

IOS6时代的JAVASCRIPT与OC通信

JAVASCRIPT调用OBJE-TIVE-C

```
1 function callOC2(func,param){
2     var iframe = document.createElement("iframe");
3     var url= "myapp:" + "&func=" + func;
4     for(var i in param)
5     {
6         url = url + "&" + i + "=" + param[i];
7     }
8     iframe.src = url;
9     iframe.style.display = 'none';
10    document.body.appendChild(iframe);
11    iframe.parentNode.removeChild(iFrame);
12    iframe = null;
13 }
14
15 //使用方法
16 <input type="button" value="传个字典2" onclick="callOC2('testFunc',{'param1':76,'param2':155,'param3':76})" />
```



JavaScript以及HTML代码













































































































































































JAVASCRIPT调用OBJECTIVE-C

Javascript 以及 HTML 代码

```
1  function callOC2(func,param){
2      var iframe = document.createElement("iframe");
3      var url= "myapp:" + "&func=" + func;
4      for(var i in param)
5      {
6          url = url + "&" + i + "=" + param[i];
7      }
8      iframe.src = url;
9      iframe.style.display = 'none';
10     document.body.appendChild(iframe);
11     iframe.parentNode.removeChild(iFrame);
12     iframe = null;
13 }
14
15 //使用方法
16 <input type="button" value="传个字典2" onclick="callOC2('testFunc',{'param1':76,'param2':155,'param3':76})" />
```

这里为什么使用 Iframe , 动态添加 iframe改变其地址, 发出请求 (发出的请求会被上述的代码方法拦截), 这种方法不会导致当前的页面跳转, 用户无感知, 效果更佳

[HTML<iframe> 标签](#)

OC 调用 JAVASCRIPT