

加餐一 | 浏览上下文组：如何计算Chrome中渲染进程的个数？

2019-11-15 李兵

《浏览器工作原理与实践》

课程介绍 >



讲述：李兵

时长 11:59 大小 8.24M



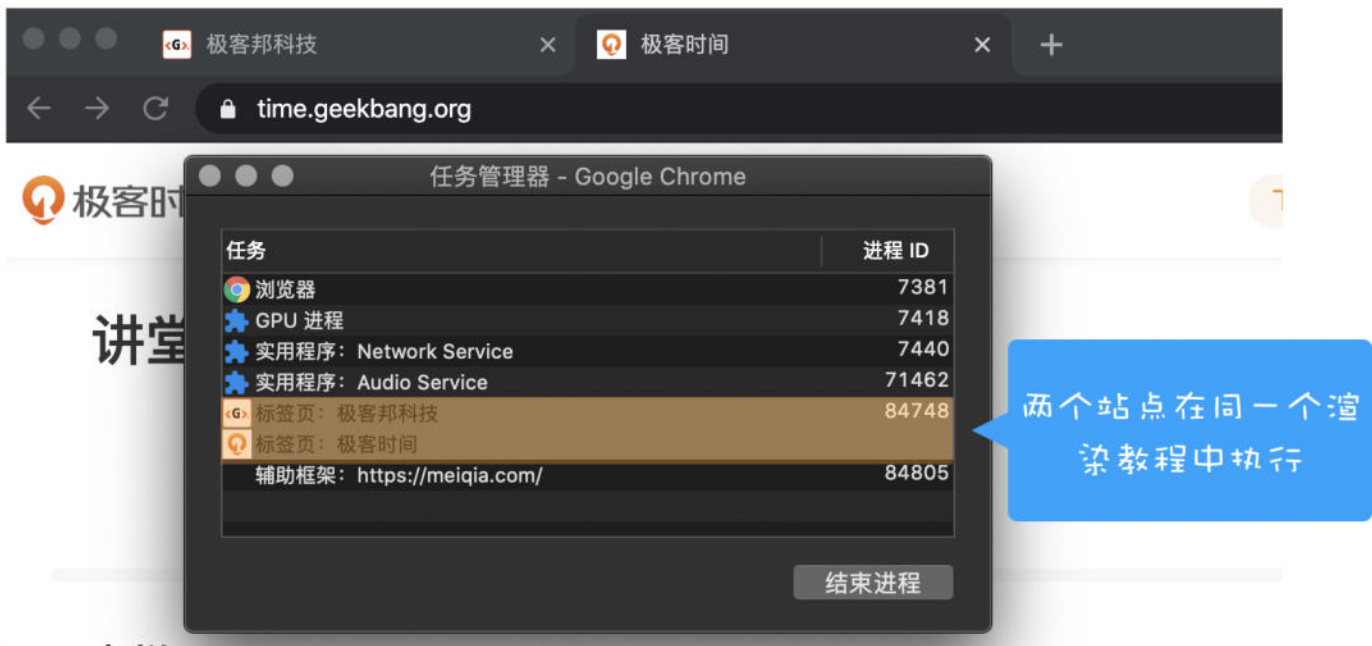
你好，我是李兵。

在留言区，经常有朋友问到如何计算 Chrome 中渲染进程个数的问题，那么今天我就来完整地解答这个问题。

在前面“[🔗 04 | 导航流程](#)”这一讲中我们介绍过了，在默认情况下，如果打开一个标签页，那么浏览器会默认为其创建一个渲染进程。不过我们在“[🔗 04 | 导航流程](#)”中还介绍了同一站点的概念，如果从一个标签页中打开了另一个新标签页，当新标签页和当前标签页属于同一站点的话，那么新标签页会复用当前标签页的渲染进程。

具体地讲，如果我从极客邦 (www.geekbang.org) 的标签页中打开新的极客时间 (time.geekbang.org) 标签页，由于这两个标签页属于同一站点 (相同协议、相同根域名)，所以他们会共用同一个渲染进程。你可以看下面这张 Chrome 的任务管理器截图：





多个标签页运行在同一个渲染进程

观察上图，我们可以看到，极客邦官网和极客时间标签页都共用同一个渲染进程，该进程 ID 是 84748。

不过如果我们分别打开这两个标签页，比如先打开极客邦的标签页，然后再新建一个标签页，再在这个新标签页中打开极客时间，这时候我们可以看到这两个标签页分别使用了两个不同的渲染进程。你可以参看下图：





多个标签页运行在不同的渲染进程中

那么到了这里，你一定会很好奇，既然都是同一站点，为什么从 A 标签页中打开 B 标签页，就会使用同一个渲染进程，而分别打开这两个标签页，又会分别使用不同的渲染进程？

标签页之间的连接

要搞清楚这个问题，我们要先来分析下浏览器标签页之间的连接关系。

我们知道，浏览器标签页之间是可以通过 JavaScript 脚本来连接的，通常情况下有如下几种连接方式：

第一种是通过[<a>标签来和新标签建立连接](#)，这种方式我们最熟悉，比如下面这行代码是从极客邦标签页里面拷贝过来的：

```
1 <a href="https://time.geekbang.org/" target="_blank" class="">极客时间</a>
```


复制代码



这是从极客邦官网中打开极客时间的链接，点击该链接会打开新的极客时间标签页，新标签页中的 `window.opener` 的值就是指向极客邦标签页中的 `window`，这样就可以在新的极客时间标签页中通过 `opener` 来操作上个极客邦的标签页了。这样我们可以说，这两个标签页是有连接的。

另外，还可以通过 JavaScript 中的 `window.open` 方法来和新标签页建立连接，演示代码如下所示：

```
1 new_window = window.open("http://time.geekbang.org")
```

 复制代码

通过上面这种方式，可以在当前标签页中通过 `new_window` 来控制新标签页，还可以在新标签页中通过 `window.opener` 来控制当前标签页。所以我们也可以说，如果从 A 标签页中通过 `window.open` 的方式打开 B 标签页，那么 A 和 B 标签页也是有连接的。

其实通过上述两种方式打开的新标签页，不论这两个标签页是否属于同一站点，他们之间都能通过 `opener` 来建立连接，所以他们之间是有联系的。在 WhatWG 规范中，把这一类具有相互连接关系的标签页称为**浏览上下文组 (browsing context group)**。

既然提到浏览上下文组，就有必要提下浏览上下文，通常情况下，我们把一个标签页所包含的内容，诸如 `window` 对象，历史记录，滚动条位置等信息称为浏览上下文。这些通过脚本相互连接起来的浏览上下文就是浏览上下文组。如果你有兴趣，可以参开下 [🔗 规范文档](#)。

也就是说，如果在极客邦的标签页中，通过链接打开了多个新的标签页，不管这几个新的标签页是否是同一站点，他们都和极客邦的标签页构成了浏览上下文组，因为这些标签页中的 `opener` 都指向了极客邦标签页。

Chrome 浏览器会将浏览上下文组中属于同一站点的标签分配到同一个渲染进程中，这是因为如果一组标签页，既在同一个浏览上下文组中，又属于同一站点，那么它们可能需要在对方的标签页中执行脚本。因此，它们必须运行在同一渲染进程中。

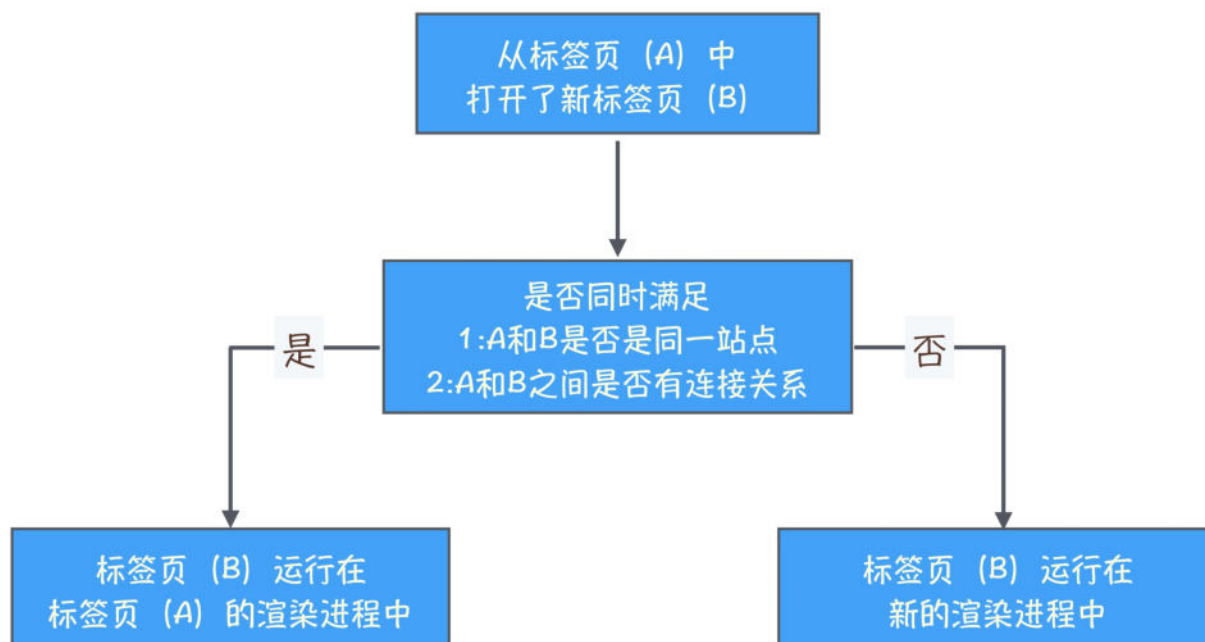
现在我们清楚了浏览器是怎么分配渲染进程的了，接下来我们就可以来分析文章开头提的那个问题了：



既然都是同一站点，为什么从 A 标签页中打开 B 标签页，就会使用同一个渲染进程？而分别打开这两个标签页，又会分别使用不同的渲染进程？

首先来看第一种，在极客邦标签页内部通过链接打开极客时间标签页，那么极客时间标签页和极客邦标签页属于同一个浏览上下文组，且它们属于同一站点，所以浏览器会将它们分配到同一个渲染进程之中。

而第二种情况就简单多了，因为第二个标签页中并没有第一个标签页中的任何信息，第一个标签页也不包含任何第二个标签页中的信息，所以他们不属于同一个浏览上下文组，因此即便他们属于同一站点，也不会运行在同一个渲染进程之中。下面是我画的计算标签页的流程图，你可以参考下：



计算标签页使用的渲染进程数目

一个“例外”

好了，现在我们清楚了 Chrome 浏览器为标签页分配渲染进程的策略了：

1. 如果两个标签页都位于同一个浏览上下文组，且属于同一站点，那么这两个标签页会被浏览器分配到同一个渲染进程中。
2. 如果这两个条件不能同时满足，那么这两个标签页会分别使用不同的渲染进程来渲染。



现在你可以想一下，如果从 A 标签页中打开 B 标签页，那我们能肯定 A 标签页和 B 标签页属于同一浏览上下文组吗？

答案是“不能”，下面我们来看个例子，在“[04 | 导航流程](#)”的留言区中，ID 为“芳华年月”的朋友就提出了这样的问题：

请问老师，<https://linkmarket.aliyun.com> 内新开的标签页都是新开一个渲染进程，能帮忙解释下吗？

我们先来复现下“芳华年月”所描述的现象，首先打开 linkmarket.aliyun.com 这个标签页，再在这个标签页中随便点击两个链接，然后就打开了两个新的标签页了，如下图所示：



“例外”情况

我通过 A 标签页中的链接打开了两个新标签页，B 和 C，而且我们也可以看出来，A、B、C 三个标签页都属于同一站点，正常情况下，它们应该共用同一个渲染进程，不过通过上图我们可以看出来，A、B、C 三个标签页分别使用了三个不同的渲染进程。

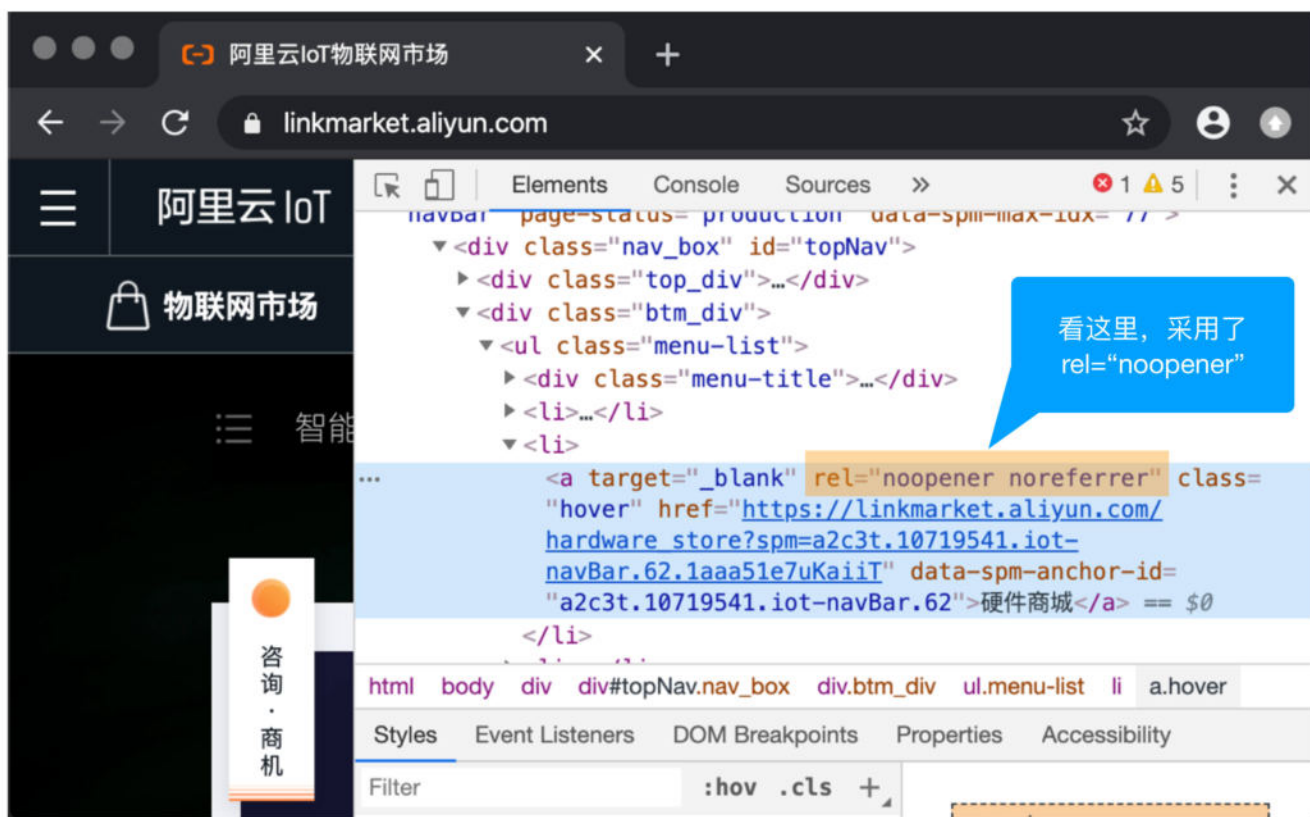
既然属于同一站点，又不在同一个渲染进程中，所以可以推断这三个标签页不属于同一个浏览上下文组，那么我们接下来的分析思路就很清晰了：

1. 首先验证这三个标签页是不是真的不在同一个浏览上下文组中；
2. 然后再分析它们为什么不在同一浏览上下文组。

为了验证猜测，我们可以通过控制台，来看看 B 标签页和 C 标签页的 opener 的值，结果发现这两个标签页中的 opener 的值都是 null，这就确定了 B、C 标签页和 A 标签页没有连接关系，当然也就不属于同一浏览上下文组了。

验证了猜测，接下来的我们就是来查查，阿里的这个站点是不是采用了什么特别的手段，移除了这两个标签页之间的连接关系。

我们可以看看实现链接的 HTML 文件，如下图所示：



通过上图，我们可以发现，a 链接的 rel 属性值都使用了 noopener 和 noreferrer，通过 noopener，我们能猜测得到这两个值是让被链接的标签页和当前标签页不要产生连接关系。

通常，将 noopener 的值引入 rel 属性中，就是告诉浏览器通过这个链接打开的标签页中的 opener 值设置为 null，引入 noreferrer 是告诉浏览器，新打开的标签页不要有引用关系。

好了，到了这里我们就知道了，通过 linkmarket.aliyun.com 标签页打开新的标签页要使用单独的一个进程，是因为使用了 rel= noopener 的属性，所以新打开的标签页和现在的标签页就没有了引用关系，当然它们也就不属于同一浏览上下文组了。这也同时解答了“芳华年月”所提出的问题。

站点隔离

上面我们都是基于标签页来分析渲染进程的，不过我在“[🔗 35 | 安全沙箱](#)”中介绍过了，目前 Chrome 浏览器已经默认实现了站点隔离的功能，这意味着标签页中的 iframe 也会遵守同一站点的分配原则，如果标签页中的 iframe 和标签页是同一站点，并且有连接关系，那么标签页依然会和当前标签页运行在同一个渲染进程中，如果 iframe 和标签页不属于同一站点，那么 iframe 会运行在单独的渲染进程中。

我们先来看下面这个具体的例子吧：

📄 复制代码

```
1 <head>
2   <title>站点隔离:demo</title>
3   <style>
4     iframe {
5       width: 800px;
6       height: 300px;
7     }
8   </style>
9 </head>
10 <body>
11   <div><iframe src="iframe.html"></iframe></div>
12   <div><iframe src="https://www.infoq.cn/"></iframe></div>
13   <div><iframe src="https://time.geekbang.org/"></iframe></div>
14   <div><iframe src="https://www.geekbang.org/"></iframe></div>
15 </body>
16 </html>
```



在 Chrome 浏览器中打开上面这个标签页，然后观察 Chrome 的任务管理，我们会发现这个标签页使用了四个渲染进程，如下图所示：

任务管理器 - Google Chrome

任务	进程 ID
浏览器	7381
GPU 进程	7418
实用程序: Network Service	7440
实用程序: Audio Service	71462
标签页: 站点隔离 demo	125
辅助框架: https://infoq.cn/	127
辅助框架: https://geekbang.org/	132
辅助框架: https://meiqia.com/	58649

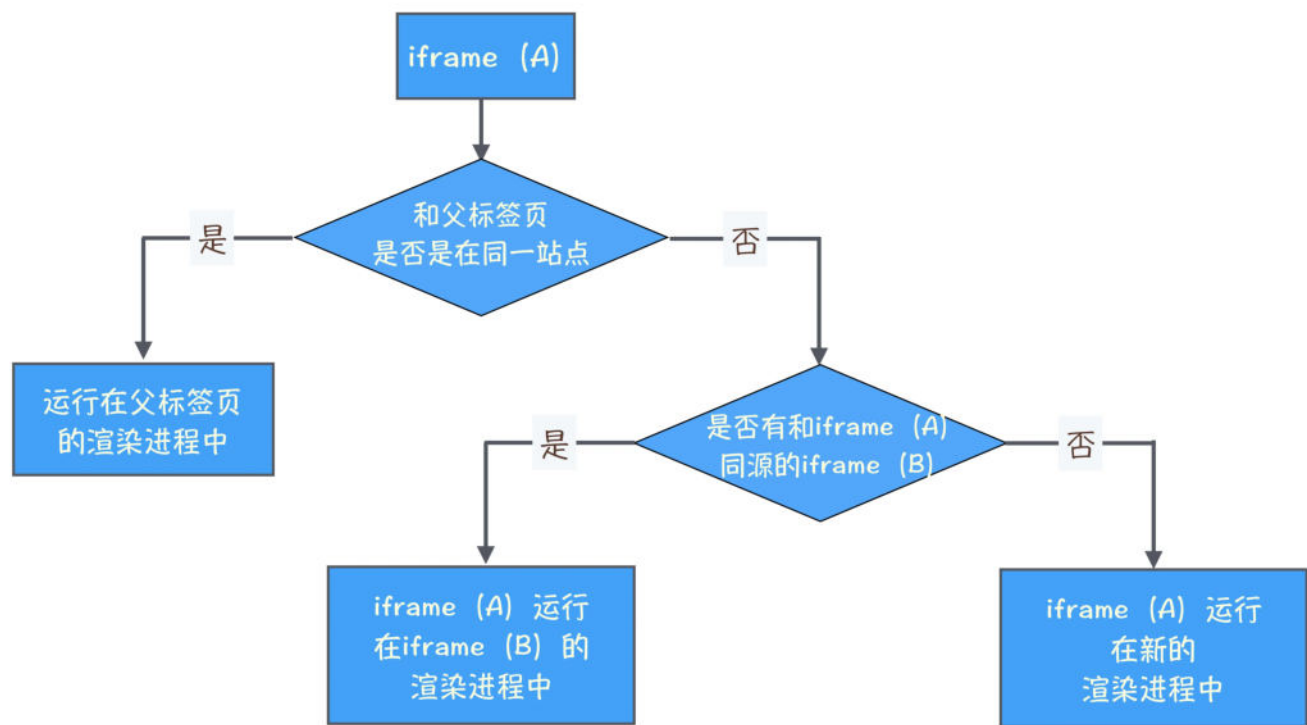
结束进程

我们可以看到，标签页中的属于同一站点iframe都使用了一个渲染进程

iframe 使用单独的渲染进程

结合上图和 HTML 代码，我们可以发现，由于 InfoQ、极客邦两个 iframe 与父标签页不属于同一站点，所以它们会被分配到不同的渲染进程中，而 iframe.html 和源标签页属于同一站点，所以它会和源标签页运行在同一个渲染进程中。下面是我画的计算 iframe 使用渲染进程数目的流程图，你可以对照着参考下：





计算 iframe 所使用的渲染进程数目

总结

好了，本节的内容就介绍到这里，下面我来总结下本文的主要内容：

首先我们使用了两种不同的方式打开两个标签页，第一种是从 A 标签页中通过链接打开了 B 标签页，第二种是分别打开 A 和 B 标签页，这两种情况下的 A 和 B 都属于同一站点。

通过 Chrome 的任务管理器我们发现，虽然 A 标签页和 B 标签页都属于同一站点，不过通过第一种方式打开的 A 标签页和 B 标签页会共用同一个渲染进程，而通过第二种方式打开的两个标签页却分别使用了两个不同的渲染进程。

这是因为，使用同一个渲染进程需要满足两个条件：首先 A 标签页和 B 标签页属于同一站点，其次 A 标签页和 B 标签页需要有连接关系。

接着，我们分析了一个“例外”，如果在链接中加入了 `rel=noopener` 属性，那么通过链接打开的新标签页和源标签页之间就不会建立连接关系了。

最后我们还分析了站点隔离对渲染进程个数的影响，如果 A 标签页中的 iframe 和 A 标签页属于同一站点，那么该 iframe 和 A 标签页会共用同一个渲染进程，如果不是，则该 iframe 会使用单独的渲染进程。



好了，到了这里相信你已经会计算渲染进程的个数了。

在最后我们还要补充下同源策略对同一站点的限制，虽然 Chrome 会让有连接且属于同一站点的标签页运行在同一个渲染进程中，不过如果 A 标签页和 B 标签页属于同一站点，却不属于同源站点，那么你依然无法通过 opener 来操作父标签页中的 DOM，这依然会受到同源策略的限制。

简单地讲，极客邦和极客时间属于同一站点，但是他们并不是同源的，因为同源是需要相同域名的，虽然根域名 geekbang.org 相同，但是域名却是不相同的，一个是 time.geekbang.org，一个是 www.geekbang.org，因此浏览器判断它们不是同源的，所以依然无法通过 time.geekbang.org 标签页中的 opener 来操作 www.geekbang.org 中的 DOM。


思考题

那么今天留给你的思考题是，你认为 Chrome 为什么使用同一站点划分渲染进程，而不是使用同源策略来划分渲染进程？

欢迎在留言区与我分享你的想法，也欢迎你在留言区记录你的思考过程。感谢阅读，如果你觉得这篇文章对你有帮助的话，也欢迎把它分享给更多的朋友。

分享给需要的人，Ta订阅超级会员，你将得 50 元

Ta单独购买本课程，你将得 20 元

 生成海报并分享

 赞 10  提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

[上一篇](#) 结课测试 | 这些浏览器的知识你都掌握了吗？

[下一篇](#) 加餐二 | 任务调度：有了setTimeout，为什么还要使用rAF？



JVM + NIO + Spring

各大厂面试题及知识点详解

限时免费 



精选留言 (25)

 写留言



tokey

2019-11-20

老师，阿里为什么要把同一站点的tab签做成无连接的，会避免什么安全隐患啊？

作者回复: 如果多个标签在同一个进程中，那么一个标签沦陷了，其它标签都会沦陷的



 57



Geek_177f82

2019-11-20

老师之前知否自己出过课程？或者书籍，博客之类的。希望老师提供下。

作者回复: 之前都是工作在一线，也没精力输出内容。

主要是现在自己搞了个小公司，时间上可以自由点了，那会刚好winter老师推荐我来极客时间写个浏览器专栏，我也没想就同意了。

博客我没有的，不过B站上到维护了一个up号，叫 二进制学院，那个账号翻译内容偏多！

关于资料，我计划在加餐结束后、整理份详细的发出来！





淡

2019-11-15

同源要求协议、域名以及端口均一样才行；同一站点只要求协议，根域名相同即可。也就是同源的要求太严格，导致复用同一渲染进程的条件比较难满足，所有条件放宽至同一站点？

作者回复：第一原因是通常同一站点安全性是有保障的
第二个原因就是提到的资源的复用了



15



Geek_259055

2019-11-15

老师期待你的Proformance加餐哦

作者回复：在规划中，一篇篇来

共 3 条评论 >

6



-_-III

2019-12-20

可能在编写“阿里云lot”这个项目是配置了eslint，所有的a标签必须加上 `rel='noopener noreferrer'`

共 1 条评论 >

6



大贝

2020-02-18

反过来思考的话，不同站点的应用不放在同一个渲染进程的原因可能是出于不信任的因素吧。

那么大致可以认为同站点下的不同源是该公司的各个子应用，所以从安全可信的角度来讲，放在同一个渲染进程是没有问题的。

但由于本质上是不同源的，所以不可以操作对方站点的 DOM。



4



Snow同學

2019-11-24

希望老师能出一篇，如何监测收集线上用户使用网站时的性能数据。
觉得虽然开发时进行了页面性能测试，但是用户使用时，可能还会出现很多我们的盲点未考虑



到。
线上监测感觉还是很有必要的

共 1 条评论 >

👍 4



james

2020-06-12

同源要求协议、域名以及端口均一样才行；
同站点只要协议、根域名相同就行
这样子相比较下同源的要求比较难满足，通常情况下同站一点就可以保障安全性，并且条件低就更容易满足渲染进程的资源复用，提高性能，减少不必要的开销



👍 3



CC

2019-11-24

老师你好，我有一个疑问。

在课程中，网络一直是当做一个进程来看待（network process）。

但是在查询资料的过程中，看到一篇 Google 的文章，作者说网络是浏览器进程内的一个线程（network thread）。文章附在最后。

这篇文章是 2018 年 9 月写的，是因为现在 Chromium 把网络线程从浏览器进程中拆分出来了吗？

提前谢谢老师。

文章：

Inside look at modern web browser (part 2)

<https://developers.google.com/web/updates/2018/09/inside-browser-part2>



👍 2



梦已沉沦

2019-11-15

是不是在多标签页时，同一站点比同源能有效节约进程

作者回复：这也是一个原因



👍 2



暴躁小胖



2020-03-09

李老师，我最近刚好有一个项目用到了iframe，iframe与父文档属于同一站点，但是我想能不能强制让iframe使用新的渲染进程，这样的话是否对页面性能会有提升，还能使用postMessage 在父文档和iframe之间进行通信。不知道行不行的通，请李老师指点

共 1 条评论 >

👍 1



江谢木

2019-12-13

老师，我在当前页面的控制台输入window.open('www.baidu.com')没有输入协议名，为啥会打开极客时间首页的新标签。

共 1 条评论 >

👍 1



Jimmy

2019-11-20

老师，想请教一个chrome内存的问题，就是我开了chrome 的任务管理器，我有看到内存占用空间远大于js内存，和GPU内存还有图片，css, js cache 之和，那要如何排查总内存是因为什么影响呢？。具体的场景就是，我有一个登录页面，本来内存大概150M, js 内存大概60M, 我登录到主页面使用一段时间，退出登录，但是我看到js 内存已经恢复了60M 左右，说明js 是没用内存泄漏的，不过问题是主内存显示确实500M, 并没有恢复到150M, 放了三个小时依旧如何，但我看GPU内存， js 内存， 图片cache 这些都已经正常了， 所以如何排查这500M 总内存到底是如何来的呢？



👍 1



hao-kuai

2021-12-24

总结:

1. a 标签和window.open 相同点是都有opener属性指向父标；不同点是后者在父标签上有new_window属性
2. 满足同一站点（协议和根域名相通，区别于同源策略的协议、域名、端口相同）和opener 即可复用同一渲染进程
3. 可以通过 a 标签设置ref 的值为 noopener 使得新标签的opener 属性为 null ，来阻止复用渲染进程



HXL

2021-10-14

老师，为什么我在 <https://www.geekbang.org/> 中点击标签 打开 <https://time.geekbang.org/>，在任务管理器中显示的是两个渲染进程呢，通过 open 方法则属于一个。而且我看跳转的 a 标签上面页面 rel属性. 有点奇怪



👍 1





Eval

2021-08-20

老师，知乎首页的每个问题的a标签并没有使用rel属性为什么不是共用一个渲染进程



neohope

2020-07-20

补充：

1、右键，在其他tab或其他window打开，会被视为不是一个浏览上下文组，会用不同的浏览进程

2、tab是否在同一window下，不会影响渲染进程的拆分及合并



Geek_baa4ad

2020-04-25

非常负责，居然还有加餐的。大赞



Cris

2020-01-01

老师，能简单说下同一站点河同源站点吗区别吗？



冯建俊

2019-12-20

我用的搜索引擎是必应：

搜索瓜子二手车 -> 点击跳转到瓜子二手车页面，我查看chrome浏览器任务管理,发现必应和瓜子同用一个渲染进程，然后我又从必应搜索列表点击跳转到另外一个网站，发现他们三个同用一个渲染进程，为什么呢？他们有浏览上下文的关系，但是主域都不同，它们不是应该各自用各自的渲染进程吗？不知道这么描述清晰吗？

作者回复: 我在bing中打开瓜子二手车的页面，新页面直接把bing页面替换了！

共 4 条评论 >

