

## 17.9.5 转换为字符串

把值转换为字符串是编程中常见行为。手动转换的具体方法如下。

### 1. 使用加号运算符

当值与空字符串相加运算时，JavaScript 会自动把值转换为字符串。例如：

- 把数字转换为字符串：

```
var a = 123456;
a = a + "";
alert(typeof a);           // 返回类型为 string
```

- 把布尔值转换为字符串，返回字符串"true"或"false"：

```
var a = true;
a = a + "";
alert(a);                  // 返回字符串"true"
```

- 把数组转换为字符串，返回数组元素列表，以逗号分隔：

```
var a = [1,2,3];
a = a + "";
alert(a);                  // 返回字符串"1,2,3"
```

- 把函数转换为字符串，返回函数结构的代码字符串：

```
var a = function(){
    return 1;
};
a = a + "";
alert(a);                  // 返回字符串"var a = function(){ return 1;}"
```

如果把 JavaScript 内置对象转换为字符串，则只返回构造函数的基本结构代码，而自定义的构造函数，则与普通函数一样，返回函数结构的代码字符串。

```
a = Date + "";
alert(a);                  // 返回字符串" function Date () { [ native code ]}"
```

如果内置对象为静态函数，则返回字符串不同。例如：

```
a = Math + "";
alert(a);                  // 返回字符串"[object Math]"
```

如果把对象实例转换为字符串，则返回的字符串会根据不同类型或定义对象的方法和参数而不同，具体说明如下。

- 对象直接量，则返回字符串为"[object object]"。

```
var a = {
    x : 1
}
a = a + "";
alert(a);                  // 返回字符串"[object object]"
```

- 如果是自定义类的对象实例，则返回字符串为"[object object]"。

```
var a = new function(){}();  
a = a + "";  
alert(a);           // 返回字符串"[object object]"
```

- 如果是内置对象实例，具体返回字符串必须根据传递的参数而定。

**【示例】**正则表达式对象会返回匹配模式字符串，时间对象会返回当前 GMT 格式的时间字符串，数值对象会返回传递的参数值字符串或者 0 等。

```
a = new RegExp(/^\\w$/) + "";  
alert(a);           // 返回字符串"/^\\w$/"
```

### 【拓展】

加号运算符有两个计算功能：数值求和、字符串连接。但是字符串连接操作的优先级要大于求和运算。因此，在可能的情况下，即运算元的数据类型不一致时，加号运算符会尝试把数值运算元转换为字符串，再执行连接操作。在下面小节中我们还会介绍这样的应用。

但是当多个加号运算符位于同一行时，这个问题就比较复杂，例如：

```
var a = 1 + 1 + "a";  
var b = "a" + 1 + 1;  
alert(a);           // 返回字符串"2a"  
alert(b);           // 返回字符串"a11"
```

通过上面示例可以看到，加号运算符不仅仅优先于连接操作，同时还会考虑运算的顺序。对于变量 **a** 来说，按着从左到右的运算顺序，加号运算符会执行求和运算，然后再执行连接操作。但是对于变量 **b** 来说，由于 **"a" + 1** 表达式运算将根据连接操作来执行，所以返回字符串 **"a1"**，然后再用这个字符串与数值 1 进行运算，再次执行连接操作，最后返回字符串 **"a11"**，而不是字符串 **"a2"**。

如果要避免此类现象的发生，可以考虑使用小括号运算符来改变一行内表达式的运算顺序。例如：

```
var b = "a" + (1 + 1);    // 返回字符串"a2"
```

## 2. 使用 toString() 方法

当为原始值调用 **toString()** 方法时，JavaScript 会自动把它们装箱为对象。然后再调用 **toString()** 方法，把它们转换为字符串。例如：

```
var a = 123456;  
a.toString();  
alert(a);           // 返回字符串"123456"
```

使用加号运算符转换字符串，实际上也是调用 **toString()** 方法来完成。只不过是 JavaScript 自动调用 **toString()** 方法实现的。