Javascript 深拷贝

2017-05-23 [前端大全](https://mp.weixin.qq.com/s?__biz=MzAxODE2MjM1MA==&mid=2651552207&idx=1&sn=cff4b5daf2ba785d1784d0a40a24bb20&chksm=8025ae0eb75227184de8099b35b0266abcc1f67ee8065d380fb37c80136247dd3ab952298f5a&scene=0&key=b10a7c153a57fbb9cbdfdc28209d0148dfbc084d6218cbedef233f37e0144b3a2aaaa6d8a0f91d156afe212938e3185405848f9c58853ca3d7bd27c0a54f3dec7e52de271661293144cf9e6ef4222b6a&ascene=14&uin=MTc2Njg4NjgwMA%3D%3D&devicetype=Windows+7&version=6204014f&pass_ticket=3Bu3Gz5q5EP%2BPrgq9UdfXzGuKzRWhVDD0z%2BqnoNBsfrSbpO5gJ9dSPgouXNye6Yu&winzoom=1##)

（点击上方公众号，可快速关注）

作者：伯乐在线/前端-小强

[如有好文章投稿，请点击 → 这里了解详情](http://mp.weixin.qq.com/s?__biz=MzAxODE2MjM1MA==&mid=402764500&idx=1&sn=cfcc178c7718d548b7cdc04758502bd9&scene=21#wechat_redirect)

javascript深拷贝是初学者甚至有经验的开发者，都会经常遇到问题，并不能很好的理解javascript的深拷贝。

**深拷贝(deepClone)？**

与深拷贝相对的就是浅拷贝，很多初学者在接触这个感念的时候，是很懵逼的。

**为啥要用深拷贝？**

在很多情况下，我们都需要给变量赋值，给内存地址赋予一个值，但是在赋值引用值类型的时候，只是共享一个内存区域，导致赋值的时候，还跟之前的值保持一直性。

看一个具体的例子

// 给test赋值了一个对象

var test = {

a: 'a',

b: 'b'

};

// 将test赋值给test2

// 此时test和test2是共享了同一块内存对象，这也就是浅拷贝

var test2 = test;

test2.a = 'a2';

test.a === 'a2'// 为true

图解：

这下就很好理解为什么引用值类型数据相互影响问题。

实现

实现一个深拷贝函数，就不得不说javascript的数值类型。

判断javascript类型

javascript中有以下基本类型

| **类型** | **描述** |
| --- | --- |
| undefined | undefined类型只有一个值undefined，它是变量未被赋值时的值 |
| null | null类型也只有一个值null, 它是一个空的对象引用 |
| Boolean | Boolean有两种取值true和false |
| String | 它表示文本信息 |
| Number | 它表示数字信息 |
| Object | 它是一系列属性的无序集合， 包括函数Function和数组Array |

使用typeof是无法判断function和array的，这里使用Object.prototype.toString方法。 默认情况下，每个对象都会从Object上继承到toString()方法，如果这个方法没有被这个对象自身或者更接近的上层原型上的同名方法覆盖(遮蔽)，则调用该对象的toString()方法时会返回”[object type]”，这里的字符串type表示了一个对象类型

function type(obj) {

var toString = Object.prototype.toString;

var map = {

    '[object Boolean]'  : 'boolean',

    '[object Number]'   : 'number',

    '[object String]'   : 'string',

    '[object Function]' : 'function',

    '[object Array]'    : 'array',

    '[object Date]'     : 'date',

    '[object RegExp]'   : 'regExp',

    '[object Undefined]': 'undefined',

    '[object Null]'     : 'null',

    '[object Object]'   : 'object'

};

return map[toString.call(obj)];

}

**实现deepClone**

对于非引用值类型的数值，直接赋值，而对于引用值类型（object）还需要再次遍历，递归赋值。

function deepClone(data) {

var t = type(data), o, i, ni;

if(t === 'array') {

    o = [];

}else if( t === 'object') {

    o = {};

}else {

    return data;

}

if(t === 'array') {

    for (i = 0, ni = data.length; i < ni; i++) {

        o.push(deepClone(data[i]));

    }

    return o;

}else if( t === 'object') {

    for( i in data) {

        o[i] = deepClone(data[i]);

    }

    return o;

}

}

这里有个点大家要注意下，对于function类型，博主这里是直接赋值的，还是共享一个内存值。这是因为函数更多的是完成某些功能，有个输入值和返回值，而且对于上层业务而言更多的是完成业务功能，并不需要真正将函数深拷贝。

但是function类型要怎么拷贝呢？

其实博主只想到了用new来操作一下，但是function就会执行一遍，不敢想象会有什么执行结果哦！o(╯□╰)o！其它暂时还没有什么好的想法，欢迎大家指导哦！

到这里差不多也就实现完了深拷贝，又有人觉的怎么没有实现浅拷贝呢？

**浅拷贝？**

对于浅拷贝而言，可以理解为只操作一个共同的内存区域！这里会存在危险！(。﹏。\*) 。

如果直接操作这个共享的数据，不做控制的话，会经常出现数据异常，被其它部分更改。所以应该不要直接操作数据源，给数据源封装一些方法，来对数据来进行CURD操作。

到这里估计就差不多了，但是作为一个前端，不仅仅考虑javascript本身，还得考虑到dom、浏览器等。

Element类型

来看下面代码，结果会返回啥呢？

Object.prototype.toString.call(document.getElementsByTagName('div')[0])

答案是[object HTMLDivElement]

有时候保存了dom元素， 一不小心进行深拷贝，上面的深拷贝函数就缺少了对Element元素的判断。而判断Element元素要使用instanceof来判断。因为对于不同的标签，tostring会返回对应不同的标签的构造函数。

function type(obj) {

var toString = Object.prototype.toString;

var map = {

    '[object Boolean]'  : 'boolean',

    '[object Number]'   : 'number',

    '[object String]'   : 'string',

    '[object Function]' : 'function',

    '[object Array]'    : 'array',

    '[object Date]'     : 'date',

    '[object RegExp]'   : 'regExp',

    '[object Undefined]': 'undefined',

    '[object Null]'     : 'null',

    '[object Object]'   : 'object'

};

if(obj instanceof Element) {

        return 'element';

}

return map[toString.call(obj)];

}

其它方式？

1. jquery的实现

详见<https://github.com/jquery/jquery/blob/master/src/core.js>

2. underscore的实现

详见https://github.com/jashkenas/underscore/blob/master/underscore.js

3. lodash的实现

详见https://github.com/lodash/lodash/blob/master/lodash.js

4. JSON实现

先通过JSON.stringify一下，然后再JSON.parse一下，就能实现深拷贝。但是数据类型只支持基本数值类型。

var obj = {

    a: 'a',

    b: function(){console.log('b')}

}

//在JSON.stringify的时候就会把function给过滤了。

JSON.stringify(obj)// "{"a":"a"}"

**小结**

这里大概总结了一下深拷贝，以及怎么实现一个深拷贝。在不同的场景下，要根据业务场景，判断是否需要使用深拷贝。

**参考文献**

winter-JavaScript中的类型 http://www.cnblogs.com/winter-cn/archive/2009/12/07/1618281.html

觉得本文对你有帮助？请分享给更多人

**关注「前端大全」，提升前端技能**

[阅读原文](https://mp.weixin.qq.com/s?__biz=MzAxODE2MjM1MA==&mid=2651552207&idx=1&sn=cff4b5daf2ba785d1784d0a40a24bb20&chksm=8025ae0eb75227184de8099b35b0266abcc1f67ee8065d380fb37c80136247dd3ab952298f5a&scene=0&key=b10a7c153a57fbb9cbdfdc28209d0148dfbc084d6218cbedef233f37e0144b3a2aaaa6d8a0f91d156afe212938e3185405848f9c58853ca3d7bd27c0a54f3dec7e52de271661293144cf9e6ef4222b6a&ascene=14&uin=MTc2Njg4NjgwMA%3D%3D&devicetype=Windows+7&version=6204014f&pass_ticket=3Bu3Gz5q5EP%2BPrgq9UdfXzGuKzRWhVDD0z%2BqnoNBsfrSbpO5gJ9dSPgouXNye6Yu&winzoom=1##)

阅读 9633

36[投诉](https://mp.weixin.qq.com/s?__biz=MzAxODE2MjM1MA==&mid=2651552207&idx=1&sn=cff4b5daf2ba785d1784d0a40a24bb20&chksm=8025ae0eb75227184de8099b35b0266abcc1f67ee8065d380fb37c80136247dd3ab952298f5a&scene=0&key=b10a7c153a57fbb9cbdfdc28209d0148dfbc084d6218cbedef233f37e0144b3a2aaaa6d8a0f91d156afe212938e3185405848f9c58853ca3d7bd27c0a54f3dec7e52de271661293144cf9e6ef4222b6a&ascene=14&uin=MTc2Njg4NjgwMA%3D%3D&devicetype=Windows+7&version=6204014f&pass_ticket=3Bu3Gz5q5EP%2BPrgq9UdfXzGuKzRWhVDD0z%2BqnoNBsfrSbpO5gJ9dSPgouXNye6Yu&winzoom=1##)

[写留言[https://res.wx.qq.com/mmbizwap/zh_CN/htmledition/images/icon/appmsg/icon_edit25ded2.png](javascript:;)](javascript:;)