队列方法

栈数据结构的访问规则是 LIFO（后进先出），而队列数据结构的访问规则是 FIFO（First-In-First-Out，先进先出）。队列在列表的末端添加项，从列表的前端移除项。由于 push()是向数组末端添加项的方法，因此要模拟队列只需一个从数组前端取得项的方法。实现这一操作的数组方法就是 shift()，它能够移除数组中的第一个项并返回该项，同时将数组长度减 1。结合使用 shift()和 push()方法，可以像使用队列一样使用数组。

```
var colors = new Array();                    //创建一个数组
var count = colors.push("red", "green");     //推入两项
alert(count);    //2

count = colors.push("black");                //推入另一项
alert(count);     //3

var item = colors.shift();                   //取得第一项
alert(item);      //"red"
alert(colors.length); //2
```

ArrayTypeExample11.htm

这个例子首先使用 push()方法创建了一个包含 3 种颜色名称的数组。代码中加粗的那一行使用 shift()方法从数组中取得了第一项，即"red"。在移除第一项之后，"green"就变成了第一项，而"black"则变成了第二项，数组也只包含两项了。

ECMAScript 还为数组提供了一个 unshift()方法。顾名思义，unshift()与 shift()的用途相反：它能在数组前端添加任意个项并返回新数组的长度。因此，同时使用 unshift()和 pop()方法，可以从相反的方向来模拟队列，即在数组的前端添加项，从数组末端移除项，如下面的例子所示。

```
var colors = new Array();                    //创建一个数组
var count = colors.unshift("red", "green");  //推入两项
alert(count);    //2

count = colors.unshift("black");             //推入另一项
alert(count);    //3

var item = colors.pop();                     //取得最后一项
alert(item);     //"green"
alert(colors.length); //2
```

ArrayTypeExample12.htm

这个例子创建了一个数组并使用 unshift()方法先后推入了 3 个值。首先是"red"和"green"，然后是"black"，数组中各项的顺序为"black"、"red"、"green"。在调用 pop()方法时，移除并返回的是最后一项，即"green"。

IE7 及更早版本对 JavaScript 的实现中存在一个偏差，其 unshift()方法总是返回 undefined 而不是数组的新长度。IE8 在非兼容模式下会返回正确的长度值。

小节：添加（push,unshift）返回的都是数组长度，删除（pop,shift）返回的都是删除的值。