## 魔鬼程序员—美团项目规范

**最佳原则**

坚持制定好的代码规范，无论团队人数多少，代码应该同出一门

**————————————————————————————————————**

**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

**————————————————————————————————————**

**目录规范**

mg-mt/

|—static/ //项目公共目录

| |—css/ // CSS

| |—images/ //图片

|—规范.doc //文档

|—js/ //js文件夹

|—page/ //子页面

|—index.html //首页

**————————————————————————————————————**

**命名规范**

* **项目命名**
* mg-mt
* **首页页面**
* index.html
* **静态文件夹（static）**
* CSS样式文件夹（css）
* 项目公共样式（common.css）
* 页面内私有样式（main.css）
* 图片文件夹（images）
* **文档文件（docs）**
* 项目开发规范文档
* 项目分析说明文档

**————————————————————————————————————**

**UI规范**

* **字体/字号/颜色/配色...**
* 严格参照 “美团” 网样式
* **页面内容宽度**
* 首页:width:1170px;
* 列表页:width:980px;
* 详情页:width: 980px;

**————————————————————————————————————**

**HTML**

**HTML通用样式规范**

* 统一为html5声明类型<!DOCTYPE html>
* 编码统一为<meta charset="UTF-8">
* 向浏览器表示该页面是简体中文语言<html lang="zh-cmn-Hans">
* 标签语义化（所谓标签语义化，就是指使用正确的标签表达合适的网页内容）
* 非特殊情况下样式文件外链至<head>...</head>之间；JavaScript文件外链至页面底部
* 尽可能减少div嵌套，节省资源，尽可能用最少的代价做最多的事情，避免无用标签嵌套
* 书写链接地址时, 必须避免重定向，例如：href=http://www.webjx.com/css/"http://itaolun.com/", 即须在URL地址后面加上“/”
* 非特殊情况不要写行内样式即：style=””
* 为含有描述性表单元素(input, textarea)添加label, 如<p>姓名: <input type="text" id="name" name="name" /></p>尽量写成:<p><label for="name">姓名: </label><input type="text" id="name" /></p>
* 能用css写出来的图，尽量不要用图片
* 给区块代码及重要功能加上注释, 方便后台添加功能
* 特殊符号使用escape 码代替
* 书写页面过程中, 请考虑向后扩展性
* **语法、HTML5 doctype、lang属性字符编码、IE兼容模式、引入CSS、JS、HTML5规范廉洁、属性顺序、boolean属性、JS生成标签**
* 参照通用规范
* **实用高于完美（改动）**
* 学习初始阶段，严格遵循语义化
* 任何时候都要用尽量小的复杂度和尽量少的标签来解决问题。（初始阶段尽量遵守）
* 此次项目的所有HTML标签都使用div标签，不使用HTML5的新标签

**————————————————————————————————————**

**CSS**

**CSS通用书写规范**

* 编码统一为utf-8
* 除了初始化css外，禁止直接对标签定义样式
* 多个selector共用一个样式集，则多个selector必须写成多行形式
* 属性名冒号之前不加空格，冒号之后加空格
* 如果有必要加入浏览器前缀，则按照-webkit- / -moz- / -ms- / -o- / std的顺序进行添加
* 整个网站共用的common.css放在base.html的head里面整个网站共用样式，列如：网站的header，footer，还有一些初始化的一些约定字体、字号、颜色等等
* 类名，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，单词之间使用中划线“-”
* 书写代码前, 考虑并提高样式重复使用率
* 尽量为大区块样式添加注释, 小区块适量注释
* 代码缩进与格式: 多行书写，看起来舒服，查找方便，提高代码可读性

**CSS属性书写顺序**

* 位置属性（position, top, right, z-index, display, float等）
* 大小（width, height, padding, margin）
* 文字系列（font, line-height, letter-spacing, color- text-align等）
* 背景（background, border等）
* 其他（animation, transition等）

**common.css（公共样式组件块）**

* **缩进、分号、空格、引号、属性声明顺序、颜色、属性简写、媒体查询、杂项、命名**
* 参照通用规范
* **子组件块**
* normalize.css（样式初始化组件）
* common.css（通用样式组件）
* **注释**
* 以组件为单位组织代码段；
* 制定一致的注释规范
* 组件块和子组件块以及声明之间使用一空行分隔，子组件块之间两空行分隔；
* 如果使用了多个CSS文件，将其按照组件而非页面的形式拆分，因为页面会被重组，而组件只会被移动；

|  |
| --- |
| /\* ====================== 组件块 ====================== \*/    /\* ===================== 子组件块 ===================== \*/  .selector {  padding: 15px;  margin-bottom: 15px;  }  .selector-secondary {  display: block; /\* 注释\*/  }  .selector-three {  display: span;  }  /\* =====================组件块===================== \*/    /\* ===================== 子组件块 =================== \*/  .selector {  padding: 15px;  margin-bottom: 15px;  }  /\* ===================== 子组件块 ==================== \*/  .selector-secondary {  display: block; /\* 注释\*/  }  .selector-three {  display: span;  } |

**————————————————————————————————————**

**main.css（主样式组件块）**

* **缩进、分号、空格、引号、属性声明顺序、颜色、属性简写、媒体查询、杂项、命名**
* 参照通用规范
* **子组件块**
* mg-header（导航栏模块）
* mg-banner（封面模块）
* mg-service（客户服务模块）
* mg-footer（页脚模块）
* **注释**
* 以组件为单位组织代码段；
* 制定一致的注释规范
* 组件块和子组件块以及声明之间使用一空行分隔，子组件块之间两空行分隔；
* 如果使用了多个CSS文件，将其按照组件而非页面的形式拆分，因为页面会被重组，而组件只会被移动；

|  |
| --- |
| /\* =====================组件块===================== \*/    /\* =====================⼦组件====================\*/  .selector {  padding: 15px;  margin-bottom: 15px;  }  /\* ====================⼦组件块 ====================\*/  .selector-secondary {  display: block; /\* 注释\*/  }  .selector-three {  display: span;  } |

* **命名**
* 类名添加“mgssr-”前缀，使用小写字母，以中划线分隔
* id添加“mgssr-”前缀，使用小写字母，以中划线分隔

|  |
| --- |
| /\* class\*/  .mgssr-header {  ...  }  /\* id\*/  #mgssr-header {  ...  } |

**————————————————————————————————————**

**注释规范**

* html注释：注释格式<!–这儿是注释–>
* css注释:：注释格式 /\*这儿是注释\*/

**————————————————————————————————————**

**图片规范**

* 所有页面元素类图片均放入images文件夹
* 图片格式多用于.jpg，对背景有透明需求或者其他清晰度的要求用png
* 命名全部用小写英文字母、数字、中划线-的组合，其中不得包含汉字、空格、特殊字符；尽量用易懂的词汇, 便于团队其他成员理解
* 每个小组成员的模块图片归类到images中以自己名字命名的文件夹下，公共的图片放在images根目录

**————————————————————————————————————**

**JS 通用书写规范**

* 在JS⽂件中⽣成标签让内容变得更难查找，更难编辑，性能更差。应该尽量避免这
* 减少标签数量
* 在编写HTML代码时，需要尽量避免多余的⽗节点；
* 很多时候，需要通过迭代和重构来使HTML变得更少。
* 实⽤⾼于完美
* 尽量遵循HTML标准和语义，但是不应该以浪费实⽤性作为代价；
* 任何时候都要⽤尽量⼩的复杂度和尽量少的标签来解决问题

|  |
| --- |
| <!-- Not well -->  <span class="avatar">  <img src="...">  </span>  <!-- Better --> |

**————————————————————————————————————**

**Git提交备注规范**

* 第一行是标题
* 第二行是空行
* 第三行是修改的详细信息

**————————————————————————————————————**