

7.26海鹏codereview总结

昨天codereview过程中提到频率较高的问题有

| 序号 | 名称 | 频数 |
|----|----------------|----|
| 1. | 代码简化, 业务处理代码抽取 | 6+ |
| 2. | 异常处理 | 4+ |
| 3. | 编码语义化 | 3+ |

以下就昨天codereview总结分类后得出的需要改进注意的地方为：

个人习惯

1. 合理添加注释
 - 举个🌰 在定义函数时要对函数名称、形参个数、形参意义、形参类型与实现的功能和内部逻辑做简要介绍
2. 相同类型、功能的代码要统一管理起来
 - 这样可以使得代码结构更清晰, 批量改正是更方便, 代码重用性更好
 - 举个🌰 将所有tpl模板写在一起, 在新增内容时即可考虑能否调整现有部分代码即可适应新需求; 将业务功能代码声明放在一起, 这样可以一目了然的看出这一js中都解决了哪些问题, 后续需要时直接调用即可
3. 寻找Dom节点时要注意缩小范围
 - 由于我们在代码编写之初定义了命名空间【mod-xxx-xxx】，因此内部类名都相对较简短, 如果不限定查找范围, 可能会获取到并不是你需要的那个节点; 同时这样操作也使得效率更高
4. 遇到问题未解决时要有记录
 - 在开发过程中难免遇到问题, 可能当时你使用了取巧的方式临时解决了麻烦, 但是长久来看是不合适的。这个时候一定要对问题进行记录, 标明适宜的注释与【TODO】, 以便后续优化
5. 注意遵守已有的规则和使用习惯
 - 举个🌰 在使用ajax请求时, 一般情况下我们都约定使用 response.ret 来判断请求状态, 而 response.msg 一般只作为错误信息显示, 因此在开发过程中不要改变这样的习惯, 不利于其他同学查看和理解。
6. 多次使用的变量要缓存起来, 不要多次去取用
 - 举个🌰 Array.length 可能多次会用到, 那就用 var LEN = Array.length 保存起来, 后续使用时直接取用即可
7. 业务功能处理代码和逻辑代码要进行分离, 以便于后续代码抽出
 - 具体示例见后面代码简化部分

语义化

1. 标签使用时要注意规范与语义化
 - 例如何时使用 `<label>`、``、`<a>` 等标签, 注意使用场景, 反例 ``
2. 变量(首字母小写后续驼峰)、常量(大写)、tpl模板(语义化内容)、Dom节点获取、类名(XXX-XXX)等都要注意遵守相关规范和要求
3. 起名时注意不要起带有迷惑性的易混淆的名称, 同一页面内尽量避免类似名称, 例如获取节点时 `select.region` 和 `region-select` 这样的命名方式就存在疑惑

代码简化

主要涉及三块内容：模板、异步请求、业务逻辑处理

1. 编写tpl模板时要考虑多情况下的重用性, 避免冗余
2. 当页面中多次用到ajax请求时封装为syncData方法比较合适, 只需传入相应的参数即可实现功能, 业务处理放在另外的函数中进行
3. 业务处理代码单独封装为函数, 需要处理时直接调用, 使得代码模块化, 调用方便结构清晰

异常处理

主要是在[数据来源不可靠](#)(例如后端传递、用户输入等)时要着重思考异常处理, 这一步骤并非必须, 一定要综合实际情况来进行分析, 若需要进行则主要注意以下几方面内容:

1. 是否传入了参数
 - 举个🌰 `if(! opts) return`
2. 参数类型是否正确
 - 举个🌰 `if(typeof callback != function) return`
3. 设置默认值提高兼容性
 - 举个🌰 `var count = response.list.length || 0`

细节处理

1. 给Dom节点赋值属性时使用 `""` 包裹, 避免低版本浏览器报错
 - 举个🌰 `<input type="text" value="{{item.value}}>`
2. 注意zepto与jQuery使用习惯差异, 两者具有的函数不完全相同
 - 举个🌰 新版本的jQuery中ajax请求已经提供了 `.done()` .`fail()` 方法的使用, 而zepto仍旧只支持 `success:function(){},error:function(){}` 的方式, 在使用过程中要注意
3. 与用户交互多的页面场景要考虑到系统关闭自带alert的情况, 尽量使用封装好的Alert组件
 - 举个🌰 TMS中每当触发错误时都会弹出相应不同的问题信息, 用户交互较多, 因此要使用封装好的Alert组件
4. 函数定义方式使用 `声明` 或 `字面量` 方式要考虑实际应用场景
 - 一般情况下的编码逻辑为定义函数后才会使用, 很少出现先使用后定义的情况, 当然如果使用声明的方式定义函数, 这样使用也不会报错。因为会有声明提前

jQuery方法使用问题

这一块的内容主要是给大家引出问题(因为很多我不是很清楚, 就不误导大家了, 希望大家能多查询相关的文章做一下研究)

1. `html()` `text()` 的效率问题
 - 只涉及到文本编辑时要使用 `text()` 方法, 效率更高, 因为 `html()` 要考虑的兼容性情况更多
 - 在使用 `html()` 或 `append()` 方法时要做分析, 当然要少的使用 `html()` 方法, 因为个人对这里理解不多, 因此算是给大家引出这个内容, 需要自己去查一下相关的文章来看下
2. `data()` `attr()` `prop()` 方法的异同与效率问题
 - 这个问题也是作为给大家引出的问题, 希望大家都能自己查一下相关的文章
3. `each()` 与 `map()` 方法的适用场景
4. `jQuery` 中哪些方法效率最低应该避免使用? (老丁提出了`attr`、`css`两个方法, 更多的也需要查一下)