

第二篇 ActionScript 数据操作篇

第 5 章 数 组

数组提供了一种理想的方法来存储、操作、排序和获取数据集。ActionScript 也不例外，它支持与数组操作有关的大量行为和函数，在 ActionScript 中提供了超过 20 个数组相关函数，其中很多通用的函数用于对数组元素排序、增加或删除数组元素、查询数组元素等。

本章将对 ActionScript 3.0 中的创建数组、数组赋值、Array 类的属性、对数组的操作以及关联数组和 multidimensional 数组进行详细地介绍。

5.1 Array 数组问题

5.1.1 问题描述

各位大虾，我在 ActionScript 中定义一个 String 类型的数组，那它的下标从哪里开始？请各位前辈指点，谢谢！

5.1.2 解决方法

哥们，下标是个索引，和数据类型没有任何关系。无论数组是 int、String、object 还是 Number、Boolean 等，索引永远都是从 0 开始。

例如，创建一个 String 类型的数组 myArray，代码如下所示。

```
var myArray:Array=new Array("a","b","c","d");
trace(myArray[0]);
trace(myArray[1]);
trace(myArray[2]);
trace(myArray[3]);
```

5.1.3 知识扩展——创建数组

通常情况下，用户可以通过 new 关键字和直接赋值的方式来创建数组对象。使用 new 关键字创建数组需要调用 Array 类的构造函数。Array 类构造函数的使用方式有三种，第一种，如果调用不带参数的构造函数，会得到空数组。可以使用 Array 类的 length 属性来验证数组是否不包含元素。

例如，下面的代码调用不带参数的 Array()构造函数。

```
var names:Array = new Array();  
trace(names.length); //输出: 0
```

第二种, 如果将一个数字用作 `Array()` 构造函数的唯一参数, 则会创建长度等于此数值的数组, 并且每个元素的值都设置为 `undefined`。参数必须为介于值 0 和 4 294 967 295 之间的无符号整数。

例如, 下面的代码调用带有一个数字参数的 `Array()` 构造函数。

```
var names:Array = new Array(3);  
trace(names.length); //输出: 3  
trace(names[0]); //输出: undefined  
trace(names[1]); //输出: undefined  
trace(names[2]); //输出: undefined
```

第三种, 如果调用构造函数并传递一个元素列表作为参数, 将创建具有与每个参数对应的元素的数组。下面的代码将三个参数传递给 `Array()` 构造函数。

```
var names:Array = new Array("李晨", "刘洋", "张珊");  
trace(names.length); //输出: 3  
trace(names[0]); //输出: 李晨  
trace(names[1]); //输出: 刘洋  
trace(names[2]); //输出: 张珊
```

除了使用 `new` 关键字创建数组之外, 还可以使用 `Array` 文本创建数组。可以将 `Array` 文本直接分配给数组变量, 如下面的示例所示。

```
var names:Array = ["李晨", "刘洋", "张珊"];
```

5.1.4 网络课堂



视频教学: <http://school.itzen.com/video-vid-3833-spid-61.html>

网络课堂: <http://bbs.itzen.com/thread-16559-1-1.html>

5.2 ActionScript 中的变量数组传递问题

5.2.1 问题描述

请教各位高手, 下面的语句为什么编译错误?

```
var myArray:Array=new Array(4);  
myArray={1,2,3,4};  
trace(myArray);
```

但是使用下面的语句却可以, 为什么呢?



```
var myArray:Array=new Array(1,2,3,4);
```

5.2.2 解决方法

你上面之所以编译错误，是因为{1, 2, 3, 4}只能用作数组的初始化，不能给已有数组赋值。使用下面几种方式都可以初始化并为数组赋值。

```
//使用 new 运算符初始化数组  
var myArray:Array=new Array(1,2,3,4);  
//不使用 new 运算符的初始化数组  
var myArray:Array=[1,2,3,4,5];
```

5.2.3 知识扩展——数组赋值

数组是稀疏数组，这意味着可能存在这样的情况：在索引 0 处有一个元素，在索引 5 处有另一个元素，而这两个元素之间的索引位置却是空的。在这种情况下，位置 1 至位置 4 的元素是未定义的，表示这些位置不存在元素，而不一定存在值为 `undefined` 的元素。

数组赋值是通过引用而不是通过值进行的。如果将一个数组变量赋值给另一个数组变量，则这两个变量引用同一个数组。

例如，下面代码创建两个数组类型的变量：`Array1` 和 `Array2` 分别对其赋值。

```
var Array1:Array = new Array("a", "b", "c");  
var Array2:Array = Array1;  
Array2[0] = "z";  
trace(Array1); //输出: z,b,c.
```

5.2.4 网络课堂



视频教学: <http://school.itzen.com/video-vid-3834-sp1d-61.html>

网络课堂: <http://bbs.itzen.com/thread-16560-1-1.html>

5.3 请教数组访问运算符的不解之处

5.3.1 问题描述

我想请教一下，在 `ActionScript` 中如果我声明了一个二维数组 `myArr`，那么当我打印数组中的元素时，该如何获取行和列的长度？请各位达人指点迷津，谢谢！

5.3.2 解决方法

哥们，你可以使用 `length` 属性来获取数组的维度。具体方法请看下面示例。

```
var myArr:Array=new Array();
myArr[0]=new Array(1,2);
myArr[1]=new Array(3,4);
myArr[2]=new Array(5,6);
trace(myArr[0].length);    //输出: 2
trace(myArr.length);      //输出: 3
```

5.3.3 知识扩展——Array 类的属性

`length` 属性，指定数组中元素数量的非负整数。在向数组中添加新元素时，此属性会自动更新。当给数组元素赋值（例如，`my_array[index]=value`）时，如果 `index` 是数字，而且 `index+1` 大于 `length` 属性，则 `length` 属性会更新为 `index+1`。



如果为 `length` 属性所赋的值小于现有长度，会将数组截断。

例如，下面的代码创建一个 `Array` 对象 `names`，其中包含字符串元素 `Jack`。然后它使用 `push()` 方法添加另外一个字符串元素 `Mary`。数组的长度由 `length` 属性确定，在使用 `push()` 之前是一个元素，在调用 `push()` 之后是两个元素。添加另外一个字符串 `Tom` 以使 `names` 的长度变为三个元素。然后，两次调用 `shift()` 方法以删除 `Jack` 和 `Mary`，从而使最终数组的 `length` 变为一。

```
var names:Array = new Array("Jack");
names.push("Mary");
trace(names.length);    //2
names.push("Tom");
trace(names.length);    //3
names.shift();
names.shift();
trace(names.length);    //1
```

5.3.4 网络课堂



视频教学：<http://school.itzen.com/video-vid-3835-sp1d-61.html>

网络课堂：<http://bbs.itzen.com/thread-16561-1-1.html>



5.4 如何添加数组元素

5.4.1 问题描述

请问 Array 类的 push()方法可以在数组前面添加元素吗？还是只能在数组末尾添加？要想在数组前面添加元素应该如何实现？请高手指教！

5.4.2 解决方法

哥们，push()方法不可以在数组前面添加元素！Array 类的 push()方法只能在数组末尾添加一个或者多个元素，要想实现在数组前面添加元素，可以使用 Array 类的 unshift()方法。具体示例如下所示。

```
var myArr:Array = new Array("a","b","c")
myArr.unshift("d");
trace(myArr); //d,a,b,c
```

5.4.3 知识扩展——插入数组元素

向数组添加元素，可以使用 Array 类的 push()、unshift()和 splice()三个方法将元素插入索引数组。

push()方法用于在数组末尾添加一个或多个元素。换言之，使用 push()方法在数组中插入的最后一个元素将具有最大索引号。

例如，下面的代码创建一个 Array 对象 letters，最初用以下元素填充：a。然后，使用一次 push()方法将元素 b 和 c 添加到数组的末尾。

```
var letters:Array = new Array("a");
var count:uint = letters.push("b", "c");
trace(letters); //a,b,c
trace(count); //3
```

unshift()方法用于在数组开头插入一个或多个元素，并且始终在索引号 0 处插入。

例如，下面的代码创建空的 Array 对象 letters。添加字符串 a 和 b（通过 push()方法），然后将字符串 c 和 d 添加到 letters 的开头（通过两次调用 unshift()方法）。

```
var names:Array = new Array();
names.push("a");
names.push("b");
trace(names); //a,b
names.unshift("c");
names.unshift("d");
trace(names); //d,c,a,b
```

`splice()`方法用于在数组中的指定索引处插入任意数目的项目。

例如，下面的代码创建 `Array` 对象 `letters`，其中包含元素`[a,b,c,d,e]`。然后，使用参数 2 和 2 调用 `splice()`方法，以将 `c` 和 `d` 赋予 `Arr` 数组。再次调用 `splice()`方法（使用参数 1、0 和 `Arr` 数组），以将`[c,d]`作为 `letters` 数组中的第二个元素。

```
var letters:Array = new Array("a","b","c","d","e");
var Arr:Array = letters.splice(2,2);
trace(letters);    //a,b,e
trace(Arr);       //c,d
letters.splice(1, 0, Arr);
trace(letters);   //a,c,d,b,e
```

`push()`和 `unshift()`方法均返回一个无符号整数，它们表示修改后的数组长度。在用于插入元素时，`splice()`方法返回空数组，通过使用 `splice()`方法，不仅可以将元素插入到数组中，而且还可以从数组中删除元素。用于删除元素时，`splice()`方法将返回包含被删除元素的数组。

5.4.4 网络课堂



视频教学：<http://school.itcn.com/video-vid-3836-sp1d-61.html>

网络课堂：<http://bbs.itcn.com/thread-16562-1-1.html>

5.5 ActionScript 中删除数组元素问题

5.5.1 问题描述

ActionScript 怎么删除数组中指定的元素？比如一个数组：`undefined, undefined, 18, 19, 20, 21`，我怎么删除所有 `undefined` 的元素，得到新数组 `18,19,20,21`？请高手赐教！

问题补充：

数组有可能是：`18, undefined, 19, 20, undefined, 21undefined`，元素位置和数组的长度都是没规律的。

5.5.2 解决方法

闲话不多说，具体实现方法如下所示。

```
var myPets_array:Array = new Array("cat", 'dog', "bird", "fish");
myPets_array[5] = '乌龟';
myPets_array[7] = '兔子';
myPets_array[10] = '刺猬';
```



```
var myPets:Array = new Array();
for (var i = 0; i<myPets_array.length; i++)
{
    if (myPets_array[i] != undefined)
    {
        myPets.push(myPets_array[i]);
    }
}
trace(myPets); //cat, dog, bird, fish, 乌龟, 兔子, 刺猬
```

5.5.3 知识扩展——删除数组元素

从数组中删除元素，可以使用 Array 类的三个方法 pop()、shift()和 splice()。

pop()方法用于从数组末尾删除一个元素。换言之，它将删除位于最大索引号处的元素。

例如，下面的代码创建一个 Array 对象 letters，其中包含元素 a、b 和 c。然后使用 pop()方法将最后一个元素 c 从数组中删除，并将该元素分配给 String 对象 lastletter。

```
var letters:Array = new Array("a","b","c");
trace(letters); //a,b,c
var lastletter:String = letters.pop();
trace(letters); //a,b
trace(lastletter); //c
```

shift()方法用于从数组开头删除一个元素，也就是说，它始终删除索引号 0 处的元素。

例如，下面的代码创建 Array 对象 letters，其中包含元素 a、b 和 c。然后，使用 shift()方法将第一个元素 a 从 letters 中删除，并将其分配给字符串 firstLetter。

```
var letters:Array = new Array("a", "b", "c");
var firstLetter:String = letters.shift();
trace(letters); //b,c
trace(firstLetter); //a
```

splice()方法既用来插入元素，也可以删除任意数目的元素，其操作的起始位置位于由发送到此方法的第一个参数指定的索引号处。

例如，下面的代码创建 Array 对象 letters，使用 splice()方法删除其中元素。

```
var letters:Array = ["a","b","c","d","e","f"];
letters.splice(2,2,"m","n"); //替换 c,d
var sp:Array= letters.splice(1,2);
trace(letters); //a,n,e,f
trace(sp); //b,m
```

pop()和 shift()方法均返回已删除的项。由于数组可以包含任意数据类型的值，因此返回值的数据类型为 Object。Splice()方法将返回包含被删除项的数组。



5.5.4 网络课堂



视频教学: <http://school.itzcn.com/video-vid-3837-sp1d-61.html>

网络课堂: <http://bbs.itzcn.com/thread-16563-1-1.html>

5.6 访问数组元素出错

5.6.1 问题描述

各位大哥大姐,我访问的数组以及访问数组元素的代码如下所示。

```
public function BB():void
{
    var a:Array=[1,2,3];
    for each(var i:int in a)
    {
        trace(a[i]);
    }
}
```

输出的结果如下所示。

```
2
3
undefined
```

请问各位大侠,我这样写的程序没有错吧?可这又是为什么呢?请高手指点,万分感谢!

5.6.2 解决方法

你没有想好思路就写程序了,这样很危险啊!以后再写程序的时候,一定要想好思路再编写程序。你写的代码是错误的,你误把元素值当成数组的索引了。如果非要用下标访问数组元素,可以使用 for 循环。请看下面的代码。

```
var a:Array=[1,2,3];
//直接访问数组元素值
for each(var i:int in a)
{
    trace(i);
}
```



```
//使用数组下标访问数组元素值
for(var j:int=0;j<a.length;j++)
{
    trace(a[j]);
}
```

5.6.3 知识扩展——访问数组元素

对于基本数据类型的 Array 元素，可以通过索引直接访问。在 ActionScript 3.0 中提供了对于存储复杂数据类型 Array 实例的访问。最经常使用的复杂数据类型就是 object 类的实例，常被定义为一个特殊的 Array 进行访问。代码如下所示。

```
var myObj:Object={name:"刘洋",sex:"男"};
```

使用这样的 Object 实例不需要通过 new 关键字进行初始化。访问实例中的对象也很容易，通过属性名称就可以访问到，代码如下所示。

```
var myObj:Object={name:"刘洋",sex:"男"};
myObj.name="王帅";
trace(myObj.name); //输出: 王帅
myObj["name"]="李晨";
trace(myObj["name"]); //输出: 李晨
```



访问这种简单对象可以使用两种方式：“.”操作符和“[]”操作符。“.”操作符进行操作时，不允许属性名中出现空格。

同样可以通过构建 Array 对象构建这样的实例，代码如下所示。

```
var myObj:Array=new Array();
myObj.name="李晨";
myObj["sex"]="男";
trace(myObj["name"]+", "+myObj["sex"]);
//输出: 李晨, 男
```

对于 Array 类本身存储内容可以是对象，由于 Array 类的实例也是一个对象，所以 Array 类可以存储自身的实例，形成二维以及多维数组，具体代码如下所示。

```
var myArr:Array=new Array();
myArr[0]=["李晨","刘洋","王帅"];
myArr[1]=["21","17","22"];
trace(myArr[0][0]); //输出: 李晨
trace(myArr[1][2]); //输出: 22
```

同样也可以使用 Object 实例的方式进行二维 Array 的构建和访问，代码如下所示。

```
var myArr:Array=new Array();
myArr["name"]=["李晨","刘洋","王帅"];
```



```
myArr["age"]=["21","17","22"];  
trace(myArr.name[1]); //输出: 刘洋  
trace(myArr.age[0]); //输出: 21
```

5.6.4 网络课堂



视频教学: <http://school.itzcn.com/video-vid-3838-sp1d-61.html>

网络课堂: <http://bbs.itzcn.com/thread-16564-1-1.html>

5.7 ActionScript 数组排序问题

5.7.1 问题描述

Pictures 是一个数组,保存的是图片文件的名称:

```
var pictures:Array=["scr.jpg","pic.jpg","PIC.JPG","oic.jpg"];
```

我希望得到的排序结果是:

```
oic.Jpg , pic.jpg , PIC.JPG , scr.jpg
```

我使用 `sort()` 函数无法得到这种排序结果:

```
var pictures:Array=["scr.jpg","pic.jpg","PIC.JPG","oic.jpg"];  
trace(pictures);  
pictures.sort();  
trace(pictures);
```

排序结果是:

```
PIC.JPG,oic.jpg,pic.jpg,scr.jpg
```

应该怎么排啊? 请各位大侠指教!

5.7.2 解决方法

呵呵,直接调用 `sort()` 方法肯定不行! `pictures` 数组元素的大小写不统一,而 `sort()` 方法的默认排序方式是区分大小写的 (Z 优先于 a),所以,想要实现排序还需要使用 `sort()` 方法的 `options` 参数。

多余的话不多说了,咱们直接看疗效吧。

```
var pictures:Array=["scr.jpg","pic.jpg","PIC.JPG","oic.jpg"];  
trace(pictures);
```



```
pictures.sort(Array.CASEINSENSITIVE); //使排序不区分大小写 (b 优先于 D)
trace(pictures);
```

输出结果如下所示。

```
oic.Jpg , pic.jpg , PIC.JPG , scr.jpg
```

5.7.3 知识扩展——排序数组

可以使用 Array 类的三个方法：`reverse()`、`sort()`和 `sortOn()`，通过排序或反向排序来更改索引数组的顺序。所有这些方法都用来修改现有数组。

1. `reverse()`方法

`reverse()`方法用于按照以下方式更改数组元素的顺序：最后一个元素变为第一个元素，倒数第二个元素变为第二个元素，以此类推。`reverse()`方法不带参数，也不返回值，但可以将数组从当前顺序切换为相反顺序。

例如，下面的代码创建一个 Array 对象 `letters`，其中包含元素 `a`、`b` 和 `c`。然后逆转数组元素的顺序（使用 `reverse()`方法）以生成数组`[c,b,a]`。

```
var letters:Array = new Array("a", "b", "c");
trace(letters); // a,b,c
letters.reverse();
trace(letters); // c,b,a
```

2. `sort()`方法

`sort()`方法可用来按照各种预定义的方式对数组的元素进行排序（如字母顺序或数字顺序），还可以创建自定义排序算法。

默认情况下，`sort()`方法按 ([code>方式进行排序：

- 排序区分大小写（Z 优先于 a）。
- 按升序排序（a 优先于 b）。
- 修改该数组以反映排序顺序；在排序后的数组中不按任何特定顺序连续放置具有相同排序字段的多个元素。
- 元素无论属于何种数据类型，都作为字符串进行排序，所以 100 在 99 之前，这是因为 "1" 的字符串值小于 "9" 的字符串值。

`sort()`方法具有 `options` 参数，可通过该参数改变默认排序顺序的各个特征。`options` 是由 Array 类中的一组静态常量定义的，如表 5-1 所示。

表 5-1 参数 `options` 的值与实现功能

参数 <code>options</code> 的值	实现功能
<code>Array.CASEINSENSITIVE</code>	可使排序不区分大小写（b 优先于 D）
<code>Array.DESENDING</code>	用于颠倒默认的升序排序（B 优先于 A）
<code>Array.UNIQUESORT</code>	如果发现两个相同的值，此选项将导致排序终止
<code>Array.NUMERIC</code>	使排序按照数字顺序进行（3 优先于 10）

例如，创建一个名为 `myArr` 的数组，并使用不同的参数对其进行排序，代码如下所示。

```
var myArr:Array = ["apple","Banana","Pear","grape"];
myArr.sort();           //默认排序
trace(myArr);          //输出: Banana,Pear,apple,grape
myArr.sort(Array.CASEINSENSITIVE);
trace(myArr);          //输出: apple,Banana,grape,Pear
myArr.sort(Array.DESENDING);
trace(myArr);          //输出: grape,apple,Pear,Banana
//使用两个选项
myArr.sort(Array.DESENDING | Array.CASEINSENSITIVE);
trace(myArr);          //输出: Pear,grape,Banana,apple
```

也可以编写自定义排序函数，然后将其作为参数传递给 `sort()` 方法。例如，创建一个名为 `myName` 的数组，使用自定义函数作为参数传递给 `sort()` 方法进行排序，代码如下所示。

```
var myName:Array = new Array("zhang san", "wang wu", "li si");
function orderName(a, b):int
{
    var lastName:RegExp = /\b\S+$/;
    var name1 = a.match(lastName);
    var name2 = b.match(lastName);
    if (name1 < name2)
    {
        return -1;    //表示第一个参数 a 优先于第二个参数 b
    }
    else if (name1 > name2)
    {
        return 1;    //表示第二个参数 b 优先于第一个参数 a
    }
    else
    {
        return 0;    //表示元素具有相同的排序优先级
    }
}
trace(myName);        // 输出: zhang san,wang wu,li si
myName.sort(orderName);
trace(myName);        //输出: zhang san,li si,wang wu
```

自定义排序函数 `orderName()` 使用正则表达式从每个元素中提取最后一个，以用于比较操作。`myName` 数组调用 `sort()` 方法时，函数标识符 `orderName` 用作唯一的参数。排序函数接受两个参数 `a` 和 `b`，因此它每次对两个数组元素进行操作。排序函数的返回值指示应如何对元素排序。

3. `sortOn()` 方法

`sortOn()` 方法用于对具有一个或多个公共属性的对象进行排序，排序时指定这样的属性作



为排序键。`sortOn()`方法是为包含对象的元素的 `Array` 对象设计的。这些对象应至少具有一个可用作排序键的公共属性。如果将 `sortOn()`方法用于任何其他类型的数组，则会产生意外结果。

例如，创建一个名为 `names` 的数组，以使每个元素均为对象而非字符串，代码如下所示。

```
var names:Array = new Array();
names.push({name:"Jack", born:"1989"});
names.push({name:"Mary", born:"1992"});
names.push({name:"Tom", born:"1988"});
names.push({name:"John", born:"1990"});
names.push({name:"Smith", born:"1985"});
```

可以使用 `sortOn()`方法，按照 `born` 属性对 `Array` 进行排序。`sortOn()`方法定义两个参数 `fieldName` 和 `options`。必须将 `fieldName` 参数指定为字符串。

在以下示例中，使用两个参数 `born` 和 `Array.NUMERIC` 来调用 `sortOn()`。`Array.NUMERIC` 参数用于确保按照数字顺序进行排序，而不是按照字母顺序。即使所有数字具有相同的数位，这也是一种很好的做法，因为当后来在数组中添加较少数位或较多数位的数字时，它会确保排序如期继续进行。

```
names.sortOn("born", Array.NUMERIC);
for (var i:int = 0; i<names.length; ++i)
{
    trace(names[i].name,names[i].born);
}
```

输出结果如下所示。

```
Smith 1985
Tom 1988
Jack 1989
John 1990
Mary 1992
```

通常，`sort()`和 `sortOn()`方法用于修改 `Array`。如果要对 `Array` 排序而又不修改现有数组，要将 `Array.RETURNINDEXEDARRAY` 常量作为 `options` 参数的一部分进行传递。此选项将指示方法返回反映排序的新数组，同时保留原始数组。方法返回的数组为由反映新排序顺序的索引号组成的简单数组，不包含原始数组的任何元素。

例如，若要根据出生年份对 `names` 数组排序而不修改该数组，可在传递的 `options` 参数中包括 `Array.RETURNINDEXEDARRAY` 常量。下面的示例将返回的索引信息存储在名为 `Arr` 的数组中，然后使用 `Arr` 数组和未修改的 `names` 数组按出生年份的顺序输出姓名。

```
var Arr:Array;
Arr= names.sortOn("born", Array.NUMERIC | Array.RETURNINDEXEDARRAY);
for (var i:int = 0; i < Arr.length; ++i)
{
    var index:int = Arr [i];
```



```
        trace(names[index].name, names[index].born);  
    }  
}
```

输出结果如下所示。

```
Smith 1985  
Tom 1988  
Jack 1989  
John 1990  
Mary 1992
```

5.7.4 网络课堂



视频教学: <http://school.itzcn.com/video-vid-3839-sp1d-61.html>

网络课堂: <http://bbs.itzcn.com/thread-16565-1-1.html>

5.8 如何查询数组

5.8.1 问题描述

求教, ActionScript 中如何查询数组元素? 小弟初学 ActionScript, 希望各位大哥大姐帮忙讲解一下, 最好给个实例! 万分感谢!!

5.8.2 解决方法

在 Actionscript 中, 查询数组可以使用 Array 类的四个方法, 分别是: `concat()`、`join()`、`slice()`和 `toString()`。你可以根据需要选用不同的方法, 下面看一下 `slice()`方法的小实例, 代码如下所示。

```
var myArr:Array=new Array("a","b","c","d","e","f");  
var Arr:Array=myArr.slice(2,6);  
trace(Arr); //输出: c,d,e,f
```

5.8.3 知识扩展——查询数组

Array 类的四个方法 `concat()`、`join()`、`slice()`和 `toString()`均可用于查询数组的信息, 而不修改数组。`concat()`和 `slice()`方法返回新数组, 而 `join()`和 `toString()`方法返回字符串。

`concat()`方法将新数组和元素列表作为参数, 并将其与现有数组结合起来创建新数组。`slice()`方法有两个名为 `startIndex` 和 `endIndex` 的参数, 并返回一个新数组, 新数组中包含从现



有数组“分离”的元素副本。分离从 `startIndex` 处的元素开始，到 `endIndex` 处的前一个元素结束。值得强调的是，`endIndex` 处的元素不包括在返回值中。

以下示例通过 `concat()` 和 `slice()` 方法，使用其他数组的元素创建一个新数组。

```
var arr1:Array = ["apple","banana"];
var arr2:Array = arr1.concat("grape","pear");
trace(arr2); //输出: apple,banana,grape,pear
var arr3:Array = arr1.concat(arr2);
trace(arr3); //输出: apple,banana,apple,banana,grape,pear
var arr4:Array = arr3.slice(2,5);
trace(arr4); //输出: apple,banana,grape
```

可以使用 `join()` 和 `toString()` 方法查询数组，并将其内容作为字符串返回。如果 `join()` 方法没有使用参数，则这两个方法的行为相同，它们都返回一个字符串，其中包含数组中所有元素的逗号分隔列表。与 `toString()` 方法不同，`join()` 方法接受名为 `delimiter` 的参数，可以使用此参数，选择要用作返回字符串中各个元素之间分隔符的符号。

下面的示例创建名为 `ArrName` 的数组，并调用 `join()` 和 `toString()` 以便采用字符串形式返回该数组中的值。`toString()` 方法用于返回以逗号分隔的值，而 `join()` 方法用于返回以+字符分隔的值。

```
var ArrName:Array = ["Jack", "Mary", "Tom", "Smith"];
var name1:String = ArrName.toString();
trace(name1); //输出: Jack,Mary,Tom,Smith
var name2:String = ArrName.join("+");
trace(name2); //输出: Jack+Mary+Tom+Smith
```

5.8.4 网络课堂



视频教学: <http://school.itcn.com/video-vid-3840-sp1d-61.html>

网络课堂: <http://bbs.itcn.com/thread-16566-1-1.html>

5.9 关联数组排序问题

5.9.1 问题描述

ActionScript 中，想给一个关联数组排序，有什么方法吗？类似于这样的数组：

```
array([电脑] =6 [硬盘] =10 [内存] = 4)
```

现在我想根据元素值的升序来排列这个数组，也就是成这个样子：

```
array ([内存] =4 [电脑] = 6 [硬盘] = 10)
```

有现成的函数吗？没有的话应该怎么办呢？或者是什么思路，谢谢！

5.9.2 解决方法

在 ActionScript 中，对数组按值排序可以使用 `sort()` 方法的 `Array.NUMERIC` 参数，具体实现方法如下所示。

```
var Arr:Array=new Array();
Arr["电脑"]=[6];
Arr["硬盘"]=[10];
Arr["内存"]=[4];
Arr.sort(Array.NUMERIC);
for (var key:Object in Arr)
{
    trace(key,Arr[key]);
}
```

输出结果如下所示。

```
内存 4
电脑 6
硬盘 10
```

5.9.3 知识扩展——关联数组

关联数组有时也称为“哈希”或“映射”，它使用“键”而非数字索引来组织存储的值。关联数组中的每个键都是用于访问一个存储值的唯一字符串。

关联数组为 `Object` 类的实例，也就是说每个键都与一个属性名称对应。关联数组是键和值对的无序集合，在代码中，不应期望关联数组的键按特定的顺序排列。

5.9.3.1 具有字符串键的关联数组

在 ActionScript 3.0 中有两种创建关联数组的方式。第一种方式是使用 `Object` 实例，它的优点是可使用对象文本初始化数组。`Object` 类的实例（也称为“通用对象”）在功能上等同于关联数组。通用对象的每个属性名称都用作键，提供对存储的值的访问。

下面的示例创建一个名为 `myObj` 的关联数组，并使用对象文本初始化具有两个键和值对的数组，代码如下所示。

```
var myObj:Object = {color:"Red", area:"1200x960"};
trace(myObj["color"], myObj["area"]);
//输出: Red 1200x960
```

如果在声明数组时不需要初始化，可以使用 `Object` 构造函数创建数组，代码如下所示。

```
var myObj:Object = new Object();
```



使用对象文本或 `Object` 类构造函数创建数组后，可以使用数组访问 (`[]`) 运算符或点运算符 (`.`) 向数组添加新值。以下示例将两个新值添加到 `myObj` 中，代码如下所示。

```
myObj["new color"] = "Deep Blue";
myObj.area = "1024x480";
trace(myObj["new color "], myObj.area);
//输出: Blue 1024x480
```



名为 `new color` 的键包含空格字符。可以使用数组访问 (`[]`) 运算符，但是如果尝试使用点运算符，则会生成错误。不建议在键名称中使用空格。

创建关联数组的第二种方式是使用 `Array` 构造函数（或任何动态类的构造函数），然后使用数组访问 (`[]`) 运算符或点运算符 (`.`) 将键和值对添加到数组中。如果将关联数组声明为数组类型，则将无法使用对象文本初始化该数组。

以下示例使用 `Array` 构造函数创建一个名为 `myArr` 的关联数组，并添加一个名为 `color` 的键和一个名为 `area` 的键以及它们的值，代码如下所示。

```
var myArr:Array = new Array();
myArr["color"] = "Black";
myArr["area"] = "1400 x 1100";
trace(myArr ["color "], myArr ["area"]);
//输出: Black 1400 x 1100
```

使用 `Array` 构造函数创建关联数组没有什么优势。即使使用 `Array` 构造函数或 `Array` 数据类型，也不能将 `Array` 类的 `Array.length` 属性或任何方法用于关联数组。最好将 `Array` 构造函数用于创建索引数组。

5.9.3.2 具有对象键的关联数组

可以使用 `Dictionary` 类创建使用对象而非字符串作为键的关联数组。这样的数组有时候也称作字典、哈希或映射。

以下代码创建三个用作 `Dictionary` 对象的键的 `Sprite` 对象实例。它为每个键分配了值 `A` 或 `B`。值可以是任意数据类型，但在此示例中，`A` 和 `B` 均为 `Object` 类的实例。然后，可以使用数组访问 (`[]`) 运算符访问与每个键关联的值，代码如下所示。

```
import flash.display.Sprite;
import flash.utils.Dictionary;
var myDic:Dictionary = new Dictionary();
//要用作键的对象
var s1:Sprite = new Sprite();
var s2:Sprite = new Sprite();
var s3:Sprite = new Sprite();
//要用作值的对象
var A:Object = new Object();
var B:Object = new Object();
```



```
//在字典中创建新的键-值对
myDic [s1] = A;
myDic [s2] = B;
myDic [s3] = B;
if (myDic [s1] == A)
{
    trace("s1 is in A");
}
if (myDic[s2] == B)
{
    trace("s2 is in B");
}
if (myDic[s3] == B)
{
    trace("s3 is in B");
}
```

5.9.3.3 使用对象键循环访问

可以使用 `for...in` 循环或 `for each...in` 循环来遍历 `Dictionary` 对象的内容。`for...in` 循环用于基于键进行遍历，而 `for each...in` 循环用于基于与每个键关联的值进行遍历。

使用 `for...in` 循环可以直接访问 `Dictionary` 对象的对象键。还可以使用数组访问(`[]`)运算符访问 `Dictionary` 对象的值。下面的代码使用前面的 `myDic` 字典示例来说明如何使用 `for...in` 循环来遍历 `Dictionary` 对象。

```
for (var key:Object in myDic)
{
    trace(key, myDic[key]);
}
```

输出结果如下所示。

```
[object Sprite] [object Object]
[object Sprite] [object Object]
[object Sprite] [object Object]
```

使用 `for each...in` 循环可以直接访问 `Dictionary` 对象的值。下面的代码也使用 `myDic` 字典来说明如何使用 `for each...in` 循环来遍历 `Dictionary` 对象。

```
for each (var Obj:Object in myDic)
{
    trace(Obj);
}
```

输出结果如下所示。

```
[object Object]
```

[object Object]
[object Object]

5.9.4 网络课堂



视频教学: <http://school.itzcn.com/video-vid-3841-sp1d-61.html>

网络课堂: <http://bbs.itzcn.com/thread-16567-1-1.html>

5.10 如何处理数组越界的异常

5.10.1 问题描述

ActionScript 有没二维数组, 像 VB 里的那种 `abc(j,k)`? 如何定义? 如何引用? 请高手指教!

5.10.2 解决方法

当然有了, 可以直接定义, 代码如下所示。

```
var thearray = [[1,2],[3,4],[3,4]];
```

也可以这样:

```
var thearray=new Array();  
thearray[0]=new Array(1,2);  
thearray[1]=new Array(3,4);  
thearray[2]=new Array(5,6);  
trace(thearray[0][1]); //输出: 2
```

5.10.3 知识扩展——多维数组

多维数组将其他数组作为其元素。例如, 考虑一个任务列表, 它存储有索引的字符串数组。

```
var Arr:Array = ["上班","约会"];
```

如果要将一周中每天的任务存储为一个单独的列表, 可以创建一个多维数组, 一周中的每天使用一个元素。每个元素包含一个与 `Arr` 数组类似的索引数组, 而该索引数组存储任务列表。

在多维数组中, 可以使用任意组合的索引数组和关联数组。以下部分中的示例使用了两个索引数组或由索引数组组成的关联数组。



5.10.3.1 两个索引数组

使用两个索引数组时，可以将结果呈现为表或电子表格。第一个数组的元素表示表的行，第二个数组的元素表示表的列。

例如，以下多维数组使用两个索引数组跟踪一周中每一天的任务列表。第一个数组 `myArr` 是使用 `Array` 类构造函数创建的。此数组中的各个元素分别表示一周中的各天，其中索引 0 表示星期一，索引 6 表示星期日。可将这些元素当成是表的行。可通过为 `myArr` 数组中创建的七个元素中的每个元素分配数组文本来创建每一天的任务列表。这些数组文本表示表的列。

```
var myArr:Array = new Array();
myArr[0] = ["上班","逛街"];
myArr[1] = ["上班", "睡觉","逛街"];
myArr[2] = ["上班", "体育锻炼"];
myArr[3] = ["上班"];
myArr[4] = ["上班","打扫卫生"];
myArr[5] = ["睡觉","逛街", "看电影"];
myArr[6] = ["约会", "游戏"];
```

可以使用数组访问 (`[]`) 运算符访问任意任务列表中的各个项。第一组括号表示一周的某一天，第二组括号表示这一天的任务列表。

例如，若要检索星期三的列表中的第二项任务，请首先使用表示星期三的索引 2，然后使用表示列表中的第二项任务的索引 1。

```
trace(myArr [2][1]); //输出：体育锻炼
```

若要检索星期日的列表中的第一项，请使用表示星期日的索引 6 和表示列表中的第一项任务的索引 0。

```
trace(myArr[6][0]); //输出：约会
```

5.10.3.2 具有索引数组的关联数组

要使单个数组的访问更加方便，可以使用关联数组表示一周的各天并使用索引数组表示任务列表。通过使用关联数组可以在引用一周中特定的一天时使用点语法，但要访问关联数组的每个元素还需额外进行运行时处理。

以下示例使用关联数组作为任务列表的基础，并使用键和值对来表示一周中的每一天。

```
var myObj:Object = new Object();
myObj["Monday"] = ["上班","逛街"];
myObj["Tuesday"] = ["上班", "睡觉","逛街"];
myObj["Wednesday"] = ["上班", "体育锻炼"];
myObj["Thursday"] = ["上班"];
myObj["Friday"] = ["上班","打扫卫生"];
myObj["Saturday"] = ["睡觉","逛街", "看电影"];
```



```
myObj["Sunday"] = ["约会", "游戏"];
```

点语法通过避免使用多组括号改善了代码的可读性。

```
trace(myObj.Wednesday[1]); //输出: 体育锻炼  
trace(myObj.Sunday[0]); //输出: 约会
```

可以使用 `for...in` 循环来遍历任务，但访问与每个键关联的值时必须使用数组访问 (`[]`) 运算符，而不是点语法。由于 `myObj` 为关联数组，因此不一定会按照所期望的顺序检索元素，如以下示例所示。

```
for (var day:String in myObj)  
{  
    trace(day + ": " + myObj[day])  
}
```

输出结果如下所示。

```
Monday: 上班, 逛街  
Sunday: 约会, 游戏  
Friday: 上班, 打扫卫生  
Thursday: 上班  
Wednesday: 上班, 体育锻炼  
Tuesday: 上班, 睡觉, 逛街  
Saturday: 睡觉, 逛街, 看电影
```

5.10.4 网络课堂



视频教学: <http://school.itcn.com/video-vid-3842-sp1d-61.html>

网络课堂: <http://bbs.itcn.com/thread-16568-1-1.html>