JavaScript 数组使用技巧

```
/*
 * 用于测试 JavaScript 数组的用法
 *
 * Steven
 */

//创建数组，长度由后期赋值决定
var arry1 = new Array;
for (i = 0; i < 5; i++) {
    arry1 = 2 * i + 1;
}

// length 属性，返回数组的长度
alert("arry1.length=" + arry1.length);

// toString()方法，返回数组的字符串表示
alert("arry1=" + arry1.toString());

// 构建数组并初始化，有点像 Java
arry2 = [2, 4, 6, 8];

// concat 方法，用于连接两个数组
var arry3 = arry1.concat(arry2);
alert(arry3); // 1, 3, 5, 7, 9, 2, 4, 6, 8

// concat 的另外一种用法，可以有多个参数
var arry4 = arry2.concat(1, 3, 5, 7, 9);
alert(arry4); // 2, 4, 6, 8,  1, 3,  5,  7,  9

// join 方法，用于将数组中的各个元素连接成字符串
var arry5 = ["one", "two", "three"];

// 默认用","连接
var strArry = arry5.join();
alert(strArry); // "one,two,three"
```

```
// 也可以指定连接的字符
  strArry = arry5.join("|");
  alert(strArry); // "one|two|three"


  // sort 方法用于将数组排序
  var arry6 = [2, 3, 1, 6, 5, 3, 1, 4, 7];
  arry6.sort();
  alert(arry6.toString()); // 1, 1, 2, 3, 3, 4, 5, 6, 7

  // 对于字符串数组, sort 方法使用字典的顺序排序
  var arry7 = ['BB', 'AAA', 'C'];
  arry7.sort();
  alert(arry7); // AAA, BB, C

  // 也可以通过回调函数的方式自定义排序的大小逻辑
  // 例如下面的例子将按照字符串的长度对 arry7 排序

  /*
   * sort 方法运行时将调用使用数组的元素调用 function(a1, a2){...}
   * 根据其返回的结果判断元素的大小, 其逻辑为:
   * 返回值>0 表示 a1>a2
   * 返回值<0 表示 a1<a2
   * 返回值=0 表示 a1=a2
   *
   * 这有点像 Java 语言中的 Comparable
   */
  arry7.sort(function(a1, a2) {
      return a1.length - a2.length;
  });
  alert(arry7); // C, BB, AAA

  // 调用 reverse 方法, 可以将数组反转
  arry8 = [1,2,3,4,5];
  arry8.reverse();
  alert(arry8.toString()); // 5, 4, 3, 2, 1
```

```
// slice 方法，用于截取数组的一部分并以数组的形式返回
    // 两个参数分别为截取部分的上界和下界(前包括，后不包括，和 Java 一样)
    var arry9 = ['一','二','三','四','五'];
    var arry10 = arry9.slice(1,3);
    alert(arry10); // 二，三

    // splice 方法，用于删除原数组的一部分内容，并用指定的元素替换

    /*
     * splice 方法的前 2 个参数表示删除部分的上界和下界(前后都包括)
     * 从 splice 方法的第 3 个的参数开始的多个参数为替换成的内容
     */
    var arry11 = ['A', 'B', 'C', 'D', 'E', 'F', 'G'];
    arry11.splice(1, 5, '&','%','#');
    alert(arry11); // A, &, %, #, G


    // pop 和 push 方法，可以利用数组实现栈(先进后出)的操作

    /*
     * push 方法，将元素追加到数组尾端(进栈)，可以支持多参数，每次调用的返回值为此时数组的长度
     * pop 方法，从数组的尾端取出元素(出栈)，返回值为出栈的元素
     */

    var arry12 = new Array;
    // 进栈一个元素
    arry12.push('one');

    // 进栈多个元素
    var size = arry12.push('two','three','four');
    alert(size); // 4
    size = arry12.push('five','six','seven','eight','night','ten');
    alert(size); // 10

    for(i=0; i<size; i++) {
     alert(arry12.pop());
    }
```

```
/*
 * ********************
 *  以下方法是 JS 框架 Prototype.js  提供的用于对数组进行操作的方法
 *  使用时要引用 Prototype.js
 * ********************
 */
//Prototype 的 clone 方法，用于实现数组的复制(克隆)，返回值为复制后的新数组
var str = "how long no see you";
// string 的 split 方法有点像 Java 语言
var arry13 = str.split(' ');
alert(arry13); // how, long, no, see, you
var arry14 = arry13.clone();
alert(arry14); // how, long, no, see, you

// Prototype 的 compact 方法，用于删除原数组中值为 null 和 undefined(未定义)的元素
//  返回值为删除后的新数组，原数组不变
var arry15 = ["spring",,"summer",,null,"autumn",,,"winter",null];
alert(arry15.length); //10:
var arry16 = arry15.compact();
alert(arry16.length+": "+arry16.toString()); //4: spring, summer, autumn, winter

// Prototype 的 without 方法，用于删除数组中的指定元素
// without 方法支持多参数，返回值为删除后的新数组，原数组不变
var arry17 = ['java', 'html', 'css', 'c++','js','php','c#','js'];
//  删除数组 arry18 中值为'html', 'css'  和'js'的元素
var arry18 = arry17.without('html','css','js');
alert(arry18.toString()); // java, c++, php, c#
```