

## 2.4 C 语言中的二维数组\_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 2.4 C 语言中的二维数组涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

一维数组是相同数据类型元素的集合，但是只能表示一行数据。若是存在行和列相关的信息 (例如矩阵)，我们就需要用二维数组来表示。

数据类型 数组名 [行数][列数];

示例用法：

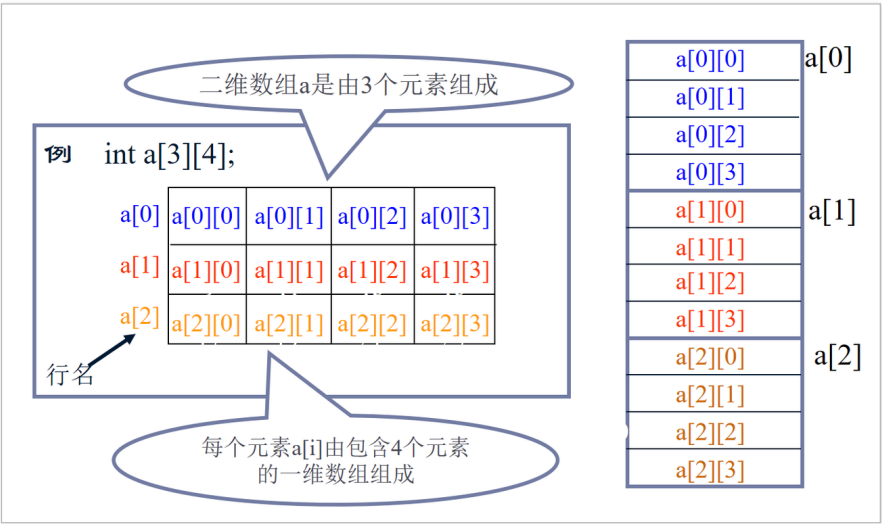
```
int a[3][4];
```

在内存中按照还是按照一维数组的顺序排序的。  
只不过，为了方便人们识别，我们是按照二维的来理解。

0xd10	1	a[0][0]
0xd14	2	a[0][1]
0xd18	3	a[0][2]
0xd1c	4	a[0][3]
0xd20	5	a[1][0]
0xd24	6	a[1][1]
0xd28	7	a[1][2]
0xd2c	8	a[1][3]
0xd30	9	a[2][0]
0xd34	10	a[2][1]
0xd38	11	a[2][2]
0xd3c	12	a[2][3]

实际我们这里来写 int a[3][4];

	0	1	2	3
0	a[0][0]	a[0][1]	a[0][2]	a[0][3]
1	a[1][0]	a[1][1]	a[1][2]	a[1][3]
2	a[2][0]	a[2][1]	a[2][2]	a[2][3]



```
int a[3][2];
```

[1]a的类型: int [3][2]  
[2]元素的表达式:a[0][0],a[0][1],a[0][2] ...a[2][3]

[3]元素的个数: 行数 \* 列数  
 [4]数组的大小: 元素的个数 \* 一个元素的大小<==>6 \* sizeof(a[0][0])<==>sizeof(a)  
 [5]数组最后一个元素: a[行数 - 1][列数 - 1]  
 [6]内存的存放方式: 按行优先存放.  
 [7]定义二维数组的时候, 行数可以省略不写, 系统会根据默认  
 初始化元素的个数来分配对应的内存空间。但是列数一定要写。  
 (因为二维数组默认按行来进行优先存放的)

例如:  
 int a[3][4];  
 int a[][4] = {1,2,3,4,5,6,7,8};

### 示例代码

```
#include <stdio.h>

int main()
{
    int a[3][2];
    int i = 0, j = 0;

    printf("please input %d data : ", 3 * 2);

    for(i = 0; i < 3; i++)
    {
        for(j = 0; j < 2; j++)
        {
            scanf("%d", &a[i][j]);
        }
    }

    for(i = 0; i < 3; i++)
    {
        for(j = 0; j < 2; j++)
        {
            printf("%d ", a[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

### 运行结果:

```
please input 6 data : 10 20 30 40 50 60
10 20
30 40
50 60
```

```
int a[3][3] = {1,2,3,4,5,6,7,8,9};
int b[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
```

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

### 示例代码:

```
#include <stdio.h>

int main()
{
    int a[3][2] = {{1,2},{3},{5,6}};
    int i = 0, j = 0;

    printf("please ouput %d data : \n", 3 * 2);

    for(i = 0; i < 3; i++)
    {
```

```
for(j = 0; j < 2; j++)  
{  
    printf("%d ", a[i][j]);  
}  
printf("\n");  
}  
return 0;  
}
```

运行结果:

```
please ouput 6 data :  
1 2  
3 0  
5 6
```

定义一个int a[10][10]的数组, 利用二维数组输出下列杨辉三角的图案.

```
1  
1 1  
1 2 1  
1 3 3 1  
1 4 6 4 1  
...
```

---

全文完

本文由 简悦 SimpRead 优化, 用以提升阅读体验

使用了 全新的简悦词法分析引擎 beta, 点击查看详细说明

