

《WEB标准和网站重构》

- 王本锐
- 2005/07/09 ~ 2005/07/22

www.iselong.com

万千英语族

基础知识：什么是W3C？

- W3C (World Wide Web Consortium, <http://www.w3.org/>) 创建于 1994年, 研究Web规范和指导方针, 致力于推动Web发展, 保证各种Web技术能很好地协同工作。大约500名会员组织加入这个团体, 它的主任Tim Berners-Lee (<http://www.w3.org/People/Berners-Lee/>) 在1989年发明了Web。W3C推行的主要规范有HTML, CSS, XML, XHTML和DOM (Document Object Model)。
- 多年以来, W3C把那些没有被部分会员公司 (如Netscape和Microsoft) 严格执行的规范定义为“推荐” (Recommendations)。自1998年开始, “Web标准组织” (www.webstandards.org) 将W3C的“推荐”重新定义为“Web标准”, 这是一种商业手法, 目的是让制造商重视并重新定位规范, 在新的浏览器和网络设备中完全地支持那些规范。
- 其他的标准组织包括European Computer Manufacturers Association (ECMA) 将 ECMAScript 定义为“标准JavaScript”。

两个问题

- 你用过那些浏览器？

- 常见的网页是通过什么手段进行布局的？

当今过时的网站设计思路的始作俑者

- 大概 1997 年的时候，David Siegel 出版了一本里程碑式的书《Creating Killer Web Sites》，它在当时有限的浏览器功能和 W3C 标准之下，设计出非常华丽的网页效果。(Netscape 2 和 3)
- 这些效果是如此漂亮，以至于到今天，它们还是最流行的网页排版方式。
- 用一句话概括这本书：**用表格和分隔 GIF 可以设计出魔鬼般迷人的站点。**

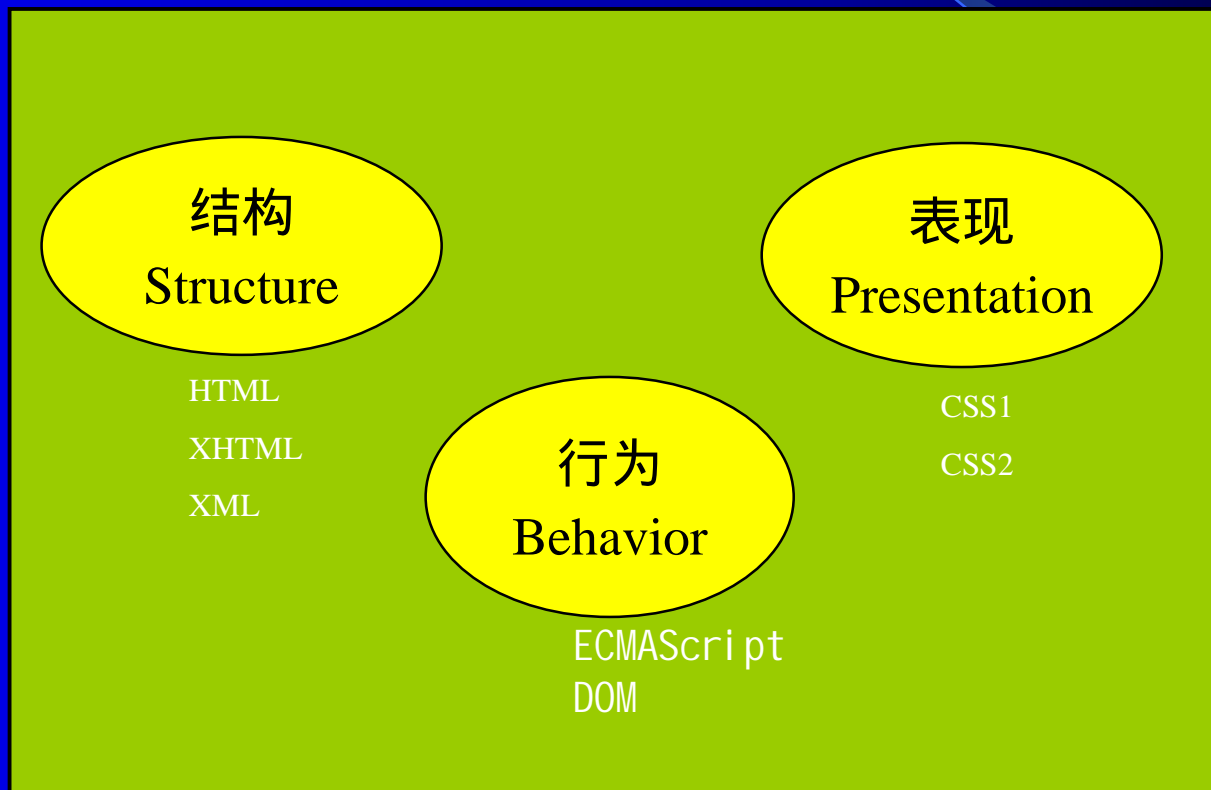
WEB标准(Web Standards)的历史

- 在上个世纪90年代后期，当互联网和web逐渐成为主流时，Web浏览器(包括当时的Netscape 4及以下版本，IE 4及以下版本。行话叫Version 4浏览器)的开发商还没有完全的支持CSS（层叠式样式表，对于Web开发人员来说，他们可以用CSS来控制Html文档的表现）。考虑到CSS1是在1996年制定的，而CSS2是在1998年才制定的，所以这种对CSS支持的不足也是可以理解的。
- 由于浏览器对CSS的支持不够，再加上一些平面设计师的要求（这些要求与他们经常与印刷品打交道有关）导致他们为了控制网页的视觉表现而滥用HTML。一个典型的例子就是，当设计师可以用 `border="0"` 来隐藏表格的边框时，用隐藏表格来控制布局的方法同样被使用。另一个例子是对“transparent”（透明）的使用，同样是不可见，他们却使用空白的GIF图片来控制布局。
- 由于HTML从来就没被用来控制一个文档的表现，导致大量混乱代码、非法代码、浏览器的专用代码和属性就被随意的使用了。“校验”这个词也很少被人问津。对于这些代码来说，标签大杂烩（tag soup）是一个很形象的名字。

什么是“WEB标准”？ <1>

- Web标准，近年来在国外已经得到了很大的重视。尤其是美国颁布了 501号法案以后，皆要求站点必须为残障人士提供与普通人士相当的可访问性支持。要知道，传统的布局方法是很难达到这样的要求的。
- 美国的多数技术性站点都开始向Web标准转型。许多大型站点也已经使用新方法重构。例如就在最近(2005年1月)，微软公司的msn站点使用XHTML Strict+CSS完全重写。
- 网站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：

在WEB标准中构成网页的三个主要部分： 结构、表现、行为



什么是WEB标准<2>: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。

什么是WEB标准<3>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。

什么是WEB标准<2>: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、脚本和表现层对象。

- (2) ECMAScript JavaScript的标准化版本

- 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>）。

名词解释：XHTML

- 2000年底，国际W3C(World Wide Web Consortium)组织公布发行了XHTML 1.0版本。XHTML 1.0是一种在HTML 4.0基础上优化和改进的新语言，目的是基于XML应用。XHTML是一种增强了的HTML,它的可扩展性和灵活性将适应未来网络应用更多的需求。下面是W3C的HTML工作组主席Steven Pemberton 回答的关于XHTML的常见基础问题。
- 问：什么是XHTML?
- 答：XHTML是一种为适应XML而重新改造的HTML。当XML越来越成为一种趋势，就出现了这样一个问题：如果我们有了XML，我们是否依然需要HTML?为了回答这个问题，1998年5月我们在旧金山开了两天的工作会议，会议的结论是：需要。我们依然需要使用HTML。因为大量的人们已经习惯使用HTML来作为他们的设计语言，而且，已经有数以百万计的页面是采用HTML编写的。
- 问：为什么XHTML 1.0相对HTML 4.0独立发展?
- 答：并不是这样。XHTML恰恰就是HTML 4.0的重新组织，(确切的说它是HTML 4.01，是一个修正版本的HTML 4.0，只不过以XHTML 1.0命名发行。)它们在XML里的解释会有一些必要的差别，但另一方面，它们依然非常相似，我们可以把XHTML的工作看作是HTML 4.0基础上的延续。
- 问：XHTML 1.0如何实现XML标准?
- 答：XHTML就是一种XML应用。它采用XML的DTD文件格式定义，并运行在支持XML的系统上。这里要感谢XML的Namespaces功能，浏览器制造商不需要再创造新的私有标签(tags)，他们只需要在XHTML代码里包含XML代码片段，或者XML代码里包含XHTML代码片段。

名词解释：XML

XML即可扩展标记语言（eXtensible Markup Language）。标记是指计算机所能理解的信息符号，通过此种标记，计算机之间可以处理包含各种信息的文章等。如何定义这些标记，既可以选择国际通用的标记语言，比如“HTML”，也可以使用象XML这样由相关人士自由决定的标记语言，这就是语言的“可扩展性”。XML是从SGML中简化修改出来的。

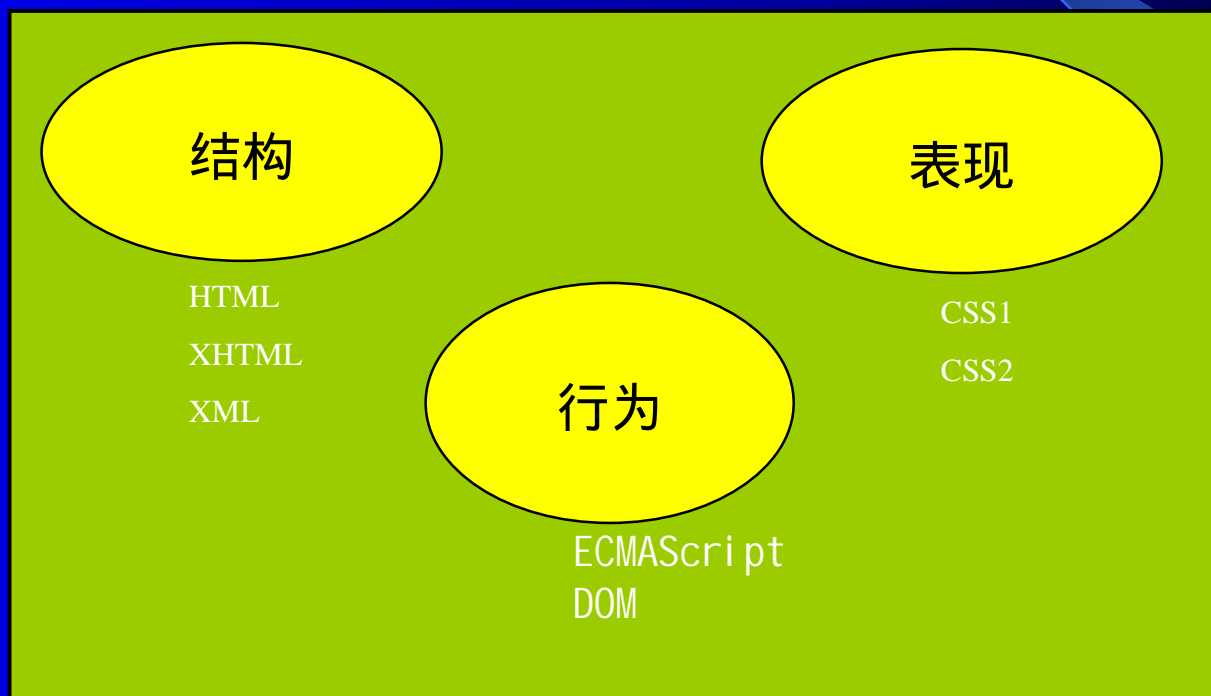
几个容易混淆的概念：

- 1.XML并不是标记语言。它只是用来创造标记语言(比如HTML)的元语言。XML和HTML是不一样的，它的用处途比HTML广泛得多。
- 2.XML并不是HTML的替代产品。XML不是HTML的升级，它只是HTML的补充，为HTML扩展更多功能。我们仍将在较长的一段时间里继续使用HTML。（但值得注意的是HTML的升级版XHTML的确正在向适应XML靠拢。）

名词解释：CSS

- ⑩ CSS是英语Cascading Style Sheets（层叠样式表）的缩写，它是一种用来表现HTML或XML等文件式样的计算机语言。它的定义是由W3C来制定的。

重新检视：在WEB标准中构成网页的三个主要部分：结构、表现、行为



表现层和结构层分离的Web标准设计

2005-7-30

《Web标准与网站建构》

www.iselong.com

14

符合“WEB标准”的浏览器(现状)

- 是指该浏览器能够理解和支持HTML和XHTML , Cascading Style Sheets (CSS) , ECMAScript 及W3C Document Object Model (DOM) 标准。所有这些形成的标准, 允许我们超越表现层的标记、不兼容的脚本语言, 以及它们所造成的永久的淘汰怪圈。主流的“新版本浏览器”包括下面列出的这些以及它们的更新版本:
 - Mozilla(Firefox) 1.0和更高版本
 - Netscape Navigator 6及其更高版本
 - Windows系统下的IE 6及其更高版本
 - Macintosh系统下的IE 5及其更高版本
 - Opera 7
- 新版本的web浏览器的发布, 使得对CSS的支持得到了加强与扩展, 尽管还没有达到它应有的水平。虽然浏览器的开发商对CSS支持的步伐很缓慢, 但是现在已经有许多浏览器选择了支持CSS, 此时, 不应该再有任何理由再像以前那样使用HTML了, 应该让它恢复本来的面貌: 去描述文档的结构, 而不是它的表现。正因为此, 我们现在才能够用CSS来达到其本来的目的了。

创建符合WEB标准的网站已经成为一种可能

按照WEB标准建设网站的目的

- 向后兼容
什么是向后兼容？我们认为是：采用正确的方法设计和建设，发布的任何文档可以正确显示在多种浏览器、平台、设备上，并且能继续在未来发明的新浏览器和设备中工作。开放的标准使之成为可能。
- 节约维护成本
- 方便的维护多个网站
对WEB的内容进行多次的复用
 - XML + RSS

符合WEB标准的网站可以：

- — 在图形桌面浏览器上达到更精确的控制、定位和排版，允许用户使用适合他们的表达方式进行编辑。
- — 可以开发工作在多浏览器和平台的复杂交互行为。
- — 遵守可访问性原则和指南，而不需要牺牲美观、性能或者精巧性。
- — 以前重新设计网站需要几天或者几星期，现在只需要几小时，从而减少成本和避免工作烦恼。
- — 支持多种浏览器，从而不需要争论和考虑多版本的成本，很少或根本就不需要代码分支。
- — 支持非传统的设备，从无线设备到孩子们想像到的、可以上网的智能手机，以及盲人阅读器、屏幕阅读器等残疾人士使用的设备，都不需要再争论开发特殊版本的费用。
- — 为任何网页提交适合打印的版本，不需要建立通常的“专门打印页”或者依赖昂贵的私人出版系统来建立类似的版本。
- — 通过把样式从结构、行为中分离，以及严格的文档结构，易于在高级发布流程中重新设计Web文档。
- — 从老的Web语言HTML转换到更强大的以XML为基础的置标语言。
- — 可以在当前符合标准的浏览器和平台上确保正确地工作，也可以在老浏览器中工作。
- — 保证这样设计的站点将能继续工作在将来的浏览器和设备上，包括那些还没有发明和仍在想像中的设备，这是向后兼容的许诺。

什么是网站重构？

Website Reconstruction

未采用CSS，大量使用HTML进行定位、布局，或者虽然已经采用CSS，但是未遵循HTML结构化标准的站点。

开发工程师
UI设计师

让标记回归标记的原本意义。

通过在 HTML文档中使用结构化的标记以及用CSS控制页面表现，使页面的实际内容与它们呈现的格式相分离。

为什么要对网站进行重构？

- 使页面加载得更快速
- 降低带宽带来的费用 - 节约成本
- 让你在修改设计时更有效率而代价更低
- 帮助你的整个站点保持视觉的一致性
- 更利于搜索引擎的检索（符合SEO的规范）
- 令站点更容易被各种浏览器和用户访问(包括手机、PDA和残障人士使用的文字浏览器)
移动互联网：中国的PC保有量约7000万台，但手机保有量已经接近3亿部
- 兼容不容忽视的 Mozilla 系浏览器（Firefox 份额）
- 在世界上越来越多人采用 Web 标准时，它还能提高你的职场竞争实力（事实上也就是降低失业的风险）。

比较优势：非常节省人力且大幅提高效率

- 把页面中的外观标记去除以后，重新设计现有的站点和内容将变得非常节省人力(同时便宜得多)。如果想改变站点的输出，你只需要改动一下样式表就行了，而完全不必改动页面本身。
- 以XX现有的CMS为例：
 - 改版需要重新设计模板，然后利用新的模板对所有页面进行重新生成，发布并上传成千上万个文件。
 - 采用WEB标准之后，改版只需要修改一个CSS文件，无需重新发布、上传上成千上万个文件。

比较优势：节约带宽且页面下载的更快

- 运用 Web 标准来缩减网页文件的大小，这样用户将不必访问每页都下载一次外观的数据。控制输出效果的样式表(.css 文件)是由用户的浏览器自动缓存的。
- 缩减文件大小意味着更快地载入和更少的流量费用。
- 网站首页大小
- 采用WEB标准后保守估计减少10k
- $10K \times 300,000,000pv = 3,000,000,000k = 2861G$

比较优势：帮助你的整个站点保持视觉的一致性

- 运用 Web 标准还能使保持整个站点的视觉一致性变得非常简单。因为页面使用同样的 CSS 文档来进行排版，所以它们会被格式化为相同的风格。
- 这样加深了你的站点给人的印象还能使它更便于使用(user friendly)。

比较优势：一次编辑，到处应用，人人皆然

- 运用 Web 标准使得你的站点对残疾人和使用手机或 PDA 的 Web 浏览者更具亲和力。
- 使用读屏器* 的访问者 (或者连接速度比较慢的访问者) 不希望在下载了数不尽的表格单元和分隔图片后才获得我们页面的实际内容。
- 换句话说，把内容与格式分开使得你的内容与设备无关。
- 注：读屏器(screen reader)是盲人等无法观看显示器的用户使用的一种把当前屏幕上的内容朗读出来的设备。

比较优势：更利于搜索引擎的检索（符合SEO的规范）

- 保持网页的亲合力、尽量减少它的标记、并合理的运用网页头部的标签同样会帮助你提高搜索引擎上的排名(Rank)。
- 降低代码/内容比、在头部标签中加入关键字，把网页顶部的标志图片改为文本.....这些行为都可以帮助你的站点在搜索引擎中获得更好的搜索结果。
- SEO -> 另外一项很有意思的工作。

矫枉不必过正：仍然可以运用表格

- 你仍然可以在需要时运用表格，别用得太多就行了
- 这对于那些固执于老式的第四代浏览器的浏览者 会比较好接受，而在直到他们回心转意之前，其他人至少也能接受。
- 运用表格的技巧：后面讲述

传统WEB的弊端：表格带来的问题

- 把格式数据混入你的内容中。
- 这使得文件的大小无谓地变大，而用户访问每个页面时都必须下载一次这样的格式信息。
- 带宽并非免费。
- 这使得重新设计现有的站点和内容极为消耗劳力（且昂贵）。
- 这还使我们保持整个站点的视觉的一致性极难，花费也极高。
- 基于表格的页面还大大降低了它对残疾人和用手机或 PDA 浏览者的亲和力。

对网站进行重构需要具备那些知识

- 策划人员必须提出网页的结构
- UI设计师需要对所设计的内容、结构进行充分的理解
- 程序开发人员也需要了解WEB标准的知识
- 具备HTML基础
- 现在开始学习CSS

如何进行网站重构？

两项最主要的工作

- 一. 恢复HTML本来的面貌：用HTML去描述文档的结构。
- 二. 掌握CSS的应用：利用CSS对内容进行布局。

结构化HTML<1>

- 我们在刚学习网页制作时，总是先考虑怎么设计，考虑那些图片、字体、颜色、以及布局方案。然后用Photoshop或者Fireworks画出来、切割成小图。最后再通过编辑HTML将所有设计还原表现在页面上。
- 如果你希望你的HTML页面用CSS布局(是CSS-friendly的)，你需要回头重来，先不考虑“外观”，要先思考你的页面内容的语义和结构。
- 外观并不是最重要的。一个结构良好的HTML页面可以以任何外观表现出来，CSS Zen Garden(www.csszengarden.com)是一个典型的例子。CSS Zen Garden帮助我们最终认识到CSS的强大力量。 -> DEMO
- HTML不仅仅只在电脑屏幕上阅读。你用photoshop 精心设计的画面可能不能显示在PDA、移动电话和屏幕阅读机上。但是一个结构良好的HTML页面可以通过CSS的不同定义，显示在任何地方，任何网络设备上。

结构化HTML<2>

- 首先要学习什么是“结构(structural)”，一些作家也称之为“语义(semantic)”。这个术语的意思是你要分析你的内容块，以及每块内容服务的目的，然后再根据这些内容目的建立起相应的HTML结构。
- 如果你坐下来仔细分析和规划你的页面结构，你可能得到类似这样的几块：
 - 标志和站点名称
 - 主页面内容
 - 站点导航(主菜单)
 - 子菜单
 - 搜索框
 - 功能区(例如购物车、收银台)
 - 页脚(版权和有关法律声明)
- 我们通常采用DIV元素(TIPS:可以把DIV理解为“盒子”或者“容器”)来将这些结构定义出来：
- ```
<div id="head"> </div>
<div id="content"> </div>
<div id="globalnav"> </div>
<div id="subnav"> </div>
<div id="search"> </div>
<div id="shop"> </div>
<div id="foot"> </div>
```
- 这不是布局，是结构。
- 这是一个对内容块的语义说明。当你理解了你的结构，就可以加对应的ID在DIV上。DIV容器中可以包含任何内容块，也可以嵌套另一个DIV。内容块可以包含任意的HTML元素---标题、段落、图片、表格、列表等等。
- 根据上面讲述的，你已经知道如何结构化HTML，现在你可以进行布局和样式定义了。每一个内容块都可以放在页面上任何地方，再指定这个块的颜色、字体、边框、背景以及对齐属性等等。

[WEB标准与网站建设]

# 例子：表格是CSS布局的对比

- [一个看起来不错的表格.htm](#)

# 过渡性的设计

- 使用 margin 和 padding 来代替多余的表格单元和间隔 GIF。
- 使用 link 和 @import 来载入样式。前者用于早期浏览器，后者给现在的浏览器。
- `<link href="basic.css" rel="stylesheet" type="text/css">`
- `<style type="text/css" media="screen"><!--`
- `@import url(modern.css) screen;`
- `--></style>`
- 一些实例：
  - Santa Cruz Montessori School <http://www.scms.org/ycc.html>
  - k10k <http://k10k.net/>
  - Fox Searchlight Pictures <http://foxsearchlight.com/>
- 想了解更多关于过渡性设计的信息，请阅读 Eric Meyer on CSS 的第一章。和 Jeffrey Zeldman 的 Designing with Web Standards 的大部分内容。



# CSS 排版：并没有想象的难

- 大部分用户使用的浏览器都有很好的 CSS 支持，当然它们也有个各自的脾性，不过只要你习惯了它们，就能够处理得很好。
- 编写 CSS 代码很简单。
- 每条 CSS 规则都有一个选择符和一个声明。声明是由 属性名(property) 和 属性值(value) 组成的。属性名一般使用连字符(-)连接。
- ```
body {margin:0; padding:0}
```
- ```
.related {float:right; width: 15em; margin-left:1em; margin-bottom: 1em; color:blue}
```
- ```
#footer {color: gray; font-size: 0.6em; line-height: 1.2em; background-color: white; margin: 0}
```

结构化标记：所写即所想，所想即所写

- 即便书写 CSS 很简单，使用 CSS 来排版却真的需要一种和我们以往有些不同的思维方式。
- 我们排版时考虑的不是“这个东西放在这儿，那个东西放在那儿”，而是页面中信息的类别和信息结构。
- 我们用 `<h1>` 标记来标示最重要的头条；次一级的条目则用 `<h2>` 来标记，以此类推；而段落则放在 `<p>` 标记中。
- 这就是我们称之为“结构标记”或“语义标记”的东西。
- 你的内容将不放在表格和表格元素中，取代它们的是 `div` 元素。还要给你的 `div` 元素安排一个 `id` 或 `class`，不过这是为了描述它们的内容或功能，而非它们的外观。

避免 ed 与
eakfast 标记

- 且想想为何你希望某个对象按特定的方式出现；它有什么含义？你的标记能够也必须表达意思，即便对于那些无法观看页面的人来说也是如此。语义性的标记使得你的页面对一切受众都更有亲和力，这还包括了搜索引擎。
- 当你把某个对象设为斜体时，大概要么就是想强调它，要么是想引用某个书中标题吧？如果是前者，应该用 ；如果是后者，其实得用 <cite>。
- 如果某个对象被标记为粗体，事实上它应该被标记的是， 。
- 如果你希望在某处添加一个换行，这大概是开始了一个新的开头。如果不是开头，会不会是某种在你的站点中出现过的class (类型)？在上面两种情况下，你都应该用 CSS 替换
。
- .foo {display:block}
- 如果希望了解更多，请阅读 [Bed and BReakfast markup \(B&BR\)](#)，[Tantek ?elik](#) 作品。

如何用css做导航条

- DEMO

当然，CSS 总会有些事情做得不如表格好。

- 略
- 瑕不掩瑜☺

从玩弄表格技巧到遵循 Web 标准：通盘考虑

- 首先，你得考虑好一个移植策略。是一口气把整个站点移植了呢，还是逐个逐个部分地一步步完成？
- 检查一下，找出会因为移植而受益的页面和分支。自然而然地，从首页、新产品通告之类的页面开始检查会比较好。
- 分析你的站点的内容分别属于下面的哪些类型：
 - 产品信息
 - 价格信息
 - 公司信息
 - 报表
 - 服务信息
 - 投资信息
 - 购物篮
 - 用户论坛
 - 等等

分解你的页面，找出逻辑结构

- 如果你已经把站点的内容分好了类，那么是时候分析每个页面，把它们按照逻辑关系进行分解了。
 - 主导航条
 - 子导航条
 - 页眉与页脚
 - 内容(以新闻为例：标题、副标题、作者、信息源、发布时间、新闻正文等)
 - 相关信息
 - 其他
- 注意分析由嵌套表格、空白分隔符和边框元素组成的表格结构(因为我们希望把它们替换为用 div 标记组织的那种简单得多的表格结构)。

改变古老的WEB建设习惯，注意以下标签

- 一旦你分析好页面的结构，就可以着手揭开表象，分析你现在的页面代码中可以转换成结构化标记的那些地方。
- 消灭所有的 标记和间隔 GIF ！（W3C的说法是：请不要使用这种过时垃圾）
- 搞定 ed 和
eakfast 标记。
- 去除表格中的那些外观标记 (bgcolor, background, 等等)。
- 把纯粹用来表示外观的那些 CSS 调用 (比如) 改成结构化的标记。（Tantek ?elik 在他的文章 [A Touch of Class](#) 里详细讨论了这方面的内容。）

把描述外观的标记改为描述结构的标记

- 你可以使用“查找加替换”(或者正则表达式), 不过最好的方法还是在浏览器里打开这个页面, 然后把文本复制粘贴到你的 HTML 编辑器里。
- 关键还是要用结构化的方式来思考! 仅仅把 `` 标记替换为 `` 是远远不够的。
- 最重要的条目是哪个? 用 `<h1>` 来标记它。次要一点的就用 `<h2>`, 依此类推。用 `<p>` 来标记段落。把导航条标记为无序列表。
- 选定一个 DOCTYPE 来使用。(我们推荐 XHTML transitional, 除非你是此道高手, 否则别用 XHTML strict。)

把页面的各个元素合理分布在 div 中

- 把你的主导航条放在一个 id 属性设为 mainnav 的 div 中，子导航条则放在一个 id 或者 class 是 subnav 的 div 中。而页脚类似这个形式：`<div id="footer">`，至于整个页面，它被放在 `<div id="content">` 里边。
- 现在的页面还比较简陋，但只要你开始给样式表添加规则，它就会马上变得可爱起来。

开始为我们准备好的内容编写CSS

- 首先，给每个 div 设置一个边界。例如：`div {border: 1px dotted gray; padding: .5em}` 这可以帮助你完完整整地看到里边的内容，也不再需要滚动屏幕了。
- 首先为以定义的元素编写 CSS (`<html>`, `<body>`, `<p>`, `<h1>`, `<h2>`, ``, ``, 等等。)
- 尽量运用上下文相关或者有递归下降结构的选择符。这可以使得你的标记更清晰。例如：`#subnav li {border: 1px solid black; padding: .5em; display: inline}` 将能够只能影响你的子导航栏的 div 中的列表项。
- 记得在尽可能多的浏览器下进行测试。

本次讲座未深入阐述但需要了解的知识

- XHTML
- CSS语法 熟练掌握
- W3C DOM 了解
- ECMAScript 掌握

总结：本次讲座的主要内容

- 恢复HTML本来的面貌：用HTML去描述文档的结构。
- 掌握CSS的应用：利用CSS对内容进行布局。



网站重构要缓行，不能一蹴而就。

- 对于已经采用老的table定位的网站来说，重构是一场噩梦
- 1、UE(user experience)人员的再学习是一个过程
- 2、不是什么网站都适合重构
- 3、原有页面设计的不规范，比如div滥用等。。。会导致重构不可能实现，或者实现出来的效果很差。
- 4、重构对cms来讲也是个挑战啊，编辑的知识可能不够用了。也要update了。
- 还有，就是现在的可视化编辑工具，是否都已经支持div + css

其它信息：资料

- 英文资源：
 - www.w3.org w3c官方网站
 - www.cssvault.com 按照web标准建设的优秀网站集锦

- 中文资源：
 - www.w3cn.org 网页设计师：中文资料 阿捷
 - <http://www.onestab.net/>
 - <http://www.blueidea.com/bbs/newsdetail.asp?id=1671022&page=1&posts=&Daysprune=&lp=1> 重构之美 - 迎接网站标准化设计的来临

- 其它参考资料：
 - 《网站重构-应用Web标准进行设计》

其它信息：用CSS排版的站点范例

- [CSS Zen Garden](#)
- [Adobe Studio](#)
- [Wired News](#)
- [ESPN](#)
- [Quark](#)
- [Inc.com](#)
- [PGA Open Championship](#)
- [phish.com](#)
- [CSS Edge](#)

其它信息：国内外web标准的采用情况

- 阿里巴巴中文版用web标准重构完毕。(2005/04)
- MSN按xhtml1-strict标准重构完毕。(2005/04)
- 网易首页按web标准重构完毕。(2005/03)
- 网易的新闻频道首页按web标准重构完毕。(2005/02)
- Techweb.com.cn (2005/01)
- 闪客帝国用web标准进行了网站重构。(2004/10)
- CSDN开设设计革命专题 (2004/05)
- blogger.com 采用css和xhtml strict 1.0标准重新设计完成。在stopdesign 上还介绍了设计中用到的一些技术。(2004/05)
- 2004 SXSW Web设计大奖CSS Zen Garden获得了“Developer’s Resource”和“Best of Show”两个奖项。(2004/05)
- MP3.com网站采用web标准重新设计完成 (2004/05)
- 美国国会图书馆(Library Of Congress)网站采用web标准重新设计完成 (2004/05)

Q&A

谢谢大家！

- 王本锐