

作为值的函数

因为 ECMAScript 中的函数名本身就是变量，所以函数也可以作为值来使用。也就是说，不仅可以像传递参数一样把一个函数传递给另一个函数，而且可以将一个函数作为另一个函数的结果返回。来看一看下面的函数。

```
function callSomeFunction(someFunction, someArgument){
    return someFunction(someArgument);
}
```

这个函数接受两个参数。第一个参数应该是一个函数，第二个参数应该是要传递给该函数的一个值。然后，就可以像下面的例子一样传递函数了。

```
function add10(num){
    return num + 10;
}


var result1 = callSomeFunction(add10, 10);
alert(result1);    //20

function getGreeting(name){
    return "Hello, " + name;
}

var result2 = callSomeFunction(getGreeting, "Nicholas");
alert(result2);    //"Hello, Nicholas"
```

这里的 `callSomeFunction()` 函数是通用的，即无论第一个参数中传递进来的是什么函数，它都会返回执行第一个参数后的结果。还记得吧，要访问函数的指针而不执行函数的话，必须去掉函数名后面的那对圆括号。因此上面例子中传递给 `callSomeFunction()` 的是 `add10` 和 `getGreeting`，而不是执行它们之后的结果。

当然，可以从一个函数中返回另一个函数，而且这也是极为有用的一种技术。例如，假设有一个对象数组，我们想要根据某个对象属性对数组进行排序。而传递给数组 `sort()` 方法的比较函数要接收两个参数，即要比较的值。可是，我们需要一种方式来指明按照哪个属性来排序。要解决这个问题，可以定义一个函数，它接收一个属性名，然后根据这个属性名来创建一个比较函数，下面就是这个函数的定义。



```
function createComparisonFunction(propertyName) {
    return function(object1, object2){
        var value1 = object1[propertyName];
        var value2 = object2[propertyName];

        if (value1 < value2){
            return -1;
        } else if (value1 > value2){
            return 1;
        } else {
            return 0;
        }
    };
}
```

这个函数定义看起来有点复杂，但实际上无非就是在一个函数中嵌套了另一个函数，而且内部函数前面加了一个 `return` 操作符。在内部函数接收到 `propertyName` 参数后，它会使用方括号表示法来取得给定属性的值。取得了想要的属性值之后，定义比较函数就非常简单了。上面这个函数可以像在下面例子中这样使用。

```
var data = [{name: "Zachary", age: 28}, {name: "Nicholas", age: 29}];

data.sort(createComparisonFunction("name"));
alert(data[0].name); //Nicholas

data.sort(createComparisonFunction("age"));
alert(data[0].name); //Zachary
```

这里，我们创建了一个包含两个对象的数组 `data`。其中，每个对象都包含一个 `name` 属性和一个 `age` 属性。在默认情况下，`sort()` 方法会调用每个对象的 `toString()` 方法以确定它们的次序；但得到的结果往往并不符合人类的思维习惯。因此，我们调用 `createComparisonFunction("name")` 方法创建了一个比较函数，以便按照每个对象的 `name` 属性值进行排序。而结果排在前面的第一项是 `name` 为 "Nicholas"，`age` 是 29 的对象。然后，我们又使用了 `createComparisonFunction("age")` 返回的比较函数，这次是按照对象的 `age` 属性排序。得到的结果是 `name` 值为 "Zachary"，`age` 值是 28 的对象排在了第一位。