

Boolean 对象

概述

`Boolean`对象是 JavaScript 的三个包装对象之一。作为构造函数，它主要用于生成布尔值的包装对象实例。

```
```javascript
var b = new Boolean(true);

typeof b // "object"
b.valueOf() // true
```
```

上面代码的变量`b`是一个`Boolean`对象的实例，它的类型是对象，值为布尔值`true`。

注意，`false`对应的包装对象实例，布尔运算结果也是`true`。

```
```javascript
if (new Boolean(false)) {
 console.log('true');
} // true

if (new Boolean(false).valueOf()) {
 console.log('true');
} // 无输出
```
```

上面代码的第一个例子之所以得到`true`，是因为`false`对应的包装对象实例是一个对象，进行逻辑运算时，被自动转化成布尔值`true`（因为所有对象对应的布尔值都是`true`）。而实例的`valueOf`方法，则返回实例对应的原始值，本例为`false`。

Boolean 函数的类型转换作用

`Boolean`对象除了可以作为构造函数，还可以单独使用，将任意值转为布尔值。这时`Boolean`就是一个单纯的工具方法。

```
```javascript
Boolean(undefined) // false
Boolean(null) // false
Boolean(0) // false
Boolean('') // false
Boolean(NaN) // false

Boolean(1) // true
Boolean('false') // true
Boolean([]) // true
Boolean({}) // true
Boolean(function () {}) // true
```
```

```
Boolean(/foo/) // true
```
```

上面代码中几种得到`true`的情况，都值得认真记住。

顺便提一下，使用双重的否运算符（!）也可以将任意值转为对应的布尔值。

```
```javascript
!!undefined // false
!!null // false
!!0 // false
!!'' // false
!!NaN // false

!!1 // true
!!'false' // true
!![] // true
!!{} // true
!!function(){} // true
!!/foo/ // true
```
```

最后，对于一些特殊值，`Boolean`对象前面加不加`new`，会得到完全相反的结果，必须小心。

```
```javascript
if (Boolean(false)) {
  console.log('true');
} // 无输出

if (new Boolean(false)) {
  console.log('true');
} // true

if (Boolean(null)) {
  console.log('true');
} // 无输出

if (new Boolean(null)) {
  console.log('true');
} // true
```
```