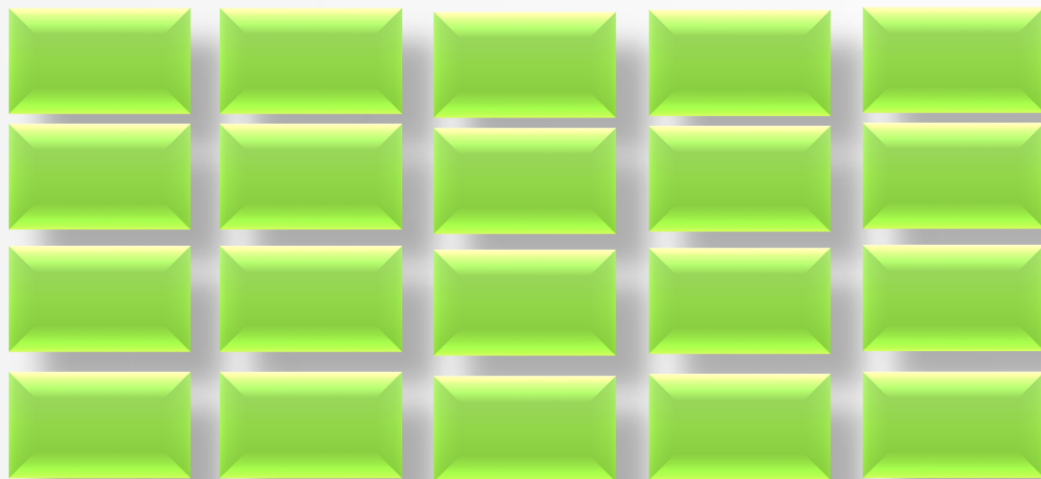


二维数组与多维数组

ARRAY STRUCTURE 4 数组与结构



一
维
数
组



二
维
数
组

一维数组的元素是一个一维数组——二维数组

一维数组的元素是一个二维数组——三维数组

一维数组的元素是一个 $n-1$ 维数组—— n 维数组

类型定义符 数组名 [元素个数1] [元素个数2]... [元素个数n];

二维数组:

```
int two[10][20];  
two[2][8]=1;
```

类型定义符 数组名 [元素个数1] [元素个数2]... [元素个数n];

三维数组:

```
int student[4][5][3];  
student[i][j][k]=3;
```

4.2 二维数组

二维数组的存储

一个两行三列的数组的逻辑存储结构

	第0列	第1列	第2列
第0行	<code>a[0][0]</code>	<code>a[0][1]</code>	<code>a[0][2]</code>
第1行	<code>a[1][0]</code>	<code>a[1][1]</code>	<code>a[1][2]</code>

二维数组的存储空间大小：
数据类型大小 * 一维数组元素数量 * 二维数组元素数量



4.2 二维数组

二维数组的初始化

按元素初始化

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

按行初始化

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

省略行数的初始化

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

4.2 二维数组

▣ 二维数组的初始化

部分元素初始化 •

```
int a[3][3] = { 1,2}
```

按行部分初始化 •

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

省略行数的部分初始化 •

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





二维数组的初始化

1	2	3
4	5	0
6	0	0
0	0	0

【例】以下程序的运行结果是什么？

```
int main()
{
    int a[][3]={{1,2,3},{4,5},{6},{0}};
    printf("%d,%d,%d\n",a[1][1],a[2][1],a[3][1]);
    return 0;
}
```

结果： , ,



二维数组的初始化

【例】若 `int a[][3]={1, 2, 3, 4, 5, 6, 7}` ,
则a数组的第一维大小是多少？

1	2	3
4	5	6
7	0	0