

转换方法

如前所述，所有对象都具有 `toLocaleString()`、`toString()` 和 `valueOf()` 方法。其中，调用数组的 `toString()` 方法会返回由数组中每个值的字符串形式拼接而成的一个以逗号分隔的字符串。而调用 `valueOf()` 返回的还是数组。实际上，为了创建这个字符串会调用数组每一项的 `toString()` 方法。来看下面这个例子。

```
var colors = ["red", "blue", "green"]; // 创建一个包含 3 个字符串的数组
alert(colors.toString()); // red,blue,green
alert(colors.valueOf()); // red,blue,green
alert(colors); // red,blue,green
```

另外，`toLocaleString()` 方法经常也会返回与 `toString()` 和 `valueOf()` 方法相同的值，但也不总是如此。当调用数组的 `toLocaleString()` 方法时，它也会创建一个数组值的以逗号分隔的字符串。而与前两个方法唯一的不同之处在于，这一次为了取得每一项的值，调用的是每一项的 `toLocaleString()` 方法，而不是 `toString()` 方法。请看下面这个例子。

```
var person1 = {
    toLocaleString : function () {
        return "Nikolaos";
    },

    toString : function() {
        return "Nicholas";
    }
};

var person2 = {
    toLocaleString : function () {
        return "Grigorios";
    },

    toString : function() {
        return "Greg";
    }
};

var people = [person1, person2];
alert(people); //Nicholas,Greg
alert(people.toString()); //Nicholas,Greg
alert(people.toLocaleString()); //Nikolaos,Grigorios
```

数组继承的 `toLocaleString()`、`toString()` 和 `valueOf()` 方法，在默认情况下都会以逗号分隔的字

符串的形式返回数组项。而如果使用 `join()` 方法，则可以使用不同的分隔符来构建这个字符串。`join()` 方法只接收一个参数，即用作分隔符的字符串，然后返回包含所有数组项的字符串。（不传默认使用逗号分隔）



如果数组中的某一项的值是 `null` 或者 `undefined`，那么该值在 `join()`、`toLocaleString()`、`toString()` 和 `valueOf()` 方法返回的结果中以空字符串表示。