

Boolean 对象

`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。

```
let b = new Boolean(true);
b;           // Boolean {true}
typeof b;    // "object"
b.valueOf(); // true
```

`false` 对应的包装对象实例，布尔运算结果也是 `true`。

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

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

上例第一段 `false` 对应的包装对象的实例是一个对象，进行逻辑运算时，对象被转换为布尔值 `true`（所有对象对应的布尔值都是 `true`）。第二段实例对象的 `valueOf()` 方法返回实例对象的原始值（`false`）。

1. Boolean 函数的类型转换

`Boolean` 对象除了可以作为构造函数外，还可以单独使用，将任意的值转换为布尔值。这是 `Boolean` 就是一个单纯的工具函数。

只有六种值会被转换为 `false` 其它全部转换为 `true`。

```
Boolean(0);           // false
Boolean(NaN);         // false
Boolean("");          // false
Boolean(false);       // false
Boolean(undefined);   // false
Boolean(null);        // false
```

使用双重的否定运算符，也能将任意值转换为对应的布尔值。

```
!!0;           // false
!!NaN;         // false
!!"";          // false
!!false;       // false
```

```
!!undefined; // false
!!null;      // false
```

一些特殊的值，前面不加 `new` 会得到完全相反的结果。

```
!!Boolean(null);           // false
!!new Boolean(null);       // true, Boolean 对象对应的布尔值都是 true

!!Boolean(false);          // false
!!new Boolean(false);      // true, Boolean 对象对应的布尔值都是 true

!!Boolean(undefined);      // false
!!new Boolean(undefined);  // true, Boolean 对象对应的布尔值都是 true
```