

前端攻城狮规范

HTML/CSS 通用规则

文件编码

- 在 HTML中指定编码 `<meta charset="utf-8">`;
- 在外部CSS文件中指定`@charset "utf-8";`;

关注分离：结构、表现、行为分离

- 全局样式写入`project.css`文件中，当前页面的样式写入`<style>`标签中。
- 全局函数/方法，写入`project.js`文件中，当前页面的脚本写入`<script>`标签中。
- 引入文件的顺序
 - 首先在`<head>`标签里引入css文件
 - 最后在`<body>`的末尾引入js文件

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width,minimum-
scale=1.0,maximum-scale=1.0,user-scalable=no">

    <title>Title</title>

    <!-- 外部css样式文件 -->

    <link rel="stylesheet" href="css/project.css">

    <!-- 当前页面的css样式 -->

    <style>

        body{ background-color: #f2f3f8; }

    </style>

</head>
```

```
<body>

<div id="app">

  ...

</div>

<!-- 外部js文件 -->

<script src="js/project.js"></script>

<!-- 当前页面的js脚本 -->

<script>

  ...

</script>

</body>

</html>
```

引用文件

- 尽量使用相对路径

在同一网站中，推荐使用相对路径链接资源，并避免使用层级过深的文件夹，以提高文件查找效率、减小文件体积。

- 添加必要的注释
- 省略外链资源URL协议部分

```
<!-- 百度地图-->

<script src="//lbsyun.baidu.com/"></script>
```

代码格式

- 缩进2个空格

```
<ul>

  <li>Fantastic</li>

  <li>Great</li>

</ul>
```

```
.example {  
  
  color: blue;  
  
}
```

- 一律使用小写字母

```
<!-- 推荐写法-->  
  
  
  
color: #e5e5e5;  
  
  
  
<!-- 不推荐写法 -->  
  
<A HREF="/">Home</A>  
  
color: #E5E5E5;
```

添加必要的注释

- HTML内容块注释

```
<!-- 致家长的信 -->  
  
<div>...</div>  
  
  
  
<!-- 保险产品列表 -->  
  
<div>...</div>
```

- CSS需按照用途、模块分别注释

```
/* 工具样式--start */  
  
...  
  
/* 工具样式--end*/  
  
  
  
/* 布局样式--start */
```

```

...

/* 布局样式--end*/

/* 微信alert弹框--start */

/* 标题 */

.weui_dialog {...}

/* 微信alert弹框--end */

```

- JS函数需注释用途、参数、返回值

```

/* 让指定元素可以在页面上拖拽

* @param id 指定元素的id

* @return 无

*/

function initDrag ( id ) {

...

}

/* 获取保险产品信息 */

$.get('./data/product.json', function ( data ) {

...

});

```

- 引入外部css/js 文件需注释用途、来源

```

<!-- 城市选择器, jquery.weui.js 配套文件, http://jqweui.com/download -->

<script src="js/libs/city-picker.js"></script>

```

未完成条目

- 用 **TODO** 标记待办事项和活动的条目。

```
// TODO(@周伦): 完成直播XX
```

HTML规范

充分考虑结构的语义化

是段落的，你就用 `<p>`；是标题的，就用 `<h1>~<h6>`；是引用的，就用 `<blockquote>`，而不是简单粗暴的用 `<div>`、``。语义化的目的，一方面是抽去 CSS 以后，页面还是一个结构良好、可读的页面；另一方面，这也是 SEO 的最基本要求。

移动端禁止缩放

```
<meta name="viewport" content="width=device-width,minimum-scale=1.0,maximum-scale=1.0,user-scalable=no">
```

属性值使用双引号""

`<!-- 推荐写法-->`

```
<a class="maia-button">Sign in</a>
```

HTML5 规范中 `disabled`、`checked`、`selected` 等属性为 `true` 时不用设置值

```
<input type="text" disabled>
```

```
<input type="checkbox" value="1" checked>
```

```
<select>
```

```
<option value="1" selected>1</option>

</select>
```

CSS规范

命名规则

- 指定全局命名空间

以项目名或者公司名作为全局命名空间，如`qc`，所有的`class`都需要带上`qc`作为前缀

- 模块化，语义化，望文见义

如 `qc-tab`、`qc-nav`，不要使用 `red`、`left` 等表象的词命名。

常用模块名有：

`bd`(body)，`cnt`(content)，`hd`(header)，`text`(txt)，`img`(images/pic)，`title`，`item`，`cell` 等， 词义表达组件要实现的功能。

- 模块状态： {命名空间}-{模块名}-{状态描述}

如`qc-nav-hover`、`qc-tab-selected`、`qc-btn-disabled`

常用状态有： `hover`，`current`，`selected`，`disabled`，`focus`，`blur`，`checked`，`success`，`error` 等

- 子模块： {命名空间}-{模块名}-{子模块名}，

如`qc-tab-item`、`qc-cell-hd`、`qc-cell-bd`

- 模块内部的类名继承自父级。

```
<div class="qc-box">
```

```
<h2 class="qc-box-hd">About the Site</h2>
```

```
<p class="qc-box-bd">This is my blog where I talk about only the bestest
things in the whole wide world.</p>
```

```
</div>
```

上面的代码中，模块的名为 `box`，模块最外层使用 {命名空间}-{模块名} 的方式命名

Class。模块子元素以在此基础上进行命名。如果不继承父级的类名，很容易造成命名冲突。

编写规则

- 避免不必要的 CSS 选择符嵌套。

Class 已经模块化命名，从类名上已经可以清晰的分辨元素的从属，一般情况下也不会造成类名冲突，没有必要再进行选择器嵌套，保持 css 结构清晰，提高渲染效率。特殊情况可以嵌套（如提高权重、主题之间代码隔离），但应避免过多层级。

/* 推荐写法 */

```
.qc-box {}
```

```
.qc-box-hd {}
```

```
.qc-box-bd {}
```

/* 不推荐写法 */

```
.qc-box .qc-box-hd {}
```

```
.qc-box .qc-box-bd {}
```

- 与 JS 交互时，在模块 HTML 结构的最外层添加状态，而非给模块每个子元素单独添加元素。给最外层添加状态类以后，整个模块的样式都能控制，减少操作，提高性能。

比如，可以这样写：

```
<div class="qc-box qc-box-active">
```

```
  <h3 class="qc-box-title"></h3>
```

```
  <p class="qc-box-content"></p>
```

```
</div>
```

但不要这样写（效率更低）：

```
<div class="qc-box">
```

```
  <h3 class="qc-box-title qc-box-title-active"></h3>
```

```
  <p class="qc-box-content qc-box-content-active"></p>
```

</div>

JavaScript 规范

命名规则

- 常量

使用全大写字母，并用下划线分隔单词，形如 `CONST_NAME_LIKE_THIS`；

- 变量

变量采用小驼峰式命名，如 `myVar`。

- 私有变量

私有属性、变量和方法以下划线 `_` 开头，形如 `_privateMethod`。

- 类名

采用大驼峰式（[帕斯卡命名法](#)）命名，如 `MyClass`。

- 函数

函数命名需语义化，采用大驼峰式（[帕斯卡命名法](#)）命名，如 `initDrag`

前端开发资源推荐

JS框架

- [vue.js](#) 易用、灵活、高性能的渐进式JavaScript框架，[github](#) stars **38000+**。自从用了vue.js,腰不疼，腿不酸，一口气写完整个项目都不喘气了。

UI框架

- PC端:
 - [element-ui](#) 饿了么出品, 基于vue.js, 组件丰富, 文档详细, [github](#) stars 6500+。
 - [amaze-ui](#) 组件丰富, 文档详细, 可定制样式, [github](#) starts 8500+。
 - [bootstrap](#) 老牌的UI框架, github starts 10万+
- 移动端:
 - [mint-ui](#) 饿了么出品, 基于vue.js, 组件丰富, 文档详细
 - [weui](#) 微信团队出品, 唯一不足之处是组件太少
 - [jquery-weui](#) 升级版的weui, 封装一些额外的组件

前后端接口管理

- [小玄鸡](#)

自动化开发工具

- [webpack](#)
- [gulp](#)

可自动完成

- js/css 文件压缩、文件合并
- css 自动添加浏览器厂商前缀
- js 代码检测
- 图片压缩
- 代码热更新
- 手机端预览
- ...

其他插件

- [hammer.js](#) 移动端拖拽插件
- [Parsley.js](#) 表单验证插件
- [Animate.css](#) 高性能动画插件
- [swiper](#) 轮播图插件

字体图标

- [iconfont](#) 阿里巴巴字体图标库