



分类：[IT技术](#)

上一篇：[Javascript面向对象编程（二](#)

下一篇：[上海的酸雨](#)

Javascript面向对象编程（三）：非函数对象的继承

作者：阮一峰

日期：2010年5月24日

昨天，我本来打算，把这个话题结束了。

但是，写到一半的时候，我突然意识到，Javascript的[继承](#)有两种。一种是基于"函数对象"的继承，也就是一个function去继承另外一个function，这个我已经在昨天介绍过了。

另一种则是基于"非函数对象"的继承，不涉及function。它的做法与前一种情况完全不同。

一、什么是"非函数对象"的继承？

比如，现在有一个对象，叫做"中国人"。

```
var Chinese = {  
  nation:'中国'  
};
```

还有一个对象，叫做"医生"。

```
var Doctor = {  
  career:'医生'  
}
```

请问，我怎样才能让"医生"去继承"中国人"，也就是说，我怎样才能生成一个"中国医生"的对象？

二、**object()** 方法

json格式的发明人Douglas Crockford，提出了一个object()函数，可以做到这一点。

```
function object(o) {  
  
    function F() {}  
  
    F.prototype = o;  
  
    return new F();  
  
}
```

这个object()函数，其实只做一件事，就是把子对象的prototype属性，指向父对象，从而使得子对象与父对象连在一起。

使用的时候，第一步先在父对象的基础上，生成子对象：

```
var Doctor = object(Chinese);
```

然后，再加上子对象本身的属性：

```
Doctor.career = '医生';
```

这时，子对象已经继承了父对象的属性了。

```
alert(Doctor.nation); // 中国
```

三、浅拷贝

除了使用"prototype链"以外，还有另一种思路：把父对象的属性，全部拷贝给子对象，也能实现继承。

下面这个函数，就是在做拷贝：

```
function extendCopy(p) {  
  
    var c = {};  
  
    for (var i in p) {
```

```
        c[i] = p[i];  
    }  
  
    c.uber = p;  
  
    return c;  
}
```

使用的时候，这样写：

```
var Doctor = extendCopy(Chinese);  
  
Doctor.career = '医生';  
  
alert(Doctor.nation); // 中国
```

但是，这样的拷贝有一个问题。那就是，如果父对象的属性等于数组或另一个对象，那么实际上，子对象获得的只是一个内存地址，而不是真正拷贝，因此存在父对象被篡改的可能。

请看，现在给Chinese添加一个"出生地"属性，它的值是一个数组。

```
Chinese.birthPlaces = ['北京','上海','香港'];
```

通过extendCopy()函数，Doctor继承了Chinese。

```
var Doctor = extendCopy(Chinese);
```

然后，我们为Doctor的"出生地"添加一个城市：

```
Doctor.birthPlaces.push('厦门');
```

发生了什么事？Chinese的"出生地"也被改掉了！

```
alert(Doctor.birthPlaces); // 北京, 上海, 香港, 厦门
```

```
alert(Chinese.birthPlaces); // 北京, 上海, 香港, 厦门
```

所以，extendCopy()只是拷贝基本类型的数据，我们把这种拷贝叫做"浅拷贝"。这是早

期jQuery实现继承的方式。

四、深拷贝

所谓"深拷贝"，就是能够实现真正意义上的数组和对象的拷贝。它的实现并不难，只要递归调用"浅拷贝"就行了。

```
function deepCopy(p, c) {  
  
    var c = c || {};  
  
    for (var i in p) {  
  
        if (typeof p[i] === 'object') {  
  
            c[i] = (p[i].constructor === Array) ? [] : {};  
  
            deepCopy(p[i], c[i]);  
  
        } else {  
  
            c[i] = p[i];  
  
        }  
    }  
}
```

```
    }  
  
    return c;  
}
```

使用的时候这样写：

```
var Doctor = deepCopy(Chinese);
```

现在，给父对象加一个属性，值为数组。然后，在子对象上修改这个属性：

```
Chinese.birthPlaces = ['北京','上海','香港'];  
  
Doctor.birthPlaces.push('厦门');
```

这时，父对象就不会受到影响了。

```
alert(Doctor.birthPlaces); // 北京, 上海, 香港, 厦门
```



```
alert(Chinese.birthPlaces); // 北京, 上海, 香港
```

目前，jQuery库使用的就是这种继承方法。

(完)

文档信息

- 版权声明：自由转载-非商用-非衍生-保持署名 | Creative Commons BY-NC-ND 3.0
- 原文网址：http://www.ruanyifeng.com/blog/2010/05/object-oriented_javascript_inheritance_continued.html
- 最后修改时间：2010年5月24日 23:28

宅男福音！袜子也能包年?!

Google

男人袜

搜索一下



相关文章

- **2010.12.27:** [PHP最佳实践](#)

今天下午，我在读下面这篇文章。

- **2010.12.07:** [《黑客英雄》书摘](#)

《黑客与画家》翻译完成后，我还欠出版社一篇译者序。

功能链接

- 前一篇：[Javascript面向对象编程（二）：继承](#)
- 后一篇：[上海的酸雨](#)
- 更多内容请访问：[首页](#) » [档案](#) » [IT技术](#)

■ 站内搜索：

■ **Feed** 订阅：

广告

留言（7条）

jlake 说：

不错，概念解释得很清楚。

Micky 说：

Thanks！虽然原理和上一篇大致相同，但还是对我有所启发。尤其最后一个深拷贝方法。

2010年5月25日 09:07 | 档案 | 引用

netwix 说：

拷贝法很适合用在传递参数

比如

```
function foo(options){  
  options=$.extend({  
    name:'Default Name',  
    age:22,  
    man:true  
  },options);  
  // options.name ....  
}
```

需要深拷贝给\$.extend的第3个参数true

2010年5月25日 22:39 | 档案 | 引用

谦 说：

很好的专题

2010年5月29日 10:07 | 档案 | 引用

xniupp 说：

学到了好多东西。。

2010年6月 7日 18:27 | 档案 | 引用

simaopig 说：

在《JavaScript 设计模式》一书中有介绍，只不过方法叫clone

同时，对于这种引用型数据的继承，书中介绍的是采用构造函数的方法返回

例如：

```
function clone(obj){
```

```
function F(){};
F.prototype = obj;
return new F;
}
```

```
var Chinese = {
    nation:'中国',
    createBirthPlaces : function(){
        return ['北京','上海','香港'];
    }
};
```

```
var Doctor = clone(Chinese);
Doctor.career = '医生';
Doctor.birthPlaces = Chinese.createBirthPlaces();
//3
alert(Doctor.birthPlaces.length);
Doctor.birthPlaces.push('大连');
//4
alert(Doctor.birthPlaces.length);
```

//3

```
alert(Chinese.createBirthPlaces().length);
```

2010年7月22日 10:32 | 档案 | 引用

呵呵 说：

这个object()函数，其实只做一件事，就是把子对象的prototype属性，指向父对象，从而使得子对象与父对象连在一起。

这段话好像说的不太对，这个object函数在我看来是临时声明一个构造函数，使这个构造函数的PROTOTYPE属性指向需要继承的对象，然后生成一个实例并返回。

2010年9月25日 15:41 | 档案 | 引用

我要发表看法

您的留言（HTML标签部分可用）

您的大名：

«- 必填

电子邮件：

«- 必填，不公开

个人网址：

«- 我信任你，不会填写广告链接

记住个人信息？☐

发表

«- 点击按钮

