








-  我的文章
-  发起活动
-  我的问题
-  我的收藏
-  我的圈子
-  我的博客
-  个人设置

退出



我要



花梅
 2017-01-25 11:12:55 (最初创作于: 2017-01-24 21:12:08)
 发表于: [AllExpress 技术团队](#) >> [前端技术](#)

50 阅读

0 收藏

知识体系: [修改知识体系](#)

文章标签: 前端与交互设计 [修改标签](#) [标签历史](#)

附加属性: [作者原创](#)

# \$(document).ready()与window.onload()的执行时机

## 一、概要

\$(document).ready()就是我们经常使用的\$(function()), 每当我们写这样的代码时, 不知道是否足够清楚到底会在什么时候执行, 它与window.onload有什么区别? 今天深入了一下, 将结果分享给和我有同样困惑的人。

## 二、理论知识

网上搜了很多关于\$(document).ready()与window.onload()的执行时机的资料, 比较权威的回答可以参考这个:

<http://stackoverflow.com/questions/3698200/window-onload-vs-document-ready>。

要点如下:

ready是jquery特有的, 它在html document被加载/解析完时执行, 而onload是dom的标准事件, 它在所有内容 (如img) 被加载后执行。

对于前端知识不扎实的我来说, 问题来了, onload的解释是容易理解的, 全部资源下载完成嘛 (浏览器标签页的表示正在下载的菊花没了), 但什么叫"html document被加载完时执行", 表示没看懂, 比如css要不要下载完成, 在\$(function())外面的js代码要不要执行完成? 于是, 为了理解这句话, 我知道还得去了解html解析过程才行, 关于html解析过程可以参考这两篇文章:

<http://blog.csdn.net/zzaquarius/article/details/6532299/>

<http://www.cnblogs.com/dojo-lzz/p/3983335.html>

说下主要结论:

- 根据你返回的html, 浏览器从上往下解析dom
- js的执行会阻塞dom解析 (因为js可能会改变dom)
- 外部css样式表的下载会阻塞js的执行 (注意不会阻塞下载)

根据上述信息, 我们推测, ready的触发时机, 需要外部css下载全部完成且同步调用的js全部执行完成

## 三、实践验证

没有验证过的理论知识总觉得不可靠, 我们写一段代码来验证下, 代码如下:

```

<html>
  <head>
    <script src="jquery.js"></script>
  </head>
  <body>
    <img id="myImg"></img>
    <script type="text/javascript">
      console.info("[outside function 1]entry time "+new Date());
    </script>
    <!--这个js存在, 下载会很快-->
    <script src="https://i.alicdn.com/ae-store-ui/?hashmap.360e9b27.js"></script>
    <!--这个css不存在, 下载会挂起很长时间, 用来验证是否会阻塞后续js的执行-->
    <link rel="stylesheet" href="http://www.facebook.com/daq.css"></link>
    <span>i'm here</span>
    <script type="text/javascript">
      $(function(){
        console.info("[ready function]entry time "+new Date());
        var sleepTime=5000;
        console.info("[ready function]i'm ready to sleep "+sleepTime+" ms!");
        sleep(sleepTime);
        console.info("[ready function]i'm trying to load img,time: "+new Date());
        //动态下载图片, 看是否会阻塞onload事件
        $("#myImg").attr("src","http://www.facebook.com/adsada1.png");

      });
      $(window).load(function(){
        console.info("[load function]entry time "+new Date());
      });
      console.info("[outside function 2]entry time "+new Date());
      var sleepTime=3000;
      console.info("[outside function 2]i'm ready to sleep "+sleepTime+" ms!");
      sleep(sleepTime);
      function sleep(numberMillis) {
        var now = new Date();
        var exitTime = now.getTime() + numberMillis;
        while (true) {
          now = new Date();
          if (now.getTime() > exitTime)
            return;
        }
      }
    </script>
  </body>
</html>

```

执行结果 (取chrome上验证结果, firefox和ie也验证过, ie略有不同) :

```

Uncaught ReferenceError: module is not defined
    at jquery.js:9475
[outside function 1]entry time Tue Jan 24 2017 16:47:32 GMT+0800 (中国标准时间) 这个在js和css下载之前, 其中css下载会很慢
Uncaught ReferenceError: seajs is not defined
    at 2?hashmap.360e9b27.js:2
GET https://www.facebook.com/daq.css net::ERR_CONNECTION_TIMED_OUT
[outside function 2]entry time Tue Jan 24 2017 16:47:53 GMT+0800 (中国标准时间) 这个在js和css下载之后, 之所以这么久是因为需要等css下载完成
[outside function 2]i'm ready to sleep 3000 ms!
[ready function]entry time Tue Jan 24 2017 16:47:56 GMT+0800 (中国标准时间) 这个是在read函数中, ready函数外有个sleep, 证明了需要外面的js执行完才能讲ready!
[ready function]i'm ready to sleep 5000 ms!
[ready function]i'm trying to load img,time: Tue Jan 24 2017 16:48:01 GMT+0800 (中国标准时间)
GET https://www.facebook.com/adsada1.png net::ERR_CONNECTION_TIMED_OUT
[load function]entry time Tue Jan 24 2017 16:48:22 GMT+0800 (中国标准时间) 这个是load函数, 与ready函数间隔很长, 因为ready函数加载了一个img, img超时后才执行

```

## 四、结论

根据理论知识和实验结果, 我们暂时可以得出如下结论:

- 1、ready是jquery特有的, 它在html document被加载/解析完时执行, 根据html的渲染过程原理, 这意味着您的外部链接或内链的css文件和js文件需要全部下载完成 (或下载失败) ,并解析/执行完成。
- 2、onload是dom标准事件, 在所有内容被加载/解析完后执行, 包括img这些资源。这意味着不会早于ready发生。

document\_ready与window\_onload的执行时机.docx.gzip

[评论文章 \(0\)](#)
[👍 0](#)
[🔄 0](#)
[➡](#)
[0 收藏](#)

相似文章

- [百阿自动填写反馈表](#)
- [Midway 插件和应用异步初始化方案](#)
- [Zone.js 简介 & 抛砖引玉](#)

- [非主流node.js异步转同步](#)
- [iOS程序猿的前端开发之路](#)
- [不见得知全豹而窥一斑 - 一周koa, e...](#)

上一篇: [XSS攻击原理与防御实践](#)

下一篇: [AE触达业务介绍-通知消息业务](#)

写下你的评论...



评论