

Boolean类型

Boolean 类型是与布尔值对应的引用类型。要创建 Boolean 对象，可以像下面这样调用 Boolean

构造函数并传入 true 或 false 值。

```
var booleanObject = new Boolean(true);
```

Boolean 类型的实例重写了 valueOf() 方法，返回基本类型值 true 或 false ；重写了 toString()

方法，返回字符串 "true" 和 "false" 。可是， Boolean 对象在 ECMAScript 中的用处不大，因为它经

常会造成人们的误解。其中最常见的问题就是在布尔表达式中使用 Boolean 对象，例如：

```
var falseObject = new Boolean(false);
```

```
var result = falseObject && true;
```

```
alert(result); //true
```

```
var falseValue = false;
```

```
result = falseValue && true;
```

```
alert(result); //false
```

BooleanTypeExample01.htm

在这个例子中，我们使用 false 值创建了一个 Boolean 对象。然后，将这个对象与基本类型值 true

构成了逻辑与表达式。在布尔运算中， false && true 等于 false 。可是，示例中的这行代码是对

falseObject 而不是对它的值（ false ）进行求值。前面讨论过，布尔表达式中的所有对象都会被转

换为 true ，因此 falseObject 对象在布尔表达式中代表的是 true 。结果， true && true 当然就等

于 true 了。

基本类型与引用类型的布尔值还有两个区别。首先， typeof 操作符对基本类型返回 "boolean" ，

而对引用类型返回 "object" 。其次，由于 Boolean 对象是 Boolean 类型的实例，所以使用 instanceof

操作符测试 Boolean 对象会返回 true ，而测试基本类型的布尔值则返回 false 。例如：

```
alert(typeof falseObject); //object
```

```
alert(typeof falseValue); //boolean
```

```
alert(falseObject instanceof Boolean); //true
```

```
alert(falseValue instanceof Boolean); //false
```

理解基本类型的布尔值与 Boolean 对象之间的区别非常重要——当然，我们的建议是永远不要使用 Boolean 对象。