# 图片方面

### 小图拼大图

有些网页上有多个小图片、图标按钮的网页，有多少图片，浏览器就需要请求多少将这些小文件，请求这些小图片文件将明显影响网页的加载速度。所以我们应该尽可能将小图片拼合一个PNG大图片上，然后通过坐标来显示图标。一次请求一个大图片比多次请求小图片速度要快不少。

这也是减少HTTP请求；

### 图片使用height和width属性

这两个属性可以让浏览器在加载图片之前就知道图片的长和宽，并预留出指定的长宽待图片加载后显示。如果没有这两个属性，浏览器还需要在读取图片成功后再处理一次页面布局样式，这无疑减慢了网页加载速度。所以在固定图片大小的情况下最好都使用上长和宽属性。

### 图片格式

PEG一般是用来存储照片或全彩色图片的，比如照片、屏幕截图等。GIF图片格式的颜色要比JPEG少，适合做小图，如制作按钮、Logo等，另外GIF支持动态效果。PNG跟GIF比较相似，但它的尺寸较大，支持的颜色也比GIF要多，并且PNG支持背景透明。我们可以试试使用一种不同的格式保存图片试下，如将PNG和JPEG换成GIF试试。

### 图片尺寸

尽量减小图片尺寸

# CSS/JS方面

### 存放位置

将样式表放在顶部；

脚本放在底部；

### 移除重复脚本

页面中脚本越多，就越有可能将一个脚本包含两次，引起不必要的HTTP请求和执行JS所浪费的时间；

### 避免CSS表达式

### 减少HTTP请求

现在的网页都有多个图片、CSS外部文件链接、Javascript外部脚本链接。所以当访问一个网页时浏览器需要多次向服务器请求这些文件。在请求和加载之间会产生不少的时间差。特别是一些网页上有多个小图片、图标按钮的网页，有多少图片，浏览器就需要请求多少将这些小文件，多将请求这些小图片文件将明显影响网页的加载速度。所以我们应该尽可能将小图片拼合一个PNG大图片上，然后通过坐标来显示图标。一次请求一个大图片比多次请求小图片速度要快不少。同样，最好将CSS和Javascript尽可能地整合到一个文件中都有助于加快网页载入速度。

### 精简JS和CSS文件

从代码中移除不必要的字符以减少其大小，减少加载时间；

### 减少JS对DOM的读取

用脚本读取Dom会使页面变慢，要注意以下几点

* 读取了元素之后缓存起来
* 将节点更新好之后再加入dom树
* 避免用脚本做页面的layout修改

### 目录地址后加上斜杠（/）

如访客点击访问这样一个目录地址，去打开这个目录下的index.html文档。当服务器收到请求后它需要消耗一些时间来分析这是一个文件还是一个目录。但是如果我们在最后加上一个斜杠（/），服务器就知道你是在访问一个目录地址，然后就直接加载默认文档index.html或index.php就行了。这样服务器就不用花时间来分析这个地址，也起到了一定加速的作用。

### 一个link标签摘引多个外部样式

以下是从网上找的资料，未经验证，但是很多网站都是这样弄的

一个link标签引用多个外部样式：

<linkhref="http://t5.zbjimg.com/min/?b=v&amp;f=g.css,base.css,module/tips.css,tips-c.css,module/window.css,head.css,footer.css,main.css,pub.css&amp;v=1.002" type="text/css" rel="stylesheet">

**【说明】**

应该是由js解析这个link标签传递的参数得到最终导入的样式，link不具有一次导入多个样式的功能，这个link的href里不是原来的意思了。它应该是请求的后台的一个文件（应该是mvc格式的），在这个请求里有参数跟版本号，很明显，这个请求的功能就是返回一整段css代码（它包含参数里面的所有css文件的代码），其实 js文件也可以这样做（通过一个请求返回所有必要的css或者js文件），这样的目的当然是减少请求个数，缩短反应时间

将几个css文件合并成了一个css

window.css,head.css,footer.css,main.css,pub.css

从命名上就可以看出来

**【好处】**

这样可以减少请求数，提升访问速度

# css sprite

## CSS Sprites简介

[CSS](http://baike.baidu.com/view/15916.htm" \t "_blank)Sprites在国内很多人叫[css](http://baike.baidu.com/view/15916.htm" \t "_blank)[精灵](http://baike.baidu.com/view/49189.htm" \t "_blank)，是一种网页图片应用处理方式。它允许你将一个页面涉及到的所有零星图片都包含到一张大图中去，这样一来，当访问该页面时，载入的图片就不会像以前那样一幅一幅地慢慢显示出来了。对于当前网络流行的速度而言，不高于200KB的单张图片的所需载入时间基本是差不多的，所以无需顾忌这个问题。

加速的关键，不是降低质量，而是减少个数。传统切图讲究精细，图片规格越小越好，重量越小越好，其实规格大小无所谓，计算机统一都按[byte](http://baike.baidu.com/view/44243.htm" \t "_blank)计算。[客户端](http://baike.baidu.com/view/930.htm" \t "_blank)每显示一张图片都会向服务器发送请求。所以，图片越多请求次数越多，造成延迟的可能性也就越大。

## CSS Sprites原理

CSS Sprites其实就是把网页中一些背景图片整合到一张图片文件中，再利用CSS的“background-image”，“background- repeat”，“background-position”的组合进行背景定位，background-position可以用数字精确的定位出[背景图片](http://baike.baidu.com/view/950110.htm" \t "_blank)的位置。

## CSS Sprites优缺点

利用CSS Sprites能很好地减少网页的http请求，从而大大的提高页面的性能，这也是CSS Sprites最大的优点，也是其被广泛传播和应用的主要原因；

CSS Sprites能减少图片的字节，曾经比较过多次3张图片合并成1张图片的字节总是小于这3张图片的字节总和。

解决了[网页设计](http://baike.baidu.com/view/8972.htm" \t "_blank)师在图片命名上的困扰，只需对一张集合的图片上命名就可以了，不需要对每一个小元素进行命名，从而提高了网页的制作效率。

更换风格方便，只需要在一张或少张图片上修改图片的颜色或样式，整个网页的风格就可以改变。维护起来更加方便。

诚然CSS Sprites是如此的强大，但是也存在一些不可忽视的缺点，如下：

在图片合并的时候，你要把多张图片有序的合理的合并成一张图片，还要留好足够的空间，防止板块内出现不必要的背景；这些还好，最痛苦的是在[宽屏](http://baike.baidu.com/view/300568.htm" \t "_blank)，高分辨率的屏幕下的自适应页面，你的图片如果不够宽，很容易出现背景断裂；

CSS Sprites在开发的时候比较麻烦，你要通过[photoshop](http://baike.baidu.com/view/2929.htm" \t "_blank)或其他工具测量计算每一个背景单元的精确位置，这是针线活，没什么难度，但是很繁琐；幸好[腾讯](http://baike.baidu.com/view/1591.htm" \t "_blank)的鬼哥用ADOBE AIR开发了一个CSS Sprites 样式生成工具，虽然还有一些使用上的不灵活，但是已经比photoshop测量来的方便多了，而且样式直接生成，复制，拷贝就OK！

CSS Sprites在维护的时候比较麻烦，如果[页面背景](http://baike.baidu.com/view/7105320.htm" \t "_blank)有少许改动，一般就要改这张合并的图片，无需改的地方最好不要动，这样避免改动更多的[css](http://baike.baidu.com/view/15916.htm" \t "_blank)，如果在原来的地方放不下，又只能（最好）往下加图片，这样图片的字节就增加了，还要改动css。

CSS Sprites非常值得学习和应用，特别是页面有一堆ico（图标）。总之很多时候大家要权衡一下利弊，再决定是不是应用CSS Sprites。

## CSS Sprites的使用

有几篇关于CSS Sprites的文章，基本上把其原理和机制说明得很清楚。

What Are CSS Sprites?

How to create CSS sprites

Creating Rollover Effects with CSS Sprites

Building a Dynamic Banner with CSS Sprites

High Performance Web Sites中关于CSS Sprites的内容3.2. CSS Sprites

## CSS Sprite的例子

.bg\_sprite{background-image:url(/整图地址); background-repeat:no-repeat}

引用该类 .. 然后在元素中逐一定义背景坐标 .. 以下为关键属性 ..

#ico1 {width:容器宽度;height:容器高度;background:-position:X坐标 Y坐标}

#ico2 {width:容器宽度;height:容器高度;background:-position:X坐标 Y坐标}

#ico3 {width:容器宽度;height:容器高度;background:-position:X坐标 Y坐标}

.nav {width:容器宽度;height:容器高度;background:-position:X坐标 Y坐标}

必须要限定容器大小符合背景图元素位置 .. 另外 .. XY轴是相对于整张图片的左上角来算的 .. 所以取值一定要算清楚。

## CSS Sprites的问题

由于[IE](http://baike.baidu.com/view/703.htm" \t "_blank)6存在的background的flicker问题IE6/Win, background image on , cache=‘check every visit’: flicker!，有人针对此问题提出了解决方案Fast Rollovers Without Preload

关于[IE6](http://baike.baidu.com/view/476980.htm" \t "_blank)的flicker问题，fivesevensix上有一篇很不错的研究文章Minimize Flickering CSS Background Images in IE6

另外：brunildo的CSS tests and experiments是关于[css](http://baike.baidu.com/view/15916.htm" \t "_blank)各种功能不错的参考手册和测试工具。

# DEMO HTML

## 关键字加链接

关键字标题要加链接，此处指的是一般搜索会用到的关键字

## 更多链接

更多 链接 要加上属性 rel=”nofollow”，用于避开搜索的；

空链接要写成href=”#”，不能写成href=””

## 图片alt属性

一定要加此属性，此属性中必须写内容，写哪些内容问优化

## Head中的标签

<title> <keyword> <description>

这三个标签中必须写内容，写哪些内容问优化

每个页面<title>标签中的标题应不同；

## CSS3

尽量用CSS3，在高级浏览器中作一些特效；

## 广告图片

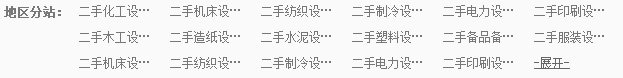
广告图片也放JS中生成

## 顶部topbar

top内容放JS中生成，加快读取速度（默认不读取JS，HTML会全部读取，即使是通过JS引用的HTML文件）

## 百度快照相关

默认隐藏部分不要在CSS中设置隐藏，要在JS中设置，这样在百度快照中，隐藏部分会全部显示出来，因为快照会忽略JS



|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
|  |
|  |
|  |
| 图片延迟加载 |
| 留广告位或去掉广告位也不影响布局 |