



imweb-next @imweb-teacher 在 1 个月之前 评论

主程序员 ☐ ☐ ☐

总体评价

基本功能都实现了，在代码中还看到你使用了ES6的字符串模板的语法，让我有眼前一亮的感觉，这些都是值得肯定。
接下来说说问题，在批改你的代码过程中，发现部分代码写得还不错（如放大功能和模板函数的实现），但是还是暴露了许多的问题。如一些代码写法不太好（即代码规范）和代码的组织有点乱（逻辑有点混乱的）
对于整个项目完成我比较关注两个点的完成，一个是功能上的，一个是代码组织上的，因此针对这两个方面列出下面的一些问题和建议哈。

功能问题

功能上还没有完成实现好，如下：

1. 点赞人问题

点赞人应该是当前操作的人，而不是发朋友圈的人。如这里点赞阳和和伟科大人的朋友圈，不应该是阳和点赞阳和的朋友圈，伟科大人点赞伟科大人的朋友圈，如下所示：



2. 先评论再点赞展示错误

如果在没有点赞和评论的朋友圈消息中，先进行评论，评论后再点击点赞，会出现点赞人展示面板置于评论面板下面，如下所示：





代码问题

以下是

1. 函数命名尽量驼峰命名，只有是类的构造函数才首字母大写

如 [第159行] 下面的代码应该统一用驼峰的命名方式。

```
36 }
37 /**
38  * 多张图片消息模版 (可参考message.html)
39  * @param {Object} pics 多张图片消息的图片列表
40  * @return {String} 返回html字符串
41  */
42 function multiplePicTpl(pics) {
43     var htmlText = [];
44     if(pics.length>0){
45         htmlText.push('<ul class="item-pic">');
46         for (var i = 0, len = pics.length; i < len; i++) {
47             htmlText.push('')
48         }
49         htmlText.push('</ul>');
50     }
51     return htmlText.join('');
52 }
53 /**
54  * 实现分享消息模版
55  * @param {Object} pic 实现分享消息的图片列表
56  * @param {Object} text 分享消息文本
57  * @return {String} 返回html字符串
58  */
59 function Sharemessage(pic,text) {
60     var htmlText = [];
61     htmlText.push('<ul class="item-pic">');
62     htmlText.push('<span class="pic-text">'+
63     htmlText.push('</ul>');
64     return htmlText.join('');
65 }
66 /**
67  * 单一图片消息模版
68  * @param {Object} pics 单一图片消息的图片列表
69  * @return {String} 返回html字符串
70  */
71 function Singlemessage(pics) {
72     var htmlText = [];
73     htmlText.push('<ul class="item-pic">');
74     htmlText.push('')
75     htmlText.push('</ul>');
76     return htmlText.join('');
77 }
```

2. 要善用库提供的工具函数

如果我们使用到 jQuery 库，应当活用库，如 [第256行] 代码这里应当使用 jQuery 的 `.hasClass` 方法来判断是否是含有类名,而不是使用 `className` 属性来判断。(并且这里使用 `e.target.className != "item-reply"` 不能判断元素拥有多个 class 的情况。)

3. 代码功能尽量单一

绑定事件做了太多事情了，不同的事件绑定应尽量拆分来写，这样更具可读性。如下面的代码：

```
4 $( '.moments-item' ).on( 'click', function( e ) {
5     let status=0;
6     if( e.target.className != 'item-reply' ) {
7         $( '.item-reply-btn' ).siblings( '.com-status' ).removeAttr( 'style' );
8         var item_name=$( e.target ).parents( '.item-ft' ).siblings( '.item-name' ).html();
9         if( e.target.className == 'ck-zan' ) {
10             let nuRel=$( e.target ).attr( 'rel' );
11             if( !nuRel ) {
12                 status++;
13                 $( e.target ).attr( 'rel', status ).html( '取消' );
14                 let zone=$( e.target ).parents( '.item-ft' ).siblings( '.reply-zone' ).find( '.reply-like' );
15                 if( !zone.length ) {
16                     $( e.target ).parents( '.item-ft' ).siblings( '.reply-zone' ).append( '<div class="reply-like"><i class="icon-like-blue"></i>
17                     <a class="reply-who" href="#">${$.trim(item_name)}</a></div>' );
18                 } else {
19                     $( zone ).append( '<a class="reply-who" href="#">${$.trim(item_name)}</a>' );
20                 }
21             } else {
22                 $( e.target ).attr( 'rel', '' ).html( '点赞' );
23                 let reply_who=$( e.target ).parents( '.item-ft' ).siblings( '.reply-zone' ).find( '.reply-who' );
24                 if( reply_who.length > 1 ) {
25                     for( var i=0, j=reply_who.length; i<j; i++ ) {
26                         console.log( $( reply_who ).eq( i ).html() );
27                         if( $( reply_who ).eq( i ).html() == ' ' + item_name ) {
28                             $( reply_who ).eq( i ).remove();
29                         }
30                     }
31                 }
32             }
33         }
34     }
35 }
```



```
var $target = $(e.target);
var $replyZone = $(e.target).parents('.item-ft').siblings('.reply-zone');
```

5. 使用数据来维护状态

在基础代码中，已经提供了一个 data 的数据变量，应该通过数据变量来判断当前用户是否已经点赞了该朋友圈。而不是默认设定为所有朋友圈都是没有点赞过的。如代码中设定一个属性值 rel 来判断用户是否是点赞，该值初始化的值应当设置为数据 data 中每条消息的 hasLiked 的值。

```
contentHtml = sharemessage(content.share_pic,content.share_text);
break;
case 2:
// TODO: 实现单张图片消息
contentHtml = Singlemessage(content.pics);
break;
case 3:
// TODO: 实现无图片消息
contentHtml = multiplePicTpl(content.pics);
break;
}
htmlText.push(contentHtml);
// 消息时间和回复按钮
htmlText.push('<div class="item-ft">');
htmlText.push('<span class="item-time">' + content.timeString + '</span>');
htmlText.push('<div class="com-status"><span class="ck-zan" rel="">点赞</span><span class="ck-pin">评论</span></div><div class="item-r');
htmlText.push('<span class="item-reply"></span></div>');
htmlText.push('</div></div>');
// 消息回复模块（点赞和评论）
htmlText.push(replyTpl(messageData.reply));
htmlText.push('</div></div>');
return htmlText.join('');
```

不能认为默认就是空，应当取决 hasLiked 的值

data中每条消息是提供一个标识符变量 hasLiked 来表示是否点赞的

reply 对象组成说明

属性名	类型	备注
hasLiked	Boolean	自己是否有对这条消息进行点赞
likes	Array	消息已点赞的用户列表
comments	Array	消息相关的评论信息列表

6. 命名可读性

在代码中，变量命名追求可读性和统一性，如下面的代码，定义的 fstatus 和 _pinthis ,单独看命名上较难明白变量的用意且这里在命名前一时 _ 作为前缀用意是？如果想不到很好的变量命名，可通过注释来提高变量的可读性。

7. 代码空格问题

这边发现代码中空格使用的比较少，许多表达式都是十分紧凑的，虽然不空格不会影响代码执行，但适当的空格能让代码看起来更舒服直观（毕竟代码需要给其他人看的），如【第275行】的这段代码：

```
for(var i=0,j=reply_who.length;i<j;i++){
}

// 可修改为
for(var i = 0, j = reply_who.length; i < j; i++) {
}
```

8. 流程控制有点乱

如下面【第289行】代码，在 if 和 else 两条分支下，都执行了同样的操作。

```
}
}else{
let fstatus=$(e.target).parent('.item-reply-btn').siblings('.com-status');
if(fstatus.attr('style')){
$('.item-reply-btn').siblings('.com-status').removeAttr('style');
}else{
$('.item-reply-btn').siblings('.com-status').removeAttr('style');
$(e.target).parent('.item-reply-btn').siblings('.com-status').show().css('animation','ck-pinfir 0.3s ease 1');
}
}
if(e.target.className!='ck-pin'){
$('body').find('.bot-pintext').remove();
}
```

两种情况都符合 则不需要放在流程控制中

这里可以把代码改成

```
// 都会执行的操作，故不放在if else中
$('.item-reply-btn').siblings('.com-status').removeAttr('style');
// 如果没有style
if(!fstatus.attr('style')){
$(e.target).parent('.item-reply-btn').siblings('.com-status').show().css('animation','ck-pinfir 0.3s ease 1');
}
```

9. ninstatus 变量无意义

5. pinstatus 变量初始化

如 [第316行] 开始的下面的事件判定中 pinstatus 的变量是否是无意义的？没有这个判断视乎不会影响代码执行结果

```
/**
 * 发送评论信息按钮事件
 */
$('body').on('click', '.bot-pintext button', function(){
  /**
   * 通过评论状态来控制评论区的初始状态
   */
  let pinstatus=false; // 初始设置 false
  /**
   * 获取评论内容和确定是由哪里来的评论，把评论放上去。
   */
  if(!pinstatus){ // 则 pinstatus 肯定为 true
    let vtext=$(this).siblings('input').val();
    let pinname=_pinthis.parents('.item-ft').siblings('.item-name').html();
    _pinthis.parents('.item-ft').siblings('.reply-zone').find('.reply-comment').show();
    _pinthis.parents('.item-ft').siblings('.reply-zone').find('.reply-comment').append('<div class="comment-item">
      <a class="reply-who" href="#">${pinname}</a>: ${vtext}</div>');
    pinstatus=true; // 再设置为true
  }
  /**
   * 评论发送完成之后把input里的内容清空，并把评论区给删除掉。
   */
  if(pinstatus){ // 则 pinstatus 肯定为 true
    $(this).siblings('input').val('');
    $('body').find('.bot-pintext').remove();
  }
});
```

建议

在代码的组织和写法上中存在着较多的问题，这里建议使用 一对一辅导 的机会。老师给你整体梳理一下这个项目可能会有更多的理解。
编辑时间 1 个月之前

imweb-next @imweb-teacher 已关闭 1 个月之前 评论



编写 预览

支持 Markdown 及 slash commands 格式

重新打开 merge request