**WEB前端开发规范**

1. **基本原则**

\* 遵循内容（HTML）、显示（CSS）、行为（JavaScript）分离的代码组织模式

\* 代码格式化，保持干净整洁

1. **HTML部分**
   1. 添加必须的utf-8的字符集，并且使用HTML5的简洁方式：

<meta charset="utf-8" />

* 1. 遵循xhtml 1.0规则：

1. 所有标签必须结束
2. 所有标签必须小写
3. 标签属性必须使用成对引号（单引号或双引号）
4. 标签属性必须有值：

<select>

<option selected="selected"></option>

</select>

<input type="checkbox" checked="checked" />

1. 所有特殊符号必须转义（&、<、>、©、»…）
   1. 标签属性命名规范

id： 连接符命名法“hello-world”

class: 连接符命名法“hello-world”

name： 骆驼式命名法“helloWorld”

1. 表单元素的id必须加以下前缀

label： lbl

text： txt

password： txt

textarea： txt

file： txt

radio： rad

checkbox： chk

submit： btn

reset： btn

button： btn

hidden： hid

1. 用于结构布局的元素id命名

主容器： main

页头： header

页脚： footer

内容区域： content

LOGO： logo

主导航： main-nav

二级导航： sub-nav

1. name命名

一般用于表单元素，根据当前元素id属性值命名，去掉id属性值的前缀和所有连接符，使用骆骆式命名法命名，例如id=”txt-id-card”，那么name=”idCard”。

* 1. 合理使用标签，语义化结构

1. 标签合理嵌套

a、span、strong、em、p、h1~h6等元素不能包含：div、ul、ol、dl、p

1. 严禁多div症、多span症、多table症，正确使用标签表现DOM结构，在文档去除css的情况下，任然具有结构和可读性，以下是html标记的使用规范：

p：文本段落；

dl：定义列表，可包括标题和内容简介的列表；

ul：无序列表；

ol：有序列表；

h1~h6：文章标题、内容区块标题，根据重要性由大到小区分，h1一个页面只出现一次；

strong/em：强调文本；

img：图像，必须加上alt属性，当图像无法显示时，可表示图像信息，背景和按钮使用css处理，不使用img元素；

table：数据网格，规则的分栏布局，尽可能显性的定宽和定高。

1. 合理化表单结构
   1. 使用fieldset元素包裹相同类别的字段；
   2. 使用legend元素表示字段类别名称；
   3. 使用label表示字段文本，添加必要的for属性，以在点击字段文本时，文本框能获得焦点；
   4. 文本框不使用size属性定义宽度，而使用css的width属性；
   5. 添加maxlength属性限制输入字符的长度。
2. 严禁使用已在xhtml 1.0中已移除的的标签：

s、i、b、font

* 1. 控制页面默认显示状态

使用js来控制和切换显示的区块、导航、tab，必须先用css处理DOM默认显示状态。

应用示例：

#tabs div **{** display**:**none **}**

#tabs div.selected **{** display**:**block **}**

<div id="tabs">

<ul>

<li><a href="#tab-1">item1</a></li>

<li><a href="#tab-2">item2</a></li>

<li><a href="#tab-3">item3</a></li>

</ul>

<div id="tab-1" class="selected"></div>

<div id="tab-2"></div>

<div id="tab-3"></div>

</div>

1. **CSS部分**
   1. Css 命名规则
2. 样式类名全部用小写，首字符必须是字母，禁止数字或其他特殊字符。由以字母开头的小写字母（a-z）、数字（0-9）、下划线（\_）组成。
3. 可以是单个单词，也可以是组合单词，要求能够描述清楚模块和元素的含义，使其具有语义化。避免使用 123456…red,blue,left,right之类的（如颜色、字号大小等）矢量命名，如class=”left\_news”、class=”2” ，以避免当状态改变时名称失去意义。
4. 尽量用单个单词简单描述class名称。
5. 双单词或多单词组合方式：形容词\_名词、命名空间\_名次、命名空间\_形容词\_名词。例如：news\_list、mod\_feeds、mod\_my\_feeds、cell\_title
   1. Class和ID的使用方法

把id留给后台开发和JS使用，除此之外页面的page id(如首页的外层需要一个ID id=”page\_index”)，页面结构（header main footer）允许用id命名。其他禁止id使用在样式表CSS命名中，一律使用class命名

* 1. 命名空间

在编码思想上，我们可以将页面拆分成不同的层级（布局、模块、元件）。

什么是CSS命名空间？

通过统一的命名规范定义命名的范围，成为CSS class & id命名空间。

**布局:** 以语义化的单词layout作为命名空间，例如主栏布局命名 layout\_main，

只改变layout\_命名空间后面的命名，layout始终保留。布局的命名空间为layout\_xxx。

**模块：**页面是由一个或多个模块组成，模块的英文单词是module，规范简写成mod，如新闻模块mod\_news，照片展示模块mod\_photo\_show。模块的命名空间为mod\_xxx 。

**元件：**元件是属于模块内部的，也可以说模块是由元件和它内部的自有元素组成。如用户照片信息元件cell\_user\_photo。元件的命名空间为cell\_xxx 。

* 1. 添加文档样式

1. 引用外部文件方式添加样式
2. 严禁编写标记内代码，比如<div style="display:none;">就应该写成<div class="hide">，然后在样式表中去完成样式代码编写。
3. 严禁在文档中使用<style type="text/css"></style>代码块。
4. 分割样式表，降低代码复杂性，方便管理和维护。但不分割的太细碎，应考虑以后的维护和管理。
5. 如果是发布版本，请压缩合并代码，将多个样式文件合并为单个文件，在线css代码压缩工具：<http://www.csscompressor.com/>
   1. 属性简写

为了节省字节数及文件大小，以下属性请使用简写方式：

padding: top right bottom left;

margin: top right bottom left;

border: style width color;

border-top: style width color;

border-right: style width color;

border-bottom: style width color;

border-left: style width color;

border-color: top right bottom left;

border-style: top right bottom left;

border-width: top right bottom left;

background: color url(image) repeat position;

list-style: type position url(image);

font-weight: 400 / 700;

* 1. 缩写16进制色值

color和background-color的属性值如果使用的是16进制色值，当6个数字两两相等时，请使用缩写方式编写，比如：#996600请缩写为#960

* 1. 字体：

1. 全局定义字体：body{font: 12px arial, helvetica, sans-serif; line-height: 1.5;}
2. font-family:
   1. 等宽字体组合：Arial, Helvetica, sans-serif;
   2. 不等宽（宽扁）字体组合： Verdana, Trebuchet MS, sans-serif;
   3. 中文字体：除非内容文本需要，不推荐强制定义
   4. 页面采用固定流式布局，使用像素（px）固定元素尺寸。
   5. 编写兼容的CSS代码
3. 页面必须在ie6~8、firefox、opera、safari、chrome下显示兼容；
4. 不使用IE有条件注释方式，对某一版本浏览器编写额外的样式表；

针对某一版本浏览器编写额外的样式表，会对维护和管理照成困难。如出现有显示不兼容现象应首先考虑代码是否有问题（如属性是否对当前浏览器支持），必须编写与常用浏览器都兼容的代码；

不推荐的引用方式：

<!--[if IE 6]>

<link type="text/css" rel="stylesheet" href="ie6.css" />

<![endif]-->

<!--[if lte ie 7]>

<link type="text/css" rel="stylesheet" href="ie7.css" />

<![endif]-->

1. 不要使用!important或下划线等招数编写代码。
   1. 添加必须的注释

可根据全局、布局、区块、功能等进行分类和添加注释，方便维护和代码搜索。

* 1. CSS 通用命名

(1)页面框架命名，一般具有唯一性，推荐用ID命名

|  |  |  |  |
| --- | --- | --- | --- |
| **ID名称** | **命名** | **ID名称** | **命名** |
| 头部 | header | 主体 | main |
| 脚部 | footer | 容器 | wrapper |
| 侧栏 | sidebar | 栏目 | column |
| 布局 | layout |  |  |

(2)模块结构命名

|  |  |  |  |
| --- | --- | --- | --- |
| **CLASS名称** | **命名** | **CLASS名称** | **命名** |
| 模块(如：新闻模块) | mod (mod\_news) | 标题栏 | title |
| 内容 | content | 次级内容 | sub\_content |

(2)导航结构命名

|  |  |  |  |
| --- | --- | --- | --- |
| **CLASS名称** | **命名** | **CLASS名称** | **命名** |
| 导航 | nav | 主导航 | main\_nav |
| 子导航 | sub\_nav | 顶部导航 | top\_nav |
| 菜单 | menu | 子菜单 | sub\_menu |

(3)一般元素命名

|  |  |  |  |
| --- | --- | --- | --- |
| **CLASS名称** | **命名** | **CLASS名称** | **命名** |
| 二级 | sub | 面包屑 | breadcrumb |
| 标志 | logo | 广告 | Bner  (禁用banner或ad) |
| 登陆 | login | 注册 | regsiter/reg |
| 搜索 | search | 加入 | join |
| 状态 | status | 按钮 | btn |
| 滚动 | scroll | 标签页 | tab |
| 文章列表 | list | 短消息 | msg/message |
| 当前的 | current | 提示小技巧 | tips |
| 图标 | icon | 注释 | note |
| 指南 | guide | 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 | 演示 | demo |
| 下拉框 | select | 摘要 | summary |
| 翻页 | pages | 主题风格 | themes |
| 设置 | set | 成功 | suc |
| 按钮 | btn | 文本 | txt |
| 颜色 | color/c | 背景 | bg |
| 边框 | border/bor | 居中 | center |
| 上 | top/t | 下 | bottom/b |
| 左 | left/l | 右 | right/r |
| 添加 | add | 删除 | del |
| 间隔 | sp | 段落 | p |
| 弹出层 | pop | 公共 | global/gb |
| 操作 | op | 密码 | pwd |
| 透明 | tran | 信息 | info |
| 重点 | hit | 预览 | pvw |
| 单行输入框 | input | 首页 | index |
| 日志 | blog | 相册 | photo |
| 留言板 | guestbook | 用户 | user |
| 确认 | confirm | 取消 | cancel |
| 报错 | error |  |  |

1. **图片命名**
2. 背景图片：bg001,bg002,bg003……
3. 一般图片：img001,img002,img003……
4. 特定图片：如banner,logo按照具体情况命名
5. 按钮图片：btn\_submit,btn\_cancel…….
6. **JavaScript部分**
   1. 在文档中引用js
7. 使用外部文件方式引用js；
8. 将引用js的代码集中放置在一起，可放置在</head>之前或</body>之前，严禁在body间分散放置；
9. 使DOM结构和js代码分离，禁止写在标记内部；
10. 如果是发布版本，请将多个稳定版本的js文件压缩、归类放置到单个文件内，**压缩和最小化js文件**，在线packer压缩工具：<http://dean.edwards.name/packer/>。
    1. 不直接操作DOM样式，使用已在样式表中定义的class来操作DOM样式
    2. DOM加载

把需要在DOM加载后立即执行的代码封装在同一个函数内执行，不要编写多个window.onload事件或jQuery的ready()事件。

* 1. 严禁使用多个library库
  2. 优化jQuery代码，提高性能：

1. 选择器从最近的ID开始继承或直接使用ID选择器：$(”#id tag”)；
2. 选择器在使用class前加上标签名：$(”span.span1”)；
3. 尽量使用ID选择器代替class；
4. 要获取子元素请使用子选择器，而不要使用后代选择器：$(”#id > span”)；
5. 缓存jQuery对象，不要在代码中重复出现相同的选择器：var btn=$(”#id”)；
6. 使用data()存储临时变量；
7. 限制直接操作DOM，在更新DOM前应该准备好需要的东西；
8. 避免使用live()方法绑定事件；
9. 在父级元素监听事件，对目标元素进行操作：

$(”#id”).click(function(e){var input=$(e.target);});

1. 推迟加载拖放、动画、视觉特效等代码，把可能会影响页面加载速度的代码绑定到$(window).load()事件中。