普通的时候，我们对如下的代码：

[?](http://my.oschina.net/mingtingling/blog/113818)

|  |  |
| --- | --- |
| 1  2  3  4 | <ul>      <li></li>      <li></li>  </ul> |

        如果要监听li的click事件怎么做，有童鞋可能就说了，首先获取li这个元素，如document.getElementsByTagName，然后对这两个元素分别监听click,如果IE就是用attachEvent，如果是w3c浏览器，就使用addEventListener，如果是比较老的浏览器就使用xx.onclick=yy。

       如果现在在文档中多添加了几个li，那么是不是又要重新获取新增的几个元素，并且监听click？

       这样做还存在一个问题，就是效率很低下。

       那么我们可不可以换一种思路呢，监听ul，因为浏览器的事件都支持冒泡，li的click会冒泡到ul，只要在ul click的时候判定事件的target即可。

        在获取target的时候，由于ie的event对象有srcElement属性但没有target属性，w3c的浏览器有target属性，但没有srcElement，那么我们可以写一个函数：

 　　　function \_getTarget(event) {return event.target || event.srcElement;}

       那么上述的事件监听改为如下：

[?](http://my.oschina.net/mingtingling/blog/113818)

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | on(ul,"click",function(e) {      var target = \_getTarget(e);      if(target.tagName.toLowerCase() == "li") {          XXXXX      }  }); |

        这里的on实际上是监听事件的函数，ul是通过某种方式抓取到的ul这个元素。

         上面这种方式就是事件代理。