**[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)**

**前言**

网站设计的优化是一个很大的话题,有一些通用的原则,也有针对不同开发平台的一些建议。这方面的研究一直没有停止过，我在不同的场合也分享过这样的话题。

作为通用的原则，雅虎的工程师团队曾经给出过35个最佳实践。这个列表请参考

Best Practices for Speeding Up Your Web Site  <http://developer.yahoo.com/performance/rules.html>

同时，他们还发布了一个相应的测试工具Yslow <http://developer.yahoo.com/yslow/>

我强烈推荐所有的网站开发人员都应该学习这些最佳实践，并结合自己的实际项目情况进行应用。

接下来的一段时间，我将结合ASP.NET这个开发平台，针对这些原则，通过一个系列文章的形式，做些讲解和演绎，以帮助大家更好地理解这些原则，并且更好地使用他们。

**准备工作**

为了跟随我进行后续的学习，你需要准备如下的开发环境和工具

1. Google Chrome 或者firefox ，并且安装 Yslow这个扩展组件.请注意，这个组件是雅虎提供的，但目前没有针对IE的版本。
   1. <https://chrome.google.com/webstore/detail/yslow/ninejjcohidippngpapiilnmkgllmakh>

Technorati Tags: [Performance](http://technorati.com/tags/Performance),[Web design](http://technorati.com/tags/Web+design),[ASP.NET](http://technorati.com/tags/ASP.NET)

* 1. <https://addons.mozilla.org/en-US/firefox/addon/yslow/>
  2. 你应该对这些浏览器的开发人员工具有所了解，你可以通过按下F12键调出这个工具。

1. Visaul Studio 2010 SP1 或更高版本，推荐使用Visual Studio 2012
   1. <http://www.microsoft.com/visualstudio/eng/downloads>
2. 你需要对ASP.NET的开发基本流程和核心技术有相当的了解，本系列文章很难对基础知识做普及。

**本文要讨论的话题**

这一篇我来和大家讨论第六个原则：[Put Scripts at Bottom](http://developer.yahoo.com/performance/rules.html#js_bottom)  （在文档底部放置脚本定义或引用）。

我在上一篇和大家探讨到了[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) ，那是一个给我留下深刻印象的原则，之所以深刻的原因并不是因为这个原则定义有多深刻，而在于以前都这么多，但并没有意识到这是一个不错的做法。

那么，这一篇所谈到的“在文档底部放置脚本定义或引用”这个原则，则是在相当长一段时间内，我都没有注意到的一条原则。换句话说，我以前更多的是将脚本或脚本引用放在HEAD里面，例如下面这样：

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="WebApplication1.Default" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

<script src="Scripts/jquery-2.0.0.min.js"></script>

<script src="Scripts/knockout-2.2.1.js"></script>

<script src="Scripts/modernizr-2.6.2.js"></script>

<script src="default.js"></script>

</head>

<body>

<form id="form1" runat="server">

<div>

</div>

</form>

</body>

</html>

相当长一段时间里，我都没有意识到这可能会成为一个问题。后来也是在研究有关性能有关的问题时，我逐渐注意到了它，原来将脚本定义或引用放在文档底部有助于提高页面加载时的并行度，从而实现提速的目的。

[HTTP /1.1](http://www.w3.org/Protocols/rfc2616/rfc2616-sec8.html#sec8.1.4) 文档中有下面这一段描述：

Clients that use persistent connections SHOULD limit the number of simultaneous connections that they maintain to a given server. A single-user client SHOULD NOT maintain more than 2 connections with any server or proxy. A proxy SHOULD use up to 2\*N connections to another server or proxy, where N is the number of simultaneously active users. These guidelines are intended to improve HTTP response times and avoid congestion

我们知道HTTP本身是无状态的，这意味着每个请求都需要建立独立的连接。但从HTTP /1.1开始对此作了改进，允许在客户端和服务器之间维持连接（Keep-Alive)，这样做的目的是因为在页面加载的时候，往往需要加载很多附加的资源（例如脚本，样式表，图片等），如果每个请求都需要建立连接，显然是不合适的。维持连接（Keep-Alive)的特性，使得连接可以重用，即一段时间内，对同一域的访问请求可以不需要创建新的连接，而是使用之前的连接。

但这里有一个问题，如果维持过多的连接，显然对于客户端和服务器来说都不是好事（这需要耗用资源），所以HTTP /1.1 中建议针对同一个域不要维护超过2个连接。实际上这就是限制了并行度。虽然看起来我们可以通过将这些脚本放在不同的域，来绕开这个并行的问题，但要命的是，只要浏览器开始下载脚本文件，那么它就不可能使用超过2个连接进行同时工作，哪怕这些资源是放在不同的域的。

但据我所知，现代的浏览器对此做了一些突破（可能不止2个连接），但无论如何，这总是有限的。[有一些方法](http://www.impng.com/web-dev/browser-max-parallel-connections.html)可以手工修改浏览器的设置，但我不建议这么做。

如果脚本文件放在HEAD里面，而且如果脚本很多的话，那么下载这些脚本文件就将占用本来就不多的几个连接，那么就必须等这些脚本全部下载完之后，才有可能下载页面其他的部分。这可能是造成页面停滞的一个用户体验。

那么，我们可以怎么使用这个原则呢？很简单，修改页面代码如下

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="WebApplication1.Default" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

</head>

<body>

<form id="form1" runat="server">

<div>

</div>

</form>

<!--将脚本移动到文档的底部有助于提高页面加载速度-->

<script src="Scripts/jquery-2.0.0.min.js"></script>

<script src="Scripts/knockout-2.2.1.js"></script>

<script src="Scripts/modernizr-2.6.2.js"></script>

<script src="default.js"></script>

</body>

</html>

有意思的是，虽然这样的移动并不难，但真的有人不愿意做这样的改变（有一部分理由是因为这样做了之后，文档结构不够清晰了）。作为妥协的一种方案，现在允许在添加脚本的时候，设置一个[DEFER](http://www.websiteoptimization.com/speed/tweak/defer/)属性，标识这个脚本可以延迟加载。

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="WebApplication1.Default" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

<!--使用defer属性延迟加载脚本-->

<script src="Scripts/jquery-2.0.0.min.js" defer="defer"></script>

<script src="Scripts/knockout-2.2.1.js" defer="defer"></script>

<script src="Scripts/modernizr-2.6.2.js" defer="defer"></script>

<script src="default.js" defer="defer"></script>

</head>

<body>

<form id="form1" runat="server">

<div>

</div>

</form>

</body>

</html>

看起来是不错的，但**不幸的是，并非所有浏览器都支持这个特性**。网络开发的复杂性就在于此，我几乎看到你在微笑着点头了。那么，为什么不可以接受将脚本放在文档底部的这一点稳妥的做法呢？

值得一说的是，并不是所有代码都适合从顶部移动到文档底部。例如这些脚本需要在加载过程中动态添加HTML的元素（可能会调用document.write方法），那么就会存在问题。看起来这是不少人固守着原先做法的原理，如果你没有用过类似[JQuery](http://jquery.com/)这一类框架的话，你是情有可原的，反过来说你就不应该这么想了。而如果你真的没有听过和用过JQuery，那么你就离大部队越来越远了。

其实，我们在实践中逐渐发现，绝大部分情况下，我们使用Javascript的时候，都应该等到页面已经准备好之后才开始工作，因为如果你在页面都没有全部呈现出来（并不是代表用户看到，而是呈现）之前就开始操作，你将无法预知会发生什么事情。JQuery创造性地引入了[document.ready](http://docs.jquery.com/Tutorials:Introducing_%24%28document%29.ready%28%29)这样一个事件，而我可以负责任地说，只要你的脚本是在document.ready之后才应该执行的，那么就都可以移动到文档底部。