检测类型

虽然在检测基本数据类型时 typeof 是非常得力的助手，但在检测引用类型的值时，这个操作符的用处不大。通常，我们并不是想知道某个值是对象，而是想知道它是什么类型的对象。为此，ECMAScript 提供了 instanceof 操作符，其语法如下所示：

```
result = variable instanceof constructor
```

如果变量是给定引用类型（根据它的原型链来识别；第 6 章将介绍原型链）的实例，那么 instanceof 操作符就会返回 true。请看下面的例子：

```
alert(person instanceof Object);    // 变量 person 是 Object 吗?
alert(colors instanceof Array);     // 变量 colors 是 Array 吗?
alert(pattern instanceof RegExp);   // 变量 pattern 是 RegExp 吗?
```

根据规定，所有引用类型的值都是 Object 的实例。因此，在检测一个引用类型值和 Object 构造函数时，instanceof 操作符始终会返回 true。当然，如果使用 instanceof 操作符检测基本类型的值，则该操作符始终会返回 false，因为基本类型不是对象。

使用 **typeof** 操作符检测函数时，该操作符会返回**"function"**。在 Safari 5 及之前版本和 Chrome 7 及之前版本中使用 **typeof** 检测正则表达式时，由于规范的原因，这个操作符也返回**"function"**。ECMA-262 规定任何在内部实现**[[Call]]**方法的对象都应该在应用 **typeof** 操作符时返回**"function"**。由于上述浏览器中的正则表达式也实现了这个方法，因此对正则表达式应用 **typeof** 会返回**"function"**。在 IE 和 Firefox 中，对正则表达式应用 **typeof** 会返回**"object"**。