

17.9.7 转换为布尔值

在 JavaScript 中，任何数据都可以被自动转换为布尔值，这种转换往往都是自动完成的。例如，把值放入条件或循环结构的条件表达式中，或者参与到逻辑运算时，JavaScript 解释器都会自动把它们转换为布尔值。用法可以手动进行转换，具体方法如下。

1. 使用双重逻辑非

任何一个值如果在前面加上一个逻辑非运算符，JavaScript 都会把这个表达式看做是逻辑运算。执行运算时，先把值转换为布尔值，然后再执行逻辑非运算。例如：

```
var a =!0;           // 返回 true
var b =!1;           // 返回 false
var c =!NaN;         // 返回 true
var d =!null;        // 返回 true
var e =!undefined;   // 返回 true
var f =![];          // 返回 false
var g =!{};          // 返回 false
```

如果再给这个表达式添加一个逻辑非运算符，所得的布尔值就是该值被转换为布尔型数据的真实值了。例如：

```
var a =!!0;          // 返回 false
var b =!!1;          // 返回 true
var c =!!NaN;        // 返回 false
var d =!!null;       // 返回 false
var e =!!undefined;  // 返回 false
var f =!![];         // 返回 true
var g =!!{};         // 返回 true
```

2. 使用 Boolean()构造函数转换

使用 Boolean()构造函数转换的方法如下：

```
var a =0;
var b =1;
a = new Boolean(a);    // 返回 false
b = new Boolean(b);    // 返回 true
```

不过这种方法会把布尔值包装为引用型对象，而不再是原始值了。使用 typeof 运算符检测如下：

```
var a =0;
var b = !!a;
var c = new Boolean(a);
alert(typeof b);       // 返回 boolean
alert(typeof c);       // 返回 object
```