Web 前端开发规范教程

编 写	邱俊波 **Kane (** www.iamkane.com **)**
日 期	**2009.4**
参 考	部分内容来自互联网 www.w3.org www.w3cn.org www.iamkane.com www.cssforest.org www.blueidea.com www.lanrentuku.com www.ibm.com/developerworks/cn/web/

一．背景知识

1.1 WEB 标准

WEB 标准不是某一个标准，而是一系列标准的集合。网页主要由三部分组成：结构（Structure）、表现（Presentation）和行为 （Behavior）。对应的标准也分三方面：结构化标准语言主要包括 XHTML 和 XML，表现标准语言主要包括 CSS，行为标准主要包括对象模型（如 W3C DOM）、ECMAScript 等。这些标准大部分由 W3C 起草和发布，也有一些是其他标准组织制订的标准，比如 ECMA（European Computer Manufacturers Association）的 ECMAScript 标准。我们来简单了解一下这些标准：

1.11 结构标准语言

（1）XML

XML 是 The Extensible Markup Language(可扩展标识语言)的简写。目前推荐遵循的是 W3C 于2000年10月6日发布的 XML1.0，参考（www.w3.org/TR/2000/REC-XML-20001006）。 和 HTML 一样，XML 同样来源于 SGML，但 XML 是一种能定义其他语言的语。XML 最初设计的目的是弥补 HTML 的不足，以强大的扩展性满足网络信息 发布的需要，后来逐渐用于网络数据的转换和描述。关于 XML 的好处和技术规范细节这里就不多说了，网上有很多资料，也有很多书籍可以参考。

（2）XHTML

XHTML 是 The Extensible HyperText Markup Language 可扩展标识语言的缩写。目前推荐遵循的是 W3C 于2000年1月26日推荐 XML1.0（参考 http://www.w3.org/TR /xhtml1）。XML 虽然数据转换能力强大，完全可以替代 HTML，但面对成千上万已有的站点，直接采用 XML 还为时过早。因此，我们在 HTML4.0的基础上，用 XML 的规则对其进行扩展，得到了 XHTML。简单的说，建立 XHTML 的目的就是实现 HTML 向 XML 的过渡。

1.12 表现标准语言

CSS 是 Cascading Style Sheets 层叠样式表的缩写。目前推荐遵循的是 W3C 于1998年5月12日推荐 CSS2（参考 http://www.w3.org/TR/CSS2/）。W3C 创建 CSS 标准的目的是以 CSS 取代 HTML 表格式布局、帧和其他表现的语言。纯 CSS 布局与结构式 XHTML 相结合能帮助设计师分离外观与结构，使站点的访问及维护更加容易。

1.13 行为标准

（1）DOM

DOM 是 Document Object Model 文档对象模型的缩写。根据 W3C DOM 规范（http://www.w3.org/DOM/），DOM 是一种与浏览器，平台，语言的接口，使得你可以访问页面其他的标准组件。简单理解，DOM 解决了 Netscaped 的 Javascript 和 Microsoft 的 Jscript 之间的冲突，给予 web 设计师和开发者一个标准的方法，让他们来访问他们站点中的数据、脚本和表现层对像。

(2) ECMAScript

ECMAScript 是 ECMA(European Computer Manufacturers Association)制定的标准脚本语言（JAVAScript）。目前推荐遵循的是 ECMAScript 262（http://www.ecma.ch/ecma1/STAND/ECMA-262.HTM）。

二．如何实现 Web 标准

2.1 软件环境

2.11 制作软件

制作网页，我们都知道用 Dreamweaver 和 Frontpage 等直接视图进行网页制作。而要实现 Web 标准，这些软件或者工具都必须升级到最新版。如 Adobe Dreamweaver CS4、 Microsoft Expression web 2。此外我们推荐另外一个功能强大的软件- Apatana Studio 1.2. 使用最新的软件，更有助编写出严格标准的页面。同时在此，我们希望所有人可以暂时抛弃视图制作页面，改为纯手写 HTML 和 CSS。培养良好的书写代码习惯。

- Apatana Studio 1.2 (免费)
- Adobe Dreamweaver CS4
- Microsoft Expression Web2

2.12 测试软件

由于存在各种浏览器，所以我们制作出来的页面必须兼容各种浏览器。首先我们必须兼容最常见的几款浏览器： IE6、IE7、IE8、Firefox3 。当你的页面都支持以上 4 种浏览器 ，Opera 、Safari 、Chrome 等其他浏览器基本上已经都没问题了。这里我们推荐几款辅助工具。

- IETester - 集成 IE5.5-IE8 http://www.my-debugbar.com/wiki/IETester/HomePage
- IE Developer toolbar - 微软出的 IE 下查看页面 DOM 结构的 IE 插件
- Firefox Firebug 插件 -Firefox 下的 DOM 查看插件
- Debugbar - 和 IE developer toolbar 差不多

2.2 网站架构的流程

网站的架构流程必须按照步骤走，步骤大概分为 4 步

1. 设计稿的分析
2. 切图
3. 编写 xhtml 和 css
4. 编写 JavaScript 脚本

2.21 设计稿的分析

1. 能分清设计稿中的公共与私有的部分

从最基本的开始，分清公共部分如菜单、导航、大框架和每个独立页面所用到的部分等，至少在想法上做不同的对待。

2. 在 1 的基础上对各部分的实现方式有一个初步的方案（包括如何切图、写结构、写样式）

在分清公共和私有部分后，分析最简单的实现方法，如哪些部分是可以平铺的，哪些是可以重复被使用的等等。

3. 在 1 的基础上，准确的给出各部分的实现方案（包括如何切图、写结构、写样式）

在分清公共和私有部分后，能准确的给出各部分的实现方案，如“滑动门技术”的实现方法有 2 种，选择哪种方法更合适项目？图片应该如何切？应该使用哪种 HTML 和 CSS 的写法？

4. 在 3 的基础上，能同时考虑方案的扩展性、复用性及页面性能（包括如何切图、写结构、写样式）

在给出的方案中考虑是否可扩展、如何重复使用、将哪一类的图合并可以最大化页面的性能。这里还要注意有些模块的内容可能是动态的，当内容改变后如何兼容。

5. 在 4 的基础上，考虑整站的结构分布（包括文件分布、目录结构）

考虑上面方案的综合效率，如维护所需要的成本、页面打开速度、带宽成本、服务器开销等等。

1～3 点应该成为基本的技能，4～5 属于更高的要求。这块的熟练度会影响到后继切图和写代码的效率、返工的机率及维护成本的高低。

2.22 切图

1. 切成所需要的图片（如何将需要的部分切出来）

最基本的，将需要的图切出来，有时候会需要 PS 出自己需要的图。

2. 在 1 的基础上，对切出来的图片进行一些优化（包括压缩文件大小、选择图片类型）

压缩输出的图片，在不影响画面质量的前提下，尽可能的减少文件字节数。这个工作很重要，优化一张图片所带来的效果可能比优化很多的代码所带来的效果要明显得多。

3. 在 2 的基础上，规划切出来的图片（包括文件分布）

规划切出来的图片，哪些图应该被合并，存放于哪个目录等等。

4. 在 3 的基础上，考虑整体的性能（包括合并图片、压缩文件大小）

同样是综合整体的性能、效率。

1～3 点应该成为基本的技能，4 属于更高的要求。这块会影响到 HTML 和 CSS 的写法，还有页面的性能（流量、链接数）。

2.23 HTML 和 CSS 的编写

1. 还原设计稿视觉效果，并通过标准验证（HTML）

还原设计稿，是页面制作最基本的要求，不管设计稿是否符合自己的审美观，做为页面重构工作者，还原设计稿是一项职业素质。通过标准验证是检验我们输出的质量很重要的一个方法。虽然最终的页面不一定可以通过验证，但我们所输出的静态页面大部分是可以做到通过验证的，除非有特殊的需求。当然我们不是为了过验证而做页面，是为了标准化。

2. 在 1 的基础上，实现多浏览器的兼容（HTML）

兼容多浏览器，要实现多浏览器的兼容，少不要了解下各浏览器的习性。不过对于什么才是兼容，在《 中国式 WEB 标准 》中有讨论。《 中国式 WEB 标准 》http://www.cssforest.org/blog/index.php?id=93

3. 在 2 的基础上，标签语义化（HTML）

标签语义化，是 WEB 标准的核心内容，只有做好了语义化，才能说得上做到了 WEB 标准。虽然在国内没有统一的标准，不过一些基本的语义标签的使用还是可以定下的，如段落、列表、表格等等。

4. 在 3 的基础上，选择较优的实现方式（包括模块化结构，方便程序脚本使用，HTML 和 CSS）

做好了上面 3 点，只能说单一的页面做好了，下面得考虑两个以上的页面了。模块化就是为了更好的提高复用，减少重复开发所带来的浪费。模块化也是被越来越多的人所关注，可以说是发展的一个趋势，随着大家对 HTML 和 CSS 掌握越多，如何更好的发挥它们也成了提高工作效率的重点，其中模块化就是很好的一种方式，打算在之后写相关的内容。

5. 在 4 的基础上，考虑到扩展性、复用性和可维护性（HTML 和 CSS）

做好了模块化，并不一定就是最优化的，如果考虑上扩展性、复用性和可维护方面的内容，模块化有时反而会不利于这几个方面，如何平衡这几方面，是一个更高的要求。

6. 在 5 的基础上，考虑到整站的样式分布（包括如何实现分布）

这个算是整站的规划了，需要多少个样式，多少个目录，这些样式文件分别存于哪个目录（当然同时也需要考虑图片的分布）

7. 在 6 的基础上，样式写法的优化（包括技巧的应用）

这点需要跟上面的第 4、5、6 点结合，需要做综合的考虑，使用更合适于项目的样式写法。

1～3 点为基本的技能，4～7 属于页面优化方面的内容。这块影响了一个页面甚至一个站点从无到有、从有到优。掌握好各个点的知识，会让页面在越短的时间内达到最优的状态。当然这也是个人能力的体现。

2.3 XHTML 的编写代码规范

在开始设计 XHTML 页面以前，我们必须先了解有关 XHTML 的代码规范，养成良好的书写习惯。在上一篇文章中我们也看到标签不同于 HTML 中的的写法，下面我们详细介绍一下 XHTML 代码的书写规范。

2.31 所有的标签都必须使用结束标记

在 XHTML 中如果出现开始标签，就必须有相对应的结束标签。如： ， 等。如果使用单体标签，则必须用“/”斜线来结束。如：
，等。

2.32 所有标签和属性名称都必须小写

在 XHTML 中标签和属性名称大写或大小写混杂是不被允许的。如：在 HTML 中<DIV Class="Abc">是没错的，但在 XHTML 中必须写成： 。div 是标签元素，class 是属性名称都必须小写，而 Abc 是属性值，在 XHTML 中属性值是不被限制的。但为了不易出错，建议全部小写。
某些开发工具自动生成的如：onMouseOver 也必须写成：onmouseover。

2.33 属性值必须使用双引号括起来

在 HTML 中可以写成 ，但 XHTML 规定必须写成： 。

2.34 不允许使用属性简写

尤其是在表单元素中，以前 HTML 允许写成如：<input checked>、<option selected>等，但 XHTML 规定所有属性必须被赋值。正确的写法是：<input checked="checked" />、<option selected="selected" />。

2.35 所有标签都必须合理嵌套

XHTML 书写结构的要求是非常严谨的，因此所有的嵌套都必须按顺序，即最先出现的标签，最后结束 。正确的写法是： XHTML 代码规范 。

2.36 不是标签一部分的特殊符号都用编码表示

出现在内容中的特殊符号都需要用编码形式表现出来。

1、任何（<），不是标签的一部分，都必须被编码为：<

2、任何（>），不是标签的一部分，都必须被编码为：>

3、任何（&），都必须编码为：&

4、任何（"），不是标签的一部分，都必须被编码为："

如 果要在网页中正确显示 HTML 代码：近水社区，在开发工具中正确的写法为：< href="http://www.w3cui.org">近水社区</a& gt;。

2.37 图片标签必须要有 ALT 属性

为了使浏览者在图片未显示的情况下依然可以了解要表现的意义，XHTML 规定没一个图片标签都要有 alt 标签。如只起修饰作用没有任何意义的图片也要设置 alt 属性，属性值为空。正确写法是：。

2.38 不能在注释中使用两个以上的破折号“--”

在 注释中的内容中，不能出现两个或两个以上破折号“--”，只能出现在注释的开头和结束，在内容中它们不再有效。如：<!-- 这里是注释 -- 这里是注释 -->是错误的写法，其中的“--”可以用空格或等号“==”替代。正确的写法是<!-- 这里是注释 -->，或<!-- 这里是注释 == 这里是注释 -->。

以上的写法虽然不会对网页的显示造成影响，但在进行 W3C 效验的时候却会出很多莫名其妙的问题。为了使代码更加规范，易于阅读和维护，为转向 XML 做准备，养成良好的书写规范还是很有必要的。

2.4 CSS 相关规范

2.41 选择 DOCTYPE

同样 CSS 在不同声明中的页面所起的效果也许会不一致。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)。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“>

理想情况当然是严格的 DTD，但对于我们大多数刚接触 web 标准的设计师来说，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。

2.42 指定语言及字符集

为文档指定语言：

<html xmlns=”http://www.w3.org/1999/xhtml” lang=”en”>

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″ />

为提高字符集，建议采用“utf-8”，另外除了代码的编码使用 utf-8，页面保存的格式也应用 utf-8 否则在某些编辑器下会出现乱码的情况。

关于 UTF-8 字符处理在 Web 开发中的应用的可以参考 IBM 的文档

http://www.ibm.com/developerworks/cn/web/wa-lo-utf8/index.html

2.43 调用样式的方法比较

大家知道页面要使用外部 css 文件的方式有两种，一种是引用(link)，例如：<link rel=’stylesheet’ href=’a.css’>；另一种是导入(@import)，例如：<style>@import url(’a.css’);</style>；两者引用的方式在页面上的展现效果却是一样的，但是两者又有着很重大的区别：就是页面的性能问题！

根据数据的比较（http://www.stevesouders.com/blog/），我们强力建议使用 link 方式调用样式。不要使用@import 方法。

另外，请不要在 html 页面内书写 css 样式。除非页面是单一独立的。

2.44 编写样式的习惯

我们要养成手写 CSS 的习惯，在调式的时候可更快更有效的实现效果。注意的是：

- 适当写好相关注释
- 适当的使用 ID 选择符（#element）、类选择符(.class)、包含选择符(#element span) 、选择符分组(#element,.class,span)
- 使用 reset 将默认的 css 重置
- 将常用的布局方法和常用颜色等写成全局类选择符
- 将部分页面可复用的部分分离出来
- Css Hack 的最有效方法 #element{color:red;*color:blue;_color:black}

```
#element{
/*顺序是FF&IE8,IE7,IE6 不能颠倒! */
height:20px /* 只对FF和IE8有效*/
*height:20px /*只对IE7有效*/
 height:20px /*只对IE6有效*/
}
```

- 每个属性占一行，而不要将全部属性写在一行了。
- 属性较多的时候可以按照 a-z 的顺序排列，方便修改的时候更快找到所需的属性和避免多写。

2.5 Javascript 的编写代码规范

Javascript 是 ECMAscript 的一种，Web2.0 的出现给 JavaScript 带来了重生。XHTML 和 JavaScript 的有效结合，可以大大提升网站用户的体验感。为了更有效的编写 JavaScript，缩短开发周期，出现了各种各样的 JavaScript 框架。下面我们会介绍一下各种库和编写 JavaScript 的规范。

2.51 选择适合自己用的 JavaScript 框架（库）

目前比较流行的 JavaScript 库有：

- Jquery http://www.jquery.com
- Prototype http://www.prototypejs.org
- script.aculo.us http://script.aculo.us/
- Dojo http://www.dojotoolkit.org/
- Yui-ext http://extjs.com/
- Moo Tools http://www.mootools.net/
- Spry Framework http://www.adobe.com/

框架的数目非常多，如何选择呢？我们可以参考以下规则：

- **项目需求**。“这个 Web 站点或者 Web 应用，是否需要 AJAX，健壮的事件处理？是否需要特效库？”框架直接提供的功能的总量，以及使用框架需要的经验同样需要考虑。
- **对浏览器的支持**。虽然大多数框架能够支持大多数浏览器，“……但是通常会有不易察觉的例外——Mac 操作系统上的 Safari 浏览器常有这样的陷阱”。
- **开发团队对框架的支持力度**。有一个核心开发团队来维护的框架是最好的。这样 Bug 报告和问题会有更快的响应速度，而且测试会更加严格，会更好地遵守开发指导方针。
- **框架的成熟度**。“框架的成熟度是最能说明其寿命的指标，同时也是框架的坚实基础。一个成熟的框架不会是 beta 版的……”一个兴旺的社区，以及提供 Subversion 或 CVS 代码库，也是成熟标志 。
- **公开更新和发布的频率**。长时间的延迟和臃肿的发行版，都是你在将来得不到框架的有效支持的明确信号。反之，过多的公开版本意味着不稳定，或者项目不够专注。”
- **文档质量**。文档是一个重要的区分指标。强健的文档包括 API、书籍、教程和博客，而“只谈论句法的文档是最差劲的”。每个方法和属性的例子也是很有帮助的。
- **存在一个活跃的社区**。“有经验的用户是愿意伸出援助之手还是打发你去其他地方寻找帮助？是否有开发者为框架开发扩展或者为核心框架作出贡献？”社区的个性也是判断社区是否可依靠的风向标。
- **基准测试**。基准测试可以帮助我们对于框架的性能方面得到概括的认识。基准测试还说明了框架采用了某些质量保证方面的最佳实践。另外，“……即使是速度上的略微提高，或者软件的尺寸缩小了，都可以视作一种正面的改进。”
- **框架的可扩展性**。“支持插件对于任何 JavaScript 框架来说都绝对是加分的，但是通常开发者仅仅想知道——在核心库中诊断问题所在有多困难？”
- **API 风格**。“这是一个重要的问题，但是是个复杂的问题。对于大多数开发者来说，只有在很多项目中使用过几个 JavaScript 框架之后，才会对这个问题有所认识。简洁和连贯性(chainability)是两个非常重要的特征，不要忽视。”

而目前 VS2008 、Aptana 等几大 Web 开发软件都内置了 jquery 或 prototype 框架，所以我们可以选择

jquery 或 prototype 来做框架。 这2个框架各有各的好处，最终看个人比较精通哪个框架。

2.52 JavaScript 的编写规范

- **不要在页面包含不必要的 javascript 代码，尽可能将其外部化。**
  JavaScript 代码不应该嵌入在 HTML 文件里，除非那些代码是一个单独的会话特有的。HTML 里的 JavaScript 代码大大增加了页面的大小，并且很难通过缓存和压缩来缓解。

- **尽可能使用语义化名称来表达函数方法，并且写上相应的注释。**
  代码质量在软件质量中占很大比例。在软件生命周期里，一个程序会被许多人接手。如果一个程序可以很好的表达自己的结构和特性，则在不久的将来修改它时就会减少程序崩溃的可能。当 js 代码量大的时候，注释会起了很大的重用，所以请不要忽视注释。

- **减少冗余代码，将可共用的部分独立出来实现重用。**
  页面要实现用户体验效果，尽量把函数写得通用，在不同的页面之间可以重复使用。

- **代码的缩进和换行**
  代码不要全写在一行，每一条语句应该用分号结束并换行。适当的使用缩进，让代码容易看。

- **应遵循 javascript 的编程规范，避免出现不必要的错误。**
  可参考：http://www.ibm.com/developerworks/cn/java/j-cb12196/index.html
  http://www.blueidea.com/tech/web/2006/3567.asp
  《Javascript Dom 编程艺术》一书

三．目录结构和命名规范

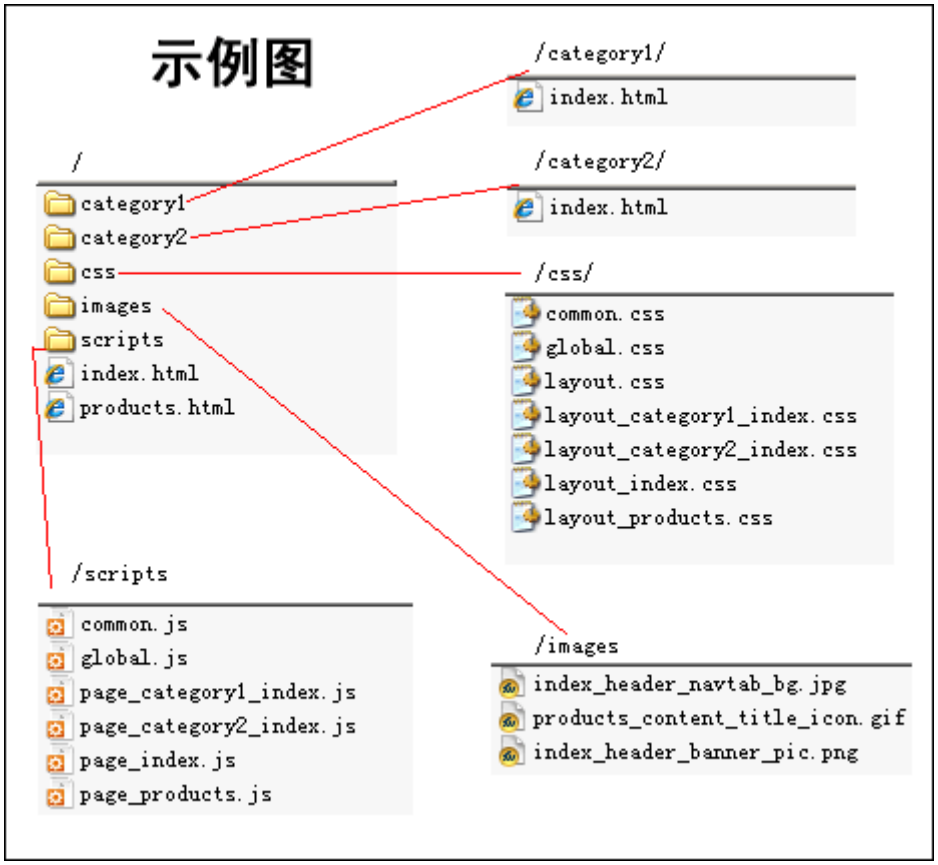
我们单独一节来说明网站目录结构和命名规范的目的是让我们重视。因为无论你的 XHTML、CSS、Javascript 写得多熟练多好，而网站的目录结构和其命名是让人和搜索引擎读不懂的，那么网站就没有真正做到标准化而且整个网站的后期扩展和维护的成本和代价会很大。

还有一点需重视就是 xhmtl css js 的代码注意缩进，并保持**格式整齐**的并且提供注释，保证可阅性 。同时为后期编写程序提供良好的开发条件。

首先网站的目录结构和文件命名清晰、XHTML 里面的元素命名清晰会给 SEO 带来好处。例如文件的命名，如果使用全拼，那么 Google 是自动识别拼音进行排名的。关于 SEO 相关的知识，就不在这里一一阐述了。接下来我们详细介绍目录结构和命名规范。

3.1 网站的目录结构和文件命名

以下是一个大概的目录结构示例图



3.11 目录结构

目录结构主要分为四类，需要注意的是，所有命名必须为小写英文，不能大写或中文。

- categorys(目录)

  目录的命名尽可能使用英文或者全拼表达目录内的页面作用（语义化），需要注意的是不要使用中文词组简拼（eg: 目录 --> ml）。简拼容易出现重复、或者目录结构复杂的时候容易出现混乱，给后期维护带来很大的麻烦。

- css

  css 的目录命名可以为 style、css、skin 等，如果网站的目录结构不是很复杂的，尽量把 css 统一放在跟目录。这样可以方便后期的维护操作。如果网站的目录结构很复杂，层次超过 3 层以上的，可以在对应的层设置目录页面结构（layout）的 css。

- js

  js 的目录命名一般用 scripts 或者其个容易让人知道里面是放 js 脚本的名称。同样 js 的目录结构也是和 css 一样。

- images

  images 根据网站规模来调整放图片的目录。，一般根目录设置的 images 是存放整站共用的图片（包括图片图标背景等），而各二级三级目录里面也可以设置相应的 images 目

录存放当前级的图片。

当页面在引用 css 或者 js 的时候，大型的门户站点一般会在引用加上版本号或者日期。如：<script type="text/javascript" src="xxx.js?v=1.0"></script> ，这样的做法是为了维护的时候可以更清晰知道所引用的脚本或样式是什么时候什么版本的。

3.12 html 命名

html 应遵循页面的内容或用途（SEO）进行命名（语义化）。不能使用中文词组的简拼进行命名。当使用英文或者中文词组全拼的时候，同样会给 SEO 带来好处。

另外需注意的是，整个网站的 html 后缀要统一，避免同时出现 html 、htm 两种不同的后缀 。

3.13 css 命名

css 应按照内容和功能进行命名。一般可以分 2 种类型。

- 功能性质

  功能性质一般指： reset.css（重置默认的样式-属性选择符）、global.css (全局使用的类选择符) 、common.css (部分页面可共同使用的类选择符) 等各种按功能分类的 css。一般小型的站点甚至可以将 连接 、段落、颜色等样式分离出来。

- 布局页面

  布局页面一般指: layout.css (全站的整体框架布局) layout_category_index.css (针对某个目录页面进行的布局)。

3.14 js 命名

js 命名规范也和 css 的命名规范差不多。同样分 2 种类型。

- 功能性质

  功能性质一般指: jquery.js (js 库或框架) 、global.js(全局使用的脚本) 、common.js (部分页面需要用的脚本)

- 针对页面

  针对某个目录的页面：page_category_index.js ,前面的 page 是为了统一所有针对页面而定出来的。可以根据个人的情况把 page 改成自己所定义的单词。

3.15 图片的命名

首先我们这里需要注意的是，切图的时候，可以去参考一些大型的网站如 yahoo 等的切图的方法。一般熟练 css 布局的都会将许多的小图标，背景图片集合到一张图，通过 css 来控制到具体

的元素使用哪个图标或背景。另外目前国外和国内高标准的网站，一般都采用 png 图使用。

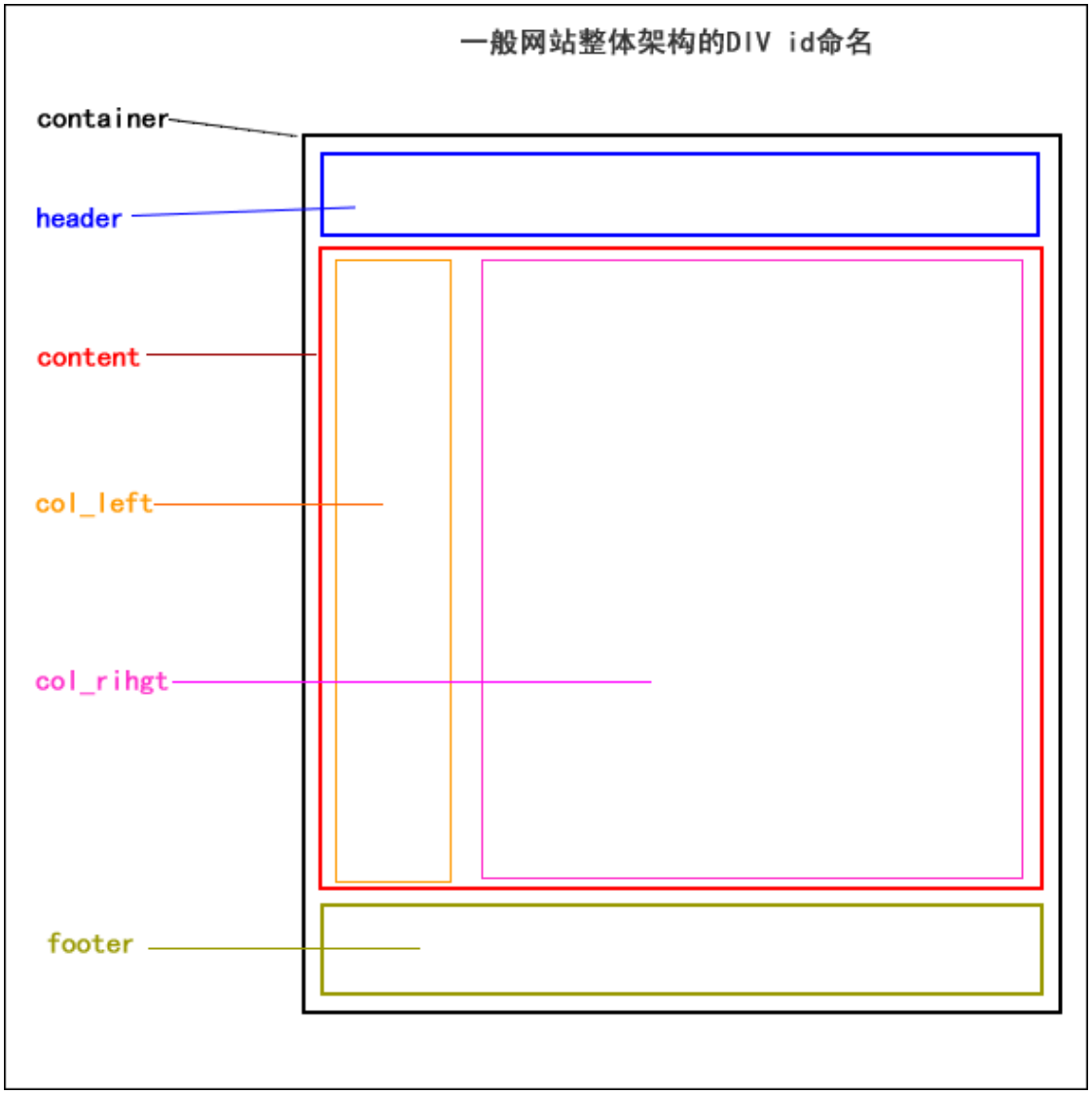
图片根据图片的所处的页面位置名称作用来命名。图片的命名规范化更有利于 css 的编写。可以举几个例子:

- index_header_navtab_bg.jpg
  字面理解到图片是 index 页面 headher 中 导航 TAB 的背景
- products_content_title_icon.gif
  字面理解为 products 页面 content 中 标题的 icon
- index_header_banner_pic.png
  字面理解为 index 页面 header 的 banner 图片

只有当图片的名称语义化了，在你写 css 或者页面需用到图片的时候，可以更快速的找到所需的。如果还沿用以前的 icon01.gif bg.jpg 这类的命名，会给前端开发带来不便，更不用说后期维护了。

3.2 XHTML 元素的命名参考

3.21 布局页面的 div id 命名



3.22 页面的各种元素 id 命名和 class 命名

元素 ID（规则用英文单词命名 id 的功能内容）	元素 class
容器：container 页头：header 登录条：loginBar 标志：logo 侧栏：sideBar 广告：banner 导航：nav 子导航：subnav 菜单：menu 子菜单：submenu 搜索：search 滚动：scroll 页面主体：main 内容：content 列：column 左边列：col_left 右边列：col_right 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 热点：hot 新闻：news 下载：download 注册：register 状态：status 按钮：btn 投票：vote 合作伙伴：partner 友情链接：friendlink 页脚：footer 版权：copyright	.bold{font-weight:bold;} .cl{clear:left;} .cr{clear:right;} .cb{clear:both;} .fl{float:left;} .fr{float:right;} .fn{float:none;} .fontn{font-weight:normal;} .txtl{text-align:left;} .txtr{text-align:right;} .txtc{text-align:center;} .inbox{padding:10px;} .pointer{cursor:pointer;} 当前：current 内容器： inbox 或 innerbox 标题：title 内容：content 输入框：textbox 按钮：btn 当鼠标 over：xxx_over 当鼠标 out：xxx_out