1. **前言**

为提高项目质量以及项目的可维护性，规范前端代码避免不必要的bug产生，节省人力成本和时间成本，书写此规范，开发人员务必按照此规范进行前端开发。此文档如有不对的或者补充的地方请指出。

1. **整体规范**
   1. 代码缩进必须使用空格字符的tab，使用代码格式化工具进行代码格式化
   2. Html，css，js代码分离，尽量避免出现行内样式 。js代码放在页面下方，css代码放在head标签内。
   3. 在编写网页的时候，前端代码应该保持简洁与克制，即用最简单的代码完成布局要求，用css代码代替js动画。
   4. 书写js代码时保证业务代码与逻辑代码的拆分，有必要的提取为公共js，css同理
   5. 布局动态化，即时刻考虑文本内容与屏幕环境对布局的影响
   6. 兼容性测试，即保持跨浏览器表现的一致性，使得浏览器能兼容各个版本的浏览器（原则上不考虑IE8）, 优雅降级
   7. Css类代码使用中线式命名法 (“test-demo”) , id属性使用驼峰式命名 （”testDemo”）。
   8. 文件命名语义化，禁止出现空格。
   9. 禁止随意更改框架或插件源码，根据业务逻辑选择插件，导入插件，不得滥用。
2. **HTML规范**
   1. 页面结构符合web标准，以还原设计稿为目标。
   2. 标签语义化，禁止自定义标签。
   3. 标签必须闭合。
   4. 页面id不得重复。
   5. 页面标签层次结构保证清晰，使用注释,避免使用空格占位。
   6. 在有框架的情况下（如bootstrap），尽量套用框架代码。
   7. meta标签声明页面字符集 (<meta charset="UTF-8">)
   8. 使用H5标准文档格式 (<!DOCTYPE html>)
   9. 兼容ms 浏览器兼容模式 （<meta http-equiv="X-UA-Compatible" content="ie=edge">）
   10. 360浏览器需要强制开启极速模式 (<meta name="renderer" content="webkit">)
   11. 手机端meta标签参考
       1. 
3. **CSS 规范**
   1. 使用css3 时考虑个版本浏览器兼容性，可使用autoprefixer做代码补全。
   2. CSS代码与页面分离，尽量不用import导入css。
   3. 流式布局与响应式布局优先，按需使用绝对定位，绝对定位方式作为最后的方案。
   4. 减少css后代选择器的选择层级。
   5. 使用类选择器代替id选择器添加样式。类的命名应该简洁明了。
   6. 在没使用到框架的情况下，考虑使用reset或者normalize重置样式。
   7. 针对动画数量较多，页面动画卡顿的情况，按需使用css开启硬件加速。
   8. 导入页面的css文件，禁止出现重复导入的情况。禁止导入无用的css。
   9. 带着全局意识编写css代码，做到类最大限度的复用，减少css的代码量与文件体积。
   10. CSS代码必须使用 ‘ ；’ 闭合。
   11. 如果类属性为 ‘0’ 则不加单位，尽量做到属性合并（如margin-top与margin-bottom）
   12. 根据场景使用！important。
   13. 使用十六进制色彩表示法时，做简化处理（如#ffffff = >#fff）
   14. 使用带前缀的命名空间可以防止命名冲突，同时提高代码可维护性。
   15. 类属性层级清晰，可遵循的方式
       1. 根据属性作用分类（如color与backgournd为一类）
       2. 根据字母顺序分类（a>b>c）
4. **Javascript 规范**
   1. 目前无前端工程化的项目使用ES5（ES6只有少数浏览器支持）。
   2. 功能代码与功能代码分离。
   3. 文件头部添加‘use strict’声明。
   4. 减少全局变量的使用。
   5. 组件中使用请求的务必使用异步。
   6. Ajax请求如非必要，禁止使用同步。同步场景可用promise或defer替代。
   7. 业务代码添加说明注释，功能代码添加参数说明。减少多余的命名前缀，函数名简洁易懂。
   8. 考虑代码安全性，根据使用场景使用try…catch，捕获异常。
   9. 避免使用直接使用undefined或者null做真假判断 如： var a = null; if(a==null){//to do} 。
   10. 写功能代码时尽量做到代码解耦，这样可提高代码的复用性。
   11. 数据动态化，常量使用大写字母声明。
   12. 原生js代码考虑浏览器事件兼容性。
   13. 使用分号“ ；”结束。
   14. Js公共方法应尽量减小颗粒度，遵循软件工程的一个函数制作一件事情的原则。这样有利于代码复用和后期的维护升级。
5. **网页性能规范**
   1. 图标尽量使用字体图标或者精灵图代替。
   2. 多图网站图片设置懒加载。
   3. 使用cdn作为静态公共资源的引入。
   4. 必要的地方考虑使用按需异步加载。
   5. 样式文件与js文件注意摆放位置。
   6. 按场景合并样式文件与js文件。
6. 文档注释规范

对于JavaScript的公共方法应该有文档注释说明，以提高代码的可读性。

/\*\* 此处为 js 代码\*/

对于包含参数的方法，使用@params 添加参数类型以及参数说明

对于包含返回值的方法，使用@return 添加返回值类型及说明

如需提供方法使用示例，使用@example 添加方法使用说明

示例如下：

