

绝大多数情况下我们总是让JavaScript在DOM载入后再开始执行。不管是直接用 DOM API 实现还是使用 jQuery，最终都是DOMContentLoaded事件在起作用。本文讨论一个我们习以为常却很少了解的问题：样式文件的载入会延迟脚本执行，以及DOMContentLoaded事件的触发。

DOMContentLoaded事件

页面文档（DOM）完全加载并解析完毕之后，会触发DOMContentLoaded事件，HTML文档不会等待样式文件，图片文件，Iframe页面的加载。但DOM树已被创建，多数JavaScript已经操作DOM并成功能了。

This (DOMContentLoaded) event fires after the HTML code has been fully retrieved from the server, the complete DOM tree has been created and scripts have access to all elements via the DOM API. – molily.de

然而在绝大多数场景下，样式文件的载入会延迟DOMContentLoaded事件的触发。其实这样的行为正是开发者所希望的，为什么呢？

事实上，老版本的IE中DOMContentLoaded事件存在兼容性问题。参见：[兼容所有浏览器的 DOM 载入事件](#)。

浏览器为何延迟DOMContentLoaded

对于很多脚本而言，它们被编写时就希望在样式载入之后再开始执行。JavaScript的作者往往会假设CSS规则已经生效，尤其是在进行一些显示相关的操作时，比如需要得到DOM元素的位置和大小的场景。事实上，在多数浏览器中DOMContentLoaded事件的触发会考虑到外部样式文件（CSS）的载入，以及在HTML中脚本标签和样式标签的相对位置。如果脚本位于样式之后，浏览器通常会认为该脚本依赖于样式

的渲染结果，也就更倾向于延迟脚本的执行（直到样式渲染结束）。

不同浏览器的行为

既然浏览器有时会延迟`DOMContentLoaded`事件，但是何时会延迟`DOMContentLoaded`事件，还取决于行内脚本还是外部脚本，以及脚本与样式标签的相对位置。不同的浏览器渲染引擎也有不同的行为。在<http://molily.de/domcontentloaded/>一文中对该问题有详尽的阐述和测试，本文取其结论。

下表描述了各种情况下脚本是否会被延迟执行，进而延迟触发`DOMContentLoaded`事件。

渲染引擎	样式表之前的脚本	样式表之后的外部脚本
Presto (Opera)	否	否
Webkit (Safari, Chrome)	否	是
Gecko (Firefox)	否	是
Trident (MSIE)		是

HTML5标准及最佳实践

其实`DOMContentLoaded`是Firefox中最先提出的，此后JavaScript社区发现它确实比`load`事件（要求所有资源完全载入）更好，于是Apple和Opera相继开始支持该事件。但不同浏览器的实现方式有所区别，于是产生了上表所示的复杂情况。

在HTML5标准中情况有所好转：`DOMContentLoaded`是一个纯DOM事件，与样式表无关。与此同时，HTML5要求：

- 脚本执行前，出现在当前之前的必须完全载入。
- 脚本执行会阻塞DOM解析。

这样的话，假如脚本和样式一起放在HTML中，DOM解析到标签时会

阻塞DOM解析，开始如下操作：

1. 获取当前的脚本文件；
2. 获取并载入前面的所有。
3. 执行当前脚本文件。

这些操作完成之后才能继续进行DOM解析，解析完毕时再触发

`DOMContentLoaded`事件。如果将样式和脚本都放到中，会使浏览器在渲染前载入并执行所有样式和脚本。页面的显示延迟会很高，多数情况下用户体验都很糟糕。因此在HTML5标准的HTML页面中，最佳实践应当是：所有样式放在中；所有脚本放在最后。

[jQuery文档](#)中也推荐这样的实践方式。

参考阅读

- MDN `load` 事件：<https://developer.mozilla.org/en-US/docs/Web/Events/load>
- MDN `DOMContentLoaded`事件：<https://developer.mozilla.org/zh-CN/docs/Web/Events/DOMContentLoaded>

本文采用 [知识共享署名 4.0 国际许可协议](#)（CC-BY 4.0）进行许可，转载注明来源即可：<https://harttle.land/2016/05/15/stylesheets-delay-domcontentloaded.html>。学识粗浅写作仓促，如有错误辛苦评论或 [邮件](#) 指出。