

魔鬼程序员-“豆瓣” 首页项目规范

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。

目录规范

```
mg_app/
├── static/           //项目公共目录
│   ├── css/         //CSS
│   └── images/
├── docs/             //文档
└── index.html        //文档首页
```

命名规范

- 项目命名

- mgdb_app

- 首页页面

- mgdb_home.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“豆瓣”首页官网样式
 - 页面内容宽度
 - min-width: 965px;
 - max-width: 1140px;
-

HTML

- 语法、HTML5 doctype、lang属性字符编码、IE兼容模式、引入CSS, JS、HTML5 规范链接、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）
 - 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；
 - 学习初始阶段，需严格遵循语义化；
 - 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）
-

common.css（公共样式组件块）

- 缩进、分号、空格、空行、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 组件（样式初始化组件）
 - common.css 组件（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块之间使用一空行分隔，子组件块之间三空行分隔；

- 如果使用了多个 CSS 文件，将其按照组件而非页面的形式分拆，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;

}

/* 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、空行、引号、属性声明顺序、颜色、属性简写、媒体查询、杂项
 - 参照通用规范
- 子组件块
 - header 组件（头部样式组件）
 - 匿名登录 组件
 - 内容 组件
 - footer 组件（底部版权声明）

• 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块之间使用一空行分隔，子组件块之间三空行分隔；

- 如果使用了多个 CSS 文件，将其按照组件而非页面的形式分拆，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;  
}  
  
/* 子组件块  
=====   
===== */  
.selector-secondary {  
  display: block; /* 注释*/  
}  
  
.selector-three {  
  display: span;  
}
```

• 命名

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

```
/* class */  
  
.mgdb-content {  
  
  ...  
  
}  
  
/* id */  
  
#mgdb-dialog {
```

...

}