

数组

概念

数组是一组**相同类型元素**的集合

可以存放一个或多个数据，但是不能是零个

分为**一维数组**和**多维数组**

一维数组

数组的创建

```
//创建一个数组，存放班级中50个人的数学成绩
int math[50];
```

数组的初始化

```
//完全初始化，完完全全塞满整个数组
int array[5] = { 1,2,3,4,5 };
```

```
//数组的不完全初始化，剩下的默认初始化为0
int a[5] = { 1 };
```

数组的下标 & 使用

可以将下标理解成**门牌号**

数组通过下标访问具体元素，下标也叫**索引**

整形数组

```
int main() {
    int a[5] = { 1,2,3,4,5 };
    //      0 1 2 3 4
    printf("%d", a[0]);
    printf("%d", a[1]);
    printf("%d", a[2]);
    printf("%d", a[3]);
    printf("%d", a[4]);
    a[2] = 30;           //重新为a[2]赋值
    printf(" %d",a[2]);
    return 0;
}
输出结果：
12345 30
```

```
//通过循环输出
int main() {
    int a[5] = { 1,2,3,4,5 };
    for ( int i = 0; i < 5; i++)
    {
        printf("%d ", a[i]);
    }
    return 0;
}
```

```
//输入数据
//整型数组 的操作是逐个元素去处理的，没办法一次性输入，也没办法一次性输出，必须使用循环
int main() {
    int a[] = { 1,2,3,4,5 };
    for (int i = 0; i < 5; i++)
    {
        scanf("%d",&a[i]);           //输入，这是一个数据元素，所以要取地址
    }
    for ( int i = 0; i < 5; i++)
    {
        printf("%d ", a[i]);
    }
    return 0;
}
```

整型数组 的操作是逐个元素去处理的，没办法一次性输入，也没办法一次性输出，必须使用循环

字符数组

一维数组中的所有元素在内存中是**连续存放**的，随着数组下标的增长而增长

数组在内存中的关系

```
#include <stdio.h>
int main() {
    int arr[5] = { 1,2,3,4,5 };
    for (int i = 0; i < 5; i++)
    {
        printf("arr[%d] = %p\n", i, &arr[i]);
    }
    return 0;
}
```

输出结果：

```
//x86架构
arr[0] = 010FFD8C      //每次加4
arr[1] = 010FFD90
arr[2] = 010FFD94
arr[3] = 010FFD98
arr[4] = 010FFD9C
```

//x64架构

```
arr[0] = 0000009FE80FFCA8  
arr[1] = 0000009FE80FFCAC  
arr[2] = 0000009FE80FFCB0  
arr[3] = 0000009FE80FFCB4  
arr[4] = 0000009FE80FFCB8
```

x86和x64位的差别是一个是32位一个是64位

sizeof计算数组元素个数

```
#include <stdio.h>  
int main() {  
    int a[10] = { 0 };  
    printf("数组的总长度为: %zu\n", sizeof(a));           //计算数组a的总长度, 10  
    * 4 = 40 字节  
    printf("数组的个数为: %zu\n", sizeof(a) / sizeof(a[0])); //数组的总长度 ÷ 单  
    个数组元素的长度  
    return 0;  
}
```

输出结果:

数组的总长度为: 40

数组的个数为: 10