

前端 HTML 页面开发规范

编 写	Tom			编写时间	2016 年 5 月 20 日
适用对象	适用于所有参与 HTML 前端开发工作的人员。				
修订历史					
版本	章节	类型	日期	作者	说明
V1.0	ALL	C	2016-05-20	Tom	创建第一版本。

说明：创建(C)、修改(U)、删除(D)、增加(A)

目录

一、编写目的	4
二、基本准则	4
三、开发和部署	4
五、目录和文件组织结构	4
1、根目录	4
2、源文件夹	4
六、命名规范	5
1、html 文件命名	5
2、css 文件命名	6
3、图片命名	6
七、HTML 中 DOM 结构设计规范	6
1、文档类型声明及编码	6
2、Style 引用位置	7
3、所有编码均遵循 XHTML 标准	7
4、充分利用无兼容性问题的 HTML 自身标签	7
5、语义化 HTML	7
6、尽可能减少 div 嵌套	7
7、JS 库文件引用位置	7
8、书写链接地址时，必须避免重定向	7
9、在页面中尽量避免使用 style 属性	8
10、严格区分作为内容的图片和作为背景的图片	8
11、图片必须加上 alt 属性 给重要的元素和截断的元素加上 title:	8
12、给区块代码及重要功能(比如循环)加上注释	8
13、特殊符号使用	8
14、书写页面过程中，请考虑向后扩展性	8
八、CSS 代码编写规范	8
1、统一编码	8
2、class 与 id 选择器的使用	9
3、class 命名原则	9
4、css 属性书写顺序	9
5、避免滥用自选择器	9
6、样式表中的中文处理	9
8、兼容处理	10
9、PNG 图片处理	10
10、兼容性属性的使用	10
11、减少使用影响性能的属性	10
12、添加注释	10

13、代码缩进与格式.....	10
九、JS 代码编写规范.....	10
1、统一编码.....	10
2、库引入.....	10
3、命名原则.....	10
4、代码结构.....	10
5、注重逻辑于 HTML 分离.....	11
6、不要忘记加:号.....	11
7、面向对象编程.....	11
十、图片使用规范.....	12
1、存放位置.....	12
2、格式限制.....	12
3、命名规范.....	12
4、图片大小规定.....	12
5、透明背景图片的使用.....	12
6、CSS Sprite 使用.....	12
十一、注释规范.....	12
1、html 注释.....	12
2、CSS 注释.....	13
3、JavaScript 注释.....	13
十二、开发及测试工具约定.....	14
十三、其他规范.....	14
十四、CSS Hack.....	14

一、编写目的

为提高团队协作效率，便于后台人员添加功能及前端后期优化维护，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，经讨论决定后可以更改此文档。

二、基本准则

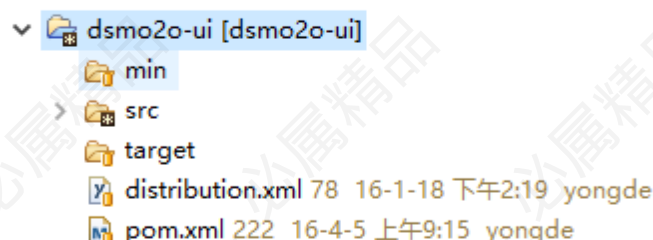
符合 WEB W3C 标准，语义化 HTML，结构表现行为分离，兼容性优良，页面性能高效，代码要求简洁明了有序，尽可能的减小服务器负载，保证最快的解析速度。

三、开发和部署

- 1、操作系统：window7 x64 及以上版本
- 2、Maven 客户端下载地址：<http://maven.apache.org/download.cgi>
- 3、开发工具(STS)下载地址：<http://spring.io/tools/sts/>
- 4、HTML 页面做为一个独立的子项目单独部署，命名为 dsmo2o-ui。
- 5、框架依赖：jQuery2.0.3、Bootstrap3.2.0

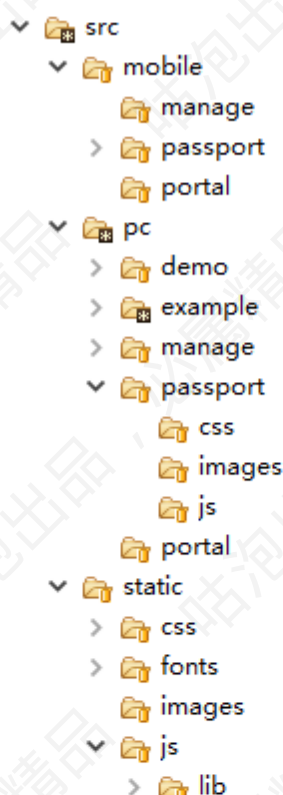
五、目录和文件组织结构

1、根目录



min/	js、css、html 等静态文件压缩后的存放目录（采用 yuicompressor 压缩）。
src/	放置所有的源代码。
pom.xml	配置所有的依赖及开发和部署环境。

2、源文件夹



src/mobile/	移动端版本的源代码。
src/mobile/manage	移动端后台管理相关页面，对应 manage 子项目的 UI。
src/mobile/passport	移动端登录相关页面，对应 passport 子项目的 UI。
src/mobile/portal	移动端门户网站的相关页面，对应 portal 子项目的 UI。
src/pc/	pc 版的源代码。
src/pc/demo	pc 端演示页面，对应 demo 子项目。
src/pc/manage	pc 端后台管理相关页面，对应 manage 子项目。
src/pc/passport	pc 端登录的相关页面，对应 passport 子项目的 UI。
src/pc/passport/css	pc 端 passport 子项目的 css 源代码。
src/pc/passport/images	pc 端 passport 子项目的图片文件。
src/pc/passport/js	pc 端 passport 子项目的 js 源代码。
src/static	公共的静态资源依赖包。
src/static/css	公共的 css 源代码。
src/static/css/fonts	公共的字体。
src/static/images	公共的图片文件。
src/static/js	公共的静态 js 源代码。
src/static/js/lib	公共的 js 框架。

六、命名规范

1、html 文件命名

命名以中文命名，依实际模块命名，每个页面命名开始为模块名的首字母大写，如果是同一模块中的小模块以“模块_子模块_页面名”来组合命名，以方便添加功能时查找对应页面，团队里的相互理解。



例如：空间里的 TAB

应该命名为：模块首字母大写_大模块_小模块_页面名.html，例如 K_空间_关注的微博.html

2、css 文件命名

1、static 目录下的 css 文件命名

static 中的 css 采用英文命名，后缀 css，文件分为三种类别：布局样式 layout、修饰样式、公共插件附加样式。例如：公共修饰样式命名为 xxx.css，公共布局样式命名为 xxx.layout.css。公共插件附加样式名为 jquery.timeline.css

2、子项目中 css 文件命名

- 1) 原则上每个页面使用一个 css 文件，命名为“页面名.css”。
- 2) 如果该 css 文件可以通用，则命名为“用途.common.css”，如分页插件：page.common.css。

2、js 文件命名

采用英文命名，后缀 .js 公共 js 使用 xxx.js 命名，每个页面的必须附带一个页面名.util.js，其他依照实际模块功能或需求命名。

例如：首页幻灯片插件

前缀.插件名.js，例如 jquery.slider.js

例如：处理首页业务逻辑的 js，例如 index.util.js

3、图片命名

采用英文命名，依实际模块或需求命名，用_隔开，图片后缀全部采用小写。

例如首页头部背景图片，则命名为“页面名_模块名_功能说明.图片格式”，如：index_header_bg.png

七、HTML 中 DOM 结构设计规范

1、文档类型声明及编码

文档类型声明统一使用如下代码：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

文档编码统一使用如下代码：


```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

目的：统一性和网站提高开发合作效率。

2、Style 引用位置

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 文件必须外链至页面底部</body>之前：

目的：实现页面的优化。

3、所有编码均遵循 XHTML 标准

标签 & 属性 & 属性命名必须由小写字母及下划线数字组成，且所有标签必须闭合，包括
 (
), <hr>(<hr/>)等；属性值必须用双引号包括：

目的：更加符合 web 标准(w3c), 也有利于 seo。

4、充分利用无兼容性问题的 HTML 自身标签

比如 span, em, strong, optgroup, label, 等等目的：减少代码量。

5、语义化 HTML

如标题根据重要性用 h* (同一页面只能有一个 h1)，段落标记用 p，列表用 ul，内联元素中不可嵌套块级元素。

目的：减少代码量，也有利于 seo

6、尽可能减少 div 嵌套

例如，有以下代码：

```
<div class="box"><div class="welcome">欢迎访问XXX，您的用户名是<div class="name">用户名</div></div></div>
```

完全可以用以下代码替代：

```
<div class="box"><p>欢迎访问XXX，您的用户名是<span>用户名</span></p></div>
```

目的：减少代码量，也有利于 seo

7、JS 库文件引用位置

文件名须包含库名称及版本号及是否为压缩版，比如 jquery-2.0.3.min.js；引入插件，文件名格式为库名称+插件名称，比如 jquery.cookie.js；

8、书写链接地址时，必须避免重定向

例如：href="http://www.haogu.com/"，即须在 URL 地址后面加上 "/"；

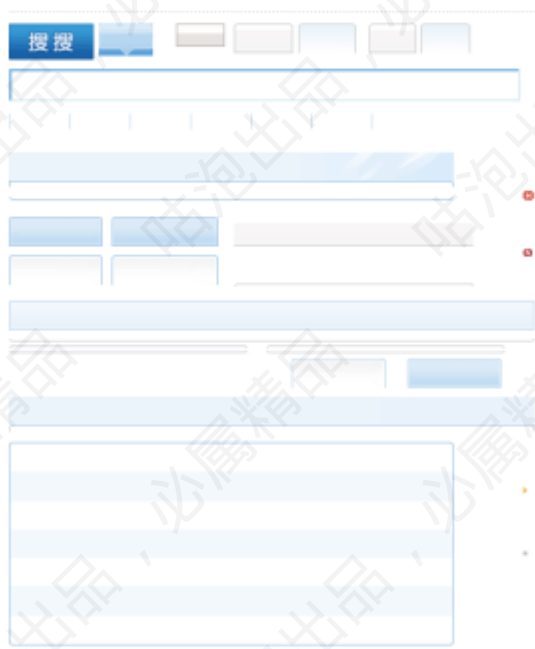
9、在页面中尽量避免使用 style 属性

即 style="..." 除非考虑网站 http 请求，网站响应速度等因素比重情况具体分析：

10、严格区分作为内容的图片和作为背景的图片

作为背景的图片采用 css sprite 技术，放在一张大图里。css sprite 技术的优点是减少了 http 请求数，但使图片面向 css 的 background-position 增加了耦合度，也增加了维护成本。如果图片有修改，不要删除已添加的图片，在空白处新增修改后的图片，减少修改风险。

下图为腾讯的 css sprite 背景图。



11、图片必须加上 alt 属性

给重要的元素和截断的元素加上 title：

目的：有利于 seo，搜索引擎的爬虫。

12、给区块代码及重要功能(比如循环)加上注释

方便后台程序员嵌套模版：

13、特殊符号使用

尽可能使用代码替代：比如 <(<) & >(>) & 空格() & »(») 等等：

14、书写页面过程中，请考虑向后扩展性

八、CSS 代码编写规范

1、统一编码

编码统一为 utf-8:都用小写，如：


```
@charset "utf-8";
```

2、class 与 id 选择器的使用

一般都使用 class，特殊除外，因 id 的优先级比 class 的高

3、class 命名原则

样式命名推荐使用英文避免使用汉语拼音，尽量使用简易的单词组合：命名方式参照《WEB 前端开发 CSS 命名参考》。命名方式采用驼峰命名法和划线命名法两种，提高可读性。

例如：dropMenu、subNavMenu、drop-menu、sub_nav_menu。驼峰命名法用来区别不同的单词。划线命名法表明从属关系。如：“.timeList”和“.time_list”分别表示时间列表和时间部分下的列表。

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。

- 1) 定位属性（比如：display, position, float, clear, visibility, table-layout 等）
- 2) 自身属性（比如：width, height, margin, padding, border 等）
- 3) 文本属性（比如：font, line-height, text-align, text-indent, vertical-align 等）
- 4) 其他属性（比如：color, background, opacity, cursor, content, list-style, quotes 等）

5、避免滥用自选择器

#test span {} 和 #test .span {} 尽量选择后者。采用继承属性还是新加 class 根据不同情况灵活运用。新加 class 便于扩展维护。

继承属性示例：

```
<ul class="list"><li>这儿是标题列表<span>2010-09-15</span></li></ul>
```

样式表：

```
ul.list li{position:relative} ul.list li span{position:absolute; right:0}
```

即可实现日期居右显示

6、样式表中的中文处理

中文字体名，请务必转码成 unicode 码，以避免编码错误时乱码。

7、TABLE 标签的使用

使用 table 标签时，请不要用 width/ height/cellspacing/cellpadding 等 table 属性直接定义表现，应尽可能的利用 table 自身私有属性分离结构与表现，如 thead, tr, th, td, tbody, tfoot, colgroup, scope, cellspaing 及 cellpadding 的 css 控制方法：

```
table{border:0;margin:0;border-collapse:collapse;} table th, table td{padding:0;}
```

一般在项目的公共 css 文件中会初始化表格样式。

8、兼容处理

尽量不使用<meta http-equiv=" X-UA-Compatible" content=" IE=7" /> 兼容 ie8,和代码*html 写法。

9、PNG 图片处理

用 png 图片做图片时，要求图片格式为 png-8 格式，若 png-8 实在影响图片质量或其中有半透明效果。

10、兼容性属性的使用

比如 text-shadow（文字阴影）、css3 的相关属性在保证目前主流浏览器使用正常的情况下可以使用来对页面进行锦上添花式的设计。

11、减少使用影响性能的属性

比如 position:absolute 或者 float;这类属性一般只用于整体布局。

12、添加注释

必须为大区块样式添加注释，小区块适量注释。

13、代码缩进与格式

单行书写，统一使用 tab 进行缩进。

目的：css 书写规范主要提高效率方便面想关人员的修改，提高网站的扩展性（根据网站情况不一样，要灵活应用）。

九、JS 代码编写规范

1、统一编码

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；要检查网上 down 下来的代码造成的代码污染（沉冗冲突，代码注入...）等，有的可以 firefox 的一些扩展插件分析。

2、库引入

原则上仅引入 jQuery 库，若需引入第三方库，须与团队其他人员讨论决定。

3、命名原则

命名语义化，尽可能利用英文单词或其缩写。

4、代码结构

代码结构明了，加适量注释，提高函数重用率。

5、注重逻辑于 HTML 分离

注重与 html 分离，减小 reflow，注重性能。

6、不要忘记加分号

每一句代码结束时，记得加上分号。虽然很多浏览器解析 js 时，不加分号不会出现语法错误，但项目发布时会将源代码进行压缩，即去除了所有的空格。因此，访问发布后的页面时，就莫名的报语法错误了。

定义方法同样也要加分号。例如：

```
var get = function(){  
    return;  
};
```

7、面向对象编程

尽量使用面向对象的编程思想编写 javascript 代码。如果使用面向对象的思想定义一个资源管理器的功能，代码如下：

```
/**  
 * 文件管理器方法定义  
 */  
var explorer = (function() {  
    //在这里定义所有的私有方法  
    var self = this;  
    this.logout = function(){//退出登录定义  
    };  
    this.changeDir = function (path) { //切换文件夹  
    },  
    this.rename = function(oldpath, newpath){//重命名  
    },  
    this.remove = function(path){//删除  
    },  
    this.download = function(path){//下载  
    },  
    this.createFolder = function(name){ //创建文件夹  
    },  
    this.loadFiles = function () { //加载文件列表  
    };  
    //如果要开发访问方法，在这里加上返回输出  
    return {  
        logout : logout,  
        loadFiles : loadFiles,
```

```

        changeDir : changeDir,
        createFolder : createFolder
    });
})();

//使用赋值
window.explorer = explorer;

//调用
$(document).ready(function(){
    explorer.loadFiles();
});

```

目的：避免带来的问题及冲突，网站的优化。

十、图片使用规范

1、存放位置

所有页面元素类图片均放入 `images` 文件夹，

2、格式限制

图片格式仅限于 `gif` || `png` || `jpg`;

3、命名规范

命名全部用小写英文字母 || 数字 || `_` 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；另，命名分头尾两部分，用下划线隔开，比如 `ad_left01.gif` || `btn_submit.gif`;

4、图片大小规定

在保证视觉效果的情况下选择最小的图片格式与图片质量，以减少加载时间；

5、透明背景图片的使用

尽量避免使用半透明的 `png` 图片（若使用，请参考 `css` 规范相关说明）；

6、CSS Sprite 使用

运用 `css sprite` 技术集中小的背景图或图标，减小页面 `http` 请求，但注意，请务必在对应的 `sprite psd` 源图中划参考线，并保存至 `img` 目录下。

十一、注释规范

1、html 注释

注释格式 `<!--这儿是注释-->`，“-”只能在注释的始末位置，不可置入注释文字区域。如：

```
<!-- 这儿是注释-->
```

2、CSS 注释

注释格式 `/*这儿是注释*/`，如：

```
/*这儿是注释*/
```

3、JavaScript 注释

单行注释使用

```
//这儿是单行注释
```

多行注释使用

```
/* 这儿有多行注释 */
```

全局变量注释

```
/* 这儿是变量注释*/
```

方法注释

```
/**  
 * 方法功能说明  
 * @param 参数名 参数说明  
 * @author 编写者  
 * @return 返回值说明  
 */
```

代码维护注释

```
/* 修改原因 modify by 作者 at 日期 Begin */
```

```
...
```

```
/* 修改原因 modify by 作者 at 日期 End */
```

十二、开发及测试工具约定

- 1、编码必须格式化，比如缩进；
- 2、测试工具：前期开发仅测试 FireFox & IE6 & IE7 & IE8，后期优化时加入 Opera & Chrome。
- 3、建议测试顺序：FireFox -> IE7 -> IE8 -> IE6 -> Opera -> Chrome，建议安装 firebug 及 IE Tab Plus 插件。安装调试利器 IETester。

十三、其他规范

- 1、开发过程中严格按分工完成页面，以提高 css、js 复用率，避免重复开发。秉着从整体出发，换位思考的原则。
- 2、减小沉冗代码，书写所有人都可以看懂的代码。简洁易懂是一种美德。为用户着想，为服务器着想。

十四、CSS Hack

即便是完美的 CSS 也未必能在目前众多的终端浏览器中呈现一致的效果，所以，CSS Hack 在很多情况下都是必要的，建议先以对 CSS 标准支持得比较好的浏览器（比如“Firefox”或者“Chrome”）为主编辑 CSS，最后再处理 IE 的兼容性——单独为 IE 建立一个 CSS 文件（比如 for-ie.css, fuck-ie.css, ie-hack.css 等），最后在 HTML 文件中，通过 IE 的条件注释按需引用。