
前端开发设计规范

目录

前端开发设计规范	1
一、HTML 使用规范	1
1.1、页面文件命名规范	1
1.2、页面 head 部分书写规范	1
1.3、HTML 元素开发规范	2
1.3.1、HTML 元素书写规范	2
1.3.2、HTML 元素命名规范	3
二、WEB 页面开发规范	4
2.1、错误跳转页面的处理	4
2.2、提示信息的处理	4
2.3、页面的返回	4
2.4、提交前数据的判断验证	4
2.5、删除操作	5
2.6、页面中 java 代码的使用	5
2.7、网站页面布局规范	5

2.7.1、前台页面尺寸.....	5
2.7.2、标准网页广告图标规格（参考）.....	6
2.7.3、页面字体.....	6
2.7.4、字体颜色.....	7
三、javaScript 开发规范.....	7
3.1、javaScript 文件命名规范：.....	7
3.2、javaScript 开发规范.....	7
3.2.1、javaScript 书写规范.....	7
3.2.2、javaScript 命名规范.....	8
四、css 样式规范.....	9
4.1、css 样式文件命名规范.....	9
4.1.1、通用样式文件命名规范：.....	9
4.1.2、业务类样式文件命名规范.....	10
4.1.3、css 样式文件命名须知.....	10
4.2、css 样式文件存放目录规范.....	10
4.3、css 样式定义规范.....	11
4.3.1、css 样式内容顶部注释规范.....	11
4.3.2、css 样式内容注释规范.....	11
4.3.3、css 样式定义规范.....	12

4.3.4、css 样式常用 id 的命名	13
4.3.5、css 样式常用 class 的命名	14
4.4、css 样式书写规范	15
4.4.1、css 样式排版规范	15
4.4.2、css 样式书写风格规范	15
4.4.3、css 样式属性定义顺序规范	16
4.4.4、css 样式其他规范	16
4.4.5、css 样式 Hack 的使用	17
4.4.6、字体定义规范	18
4.4.7、css 样式检测	18
4.4.8、注意事项	18
4.5、css 样式引用规范	19
4.6、媒体内容命名规范	19
五、项目文件存放规范	19
六、前端开发规则	20

一、HTML 使用规范

1.1、页面文件命名规范

命名格式为：项目名缩写_所属功能_所属功能子项_... .jsp/html...，文件命名下划线不能超过三个，命名尽量使用简短的能明确表明文件用途的英文或者英文简写。

1.2、页面 head 部分书写规范

- 1) 、JSP 页面：需要在页面的最开始部分增加以下语句：

```
<%@ page language="java" import="java.util.*" pageEncoding="UTF-8"%>
```

- 2) 、HTML 页面：需要在页面的最开始部分增加以下语句：

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

- 3) 、HTML5 页面：页面添加编码格式可简写为：

```
<meta charset="utf-8">
```

- 4) 、响应式的网页添加如下语句：

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

- 5) 、title 元素：一般网页必须添加 title 元素，若为框架页面，则可以不写。title 统一使用中文，title 内容要简洁明了，不能超过 20 个字。

- 6) 、外部 js 的引用：页面加载时需要用到的 js 文件写在 head 中，引用时不用写 language 属性，HTML5 可以省略 type 属性，如。

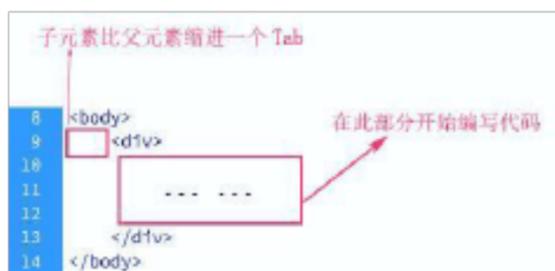
-
- 7) 、外部 CSS 文件的引用：必须使用 link 方式引入，HTML5 可以省略 type 属性，CSS 文件引入要放在 js 文件前。

```
<link rel="stylesheet" href="main.css"/>
```

1.3、HTML 元素开发规范

1.3.1、HTML 元素书写规范

- 1) 、代码的结构要保持完整性，单个标签必须要关闭，如：<div></div>，
等。
- 2) 、子元素要比父元素缩进两个字符。
- 3) 、body 中的所有内容不能直接书写在<body></body>标签中，需要在 body 中嵌入一层 div，所有的元素需要写在改 div 中。



- 4) 、除非必要，所有标签元素的样式都需要使用 CSS 文件来定义。
- 5) 、img 元素：所有展示用图片都要使用 alt 属性添加能简要描述图片的文字说明，如首页的广告图片等，必须具有 width 和 height 属性。
- 6) 、所有 Form 都要指定 action 属性，但属性值需要则填写，不需要则留空，method 属性统一使用 POST；所有 form 表单都要在提交前对输入的数据前进行验证，验证使用 jQuery validate 插件，书写规范如下：



- 7) 、所有不可更改的 input 元素都要设置 readonly 属性。
- 8) 、按功能模块添加简单明了的注释，在功能模块的开始标明模块开始，结束时标明模块结束，注释单独占一行；模块之间留行间隔便于查看代码。
- 9) 、已过时的元素标签使用 CSS 样式来代替，已过时的标签属性禁止使用，使用 CSS 样式定义来实现。



1.3.2、HTML 元素命名规范

涉及到跟服务端交互的元素，元素命名应当与服务端程序中定义的对应变量的名称相同，或使用对象名. 变量名的形式。

页面上使用到的不涉及与服务端交互的 HTML 元素其 id 和 name 需统一，所有命名使用元素标签的缩写_后缀方式，后缀命名使用小驼峰命名法即第一个单词全部使用小写，其他单词首字母大写，常见元素命名规则如下表所示：

HTML 元素	缩写	示例
text 输入框	txt	txt_userName
button 按钮	btn	btn_check
select 下拉选择	sel	sel_beginTime

checkbox 多选项	chk	chk_departmentNumber
div 标记	div	div_result
submit 提交按钮	sub	sub_register
hidden 隐藏值	hdn	hdn_userId

二、WEB 页面开发规范

2.1、错误跳转页面的处理

400、401、403、404、405、406、407、410、412、414、500、501、502 等错误代码使用其相应的统一页面，所有错误信息全部使用中文错误信息，标点符号使用中文半角符号。页面设计与项目功能相匹配，做到简洁友好。

2.2、提示信息的处理

成功提示信息可以使用弹窗或跳转统一成功页面的方式，成功页面的风格与网站总体风格相同，格式如下："成功："+提示信息+"！"。

2.3、页面的返回

所有需要返回上一页的时候使用 `history.back()`；不使用 `history.go(-1)`。

2.4、提交前数据的判断验证

1) 、所有由用户输入的数据在提交前都要进行验证。

2) 、验证方式使用 jQuery validate 插件，需要验证的项：

长度验证：默认长度为数据库中数据的最大长度，中文占两个字符，英文占一个字符。

为空验证：所有不允许为空的输入内容为空时不允许提交。

其他验证：需要根据输入内容的不同设定合适的验证，如 Email 格式是否正确，身份证号格式是否正确等。

3) 、验证后发现错误，需要提示明确的错误信息。错误提示信息样式：字体颜色#ff3366，文字前面加红色感叹号小图标。

4) 、可输入表单需要具有输入内容的提示信息，可使用 placeholder 属性来定义或者自定义，颜色必须使用灰色，字体比网站使用字体小。

2.5、删除操作

所有涉及删除的操作，需要用户进行确认之后才能进行操作。

2.6、页面中 java 代码的使用

页面中不允许使用<% %>的方式嵌入 java 代码。

2.7、网站页面布局规范

2.7.1、前台页面尺寸

1) 、800*600 下，网页宽度保持在 778 以内，就不会出现水平滚动条，高度则视版面和内容决定，1024*768 下，网页宽度保持在 1002 以内，不会出现水平滚动条，高

度同样视版面内容决定。

- 2) 、根据第一条原则，规定网页的尺寸为 width=960px,height=600px。
- 3) 、页面长度原则上不超过 3 屏（可根据实际情况设定），宽度不超过 1 屏。
- 4) 、全尺寸 banner 为 468*60px,半尺寸 banner 为 234*60px,小 banner 为 88*31px。
- 5) 、每个非首页静态页面含图片字节不超过 60K,全尺寸 banner 不超过 14K。

2.7.2、标准网页广告图标规格（参考）

- 1) 、120*120,适用于产品或新闻照片展示。
- 2) 、120*60,主要用于做 LOGO 使用。
- 3) 、120*90,主要应用于产品演示或大型 LOGO。
- 4) 、125*125,适于表现照片效果的图像广告。
- 5) 、234*60,适用于框架或左右形式主页的广告链接。
- 6) 、392*72,主要用于有较多图片展示的广告条,用于页眉或页脚。
- 7) 、468*60,应用最为广泛的广告条尺寸,用于页眉或页脚。
- 8) 、88*31,主要用于网页链接,或网站小型 LOGO。

2.7.3、页面字体

正文内容中文统一使用宋体（可根据需要设定），大小为 12px，标题使用 14px 加粗，不建议使用 13px 字，英文字体使用 Arial 和 Helvetica/Univers。

2.7.4、字体颜色

- 1) 、正文使用灰黑色#333333。
- 2) 、超链接可以使用蓝色#1f376d 或#425c9e，也可以根据整体页面布局进行调整，整个网站的文字超链接样式要统一。

三、javaScript 开发规范

3.1、javaScript 文件命名规范：

- 1) 、可通用的 javaScript 文件：项目名称缩写-文件作用.js。
- 2) 、其他 javaScript 文件：所属功能-文件作用.js。

3.2、javaScript 开发规范

3.2.1、javaScript 书写规范

- 1) 、javaScript 代码都需要写在 javaScript 文件中，在页面中进行调用，调用代码除页面加载时需要用到的外全都写在页面最低端。
- 2) 、如验证用 javaScript 代码等，每个页面不同且比较短的，可以写在页面的最底端，从最后一个外部 javaScript 引入命令后开始。
- 3) 、脚本变量开发规范：变量的使用尽量缩小到小的作用域。如循环使用。
- 4) 、尽量避免使用全局变量。

- 5) 、每一句语句都要以分号 “;” 结束。
- 6) 、函数程序体缩进四个空格，Tab 键为 4 个空格，以 Tab 键作为缩进单位。
- 7) 、函数名与 “(” 之间不应该有空格，“)” 与 “{” 之间加空格。
- 8) 、函数体之间应当加空行。
- 9) 、如果代码本身是清楚的，则只需要在函数定义前，添加简单注释说明该函数的作用，若代码本身比较复杂，可在个别语句后添加简洁明了的注释。

3.2.2、JavaScript 命名规范

- 1) 、常量以及全局变量名必须全部使用大写字母。
- 2) 、不要再命名中使用 “\$” 和 “\” 等特殊字符，不要把 “_” 作为变量名的第一个字符和最后一个字符。
- 3) 、变量名必须使用其类型的缩写字符串开始。各种类型的缩写字符串如下：

变量类型	缩写	变量类型	缩写
整型变量	int	长整型变量	lng
浮点型变量	flt	双精度变量	dbl
对象变量	obj	字符串变量	str
Date 类型变量	dtm	数组	ary
临时变量	tmp		

- 4) 、变量名必须采用有意义的单词命名，如：strUserName、lngArrayIndex，以及不要出现数字编号命名，如：value1，value2...
- 5) 、如果变量名过长可以使用单词缩写，除了被广泛了解的单词缩写以外，所有使用单词缩写的变量名必须在定义时给出注释，如：

```

2 function getUserId() {
3     var strAdName = null; //用于表Administrator帐户的名称
4     var strAdminName = null; //不用给出注释，Admin够广泛了解
5 }

```

6) 、参数变量命名必须加前缀：p_。

7) 、function 命名规范：使用动词+名词的方式，并且命名需要明确指出其作用，动词前缀必须是同函数功能相关的完整动词，如 getUserId ()，用来取用户 Id。

8) 、所有命名使用小驼峰命名法：第一个单词全部使用小写，其他单词首字母大写，如：myClassName。

The diagram shows two function definitions with annotations in red:

- Annotation 1:** "函数功能简要注释" (Brief function description) points to the comment `//获取用户Id` above the first function.
- Annotation 2:** "“)”与“{”之间间隔一个空格" (One space between ")" and "{") points to the space between `)` and `{` in `function getUserId() {`.
- Annotation 3:** "函数之间间隔一行" (One line between functions) points to the blank line between the two function definitions.
- Annotation 4:** "缩进一个 Tab" (Indent one tab) points to the indentation of the `var strUserName;` line inside the second function.

四、css 样式规范

4.1、css 样式文件命名规范

4.1.1、通用样式文件命名规范：

1) 、整个项目通用的 css 布局样式文件命名为：layout.css。

如：通过于整个网站中使用的 div、h1、img 等的定义文件。

2) 、对引用的开源 css 样式重定义的文件命名为：开源的 css 样式文件名-reset.css

如：bootstrap.css 为引用的 css 样式文件，现在对它的.btn 重新定义样式内容，这类样式的文件名命名为：开源的 css 样式文件名-reset.css。

一个项目建议最多使用一个 css 框架。

-
- 3) 、外部引用的字体 css 样式文件命名为：font-加引用的字体名字.css，若有多个字体引用则使用：font-style.css。
 - 4) 、自定义的响应式布局样式命名为：项目名称缩写 - responsive.css。
 - 5) 、跨项目通用功能定义的 css 样式文件命名为：实现的功能.css。
 - 6) 、为 IE 浏览器做兼容处理的样式命名为：ie-版本号.css。

4.1.2、业务类样式文件命名规范

自定义的样式统一定义在一个文件内，文件命名为：项目名称-style.css。

4.1.3、css 样式文件命名须知

- 1) 、以上所有的 css 样式文件命名必须是小写字母，不允许中文、大写字母及其他特殊字符等。如果有样式版本共存或更新需要保留老版本样式，请在样式文件名后面加上“-版本号”，最新版本样式文件除外，同样版本号只允许是数字或小写字母。
- 2) 、以上 css 文件命名制定的规范只适用普遍的情况，特殊情况下请开发人员根据实际情况自行命名，但必须遵行 1) 的规定。

4.2、css 样式文件存放目录规范

项目文件存放目录将在后面说明，此处只针对 css 样式文件存放规定。

- 1) 、字体样式文件存放文件夹：/font/，包括字体的源文件。
- 2) 、引用的样式保持原来的目录结构，如果有多个引用的样式则按功能新建文件夹来分别存放。

-
- 3) 、其他样式文件不用单独新建文件夹。

4.3、css 样式定义规范

4.3.1、css 样式内容顶部注释规范

- 1) 、请使用 utf-8 编码。
- 2) 、申明 css 定义的内容概述或目的，申明编写的人员、更新日期。
- 3) 、请谨慎使用@import url(……)引入其他 css 样式文件。

4.3.2、css 样式内容注释规范

- 1) 、模块注释必须单独写在一行，简要说明模块的功能。

```
1.  /* Header */  
2.  /* Footer */  
3.  /* Gallery */
```

- 2) 、单行注释可以写在单独一行，也可以写在行尾，注释中的每一行长度不超过 30 个汉字，或者 60 个英文字符。

```
1.  /* this is a short comment */
```

- 3) 、多行注释必须写在单独行内，即/*后换行写注释，*/单独写一行。

```
1.  /*  
2.  * this is comment line 1.  
3.  * this is comment line 2.  
4.  */
```

- 4) 、用于标注修改、待办等信息的注释以单行注释为基准。

```
1.  /* TODO: xxxx by name 2013-04-13 18:32 */  
2.  /* BUGFIX: xxxx by name 2012-04-13 18:32 */
```

-
- 5) 、对一个代码区块注释（可选），将样式语句分区块并在新行中对其注释。

```
1.  /* module: module1 by 张三 */  
2.  ...  
3.  /* module: module2 by 张三 */
```

4.3.3、css 样式定义规范

- 1) 、使用有意义的或通用的 id 和 class 命名：id 和 class 的命名应反映该元素的功能或使用通用名称，而不要用抽象的晦涩的命名。反映元素的使用目的是首选；使用通用名称代表该元素不表特定意义，与其同级元素无异，通常是用于辅助命名；使用功能性或通用的名称可以更适用于文档或模版变化的情况。
- 2) 、id 和 class 命名越简短越好，只要足够表达涵义。这样既有助于理解，也能提高代码效率。
- 3) 、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。
- 4) 、类型选择器避免同时使用标签、id 和 class 作为定位一个元素选择器，可以使用 class 层级来代替。
- 5) 、规则命名中，一律采用小写加中划线的方式，不允许使用大写字母或下划线。
- 6) 、命名避免使用中文拼音，应该采用更简明有语义的英文单词进行组合。
- 7) 、命名注意缩写，但是不能盲目缩写，具体请参见常用的 CSS 命名规则。
- 8) 、不允许通过 1、2、3 等序号进行命名，避免 class 与 id 重名。
- 9) 、id 用于标识模块或页面的某一个父容器区域，名称必须唯一，不要随意新建 id。

10) 、class 用于标识某一个类型的对象，命名必须言简意赅，尽可能提高代码模块的复用，样式尽量用组合的方式。

4.3.4、css 样式常用 id 的命名

1) 、 页面结构命名

结构	命名	结构	命名
容器	container	页头	header
页面主体	main	页尾	footer
侧栏	sidebar	栏目	column
内容	content/container	内容	content/container
外围布局	wrapper	左右中	left right center

2) 、 导航命名

结构	命名	结构	命名
导航	nav	主导航	mainbav
子导航	subnav	顶导航	topnav
边导航	sidebar	左导航	leftsidebar
右导航	rightsidebar	菜单	menu
子菜单	submenu	标题/摘要	title/summary

3) 、 功能命名

结构	命名	结构	命名
标志	logo	广告	banner
登陆	login	登录条	loginbar
注册	regsiter	搜索	search
功能区	shop	标题	title
加入	joinus	状态	status

按钮	btn	滚动	scroll
标签页	tab	文章列表	list
提示信息	msg	当前的	current
小技巧	tips	图标	icon
注释	note	指南	guild
服务	service	热点	hot
新闻	news	下载	download
投票	vote	合作伙伴	partner
友情链接	link	版权	copyright

以上样式定义须符合 css 样式定义规范。

4.3.5、css 样式常用 class 的命名

- 1) 、颜色：单独定义颜色时使用颜色的名称，复合颜色用中划线连接。如：.blue
或.blue-green。
- 2) 、字体大小：单独定义时使用 font-字号；如：.font-12。
- 3) 、对齐样式：使用对齐目标的英文名称。如：.left。
- 4) 、其他样式：功能缩写-当前的作用区域；如果有多层级，则下一层级定义为：上一级的样式名-目标区块命名；在嵌入其他门户网站中的应用，如果门户网站没有特别规定，为了避免样式冲突必须为每一个样式加上前缀，前缀可以按实际情况来设置。

4.4、css 样式书写规范

4.4.1、css 样式排版规范

- 1) 、使用一次 tab 键或者 4 个空格作为缩进。

```
@charset "utf-8";  
/* CSS Document */  
body {  
    line-height:1  
}
```

- 2) 、规则可以写成单行，或者多行，但是整个文件内的规则排版必须统一。
- 3) 、对于可以发布的网站样式建议进行压缩，直接使用网上的 css 格式化工具进行压缩。

4.4.2、css 样式书写风格规范

- 1) 、如果是在 html 页面中定义的 css 样式，则必须写成单行，不建议在 html 页面中定义 css 样式。
- 2) 、每一条规则的大括号 { 前后加空格，每一条规则结束的大括号 } 前加空格。
- 3) 、属性名冒号之前不加空格，冒号之后加空格。
- 4) 、每一个属性值后必须添加分号；并且分号后空格。
- 5) 、多个 selector 共用一个样式集，10 个以内的 selector 必须写成多行形式，10 个以上的 selector 写成一行。
- 6) 、多行书写时每一条规则结束的大括号 } 必须与规则选择器的第一个字符对齐。



4.4.3、css 样式属性定义顺序规范

- 1) 、显示属性：display/list-style/position/float/clear …。
- 2) 、自身属性（盒模型）：width/height/margin/padding/border。
- 3) 、背景：background；行高：line-height。
- 4) 文本属性：color/font/text-decoration/text-align/text-indent/ white-space。
- 5) 、其他：cursor/z-index/zoom/overflow。
- 6) 、CSS3 属性：transform/transition/animation/box-shadow/border-radius。
- 7) 、使用 CSS3 的属性，如果有必要加入浏览器前缀，则按照 -webkit- / -moz- / -ms- / -o-的顺序进行添加，标准属性写在最后。
- 8) 、属性定义顺序规范不作严格要求，但在使用 css3 时最好加上前缀。

4.4.4、css 样式其他规范

- 1) 、使用单引号，不允许使用双引号。
- 2) 、每个声明结束都应该带一个分号，不管是不是最后一个声明。
- 3) 、除 16 进制颜色和字体设置外，CSS 文件中的所有的代码都应该小写。
- 4) 、选择器应该在满足功能的基础上尽量简短，减少选择器嵌套，查询消耗。但是一

定要避免覆盖全局样式设置。

- 5) 、 background、 font、 margin、 padding、 border 等可以缩写的属性，尽量使用缩写形式。
- 6) 、 在 css 中谨慎使用*选择符。
- 7) 、 除非必须，否则，一般有 class 或 id 的，不需要再写上元素对应的 tag。
- 8) 、 0 后面不需要单位，比如 0px 可以省略成 0，0.8px 可以省略成.8px。
- 9) 、 如果是 16 进制表示颜色，则颜色取值字母应该大写；如果可以，颜色尽量用三位字符表示，例如#AABBCC 写成#ABC。
- 10) 、 如果没有边框时，不要写成 border:0，应该写成 border:none。
- 11) 、 在保持代码解耦的前提下，尽量合并重复的样式。
- 12) 、 除了重置浏览器默认样式外，禁止直接为 html tag 添加 css 样式设置。

4.4.5、css 样式 Hack 的使用

Css hack 仅仅是为了兼容浏览器对 css 的解析而设置，请不要使用浏览器检测和 CSS Hacks 作为解决浏览器解析差异的首选方式，但应被视为最后的手段。允许使用 hack 只会带来更多的 hack，你越是使用它，你越是会依赖它。

区别属性:	
IE6	_property:value
IE6/7	*property:value
IE6/7/8/9	property:value\9

区别规则:	
IE6	* html selector [...]
IE7	*:first-child+html selector [...]
非IE6	htmlbody selector { ... }
firefox only	@-moz-document url-prefix() [...]
saf3+/chrome1+	@media all and (-webkit-min-device-pixel-ratio:0) { ... }
opera only	@media all and (-webkit-min-device-pixel-ratio:10000),not all and (-webkit-min-device-pixel-ratio:0) [...]
iPhone/mobile webkit	@media screen and (max-device-width: 480px) { ... }

4.4.6、字体定义规范

- 1) 、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)，详细介绍参详见附件。
- 2) 、字体粗细采用具体数值，粗体 bold 写为 700，正常 normal 写为 400。
- 3) 、font-size 必须以 px 或 pt 为单位，推荐用 px(注：pt 为打印版字体大小设置)，不允许使用 small/medium/large/x-large/xx-large 等不确定的值。
- 4) 、为了对 font-family 取值进行统一，更好的支持各个操作系统上各个浏览器的兼容性，font-family 不允许在业务代码中随意设置。

4.4.7、css 样式检测

写好一份 css 样式文件之后，如果不完全确定定义是否符合 W3C 标准，可借助在线检测工具来检测。常用的测试工具：W3C CSS validator、CSS Lint 等不限制。

4.4.8、注意事项

- 1) 、不要轻易改动全站级 CSS 和通用 CSS 库。改动后，要经过全面测试。
- 2) 、避免在 CSS 中使用 expression，避免使用 filter，尽量不要在 CSS 中使用 !important。
- 3) 、避免过小的背景图片平铺，允许使用 1px 的纯色或渐变色图片平铺作为背景。
- 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)，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。

- 5) 、背景图片请尽可能使用 sprite 技术，减小 http 请求，考虑到多人协作开发，sprite 按照模块、业务、页面来划分均可。

4.5、css 样式引用规范

- 1) 、除必需的情况下，所有的 css 样式都定义在单独的 css 文件中作为外联样式使用 link 引用到页面，当前页用不到的样式禁止在页面上引用。
- 2) 、在页面中禁止使用 @import 来关联其他外部样式。

4.6、媒体内容命名规范

- 1) 、图片优先考虑 .png 格式，请尽可能使用 sprite 技术，同一作用的图片放在以功能命名的文件夹中。图片以“功能缩写-子功能缩写”来命名；轮播图片可以用功能缩写-数字来命名。
- 2) 、视频、音频、flash 文件以语义明确的英文来命名。
- 3) 、所有的多媒体文件命名不允许出现中文、大写字母和特殊字符。

五、项目文件存放规范

- 1) 、所有 css 样式文件放在 webRoot 目录下的 css 文件内。
- 2) 、所有图片放在 webRoot 目录下的 images 文件夹中，引用样式中用到的图片允许放到 css 目录下按照 css 文件夹目录规范存放。允许在文件夹中按照功能、模块等

创建文件夹存放。

- 3) 、多媒体文件放在 webRoot 目录下 media 文件夹中按照媒体类型创建文件夹存放。
- 4) 、js 文件放在 webRoot 目录下 js 文件夹中,具体的按照 js 文件夹目录规范存放。
- 5) 、与服务端交互的页面放到 webRoot 目录下 WEB-INF 目录中,单独创建文件夹存放;
将静态页面直接创建文件夹放在 webRoot 目录下。
- 6) 、外部引用的组件放在 webRoot 目录下以组件命名的文件夹中,目录结构保持不变。

六、前端开发规则

- 1) 、开发前需要编写设计文档、开发工作计划,项目开发按照设计文档进行开发,开发过程中遇到需要修该的问题及时向上级反映,讨论决定再进行修改。
- 2) 、在开发工程每完成一个功能都必须对该功能进行测试,在完成整个项目放到测试服务器之前,开发人员自己必须完整测试,允许不全面的测试,但是不允许存在严重的 bug。
- 3) 、本文档编写的规范仅适用于开发人员前端设计的部分,外部引用的文件不在本文档规定范围。