



链滴


# JS 深浅拷贝

作者: [xfy196](#)

原文链接: <https://ld246.com/article/1586176881758>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

```

} else {<br>
  _obj[attr] = obj[attr];<br>
}<br>
}<br>
return _obj;<br>
}<br>
}<br>
[]<br>
var obj = {<br>
  a: 1,<br>
  b: {<br>
    str: "hhhh"<br>
  }<br>
}<br>
var newObj = deepClone(obj);<br>
console.log(newObj, newObj.b === obj.b) // false</p>
<ul>
<li>案例(拷贝数组)</li>
</ul>
<p>function deepClone(obj, type) {<br>
  []<br>
  var _content = type === "array" ? [] : {};<br>
  for (var attr in obj) {<br>
    []<br>
    for (var attr in obj) {<br>
      // 如果不是引用数据类型我们直接复制就行了 如果是引用数据类型我们需要判断然后创建一个新的象在进行赋值<br>
      if (typeof obj[attr] === "object" && obj[attr] !== null && obj[attr].constructor && obj[attr].constructor === Object) {<br>
        // 是对象<br>
        _content[attr] = deepClone(obj[attr]);<br>
      } else if (Array.isArray(obj)) {<br>
        _content[attr] = deepClone(obj[attr], "array");<br>
      }<br>
    } else {<br>
      _content[attr] = obj[attr];<br>
    }<br>
  }<br>
  return _content;<br>
}<br>
}<br>
var obj = {<br>
  a: 1,<br>
  b: [1, 2, 3, 4, 5],<br>
  c: {<br>
    str: "hello"<br>
  }<br>
}<br>
var newObj = deepClone(obj);<br>
console.log(newObj);</p>
<ul>
<li>案例(深拷贝黑科技)</li>
</ul>
<p>// 函数在拷贝过程中怎么处理 这时候有个滑头的操作<br>

```

```
var obj1 = {<br>
foo : function(){<br>
console.log(1);<br>
}<br>
}<br>
// 将函数先转成字符串在使用 eval 将可执行的 js 代码成功转换<br>
var newFn = eval("(" +obj1.foo.toString()+")");<br>
console.log(newFn)<br>
[]<br>
// 黑科技深拷贝 一般人我不告诉他<br>
var obj1 = {<br>
a: 1,<br>
b: 2,<br>
c: {<br>
str: "hello"<br>
},<br>
d: [1, 2, 3, 4],<br>
f: function () {<br>
console.log(1);<br>
}<br>
}<br>
var obj2 = JSON.parse(JSON.stringify(obj1));<br>
console.log(obj2); // 虽然这个黑科技好用但是也要慎重使用因为是 ES5 中支持的而且有的时候在处数据的转换的时候会出现异常情况</p>
<h3 id="总结">总结</h3>
<p>我觉得深浅拷贝的含义并不难理解，只需要搞懂 JS 的数据类型的存储结构和特点就可以轻松理什么是深浅拷贝了。</p>
```