

## 前端的性能优化都有哪些东西？

# 前端的性能优化都有哪些东西？

前端性能优化是个巨大的课题，如果要面面俱到的说的话，估计三天三夜说不完。所以我们就从实际的工程应用角度出发，聊我们最常遇见的前端优化问题。

Yslow是雅虎开发的基于网页性能分析浏览器插件，可以检测出网页的具体性能值，并且有著名的Yslow 23条优化规则，这23条，就够我们玩的了。

### 1. 减少HTTP请求次数

尽量合并图片、CSS、JS。比如加载一个页面，如果有5个css文件的话，那么会发出5次http请求，这样会让用户第一次访问你的页面的时候会长时间等待。而如果把这5个文件合成一个的话，就只需要发出一次http请求，节省网络请求时间，加快页面的加载。

### 2. 使用CDN

网站上静态资源即css、js全都使用cdn分发，图片亦然。

### 3. 避免空的src和href

当link标签的href属性为空、script标签的src属性为空的时候，浏览器渲染的时候会把当前页面的URL作为它们的属性值，从而把页面的内容加载进来作为它们的值。所以要避免犯这样的疏忽。

### 4. 为文件头指定Expires

Expires是用来设置文件的过期时间的，一般对css、js、图片资源有效。他可以使内容具有缓存性，这样下回再访问同样的资源时就通过浏览器缓存区读取，不需要再发出http请求。如下例子：

新浪微博的这个css文件的Expires时间是2016-5-04 09:14:14.



### 5. 使用gzip压缩内容

gzip能够压缩任何一个文本类型的响应，包括html，xml，json。大大缩小请求返回的数据量。

### 6. 把CSS放到顶部

网页上的资源加载时从上网下顺序加载的，所以css放在页面的顶部能够优先渲染页面，让用户感觉页面加载很快。

### 7. 把JS放到底部

加载js时会对后续的资源造成阻塞，必须得等js加载完才去加载后续的文件，所以就把js放在页面底部最后加载。

### 8. 避免使用CSS表达式

举个css表达式的例子

```
font-color: expression( (new Date()).getHours()%3 ? "#FFFFFF" : "#AAAAAA" );
```

这个表达式会持续的在页面上计算样式，影响页面的性能。并且css表达式只被IE支持。

### 9. 将CSS和JS放到外部文件中

目的是缓存文件，可以参考原则4。但有时候为了减少请求，也会直接写到页面里，需根据PV和IP的比例权衡。

### 10. 权衡DNS查找次数

减少主机名可以节省响应时间。但同时，需要注意，减少主机会减少页面中并行下载的数量。

IE浏览器在同一时刻只能从同一域名下载两个文件。当在一个页面显示多张图片时，IE用户的图片下载速度就会受到影响。所以新浪会搞N个二级域名来放图片。

下面是新浪微博的图片域名，我们可以看到他有多个域名，这样可以保证这些不同域名能够同时去下载图片，而不用排队。不过如果当使用的域名过多时，响应时间就会慢，因为不用响应域名时间不一致。

Request URL: <http://ww4.sinaimg.cn/orj480/736f0c7ejw1f31vf7u4ntj20dc0dcmxt.jpg>

Request URL: <http://ww3.sinaimg.cn/orj480/67dd74e0gw1f329nmuoxvj20hs53e1kx.jpg>

## 11. 精简CSS和JS

这里就涉及到css和js的压缩了。比如下面的新浪的一个css文件，把空格回车全部去掉，减少文件的大小。现在的压缩工具有很多，基本主流的前端构建工具都能进行css和js文件的压缩，如grunt，glup等。

## 12. 避免跳转

有种现象会比较坑爹，看起来没什么差别，其实多次了一次页面跳转。比如当URL本该有斜杠 (/) 却被忽略掉时。例如，当我们要访问 baidu.com 时，实际上返回的是一个包含301代码的跳转，它指向的是 baidu.com/（注意末尾的斜杠）。在nginx服务器可以使用rewrite；Apache服务器中可以使用Alias 或者 mod\_rewrite或者the DirectorySlash来避免。

另一种是不用域名之间的跳转，比如访问 baidu.com/bbs 跳转到bbs.baidu.com/。那么可以通过使用Alias或者 mod\_rewrite建立CNAME（保存一个域名和另外一个域名之间关系的DNS记录）来替代。

## 13. 删除重复的JS和CSS

重复调用脚本，除了增加额外的HTTP请求外，多次运算也会浪费时间。在IE和Firefox中不管脚本是否可缓存，它们都存在重复运算JavaScript的问题。

## 14. 配置ETags

它用来判断浏览器缓存里的元素是否和原来服务器上的一致。比last-modified date更具有弹性，例如某个文件在1秒内修改了10次，Etag可以综合Inode(文件的索引节点(inode)数)，MTime(修改时间)和Size来精准的进行判断，避开UNIX记录MTime只能精确到秒的问题。服务器集群使用，可取后两个参数。使用ETags减少Web应用带宽和负载

## 15. 可缓存的AJAX

异步请求同样的造成用户等待，所以使用ajax请求时，要主动告诉浏览器如果该请求有缓存就去请求缓存内容。如下代码片段，cache:true就是显式的要求如果当前请求有缓存的话，直接使用缓存

```
$.ajax({ url: 'url', dataType: "json", cache: true, success: function(son, status){
```

## 16. 使用GET来完成AJAX请求

当使用XMLHttpRequest时，浏览器中的POST方法是一个“两步走”的过程：首先发送文件头，然后才发送数据。因此使用GET获取数据时更加有意义。

## 17. 减少DOM元素数量

这是一门大学问，这里可以引申出一堆优化的细节。想要具体研究的可以看后面推荐书籍。总之大原则减少DOM数量，就会减少浏览器的解析负担。

## 18. 避免404

比如外链的css、js文件出现问题返回404时，会破坏浏览器的并行加载。

## 19. 减少Cookie的大小

Cookie里面别塞那么多东西，因为每个请求都得带着他跑。

## 20. 使用无cookie的域

比如CSS、js、图片等，客户端请求静态文件的时候，减少了Cookie的反复传输对主域名的影响。

## 21. 不要使用滤镜

IE独有属性AlphaImageLoader用于修正7.0以下版本中显示PNG图片的半透明效果。这个滤镜的问题在于浏览器加载图片时它会终止内容的呈现并且冻结浏览器。在每一个元素（不仅仅是图片）它都会运算一次，增加了内存开支，因此它的问题是多方面的。

完全避免使用AlphaImageLoader的最好方法就是使用PNG8格式来代替，这种格式能在IE中很好地工作。如果你确实需要使用AlphaImageLoader，请使用下划线\_filter又使之对IE7以上版本的用户无效。

## 22. 不要在HTML中缩放图片

比如你需要的图片尺寸是50\* 50

那就不用用一张500\*500的大尺寸图片，影响加载

## 23. 缩小favicon.ico并缓存

以上是Yslow的23个优化原则，基本可以涵盖现在前端大部分的性能优化原则了，很多更加geek和精细优化方法都是从这些原则里面延伸出来的。具体想了解更多优化细则的童鞋建议去看看下面的一本书，毕竟页数多讲的也细嘛：

《[高性能网站建设指南（第二版）](#)》，这里面其实就是细化的讲解了上面的23原则。

前端优化是条漫长的路，不是说一天两天就能全部做完的。我们可以参考上面的准则去把我们目前能做的都给优化了，剩下的更加小的一些细节点不用太过着急，毕竟也是要考虑优化性价比的。比如为了减小一个文件几个字节花上个把月根本不值得。这些优化的东西都可以在我们的工作中慢慢去通过积累，去通过google解决。