



伽然前端开发规范

第一版

(2013年6月)

伽然信息科技有限公司（上海）有限公司

目录

第一版	1
【 命名规范 】	- 1 -
命名基本规则	- 1 -
HTML 文件命名规范	- 1 -
■ 通用页面命名:	- 1 -
■ 电商类页面命名:	- 2 -
CSS 文件命名规范	- 2 -
CSS 选择器命名规范	- 3 -
JS 文件命名规范	- 5 -
JS 变量、函数命名规范	- 5 -
图片命名规范	- 5 -
【 注释规范 】	- 7 -
HTML 文件注释	- 7 -
CSS 文件注释	- 9 -
JS 文件注释	- 9 -
【 HTML 规范 】	- 12 -
【 CSS 规范 】	- 14 -
【 JS 规范 】	- 15 -
【 图片规范 】	- 17 -
【 SEO 针对性规范 】	- 18 -
一、 META 标签说明	- 18 -
二、 标题规范	- 18 -
三、 元标签规范	- 18 -
四、 图片规范	- 19 -
五、 关键字规范	- 19 -
六、 链接规范	- 19 -
七、 URL 规范	- 20 -
八、 代码优化规范	- 21 -



【 命名规范 】

命名基本规则

1. 命名要有意义，不要使用没有意义的命名；
2. 命名清晰，不怕命名长，短命名容易冲突，长命名可以保证不会产生冲突；
3. 公共组件因为高度重用，命名从简，不加前缀；
4. 名称全部用**英文字母、数字**的组合，**驼峰式命名规则**，第一个单词全部小写，其后的单词首字母大写，其中不得包含汉字、空格和特殊字符；
5. 尽量用一些大家都能看懂，朗朗上口的词汇；

HTML 文件命名规范

■ 通用页面命名：

首页：index.html

列表页：***List.html

详细页：***Detail.html

公司介绍：profile.html

联系我们：contact.html

关于我们：about.html

人才招聘：job.html

发展历程：history.html

新闻列表：newsList.html

新闻详情：newsDetail.html

产品列表：productList.html

产品详情：productDetail.html

客户留言：leaveMsg.html

常见问题：faq.html

下载：download.html



企业文化: culture.html
公司环境: environment.html
视频列表: videoList.html
视频详情: videoDetail.html
网站地图: siteMap.html
隐私政策: privacy.html

■ 电商类页面命名:

购物车: cart.html
审核订单: orderCheck.html
提交订单: orderSubmit.html
支付订单: orderPay.html
找回密码: findPwd.html
发送邮件: sendEmail.html
登陆: login.aspx
注册: register.html
用户中心: uesrCenter.html
订单列表: orderList.html
订单详情: orderDetail.html
收货地址: address.html
我的收藏: favrite.html
我的评论: comment.html
我关注的: focus.html
退换货: allRepairs.html
浏览历史: historyList.html
我的积分: integral.html

其他页面使用通俗易懂的英文单词命名，且遵循驼峰式命名规则。

CSS 文件命名规范



公用样式: common.css (base.css\reset.css)

产品模块: product.css

会员中心模块: usercenter.css

登陆注册模块: login.css

关于我们: about.html

自适应: responsive.css

补丁样式: append.css

其他样式文件使用和页面名称相同的命名。

CSS 选择器命名规范

1. 减少 CSS 选择器层级，确保 CSS 优先级权重足够低，方便扩展时覆盖操作；
2. 层级一般不要超过四个；
3. 选择器首选类选择器，预留权重更高的 id 选择器方便后期扩展时样式层叠覆盖；
4. 使用“类别+功能”的方式命名；

常规命名：

顶部边栏: topBar;

顶部: top(head);

底部: bottom(foot);

底部边栏: bottomBar;

页面主体: main;

容器: wrap;

栏目: column;

摘要: summary;

登录条: loginBar;

左侧栏: leftBar;

右侧栏: rightBar;

侧栏: sideBar;



标志: logo;
导航: nav;
子导航: subNav;
左侧导航: leftNav;
右侧导航: rightNav;
侧导航: sideNav;
菜单: menu;
子菜单: subMenu;
搜索: search;
当前位置: position;
广告, 标语: banner;
选项卡: tab;
列表: list;
提示信息: msg;
小技巧: tips;
标题: title(t);
加入: join;
热点: hot;
指南: guide;
服务: service;
下载: download;
新闻: news;
登陆: login;
注册: register;
按钮: btn;
状态: status;
投票: vote;
合作伙伴: partner;
友情链接: fLink;



版权信息: copyRight;

当前: current;

购物车: cart;

JS 文件命名规范

公用 js: common.js;

页面 js: *.js(*为栏目名或首字母缩写);

JS 变量、函数命名规范

1. 使用驼峰命名法;
2. 常量及全局变量采用全部大写的形式;
3. 变量的命名使用驼峰命名法;
例如: strParsor.maxLength;
4. 类的命名使用骆驼命名法,其中首字母大写:
例如: AccountInfo: EventHandler
5. 方法的命名必须为动词或者是动词短语:
例如: obj.getSomeValue()
6. 私有类的成员变量命名前面加下划线 (_):
例如:

```
var MyClass = function(){  
    var _buffer;  
    this.doSomething = function(){  
    };  
}
```

又如:
this._somePrivateVariable = statement;
7. 前缀为 "is"或 "has"的变量应该为布尔值;
8. 术语 "init" 作为变量名应为初始化方法;
9. 迭代临时变量建议使用 "i", "j", "k" (依次类推) 等名称;
10. 异常处理类建议在变量名称后加上 "Exception" 或者 "Error";

图片命名规范



公用背景图片: common.gif

广告图片: ad_*

标志性图片: logo_*

导航图片: nav_*

菜单图片: menu_*

装饰用的图片: pic_*

不带链接表示标题的图片取名: tit_*

小图标: ico_*

背景: bg_*

*用来表示图片的具体含义, 例如:

menu_aboutus.gif、menu_job.gif、pic_people.jpg、pic_hill.jpg



【 注释规范 】

所有文件的维护注释方法：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。下面为 CSS 文件的维护注释示例：

在需要修改的地方添加说明文字为：

```
/* 修改 by 张三
 * 原代码如下：
    CSS 原代码
 * 新代码如下：
 */
CSS 新代码
```

修改时添加的注释，在项目通过测试之后，上线前可以优化掉。

HTML 文件注释

我们经常要在一些代码旁做一些 HTML 注释，这样做的好处很多，比如：方便查找，方便对比，方便项目组里的其它程序员了解你的代码，而且方便以后对自己代码的理解与修改等等。

1. 常规注释：

```
<!--注释掉的内容 -->
```

被注释掉的内容在浏览器中是无法显示的；如果页面出现乱码无法解决，可以使用逐步注释的方法进行调试，找到问题。

2. 将注释作为文档模块的说明性提示，说明该木模块代码的功能或者起始位置：

```
<!--something 开始 -->
```

```
Your HTML Code
```

```
<!--something 结束 -->
```

或者：



<!-- 模块名: 功能说明 开始 -->

Your Module

<!-- 模块名 结束 -->

3. IE 条件注释: 上述注释方法所有浏览器都可识别, 但下面的注释方法只有 IE 浏览器才能识别;

<!--[if IE]>Code<![end if]--> // IE 条件注释

<!--[if ! IE]><![end if]--> //非 IE 条件注释

特定版本 IE 条件注释:

<!--[if IE 6]> <![end if]--> // 仅 IE6 可识别

<!--[if lte IE 6]> <![end if]--> //IE6 及其以下版本可识别

<!--[if lt IE 6]> <![end if]--> //IE6 以下版本可识别

<!--[if gte IE 6]> <![end if]--> //IE6 及其以上版本可识别

<!--[if gt IE 6]> <![end if]--> //IE6 以上版本可识别

说明:

lte: 小于等于

lt: 小于

gte: 大于等于

gt: 大于

4. 条件注释和 CSS HACKS 的区别;

- Hacks 是基于浏览器的 bug, 而这些 bug 最终可能会被修复。
- 条件注释是基于 IE 特定代码, 这种识别机制任何时候都不会被移除。
- 每一个浏览器都能看见你的 hacks, 或许下一个版本或者一个新的浏览器会在你的 hacks 代码上出错。
- 只有 IE 才能看到条件注释, 通过额外的“IE 文件”来影响页面, 其他浏览器根本就不会下载它。
- Hacks 不能确保对哪些浏览器生效而对哪些不, 用的 hacks 越多, 代码越混乱。



- 条件注释利用版本匹配使得作者可以容易地对特定版本写代码。

CSS 文件注释

就像上述 HTML 注释中描述的一样，在 CSS 文档中注释也起到很重要的作用，可以帮助我们记起 CSS 的含义，加载在 HTML 文档的位置等。

1. CSS 文件使用 `/* CSS CODE */`（斜杠星）的注释方法；
2. 在文件头部加注释说明：

```
/**  
 * CSS 文档用途  
 * 作者姓名、联系方式（电话、QQ、邮箱等）  
 * 制作日期  
 **/
```

3. 对每一个功能模块使用注释，方便模块移植：

```
/* 模块名/  
CSS DODE
```

4. 新增加的代码：

```
/* 新增 模块名 by 添加人 时间 */
```

例如：

```
/* 新增 Login 样式 by 张三 2013-6-30 wangzili@besture.com */
```

JS 文件注释

给 JS 做注释同样有很多好处，比如：方便查找，方便比对，方便项目组里的其它程序员了解你的代码，而且方便以后对自己代码的理解与修改等等。

注：

- 过多的 JavaScript 注释会降低 JavaScript 的执行速度与加载速度，因此应在发布网站时去掉 JavaScript 注释。
- 注释块 (`/* ... */`) 中不能有 (`/*或*/`)，JavaScript 正则表达式中可能产生这种



代码), 这样会产生语法错误。

1. JS 中有两种注释方法:

a) 单行注释 (双斜杠):

```
// JS CODE
```

注释单独一行, 不要在代码后的同一行内加注释

错误示例:

```
var a='张三' // 声明变量 a
```

正确示例:

```
// 声明变量 a
```

```
var a='张三'
```

b) 多行注释 (与 CSS 注释方法一样):

```
/*
```

```
JS CODE
```

```
*/
```

2. 在文件头部加注释说明:

```
/**
```

```
* JS 文档用途
```

```
* 作者姓名、联系方式 (电话、QQ、邮箱等)
```

```
* 制作日期
```

```
**/
```

3. 对每一个功能模块加注释:

```
// = = = = =
```

```
//模块功能描述
```

```
//= = = = =
```

或者

```
/**
```

```
* 模块功能说明
```

```
* @参数 1: 参数说明 ( 类型、用途 )
```

```
* @参数 2: 参数说明 ( 类型、用途 )
```



* @返回值类型

**/

4. 新增加功能模块:

a) 单行:

// 新增 模块名 by 添加人 时间

例如:

// 新增 Login 样式 by 张三 2013-6-30 wangzili@besture.com

b) 多行

/**

* 新增 模块名 by 添加人 时间

* 参数 1: 参数说明 (类型、用途)

* 参数 2: 参数说明 (类型、用途)

* 模块功能说明

**/



【 HTML 规范 】

1. DTD 声明统一使用 XHTML 1.0 Transitional DTD ；
2. head 元素中必须包含 title ， keywords, descriptin ；
3. 统一使用 UTF-8 编码，设置 charset=utf8 ；
4. 代码编写采用 Tab 键缩进，保证代码层次结构清晰，禁止使用连续空格缩进代码 ；
5. 当 link 元素用于引用 CSS 文档时，必须指明 media 属性 ；
6. 样式分离：
 5. 非特殊情况下必须把 CSS 外部链接放到页面的 head 标签之间 ；
 6. 行为分离：
 7. 非特殊情况下必须把 JS 代码或外部链接放到页面的底部（</ body>标签之前） ；
 8. 尽量减少页面的 HTTP 请求：
 9. 合并压缩的多个 JS 到一个文件中 ；
 10. 合并压缩的多个 CSS 到一个文件中 ；
 11. 合并多个背景图片，使用 css sprite 技术 ；
 12. HTML 应在保证弹性的基础上尽量减少嵌套层数 ；
 13. 使用正确的注释方法给区块代码加上注释 ；
 14. 所有 HTML 代码必须小写 ；
 11. HTML 标签的非样式属性值必须用半角英文双引号括起来，并且一定要有值，样式属性尽可能的写在 css 文件中 ；
 12. 双标记标签都要有开始和结束标签，单标记标签的后面一定要加 “/” ；
 13. 同一页面不要重复使用同一 id ；
 14. 所有标签必须进行合理的嵌套，体现文档的结构，有利于搜索引擎的抓取 ；
 15. 非特殊情况不使用表格，必须使用的情况要提前规划好，避免使用合并单元格的形式；
 16. flash 的嵌入使用<object>标签 ；
 17. 在页面中给单选框、复选框加上<lable></lable>标签 ；
 18. 在属性值和元素内容出现的地方，如果有特殊字符 “， &， <， >”，那么应当分别使用 "， &， <， > ；



19. table 必须指定 summary 属性 ；
20. table 必须显示包含 tbody 元素 ；
21. a 标签关键的链接处应当指定 accesskey 属性 ；
22. form 中严禁再嵌套 form 元素 ；
23. 一个 form 有且只有一个 type="submit" 的 input 按钮 ；
24. 禁止 submit 上绑定 onclick 等事件使用 submit 方法提交表单 ；
25. 与控件对应的文本必须使用 label 元素标记，且使用 for 属性指向该控件 id ；
26. 尽量减少 iframe 元素的使用 ；



【 CSS 规范 】

1. 文件编码为 UTF-8;
2. CSS RESET 用 YUI 的 CSS RESET;
3. CSS 用一行的写法，避免行数太长，不利查找;
4. 代码模块化;
5. 为方便组件模块化和提高弹性，正常情况下，为避免外边界冲突，组件不设置外边界，外边界用组合 CSS 的方式实现;
6. 为避免组件的上下外边距重合问题和 IE 的 haslayout 引发的 Bug, 各模块除特殊需求，一律采用 `marginTop` 设置上下外边距;
7. CSS 代码书写顺序：显示样式 > 自身样式 > 文本样式
8. 显示样式： `position; display; float; list-style; clear;`
9. 自身样式： `width; height; margin; padding; border; background;`
10. 文本属性： `text-align; text-indent; line-height; font; vertical-align; text-decoration; color; cursor;`
11. 每个页面只能有一个 ID;
12. `font-weight` 的值避免使用英文而用数字表示，减少 css 解析步骤，加快渲染速度；500 表示 normal，700 表示 bold;
13. 由于部分浏览器不支持中文，所以设置 `font-family` 是避免中文书写，最有写法为使用 unicode 编码，同时也可以减少 css 解析步骤，加快渲染速度。

微软雅黑	\5FAE\8F6F\96C5\9ED1	Microsoft YaHei
宋体	\5B8B\4F53	SimSun
黑体	\9ED1\4F53	SimHei



【 JS 规范 】

制订此规范的初衷是让所有的代码都变成他人容易阅读,并利于维护的。

1. 所有变量用 var 声明;
2. 所有变量在最开始处声明;
3. JS 尽量避免使用全局变量,复杂应用写成组件,通过构造函数实现多态,写在公共组件或外部 JS 中,简单应用直接写在 init 函数中,通过命名空间或匿名函数将变量包进闭包中;
4. 变量必须在声明初始化以后才能使用,即便是 NULL 类型;
5. 变量名请勿使用系统保留关键字;
6. 变量应该尽量保持最小的生存周期,及时销毁防止内存泄露;
7. 浮点变量必须指明小数点后一位(即使是 0);
8. 浮点变量必须指明实部,即使为零(使用 0. 开头)。
9. 书写过程过, 每行代码结束必须有分号;
10. 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;
11. 变量命名: 驼峰式命名;
12. 原生 JavaScript 变量要求是纯英文字母, 首字母须小写;
13. jQuery 变量要求首字符为 '\$ ',
14. 其他与原生 JavaScript 规则相同, 变量要集中声明, 避免全局变量;
15. 类命名: 首字母大写, 驼峰式命名;
16. 函数命名: 首字母小写驼峰式命名;
17. 命名语义化, 尽可能利用英文单词或其缩写;
18. 尽量避免使用存在兼容性及消耗资源的方法或属性, 比如 eval() & innerText;
19. 代码结构明了, 加适量注释. 提高函数重用率;
20. 类/函数/方法前需添加注释,并注明参数类型,返回值类型;
21. 空格的使用:



操作符使用空格隔开（包括三元操作符），如：

```
var x = a - b;
```

```
var y = result == true ? 1 : 0;
```

逗号 (,) 与 for 循环中的分号 (;) 后添加空格。例如：

```
var test = Array(), test2;
```

```
for(i = 0; i < 100; i++) {
```

```
    Statements;
```

```
}
```

冒号 (:) 后添加空格

逻辑块之间以空行隔开



【 图片规范 】

1. 图片统一放在 images 文件夹下面 ；
2. IMG 标签没有结束标签，但在结尾要加斜线：
3. 必须指定 alt 属性；
4. 图片格式：图片只允许使用 jpg、gif、png 压缩格式，权衡图片质量和文件大小，保证不过分失真的情况下是图片越小越好。例如，色彩丰富的图片使用 jpg 格式比较好，在 ps 下图片品质选 70 左右，图片既不会失真，而且图片也足够小；在色彩不是很丰富的图片用 gif 和 png-8 格式比较好；如果图片有半透明效果则保存为 png-24 格式；
5. 严格区分内容图片和背景图片。做为背景的图片采用 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 耦合度增加了，增加了维护成本。如果图片有修改，不要删已添加的图片，在空白处新增修改后的图片，减小修改的风险 ；
6. 图片大小一定的情况下，在 IMG 标签中指定 width 和 height 属性，减少浏览器的重绘次数，有利于页面的加载；



【 SEO 针对性规范 】

一、 META 标签说明

为了能够更好的为搜索引擎准备内容，meta 标签采用如下的定义方式，允许搜索机器人搜索站内所有链接。

```
<meta name=" robots" content=" all" />
```

设置站点作者信息

```
<meta name=" author" content=" " />
```

设置站点版权信息

```
<meta name=" Copyright" content=" ? 版权" />
```

二、 标题规范

1. 为所有页加上标题。标题简短精练、高度概括，标题长度一般不超过 30 个汉字；
2. 标题要含有关键字，关键字个数不超过 3 个；
3. 频道首页标题格式：频道名_站点名；
4. 栏目首页标题格式：栏目名_频道名_站点名；
5. 新闻最终页标题格式：网页标题_频道名_站点名；
6. 非新闻最终页标题格式：核心关键字_栏目名_频道名_站点名；
7. 专题标题格式：专题名_频道名_站点名；

三、 元标签规范

1. 为所有页加上描述标签，<meta name="description" content=" ">。
2. 描述要清晰明了地写出网页内容，连贯通顺，描述中出现关键字，与正文内容相关。这部分内容是给人看的，要写得详细，让人感兴趣，吸引用户点击。
3. 遵循简短原则，字数一般不超过 120 个汉字。
4. keywords 标签规范
5. 每个频道和栏目可以加 keyword 标签，但是文章页就不要加了



四、 图片规范

1. 在保持图片质量的情况下尽量压缩图像的文件大小。
2. ALT 属性，为图片加上 ALT 属性文字说明，，ALT 说明包含该页的核心关键字在其中；
3. 为图片链接加上 ALT 说明，ALT 说明加入和目标页主题相关的热门关键字。
4. 尽量在图片和图片链接的上方或下方加上包含核心关键字的图片文字说明。
5. 为频道 logo 增加 ALT 说明。

五、 关键字规范

1. 页面关键词只涉及到当前页面的内容，而不去涉及整个网站、所在频道等内容。
2. 在关键词选择上，可以采取具体关键词+主关键词的形式，在某种情况下，标题也可以作为一个比较具体的关键词。
3. 站在受众的角度考虑，结合用户的一些搜索习惯，来填写适当的关键词。（可参照百度的相关搜索）。不同的关键词一定要用英文下的逗号(,)隔开。
4. 关键词以 3-5 个为宜。

六、 链接规范

（一） 链接基本规范

1. 链接要避免链接错误，比如调用不存在的图片和链接到不存在的网页。
2. 确保每个页面都可以通过一个文本链接到达。
3. 锚文本要准确，要能反映目标页主题。导出、导入和内部链接的锚文本要尽量包含目标页的核心关键字。例如：“中国足球”比“国内足坛”更常用、更适合做锚文本。
4. 不要使用图片来显示重要名称、内容或链接。如果网页上的重要内容和关键字不能以普通的 HTML 格式显示，使用 ALT 标注。避免用 JS、Flash、大幅图片来制作整个页面；如果一定要，必须在页面中留下文本内容。
5. 在最终页、专题页、栏目页、频道页放置和本页内容相关的新闻、搜索、论坛链接。
6. 制作 404 错误引导页，引导用户跳转其它页面，比如网站地图页，不建议全部跳转到首页



（二） 文内链接规范

1. 在正文页内给热门关键字加文内链接：一篇文章内，文内链接不超过三个最优。
2. 关键字链接遵循最大优先原则，如果文内同时出现母关键字和子关键字，优先给予关键字加链接。例如：文内出现“北京奥运会”，给“北京奥运会”加链接，而不是“奥运”或“北京奥运”。
3. 优先给段落标题加关键字链接；优先给正文前面内容加关键字链接；优先给搜索排名靠后、频道要推广的热门关键字加链接。
4. 单个关键字不得在同一文内出现两处链接。如果出现多处，优先给正文前面、段落标题中的关键字加链接。
5. 各频道交叉加文内关键字链接。关键字为各频道推广热门关键字，如果其它频道的推广关键字和本频道的冲突，优先加其他频道的关键字。
6. 建议文章内部链接为绝对路径的 URL，这样被转载后可以增加外链，也可避免链接错误。

（三） 外链规范

1. 友情链接等要寻找相似性高，质量高的网站链接，
2. 在网站的导出链接里参数里，一定要加上 target="_blank"，
3. 不使用外链发布软件，在外部发链接时要注意该网站的相关性，不宜在短时间内发出大量外链

（四） 导航规范

1. 导航要清晰，所有页面都能链回首页、频道首页；使用文字链接，不要使用嵌入 JS 文件的方式实现网站导航，尽量避免使用 flash 及图片导航。为所有正文页加小导航。使用文字链接；列出当前页的所有上级页（逻辑结构）的名称及链接，包括频道名、栏目名、子栏目名（或专题名）
2. 导航链接不得包含重定向 URL 链接。重定向发生时，把旧 URL 永久重定向到新 URL；逐步把站内的相关链接由新 URL 替换；确保导航和小导航第一时间完成替换；尽量把站外的友情链接协商改成新 URL。

七、 URL 规范

1. 缩短 URL 长度：长度保持在 65 个英文字母之内。调整网站结构，避免不必要地设置过多层次的目录，目录最好不要超过三级。动态页面的 URL 的参数要尽量少于 3 个。



2. 首页带斜杠，不加 index 后缀。

八、代码优化规范

1. 代码严格遵守 HTML 规范，确保搜索引擎能够正确分析页面代码。
2. 用 Web 标准 (DIV+CSS) 重构页面模板，不使用 table 控制排版，不使用 table 嵌套。
3. 字体设置，文章、段落标题等用<h1><h2>...<h6>依此显示重要性的递减。黑体字用文字标签，尽量避免使用 CSS 中的 style="font-weight:bold"。
4. CSS、JS 代码采用外部调用。
5. 尽量少使用 frame 框架结构。如果网页使用了框架 (frame)，则必须在代码中使用 "Noframes" 标签进行优化。在<Noframes></Noframes>区域中包含指向 frame 页的链接以及带有关键字的描述文本。
6. 广告和与正文无关的内容尽量用 iframe、JS 等方式调用显示。

补充：制作好一个专业的 404，网站 iss 日志放到根目录，网站还没有修改完整的时候，要用 robots 屏蔽蜘蛛，避免过早的被搜索引起收录。影响网站在搜索引擎中的权重。网站上线之后 ftp 中多余的文件都应删除，避免蜘蛛抓取一些无关紧要的页面。