Web标准及浏览器兼容性解决方案

一：优秀的网页设计师

经常有人会问做为一个优秀网页设计师要具备哪些方面的知识以及技能？

1.掌握一款良好的图形图形处理软件（例如：Photoshop，Fireworks ，AI...）

2.掌握可视化编辑工具或者编辑器（例如：TopStyle，Dreamweaver ，EditPlus...）

3.掌握最基本的实质性知识，也就是网页的三层结构Xhtml,CSS,Javascript(结构，表现，行为)

4.掌握网站据有分析策划的能力

5.掌握具有美术的一些基础配色知识最起码不要让网页的颜色太难看

6.掌握一款网络动画软件（例如Flash）

如果你可以达到以上的要求，可以说的上你算一个网页设计师，不要认为一个网页设计师只会用PS做出一张图片然后用PS直接分割另存为html页面的就是网页设计师，一个会用Dreamweaver可视化软件做出一个垃圾网站（传统的table+gif技术实现的网站）的就叫网页设计师。

二：浏览器内核

浏览器内核

通常所谓的浏览器内核也就是浏览器所采用的渲染引擎，渲染引擎决定了浏览器如何显示网页的内容以及页面的格式信息。不同的浏览器内核对网页编写语法的解释也有不同，因此同一网页在不同的内核的浏览器里的渲染（显示）效果也可能不同，这也是网页编写者需要在不同内核的浏览器中测试网页显示效果的原因。

主流浏览器内核

Trident：IE浏览器使用的内核,该内核程序在1997年的IE4中首次被采用，是微软在Mosaic代码的基础之上修改而来的，并沿用到目前的IE7。Trident实际上是一款开放的内核，其接口内核设计的相当成熟，因此才有许多采用IE内核而非IE的浏览器涌现（如 Maxthon、TheWorld 、TT、GreenBrowser、AvantBrowser等）。

Geckos： Netcape6开始采用的内核，后来的Mozilla FireFox也采用了该内核，Geckos的特点是代码完全公开，因此，其可开发程度很高，全世界的程序员都可以为其编写代码，增加功能。

Presto： 目前Opera采用的内核，该内核在2003年的Opera7中首次被使用，该款引擎的特点就是渲染速度的优化达到了极致，也是目前公认网页浏览速度最快的浏览器内核，然而代价是牺牲了网页的兼容性。

Webkit：Safari浏览器使用的内核。Webkit引擎包含WebCore排版引擎及JavaScriptCore解析引擎，均是从KDE的KHTML及KJS引擎衍生而来，它们都是自由软件，在GPL条约下授权，同时支持BSD系统的开发。所以Webkit也是自由软件，同时开放源代码。在安全方面不受IE、Firefox的制约，所以Safari浏览器还是比较安全的浏览器。(Google的Chrome浏览器也是采用Webkit)

注意，牺牲了网页的兼容性是说牺牲不好的网页兼容，因为Opera浏览器以开始就对标准支持的非常良好，所以对写法要求比较严格，那些不符合标准的网站在Opera下自然也就不能运行。

三：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。

　　1．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。

　　（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。

　　2. 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。

　　3.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

四：采用Web标准的好处

　　对于访问者：

　　● 文件下载与页面显示速度更快。

　　● 内容能被更多的用户所访问（包括失明、视弱、色盲等残障人士）。

　　● 内容能被更广泛的设备所访问（包括屏幕阅读机、手持设备、搜索机器人、打印机、电冰箱等等）。

　　● 用户能够通过样式选择定制自己的表现界面。

　　● 所有页面都能提供适于打印的版本。

　　对于网站所有者：

　　● 更少的代码和组件，容易维护。

　　● 带宽要求降低（代码更简洁），成本降低。举个例子：当 ESPN.com 使用 CSS改版后，每天节约超过两兆字节（terabytes）的带宽。

　　● 更容易被搜寻引擎搜索到。

　　● 改版方便，不需要变动页面内容。

　　● 提供打印版本而不需要复制内容。

● 提高网站易用性。在美国，有严格的法律条款（Section 508）来约束政府网站必须达到一定的易用性，其他国家也有类似的要求。

五：如何学DIV+CSS技术

有好多人也看了，但是没有什么进步，我分析主要有以下几个原因：

原因一：压力不够，因为有不少人比如美工学CSS，因为是看别人学，所以学之，不学就没优势！这样的压力是很小的。

原因二：只是业余学习，这样学还只是玩一玩，因为你不一定非要以这门技术吃饭。

原因三：方法不对，有的人只是在看教程，但他不动手去做，看的太多，做的太少，所以有了实践你才能把理论的东西揉进去。

六：掌握CSS的四个学习阶段

有一次我问一个网友，学CSS难吗？她说：“不难”我说你学多久了？她说：“刚学”。其实她说的也没错！如果有人问我学CSS 难吗？我会说：“难！”为什么我会说难呢？

学CSS可以分成以下几个阶段：

阶段一：不去想浏览器的兼容性问题的情况下能做出页面，但页面里到处用TABLE的 模式DIV写出的页面。

阶段二：想到了兼容性问题，但无法提前预防，只会修补问题，大量使用HACK技术（我对HACK技术只是知道，但用的很少）

阶段三：可提前预防BUG，但样式表中大量应用了ID,CLASS，CSS译过来是重叠样式表，比如“DIV P SPAN”。这句代码是标明DIV子元素中P的子元素SPAN，这样写就可以定义SPAN的样式了，不用在SPAN中加入CLASS了。这也就是CSS的优点所在，为什么不好好利用而一定要定义一个CLASS呢！

阶段四：这一阶段是最难的，良好的HTML语义结构、合理的CSS、可重复利用的样式。良好的语义有这方面的文章,合理的CSS，这两方面以后一定会写专题来研究。

七：用什么软件编辑CSS代码

这个问题好象是问过频率最高的！先说我自己吧，刚开始是用DreamWeaver写代码，由于DreamWeaver软件本身是一个很耗CPU和内存的软件，运行时编辑的速度就可想而知了。走了好长一段时间之后我才过度到了“半手写”状态，“半手写”就是利用DW提供的代码提示功能，当你写“F”时有关“F”的样式就会立刻在后面提示出来，方便你选取。如果再写上几个字母，你要的那个样式就很快找到了，这样可以加快你的写样式的速度，但有一个坏处就是如果你脱离了代提示的编辑软件你就不会写样式了，比如你在程序员那里修改微调样式时，他用的可不一定是你用的那个软件，不一定有代码提示功能，所以你只能回到自己的机器上编辑好了然后发给程序员！这样做显然是不专业的！所以如果想成为优秀的人就要做到够专业！

八：结构和表现分离

一定要做到从思想上分离，书写代码的时候

九：web标准有关XHTML代码规范

　　1.所有的标记都必须要有一个相应的结束标记 :

　　以前在HTML中，你可以打开许多标签，例如<p>和<li>而不一定写

　　对应的</p>和</li>来关闭它们。但在XHTML中这是不合法的。XHTML要求有

　　严谨的结构，所有标签必须关闭。如果是单独不成对的标签，在标签最后加

　　一个"/"来关闭它。例如:

　　<br /><img height="80" alt="网页设计师"

　　src="../images/logo\_w3cn\_200x80.gif" width="200" />

　　2.所有标签的元素和属性的名字都必须使用小写 :

　　与HTML不一样，XHTML对大小写是敏感的，<title>和<TITLE>是不

　　同的标签。XHTML要求所有的标签和属性的名字都必须使用小写。例如：

　　<BODY>必须写成<body> 。大小写夹杂也是不被认可的，通常dreamweaver

　　自动生成的属性名字"onMouseOver"也必须修改成"onmouseover"。

　　3.所有的XML标记都必须合理嵌套 :

　　同样因为XHTML要求有严谨的结构，因此所有的嵌套都必须按顺序，以

　　前我们这样写的代码：

　　<p><b></p>/b>

　　必须修改为：

　　<p><b></b></p>

　　就是说，一层一层的嵌套必须是严格对称。

　　4.所有的属性必须用引号""括起来:

　　在HTML中，你可以不需要给属性值加引号，但是在XHTML中，它们必须被加

　　引号。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

　　&apos;，例如：

　　<alt="say&apos;hello&apos;">

　　5.把所有<和&特殊符号用编码表示 :

　　任何小于号（<），不是标签的一部分，都必须被编码为& l t ;

　　任何大于号（>），不是标签的一部分，都必须被编码为& g t ;

　　任何与号（&），不是实体的一部分的，都必须被编码为& a m p;

　　注：以上字符之间无空格。

　　6.给所有属性赋一个值 :

　　XHTML规定所有属性都必须有一个值，没有值的就重复本身。例如：

　　<td nowrap> <input type="checkbox" name="shirt" value="medium"

　　checked>

　　必须修改为：

　　<td nowrap="nowrap"> <input type="checkbox" name="shirt"

　　value="medium" checked="checked">

　　7.不要在注释内容中使“--” :

　　“--”只能发生在XHTML注释的开头和结束，也就是说，在内容中它们不再

　　有效。例如下面的代码是无效的:

<!--这里是注释-----------这里是注释-->

8.else

XHTML 1.0文件类别宣告的正确写法 (不可小写)

　　过度标准（Transitional）

　　<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

　　"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

　　框架标准（Frameset）

　　<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"

　　"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">

　　严格标准（Strict） 包含以上须注意的问题,还有其他更严格的标准

　　<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"

　　"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

头文件问题

　　所有的网页头文件都一律都改为标准形式，写法如下：

　　<head>

　　<meta http-equiv="content-type" content="text/html; charset=gb2312" />

　　<meta http-equiv="content-language" content="zh-cn" />

　　<meta name="keywords" content="..." />

　　<meta name="description" content="..."/>

　　<title>...</title>

　　</head>

不允许使用target="\_blank"

同一个id选择器不可重复使用

JavaScript写法

　　Javascript我们通常会写为

　　错误 <script language="javascript">

　　W3C标准必须为程式指定类型type=text/javascript,所以要写为

　　正确 <script type="text/javascript">

　　或者 <script language="javascript" type="text/javascript">

　　载入外部.js独立档案的写法

　　正确 <script type="text/javascript" src="script.js"></script>

图片标签加上文字说明alt="说明"

背景音乐不允许使用 bgsound 标签

不允许使用框架标签<IFRAME>

正确使用CSS样式表

　　一定要放在<head></head>之间

　　<style type="text/css">

　　<!—

　　@import Url(“style.css”);

　　body{font-size:9pt;}

　　-->

　　</style>

非标签一部分的符号以编码表示

　　表单内包含以下符号也必须用编码表示

　　< 以 &lt; 表示

　　> 以 &gt; 表示

　　& 以 &amp; 表示

　　程式中的连结 & 也要改用 &amp;

　　错误 <a href="foo.cgi?chapter=1&ion=2">

　　正确 <a href="foo.cgi?chapter=1&amp;ion=2">

十：浏览器BUG

　　1.不同浏览器默认样式表不同

　　2.IE double margin

九：初学学习Web标准的误区

　　1.不是为了通过校验才标准化。

　　web标准的本意是实现内容(结构)和表现分离，就是将样式剥离出来放在单独的css文件中。这样做的好处是可以分别处理内容和表现，也方便搜索和内容的再利用。

　　W3C校验仅仅是帮助你检查XHTML代码的书写是否规范，css的属性是否都在CCS2的规范内。代码的标准化仅仅是第一步，不是说通过的校验，我的网页就标准化了。我们不是为了虚名，或者向别人炫耀：“看我的页面通过了校验”而去标准化，我们的目的是为了使自己的网页设计工作更有效率，为了缩小网页尺寸，为了能够在任何浏览器和网络设备中正常浏览。

　　请大家沉下心来，仔细研究和理解web标准的内涵。网络上已经充满了不要浮躁和虚荣，请不要将这种风气带到web标准应用中来，这样做只会影响和损害web标准的推广。

　　2.不要用传统的表格思维来套div

　　“CSS布局就是将原来用table的地方用div来替代，原来是表格嵌套，现在是DIV嵌套。”这种观点是错误的！

　　请跳出原来表格布局的禁锢，抛弃一个td接一个td放置图片和内容的思维方式。我们上面说过web标准的目的是分离内容和表现，你可以这样思考，页面里有的仅仅是内容，没有修饰的情况下，它看上去就是一张白白的页面，上有一些文字和图片(这个图片是指内容中的图片，是有真实意义的图片)。这些文字图片仅仅是依次罗列下来，只有结构，没有任何样式。然后加入表现，将所有修饰的图片作为背景，用C SS来定义每一块内容的位置、字体、颜色等。

　　这样制作的页面才是内容与表现分离的，就是说，当你抽掉css文件，剩下的就是干净的内容。这样才能在文本浏览器中阅读，才能在手机、PDA中阅读，才能随时修改CSS实现改版。

　　3.不必为每块内容都建立一个id

　　有网友抱怨css文件太大、太复杂。仔细看他们的代码，发现原因是他们对每块内容、甚至每句话都定义一个div，建立一个ID。这一方面是对样式表应用的不熟悉，另一方面也表现出对web标准没有彻底理解。

　　我们知道内容都是有结构的(如果不明白，请阅读：[url=http://www.w3cn.org/article/tips/2004/43. html]理解表现和结构相分离 ),相同的结构的内容我们可以用同一个样式来定义，比如相同级别的标题、正文、图片。对于多次引用的样式可以用class来定义，不需要每个都用id；另外也不是说一定要用，你完全可以用来代替，同样都是块级元素，一样有盒模型的七个参数，仅仅方便浮动。

　　至于仅仅为了行高、间距、一个修饰图片而增加额外的div，我想随着对CSS的应用和理解，你很快就可以省略掉这些。我们反对用DIV的嵌套取代table的嵌套，这没有任何意义，不能体现内容的结构化。希望大家多研究CSS，写出最简练，最有效的样式表。

　　4.不要因为一点挫折就轻言放弃

　　我想这是一个态度和处事方法问题，这不仅仅针对学习web标准。学习任何新知识，接受新概念，都有困难，特别是需要你改变原来已有的习惯和思维时。唯一的问题就是：你认为学习web标准是否值得？你可以问问自己：你是否认可web标准带来的这些好处(简化代码、加速开发、压缩文件尺寸、提高下载速度、更好的易用性、获得更多用户、更易于维护、多平台兼容)？你是否觉得世界通用的XML离你还很远？你是否认为现在掌握的HTML知识就已经足够了？

　　自从99年以后，网页制作技术在国内就没有怎么进步和变革过(除了flash的兴起)，而在国外前进的脚步就一直没有停止过，xhtml1.0、xhtml1.1、xml、xsl...，W3C和各大软件公司都在不断的研究和推进web技术，使它更方便、更有效、更强大。web标准的应用和推广国外已经开展了八年多(以zeldman的web标准组织网站为起点)，虽然困难重重，但正在逐步被人们接受和认可。为什么我们依然陶醉在网站表面的“繁华”上，不去思考页面背后本质的东西。