

# Bài: Mảng 2 chiều trong C#

Xem bài học trên website để ủng hộ Kteam: [Mảng 2 chiều trong C#](#)

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage [How Kteam](#) nhé!

## Dẫn nhập

Ở các bài học trước, chúng ta đã cùng nhau tìm hiểu về [MẢNG MỘT CHIỀU TRONG C#](#). Hôm nay chúng ta sẽ cùng tìm hiểu về **mảng hai chiều trong C#**.

## Nội dung

Để đọc hiểu bài này tốt nhất các bạn nên có kiến thức cơ bản về các phần:

- [CẤU TRÚC CƠ BẢN MỘT CHƯƠNG TRÌNH TRONG C# console application](#)
- [BIẾN](#) và [KIỂU DỮ LIỆU](#) trong C#
- [TOÁN TỬ TRONG C#](#)
- [CÁU ĐIỀU KIỆN TRONG C#](#)
- [CẤU TRÚC CƠ BẢN VÒNG LẶP](#)
- [CẤU TRÚC CƠ BẢN CỦA HÀM TRONG C#](#)

Trong bài học này, chúng ta sẽ cùng tìm hiểu các vấn đề:

- Khai báo, khởi tạo mảng 2 chiều trong C#
- Sử dụng mảng 2 chiều trong C#

## Khai báo, khởi tạo mảng 2 chiều trong C#

### Khai báo mảng 2 chiều


**Mảng 2 chiều** được hình dung như một bảng có **m** dòng và **n** cột với một số đặc trưng sau:

- Mảng 2 chiều mang những đặc trưng cơ bản của một mảng bình thường (đặc trưng về mảng đã trình bày trong bài [MẢNG 1 CHIỀU TRONG C#](#))
- Các phần tử trong mảng 2 chiều được truy xuất thông qua 2 chỉ số phần tử (tạm gọi là chỉ số dòng và chỉ số cột)
- Hình ảnh minh họa mảng 2: Giả sử ta có mảng A có 5 dòng và 8 cột

chỉ số cột N

	0	1	2	3	4	5	6	7
0	A[0,0]	A[0,1]	A[0,2]	A[0,3]	A[0,4]	A[0,5]	A[0,6]	A[0,7]
1	A[1,0]	A[1,1]	A[1,2]	A[1,3]	A[1,4]	A[1,5]	A[1,6]	A[1,7]
2	A[2,0]	A[2,1]	A[2,2]	A[2,3]	A[2,4]	A[2,5]	A[2,6]	A[2,7]
3	A[3,0]	A[3,1]	A[3,2]	A[3,3]	A[3,4]	A[3,5]	A[3,6]	A[3,7]
4	A[4,0]	A[4,1]	A[4,2]	A[4,3]	A[4,4]	A[4,5]	A[4,6]	A[4,7]

chỉ số dòng M



- Các dòng và các cột được đánh số từ 0 và tăng dần. Mỗi phần tử là giao nhau của dòng và cột tương ứng đồng thời ta sử dụng chỉ số dòng cột đó để truy xuất đến phần tử của mảng 2 chiều.
- **Ví dụ:** A[1, 2] là cách truy xuất đến phần tử ở dòng thứ 2 cột thứ 3 (do chỉ số được đánh số từ 0)

### Cú pháp:

```
<kiểu dữ liệu> [ , ] <tên mảng>;
```

Trong đó:

- <kiểu dữ liệu> là kiểu dữ liệu của các phần tử trong mảng.
- Cặp dấu [ , ] là ký hiệu cho khai báo mảng 2 chiều.
- <tên mảng> là tên của mảng, cách đặt tên mảng cũng như cách đặt tên biến (quy tắc đặt tên biến đã trình bày trong [BIẾN TRONG C#](#)).

Để sử dụng được mảng ta phải khởi tạo giá trị hoặc cấp phát vùng nhớ cho mảng.

### Cấp phát vùng nhớ

Được thực hiện thông qua toán tử `new` (đã trình bày trong bài [TOÁN TỬ TRONG C#](#)). Lưu ý là khi cấp phát vùng nhớ cho mảng 2 chiều ta cần chỉ ra số dòng và số cột tối đa của mảng.

#### Ví dụ:

C#:

```
/*
    * Khai báo mảng 2 chiều kiểu string và có tên là Kteam.
    * Sau đó thực hiện cấp phát vùng nhớ với số dòng là 2 và số cột là 3.
    */
string[,] Kteam = new string[2, 3];
```

Sau khi mảng được cấp phát vùng nhớ thì các phần tử trong mảng sẽ mang giá trị mặc định:

- Đối với số nguyên là 0
- Đối với số thực là 0.0
- Đối với kiểu ký tự là "" (ký tự rỗng)
- Đối với kiểu tham chiếu là `null`

Chúng ta có thể khởi tạo giá trị khác mà chúng ta mong muốn ngay khi cấp phát vùng nhớ bằng cú pháp sau:

```
<kiểu dữ liệu>[,] <tên mảng> = new <kiểu dữ liệu>[]
{
    { <giá trị dòng 1 cột 1>, ..., <giá trị dòng 1 cột n> },
    ...
    { <giá trị dòng m cột 1>, ..., <giá trị dòng m cột n> }
};
```

- Vì đây là mảng 2 chiều nên chúng ta sẽ không khởi tạo giống mảng 1 chiều được mà phải khởi tạo giá trị theo từng dòng một. Xem từng dòng là một mảng 1 chiều và khởi tạo như mảng 1 chiều.
- Các giá trị khởi tạo nằm trong cặp dấu ngoặc nhọn `{ }` và cách nhau bởi dấu phẩy.

- Chúng ta không cần cung cấp số dòng và số cột tối đa mà trình biên dịch sẽ tự đếm xem bạn đã khởi tạo bao nhiêu dòng và mỗi dòng bao nhiêu giá trị rồi xem nó như số dòng số cột tối đa.

## Khởi tạo giá trị

### Cú pháp:

```
<kiểu dữ liệu>[,] <tên mảng> =
{
    { <giá trị dòng 1 cột 1>, ..., <giá trị dòng 1 cột n> },
    ...
    { <giá trị dòng m cột 1>, ..., <giá trị dòng m cột n> }
};
```

### Ví dụ:

#### C#:

```
int[,] IntArray =
{
    {1, 2},
    {3, 4},
    {5, 6}
};
```

Về bản chất thì cách này trình biên dịch vẫn xem xét số phần tử khởi tạo và cấp phát vùng nhớ cho biến mảng sau đó thực khởi tạo giá trị cho các phần tử trong mảng. Nhưng cách viết này có vẻ nhanh và gọn hơn so với cách cấp phát vùng nhớ rồi mới khởi tạo giá trị.

Tóm lại, cũng như mảng 1 chiều, mảng 2 chiều cũng có 3 cách khai báo và khởi tạo sau:

- Khai báo và cấp phát vùng nhớ

#### C#:

```
string[,] Array = new string[2, 3];
```

- Khai báo, cấp phát và khởi tạo giá trị cho mảng

#### C#:

```
string[,] Kteam = new string[,]
{
    { "HowKteam", "Free Education" },
    { "HowKteam.com", "Share to be better" }
};
```

- Khởi tạo giá trị cho mảng

#### C#:

```
int[] IntArray =
{
    {1, 2},
    {3, 4},
    {5, 6}
};
```

## Sử dụng mảng 2 chiều

Tương tự như mảng 1 chiều, kiểu mảng 2 chiều cũng có thể dùng làm:

- Kiểu dữ liệu cho biến.
- Kiểu trả về cho hàm.
- Tham số truyền vào cho hàm.
  - Các phần tử của mảng được truy xuất thông qua chỉ số dòng và chỉ số cột (chỉ số dòng viết trước, chỉ số cột viết sau ngăn cách nhau bởi dấu ,) và cặp dấu []. Có thể xem các phần tử của mảng như là các biến đơn và thao tác như thao tác với biến bình thường.


**C#:**

```
// Khai báo, cấp phát và khởi tạo mảng 2 chiều kiểu string và tên là Kteam
int[,] Kteam =
{
    {1, 2},
    {3, 4},
    {5, 6}
};

/*
 * Vì chỉ số phần tử được đánh số từ 0 nên muốn truy xuất đến phần tử dòng 1 cột 2 của mảng
 thì chỉ số dòng là 0 và chỉ số cột là 1
 */
Console.WriteLine(Kteam[0, 1]);
```

Một số thuộc tính và phương thức đặc trưng của mảng 2 chiều:

Tên thuộc tính hoặc phương thức	Ý nghĩa
Length	Thuộc tính trả về số nguyên kiểu <code>int</code> là số phần tử tối đa của mảng (số phần tử của mảng 2 chiều là tích số dòng và số cột của mảng)
LongLength	Tương tự như thuộc tính Length nhưng trả về số nguyên kiểu <code>long</code>
GetLength(<số chiều>)	Trả về số nguyên kiểu <code>int</code> là số phần tử trong chiều đã xác định. Lưu ý chiều của mảng là các số nguyên và được đánh số từ 0. Đối với mảng 2 chiều thì GetLength(0) là độ dài chiều đầu tiên tương ứng với số dòng và GetLength(1) là độ dài chiều thứ 2 tương ứng với số cột.
GetLongLength(<số chiều>)	Tương tự GetLength nhưng trả về số nguyên kiểu <code>long</code>

Rank	Thuộc tính trả về số nguyên int là số chiều của mảng
 Clone()	Thực hiện copy giá trị của mảng ra một vùng nhớ mới (phép gán thông thường thì 2 đối tượng sẽ dùng chung vùng nhớ rất nguy hiểm vì đối tượng này thay đổi dẫn đến đối tượng kia cũng thay đổi)

Còn rất nhiều thuộc tính và phương thức khác, mình chỉ giới thiệu một số cái hay dùng còn lại các bạn có thể tự khám phá.

## Cách duyệt mảng 2 chiều:

### Ý tưởng:

- Tương tự như ý tưởng duyệt mảng 1 chiều (đã trình bày trong bài [MẢNG MỘT CHIỀU TRONG C#](#))
- Nhưng do mảng 2 chiều có 2 chỉ số là chỉ số dòng và chỉ số cột nên chúng ta cần 2 vòng lặp lồng vào nhau.

### Ví dụ:


#### C#:

```
int[,] IntArray = new int [9, 10];
/*
 * Sử dụng 2 vòng for lồng vào nhau để duyệt mảng 2 chiều
 * Vòng lặp ngoài là vòng lặp duyệt mỗi dòng của của mảng 2 chiều
 * Với mỗi dòng thì vòng lặp trong là vòng lặp duyệt các phần tử trên dòng đó (duyet từng cột trên dòng hiện tại)
 */
for (int i = 0; i < 9; i++)
{
    for (int j = 0; j < 10; j++)
    {
        /*
         * Với cách duyệt này thì IntArray[i, j] sẽ là phần tử hiện tại mình đang xét
         * Code xử lý sẽ viết ở đây
         */
    }
}
```

Hình ảnh minh họa:

Biến **i** trong vòng lặp  
đầu tiên tượng trưng cho  
chỉ số dòng của mảng

IntArray[**i**, **j**]

 Biến **j** trong vòng lặp  
thứ hai tượng trưng cho  
chỉ số cột của mảng

- Cách duyệt này sẽ duyệt tuần tự các dòng trong mảng 2 chiều, ở mỗi dòng sẽ duyệt từ đầu dòng đến cuối dòng. Bạn hoàn toàn có thể duyệt theo ý mình bằng cách thay đổi giá trị trong vòng lặp.
- Ví dụ sau sẽ duyệt theo cột. Tức là duyệt tuần tự từng cột rồi ở mỗi cột duyệt từ trên xuống dưới.

#### C#:

```

/*
 * Duyệt mảng 2 chiều theo cột
 * Các bạn để ý sự thay đổi trong 2 vòng lặp
 */
for (int i = 0; i < 10; i++)
{
    for (int j = 0; j < 9; j++)
    {
        /*
         * Lưu ý là các phần tử được truy xuất là IntArray[j, i] thay vì IntArray[i, j]
         * Code xử lý
         */
    }
}

```

- Để ý là cả 2 cách duyệt mình đều vi phạm 1 điều mà ở phần duyệt mảng 1 chiều của bài trước mình đã trình bày. Đó là nên sử dụng hàm trả về số dòng, số cột thay vì viết cứng một con số cụ thể.
- Lúc này hàm **GetLength(<số chiều>)** thực sự phát huy tác dụng. Các bạn cùng xem nhé:

**C#:**

```

/*
 * Như đã trình bày ở phần trước thì:
 * GetLength(0) sẽ trả về số dòng của mảng 2 chiều
 * GetLength(1) sẽ trả về số cột của mảng 2 chiều
 */
for (int i = 0; i < IntArray.GetLength(0); i++)
{
    for (int j = 0; j < IntArray.GetLength(1); j++)
    {
        // Code xử lý
    }
}

```

## Ví dụ chương trình sử dụng mảng 2 chiều

Ta thử xét 1 ví dụ đơn giản đó là viết chương trình cho phép nhập vào giá trị số nguyên cho 1 mảng 2 chiều bất kỳ sau đó in ra màn hình mảng đã nhập kèm theo tổng tất cả các giá trị trong mảng.

Chương trình minh họa:

**C#:**

```

Console.Write(" Moi ban nhap so dong cua mang: ");
int Rows = int.Parse(Console.ReadLine());
Console.Write(" Moi ban nhap so cot cua mang: ");
int Columns = int.Parse(Console.ReadLine());

// Tạo 1 mảng 2 chiều với số dòng và số cột đã nhập
int[,] IntArray = new int[Rows, Columns];
/*
 * Duyệt mảng để nhập giá trị cho các phần tử
 * Ở đây mình muốn minh họa cách sử dụng mảng nên mình bỏ qua các bước kiểm tra dữ liệu mà ép kiểu trực tiếp
 * Điều này có thể gây lỗi khi nhập sai nên các bạn hãy cải tiến chương trình này cho đầy đủ nhé!
 */
for (int i = 0; i < IntArray.GetLength(0); i++)
{
    for (int j = 0; j < IntArray.GetLength(1); j++)
    {
        Console.Write(" Moi ban nhap phan tu IntArray[{0}, {1}] = ", i, j);
        IntArray[i, j] = int.Parse(Console.ReadLine());
    }
}

/*
 * In mảng 2 chiều đã nhập ra màn hình
 * Để tính tổng các giá trị trong mảng ta chỉ cần duyệt qua các phần tử và cộng chúng lại với nhau
 * Tận dụng lúc duyệt mảng để in giá trị ta sẽ thực hiện tính tổng luôn để tránh phải duyệt lại mảng thêm lần nữa.
 */
int Sum = 0;

Console.WriteLine("\n Mang ban vua nhap la: ");
for (int i = 0; i < IntArray.GetLength(0); i++)
{
    for (int j = 0; j < IntArray.GetLength(1); j++)
    {
        Console.Write(IntArray[i, j] + " ");
        Sum = Sum + IntArray[i, j];
    }
    // Sau khi in xong mỗi dòng ta thực hiện xuống dòng rồi mới in tiếp
    Console.WriteLine();
}
Console.WriteLine(" Tong cac gia tri trong mang: " + Sum);

```

Kết quả sau khi chạy chương trình trên:

```

file:///C:/Users/HT/documents/visual studio 2010/Projects/Mang2Chieu
Moi ban nhap so dong cua mang: 3
Moi ban nhap so cot cua mang: 3
Moi ban nhap phan tu IntArray[0, 0] = 1
Moi ban nhap phan tu IntArray[0, 1] = 2
Moi ban nhap phan tu IntArray[0, 2] = 3
Moi ban nhap phan tu IntArray[1, 0] = 4
Moi ban nhap phan tu IntArray[1, 1] = 5
Moi ban nhap phan tu IntArray[1, 2] = 6
Moi ban nhap phan tu IntArray[2, 0] = 7
Moi ban nhap phan tu IntArray[2, 1] = 8
Moi ban nhap phan tu IntArray[2, 2] = 9

Mang ban vua nhap la:
1 2 3
4 5 6
7 8 9
Tong cac gia tri trong mang: 45

```

## Kết luận

Nội dung bài này giúp các bạn nắm được:

- Khai báo, khởi tạo mảng 2 chiều trong C#.
- Sử dụng mảng 2 chiều trong C#.

Bài sau chúng ta sẽ tìm hiểu về [MẢNG NHIỀU CHIỀU TRONG C#](#).

Cảm ơn các bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên "**Luyện tập – Thử thách – Không ngại khó**".

