**规范目的**

**为提高团队协作效率, 便于后台人员添加功能及前端后期优化维护, 输出高质量的文档, 特制订此文档. 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. 本文档如有不对或者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**

**符合web标准, 语义化html, 结构表现行为分离, 兼容性优良. 页面性能方面, 代码要求简洁明了有序, 尽可能的减小服务器负载, 保证最快的解析速度.**

**文件规范**

**1. html, css, js, images文件均归档至<系统开发规范>约定的目录中;**

**2. html文件命名: 英文命名, 后缀.htm. 同时将对应界面稿放于同目录中, 若界面稿命名为中文, 请重命名与html文件同名, 以方便后端添加功能时查找对应页面;**

**3. css文件命名: 英文命名, 后缀.css. 共用base.css, 首页index.css, 其他页面依实际模块需求命名.;**

**4. Js文件命名: 英文命名, 后缀.js. 共用common.js, 其他依实际模块需求命名.**

**html书写规范**

**1. 文档类型声明及编码: 统一为html5声明类型<!DOCTYPE html>; 编码统一为<meta charset="gbk" />, 书写时利用IDE实现层次分明的缩进;**

**2. 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;**

**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>**

**4. 引入JS库文件, 文件名须包含库名称及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;**

**5. 所有编码均遵循xhtml标准, 标签 & 属性 & 属性命名 必须由小写字母及下划线数字组成, 且所有标签必须闭合, 包括 br (<br />), hr(<hr />)等; 属性值必须用双引号包括;**

**6. 充分利用无兼容性问题的html自身标签, 比如span, em, strong, optgroup, label,等等; 需要为html元素添加自定义属性的时候, 首先要考虑下有没有默认的已有的合适标签去设置, 如果没有, 可以使用须以"data-"为前缀来添加自定义属性，避免使用"data:"等其他命名方式;**

**7. 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;**

**8. 尽可能减少div嵌套, 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>;**

**9. 书写链接地址时, 必须避免重定向，例如：href="http://itaolun.com/", 即须在URL地址后面加上“/”；**

**10. 在页面中尽量避免使用style属性,即style="…";**

**11. 必须为含有描述性表单元素(input, textarea)添加label, 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>**

**12. 能以背景形式呈现的图片, 尽量写入css样式中;**

**13. 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;**

**14. 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;**

**15. 特殊符号使用: 尽可能使用代码替代: 比如 <(<) & >(&gt;) & 空格( ) & »(») 等等;**

**16. 书写页面过程中, 请考虑向后扩展性;**

**17. class & id 参见 css书写规范.**

**css书写规范**

**1. 编码统一为utf-8;**

**2. 协作开发及分工: i会根据各个模块, 同时根据页面相似程序, 事先写好大体框架文件, 分配给前端人员实现内部结构&表现&行为; 共用css文件base.css由i书写, 协作开发过程中, 每个页面请务必都要引入, 此文件包含reset及头部底部样式, 此文件不可随意修改;**

**3. class与id的使用: id是唯一的并是父级的, class是可以重复的并是子级的, 所以id仅使用在大的模块上, class可用在重复使用率高及子级中; id原则上都是由我分发框架文件时命名的, 为JavaScript预留钩子的除外;**

**4. 为JavaScript预留钩子的命名, 请以 js\_ 起始, 比如: js\_hide, js\_show;**

**5. class与id命名: 大的框架命名比如header/footer/wrapper/left/right之类的在2中由i统一命名.其他样式名称由 小写英文 & 数字 & \_ 来组合命名, 如i\_comment, fontred, width200; 避免使用中文拼音, 尽量使用简易的单词组合; 总之, 命名要语义化, 简明化.**

**6. 规避class与id命名(此条重要, 若有不明白请及时与i沟通):**

**a） 通过从属写法规避, 示例见d;  
b）取父级元素id/class命名部分命名, 示例见d;  
c）重复使用率高的命名, 请以自己代号加下划线起始, 比如i\_clear;**

**d）a,b两条, 适用于在2中已建好框架的页面, 如, 要在2中已建好框架的页面代码<div id="mainnav"></div>中加入新的div元素,**

**按a命名法则: <div id="mainnav"><div class="firstnav">...</div></div>,**

**样式写法: #mainnav .firstnav{.......}**

**按b命名法则: <div id="mainnav"><div class="main\_firstnav">...</div></div>,**

**样式写法: .main\_firstnav{.......}**

**7. css属性书写顺序, 建议遵循: 布局定位属性-->自身属性-->文本属性-->其他属性. 此条可根据自身习惯书写, 但尽量保证同类属性写在一起. 属性列举: 布局定位属性主要包括: display & list-style & position（相应的 top,right,bottom,left） ＆ float & clear ＆ visibility ＆ overflow； 自身属性主要包括: width & height & margin & padding & border & background; 文本属性主要包括：color & font & text-decoration & text-align & vertical-align & white- space & 其他 & content; 我所列出的这些属性只是最常用到的, 并不代表全部;**

**8. 书写代码前, 考虑并提高样式重复使用率;**

**9. 充分利用html自身属性及样式继承原理减少代码量, 比如:<ul class="list"><li>这儿是标题列表<span>2010-09- 15</span></ul>**

**定义**

**ul.list li{position:relative} ul.list li span{position:absolute; right:0}**

**即可实现日期居右显示**

**10. 样式表中中文字体名, 请务必转码成unicode码, 以避免编码错误时乱码;**

**11. 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按模块制作;**

**12. 使用table标签时(尽量避免使用table标签), 请不要用width/ height/cellspacing/cellpadding等table属性直接定义表现, 应尽可能的利用table自身私有属性分离结构与表现 , 如thead,tr,th,td,tbody,tfoot,colgroup,scope; (cellspaing及cellpadding的css控制方法:table{border:0;margin:0;border-collapse:collapse;} table th, table td{padding:0;}, base.css文件中我会初始化表格样式)**

**13. 杜绝使用<meta http-equiv="X-UA-Compatible" content="IE=7" /> 兼容 ie8;**

**14. 用png图片做图片时, 要求图片格式为png-8格式,若png-8实在影响图片质量或其中有半透明效果, 请为ie6单独定义背景:\_background:none;\_filter:progid:DXImageTransform.Microsoft.AlphaImageLoader (sizingMethod=crop, src=’img/bg.png’);**

**15. 避免兼容性属性的使用, 比如text-shadow || css3的相关属性;**

**16. 减少使用影响性能的属性, 比如position:absolute || float ;**

**17. 必须为大区块样式添加注释, 小区块适量注释;**

**18. 代码缩进与格式: 建议单行书写, 可根据自身习惯, 后期优化i会统一处理;**

**JavaScript书写规范**

**1. 文件编码统一为utf-8, 书写过程过, 每行代码结束必须有分号; 原则上所有功能均根据XXX项目需求原生开发, 以避免网上down下来的代码造成的代码污染(沉冗代码 || 与现有代码冲突 || ...);**

**2. 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;**

**3. 变量命名: 驼峰式命名. 原生JavaScript变量要求是纯英文字母, 首字母须小写, 如iTaoLun;**

**jQuery变量要求首字符为'\_', 其他与原生JavaScript 规则相同, 如: \_iTaoLun;另, 要求变量集中声明, 避免全局变量.**

**4. 类命名: 首字母大写, 驼峰式命名. 如 ITaoLun;**

**5. 函数命名: 首字母小写驼峰式命名. 如iTaoLun();**

**6. 命名语义化, 尽可能利用英文单词或其缩写;**

**7. 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\_r() & innerText;**

**8. 后期优化中, JavaScript非注释类中文字符须转换成unicode编码使用, 以避免编码错误时乱码显示;**

**9. 代码结构明了, 加适量注释. 提高函数重用率;**

**10. 注重与html分离, 减小reflow, 注重性能.**

**图片规范**

**1. 所有页面元素类图片均放入img文件夹, 测试用图片放于img/demoimg文件夹;**

**2. 图片格式仅限于gif || png || jpg;**

**3. 命名全部用小写英文字母 || 数字 || \_ 的组合，其中不得包含汉字 || 空格 || 特殊字符；尽量用易懂的词汇, 便于团队其他成员理解; 另, 命名分头尾两部分, 用下划线隔开, 比如ad\_left01.gif || btn\_submit.gif;**

**4. 在保证视觉效果的情况下选择最小的图片格式与图片质量, 以减少加载时间;**

**5. 尽量避免使用半透明的png图片(若使用, 请参考css规范相关说明);**

**6. 运用css sprite技术集中小的背景图或图标, 减小页面http请求, 但注意, 请务必在对应的sprite psd源图中划参考线, 并保存至img目录下.**

**注释规范**

**1. html注释: 注释格式 <!--这儿是注释-->, '--'只能在注释的始末位置,不可置入注释文字区域;**

**2. css注释: 注释格式 ;**

**3. JavaScript注释, 单行注释使用'//这儿是单行注释' ,多行注释使用 ;**

**开发及测试工具约定**

**建议使用Aptana || Dw || Vim , 亦可根据自己喜好选择, 但须遵循如下原则:**

**1. 不可利用IDE的视图模式'画'代码;**

**2. 不可利用IDE生成相关功能代码, 比如Dw内置的一些功能js;**

**3. 编码必须格式化, 比如缩进;**

**测试工具: 前期开发仅测试FireFox & IE6 & IE7 & IE8 , 后期优化时加入Opera & Chrome & Safari;**

**建议测试顺序: FireFox-->IE7-->IE8-->IE6-->Opera-->Chrome-->Safari, 建议安装firebug及IE Tab Plus插件.**

**其他规范**

**1. 开发过程中严格按分工完成页面, 以提高css复用率, 避免重复开发;**

**2. 减小沉冗代码, 书写所有人都可以看的懂的代码. 简洁易懂是一种美德. 为用户着想, 为服务器着想.**