目录结构

**# 项目布局**  
```  
.  
├── pubulic //公用文件  
├── src // 源码目录  
│ ├── components // 组件  
│ │ ├── common // 公共组件  
│ │ │ ├── js // 公用方法库  
│ │ ├── footer  
│ │ │ └── footGuide.vue // 底部公共组件  
│ │ └── header  
│ │ └── head.vue // 头部公共组件  
│ ├── constants // 基本常量配置  
│ │ ├── API.ts // api配置  
│ │ ├── TEXT.ts // 常量配置  
│ │ ├── URL.ts // 常用url  
│ ├── images // 公共图片  
│ ├── views  
│ │ ├── demo  
│ │ │ ├── Form.vue // 表格页面  
│ │ │ └── Table.vue  
.。。。

流程与信息管理部前端开发规范

目录

[前端开发设计规范 1](#_Toc390079552)

[一HTML使用规范 1](#_Toc390079553)

[1.1页面文件命名规范 1](#_Toc390079554)

[1.2页面head部分书写规范 1](#_Toc390079555)

[1.3 HTML元素开发规范 2](#_Toc390079556)

[1.3.1 HTML元素书写规范 2](#_Toc390079557)

[1.3.2 HTML元素命名规范 3](#_Toc390079558)

[二WEB页面开发规范 4](#_Toc390079559)

[2.1错误跳转页面的处理 4](#_Toc390079560)

[2.2提示信息的处理 4](#_Toc390079561)

[2.3页面的返回 4](#_Toc390079562)

[2.4提交前数据的判断验证 4](#_Toc390079563)

[2.5删除操作 5](#_Toc390079564)

[2.6页面中java代码的使用 5](#_Toc390079565)

[2.7网站页面布局规范 5](#_Toc390079566)

[2.7.1前台页面尺寸 5](#_Toc390079567)

[2.7.2标准网页广告图标规格（参考） 6](#_Toc390079568)

[2.7.3页面字体 6](#_Toc390079569)

[2.7.4字体颜色 7](#_Toc390079570)

[三javaScript开发规范 7](#_Toc390079571)

[3.1 javaScript文件命名规范： 7](#_Toc390079572)

[3.2 javaScript开发规范 7](#_Toc390079573)

[3.2.1 javaScript书写规范 7](#_Toc390079574)

[3.2.2 javaScript命名规范 8](#_Toc390079575)

[四css样式规范 9](#_Toc390079576)

[4.1 css样式文件命名规范 9](#_Toc390079577)

[4.1.1通用样式文件命名规范： 9](#_Toc390079578)

[4.1.2业务类样式文件命名规范 10](#_Toc390079579)

[4.1.3 css样式文件命名须知 10](#_Toc390079580)

[4.2 css样式文件存放目录规范 10](#_Toc390079581)

[4.3 css样式定义规范 11](#_Toc390079582)

[4.3.1 css样式内容顶部注释规范 11](#_Toc390079583)

[4.3.2 css样式内容注释规范 11](#_Toc390079584)

[4.3.3 css样式定义规范 12](#_Toc390079585)

[4.3.4 css样式常用id的命名 13](#_Toc390079586)

[4.3.5 css样式常用class的命名 14](#_Toc390079587)

[4.4 css样式书写规范 15](#_Toc390079588)

[4.4.1 css样式排版规范 15](#_Toc390079589)

[4.4.2 css样式书写风格规范 15](#_Toc390079590)

[4.4.3 css样式属性定义顺序规范 16](#_Toc390079591)

[4.4.4 css样式其他规范 16](#_Toc390079592)

[4.4.5 css样式 Hack的使用 17](#_Toc390079593)

[4.4.6字体定义规范 18](#_Toc390079594)

[4.4.7 css样式检测 18](#_Toc390079595)

[4.4.8注意事项 18](#_Toc390079596)

[4.5 css样式引用规范 19](#_Toc390079597)

[4.6媒体内容命名规范 19](#_Toc390079598)

[五项目文件存放规范 19](#_Toc390079599)

[六前端开发规则 20](#_Toc390079600)

# 一HTML使用规范

## 1.1页面文件命名规范

命名格式为：引文件统一使用index.htm index.html index.asp文件名（小写）

各子页命名的原则首先应该以栏目名的英语翻译取单一单词为名称。统一为小驼峰式命名法（Camel Case）首字母小写。

例如：studentInfo、userInfo、productInfo

关于我们 \ aboutUs

信息反馈 \ feedback

产 品 \ product

如果栏目名称多而复杂并不好以英文单词命名，则统一使用该栏目名称拼音或拼音的首字母表示；

每一个目录中应该包含一个缺省的html 文件，文件名统一用index.htm index.html index.asp；。

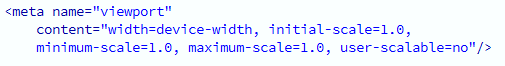
Vue组件命名参照vue2.0官方文档命名规范不在赘述。（https://cn.vuejs.org/）

## 1.2页面head部分书写规范

1. HTML页面：需要在页面的最开始部分增加以下语句：
2. HTML5页面：页面添加编码格式可简写为：



1. 响应式的网页添加如下语句：



1. 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。
2. 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。
3. 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。



## 1.3 HTML元素开发规范

### 1.3.1 HTML元素书写规范

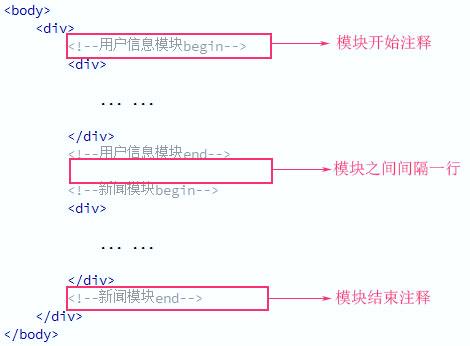
1. 代码的结构要保持完整性，单个标签必须要关闭，如：<div></div>，<br/>等。
2. 子元素要比父元素缩进两个字符。
3. body中的所有内容不能直接书写在<body></body>标签中，需要在body中嵌入一层div，所有的元素需要写在改div中。



1. 除非必要，所有标签元素的样式都需要使用CSS文件来定义。
2. 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。
3. 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，验证书写规范如下：



1. 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。
2. 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名相同，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：提示信息+"成功！"。

## 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. 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。

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

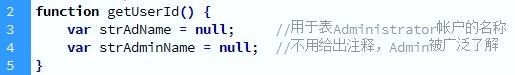
1. javaScript代码都需要写在javaScript文件中，在页面中进行调用，调用代码除页面加载时需要用到的外全都写在页面最低端。
2. 如验证用javaScript代码等，每个页面不同且比较短的，可以写在页面的最底端，从最后一个外部javaScript引入命令后开始。
3. 脚本变量开发规范：变量的使用尽量缩小到小的作用域。如循环使用。
4. 尽量避免使用全局变量。
5. 每一句语句都要以分号“；”结束。
6. 函数程序体缩进四个空格，Tab键为4个空格，以Tab键作为缩进单位。
7. 函数名与“（”之间不应该有空格，“）”与“{”之间加空格。
8. 函数体之间应当加空行。
9. 如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

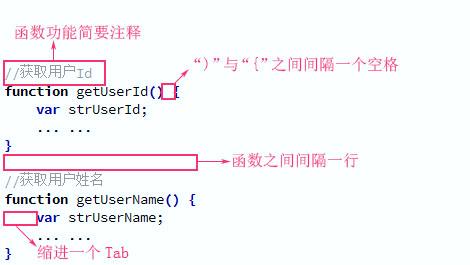
1. 常量以及全局变量名必须全部使用大写字母。
2. 不要再命名中使用“$”和“\”等特殊字符，不要把“\_”作为变量名的第一个字符和最后一个字符。
3. 变量名必须使用其类型的缩写字符串开始。各种类型的缩写字符串如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 | |  |  |

1. 变量名必须采用有意义的单词命名，如：strUserName、arrayIndex，以及不要出现数字编号命名，如：value1，value2…
2. 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：



1. 参数变量命名必须加前缀：p\_。
2. 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。
3. 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。



# 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divh1img等的定义文件。

1. 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。

1. 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。
2. 自定义的响应式布局样式命名为：项目名称缩写–responsive.css。
3. 跨项目通用功能定义的css样式文件命名为：实现的功能.css。
4. 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

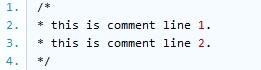
1. 模块注释必须单独写在一行，简要说明模块的功能。



1. 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. 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. 用于标注修改待办等信息的注释以单行注释为基准。



1. 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。



### 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(footft) content(cont)、menu、nav、main、submain、sidebar(side)、logo、banner title(tit)、pop(pop)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123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 |

1. 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 |

1. 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。



1. 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。
2. 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

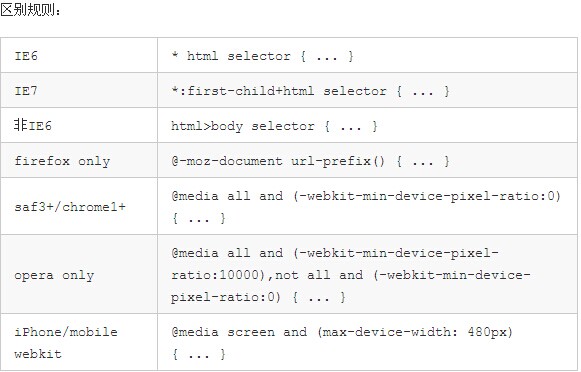
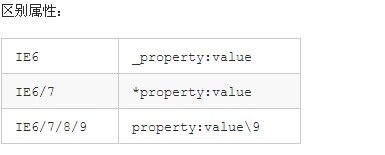
1. 显示属性：display/list-style/position/float/clear …。
2. 自身属性（盒模型）：width/height/margin/padding/border。
3. 背景：background；行高：line-height。
4. 文本属性：color/font/text-decoration/text-align/text-indent/ white-space。
5. 其他：cursor/z-index/zoom/overflow。
6. CSS3属性：transform/transition/animation/box-shadow/border-radius。
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. backgroundfontmarginpaddingborder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。



### 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rem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,移动端最好使用rem相对单位。
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 validatorCSS 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改的问题及时反映，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。