



慧聪网 WEB 前端标准规范

CSS 样式命名与编码标准规范

2011 年 2 月

慧聪网技术中心 MMT 开发部 WEB 前端研发组

目录

- 一、概述.....3
- 二、受众.....3
- 三、CSS 文件命名规范.....4
 - 1. 全局页面 CSS 文件命名：.....4
 - 2. 子页面 CSS 文件命名规则：.....4
 - 3. CSS 引用资源命名规则：.....5
- 四、CSS 类/ID 命名规范.....5
- 五、样式属性书写规范.....8
- 六、CSS 编码规范及方法.....10
 - 1. 基本规则.....10
 - 2. CSS 编码规范.....11
 - 选择 DOCTYPE.....11
 - 指定语言及字符集：.....11
 - 调用样式表：.....12
 - 选用恰当的元素：.....12
 - 派生选择器：.....13
 - 辅助图片用背影图处理：.....13
 - 结构与样式分离：.....13
 - 文档的结构化书写：.....13
 - 鼠标手势：.....14
 - 3. 注释书写规范.....14
 - 行间注释：.....14
 - 整段注释：.....14
 - 4. 样式属性代码缩写.....15
 - 不同类有相同属性及属性值的缩写：.....15
 - 同一属性的缩写：.....16
 - 内外侧边框的缩写：.....16
 - 颜色值的缩写：.....18
 - 5. hack 书写规范.....18
- 七、附录一.....21
 - 1. XHTML 校验.....21
 - 2. CSS2 校验.....22

一、概述

CSS 样式表在控制网页表现效果上有着重要的作用。本文档规定 CSS 样式表的基础标准。内容包括级联样式表的语法、各种术语以及使用规范等。本 CSS 规范主要实现以下目的：

- ✧ 缩减页面代码，提高页面浏览速度,缩减带宽成本;
- ✧ 结构清晰，容易被搜索引擎搜索到，天生优化了 seo
- ✧ 提高易用性。使用 CSS 结构化 HTML。
- ✧ 有效控制页面布局。
- ✧ 表现和内容相分离。
- ✧ 方便搜索引擎的搜索。。
- ✧ 增强浏览器兼容性。

在产品开发、维护时请遵循本文档制定的规范开发、编辑相关 CSS 文件。

本文档涉及的规范主要包含三个方面：XXXX。

二、受众

网页制作师(制作)、JS 工程师(交互)、。

三、 CSS 文件命名规范

名称全部用小写英文字母、数字、下划线的组合。

其中不得包含汉字、空格和特殊字符；目录名应以英文、拼音为主（不到万不得已不要以拼音作为目录名称，经验证明，用拼音命名的目录往往连一个月后的自己都看不懂）。尽量用一些大家都能看懂的词汇。使得你自己和工作组的每一个成员能够方便的理解每一个文件的意义。而且当我们在文件夹中使用“按名称排列”的命令时，同一种大类的文件能够排列在一起，以便我们查找、修改、替换、计算负载量等等操作。

1. 全局页面 CSS 文件命名：

Base 或 Basic - 基本样式。

Layout - 全部页面布局。除非改变布局否则不修改此文件。

General - 颜色，风格，排版，非结构的布局。本文件可据需要经常编辑。

Forms - 表单的布局和风格。如果需要可以为不同的页面而分别制作表单的 css 文件。

Index 或 Home - 用来控制主页而不考虑其他页面的样式。

Print - 基本的打印样式。

Themes - 主题样式表。

Master - 主要样式。

Font - 字体样式。

2. 子页面 CSS 文件命名规则：

文件名格式为 **module(业务模块).page(页面类型)[.subpage(子页面类型)].css**

如下示例：

my_default.css 商务中心首页

my_compass.css 黄金罗盘页面

my_goldshopkeeper.css 金牌掌柜页面

3. CSS 引用资源命名规则：

资源文件名格式为：(性质)_(描素)_(位置)_(分类)_(数量)项相结合，采用简写、组合的方式形成通用规则：

news (性质)

news_title (性质_描素)

news_title_top (性质_描素_位置)

news_title_top_01 (性质_描素_位置_数量)

news_title_top_a_01 (性质_描素_位置_分类_数量)

news_title_top_b_01 (性质_描素_位置_分类_数量)

例如：news_title_top.gif、news_title_top_b_01.jpg。

四、CSS 类/ID 命名规范

CSS 和其他程序一样，都是有作用域这个概念，有全局、类局部作用这些方式。

以下示例：

```
p{background:#f00;}/* 作用域：全局 */  
.div p{color:#000;}/* 作用域：div 类中*/
```

Css 几种编写方式和权重对比

1. 标签：权值为 0,0,0,1
2. 类：权值为 0,0,1,0
3. 属性选择：权值为 0,0,1,1
4. ID：权值为 0,1,0,0
5. important 的权值为最高 1,0,0,0

选择器要避免当状态改变时名称失去意义，最常见的就是用于布局的类名，如“left”、“right”，当左边栏不再是左边栏的时候，“left”这个名就没有实际意义了。这与我们所推荐的“命名要有意义”就相违背了，使用序号就更加有问题了。我们要避免遇到的情况就是当状态（表现）改变时，对应定义的类名不会失去意义。

所谓的状态（表现）改变，有几种情况：

1. *HTML 不变，样式定义改变。如果命名使用了表示某一状态的名称，如“red”、“font14”等，必定会引起定义与命名不符的情况，对后继的影响会造成比较大的影响。*
2. *样式定义不变，HTML 改变。HTML 改变意味着类名可更换，也就是如果类名使用了表示某一状态的名称，反而更有利于修改。*
3. *样式定义与 HTML 都改变。只需要考虑不要出现第一种情况的结果就可以了。*
 - a) *而实际情况并不是单纯的某一种情况，更多的时候是混杂着出现的。*

规则

[模块前缀]_ 类型 _ (作用 | 状态) n _ [位置 n]

(必选)：必需存在。

[可选]：可根据需要选择。

|：多选一。

n：可有多。

名词说明

模块前缀：模块定义时使用的前缀。

类型：定义类的内容类型。如输入框、文本、段落等等。

作用：定义类的作用，用于对类型的补充。

状态：定义类的状态，用于对类型的补充。

位置：定义类所使用的位置，如首页、导航等等，不排除使用左、右这样的词，但应尽量避免。

- ✧ 每项都可为自己的一个缩写表，同一名称的缩写尽量统一。
- ✧ 所选用的单词应选择不过于具体表示某一状态（如颜色、大小等）的单词，以避免当状态改变时名称失去意义。
- ✧ 由不以数字开头的小写字母（a-z）、数字（0-9）组成。
- ✧ 确保类(.class)的重用性与对象(#id)的唯一性，id 避免使用保留字。

模块前缀:	类型:	作用:	状态:	位置:
弹出 pop	按钮 bt	设置 set	成功 suc	公共 gb
公共 global, gb	文本 tx	添加 add	失败 lost	边框 bor
标题 title, tit	段落 p	删除 del	透明 tran	段落 p
提示 hint	图标 icon	操作 op		弹出 pop
菜单 menu	input input	密码 pw		标题 title, tit
信息 info	颜色 color, c	导入 inc		菜单 menu
预览 pvw	背景 bg			内容 cont
Tips tips	边框 bor			导航 nav
导航 nav				

以下示例：

中文解释	命名
文本输入框	.input_tx
密码输入框	.input_pw
登录密码输入框	.input_pw_login
文本颜色	.c_tx
段落文本颜色	.c_tx_p
相册弹出的设置层	.pop_set_photo
日志设置成功提示	.hint_suc_blogset
公共提示	.hint_gb

问几个简单的问题，可以帮助我们完成命名：

1. “什么类型的定义？”——是个输入框，input。
2. “类型补充说明”——如果一个词说明不清楚，那么补充说明类型，文本输入框，input_tx。
3. “在哪使用？”——定义要使用的位置在哪？首页的搜索文本输入框，input_search_index。

结合“模块化”相关的方法去定义，其实所需要定义的名称并不需要很多。如：

“hint_tx”表示提示模块的文字定义，“hit_tx_hint”表示提示里文字强调的定义，至于改变颜色还是加粗，这个就看不同提示模块的需要了。

五、 样式属性书写规范

属性先后顺序为：

- | | |
|--|----------|
| 1. Display & Flow | (显示属性) |
| 2. Positioning | (位置属性) |
| 3. Dimensions | (自身属性) |
| 4. Margins, Padding, Borders, Outline | (盒子属性) |
| 5. Typographic Styles | (版式属性) |
| 6. Backgrounds | (背景属性) |
| 7. Opacity, Cursors, Generated Content | (扩展属性) |

如下示例：

```
el {  
  
    display: ;  
    visibility: ;  
    float: ;  
    clear: ;  
  
    position: ;  
    top: ;  
    right: ;  
    bottom: ;  
    left: ;  
    z-index: ;  
  
    width: ;  
    min-width: ;  
    max-width: ;  
    height: ;  
    min-height: ;  
    max-height: ;  
    overflow: ;  
}
```



```
margin; ;
margin-top; ;
margin-right; ;
margin-bottom; ;
margin-left; ;

padding; ;
padding-top; ;
padding-right; ;
padding-bottom; ;
padding-left; ;

border; ;
border-top; ;
border-right; ;
border-bottom; ;
border-left; ;

border-width; ;
border-top-width; ;
border-right-width; ;
border-bottom-width; ;
border-left-width; ;

border-style; ;
border-top-style; ;
border-right-style; ;
border-bottom-style; ;
border-left-style; ;

border-color; ;
border-top-color; ;
border-right-color; ;
border-bottom-color; ;
border-left-color; ;

outline; ;

list-style; ;

table-layout; ;
caption-side; ;
border-collapse; ;
border-spacing; ;
empty-cells; ;

font; ;
```

```
font-family: ;
font-size: ;
line-height: ;
font-weight: ;
text-align: ;
text-indent: ;
text-transform: ;
text-decoration: ;
letter-spacing: ;
word-spacing: ;
white-space: ;
vertical-align: ;
color: ;

background: ;
background-color: ;
background-image: ;
background-repeat: ;
background-position: ;

opacity: ;

cursor: ;

content: ;
quotes: ;
}
```

每一个属性结尾都必须写分号，其中有的值如果不需要，可以不写。

六、CSS 编码规范及方法

1. 基本规则

- a) CSS 编码符合浏览器兼容标准；
- b) 属性名、属性值全部小写；
- c) 不允许将样式的定义写在行内标签中
- d) CSS 类命名要符合命名原则；
- e) 所有装饰性的图片（如背景图），要定义在 CSS 背景中；

f) CSS 文件发布前要进行压缩。

2. CSS 编码规范

选择 DOCTYPE

XHTML 1.0 提供了三种 DTD 声明可供选择：

过渡的(Transitional):要求非常宽松的 DTD，它允许你继续使用 HTML4.01 的标识(但是要符合 xhtml 的写法)。慧聪网以此声明为标准声明规范。完整代码如下：

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

严格的(Strict):要求严格的 DTD，你不能使用任何表现层的标识和属性，例如
。完整代码如下：

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

框架的(Frameset):专门针对框架页面设计使用的 DTD，如果你的页面中包含有框架，需要采用这种 DTD。完整代码如下：

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

理想情况是执行严格的 DTD，但过渡的 DTD(XHTML 1.0 Transitional)是目前理想选择。因为这种 DTD 还允许我们使用表现层的标识、元素和属性，也比较容易通过 W3C 的代码校验。

指定语言及字符集：

为文档指定语言：

```
<html xmlns="http://www.w3.org/1999/xhtml" lang="zh-cn">
```

为了被浏览器正确解释和通过 W3C 代码校验，所有的 XHTML 文档都必须声明它们所使用的编码语言；如：

常用的语言定义：

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

标准的 XML 文档语言定义：

```
<?xml version="1.0" encoding="utf-8"?>
```

针对老版本的浏览器的语言定义：

```
<meta http-equiv="Content-Language" content="utf-8" />
```

UTF8 是国际编码，它的通用性比较好，国外地区可以正确访问，GBK 是国家编码，通用性比 UTF8 差。

调用样式表：

外部样式表调用：

页面内嵌法：就是将样式表直接写在页面代码的 head 区。如：

```
<style type="text/css"><!-- body { background : white ; color : black ; } --> </style>
```

外部调用法：将样式表写在一个独立的.css 文件中，然后在页面 head 区用类似以下代码调用。

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

在本标准规范文档中，使用外部调用法。目的是可以不修改页面只修改.css 文件而改变页面的样式。如果所有页面都调用同一个样式表文件，那么改一个样式表文件，可以改变所有文件的样式。

选用恰当的元素：

根据文档的结构来选择 HTML 元素，而不是根据 HTML 元素的样式来选择。例如，使用 P 元素来包含文字段落，而不是为了换行。如果在创建文档时找不到适当的元素，则可以考虑使用通用的 div 或者是 span；

避免过度使用 div（div 滥用）和 span。少量、适当的使用 div 和 span 元素可以使文档的结构更加清晰合理并且易于使用样式；

尽可能少地使用标签和结构嵌套，这样不但可以使文档结构清晰，同时也可以保持文件的小巧，在提高用户下载速度的同时，也易于浏览器对文档的解释及呈视；

派生选择器：

使用派生选择器给一个元素里的子元素定义样式，在简化命名的同时也使结构更加的清晰化，如：

```
.mainMenu ul li {background:url(images/bg.gif);}
```

辅助图片用背影图处理：

这里的“辅助图片”是指那些不是作为页面要表达的内容的一部分，而仅仅用于修饰、间隔、提醒的图片。将其做背影图处理，可以在不改动页面的情况下通过 CSS 样式来进行改动，如：

```
#logo {background:url(images/logo.jpg) #FEFEFE no-repeat right bottom;}
```

结构与样式分离：

在页面里只写入文档的结构，而将样式写于 CSS 文件中，通过外部调用 CSS 样式表来实现结构与样式的分离。

文档的结构化书写：

页面 CSS 文档都应采用结构化的书写方式，逻辑清晰易于阅读。如：

html

```
<div id="mainMenu">
  <ul>
    <li><a href="#" >供应</a></li>
    <li><a href="#" >求购</a></li>
    <li><a href="#" >公司</a></li>
    <li><a href="#" >资讯</a></li>
  </ul>
</div>
```

CSS

```
/*=====主导航=====*/
#mainMenu {
    width:100%;
    height:30px;
    background:url(images/mainMenu_bg.jpg) repeat-x;
}
#mainMenu ul li {
    float:left;
    line-height:30px;
    margin-right:1px;
    cursor:pointer;
}
/*=====主导航结束=====*/
```

鼠标手势：

在 XHTML 标准中，hand 只被 IE 识别，当需要将鼠标手势转换为“手形”时，则将“hand”换为“pointer”，即“cursor:pointer;”

```
cursor: hand
cursor:pointer
```

3. 注释书写规范

行间注释：

直接写于属性值后面，如：

```
.search{
border:1px solid #fff;/*定义搜索输入框边框*/
background:url(..images/icon.gif) no-repeat #333;/*定义搜索框的背景*/
}
```

整段注释：

分别在开始及结束地方加入注释，如：

```
/*=====搜索条=====*/  
.search {  
    border:1px solid #fff;  
    background:url(..images/icon.gif) no-repeat #333;  
}  
/*=====搜索条结束=====*/
```

4. 样式属性代码缩写

不同类有相同属性及属性值的缩写：

对于两个不同的类，但是其中有部分相同甚至是全部相同的属性及属性值时，应对其加以合并缩写，特别是当有多个不同的类而有相同的属性及属性值时，合并缩写可以减少代码量并易于控制。如：

```
#mainMenu {  
    background:url(..images/bg.gif);  
    border:1px solid #333;  
    width:100%;  
    height:30px;  
    overflow:hidden;  
}  
#subMenu {  
    background:url(..images/bg.gif);  
    border:1px solid #333;  
    width:100%;  
    height:20px;  
    overflow:hidden;  
}
```

两个不同类的属性值有重复之处，则可以缩写为：

```
#mainMenu,#subMenu {  
    background:url(..images/bg.gif);  
    border:1px solid #333;  
    width:100%;  
    overflow:hidden;  
}  
#mainMenu {height:30px;}  
#subMenu {height:20px;}
```

同一属性的缩写：

同一属性根据它的属性值可以进行简写，如：

```
.search {  
    background-color:#333;  
    background-image:url(..images/icon.gif);  
    background-repeat: no-repeat;  
    background-position:50% 50%;  
}
```

```
.search {  
    background:#333 url(..images/icon.gif) no-repeat 50% 50%;  
}
```

内外侧边框的缩写：

在 CSS 中关于内外侧边框的距离是按照上、右、下、左的顺序来排列的，当这四个属性值不同时可直接缩写，如：


```
.btn {  
    margin-top:10px;  
    margin-right:8px;  
    margin-bottom:12px;  
    margin-left:5px;  
    padding-top:10px;  
    padding-right:8px;  
    padding-bottom:12px;  
    padding-left:8px;  
}
```

则可缩写为：

```
.btn {  
    margin:10px 8px 12px 5px;  
    padding:10px 8px 12px 5px;  
}
```

而如果当上边与下边、左边与右边的边框属性值相同时，则属性值直接缩写为两个，如：

```
.btn {  
    margin-top:10px;  
    margin-right:5px;  
    margin-bottom:10px;  
    margin-left:5px;  
}
```

缩写为：

```
.btn {margin:10px 5px;}
```

而当上下左右四个边框的属性值都相同时，则可以直接缩写成一个，如：

```
.btn {  
    margin-top:10px;  
    margin-right:10px;  
    margin-bottom:10px;  
    margin-left:10px;  
}
```

缩写为：

```
.btn{margin:10px;}
```

颜色值的缩写：

当 RGB 三个颜色值数值相同时，可缩写颜色值代码。如：

```
.menu { color:#ff3333;}
```

可缩写为:

```
.menu {color:#f33;}
```

5. hack 书写规范

因为不同浏览器对 W3C 标准的支持不一样，各个浏览器对于页面的解释呈视也不尽相同，比如 IE 在很多情况下就与 FF 存在 3px 的差距，对于这些差异性，就需要利用 css 的 hack 来进行调整，当然在没有必要的情况下，最好不要写 hack 来进行调整，避免因 hack 而导致页面出现问题。

1、IE6、IE7、Firefox 之间的兼容写法：

写法一：

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,

IE7 能识别*，也能识别!important;

FF 不能识别*，但能识别!important;

根据上述表达，同一类/ID 下的 CSS hack 可写为：

```
.searchInput {  
    background-color:#333;/*三者皆可*/  
    *background-color:#666 !important; /*仅 IE7*/  
    *background-color:#999; /*仅 IE6 及 IE6 以下*/  
}
```

一般三者的书写顺序为：FF、IE7、IE6。

写法二：

IE6 可识别 “_”，而 IE7 及 FF 皆不能识别，所以当只针对 IE6 与 IE7 及 FF 之间的区别时，可这样书写：

```
.searchInput {  
    background-color:#333;/*通用*/  
    _background-color:#666;/*仅 IE6 可识别*/  
}
```

写法三：

*+html 与 *html 是 IE 特有的标签, Firefox 暂不支持。

```
.searchInput {background-color:#333;}  
*html .searchInput {background-color:#666;}/*仅 IE6*/  
*+html .searchInput {background-color:#555;}/*仅 IE7*/
```

屏蔽 IE 浏览器：

select 是选择符，根据情况更换。第二句是 MAC 上 safari 浏览器独有的。

```
*:lang(zh) select {font:12px !important;} /*FF 的专用*/  
select:empty {font:12px !important;} /*safari 可见*/
```

IE6 可识别：

这里主要是通过 CSS 注释分开一个属性与值，注释在冒号前。

```
select { display /*IE6 不识别*/:none;}
```

IE 的 if 条件 hack 写法：

所有的 IE 可识别：

所有的 IE 可识别:

```
<!--[if IE]> Only IE <![end if]-->
```

只有 IE5.0 可以识别:

```
<!--[if IE 5.0]> Only IE 5.0 <![end if]-->
```

IE5.0 包换 IE5.5 都可以识别:

```
<!--[if gt IE 5.0]> Only IE 5.0+ <![end if]-->
```

仅 IE6 可识别:

```
<!--[if lt IE 6]> Only IE 6- <![end if]-->
```

IE6 以及 IE6 以下的 IE5.x 都可识别:

```
<!--[if gte IE 6]> Only IE 6/+ <![end if]-->
```

仅 IE7 可识别:

```
<!--[if lte IE 7]> Only IE 7/- <![end if]-->
```

l = less

g = greater

t = than

e = equal

lt IE 6 就是 less than ie 6 , 即小于 ie6 版本

lte 就是小于或等于 ie6

示例请见：[附件一](#)。:

2、清除浮动：

在 Firefox 中，当子级都为浮动时，那么父级的高度就无法完全的包住整个子级，那么这时用这个清除浮动的 HACK 来对父级做一次定义，那么就可以解决这个问题。

```
select:after {  
    content:".";   
    display:block;  
    height:0;  
    clear:both;  
    visibility:hidden;  
}
```

七、 附录一

1. XHTML 校验

校验网址：<http://validator.w3.org/>

校验方式:网址校验、文件上传校验

一般选择"Show Source"和"Verbose Output"可以帮助找到错误代码所在行和错误原因。

XHTML 校验常见错误原因对照表

- ✧ No DOCTYPE Found! Falling Back to HTML 4.01 Transitional--未定义 DOCTYPE。
- ✧ No Character Encoding Found! Falling back to UTF-8.--未定义语言编码。
- ✧ .end tag for "img" omitted, but OMITTAG NO was specified--图片标签没有加"/"关闭。
- ✧ an attribute value specification must be an attribute value literal unless SHORTTAG YES is specified--属性值必须加引号。
- ✧ element "DIV" undefined---DIV 标签不能用大写，要改成小写 div。
- ✧ required attribute "alt" not specified---图片需要加 alt 属性。
- ✧ required attribute "type" not specified---JS 或者 CSS 调用的标签漏了 type 属性。

其中最最常见的错误就是标签的大小写问题了。通常这些错误都是关联的，比如忘记了一个其他标签都会报错，所以不要看到一堆的错误害怕，通常解决了一个错误，其他的错误也都没有了。

2. CSS2 校验

校验网址：<http://jigsaw.w3.org/css-validator/>

校验方式:网址校验、文件上传校验、直接贴入代码校验

校验成功，会显示"恭喜恭喜，此文档已经通过样式表校验!"。

校验失败，会显示两类错误：错误和警告。错误表示一定要修正，否则无法通过校验；警告表示有代码不被 W3C 推荐,建议修改。

CSS2 校验常见错误原因对照表

- ✧ (错误)无效数字 : color909090 不是一个 color 值 : 909090 ---十六进制颜色值必须加 "#"号，即#909090
- ✧ (错误)无效数字 : margin-topUnknown dimension : 6pixels ---pixels 不是一个单位值，正确写法 6px
- ✧ (错误)属性 scrollbar-face-color 不存在 : #eeeeee --- 定义滚动条颜色是非标准的属性
- ✧ (错误)值 cursorhand 不存在 : hand 是非标准属性值，修改为 cursor:pointer
- ✧ (警告)Line : 0 font-family: 建议你指定一个种类族科作为最后的选择 --W3C 建议字体定义的时候，最后以一个类别的字体结束，例如"sans-serif"，以保证在不同操作系统下，网页字体都能被显示。
- ✧ (警告)Line : 0 can't find the warning message for otherprofile --表示在代码中有非标准属性或值，校验程序无法判断和提供相应的警告信息。
- ✧ 同样，通过检验后，可以放置一个 CSS 校验通过图标