

EXAMPLE

```
a = [  
    [1, 2, 3, 4, 1],  
    [5, 6, 7, 8, 0],  
    [9, 3, 2, 6, 0],  
    [1, 2, 1, 4, 4],  
    [1, 2, 3, 5, 3]  
]
```

```
path = [[-2, -1], [-2, 1], [-1, -2], [-1, 2], [1, -2], [1, 2], [2, -1], [2, 1]]  
i, j = 2, 2  
for x in path:  
    i1, j1 = i + x[0], j + x[1]  
    print(a[i1][j1], end = ' ')
```

1	2	3	4	1
5	6	7	8	0
9	3	2	6	0
1	2	1	4	4
1	2	3	5	3

OUTPUT

2 4 5 0 1 4 2 5