

2010/7/9

河南大
学计算
机与信
息工程
学院

XHTML 和 CSS 开发规范与命名参考

心有多大舞台就有多大 | 杜晓孔

目录

1	Web 标准.....	3
1.1	Web 标准是什么?	3
1.2	采用 web 标准的好处.....	3
2	页面的基本构成和开发标准	4
2.1	选择合适的 DOCTYPE 现阶段统一使用 XHTML1.0 Transitional 标准	5
2.2	定义名字空间 标准必须引用	5
2.3	定义语言编码 默认使用 utf-8 编码, 可以根据实际需求采用其它编码	5
2.4	设置页面标题 必须设置, 不准留空	5
2.5	调用样式表 除去简单的独立页面之外全部采用外部调用的方式加载样式文件 5	
2.6	HEAD 区域其它相关设置.....	5
2.7	CSS 统一放在<head>里, 不允许放在<body>里	6
2.8	<body>内的模块都要加注释 标准注释格式如下:	6
2.9	各个模块合理嵌套和缩进 缩进一律用 Tab, 禁止用空格	6
2.10	页面中的正式内容图片, 必须填写 alt 属性, 不允许为空。	6
2.11	每页有尽可能少的 HTTP 请求。	6
2.12	同一个页面中尽量避免重复包含 CSS 文件或 JS 文件。	6
2.13	绝对不允许引用其他网站的 css 和 Javascript 等文件。	6
3	(X)HTML 书写规范	6
3.1	基本书写规范.....	6
3.2	相关标签的语义	7
3.3	模块效果与标签写法	8
4	CSS 编码规范.....	9
4.1	基本书写规范	9
4.2	CSS 命名参考	10
5	其它规范	13
6	页面质量评估标准	13

1 Web 标准

1.1 Web 标准是什么？

“WEB 标准”是一系列标准的总称，包括 HTML4.0、XHTML1.1、CSS2.1、XML1.0、RSS2.0、ECMAScript1.1、DOM1.0 等等。一般的误区经常把 WEB 标准说成 DIV+CSS。准确的说法应该是：采用 W3C 推荐的 WEB 标准中的 XHTML1.1 结合 CSS2.0 样式表制作页面的方法，DIV 应该指的是 XHTML 标签，而 CSS 指的是样式表了。

1.2 采用 web 标准的好处

1. 节约运营成本

采用 WEB 标准制作，我们可以做到表现和形式及内容的分离，我们采用XHTML 语言来表现(数据)，用 CSS 来控制(页面元素呈现的)形式。写的好的页面，XHTML 代码中基本上都是用户要看的数据，而其他修饰性的东西，全部由CSS 来控制。这样一来(XHTML)页面的体积就大大减小了，这样带宽上的费用就会大家降低了

2. 用户友好性

第一类：普通用户(每个访问我们网站的人)；

第二类：搜索引擎；

第三类：更多的设备（手机、PDA、媒体机等）

采用 WEB 标准开发的页面，结构清晰，页面体积小，浏览器兼容性好。普通用户访问的时候，页面打开速度快，而且不管用户使用那种浏览器，都能够正常访问(显示)页面，且页面的结构清晰，要找的数据可以很方便的浏览到。

而对搜索引擎来说，一个好的采用WEB 标准开发的页面，都是做过**SEO 优化**的，访问起来很友好，很容易理解页面中哪里是标题(H1~H6 标签)，哪里是段落(p 标签)，哪里是段落里要强调的内容(strong 标签) 等，它可以很容易的分析出来。而一个SEO 好的站点，被搜索引擎收录的机会更多，这个也意味着会被更多的普通用户访问到，搜索引擎会带来更多的用户。

随着智能手机的大量普及及 Wifi 配置渐渐成为主流，手机上网需求变得越来越迫切，以使我们也不得不考虑到页面在手机上的显示，对于当前标准化形式下，基于标准，将是可行之道！

3. 内容跨平台的可用性

通过将内容与设计进行分离的方法，可以使信息以较低的成本实现跨平台的转换。

4. 加快页面解析速度

通过实际测试证明，采用内容与设计分离的结构进行页面设计，可使浏览器对网页的解析速度大幅提高，相对于老式的内容与设计混编的方法，浏览器在解析中可以更好的解析结构、内容和设计。

5. 更良好的用户体验

采用表现与内容分离技术使我们在建设网站时，能够充分的对网站的内容、设计统一性、可用性进行思考与设计，从而提高网站的可用性，改善用户体验及印象。同时更加适用于在web 2.0 浪潮中不断变化的互联网形态。

2 页面的基本构成和开发标准

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <meta http-equiv="Content-Language" content="zh-CN" />
  <title>开心网</title>
  <meta name="Keywords" content="" />
  <meta name="Description" content="" />
  <link rel="stylesheet" type="text/css" href="...(外部 CSS 文件地址)...">
  <style type="text/css">
    /*070817 M. 修改** start*/
    ...(内部 CSS 内容)...
    /*070817 M. 修改** end*/
  </style>
  <script type="text/javascript" src="...(外部 Javascript 文件地址)..."></script>
  <script type="text/javascript">
    <!--
    ...(内部 Javascript)...
  //-->
  </script>
</head>
<body>
  <!--start header-->
  <div id="header">
    ...(header 内容)...
  </div>
  <!--end header-->
  <!--start content -->
  <div id="content">
    ...( content 内容)...
  </div>
  <!--end content -->
  <!--start footer-->
  <div id="footer">
    <!--070817 M. 修改** start-->
    ...( footer 内容)...
    <!--070817 M. 修改** end-->
  </div>
  <!--end footer -->
  <!--Design by M. last update 070817-->
</body>
</html>
```

2.1 选择合适的 DOCTYPE 现阶段统一使用 XHTML1.0 Transitional 标准

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

2.2 定义名字空间 标准必须引用

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

2.3 定义语言编码 默认使用 utf-8 编码，可以根据实际需求采用其它编码

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

下面的语句是针对老版本浏览器写的，以保证各种浏览器都能正确解释页面。

```
<meta http-equiv="Content-Language" content="zh-CN" />
```

2.4 设置页面标题 必须设置，不准留空

```
<title>开心网</title>
```

页面标题标准命名原则：

各级首页：开心网 - 宣传语

子频首页：开心网 - 子频道名 - 宣传语

内容页：内容标题 - 子频道标题 - 开心网

专题页：专题标题 - 开心网

2.5 调用样式表 除去简单的独立页面之外全部采用外部调用的方式加载样式文件

页面内部嵌入法：就是将样式表直接写在页面代码的 head 区。类似这样：

```
<style type="text/css">
<!--
body { background : #ffffff ; color : #000000 ; }
-->
</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 区用以下代码调用。

```
<link rel="stylesheet" type="text/css" href="http://www.kaixin.com/style/default.css" />
```

2.6 HEAD 区域其它相关设置

收藏夹图表：

```
<link rel="icon" href="/favicon.ico" type="image/x-icon" />
<link rel="shortcut icon" href="/favicon.ico" type="image/x-icon" />
```

允许搜索机器人搜索站内所有链接

```
<meta name="robots" content="all" />
```

设置站点版权信息

```
<meta name="Copyright" content="www.kaixin.com,版权所有,违者必究" />
```

站点关键词 必须添加

```
<meta name="Keywords" content="开心网 ,开心网在线" />
```

站点的简要描述 必须添加

```
<meta name="Description" content=" 开心网在线,今日在线,明日再现" />
```

2.7 CSS 统一放在<head>里，不允许放在<body>里

2.8 <body>内的模块都要加注释 标准注释格式如下：

```
开始：<!--start 模块名称-->
结束：<!--end 模块名称-->
修改记录注释：
    <!--20100708 wuhui 修改** start-->
    ...(内容)...
    <!--20100708 wuhui 修改** end-->
作者及最后更新日期注释：<!--Design by wuhui last update 20100708-->
其中**代表具体内容，20100708 代表日期，wuhui 代表作者信息
```

2.9 各个模块合理嵌套和缩进 缩进一律用 Tab，禁止用空格

2.10 页面中的正式内容图片，必须填写 alt 属性，不允许为空。

对于页面装饰性质的图片，如虚线、分隔小图等，要定义在 CSS 中，制作过程中，如不知道 title 详细信息，可以按照下面的方式书写：title= “#”。

2.11 每页有尽可能少的 HTTP 请求。

页面内的 HTTP 请求，包括引用的外部 CSS 和外部 JS 文件、图片、Flash 文件等。

2.12 同一个页面中尽量避免重复包含 CSS 文件或 JS 文件。

2.13 绝对不允许引用其他网站的 css 和 Javascript 等文件。

3 (X)HTML 书写规范

3.1 基本书写规范

1. 所有标签的元素和属性的名字都必须使用小写。
2. 属性的值一定要用双引号(“”)括起来，且一定要有值。所有的属性必须用引号””括起来，

例如: <height=80> 必须修改为: <height="80">

特殊情况, 你需要在属性值里使用双引号, 你可以用", 单引号可以使用', 例如:

<alt="say'hello'">

3. 所有的标记都必须要有开始和相应的结束标记, 所有标签必须关闭。对于不成对的标识, 在标识最后加一个空格, 然后跟一个"/"。例如
写成
、写成, 加空格的原因是避免代码连在一起浏览器不识别。

4. 所有的 XHTML 标记都必须合理嵌套, 以前这样写的代码:

<p></p> 必须修改为: <p></p> 就是说, 一层一层的嵌套必须是严格对称。

5. 把所有<和&等特殊符号用编码表示。任何不是标签部分的小于号(<), 都必须被编码为<;任何不是标签部分的大于号(>), 都必须被编码为>;任何不是实体部分的与号(&), 都必须被编码为&;, 在 utf-8 编码时尤其需要注意。

6. 在注释内容中不能使用“--”, 如有需要, 必须用==来替换。例如下面的代码:

<!--这里是注释-----这里是注释--> 必须改成<!--这里是注释=====这里是注释-->

7. 不要使用被排斥的元素: 一些元素和属性在 HTML 4.01 Transitional 和 XHTML 1.0 Transitional 是被允许的, 而在 XHTML 1.0 Strict 下则不被支持。比如、<center>、alink、align、width、height (对于某些元素)、bgcolor 和 background。

不建议使用的标签: b、big、hr、i、small、sub、sup、tt

禁止使用的标签: basefont、center、dir、font、isindex、menu、s、strike、u、bgcolor

8. 表现结构完全分离, 代码中不涉及任何表现元素, 如 font、bgColor、border 等

9. <h1>到<h6>的定义, 应遵循从大到小的原则, 体现文档的结构, 并有利于搜索引擎的查询。

10. 给每一个表格和表单加上一个唯一的、结构标记 id ; 给重要的区块加上注释; 给图片加上 alt 标签

11. 嵌套结构控制在 5 层之内, 对于特殊页面, 可适当增加嵌套层数, 但是也应该控制在 8 层之内。

12. 标签结构尽量采用语义化标签

13. 页面中空格的使用: 全角: 中文空格 半角:

3.2 相关标签的语义

标题

为了给标题做标记, 要用 <h1>、<h2>、<h3>、<h4>、<h5> 或者 <h6>, 这完全取决于标题的等级。<h1> 是最高的等级。

段落

用 <p> 来标记段落。而不是用
 来生成段落间的空行。用 CSS 来控制段落间的空隙 (Margins), 这个空隙可以确保段落标记的正确无误。

列表

一些事物的罗列应使用列表来显示。在 XHTML 中有三种列表的方法: 无序、有序和自定义。无序列表, 就是我们所熟知的圆圈列表, 以 开始, 以结束。每一个列表项都包含在之中。

有序列表, 以开始, 以结束。

自定义列表有一些不同，可以用来标记一些列表项和描述，以<dl>开始，以</dl>结束。每一个被描述的项目，要包含在<dt>中，而描述的内容要包含在<dd>中。

强调

 是用作强调的，是用作重点强调的。 大部分浏览器用斜体显示 *强调*的内容，用粗体来显示重点强调的内容，然而，这是没有必要的，如果是为了确定强调内容的显示方式，最好的方法就是使用 CSS 来定义他们的表现。当你想要的只是视觉上的效果时，就不要使用强调了。

表格

XHTML 中的表格不应用来布局。然而如果是为了标记列表的数据，就应该使用表格了。为了使数据表格有更强的访问性， 了解和使用各种构造表格的组件就很重要了。比如表格标题（<th>）、摘要（summary 属性）和首部说明 （<caption>标签）。

布局标签 DIV

这个 div 没有什么特性的意义，可以使用在很多地方，也就是说他可以装不同的东西。他的正确的写法是<div></div>必需要有封口。大家都用来做布局之用， 也有用来作为存放文章形成段落，实际上，这个做法并不是很好，因为作为文章的分段自然有一个特定的标签来用。那就是下面要讲的<p></p>标签，不过用 DIV 来再整体的包住所有的段落。这是非常实用的。

辅助布局标签 SPAN

这也是一个非常常用的标签，这个标签可以说与 div 很像，没有什么特定的意义，只是他是一个级联元素，不是块级元素。可以把它看作是一个袋子，他不像箱子那样可以有自己的宽高，他的宽高只能随着内容的多少而定，所以很像是袋子。这个标签与 div 正好互补。

图片标签

img 是图片标签，也是个特定属性的标签。正常写法是： 这里的 src 是目标地址，alt 与 title 是替换文字，alt 是 IE 特定的，title 是其它浏览器的通用的。不过记得后面的反斜线那是一定要有的。

3.3 模块效果与标签写法

模块效果	正确标签写法
标题： 一级标题 二级标题 三级标题	<h1>页面标题</h1> 一个页面只有一个标题 <h2>模块标题</h2> <h3>栏目标题</h3>
列表： 出租丈夫能否成为另类职业 胸罩的出现是进步还是退步 北京人凭什么上北大清华？ 日常交往中应提倡 AA 制吗 打工仔为何爱看脱衣舞？	 文章标题 文章标题 文章标题 或者用 文章标题 文章标题 文章标题 或者用

	<pre><dl> <dt>...</dt> <dd>...</dd> </dl></pre>
段落： <u>世界上，有出租土地的，有出租房屋的，有出租牛羊牲畜的，如今居然连夫妻都能出租了。</u>	<pre><p> 文章内容 </p></pre>
表单： 登录论坛 用户名: <input type="text"/> 密 码: <input type="password"/> <input type="button" value="登 录"/>	<pre>提交按钮: <button type="submit">提交</button> 普通按钮: <button type="button">确定</button> 输入框: <input type="text" name="input1" /> 单选框: <label for="input"><input id="input" type="radio" name=" " value=" " />男</label> 多选框: <label for="persist"><input id="input" type="checkbox" name="t" value="" />记住帐号</label> 选择列表框: <select name="select"> <option value="" checked="checked">网页</option> ... </select> 文本框: <textarea name="cont" cols="10" rows="10"> 内容 </textarea></pre>

4 CSS 编码规范

4.1 基本书写规范

- 1、尽量不缩写，除非一看就明白的单词。
- 2、一定要使用英文；
- 3、注释写法，例：/*header*/ 注释中间不要加---
- 4、所写的 **CSS** 要符合浏览器兼容标准；
- 5、样式名称首字母统一为小写字母，不能为数字，下划线及特殊字符，且每一项 CSS 定义写成一行；
- 6、尽量不要将样式的定义写在标签中（如：style="font-color:red;"）；
- 7、所有装饰性的图片（如背景图），要定义在 CSS 中；
- 8、ID 必须是唯一的，且用在结构的定义中；
- 9、样式名需要组合拼写时，采用驼峰式拼写，即：keyWord
- 10、单位建议使用“em”或者“px”。使用 px(像素)为基本计量单位。
- 11、CSS 中谨慎使用 expression，尽量不要使用；
- 12、CSS 中尽量不要使用滤镜。

13、css 不允许随意命名，如：div1,div2,head1,head2。Css 的命名一定要有一定的实际含义，以便于他人理解和后续开发。如：head,foot。

4.2 CSS 命名参考

1、常用的 CSS 命名规则：

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 或者 wrap

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、直观命名

当在设计 Web 页面以及需要对一个 div 进行标识的时候，最自然的想法就是使用可以描述元素所在页面位置的词汇来对其命名。这种方法使得类以及 id 的名称如下面所示：

自上而下小组: `toppanel`

横向: `horizontalnav`

左面: `leftside`

中心-栏目: `centercolumn`

右面: `rightcol`

这些是 CSS 以及 XHTML 类和 id 的有效命名方式。这些词汇简单并且能够使人顾名思义, 因此满足了标识页面元素以及相应的 CSS 样式的需要。

3、结构化命名

结构化的标记意味着表达方式/位置信息同内容的完全分离——这其中包括出现在标记 (markup) 中的类和 id 名称。

有标记的相关信息都是用来描述文档的结构而不是外观。这样的特点使得我们可以通过简单的改变 CSS 的方式来对不同外观格式下的内容 (content) 以及标记 (markup) 进行重用。当你理解这种方式时, 很容易就可以发现采用页面位置来为类以及 id 命名的方式在处理如音频 (audio) 等外观格式上显得非常不合适。因此, **应当根据在文档中的使用目的而非出现位置来对类以及 id 进行结构化命名。**

可以按照如下所示的结构化方式来对类以及 id 名称命名:

顶部抢眼部分: `branding`

导航部分: `mainnav`

主要内容部分: `maincontent`

这些名字同直观命名方式一样非常易懂, 但他们描述了页面元素的作用而非位置。这使得代码更加符合使用纯粹的结构化标记 (structural markup) 的初衷, 即开发人员可以在不改变标记的情况下对各种各样媒体下的显示格式进行处理。

4、惯例

Andy Clarke 分析了 40 份由推崇标准化 Web 设计理念的开发人员所设计的 Web 站点的源代码。尽管类以及 id 名称很不统一, 但是还是发现了一些频繁出现的常用名称。这里给出了最常用类/id 名称的示例列表:

id 的命名:

(1) 页面结构

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`

(2) 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

(3) 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

class 的命名:

(1) 颜色: 使用颜色 16 进制代码, 如

```
.f60 { color: #f60; }
```

```
.ff8600 { color: #ff8600; }
```

(2) 字体大小, 直接使用“font+字体大小”作为名称, 如

```
.font12px { font-size: 12px; }
```

```
.font9pt { font-size: 9pt; }
```

(3) 对齐样式, 使用对齐目标的英文名称, 如

```
.left { float:left; }
```

```
.right { float:right; }
```

(4) 标题栏样式, 使用“类别+功能”的方式命名, 如

```
.barnews { }
```

```
.barproduct { }
```

5、自定义命名:

根据 w3c 网站上给出的,最好是用意义命名

比如: 是重要的新闻高亮显示 (像红色)

有两种

```
.red{color:red}
```

```
.important-news{color:red}
```

很显然第二种传达的意义更加明确,所以尽量不要用意义不明确的作为自己自定义的名字

5 其它规范

1. 项目完成包中, 文件夹及文件名称全部采用小写字母, 不允许使用中文文件名。
2. 非特殊情况下要求表现和内容分离, 代码中不要涉及任何表现的元素, 例如: `style`、`font` 等。
3. 图片的命名中不允许出现空格, 重要的图片的命名要有一定含义不可随意命名。
4. 遵循 w3c XHTML 代码编写规范。
5. 尽量避免使用 hack 方法来解决浏览器兼容问题。
6. 根据页面代码结构进行包含缩进, 这样代码层次结构清晰;
7. 将样式表放在顶部, 将脚本放在底部, 避免 css 表达式。

6 页面质量评估标准

1、页面布局及里面的 Javascript 特效对 IE 6.0, IE 7.0, IE8.0, Firefox 等浏览器 100%兼容。产品测试必须测试的七种浏览器: IE6、IE7、IE8、firefox (此四个为重点)、chrome、opera、safari (此三个为次要)。

判断方法: 浏览器兼容性测试

2、(X)HTML 代码结构是否清晰。

判断方法: 查看源文件, 是否符合页面标准。各个模块是否清晰可辨或是有注释说明。

3、(X)HTML 代码结构是否复杂。

判断方法: 每个模块 (x)html 代码的嵌套层次简单, 代码简洁。

4、(X)HTML 代码是否和 CSS 混杂在一起。

判断方法: 是否存在大量在标签中使用 `Style` 定义的样式。

5、(X)HTML 代码是否和大量 Javascript 混杂在一起。

判断方法: Javascript 没有集中的放在 `<head>` 中或是放在页面的底部, 而是混杂在模板中。

6、(X)HTML 代码是否大量出现不被推荐使用的标签。

判断方法: 不被推荐使用的标签见第二部分的列表, 如 ``、`` 等。

7、(X)HTML 代码是否书写规范, 是否结构化布局。

判断方法: 依照结构化代码规范和具体的语义性检查。

8、CSS 书写是否规范。

判断方法：查看 CSS 代码部分 CSS 命名是否标准，单个定义是不是过长，是否出现过多重复的定义。

9、(X)HTML 代码的格式是否混乱。

判断方法：格式是否整齐、干净。

10、 页面是否存在性能问题。

判断方法：观察 CPU 的占用情况，和页面的打开速度。页面的下载速度明显感觉很慢的不行。