

需求：

1. 点击发表评价在当前评价列表中增加一条数据
2. 点击删除评价在当前评价列表中删除一条数据
3. 点击发表和删除时评价总条数要跟着变化

传统开发方式：

给发表评价、删除两个按钮绑定事件，删除的事件还好，只需要想办法找到当前行所在元素并删除，然后把总条数减1即可，稍微一想就知道要封装到一个函数中：

function delComment(){

$(this).parents(".item").remove();

$(".total-txt.num").text(++totalAmount);

}

而发表所绑的事件就比较复杂了，需要先创建一个li，然后再创建两个元素（评论内容和删除按钮），然后在li中append进去这两个元素，最后再把li整体append到外层的ul中，并且如果不用事件委托的话还得给删除按钮绑事件，然后还不能忘了给总条数加1：

var totalAmount = 3;

$(".publish-comment").click(function(){

$(".list").append(

$("<li>").append(

$("<div>").addClass("content").text($("input.comment").val())

).append(

$("<a>").addClass("del").attr({"href": "javascript:;"}).text("删除").click(delComment)

)

);

$(".total-txt.num").text(++totalAmount);

});

可以看到，传统开发方式中的不足：

1. 操作DOM极为繁琐
2. 将DOM元素删除或添加了以后的后续工作很多，比如我们这里的更新总评价条数

这仅仅是举了一个例子，在实际的开发过程中再加上分页，再加上ajax异步操作，情况会更为复杂

此时如果你想直接刷新一遍页面就完事的话，请回家闭门思过。

后来Angular的出现为我们解决了这一痛点，Angular的基本思路如下：

首先，我们所有的评价肯定是通过ajax请求得到的，通常其格式如下：