C 语言数组的概念

我们举一个例子，是输出一个 4×4 的整数矩阵，代码如下：

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int a1=20, a2=345, a3=700, a4=22;
    int b1=56720, b2=9999, b3=20098, b4=2;
    int c1=233, c2=205, c3=1, c4=6666;
    int d1=34, d2=0, d3=23, d4=23006783;

    printf("%-9d %-9d %-9d %-9d\n", a1, a2, a3, a4);
    printf("%-9d %-9d %-9d %-9d\n", b1, b2, b3, b4);
    printf("%-9d %-9d %-9d %-9d\n", c1, c2, c3, c4);
    printf("%-9d %-9d %-9d %-9d\n", d1, d2, d3, d4);

    system("pause");
    return 0;
}
```

运行结果：

```
20        345       700       22
56720     9999      20098     2
233       205       1         6666
34        0         23        23006783
```

矩阵共有 16 个整数，我们为每个整数定义了一个变量，也就是 16 个变量。那么，为了减少变量的数量，让开发更有效率，能不能为多个数据定义一个变量呢？比如，把每一行的整数放在一个变量里面，或者把 16 个整数全部都放在一个变量里面。

我们知道，要想把数据放入内存，必须先要分配内存空间。放入 4 个整数，就得分配 4 个 int 类型的内存空间：

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

这样，就在内存中分配了 4 个 int 类型的内存空间，共 4×4=16 个字节，并为它们起了一个名字，叫 a。

我们把这样的一组数据的集合称为数组（Array），它所包含的每一个数据叫做数组元素（Element），所包含的数据的个数称为数组长度（Length），例如 int a[4];就定义了一个长度为 4 的整型数组，名字是 a。

数组中的每个元素都有一个序号，这个序号从 0 开始，而不是从我们熟悉的 1 开始，称为下标（Index）。使用数组元素时，指明下标即可，形式为：

```
arrayName[index]
```

arrayName 为数组名称，index 为下标。例如，a[0] 表示第 0 个元素，a[3] 表示第 3 个元素。

接下来我们就把第一行的 4 个整数放入数组：

```
a[0]=20;
a[1]=345;
a[2]=700;
a[3]=22;
```

这里的 0、1、2、3 就是数组下标，a[0]、a[1]、a[2]、a[3] 就是数组元素。

我们来总结一下数组的定义方式：

```
dataType  arrayName[length];
```

dataType 为数据类型，arrayName 为数组名称，length 为数组长度。例如：

```
float m[12];
```

```
char ch[9];
```

注意：
1) 数组中每个元素的数据类型必须相同，对于 int a[4];，每个元素都必须为 int。

2) 数组下标必须是整数，取值范围为 0 ≤ index < length。

3) 数组是一个整体，它的内存是连续的，下面是 int a[4];的内存示意图：

数组的初始化

上面的代码是先定义数组再给数组赋值，我们也可以在定义数组的同时赋值：

```
int a[4] = {20, 345, 700, 22};
```

{ }中的值即为各元素的初值，各值之间用,间隔。

对数组赋初值需要注意以下几点：
1) 可以只给部分元素赋初值。当{ }中值的个数少于元素个数时，只给前面部分元素赋值。例如：

```
int a[10]={12, 19, 22 , 993, 344};
```

表示只给 a[0]~a[4] 5 个元素赋值，而后面 5 个元素自动赋 0 值。

当赋值的元素少于数组总体元素的时候，剩余的元素自动初始化为 0：对于 short、int、long，就是整数 0；对于 char，就是字符 '\0'；对于 float、double，就是小数 0.0。

我们可以通过下面的形式将数组的所有元素初始化为 0：

```
int a[10] = {0};
char c[10] = {0};
float f[10] = {0};
```

由于剩余的元素会自动初始化为 0，所以只需要给第 0 个元素赋 0 值即可。

示例：输出数组元素。

```
#include <stdio.h>
int main()
{
    int a[6] = {299, 34, 92, 100};
    int b[6], i;
    //从控制台输入数据为每个元素赋值
    for(i=0; i<6; i++){
        scanf("%d", &b[i]);
    }
    //输出数组元素
    for(i=0; i<6; i++){
        printf("%d  ", a[i]);
    }
    putchar('\n');
    for(i=0; i<6; i++){
        printf("%d  ", b[i]);
    }
    putchar('\n');

    return 0;
}
```

运行结果：

```
90 100 33 22 568 10
299  34  92  100  0  0
90  100  33  22  568  10
```

2) 只能给元素逐个赋值，不能给数组整体赋值。例如给十个元素全部赋 1 值，只能写为：

```
int a[10]={1, 1, 1, 1, 1, 1, 1, 1, 1, 1};
```

而不能写为：

```
int a[10]=1;
```

3) 如给全部元素赋值，那么在数组定义时可以不给出数组的长度。例如：

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

等价于

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

最后，我们借助数组来输出一个 4×4 的矩阵：

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int a[4] = {20, 345, 700, 22};
    int b[4] = {56720, 9999, 20098, 2};
    int c[4] = {233, 205, 1, 6666};
    int d[4] = {34, 0, 23, 23006783};

    printf("%-9d %-9d %-9d %-9d\n", a[0], a[1], a[2], a[3]);
    printf("%-9d %-9d %-9d %-9d\n", b[0], b[1], b[2], b[3]);
    printf("%-9d %-9d %-9d %-9d\n", c[0], c[1], c[2], c[3]);
    printf("%-9d %-9d %-9d %-9d\n", d[0], d[1], d[2], d[3]);

    system("pause");
    return 0;
}
```