

# 第五章：数组

## 教学内容：

1. 数组的创建和声明
2. 数组的初始化和元素
3. 一维数组
4. 二维数组

## 一、 数组的创建和声明

定义：数组是一个长度固定的数据结构，它存储多个相同类型的数值。数组直接被 Java 编程语言所支持，数组也是一个对象(引用数据类型)。数组是一个 Object 类的一个不明显的扩展。

数组的特征：有序、具有相同类型、通过数组名和下标确定数组中的元素。

数组的声明：

a) 数组的元素类型     数组变量名[];

b) 数组的元素类型[]   数组变量名;

创建数组空间：创建数组空间可以在声明的时候进行创建，也可以先声明然后在创建空间。

数组名=new     数组元素类型[数组的长度];

```
int[] arr = new int[10];
```

等同于下面的

```
int[] arr;  
arr=new int[10];
```

## 二、 数组的初始化和元素

数组中的每个元素都是有先后顺序的，可以通过下标（索引）来指定对应位置的元素。下标（索引）是从 0 开始，即 0 表示第一个元素，1 表示第二个元素；

初始化数组就是要使数组中每个元素具有确定的值，如果数组元素类型为基本数据类型，由于基本数据类型都有缺省值，数组会自动将元素初始化为缺省值，对于元素类型为引用数据类型的数组，初始化数组的每个元素则是必要的，否则元素为 null。

```
int[] arr = new int[5];
System.out.println(arr[0]); //0
System.out.println(arr[1]); //0
arr[0]=10; //通过指定下标对数组的元素进行赋值
System.out.println(arr[0]); //10
```

PS：每个数组都有一个属性 length，表示该数组可以存放元素的个数。Java 环境会自定检测数组的下标是否越界。

```
//通过for循环遍历数组中的所用元素
for (int i = 0; i < arr.length; i++) {
    System.out.println(arr[i]);
}
```

```
//通过增强for循环遍历
for (int i : arr) {
    System.out.println(i);
}
```

增强 for 循环是 JDK5.0 的新特性，内部使用的是迭代的方法 iteratoer 来实现的。只可以用来遍历集合不可以对集合中的元素进行赋值。

数组的静态初始化和动态初始化：

a) 动态初始化：使用 new 关键字初始化，需要指定数组的长度。

```
int[] arr = null;
arr = new int[5];
//通过for循环对数组的元素进行赋值
for (int i = 0; i < arr.length; i++) {
    arr[i]=i+10;
}
```

b) 静态初始化：在声明的同时进行初始化，需要指定数组的元素，根据元素由系统指定它的长度。

```
int[] arr={6,5,4,3,2};
for (int i = 0; i < arr.length; i++) {
    System.out.println(arr[i]);
}
```

PS：注意静态初始化和动态初始化的区别。

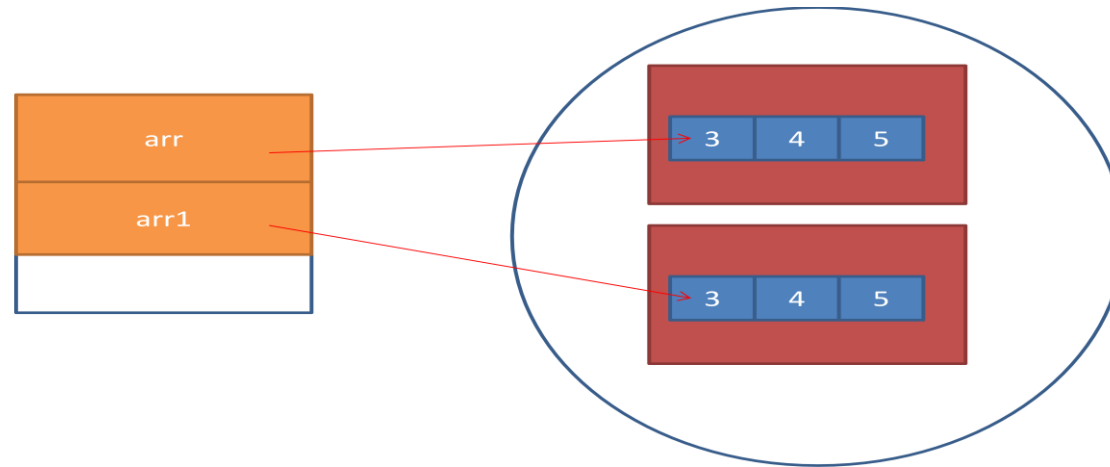
### 三、 一维数组

一维数组的内存划分：

对于基本数据类型：

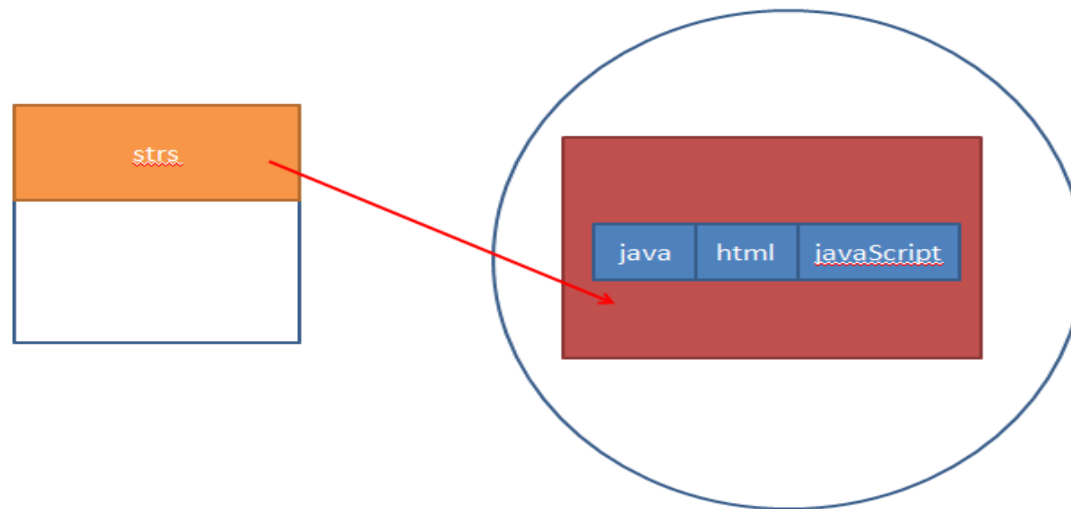
```
//静态初始化
int[] arr = {3,4,5};
//动态初始化
int[] arr1 = null;
arr1[0]=3;
arr1[1]=4;
```

`arr1[2]=5;`



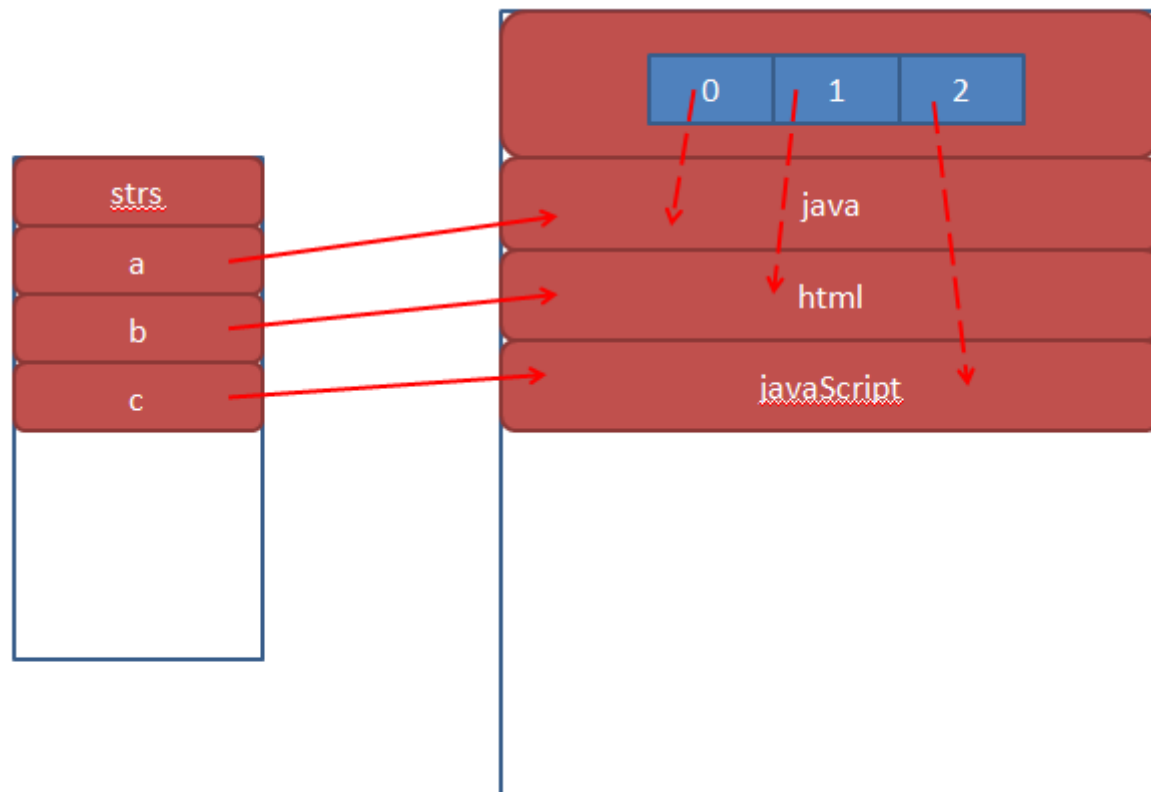
对于引用数据类型：以 String 伪列

`String[] strs = {"java", "html", "javaScript"};`



```
String[] strs = new String[3];  
String a = new String("java");  
String b = new String("html");  
String c = new String("javascript");
```

```
strs[0] = a;  
strs[1] = b;  
strs[2] = c;
```



测验：

a)

```
String[] strs = {"A","A"};

System.out.println(strs[0]==strs[1]);//true OR false
```

b)

```
String[] strs = {new String("A"),new String("A")};

System.out.println(strs[0]==strs[1]);//true OR false
```

## 四、 二维数组

静态初始化和遍历

```
int[][] arr = {{12,44,35,14,3},{76,54,31,34},{21,314},};
for (int i = 0; i < arr.length; i++) {
    for (int j = 0; j < arr[i].length; j++) {
        System.out.print(arr[i][j]+"\\t");
    }
    System.out.println();
}
```

## 动态初始化和遍历

```
int[][] arr = new int[3][];  
arr[0]=new int[2];  
arr[1]=new int[3];  
arr[2]=new int[4];  
for (int i = 0; i < arr.length; i++) {  
    for (int j = 0; j < arr[i].length; j++) {  
        System.out.print(arr[i][j]+"\\t");  
    }  
    System.out.println();  
}
```