

网站页面（前端）seo 优化方法及建议

（山东新华电脑学院 整理供稿）

很多时候，网站页面（前端）seo 优化所涉及的点在于页面精简。本教程概述了如何对网页进行精简，以及提供相关建议，加快网站加载速度，提升网站性能。

从相反方面考虑，如果网页庞杂，网页打开速度慢，会有什么后果？

首先，网页代码繁杂，会出现网站加载速度慢的直接后果。有鉴于国内网速慢的大环境，更是加剧了繁杂网页打开慢，加载慢的问题。

其次，移动端用户的崛起，造成了大量用户通过手机，平板等移动端访问网站的状况，网站打开速度慢的问题更加凸显。

再次，网站加载速度的快与慢直接影响到网站给予搜索引擎友好度的多与少。

最后，从网站管理者或者 seo 优化人员来讲，繁杂的网页会造成维护困难的问题。

实际来看，优化网站页面（前端）不是技术难题，给页面减肥也是容易操作的事情，建议是：容易的完成的事情需即可，马上完成。网站页面（前端）优化方法及建议如下：

1：启用 GZIP 压缩网页。

什么是 GZIP 压缩？GZIP 最早由 Jean-loup Gailly 和 Mark Adler 创建，用于 UNIX 系统的文件压缩。我们在 Linux 中经常会用到后缀为 .gz 的文件，它们就是 GZIP 格式的。现今已经成为 Internet 上使用非常普遍的一种数据压缩格式，或者说一种文件格式。

GZIP 压缩网页与什么用？

HTTP 协议上的 GZIP 编码是一种用来改进 WEB 应用程序性能的技术。

大流量的 WEB 站点常常使用 GZIP 压缩技术来让用户感受更快的速度。

这一般是指 WWW 服务器中安装的一个功能，当有人来访问这个服务器中的网站时，服务器中的这个功能就将网页内容压缩后传输到来访的电脑浏览器中显示出来。一般对纯文本内容可压缩到原大小的 40%。这样传输就快了，效果就是你点击网址后会很快的显示出来。当然这也会增加服务器的负载。

2：网页（前端）支持浏览器缓存以达到速度优化效果。

浏览器缓存有什么用？浏览器缓存 (Browser Caching) 是为了节约网络的资源加速浏览，浏览器在用户磁盘上对最近请求过的文档进行存储，当访问者再次请求这个页面时，浏览器就可以从本地磁盘显示文档，这样就可以加速页面的阅览。

浏览器缓存放在什么地方？

可放到文件，内存中（如 session），还有 cache（高速缓存），还有 cookie，session，viewstate，这些是我们经常用到的，但可以认为他们是缓存数据。其实 cache 跟 session 有相似功能，但 cache 可在代码中设置过期时间，依赖项。

所谓依赖项（例如：微软的类 `cachedependency sqlCacheDependency`）当依赖项变动了，系统会通知 cache 过期，无效。缓存可是有服务器缓存，客户端缓存。

举例如何使用浏览器缓存以优化网页（前端）。

使用浏览器缓存以优化网页（前端）

对于 seo 优化人员来讲，不必掌握如何使用浏览器缓存代码，但需要知晓相关原理。

3：利用内容分发网络（CDN）优化网站加载速度。



内容分发网络（CDN）概念：CDN 是构建在网络之上的内容分发网络，依靠部署在各地的边缘服务器，通过中心平台的负载均衡、内容分发、调度等功能模块，

使用户就近获取所需内容，降低网络拥塞，提高用户访问响应速度和命中率。

CDN 的关键技术主要有内容存储和分发技术。

内容分发网络（CDN）原理：CDN 的基本原理是广泛采用各种缓存服务器，将这些缓存服务器分布到用户访问相对集中的地区或网络中，在用户访问网站时，利用全局负载技术将用户的访问指向距离最近的工作正常的缓存服务器上，由缓存服务器直接响应用户请求。

内容分发网络（CDN）目的：使用户可就近取得所需内容，解决 Internet 网络拥挤的状况，提高用户访问网站的响应速度。

内容分发网络（CDN）的主体是空间商而非网站管理员本身。

4：删除多余代码。

很多时候，一个网站的网页中存在很多多余的代码。如首页用到的 js，列表页就用不到，如果全站使用头部调用，就让某些本不需要相关代码的网页也无故添加了许多代码，这些代码的精简，是有利于前端的打开速度的。也就是说：网站中的每一个页面，都无需多余代码。

5：充分利用 css 控制样式。

网站页面（前端）优化方法及建议-css 样式

一个前端代码精简的网站，其 css 代码与 div 是分离的，无需在前端对某些代码使用 style 定义样式，一切以 css 控制即可。如：某一个网页中即使用了 style

定义样式，又使用了 css 定义样式，这样就重复了代码，也会造成网站前端加载速度的慢化。

6：CSS 代码的优化与控制。

一切网站页面（前端）优化方法及建议，其最终目的都是加快网站打开速度，这些方法和建议的落地点在于精简代码，给网页前端进行减肥，css 代码优化的点包括但不限于：缩写 css 代码；排列 css 代码；同属性提取共用 css 选择器；分离网页颜色和背景设置样式（较大站点需要注意）；条理化 css 代码等。

7：前端 js 代码优化。

与 css 代码类似，前端 js 代码优化方法包括但不限于：

避免全局查找：在一个函数中会用到全局对象存储为局部变量来减少全局查找，因为访问局部变量的速度要比访问全局变量的速度更快些；

定时器：如果针对的是不断运行的代码，不应该使用 setTimeout，而应该是用 setInterval，因为 setTimeout 每一次都会初始化一个定时器，而 setInterval 只会在开始的时候初始化一个定时器；

字符串连接：如果要连接多个字符串，应该少使用 +=；如果是收集字符串，比如多次对同一个字符串进行 += 操作的话，最好使用一个缓存，使用 JavaScript 数组来收集，最后使用 join 方法连接起来；

避免 with 语句：和函数类似，with 语句会创建自己的作用域，因此会增加其中执行的代码的作用域链的长度，由于额外的作用域链的查找，在 with 语句中