**[优化网站设计系列文章总结和导读](http://www.cnblogs.com/chenxizhang/archive/2013/05/20/3088196.html)**

**概述**

其实想写这方面的文章由来已久，这个系列文章的想法是参照[雅虎团队提供的35条性能优化的最佳实践](http://developer.yahoo.com/performance/rules.html)（其实最早的时候是14条），再结合我自己多年的实际工作经验，结合具体的开发平台（ASP.NET），为读者提供既有理论知识、又有实践指导的参考资料，对于优化而言，本身是一个长期细致的工作（没有所谓的银弹），并且重在权衡利弊，选择最适合自己项目情况的解决方案。而要达到这样的目标，就需要对网站设计优化有较为系统的认识，知其然、也知其所以然，然后才可以熟练地运用它们。

我从4月30日开始写这个系列，到今天（5月20日）已经全部写完，主要利用了假日、周末以及晚上的时间），虽然比较辛苦，但自认为这个系列写的质量算是我迄今为止最好的，并且毫无疑问，我有理由借此机会对我的家人的支持表示感谢。

在这个过程中，我也对有关的知识温故而知新，所以说个人收获已经很大了，如果能对阅读者也能有所帮助，那是再好不过的了。

**导读**

这个系列的内容组织保持了雅虎团队当年所提出的35条原则的结构，以便大家对照起来进行学习。值得一说的是，这些文档虽然参照了原文，但不是对原文的翻译，更多的是加入了我的一些见解以及演绎，有些地方可能还与原文有不太一样的说明。

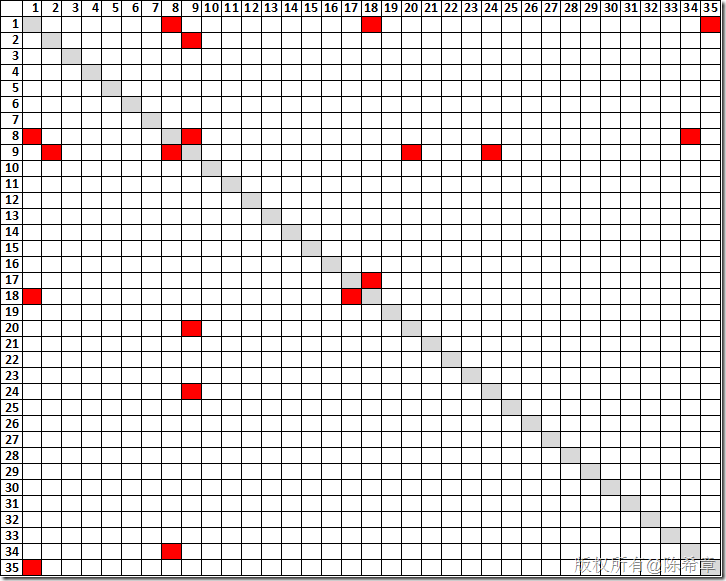
1. [优化网站设计（一）：减少请求数](http://www.cnblogs.com/chenxizhang/archive/2013/04/29/3050839.html)
2. [优化网站设计（二）：使用CDN](http://www.cnblogs.com/chenxizhang/archive/2013/04/30/3051686.html)
3. [优化网站设计（三）：对资源添加缓存控制](http://www.cnblogs.com/chenxizhang/archive/2013/04/30/3052440.html)
4. [优化网站设计（四）：对资源启用压缩](http://www.cnblogs.com/chenxizhang/archive/2013/05/01/3052881.html)
5. [优化网站设计（五）：在顶部放置样式定义](http://www.cnblogs.com/chenxizhang/archive/2013/05/01/3053216.html)
6. [优化网站设计（六）：在文档底部放置脚本定义或引用](http://www.cnblogs.com/chenxizhang/archive/2013/05/01/3053299.html)
7. [优化网站设计（七）：避免在CSS中使用表达式](http://www.cnblogs.com/chenxizhang/archive/2013/05/01/3053439.html)
8. [优化网站设计（八）：将脚本文件和样式表作为外部文件引用](http://www.cnblogs.com/chenxizhang/archive/2013/05/01/3053562.html)
9. [优化网站设计（九）：减少DNS查找的次数](http://www.cnblogs.com/chenxizhang/archive/2013/05/02/3053996.html)
10. [优化网站设计（十）：最小化JAVASCRIPT和CSS](http://www.cnblogs.com/chenxizhang/archive/2013/05/02/3054141.html)
11. [优化网站设计（十一）：避免重定向](http://www.cnblogs.com/chenxizhang/archive/2013/05/05/3060804.html)
12. [优化网站设计（十二）：删除重复脚本](http://www.cnblogs.com/chenxizhang/archive/2013/05/05/3061004.html)
13. [优化网站设计（十三）：配置ETags](http://www.cnblogs.com/chenxizhang/archive/2013/05/11/3072898.html)
14. [优化网站设计（十四）：使AJAX调用尽可能利用缓存特性](http://www.cnblogs.com/chenxizhang/archive/2013/05/12/3073690.html)
15. [优化网站设计（十五）：尽可能早地发送缓冲区内容](http://www.cnblogs.com/chenxizhang/archive/2013/05/12/3073836.html)
16. [优化网站设计（十六）：为AJAX请求使用GET方法](http://www.cnblogs.com/chenxizhang/archive/2013/05/12/3073891.html)
17. [优化网站设计（十七）：延迟或按需加载内容](http://www.cnblogs.com/chenxizhang/archive/2013/05/16/3081941.html)
18. [优化网站设计（十八）：预加载内容](http://www.cnblogs.com/chenxizhang/archive/2013/05/16/3082402.html)
19. [优化网站设计（十九）：减少DOM元素的数量](http://www.cnblogs.com/chenxizhang/archive/2013/05/17/3083162.html)
20. [优化网站设计（二十）：使用多个主机来平衡负载](http://www.cnblogs.com/chenxizhang/archive/2013/05/17/3083509.html)
21. [优化网站设计（二十一）：尽量少用iframe](http://www.cnblogs.com/chenxizhang/archive/2013/05/17/3083788.html)
22. [优化网站设计（二十二）：避免404错误](http://www.cnblogs.com/chenxizhang/archive/2013/05/17/3084052.html)
23. [优化网站设计（二十三）：减小Cookie的体积](http://www.cnblogs.com/chenxizhang/archive/2013/05/18/3085712.html)
24. [优化网站设计（二十四）：通过使用不同的主机减少对cookie的使用](http://www.cnblogs.com/chenxizhang/archive/2013/05/19/3086514.html)
25. [优化网站设计（二十五）：避免对DOM元素进行过多操作](http://www.cnblogs.com/chenxizhang/archive/2013/05/19/3086546.html)
26. [优化网站设计（二十六）：设计“智能”的事件处理程序](http://www.cnblogs.com/chenxizhang/archive/2013/05/19/3086643.html)
27. [优化网站设计（二十七）：使用link而不是@import导入样式表](http://www.cnblogs.com/chenxizhang/archive/2013/05/19/3086725.html)
28. [优化网站设计（二十八）：避免使用Filters(滤镜)](http://www.cnblogs.com/chenxizhang/archive/2013/05/19/3086817.html)
29. [优化网站设计（二十九）：优化图片](http://www.cnblogs.com/chenxizhang/archive/2013/05/19/3086865.html)
30. [优化网站设计（三十）：优化CSS sprites](http://www.cnblogs.com/chenxizhang/archive/2013/05/19/3086895.html)
31. [优化网站设计（三十一）：不要在页面中缩放图片](http://www.cnblogs.com/chenxizhang/archive/2013/05/19/3086912.html)
32. [优化网站设计（三十二）：使favicon.ico文件尽可能小并且可以缓存](http://www.cnblogs.com/chenxizhang/archive/2013/05/20/3087965.html)
33. [优化网站设计（三十三）：尽量使单个内容体积小于25KB](http://www.cnblogs.com/chenxizhang/archive/2013/05/20/3087983.html)
34. [优化网站设计（三十四）：将组件直接打包到页面](http://www.cnblogs.com/chenxizhang/archive/2013/05/20/3087997.html)
35. [优化网站设计（三十五）：避免将img的src属性设置为空白](http://www.cnblogs.com/chenxizhang/archive/2013/05/20/3088007.html)

**权衡**

我多次提到权衡这个词，就好比一个架构师被问到“什么才是最好的架构？”这样的问题的时候，最合理的回答可能真的是“这个得看情况”（虽然这样的回答可能显得有点像外交辞令），因为技术是很多的，架构也很多，没有哪个架构对所有业务场景都是合适的。

同理，我们提到了很多原则，这些原则各自有它的侧重点。很显然，他们可能还互相有冲突。那么，这能说明什么呢？这难道说明这些原则不合理吗？当然不是，正好相反，这才是合理的。

为了帮助大家更好地了解这些原则之间的关系，抑或是互斥的、抑或是互补的，我整理了一个表格如下，给大家参考。

[](http://images.cnitblog.com/blog/9072/201305/20105317-b34b340ddb57410f88028e232968923c.png)

图中红色标识出来的原则就是或多或少会有互斥的。遇到这些原则，你需要兼顾到其他的原则，两害相权取其轻，做出最适合于你的实际情况的选择。

没有标识颜色的，标识这些原则是有互补性的，或者至少不冲突。

**工具**

我在编写本系列文章的时候，用到如下的工具

1. 开发工具：Visual Studio 2012。我用的是2012，但如果想要跟着做练习，有2010也就可以了。
   * <http://www.microsoft.com/visualstudio/chs/downloads>
2. 调试工具：
   * Fiddler: ：<http://fiddler2.com/home> 这是一个独立的网络测试和调试工具。
   * IE 自带的开发工具（通过F12调出）：我用的是IE 10，最好你至少有 IE 9. <http://msdn.microsoft.com/en-us/library/ie/gg589507(v=vs.85).aspx>
   * Chrome自带的开发工具（通过F12调出）：使用最新版本即可。<https://developers.google.com/chrome-developer-tools/>
   * Chrome中安装的Yslow插件：<https://chrome.google.com/webstore/detail/yslow/ninejjcohidippngpapiilnmkgllmakh>
     + 关于Yslow的详细信息（包括计分规则等）请参考：<http://yslow.org/ruleset-matrix/>
     + Google自己也开发了一个类似的插件(功能很像)：Page speed insights <https://chrome.google.com/webstore/detail/pagespeed-insights-by-goo/gplegfbjlmmehdoakndmohflojccocli>

最后，希望这个系列文章对大家有些帮助！