

浏览器加载和渲染html的顺序

js的加载

1. 不能并行下载和解析（阻塞下载）
2. 当引用了JS的时候，浏览器发送1个js request就会一直等待该request的返回。因为浏览器需要1个稳定的DOM树结构，而JS中很有可能代码直接改变了DOM树结构，比如使用 document.write 或 appendChild,甚至是直接使用的location.href进行跳转，浏览器为了防止出现JS修改DOM树，需要重新构建DOM树的情况，所以 就会阻塞其他的下载和呈现。

HTML页面加载和解析流程

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，开始运行它；
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。