

Web 标准中的常见问题

Common Problems of Web Standard

张子阳

www.tracefact.net

jimmy_dev@163.com

引言

大概在 2004 年的时候，Web 标准的概念藉由一本名为《网站重构》的书开始被国内人所了解。随后的几年中，其更少的代码量、更好的搜索引擎友好性、更好的浏览器兼容性使得 Web 标准迅速受到重视并逐渐普及，网易、新浪等各大网站相继采用 Web 标准进行网站重构。

在今年，我浏览了不少采用Web标准进行重构的网站，并在国内著名设计与制作论坛 [蓝色理想](#) 浏览了很多网页制作人员发表的言论。发现许多制作人员对Web标准的理解不够深入，有的地方甚至存在误区，在这篇文档中，我将就我所了解的问题做一点分析，以供参考。

NOTE: 很可惜，这个文档中提到的范例页面全部都失效了，重新制作这些页面很花时间，缺少范例可能会对某些地方的理解造成影响，我多加了注释。

以访客为中心的 可访问性 和 可使用性

如果有人问我，国外的网页设计师和国内的最大区别是什么？我会毫不犹豫的告诉他，国外的设计师最注重的是网页的可访问性(Accessibility)和可用性(Usability)。而国内的设计师则很少考虑，甚至不考虑这方面的问题。

什么是可访问性(Accessibility)?

可访问性就是说你的网页可以在任何情况下、被任何人，特别是一些受到某种限制的访客所访问到。受到某种限制的意思是说：视力障碍的访客、只使用键盘不使用鼠标的访客。或者在浏览器不支持，或仅支持很少的 css 和 javascript 的情况下，页面依然可以访问。

以页面一来说，注意到左边有“相关链接”，你可能会奇怪，在上面的导航中已经可以访问到“相关链接”里的全部内容，这里为什么还要多此一举呢？为了好看么？为了可访问性！假设你只使用键盘会怎样(只能通过 TAB 遍历所有的链接和表单项)？你在这个页面该如何访问到 Music 目录下的其他页面？NO WAY！因为根本就没有鼠标的 onmouseover 和 onmouseout 事件。

NOTE: 由于页面失效，我在这里多做一点说明：这个页面在上方有一个很常见的鼠标悬停显示二级下拉菜单的导航条，而在页面左边，我又放置了一些链接，其内容与下拉菜单里的链接相同。

什么是可用性(Usability)?

举一个简单的例子：在我给大家展示的这个页面里，如果所有的 JavaScript 脚本都失效，页面会怎样？如果你想选择所有的复选框，你将不得不一个一个的去点击它，它的可用性降低了，但是可访问性无损。如果你想按时间长度排序表格，则没有办法完成，但是丝毫不影响你了解每首歌曲的长度。

看了这个例子，你应该明白可访问性和可用性的区别：可访问性关心的是某个页面是否能被用户访问到；可用性关心的是在诸如 JavaScript 等脚本失效后，页面的功能是否依然有效。

NOTE: 同样是由于页面失效的原因，我想再举一个可用性的例子，我们在提交表单的时候，很多时候都会加一个 Ctrl + Enter 直接提交的功能。这样的做法就是提高可用性的一个典型例子，而如果你的表单只能通过 Ctrl + Enter 提交，但用户的浏览器不支持 JavaScript，那么他将无法提交表单，这时对于此用户，网页的可用性可以认为是 0。

最错误的概念：Web 标准就是 Div + Css

这里首先要搞懂什么是 Web 标准。Web 标准不是一个标准，而是一个标准的集合。包括了结构化标记语言 XHTML (语义和文档结构)、样式表 CSS (表现) 以及 Javascript+DOM (行为)。

当我们用 Web 标准来进行网页设计的时候，我们首先做什么？我们首先需要的是一个结构良好的 HTML 文件。

那么什么叫结构良好的 HTML 文件？

结构良好的主要含义就是，每一个标记都是自解释的，是有语义的，放置的每一个 HTML 标记都是有确切含义的。比如说：当我写下 `<h1>标题文字</h1>` 的时候，我仅仅告诉浏览器，这里放置的，是一个标题，至于浏览器如何显示它，那就不关我的事了 (尽管几乎所有的浏览器都会以很大的黑体加粗超大来显示)。而当我想告诉浏览器这个标题该如何显示的时候 (字体有多大，白色还是黑色)，那么就到了 CSS 该出手的时候了。可是，有多少人会这样去显示标题呢？`<div style="font-size:30px;font-weight:bold">标题文字</div>`。在蓝色理想，对于这类设计者有一个很轻蔑的称谓，叫做 **Div 狂人**。

NOTE: 说到这里的时候希望不要得罪任何人，我只想就事论事的把问题讲清楚，以便大家日后的改进。

很多的网页设计人员对于此问题理解的不够深入直接导致了页面的文档结构很差。一个很简单的例子，所有的页面都没有 `<h1>`，我在这里可以很郑重地告诉大家：搜索引擎对 `<h1>` 标签中文字的重视程度不亚于页面的 Title 标记。另外，每一个页面应该有且只有一个 `<h1>`，并且 `<h1>` 中间的文字应该是在搜索引擎在正文里第一个所能找到的文字。你们可能想问，有很多的网站，合适放 `<h1>` 的位置现在放置的是一张图片，这样该如何做呢？可以这样写 CSS，`h1 {text-indent:-2000em}`，这个意思相当于说文字跑到屏幕以外，对于用户不可见，对于搜索引擎可见。你们可能会产生疑问，这样会不会受到搜索引擎惩罚？我曾经花了整整一个下午专门查阅资料以进行考证，结论是不会的，这种方式和设置文字颜色和背景色颜色一致来欺骗搜索引擎有着本质的区别，而且，国外专业讲解 Web 标准的站点采用的都是这种做法。如果你们还是觉得这样做有风险，那么我再告诉一种方法：就是在 `<h1>` 中添加一张图片 (很多人实际已经在这么做，只不过是放在 div 里)，在图片的 alt 信息里写明网页的标题。

更进一步地讲，良好的 XHTML 文档结构在脱离 CSS 的情况下，应该是一个结构良好的类似你现在读的这篇 PDF 文档。

在这里提出几点建议：

- 每个页面都应该有且只有一个 `<h1>`，且出现在页面最上方(也可以被包含在一个 `Div` 中)。
- 有了`<h1>`，自然应该有`<h2>`、`<h3>`，用标题去划分文档结构。
- 应该是段落的地方使用`<p>`标记，设置`<p>`的下间距为一行，而不要使用连续的两个`
`。
- 需要缩进段首文字的时候使用 `text-indent:2em`，而不要简单地拍两个中文全角空格，或者干脆四个` `;
- 对于有序列表，使用``，对于无序列表，使用``。对于不是列表，但是长得像列表的页面表现，不要使用 `ol` 或者 `ul`。
- 使用有语义的逻辑元素，而不要使用物理元素。逻辑元素是有语义的，物理元素是无语义的。

NOTE: 这里简单解释一下什么是逻辑元素，什么是物理元素。大家一定发现这样一个事实，``标签和``标签对于文字的效果都是加粗，`<i>`标签和``标签对于文字的效果都是斜体，它们之间有什么区别呢？在这里，`b` 代表的意思是 **Bold**(加粗)，而 `Strong` 代表着**强调**。`i` 代表的意思是 *Italic*(斜体)，而 `em` 代表的意思是 *emphasize*(强调)。看到这里你应该明白了，物理元素告诉浏览器，我希望如何显示这个文字；逻辑元素告诉浏览器，我这里放置的文字有怎么样的重要性。所以，对于搜索引擎来说，``和``之间文字的重要程度要比``和`<i>`大的多。

最普遍的误解：Web 标准就是不使用表格

《PPk on Javascript》一书的作者，Web Standards 组织的成员 Peter-Paul Koch 这样说道：

You should definitely use table to display tabular data !

我所知道的 Web 标准世界级专家，没有人在任何时间，任何地点提到说“Web 标准就是不使用表格”。Web 标准只是说：不应该使用表格去布局，而不是说表格这个标记没有存在的价值。对于表格状的页面，应该毫不犹豫的使用表格去完成它。

我举个例子，如果让我做论坛帖子显示页的表格，我会这么写(做了简化)：

```
<table>
<thead>
<tr><th colspan="6">论坛板块的标题</th></tr>
<tr>
<td>表情</td>
<td>帖子标题</td>
<td>作者</td>
<td>回贴数</td>
<td>浏览数</td>
```

```
<td>最后回复</td>
</tr>
</thead>

<tbody>
<tr><td>与上面想对应的内容区域，这里是循环显示的</td></tr>
</tbody>

<tfoot>
<tr><td>这里是分页信息，比如当前是多少到多少贴，总共多少贴等</td></tr>
</tfoot>
</table>
```

当你要对 版块名称 进行样式的时候：

```
thead tr th{样式}
```

当你要对 每一列的说明(就是表情、作者什么的) 样式的时候：

```
thead tr td{样式}
```

当你要对 循环显示的内容样式的时候：

```
tbody tr td{样式}
```

当你要对 帖子数目 及分页 样式的时候：

```
tfoot tr td{样式}
```

貌似正确的错误：结构与显示分离就是只使用外联式样式表

很多人都简单的认为既然要结构与显示分离，那么就应该绝对的把样式存成 css 文件，而在页面顶部 link 进来。实际上，这种做法是相当错误的，当你这样做了的时候，你不可避免的要使用大量的 ID 和 Class，对日后的维护产生巨大的困难(甚至只有你才愿意去维护这个页面)。

《CSS 权威指南》的作者 Eric A. Meyer 这样说: **In some situations, we should probably use inline style.**

以页面三为例：在<head>部分我使用了内联式样式表:，为什么这么做呢？为了覆盖！因为在“/music/inc/main.css”里，我已经定义了页面左下角显示的背景。但是我又不想每个页面左下角的图案都相同，所以，我把想要在此页显示的背景写在这里。而如果我这里什么也不写会怎么样呢？不用慌张，页面仍然会呈现我定义在/music/inc/main.css 文件中的背景。

再举个使用在行级的样式的例子，大家看“歌曲名 歌曲类型 歌曲长度 人气 下载”的表格标题，除了第一个歌曲名是靠左对齐的，其他都是居中对齐的。这个时候，换做你会怎么做呢？很多人毫不犹豫地给“歌曲名”加个 ID 或者 Class，然后定义在外联式样式表里，你有没有想过，如果你做的是一个非常复杂的页面，你需要定义多少个 ID 和 Class？是 50 个还是 100 个？

所以，以我的看法，合理的样式表应该是 90%外联+5%内联+5%内嵌。

文件该如何组织？是否一个页面对应一个 Css 文件

首先，我想说明，对于文件的组织，每个人都会有不同的做法，我这里只是提出我的做法，给大家参考。

很多站点的根目录都会有这样的目录：[images]、[style]、[js]、[include]、[pic]、[flash] 等，本人所在公司的网站 www.travelchinaguide.com，它就有以上几个目录，可是，把所有的 .css，.js，.swf 加起来，不足 30 个，所以，有必要把这么零散的页面分放不同文件夹么？

我的做法是这样的：

根目录下放置一个 [inc]，将所有的 .css，.js，.ascx 都放进去，而把 .swf 都放到 [images]，取消 [pic] 目录，仅留 [images] 目录，因为这两个目录功能太相似。好了，现在我们仅就 .css 文件来说，根目录的 [inc] 下的 css 文件应该有两个，一个叫 main.css，这个全站使用的样式，主要包括链接样式、字体大小，还有最重要的网页构架的样式（就是组成页面布局的 div 的样式）。另一个叫 index.css，这个主要包括了 index 页所独有的样式。

接下来我们考虑 /music/o2jam.aspx 这个页面的样式文件该如何存放。首先要注意到，o2jam.aspx 与其他同在 music 目录下的其他页面的显示方式是类似的；其次注意到，各个栏目之间的风格又可能是迥异的（比如“访客留言”和“音乐精选”），所以，我在 [music] 目录下也建立 [inc] 目录，同时，我将 o2jam.aspx 用到的样式，同时也是 music 目录其他页所用到的样式，存成 main.css，放在 /music/inc/ 目录下。此时，如果要针对 [music] 目录下的页面应用它所独有的样式时，呵呵，相信你已经知道了，直接写在 head 里，而没有必要再存成 o2jam.css，audition.css 等等，只有在行数特别多时（超过 100 行）再考虑为每个页面创建单独的 css 文件。

按照 Web 标准制作页面时需注意的两点

不要为了标准而去标准

很多人在对 Web 标准过了入门期以后，进入了一种对标准狂热推崇的状态，他们对自己的要求可以用苛刻来形容，用尽一切办法使出浑身解数保证自己制作出来的页面 100% 的符合 Web 标准，并以页面通过 W3C XHTML 和 CSS 验证为己任。

尽管这种对于 Web 标准的态度值得敬佩，但却有些得不偿失。我们有没有思考过为什么要使用 Web 标准？我想可能主要有这么几个原因：1、浏览器兼容性。2、搜索引擎友好型。3、降低

页面文件大小。4、更好的用户体验。当你为 Web 标准付出 50%的努力，以上 4 点的完成度差不多就已经达到了 90%的时候，为什么要再花出剩余 50%的精力去追求余下的 10%呢？

尽可能少用或者不用 Hacks

IE6 有个很有名的 BUG，就是对于类似于：

```
* html div#container{样式}
```

这样的样式 IE6 也可以解析通过，而 FireFox 则对这个样式视而不见。

而 IE6 对 CSS 的支持又不及 FireFox 多，对于类似于

```
html>body div#container{样式}
```

这样的样式，IE6 忽略不计，而 FireFox 却可以正确解析。

很多人了解了这个特性之后仿佛抓住了一根救命稻草，只要一出现浏览器不兼容的情况，甚至不经过思考，就使用这个 Hacks。

实际上，同一种样式的页面表现往往有很多种实现方式，当一种方式行不通的时候，应该尝试换一个思路去实现它，而不是简单的用 Hacks 去解决，这样会给代码的维护带来很大的不便。

最后，请记住《网站重构》的作者说过的一句话：

想让 Web 页面在所有的浏览器中都一摸一样是不可能的，既然这点差异对人们的浏览和使用不会产生影响，那么就让它差异去吧！

不要只知道结构与表现分离，结构与行为也需要分离

我们知道，Web 标准共分为三个部分：1，XHTML 控制文档框架，属于结构；2，CSS 控制页面显示，属于表现；3，JavaScript+DOM 控制用户交互，属于行为。我们一直在强调结构与表现分离，对于结构与行为的分离就考虑的非常少了，这样的结果就是我们的 Web 页面充斥着大量的 JavaScript 代码，既增大了页面体积、占用了带宽，对于搜索引擎也不够友好。

我们现在看下页面一，看下这个页面里的 JavaScript 都做了些什么：

- 表格的就地排序，当你点表格的标题“歌曲类型”，“歌曲长度”，“人气”，会进行自动排序。
- 歌曲名称列的 onmouseover、onmouseout 切换效果。
- 歌曲长度列的颜色效果。
- 对复选框的“全选”、“反选”、“清除”。
- 所有和<textarea />在 onfocus 和 onblur 时的样式，（在“首

- 页 上页 下页 末页”旁边)。
- 输入页数的 input 只接受数字，且长度不能超过 2。

下面是我的几点建议和想法：

1. 我们经常强调“结构与显示”分离，可曾想过“结构与行为”分离？你可曾想过你的站点在访客禁用掉 JavaScript 之后是什么样子？
2. 避免过多的在 HTML 标签内部使用 onclick、onmouseover、onmouseout 等行为，而应该在页面首部写成函数，在 window.onload 事件中加载它，这样你的代码会清晰很多，同时也做到了行为与结构的分离。
3. 在 IE7 还没有大范围普及之前，:first-child 及 a 标记以外的:hover 还是没有实际的使用价值，与其给额外的标签加 class 属性，我宁可在 css 里使用现在不被 IE6 所支持的伪类，再使用 js 来达到相同的效果，等到将来 IE7 大面积普及的时候，只要简单的去掉这些 JS 代码就好了。
4. 当你使用结构与行为分离的时候，带给你的最大好处可能就是代码的维护性和可移植性，当你需要给你的页面添加我这个页面的 textbox 显示效果时，你只需要简单的 prepareForm() 函数拷过去，就 OK 了，对于 HTML 文件，不需要做任何改动。

总结

在本文中，我首先介绍了什么是网页的可访问性和可用性。

随后说明了很多人对于 Web 标准就是 DIV+CSS 的一个普遍误解，以及这样做对搜索引擎的不友好性。

接下来，我讨论了另外一个对 WEB 标准的普遍误解：应用 Web 标准就是不使用表格。

对于样式表的实现方式 和 文件的组织我也做了细致的探讨。

最后，我以一个高级话题，即表现与行为分离，作为结尾。由于这个太前卫了，可以暂时不考虑实现它。

希望这篇文章能给你带来帮助。