1、尽可能减少HTTP请求：图片合并 （css sprites），Js脚本文件合并、css文件合并

2、减少DNS查询

DNS查询服务指域名查找，指将请求的域名转化为对应的IP地址，就如姓名和门牌号的关系。如输入www.baidu.com DNS系统会将此域名转化为119.75.217.109，然后将IP地址返回给浏览器，这个过程会花费一定的时间，影响页面的加载，所以要尽可能减少DNS查询。

减少DNS查询的方法有：　缓存 DNS查找可以改善页面性能，大多数浏览器有自己的缓存系统，缓存时间又不一致，缓存时间越长，DNS保存的时间越长。

当客户端中的 DNS缓存都为空时(浏览器和操作系统都为空)， DNS查找的次数和页面中主机名的数量相同。这其中包括页面中 URL、图片、脚本文件、样式表、Flash对象等包含的主机名。减少主机名的数量可以减少DNS查找次数。

减少主机名的数量还可以减少页面中并行下载的数量。减少 DNS查找次数可以节省响应时间，但是减少并行下载却会增加响应时间。我的指导原则是 把这些页面中的内容分割成至少两部分但不超过四部分。这种结果就是在减少 DNS查找次数和保持较高程度并行下载两者之间的权衡了。

3、将css放在页面最上面，将js放在页面最下面

css放在页面最上面可以防止页面出现白屏、闪跳的现象，即减少页面的首屏出现时间。js的下载和执行会阻塞Dom树的构建（严谨地说是中断了Dom树的更新），所以script标签放在首屏范围内的HTML代码段里会截断首屏的内容。而且js中可能会对DOM节点进行操作，而这时代码是自上向下进行执行的，这样会造成js对相应的元素操作不了。所以js放在页面的最下面

4、压缩js和css

减少文件体积，去除不必要的空白符、格式符、注释（即对代码进行格式化）

5、把js和css提取出来放在外部文件中

这一条要灵活运用，把js和css提取出来放在外部文件的优点是：减少html体积，提高了js和css的复用性，提高日后的可维护性

缺点：增加了http请求，不过这一点可以通过缓存来解决

什么情况下将js和css写在页面内呢，可以分为几种情况：js和css代码比较少；这个页面不怎么会访问

6、避免重定向

重定向就是用户请求的页面被转移到了别的地方，浏览器向服务请请求一个页面，服务器告诉浏览器请求的页面已经被转移到另外一个页面，并告知另一个页面地址，浏览器就再发送请求到重定向的地址。这样会增加服务器和浏览器之间的往返次数，影响网站性能。

重定向状态码有：301永久重定向 302临时重定向。304 not modified 并不是真的重定向，它是用来告诉浏览器get请求的文件在缓存中，避免重新下载。

7、移除重复脚本

8、配置实体标签Etag

9、使用ajax缓存

ajax的get和post方法：

只要是浏览器的get请求，浏览器都会使用缓存，对于同一地址的请求，服务器会发送304状态码到浏览器，浏览器就会使用缓存中的数据

post的请求每次都会被执行，浏览器不会缓存

10、使用Gzip压缩

11、使用CDN(内容分发网络)

12、在css中避免expression表达式(calc())

13、添加expires/cache-control头

在写代码的时候应该注意：

1.标签闭合

2.标签小写

3.不能随意嵌套

提高被搜索引擎搜到几率：

mate中的name变量【其中keywords和description尤其重要】

Meta name=”Kyewords”Lang=”EN”Content=”vacation,greece,sunshine”

Meta name=”Description” Content=”你网页的简述”

Meta name=”Robots” Content=”All|None|Index|Noindex|Follow|Nofollow”

[ all：文件将被检索，且页面上的链接可以被查询；

none：文件将不被检索，且页面上的链接不可以被查询；(和 “noindex, no follow” 起相同作用)

index：文件将被检索；（让robot/spider登录）

follow：页面上的链接可以被查询；

noindex：文件将不被检索，但页面上的链接可以被查询；(不让robot/spider登录)

nofollow：文件将不被检索，页面上的链接可以被查询。(不让robot/spider顺着此页的连接往下探找)]

Meta name=”Author” Content=”张三，abc@sina.com”

Meta name=”Copyright” Content=”本页版权归Zerospace所有。All Rights Reserved”

Meta name=”Generator” Content=”PCDATA|FrontPage|”

META name=”revisit-after” CONTENT=”7 days”

提高网页加载速度

1.把JavaScript和css写到外部文档

2.去掉不必要的空格和注释

3.尽量不使用table布局，尽量使用不嵌套的table布局

4.指定图像和table的大小

5.尽量不要使用图像来代替文本（比如在导航栏）

6.延迟脚本加载，将脚本放在页面底部

7.尽量使用png格式的图像

8.检查丢失的文件和图像

优化的方向有两个：

减少页面体积，提升网络加载

优化页面渲染

减少页面体积，提升网络加载

静态资源的压缩合并（JS 代码压缩合并、CSS 代码压缩合并、雪碧图）

静态资源缓存（资源名称加 MD5 戳）

使用 CDN 让资源加载更快

优化页面渲染

CSS 放前面，JS 放后面

懒加载（图片懒加载、下拉加载更多）

减少DOM 查询，对 DOM 查询做缓存

减少DOM 操作，多个操作尽量合并在一起执行（DocumentFragment）

事件节流

尽早执行操作（DOMContentLoaded）

使用 SSR 后端渲染，数据直接输出到 HTML 中，减少浏览器使用 JS - 模板渲染页面 HTML 的时间