

Web 前端开发规范

参考手册

一、规范目的

1.1 概述

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，经讨论决定后可以更改。

1.2 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。

二、规范细则

2.1 文件命名规则

文件名称统一用小写的英文字母、数字和下划线的组合，其中不得包含汉字、空格和特殊字符。命名原则的指导思想：

- 一是使得工作组的每一个成员能够方便的理解每一个文件的意义。
- 二是当在文件夹中使用“按名称排列”的命令时，同一种大类的文件能够排列在一起，以便查找、修改、替换、计算负载量等等操作。

1. HTML 命名原则

- 引导文件统一使用 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。例如：
 - 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。

2. 图片命名原则

- 图片的名称分为头尾两部分，用下划线隔开，头部分表示此图片的大类性质。例如：广告、标志、菜单、按钮等。
- 放置在页面顶部的广告、装饰图案等长方形的图片取名：banner。
- 标志性的图片取名：logo。
- 在页面上位置不固定并且带有链接的小图片取名：button。
- 在页面上某一个位置连续出现，性质相同的链接栏目的图片取名：menu。
- 装饰用的照片取名：pic。
- 不带链接表示标题的图片取名：title。

例如：

banner_sohu.gif

banner_sina.gif

menu_aboutus.gif

menu_job.gif

title_news.gif

logo_police.gif

logo_national.gif

pic_people.jpg

- 鼠标感应效果图片命名规范为"图片名+_on/off"。

例如：

menu1_on.gif

menu1_off.gif

3. Javascript 命名原则

例如：

- 广告条的 javascript 文件名为 ad.js
- 弹出窗口的 javascript 文件名为 pop.js

4. 动态语言文件命名原则

以性质_描述，描述可以有多个单词，用“_”隔开，性质一般是该页面得概要。例如：

register_form.asp

register_post.asp

topic_lock.asp

2.2 文件存放位置规范

HTML、CSS、js、images 文件均归档至<系统开发规范>约定的目录中。

_Root		
	cn	存放中文 HTML 文件
	en	存放英文 HTML 文件
	flash	存放 Flash 文件
	images	存放图片文件
	imagestudio	存放 PSD 源文件

	flashstudio	存放 flash 源文件
	inc	存放 include 文件
	library	存放 DW 库文件
	media	存放多媒体文件
	project	存放工程项目资料
	temp	存放客户原始资料
	js	存放 JavaScript 脚本
	css	存放 CSS 文件

2.3 CSS 书写规范

CSS 书写规范主要目的：提高效率方便相关人员修改，提高网站的扩展性(根据网站情况不一样，要灵活应用)。

1. 基本原则

CSS 样式可细分为 3 类：自定义样式、重新定义 HTML 样式、链接状态样式。

- 自定义样式：为设计师自定义的新 CSS 样式，影响使用本样式的区域，用于完成网页中局部的样式设定。样式名：“.”+“相应样式效果描述的单词或缩写”。例如：
“ .shadow ”

文字样式名：“.no”+“字号”+“行距”+“颜色缩写”。例如：“ .no12 ”、“ .no12-24 ”。

- 重新定义 HTML 样式：为设计师重新定义已有的 HTML 标签样式，影响全部的被设定标签样式，用于统一网页中某一标签的样式定义。样式名：“HTML 标签”。
例如：hr { border: 1px dotted #333333 }。
- 链接状态样式：为设计师对链接不同状态设定特殊样式，影响被使用本样式区域中的链接。

该样式写法有 2 种： a.nav:link、nav.a:link。第一种只能修饰<a>标签中；第二种可以修饰所有包含有<a>标签的其他标签。

页面内的样式加载必须用链接方式：

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

2. 应用说明

- 编码统一为 utf-8，都用小写。
- class 与 id 的使用：一般都使用 class。特殊除外，因 id 的优先级比 class 的高。
- 样式命名推荐使用英文，避免使用汉语拼音，尽量使用简易的单词组合。命名方式采用驼峰命名法和划线命名法两种，提高可读性。例如：

dropMenu、subNavMenu、drop-menu、sub_nav_menu

驼峰命名法用来区别不同的单词。

划线命名法表明从属关系。例如：“.timeList”和“.time_list”分别表示时间列表和时间部分下的列表。

- 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。
此条可根据习惯书写，但尽量保证同类属性写在一起。

- a) 定位属性（比如：display, position, float, clear, visibility, table-layout 等）
- b) 自身属性（比如：width, height, margin, padding, border 等）
- c) 文本属性（比如：font, line-height, text-align, text-indent, vertical-align 等）
- d) 其他属性（比如：color, background, opacity, cursor, content, list-style, quotes 等）

- 避免滥用自选择器。#test span { } 和 #test .span { } 尽量选择后者。采用继承属性还是新加 class，根据不同情况灵活运用。新加 class 便于扩展维护。

继承属性示例：<ul class="list">这儿是标题列表2010-09-15

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

即可实现日期居右显示

- 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。
- 使用 table 标签时，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。

例如：

thead, tr, th, td, tbody, tfoot, colgroup, scope;

cellspacing 以及 cellpadding 的 CSS 控制方法：

table{border:0;margin:0;border-collapse:collapse;} table th, table td{padding:0;}

base.css 文件中会初始化表格样式。

- 尽量不使用<meta http-equiv="X-UA-Compatible" content="IE=7" /> 兼容 ie8，和代码*html 写法。
- 用 png 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，请参考文档“IE6 下 png 透明问题解决的最佳方案”。
- 兼容性属性的使用，如 text-shadow（文字阴影）、CSS3 的相关属性在保证目前主流浏览器使用正常的情况下，可以使用来对页面进行锦上添花式的设计。
- 减少使用影响性能的属性，如 position:absolute || float。
- 必须为大区块样式添加注释，小区块适量注释。
- 代码缩进与格式：单行书写，统一使用 tab 进行缩进。

3. 注意细则

- 1) 协作开发及分工：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 由一人书写，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。
- 2) 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 原则上都是由分发框架文件时命名的，为 JavaScript 预留钩子的除外。
- 3) 为 JavaScript 预留钩子的命名，请以 js_ 起始，例如：js_hide、js_show。
- 4) class 与 id 命名：大的框架命名，如 header、footer、wrapper、left、right 之类的由负责人统一命名。其他样式名称由“小写英文 & 数字 & _”来组合命名。例如：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。
- 5) 规避 class 与 id 命名（此条重要，若有不明白请及时与负责人沟通）：

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 和 b 两条，适用于已建好框架的页面。例如，要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{.....}

6) 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：margin、padding、float（包括 clear）、position（相应的 top、right、bottom、left）、display、visibility、overflow 等。

自身属性主要包括：width、height、background、border。

文本属性主要包括：font、color、text-align、text-decoration、text-indent 等。

其他属性包括：list-style(列表样式)、vertical-align、cursor、z-index(层叠顺序)、zoom 等。

上面所列出的这些属性只是最常用到的，并不代表全部。

7) 书写代码前，考虑并提高样式重复使用率。

8) 充分利用 HTML 自身属性及样式继承原理减少代码量，例如：

```
<ul class="list"><li>这儿是标题列表<span>2010-09-15</span></li></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。

9) 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。

10) 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。

11) 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，例如：

```
thead,tr,th,td,tbody,tfoot,colgroup,scope;
```

cellspacing 及 cellpadding 的 CSS 控制方法：

```
table{border:0;margin:0;border-collapse:collapse;}
```

```
table th, table td{padding:0;}
```

base.css 文件中会初始化表格样式。

12) 杜绝使用<meta http-equiv="X-UA-Compatible" content="IE=7" /> 兼容 ie8。

13) 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');
```

- 14) 避免兼容性属性的使用，如 `text-shadow` || CSS3 的相关属性。
- 15) 减少使用影响性能的属性，比如 `position:absolute` || `float`。
- 16) 必须为大区块样式添加注释，小区块适量注释。
- 17) 代码缩进与格式：建议单行书写，可根据自身习惯，后期优化会统一处理。

4. 命名规则

1) 主体结构

头：header
内容：content/container
尾：footer
导航：nav
侧栏：sidebar
栏目：column
页面外围控制整体布局宽度：wrapper
左右中：left right center
登录条：loginbar
标志：logo
广告：banner
页面主体：main
热点：hot
新闻：news
下载：download
子导航：subnav
菜单：menu
子菜单：submenu
搜索：search
友情链接：friendlink
页脚：footer
版权：copyright
滚动：scroll
内容：content
标签页：tab
文章列表：list
提示信息：msg
小技巧：tips
栏目标题：title
加入：joinus
指南：guild
服务：service
注册：regsiter
状态：status
投票：vote
合作伙伴：partner

2) 注释的写法

```
/* Footer */  
内容区  
/* End Footer */
```

3) id 的命名

容器: container
页头: header
内容: content/container
页面主体: main
页尾: footer
导航: nav
侧栏: sidebar
栏目: column
页面外围控制整体布局宽度: wrapper
左右中: left right center

4) 导航

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: title
摘要: summary

5) 功能

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\

4. 基本样式

```
/* CSS Document */  
body {margin:0; padding:0; font:12px "5B8B\4F53",san-serif;background:#fff;}  
div,dl,dt,dd,ul,ol,li,h1,h2,h3,h4,h5,h6,pre,form,fieldset,input,textarea,blockquote,p {padding:0;  
margin:0;}  
table,td,tr,th {font-size:12px;}  
li {list-style-type:none;}  
img {vertical-align:top;border:0;}  
ol,ul {list-style:none;}  
h1,h2,h3,h4,h5,h6 {font-size:12px; font-weight:normal;}  
address,cite,code,em,th {font-weight:normal; font-style:normal;}  
.fb {font-weight:bold;}  
.fl2px {font-size:12px;}  
.fl4px {font-size:14px;}  
.left {float:left;}  
.right {float:right;}  
  
a {color:#2b2b2b; text-decoration:none;}  
a:visited {text-decoration:none;}  
a:hover {color:#ba2636;text-decoration:underline;}  
a:active {color:#ba2636;}
```

重定义的最先，伪类其次，自定义最后，便于自己和他人阅读。

不同浏览器上字号保持一致，字号建议用点数 pt 和像素 px 来定义，pt 一般使用中文宋体的 9pt 和 11pt，px 一般使用中文宋体 12px 和 14.7px 这是经过优化的字号，黑体字或者宋体字加粗时，一般选用 11pt 和 14.7px 的字号比较合适。中英文混排时，尽可能的将英文和数字定义为 verdana 和 arial 两种字体。

2.4 HTML 书写规范

1. 基本原则

- 文档类型声明统一为：


```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

编码统一为:

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

目的: 统一性和网站提高开发合作效率。

- 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</body>之前。目的: 网站的优化。
- 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 (
), hr(<hr />)等。属性值必须用双引号包括。目的: 更加符合 web 标准(w3c), 也有利于 SEO。
- 充分利用无兼容性问题的 HTML 自身标签, 例如: span、em、strong、optgroup、label 等。目的: 减少代码量。
- 语义化 HTML, 例如: 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。目的: 减少代码量, 也有利于 SEO。
- 尽可能减少 div 嵌套, 例如:

```
<div class="box"& gt;<div class="welcome">欢迎访问 XXX, 您的用户名是<div class="
name" >用户名</div></div></div>
```

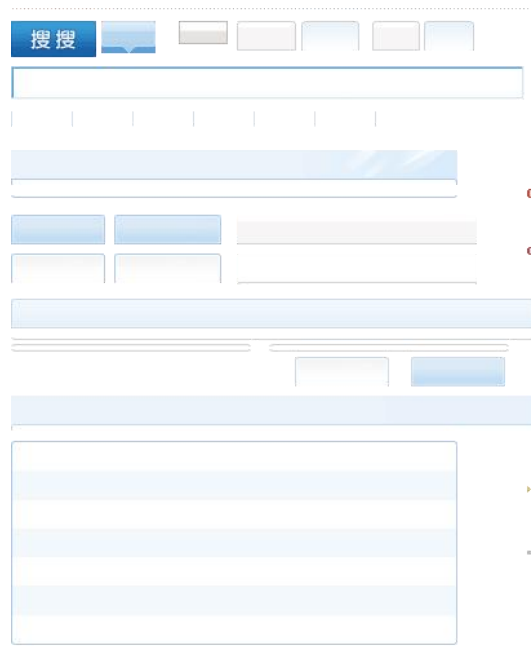
完全可以用以下代码替代:

```
<div class=" box" & gt;<p>欢迎访问 XXX, 您的用户名是<span>用户名</span></p><
/div>
```

目的: 减少代码量, 也有利于 SEO。

- 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, 例如: jquery-1.4.1.min.js。引入插件, 文件名格式为: 库名称+插件名称, 例如: jQuery.cookie.js。
- 书写链接地址时, 必须避免重定向, 例如: href="http://www.haogu.com/", 即须在 URL 地址后面加上"/"。
- 在页面中尽量避免使用 style 属性, 即 style="..."。除非考虑网站 HTTP 请求, 网站响应速度等因素比重情况具体分析。
- 严格区分作为内容的图片和作为背景的图片。作为背景的图片采用 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 增加了耦合度, 也增加了维护成本。如果图片有修改, 不要删除已添加的图片, 在空白处新增修改后的图片, 减少修改风险。

例如: 下图为腾讯的 CSS sprite 背景图。



- 图片必须加上 alt 属性，给重要的元素和截断的元素加上 title。目的：有利于 SEO，搜索引擎的爬虫。
- 给区块代码及重要功能（如循环）加上注释，方便后台程序员嵌套模版。
- 特殊符号使用：尽可能使用代码替代，如 <(<) & >(>) & 空格() & »(») 等。
- 书写页面过程中，请考虑向后扩展性。

2. head 区代码规范

head 区是指 HTML 代码的<head>和</head>之间的内容。

第一，必须加入的标签：

- 公司版权注释：

```
<!-- The site is designed by EHM,Inc 07/2015 --->
```

- 网页显示字符集：

■ 简体中文

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=gb2312">
```

■ 繁体中文

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
```

■ 英语

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
```

- 网页制作者信息：

```
<META name="author" content="webmaster@maketown.com">
```

- 网站简介：

```
<META NAME="DESCRIPTION" CONTENT="xxxxxxxxxxxxxxxxxxxxxxxxxxxx">
```

- 搜索关键字

```
<META NAME="keywords" CONTENT="xxxx,xxxx,xxx,xxxxx,xxxx,">
```

- 网页的 CSS 规范

```
<LINK href="../css/style.css" rel="stylesheet" type="text/css">
```

- 网页标题

```
<title>xxxxxxxxxxxxxxxxxxxx</title>
```

第二，可以选择加入的标签

- 设定网页的到期时间。一旦网页过期，必须到服务器上重新调阅。

<META HTTP-EQUIV="expires" CONTENT="Wed, 26 Feb 2015 08: 21: 57 GMT">

- 禁止浏览器从本地机的缓存中调阅页面内容。

<META HTTP-EQUIV="Pragma" CONTENT="no-cache">

- 用来防止别人在框架里调用你的页面。

<META HTTP-EQUIV="Window-target" CONTENT="_top">

- 自动跳转。

<META HTTP-EQUIV="Refresh" CONTENT="5;URL=http://www.yahoo.com">

指时间停留 5 秒

- 网页搜索机器人向导。用来告诉搜索机器人哪些页面需要索引，哪些页面不需要索引。

<META NAME="robots" CONTENT="none">

CONTENT 的参数有 all、none、index、noindex、follow、nofollow。默认是 all。

- 收藏夹图标

<link rel="Shortcut Icon" href="favicon.ico">

- 所有的 javascript 的调用尽量采取外部调用。

<SCRIPT LANGUAGE="JavaScript" SRC="script/xxxxx.js"></SCRIPT>

3. 字体

- 在设定字体样式时，对于文字字号样式和行间距应必须使用 CSS 样式表。禁止在页面中出现 标记。
- 在网页中中文应首选使用宋体。英文和数字首选使用 verdana 和 arial 两种字体。一般使用中文宋体的 9pt 和 11pt 或 12px 和 14.7px 这是经过优化的字号，黑体字或者宋体字加粗时，一般选用 11pt 和 14.7px 的字号比较合适。
- 为了最大程度的发挥浏览器自动排版的功能，在一段完整的文字中请尽量不要使用
 来人工干预分段。
- 不同语种的文字之间应该有一个半角空格，但避头的符号之前和避尾的符号之后除外，汉字之间的标点要用全角标点，英文字母和数字周围的括号应该使用半角括号。
- 请不要在网页中连续出现多于一个的 也尽量少使用全角空格（英文字符集下，全角空格会变成乱码），空白应该尽量使用 text-indent、padding、margin、hspace、vspace 以及透明的 gif 图片来实现。
- 行距建议用百分比来定义，常用的两个行距的值：line-height:120%/150%。
- 排版中经常会遇到需要进行首行缩进的处理，不要使用 或者全角空格来达到效果，规范的做法是在样式表中定义：p { text-indent: 2em; } 然后给每一段加上 <p> 标记，注意，一般情况下，请不要省略 </p> 结束标记 。

4. 链接

- 网站中的链接路径全部采用相对路径，一般链接到某一目录下的缺省文件的链接路径不必写全名，如不必这样：，而应该这样：，所有内页指向首页的链接写成。
- 在浏览器里，当点击空链接时，它会自动将当前页面重置到首端，从而影响用户正常的阅读内容，因此可以用代码“javascript:void(null)”代替原来的“#”标记。

5. 表格

在写 `<table>` 互相嵌套时, 严格按照的规范, 对于单独的一个`<table>`来说, `<table><tr>` 对齐, `<td>` 缩进两个半角空格, `<td>` 中如果还有嵌套的表格, `<table>`也缩进两个半角空格, 如果`<td>`中没有任何嵌套的表格, `</td>` 结束标记应该与 `<td>` 处于同一行, 不要换行,

在源代码中不应有这样的代码:

```
<td>
</td>
```

而应该是这样的:

```
<td></td>
```

这是因为浏览器认为换行相当于一个半角空格, 以上不规范的写法相当于无意中增加一个半角空格, 如果确实有必要增加一个半角空格, 也应该这样写:

```
<td> </td>
```

一个网页要尽量避免用整个一张大表格, 所有的内容都嵌套在这个大表格之内, 因为浏览器在解释页面的元素时, 是以表格为单位逐一显示, 如果一张网页是嵌套在一个大表格之内, 那么很可能造成的后果就是, 当浏览者敲入网址, 他要先面对一片空白很长时间, 然后所有的网页内容同时出现。如果必须这样做, 请使用 `<tbody>`标记, 以便能够使这个大表格分块显示

6. 下载速度

首页 Flash 网页大小应限定在 200K 以下, 尽可能的使用矢量图形和 Action 来减小动画大小。非首页静态页面含图片大小应限定在 70K 左右, 尽可能的使用背景颜色替换大块同色图片。

7. include

asp 标准写法 `<!--#include file="inc/index_top.asp" -->`

jsp 标准写法 `<%@ include file="../inc/index_top.jsp" %>`

8. Alt 和 Title

都是提示性语言标签, 注意它们之间的区别。

在我们浏览网页时, 当鼠标停留在图片对象或文字链接上时, 在鼠标的右下角有时会出一个提示信息框。对目标进行一定的注释说明。在一些场合, 它的作用是很重要的。

alt 用来给图片来提示的。Title 用来给链接文字或普通文字提示的。

用法如下:

```
<p Title="给链接文字提示">文字</p>
```

```
<a href="#" Title="给链接文字提示">文字</a>
```

```

```

9. 缓存

网页不会被缓存。

HTM 网页

```
<META HTTP-EQUIV="pragma" CONTENT="no-cache">
```

```
<META HTTP-EQUIV="Cache-Control" CONTENT="no-cache, must-revalidate">
```

```
<META HTTP-EQUIV="expires" CONTENT="0">
```

ASP 网页

```
Response.Expires = -1
```

```
Response.ExpiresAbsolute = Now() - 1
```

```
Response.cachecontrol = "no-cache"
```

10. 浏览器兼容性

创建站点时，应该明白访问者可能使用各种 Web 浏览器。在已知的其他设计限制下，尽可能将站点设计为具有最大的浏览器兼容性。

目前，使用的 Web 浏览器有二十多种，大多数已发行了多个版本。即使只针对使用 Microsoft Internet Explorer 的大多数 Web 用户，但也应明确并不是每个人都在使用这种浏览器的最新版本。

站点越复杂（在布局、动画、多媒体内容和交互方面），跨浏览器兼容的可能性就越小。例如，并非所有的浏览器都可以运行 JavaScript。不使用特殊字符的纯文本页面或许能够在任何浏览器中正确显示，但比起有效地使用图形、布局和交互的页面，这样的页面在美感上可能要差得多。所以，应尽量在最佳效果设计和最大浏览器兼容性设计之间取得平衡。

所有的 HTML 标签的属性都要用单引号或者双引号括起，即们应该写 `` 而不是 ``。

11. banner

全尺寸 banner 为 468X60px，半尺寸 banner 为 234X60px，小 banner 为 88X31px。

另外 120X90，120X60 也是小图标的标准尺寸。全尺寸 banner 不超过 14K。

普遍的 banner 尺寸 760X100、750X120、468X60、468X95、728X90、585X140。

次级页的 pic 尺寸 360X300、336X280。

游标：100X100 或 120X120。

12. LOGO 的国际标准规范

为了便于信息传播，一个统一的国际标准是需要的。实际上已经有了这样的一整套标准。其中关于网站的 LOGO，目前有三种规格：

- 88*31 这是互联网上最普遍的 LOGO 规格。
- 120*60 这种规格用于一般大小的 LOGO。
- 120*90 这种规格用于大型 LOGO。

13. 页面修饰图片处理

图片经过优化以加快下载的速度，有较佳的视觉空间效果，用图要与页面风格、页面内容相符；制作精美，细节处理得当。

2.5 JavaScript 书写规范

- 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。要检查网上 down 下来的代码造成的代码污染（沉冗冲突，代码注入...）等。
- 书写过程中，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 || ...)。
- 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。
- 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。

- 类命名：首字母大写，驼峰式命名。如 ITaoLun。
- 函数命名：首字母小写驼峰式命名。如 iTaoLun()。
- 命名语义化，尽可能利用英文单词或其缩写。
- 尽量避免使用存在兼容性及消耗资源的方法或属性，如 eval()、innerText。
- 在后期优化中，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。
- 代码结构明了，加适量注释。提高函数重用率。
- 注重与 HTML 分离，减小 reflow，注重性能。

2.6 图片规范

- 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。
- 图片格式仅限于 gif、png、jpg。
- 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，如 ad_left01.gif、btn_submit.gif。
- 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。
- 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）。
- 运用 CSS sprite 技术集中小的背景图或图标，减小页面 http 请求。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。

2.7 注释规范

- 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。
- CSS 注释：注释格式 /*这儿是注释*/。
- JavaScript 注释：单行注释使用//这儿是单行注释；多行注释使用 /* 这儿有多行注释 */。

2.8 开发及测试工具约定

- 编码必须格式化，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 等。
- 建议测试顺序：FireFox ->IE7 ->IE8 ->IE6 ->Opera ->Chrome，建议安装 firebug 及 IE Tab Plus 插件，安装调试利器 IETester。

2.9 其他规范

- 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。

- 减小冗余代码，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。

2.10 浏览器兼容性 CSS hack

即便是完美的 CSS 也未必能在目前众多的终端浏览器中呈现一致的效果，所以，CSS Hack 在很多情况下都是必要的，建议先以对 CSS 标准支持得比较好的浏览器（如 Firefox 或者 Chrome）为主编辑 CSS，最后再处理 IE 的兼容性——单独为 IE 建立一个 CSS 文件（如 for-ie.css, fuck-ie.css, ie-hack.css 等），最后在 HTML 文件中，通过 IE 的条件注释按需引用。

1. 标识区别

区别 IE6、IE7、IE8、FF。

- IE 都能识别*，标准浏览器(如 FF)不能识别*。
- IE6 能识别*，但不能识别 !important，IE6 在样式前面加_。
- IE7 能识别*，也能识别!important。
- IE8 能识别\9，例如：background:red \9。
- firefox 不能识别*，但能识别!important。
- IE6 和 firefox 的区别：

```
background:orange;*background:blue;
```

意思就是火狐浏览器的背景颜色是橙色,而 IE 浏览器的背景色是蓝色。

- IE6 和 IE7 的区别：

```
background:green !important;background:blue;
```

意思指的是:IE7 的背景颜色是绿色,IE6 的背景颜色是蓝色。

- IE7 和 FF 的区别：

```
background:orange; *background:green;
```

意思指的是:火狐浏览器的背景颜色是橙色,而 IE7 的背景颜色是绿色。

- FF、IE7、IE6 的区别：

```
background:orange;
```

```
*background:green !important;
```

```
*background:blue;
```

意思是火狐浏览器的背景橙色,IE7 浏览器的背景颜色是绿色,而 IE6 浏览器的颜色是蓝色。

2. 实践建议

- 开发平台的选择

在 Firefox 上编写 CSS，同时兼容其他浏览器的。这样做肯定会比在 IE 做好再到别的浏览器兼容来得容易，因为 IE 对老标准支持还是很不错的，而 IE 的一些特有功能却不支持。所以推荐以 Firefox 结合 Firebug 扩展作为平台。

- CSS Hack 的顺序

使用 Firefox 作为平台，只要代码写得够标准，其实要 Hack 的地方不会很多的，IE 以外的浏览器几乎都不会有问题，所以可以暂时忽略。

顺序如下：Firefox -> IE6 -> IE7 -> 其他。

- Hack 的方法

两种方法：一种是在不同文件中处理，另一种则是在同一个文件中处理。其实作用是相同的，只是出发点不一样而已。

第一，同一文件中处理

如 id="bgcolor"的控件要在 IE6 中显示蓝色，IE7 中显示绿色，Firefox 等其他浏览器中显示红色。

```
1 #bgcolor {
2     background:red !important; /* Firefox 等其他浏览器 */
3     background:blue; /* IE6 */
4 }
5 *+html #bgcolor {
6     background:green !important; /* IE7 */
7 }
```

IE6 不认 !important,也不认 *+html.所以 IE6 只能是 blue。

IE7 认 !important,也认 *+html,优先级: (*+html + !important) > !important > +html. IE7 可以是 red, blue 和 green,但 green 的优先级最高。

Firefox 和其他浏览器都认 !important. !important 优先,Firefox 可以是 red 和 blue,但 red 优先度高.上述的优先符号均是 CSS3 标准允许的,其他浏览器也还有其他的 Hack 方法,但我迄今还没遇到过 Firefox 正常, IE 以外的其他浏览器不正常的情况,所以无可分享.只要代码规范,相信这种情况的发生应该是很罕见 (JavaScript 除外)。

第二，不同文件中处理

为什么同一文件中可以处理还要写在多个文件里面针对不同的浏览器?这是为了欺骗 W3C 的验证工具。其实只需要两个文件，一个是针对所有浏览器的，一个只为 IE 服务。将所有符合 W3C 的代码写到一个里面去，而一些 IE 中必须的，又不能通过 W3C 验证的代码 (如 cursor:hand;)放到另一个文件中,再用下面的方法导入。

```
1 <!-- 放置所有浏览器的样式 -->
2 <link rel="stylesheet" href="style.css" type="text/css" />
3 <!-- 只放置 IE 必须的，而不能通过 W3C 的代码 -->
4 <!--[if IE]>
5     <link rel="stylesheet" href="style_ie.css" type="text/css" />
6 <![endif]-->
```