**[网站内容结构化探讨[1]–头部的结构化](http://www.ddcat.net/blog/?p=249" \o "点击阅读《网站内容结构化探讨[1]–头部的结构化》全文。)**

本文不是纯教程而是希望大家一起来讨论，我所给出的，只是我的某个时间的某种思维结果，也许过几天，我会推翻这个结果，所以希望大家能一起思考一起讨论，这样才会有进步！

无意间浏览到一个承接网页制作的公司的网站――[唐宋中国](http://www.tscn.cn/)。

首先网站的设计给我一个很好的印象，因此特意留意了一下它的源代码（做页面的职业病），结果发现了一些问题，而这些问题，可能正好是学习web标准的人常常会遇到的。

[第2部分：主导航的问题](http://www.ddcat.net/blog/?p=251)

懒惰了很久，正在想该写些什么，正好可以以这个网站为例，来探讨一下结构化的一些问题。

特别声明：以此网站为例，纯属偶然，请不要有其他的想法，喵喵妙。

很多人抱怨CSS难学，其实CSS不是难点，真正难的是看似很容易的XHTML――即内容的结构化。

结构化是web标准中的重点，也是难点，因为结构化的结果并不是唯一的，也就是说，相同的内容不同的制作者会产生不同的结构。

但是，由于大部分网页的内容的分类比较固定，例如：新闻列表、标题、内容文字、导航菜单等等，所以其大体的结构化方法也可以基本固定下来，但是无论任何，需要明确的一点是：以**有语意的XHTML标签**来结构化内容，而不是<div>、<span>取代一切。

结构化的第一步，不要把重点放在如何接下来用CSS美化。结构化的重要其实在于，提供基本良好的易用性和SEO基础。  
由于篇幅的关系，不能全部分析，只拿几个比较典型的部分来讨论讨论。

**一、网站头部分析**

**1、头部的结构化**

先看一下网站的头部把具体内容简化后的结构代码：

<div class="top">

<div class="top\_con">

<div class="top\_left">......</div>

<div class="top\_right">

<div class="top\_right\_top">

<div class="top\_right\_top\_l">......</div>

<div class="top\_right\_top\_r">......</div>

<div class="clear"></div>

</div>

<ul id="nav">......</ul>

</div>

<div class="clear"></div>

</div>

</div>

首先，大家可以发现，这个头部代码嵌套了很多的div，如图1所示。

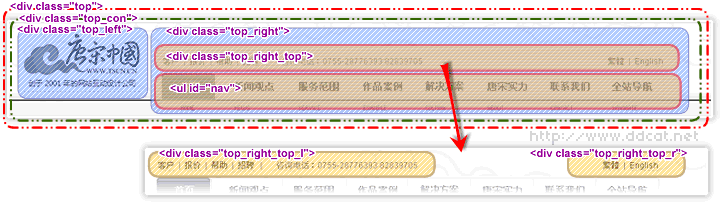


图1 网站头部结构分析

而实际上头部的内容包括哪些呢？如图2所示。



图2 网站头部实际内容

再重新看看图1中布局的目的，如图3所示。

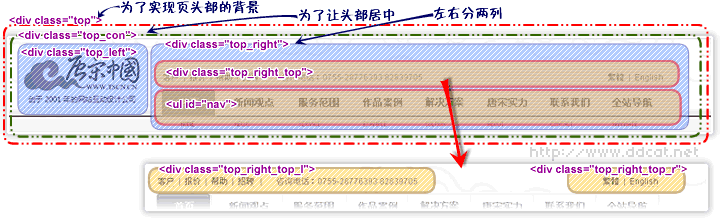


图3 网站头部结构代码用途分析

由图3大家可以发现，大部分的嵌套层只是为了实现背景、左右两列类似这样的效果，这也是在结构化的时候经常遇到的问题――当制作人员按照拿到的设计图开始结构化的时候，往往不自觉地把外观的实现当作了工作重点，所有的结构化都是围绕如何实现设计图的效果来展开，这无疑就将表现与结构混在了一处。

其实，结构化更好的方法是从网站内容策划书开始，但是目前的状况是，很多时候并没有这个东西，也许很多制作人员从来都听说过这个东西，那么笔者推荐一个做好结构化的简单方法：遵循“加法”原则。

先写出最简洁的结构，不考虑表现和行为层的东西，只是把内容用带有语意的标签包裹起来，例如导航使用ul，说明文字用p，然后再考虑这个结构是不是可用于完成“表现”，如果实在无法完成表现层的东西，再添加额外的标签，例如div或者span。

当然，此处说的最简洁的结构，也包括使用div将不同部分的内容区分开，例如页头、网页内容主体、页脚，就可以用3个div来分开。

同时，还要考虑一定的可扩展性，即当设计改变时这个结构仍然可以完成表现――当然这是一个需要经验的工作，那么在初期，也可以暂时不太考虑这个方面。

那么本例中的实际结构，其实可以简化为：

<div id="header">

<div class="logo">......</div>

<ul class="subNav">......</ul>

<div class="telNum">......</div>

<ul class="langSelect">......</ul>

<ul id="mainNav">......</ul>

</div>

此处，大家可以发现几个改动：

**1）选择器命名**

选择器命名也要遵循“表现”和“结构”分离的元素，类似“top\_left”、“top\_right”的命名，很明显是“表现”层的东西，如果某天需要将logo放到右边，如果只修改CSS，那么一个居右的logo，却有个“top\_left”的类名，这岂不是很让人疑惑的一件事儿。

**2）去掉了多余的嵌套div**

精简后的代码，只有2层嵌套结构（不包括其具体的内容元素，例如img或者a），这无疑比上一个结构更容易解读。

其实作为一个有代码洁癖的人来说，还可以更简化这个结构，但是那样不利于以后的可扩展性。

**3）去掉了用于清除浮动的div**

清除浮动的方法有很多，而作者在此不推荐使用增加额外元素的方法，因为如果不需要清除浮动的时候，还要手工去掉这些空元素。

**4）id还是class**

id和class都是最基本的选择器，究竟该用何种选择器，一直存在争论。

首先，要明确的是：id是唯一的，也就是说，同一个页面内只能使用一次（虽然浏览器不会因为多个相同id而报错），因此使用id可以很容易地区分出主要的结构。例如对头部层的定义“id=”header””和主导航“id=”mainNav””。

但是，如果页面需要和程序结合，程序员往往需要使用id来进行操作，因此，有一部分人提倡表现层使用class而行为层使用id。

当然这个问题可以通过人员之间的沟通来解决，因此，笔者推荐一种混合用法，即用id来标示页面的主体结构，例如“header”、“mainContent”、“footer”，而适当地使用class来标示大部分内容。

这并不表示每个元素都要有class或者id，要知道，灵活使用[包含选择器](http://www.ddcat.net/blog2009/?p=218)可以完成很多事情。

那么，如此简单的结构，能够完成页头部的设计要求么？[答案是肯定的](http://www.ddcat.net/web/tscn/testSite/header.html)。