# 母婴之家前端开发规范

## 规范目的

1. 为提高团队协作效率，便于后台人员添加功能及前端优化维护，输出高质量的文档，特制订此规范文档。
2. 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。
3. 本文档如有不对或不合适的地方请及时提出， 经讨论决定后方可更改。

## 基本准则

1. 符合web标准、 语义化html、 结构表现行为分离、兼容性优良。
2. 减小冗余代码， 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。
3. 代码的松耦合，高度模块化，将页面内的元素视为一个个模块，相互独立，尽量避免耦合过高的代码。html、css、js 三个层面都要考虑模块化。
4. 注意代码的弹性，提高css复用率， 避免重复开发。在性能和弹性的选择上，一般情况下以弹性为优先考虑条件，在保证弹性的基础上，适当优化性能。
5. 良好的注释。
6. 严格按照规范编写代码。

## 文件规范

**【文件命名规则】**

1. 文件名称统一用小写的英文字母、数字和下划线的组合，不得包含汉字、空格和特殊字符。
2. 命名原则的指导思想：

一是使得你自己和工作组的每一个成员能够方便的理解每一个文件的意义；二是当在文件夹中使用“按名称排例”的命令时，同一种大类的文件能够排列在一起，以便查找、修改、替换、计算负载量等等操作。

**【文件存放路径】**

1. **css：**存放样式文件
2. **images:** 存放页面样式背景图片。公用的图片直接放在images下， 私用的图片要在images建立相应的文件夹存放，例如usercenter。
3. **pic:**存放页面内容图片
4. **js:**存放脚本文件

## css规范

**【编码】**

1. **母婴之家页面使用utf-8编码，请注意保持CSS文件编码与页面编码一致。**
2. **为了防止文件合并及编码转换时造成问题，建议将样式中文字体名字改成对应的英文名字，如：黑体(SimHei)、宋体(SimSun)、微软雅黑 (Microsoft Yahei），几个单词中间有空格组成的必须加引号。**

**【命名规则】**

1. **采用通俗易懂的英文单词并按内容/功能命名，**避免使用中文拼音。严禁用缩写， 命名要语义化、简明化，尽量使用英文单词全拼组合。**严禁出现如left、right等方向名词的class/id，严禁出现如xxx1、xxx2等的数字class/id。**
2. id是唯一的并是父级的， class是可以重复的并是子级的。id仅使用在大的模块和为js预留钩子而不用于任何css。除布局、唯一独立模块外建议少用id，必须保证id唯一性。id命名采用下划线，如 xxx\_yyy，或者驼峰命名方式，如xxxYyy。为JavaScript预留钩子的id命名， 以 js\_ 起始， 比如: js\_hide， js\_shoppingCart。
3. class用在重复使用率高及子级中，样式只能写在class上，不可写在id上。
4. **class命名一律采用小写中划线方式，如 xxx-yyy，禁止出现大写字母，尽可能提高代码模块的复用。复用模块、独立模块可按xxx-mod命名（-mod可不写），mod下面再取xxx-hd（头部）、xxx-bd（内容）、xxx-ft（底部）命名。**
5. **常用命名（多记多查英文单词）：page、wrap、layout、header(head)、footer(foot、ft)、content(cont)、menu、nav、main、submain、sidebar(side)、logo、banner、title(tit)、popo(pop)、icon、note、btn、txt、iblock、window(win)、tips等。**

**【文件划分】**

1. reset.css是样式重置文件，每个页面都要引用，此文件不可修改。base.css是网站所有页面共用的样式，每个页面都要引用，此文件不可随意修改，如要修改，必须经过各位前端人员协商，确认样式必须放在base.css才可修改。
2. 根据页面模块需求确定其他css文件名，比如：栏目页category.css，表单样式form.css，购买流程shopping.css。

**【代码顺序】**

1. **css文档的顺序**
   1. **共用的代码样式放在文档最前面。**
   2. **css结构的顺序依照html的结构。**
2. **css选择器中的属性顺序：显示属性->自身属性->文本属性**
   1. **显示属性(display / position / float / clear / cursor …)**
   2. **自身属性(margin / padding / border /width / height/ background...)**
   3. **文本属性(color / font / vertical-align / text-decoration / text-align /…)**
   4. **内容管理、层级(overflow / z-index / zoom...)**

**【其他】**

1. **禁止使用****expression表达式。**
2. **禁止滥用！important（使用前必须讨论确认不会造成其他bug或者维护问题）。**
3. **尽量少的使用滤镜，尽量减少hack数，能不hack的坚决不hack。**
4. **层级(z-index)必须清晰明确，页面弹窗、气泡为最高级（最高级为999，如showWin、pop等），不同弹窗气泡之间可在三位数之间调整；普通区块为10-90内10的倍数；区块展开、弹出为当前父层级上个位增加，禁止层级间盲目攀比，**

**如以下html结构：**

<ul class="post-bd">

<li class="post-to">

<div class="post-tag"></div>

<div class="post-add"></div>

</li>

<li class="post-editor"></li>

<li class="post-app"></li>

</ul>

**错误**

.post-to {z-index:999;}

.post-tag {z-index:99;}

.post-add {z-index:9;}

.post-editor {z-index:99;}

.post-app {z-index:9;}

正确

.post-to {z-index:90;}

.post-tag {z-index:99;}

.post-add {z-index:98;}

.post-editor {z-index:80;}

.post-app {z-index:70;}

1. 保持代码的可读性与可维护性，建议单行书写, 可根据自身习惯, 后期由专人统一优化。如果是多行编写，代码必须缩进，CSS结构同html结构 (但不建议缩进太多，能大概看出结构即可)，选择器与{}之间必须有一个空格分隔，{}里不要出现多余的空格、换行。
2. 在保证正确性的前提下，去除不必要的层级选择，例如：

错误

div.post-to {}

.post-app dl dt {}

.post-app ul li.post-xxx {}

正确

.post-to {}

.post-app dt {}

.post-app .post-xxx {}

1. 字体粗细采用具体数值，粗体bold写为700，正常normal写为400
2. 能缩写的尽量缩写，去掉0后面的单位（禁止出现如0px等多余的单位）、颜色值不要出现red/blue等颜色名词（一律采用#加16进制来表示，CSS3的rgb除外），颜色值小写、6个字母的#颜色值缩写为3个字母。

错误

.post-add {margin:0px 0px 1px 1px;color:rgb(255，255，255);background: white;}

正确

.post-add {margin:0 0 1px 1px;color:#fff;background:#fff;}

错误

.post-tag{margin:2px;margin-right:5px;font-weight: bold;font-size:14px;line-height:24px;font-family:”黑体”;}

正确

.post-tag { margin:2px 5px 2px; font: 700 14px/24px SimHei; }

## html规范

**【页面头部】**

1. 每个页面必须有文档类型声明。

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

<html xmlns="http://www.w3.org/1999/xhtml">

1. head 必须包含的信息

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

<meta name="description" content="xxxxxxxxxxxxxxxxxxxxxxxxxx" />

<meta name="keywords" content="xxxx，xxxx，xxx，xxxxx，xxxx，" />

<title>xxxxxxxxxxxxxxxxxx</title>

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

**【引入】**

1. 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。
2. 引入JS库文件， 文件名须包含库名称和版本号及是否为压缩版，比如jquery.1.4.1.min.js； 引入插件， 文件名格式为库名称+插件名称，比如jquery.cookie.js。
3. 引入样式文件或JavaScript文件时，须略去默认类型声明， 写法如下:

<link rel="stylesheet" href="..." />  
<style>...</style>  
<script src="..."></script>

1. 公用部位可以引入进来，比如头部和尾部。

**【细节】**

1. 所有编码均遵循xhtml标准， 标签、属性、属性命名只能使用小写字母、下划线、中划线、数字， 大小写夹杂也是不被认可的，通常dreamweaver自动生成的属性名字onMouseOver也必须修改成onmouseover。
2. 充分利用无兼容性问题的html自身标签，比如span、 em、 strong、 label等等；需要为html元素添加自定义属性的时候，首先要考虑有没有默认的合适标签去设置。
3. 所有标签必须闭合，包括 br (<br />)，hr(<hr />)等。
4. **给所有属性赋一个值。**例如：

input type="checkbox" name="shirt" value="medium" checked />  
必须修改为：  
<input type="checkbox" name="shirt" value="medium" checked="checked" />

1. 属性值必须用双引号**""**括起来。
2. **所有<和&特殊符号用编码表示**  
   任何小于号 <，不是标签的一部分，都必须被编码为&lt ;  
   任何大于号 >，不是标签的一部分，都必须被编码为&gt ;  
   任何与号 & ，不是实体的一部分的，都必须被编码为&amp;
3. 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。
4. 尽可能减少div嵌套。例如：

<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>

1. 在页面中尽量避免使用style属性，即style="…"。
2. 必须为含有描述性表单元素(input， textarea)添加label。 例如：

<p>姓名: <input type="text" id="name" name="name" /></p>

须写成:

<p><label for="name">姓名: </label><input type="text" id="name" /></p>

1. 除了模拟页面的常规状态，还要将极端状态模拟出来，比如内容为空时及内容过多时的呈现。
2. 在页面中尽量完成每一步交互效果，包括即时响应的。
3. 提交给程序员的demo必须是连贯的，交互效果齐全，而且经过 IE6、 IE7、 IE8、IE9、 firefox、chrome、safari 等浏览器的兼容性测试。
4. 给区块代码及重要功能(比如循环)加上注释，方便后台添加功能。
5. 书写页面过程中，要考虑向后扩展性。

**【链接】**

1. 网站中的链接路径全部采用相对路径，一般链接到某一目录下的缺省文件的链接路径不必写全名，如不必这样：<a href="aboutus/index.htm">，而应该这样：<a href="aboutus/">，所有内页指向首页的链接写成<a href="/">。

2. 在浏览器里，当我们点击空链接时，它会自动将当前页面重置到首端，从而影响用户正常的阅读内容，我们用代码“javascript:void(null)”代替原来的“#”标记。

**【字体】**

1. 在设定字体样式时对于文字字号样式和行间距应必须使用CSS样式表，禁止在页面中出现<font> 标记。
2. 在网页中中文应首选使用宋体。英文和数字首选使用verdana和arial两种字体。一般使用中文宋体的9pt和11pt 或12px和14.7px，这是经过优化的字号，黑体字或者宋体字加粗时，一般选用11pt和14.7px的字号比较合适。
3. 为了最大程度的发挥浏览器自动排版的功能，在一段完整的文字中请尽量不要使用<br> 来人工干预分段。
4. 不同语种的文字之间应该有一个半角空格，但避头的符号之前和避尾的符号之后除外，汉字之间的标点要用全角标点，英文字母和数字周围的括号应该使用半角括号。
5. 请不要在网页中连续出现多于一个的 也尽量少使用全角空格（英文字符集下，全角空格会变成乱码），空白应该尽量使用 text-indent，padding，margin，&nbsp;来实现。
6. 行距建议用系数来定义，常用的两个行距的值是line-height:1.2/1.5。
7. 排版中我们经常会遇到需要进行首行缩进的处理，不要使用 或者全角空格来达到效果，规范的做法是在样式表中定义 p { text-indent: 2em; }，然后给每一段加上 <p> 标记。

**【alt和title】**

1. alt 用来给图片来提示的。title用来给链接文字或普通文字提示的。 重要图片必须加上alt属性； 给重要的元素和截断的元素加上title。

用法如下：

<p title="给链接文字提示">文字</p>

<a href="#" title="给链接文字提示">文字</a>

<img src="图片.gif" alt="给图片提示">

**【图片】**

1. 从数据库里取出来的图片属于内容，放在页面里；凡是不属于内容的图片都用背景，写在css里。
2. 链接和按钮要做上鼠标 hover 时的一个变化效果，如果 hover时是换一张背景图片，请把这两张图片整合在一张图片中，以防止在 hover 时，页面还在download 变化的那张图片，这样会出现那个按钮无图的间隔。

## 图片规范

1. 命名规则见文件规范中的文件命名规则。
2. 命名分头尾两部分，用下划线隔开，头部分表示此图片的大类性质，例如：广告、标志、菜单、按钮等等。
3. 放置在页面的广告、装饰图案等长方形的图片命名头部分以banner开始；标志性的图片命名头部分以logo开始；带有链接或提交功能的图片命名头部分以button开始；背景图片命名头部分以bg开始；导航图片命名头部分以nav开始；装饰用的图片命名头部分以icon开始；表示标题的图片命名头部分以title开始。
4. 命名范例：ad\_left01.gif、btn\_submit.gif、banner\_sohu.gif 、nav\_aboutus.gif 、title\_category.gif 、logo\_muyingzhijia.gif 、icon\_news.gif；鼠标感应效果图片命名规范为"图片名\_on/off"，例如：nav\_on.gif 、nav\_off.gif。
5. 图片格式仅限于gif、png、jpg。  
   在保证视觉效果的情况下选择最小的图片格式和图片质量， 以减少加载时间。
6. 运用css sprite技术集中小的背景图或图标，减小页面http请求，但请务必在对应的sprite psd源图中划参考线， 并保存至images目录下。

## 注释规范

1. html 注释格式:<!--这里是注释-->。可以在模块的开始和结束处各加注释，例如:

<!--header start-->

这里是header

<!--header end-->

也可以只在模块结束处加注释，例如:  
这里是header

<!--header end-->

1. css注释格式: /\*\*这里是注释\*\*/ 。
2. JavaScript注释，单行注释使用: //这里是单行注释，多行注释使用: /\*这里是多行注释\*/。单行注释不要在代码后的同一行内加注释。 例如：

正确注释：

//姓名

var name = “abc”;

错误注释：

var name =”abc”; //姓名

1. 需要在css头部加上注释说明：

/\*\*

\*文件用途说明

\*作者姓名

\*制作日期

\*\*/

1. 维护人员尽量根据注释说明，找到代码的原作者，让原作者进行维护。原作者进行维护可以无需添加额外说明，直接进行修改。如果因为特殊原因无法让原作者进行维护，需添加额外说明进行注释。说明文字为：

/\*change by xxx) 原代码如下：

<{源代码}>.新代码如下：\*/

新代码：

如：name = “abc”;这段代码，var 要将 name 由“abc”变成“123”，原作者可直接改 var name=”123”; 非原作者修改，需改成：

/\*(change by fangcao) 原代码如下：<{ var name = “abc”; }>新代码如下：\*/

var name =”123”;

1. 修改时添加的注释，在项目通过测试之后，上线之前，可以优化掉。

## 分工安排

1. 分工原则为公共组件（包括reset.css，base.css 和my.js）由专人维护，各栏目其他人负责，每个栏目正常情况下由一人负责，要详细写明注释，如果多人合作，维护的人员注意添加注释信息，具体注释方法，详见注释规则。
2. 视觉设计师完成设计图后，先和交互设计师或产品专员沟通，确定设计可行，再将设计图给公共组件维护者，看设计图是否需要提取公共组件，最后提交给相应栏目的前端工程师。如果有公共组件要提取，公共组件维护者需对栏目前端工程师说明。
3. 如果确定没有公共组件需提取，交互设计师或产品专员直接和各栏目的前端工程师交流，对照着设计图进行说明，前端工程师完成需求。
4. 前端工程师在制作页面的时候，需先去 base.css文件中查询是否已经存在设计图中的组件，如果有，直接调用；没有，则在 app.css 和 app.js 中添加相应的代码。
5. 前端工程师在制作过程中，发现有高度重用的模块，却未被加入到公共组件中，需向公共组件维护人进行说明，然后公共组件维护人决定是否添加该组件。如果确定添加，则向所有前端工程师告知添加了新组件，让前端工程师检查之前是否添加了类似组件，统一更新成新组件的用法，删除之前自定义的 css 和 js。虽然麻烦，但始终把可维护性放在首位。
6. 公共组件维护者的公共组件说明文档，需图片和说明文字配套，方便阅读。

修改记录

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 修改人 | 修改内容 | 当前状态 |
| 2012.8.20 | Clock | 拟定css规范 | 草案 |
| 2012.8.22 | Clock | 拟定html规范 | 草案 |
| 2012.8.23 | Clock | 拟定开发总则 | 草案 |
| 2012.8.24 | Jinwyp | 修改css规范 |  |
| 2012.8.24 | Clock | 将css规范、html规范、开发总则整合成该文档 | 草案 |
| 2012.8.27 | Clock | 修改该文档 | 草案 |
| 2012.8.28 | Clock | 修改该文档css部分 | 草案 |