

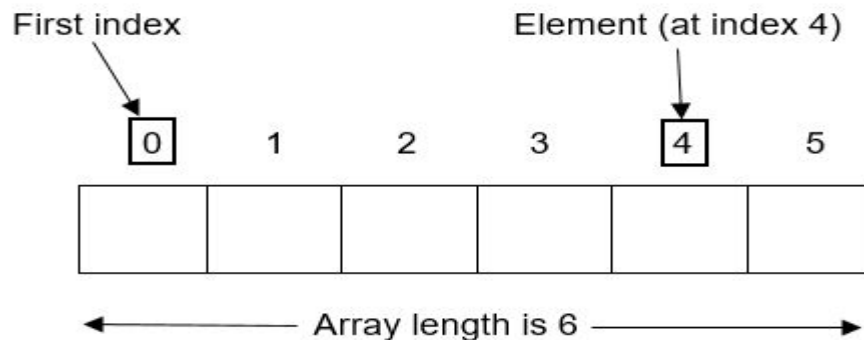
---

# Array

# Mảng (Array)

Mảng là tập hợp nhiều phần tử có cùng kiểu dữ liệu và mỗi phần tử trong mảng được truy xuất thông qua các chỉ số của nó trong mảng

Mảng trong java lưu các phần tử theo chỉ số, chỉ số của phần tử đầu tiên là 0



# Mảng (Array)



Mảng có hai loại:



Mảng một chiều



Mảng đa chiều

# Mảng một chiều



Khai báo mảng:

```
<Kiểu dữ liệu> <Tên mảng>[ ];
```

Hoặc:

```
<Kiểu dữ liệu>[ ] <Tên mảng>;
```

Cú pháp cấp phát bộ nhớ cho mảng:

```
<Tên mảng> = new <kiểu dữ liệu>[<Kích thước mảng>];
```

```
int[] a;  
a = new int[3];  
a[0] = 5;  
a[2] = 1;  
a[1] = 2;  
System.out.println("Mang a: ");  
for (int i=0; i<a.length; i++){  
    System.out.println(a[i]);  
}
```

Khai báo mảng a có kiểu dữ liệu là int

Mảng a có kích thước là 3

Gán giá trị cho mảng a

Lấy kích thước mảng

Mang a:

5  
2  
1

# Exercise



Viết chương trình tạo mảng số nguyên gồm  $n$  phần tử ( $n$  nhập vào từ bàn phím), thực hiện:

- Nhập phần tử cho mảng và in mảng ra màn hình
- Hiển thị phần tử tại vị trí  $\text{index} = 2$
- Tính tổng các phần tử trong mảng
- In ra màn hình các số chẵn và tổng các số đó
- Sắp xếp mảng theo thứ tự tăng dần
- Sắp xếp mảng theo thứ tự giảm dần

# Exercise



Tạo mảng chuỗi gồm  $n$  phần tử ( $n$  nhập vào từ bàn phím)

- Nhập thông tin cho các phần tử
- In thông tin ra màn hình
- Đếm số lần “Java” xuất hiện trong mảng
- Nhập vào từ bàn phím chuỗi bất kỳ, kiểm tra chuỗi đó ở vị trí nào của mảng



## Mảng đa chiều

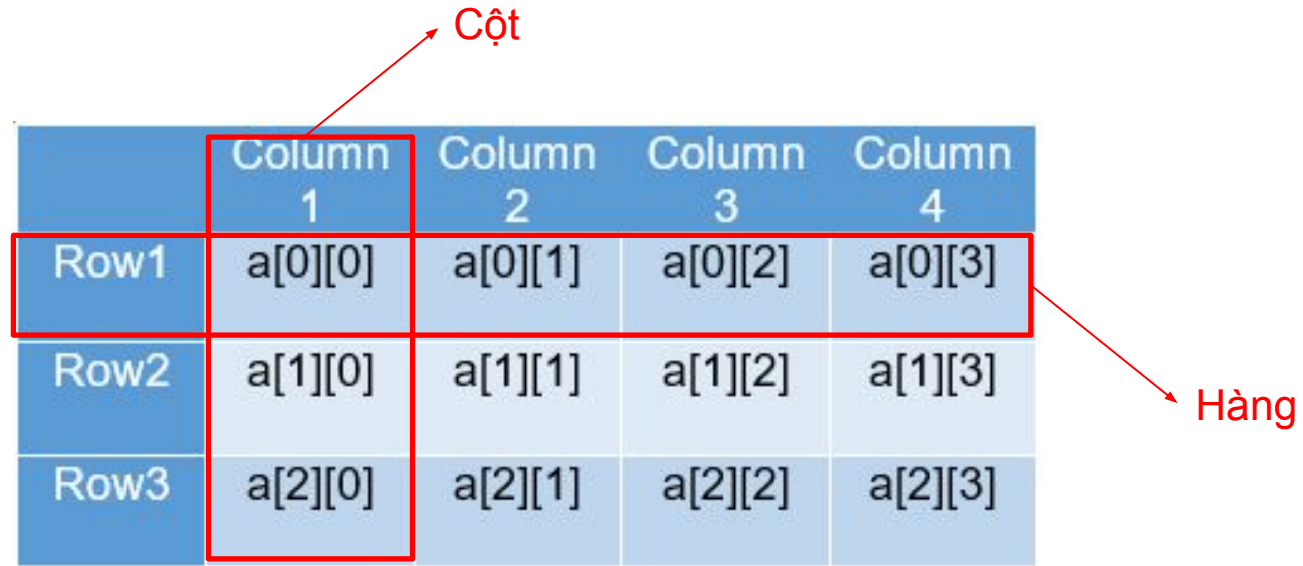
Mảng đa chiều chỉ là tăng số chiều lưu trữ nhiều chiều hơn, hay còn gọi là ma trận. Thông thường ta hay sử dụng mảng 2 chiều.

Đối với mảng 2 chiều, dữ liệu được lưu trữ theo hai chiều. Chiều thứ nhất gọi là hàng và chiều thứ hai gọi là cột.

Trong nội dung này chúng ta chủ yếu tìm hiểu về mảng 2 chiều



# Mảng đa chiều



A diagram illustrating a 2D array structure. The array is represented as a table with 3 rows and 4 columns. The columns are labeled 'Column 1', 'Column 2', 'Column 3', and 'Column 4'. The rows are labeled 'Row1', 'Row2', and 'Row3'. The elements are indexed as `a[row][column]`. A red box highlights the first column, labeled 'Cột' (Column) in red. Another red box highlights the first row, labeled 'Hàng' (Row) in red.

	Column 1	Column 2	Column 3	Column 4
Row1	<code>a[0][0]</code>	<code>a[0][1]</code>	<code>a[0][2]</code>	<code>a[0][3]</code>
Row2	<code>a[1][0]</code>	<code>a[1][1]</code>	<code>a[1][2]</code>	<code>a[1][3]</code>
Row3	<code>a[2][0]</code>	<code>a[2][1]</code>	<code>a[2][2]</code>	<code>a[2][3]</code>

# Mảng đa chiều



Khai báo mảng:

```
<Kiểu dữ liệu> <Tên mảng>[ ][];
```

Hoặc:

```
<Kiểu dữ liệu>[ ][] <Tên mảng>;
```

Cú pháp cấp phát bộ nhớ cho mảng:

```
<Tên mảng> = new <kiểu dữ liệu>[<Số dòng>][<Số cột>;
```

```
int[][] a= {{1,2,3},{4,5,6},{7,8,9}};  
for (int i=0; i<3; i++){  
    for(int j=0; j<3; j++){  
        System.out.print(a[i][j]+" ");  
    }  
    System.out.println();  
}
```

Sử dụng hai vòng lặp để  
in thông tin mảng đa chiều

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

# Exercise



1 Viết chương trình nhập vào 2 ma trận vuông A và B , in mảng đó ra màn hình.

- Thực hiện cộng 2 ma trận
- Tìm ma trận chuyển vị của 2 ma trận đó

2 Viết chương trình nhập vào 1 mảng đa chiều, in mảng đó ra màn hình. Tính tổng các phần tử chia hết cho 5 trong mảng đó



## For each

Vòng lặp for each chủ yếu được sử dụng để duyệt mảng hoặc các phần tử của collection. Với for each, thay vì khai báo hay khởi tạo biến lặp vị trí, chúng ta sẽ khai báo một biến chung kiểu dữ liệu của mảng, sử dụng biến đó để duyệt các phần tử của mảng mà không cần lấy vị trí (index) của mỗi phần tử

# For each



Cú pháp:

```
for(<Kiểu dữ liệu> <Tên biến chạy>: <Tên mảng>){  
    <Khối lệnh lặp lại>;  
}
```

```
int[] a;  
a = new int[3];  
a[0] = 5;  
a[1] = 2;  
a[2] = 1; Tên biến chạy  
  
System.out.println("Mang a: ");  
for (int i: a) {  
    System.out.println(i);  
}
```

```
5  
2  
1
```