


Object 类型

到目前为止，我们看到的大多数引用类型值都是 Object 类型的实例；而且，Object 也是 ECMAScript 中使用最多的一个类型。虽然 Object 的实例不具备多少功能，但对于在应用程序中存储和传输数据而言，它们确实是非常理想的选择。

创建 Object 实例的方式有两种。第一种是使用 new 操作符后跟 Object 构造函数，如下所示：



```
var person = new Object();
person.name = "Nicholas";
person.age = 29;
```

另一种方式是使用**对象字面量**表示法。对象字面量是对象定义的一种简写形式，目的在于简化创建包含大量属性的对象的过程。下面这个例子就使用了对象字面量语法定义了与前面那个例子中相同的 person 对象：

```
var person = {
  name : "Nicholas",
  age : 29
};
```

在使用对象字面量语法时，属性名也可以使用字符串，如下面这个例子所示。

```
var person = {
  "name" : "Nicholas",
  "age" : 29,
  5 : true
};
```

这个例子会创建一个对象，包含三个属性：name、age 和 5。但这里的数值属性名会自动转换为字符串。

另外，使用对象字面量语法时，如果留空其花括号，则可以定义只包含默认属性和方法的对象，如下所示：

```
var person = {};           //与 new Object()相同
person.name = "Nicholas";
person.age = 29;
```

虽然可以使用前面介绍的任何一种方法来定义对象，但开发人员更青睐对象字面量语法，因为这种语法要求的代码量少，而且能够给人封装数据的感觉。实际上，对象字面量也是向函数传递大量可选参数的首选方式，例如：

```
function displayInfo(args) {  
    var output = "";  
  
    if (typeof args.name == "string"){
```

图灵社区会员 StinkBC(StinkBC@gmail.com) 专享 尊重版权

5.1 Object 类型 85

```
        output += "Name: " + args.name + "\n";  
    }  
  
    if (typeof args.age == "number") {  
        output += "Age: " + args.age + "\n";  
    }  
  
    alert(output);  
}  
  
displayInfo({  
    name: "Nicholas",  
    age: 29  
});  
  
displayInfo({  
    name: "Greg"  
});
```



这种传递参数的模式最适合需要向函数传入大量可选参数的情形。一般来讲，命名参数虽然容易处理，但在有多个可选参数的情况下就会显示不够灵活。最好的做法是对那些必需值使用命名参数，而使用对象字面量来封装多个可选参数。

一般来说，访问对象属性时使用的都是点表示法，这也是很多面向对象语言中通用的语法。不过，

在 JavaScript 也可以使用方括号表示法来访问对象的属性。在使用方括号语法时，应该将要访问的属性

以字符串的形式放在方括号中，如下面的例子所示。

```
alert(person["name"]); // "Nicholas"
```

```
alert(person.name); // "Nicholas"
```

从功能上看，这两种访问对象属性的方法没有任何区别。但方括号语法的主要优点是可以通过变量来访问属性，例如：

```
var propertyName = "name";
```

```
alert(person[propertyName]); // "Nicholas"
```

如果属性名中包含会导致语法错误的字符，或者属性名使用的是关键字或保留字，也可以使用方括

号表示法。例如：

```
person["first name"] = "Nicholas";
```

由于 "first name" 中包含一个空格，所以不能使用点表示法来访问它。然而，属性名中是可以包

含非字母非数字的，这时候就可以使用方括号表示法来访问它们。

通常，除非必须使用变量来访问属性，否则我们建议使用点表示法。