# 一维数组

## 一维数组的声明

**数组：类型相同的数据元素的集合。**

这个类型可以是基本的数据类型，也可以是“构造类型” 。

**这些数据元素只能顺序的存放在内存的某段区域。**

**格式：** **类型** **数组名** **[**数组大小**];**

定义一个数组就等于同时定义多个变量。这些元素（变量）的名字：

**数组名[下标];** // 下标就是一个**整数**

例如: **int** **a[10];** //**这一行代码声明了一个 10个元素的一维整型数组。**

**相当于声明了10个变量,变量名分别为**

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

**注意 : 下标从0 开始, 共10 个变量 , 最大下标为 9**

## 一维数组的初始化

初始化方法一：**全部初始化**

**int** **arr[5]** **=** **{11,** **12,** **13,** **14,** **15};**

**注意：**

1. 初值的个数不能大于数组的长度

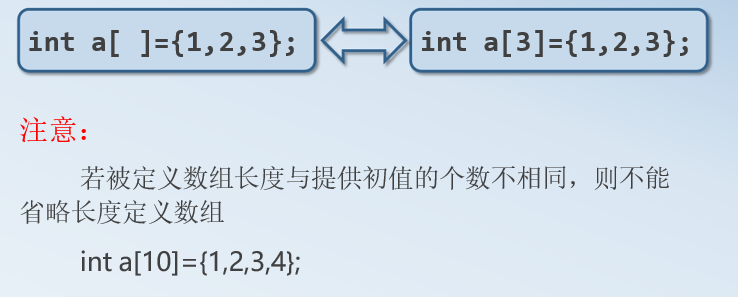
int a[3]={1,2,3,4,5};（这种初始化是不正确的）

1. 每个初值的类型最好与数组的类型一致

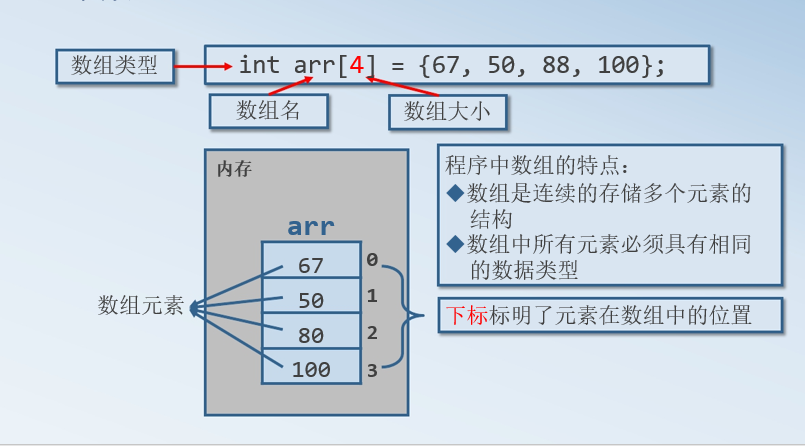
初始化方法二：**部分元素赋初值**

**int** **b[5]={1,2};//** b[0]=1 b[1]=2b [2]～b[4]都为0 未赋值的均为0

初始化方法三：省略长度赋初值



## 一维数组的存储原理



## **一维数组元素的引用**

**初始化数组**

**for(i=0;i<N;i++)**

**a[i]=0;**

**读入数据到数组**

**for(i=0;i<N;i++)**

**scanf(“%d”,&a[i]);**

**数组元素求和**

**for(i=0;i<N;i++)**

**sum += a[i];**

## **一维数组的注意事项**

**数组中必须存放同类型的对象**

**数组名的命名规则与普通标识符的命名规则相同**

**数组中对象可以是数值、字符、指针、结构体等类型**

**数组中的元素还可以是另外一个数组**

**数组中的元素地址在内存中是连续的**

# 二维数组

## 二维数组的声明

某个一维数组的每一个元素都是一个一维数组。这种数组我们叫做**二维数组**。

**定义** **类型名 数组名[ 行长度 ][ 列长度 ]**

**• 行长度和列长度：整型常量表达式**

**• 列长度不能为空**

例： int a[2][3];// **定义一个含有2行3列的整型数组 a**

|  |  |  |  |
| --- | --- | --- | --- |
| **第一行** | **a[0][0]** | **a[0][1]** | **a[0][2]** |
| **第二行** | **a[1][0]** | **a[1][1]** | **a[1][2]** |
|  | **第一列** | **第二列** | **第三列** |

## 二维数组的存储

**二维数组按行存放，先行后列**

**所以需要知道数组的每一行 有多少列，这样从起始地址开始，才能正确的读出数组的元素**

## 二维数组的初始化

**分行给二维数组赋初值**

int a[3][4]={{1, 2, 3, 4},{5, 6, 7, 8},{9,10,11,12}};

**对部分元素赋初值**

int a[3][4]={{1},{0,6},{9}};

 int b[3][4]={{1},{5,6}};

 int c[3][4]={{1},{},{0,0,11}};

**省略行长度**

 **对全部元素都赋了初值，**

• int a[ ][3]={1,2,3,4,5,6,7,8,9};

 **或分行赋初值时，在初值表中列出了全部行**

• int b[ ][3]={{1,2,3},{2},{4,5},{3}};

**错误赋值方式**

 int a[2][3]={{1,2},4,5,6};

**建议不要省略行长度**

**例7-4 请观察下面的声明语句。**

 int a1[3][4] = {{1,2,3,4}, {5,6,7,8},{9, 10, 11, 12}};

 int a2[3][4] = {1,2,3,4,5,6,7,8,9,10,11,12};

 int a3[3][3] = {{1, 2}, {3}};

 int a4[3][3] = {1, 2, 0, 3};

 int a5[][3] = {{1, 2, 3}, {4, 5}};

 int a6[][3] = {1, 2, 3, 4, 5, 0};

 char a7[3][5] ={{‘a’,‘b’,‘c’,‘d’, ‘\0’},{‘e’,‘f’,‘g’,‘h’,‘\0’},{‘i’,‘j’,‘k’,‘l’, ‘\0’}};

 char a8[3][5] ={“abce”, “efgh”, “ijkl”};

# **代码练习**

**详情请见C语言程序手册**