

浏览器加载网页的顺序

浏览器加载和渲染html的顺序

1. 用户输入网址（假设是个 HTML 页面，并且是第一次访问），浏览器向服务器发出请求，服务器返回 HTML 文件；
2. 浏览器开始载入 HTML 代码，发现 `<head>` 标签内有一个 `<link>` 标签引用外部 CSS 文件；
3. 浏览器又发出 CSS 文件的请求，服务器返回这个 CSS 文件；
4. 浏览器继续载入 HTML 中 `<body>` 部分的代码，并且 CSS 文件已经拿到手了，可以开始渲染页面了；
5. 浏览器在代码中发现一个 `` 标签引用了一张图片，向服务器发出请求。此时浏览器不会等到图片下载完，而是继续渲染后面的代码；
6. 服务器返回图片文件，由于图片占用了一定面积，影响了后面段落的排布，因此浏览器需要回过头来重新渲染这部分代码；
7. 浏览器发现了一个包含一行 JavaScript 代码的 `<script>` 标签，赶快运行它；
8. JavaScript 脚本执行了这条语句，它命令浏览器隐藏掉代码中的某个 `<div>`（`style.display="none"`）。杯具啊，突然就少了这么一个元素，浏览器不得不重新渲染这部分代码；
9. 终于等到了 `</html>` 的到来，浏览器泪流满面.....
10. 等等，还没完，用户点了一下界面中的“换肤”按钮，JavaScript 让浏览器换了一下 `<link>` 标签的 CSS 路径；
11. 浏览器召集了在座的各位 `<div>` 们，“大伙儿收拾收拾行李，咱得重新来过.....”，浏览器向服务器请求了新的CSS文件，重新渲染页面。浏览器每天就这么来来回回跑着，要知道不同的人写出来的 HTML 和 CSS 代码质量参差不齐，说不定哪天跑着跑着就挂掉了。好在这个世界还有这么一群人——页面重构工程师，平时挺不起眼，也就帮视觉设计师们切切图啊改改字，其实背地里还是干了不少实事的。

各个步骤的加载渲染时间，可以通过各个浏览器的插件跟踪到，例如IE的httpwatch、火狐的firebug等等。

二、如何加快HTML页面加载速度

1. 页面减肥

- a. 页面的肥瘦是影响加载速度最重要的因素。
- b. 删除不必要的空格、注释。
- c. 将inline的script和css移到外部文件。
- d. 可以使用HTML Tidy来给HTML减肥，还可以使用一些压缩工具来给JavaScript减肥。

2. 减少文件数量

- a. 减少页面上引用的文件数量可以减少HTTP连接数。
- b. 许多JavaScript、CSS文件可以合并最好合并，人家财帮子都把自己的JavaScript. functions和Prototype.js合并到一个base.js文件里去了。

3. 减少域名查询

- a. DNS查询和解析域名也是消耗时间的，所以要减少对外部JavaScript、CSS、图片等资源的引用，不同域名的使用越少越好。

4. 缓存重用数据

- a. 对重复使用的数据进行缓存。

5. 优化页面元素加载顺序

- a. 首先加载页面最初显示的内容和与之相关的JavaScript和CSS，然后加载HTML相关的东西，像什么不是最初显示相关的图片、flash、视频等很肥的资源就最后加载。

6. 减少inline JavaScript的数量

- a. 浏览器parser会假设inline JavaScript会改变页面结构，所以使用inline JavaScript开销较大。
- b. 不要使用document.write()这种输出内容的方法，使用现代W3C DOM方法来为现代浏览器处理页面内容。

7. 使用现代CSS和合法的标签

- a. 使用现代CSS来减少标签和图像，例如使用现代CSS+文字完全可以替代一些只有文字的图片。
- b. 使用合法的标签避免浏览器解析HTML时做“error correction”等操作，还可以被HTML Tidy来给HTML减肥。

8. Chunk your content

- a. 不要使用嵌套table，而使用非嵌套table或者div。将基于大块嵌套的table的layout分解成多个小table，这样就不需要等到整个页面（或大table）内容全部加载完才显示。

9. 指定图像和table的大小

- a. 如果浏览器可以立即决定图像或table的大小，那么它就可以马上显示页面而不要重新做一些布局安排的工作。
- b. 这不仅加快了页面的显示，也预防了页面完成加载后布局的一些不当的改变。
- c. image使用height和width。