**WEB前端开发规范文档+CSS命名规范**

**规范目的**  
　　为提高团队协作效率,便于后期优化维护, 输出高质量的文档, 特制订此文档. 本规范文档一经确认, 前端开发人员必须按本文档规范进行前台页面开发. 本文档如有不对或者不合适的地方请及时提出, 经讨论决定后方可更改.  
  
**基本准则**  
　　符合web标准, 语义化html, 结构表现行为分离, 兼容性优良. 页面性能方面, 代码要求简洁明了有序, 尽可能的减小服务器负载, 保证最快的 解析速度.  
   
**文件规范**  
　　1. html, css, js, images文件均归档至<系统开发规范>约定的目录中;  
　　2. html文件命名: 英文命名全部小写, 后缀.html. 同时将对应界面稿放于同目录中, 若界面稿命名为中文, 请重命名与html文件同名, 以方便后端添加 功能时查找对应页面;  
　　3. css文件命名: 英文命名, 后缀.css. 共用dome.css, 首页index.css, 其他页面依实际模块需求命名;  
　　4. js文件命名: 英文命名, 后缀.js. 共用common.js, 其他依实际模块需求命名.

**html书写规范**  
　　1. 文档类型声明及编码: 统一为html5声明类型<!DOCTYPE html>; 编码统一为<meta charset="utf-8" />, 书写时利用2个空格严禁使用tab实现层次分明的缩进;  
　　2. 非特殊情况下样式文件必须外链至<head>...</head>之间;非特殊情况下JavaScript文件必须外链至body标签底部;  
　　3. 页面中不要写样式;

4.不要嵌套太深了,麻烦;

5.非特殊情况不要添加空的元素来搞事;

**css书写规范**

1. 编码统一为utf-8;  
2. class与id的使用: id是唯一的并是父级的, class是可以重复的并是子级的, 所以id仅使用在大的模块上, class可用在重复使用率高及子级 中;

3. class与id命名: 大的框架命名比如header/footer/wrapper/left/right之类的在2中由i统一命名.其他样式名称由小写英文 & 数 字 & \_ 来组合命名, 如i\_comment, fontred, width200; 避免使用中文拼音, 尽量使用简易的单词组合; 总之, 命名要语义化, 简明 化;

4. css属性书写顺序, 建议遵循:  布局定位属性-->自身属性-->文本属性-->其他属性. 此条可根据自身习惯书写, 但尽量保证同类属 性写在一起;

5. 书写代码前, 考虑并提高样式重复使用率;  
6. 充分利用html自身属性及样式继承原理减少代码量;

7. 背景图片请尽可能使用sprite技术, 减小http请求, 考虑到多人协作开发, sprite按模块制作;

8. 必须为大区块样式添加注释, 小区块适量注释;适当添加注释;

9. 代码缩进与格式

**JavaScript书写规范**  
　　1. 文件编码统一为utf-8, 书写过程过, 每行代码结束必须有分号; 原则上所有功能均根据咪咕项目需求原生开发, 以避免网上down下来的代码造 成的代码污染(沉冗代码 || 与现有代码冲突 || ...);  
　　2. 库引入:框架包含所有预期需要的库, 若需引入第三方库, 须与团队其他人员讨论决定;  
　　3. 变量命名: 小驼峰式命名.

4. 类命名: 首字母大写, 驼峰式命名. 如 ITaoLun;  
　　5. 函数命名: 首字母小写驼峰式命名. 如iTaoLun();  
　　6. 命名语义化, 尽可能利用英文单词或其缩写;  
　　7. 尽量避免使用存在兼容性及消耗资源的方法或属性, 比如eval() & innerText;  
　　8. 后期优化中, JavaScript非注释类中文字符须转换成unicode编码使用, 以避免编码错误时乱码显示;  
　　9. 代码结构明了, 加注释. 提高函数重用率;  
　　10. 注重与html分离, 减小reflow, 注重性能.

js建议采用面向对象的风格,不强求哈,我也没有怎么实践过;

js文件写上作者名字;

给重要的方法加上详细的注释;写上功能，参数，返回值，重要的逻辑也加上

封装的组件尤其要写好注释，组件要写好点，不要因为不符合自己需求就重新写组件，把原来的组件改好点

**测试**

想学习的可以研究哈如何写单元测试，自动化测试可以试试puppeteer，仅针对chrome；

需要兼容性测试可以试试nightwatch

不想写就交给测试妹子

自己写好的页面多点点，注意细节