

对于 JQuery，我们真的要说再见了么？

jQuery 的辉煌

jQuery 最初诞生于 2006 年 8 月，作者是 John Resig

(<https://zh.wikipedia.org/wiki/JQuery>)。10 多年前，网页开发者（当时还没有“前端”这个概念）深受浏览器不兼容性之苦。以 jQuery 为代表的一批 JavaScript 库/框架应运而生：

Dojo

jQuery

MooTools

Prototype

Yahoo! User Interface Library (YUI)

Script.aculo.us

这些库有的像 jQuery 一样是主打通用性，比如 Dojo、Prototype；有的则走更专业化的路线，比如 YUI（组件化）和 Script.aculo.us（动画交互）。从通用功能看，这些库大都提供了如下特性：

语法糖：链式调用、实用函数

选择元素：基于各种标准选择符和库自定义的选择符，以及通过回调进行筛选操作

DOM：创建和操作元素，乃至各种属性操作

处理事件：ready 事件及各种注册和触发事件的方法，乃至委托

动画：基于 animate 的 CSS 属性动画，以及内置动画方法

Ajax：封装原生的 XMLHttpRequest API，简化请求方法及回调处理

一批宣传和推广这些库和框架的书籍相继面世，迅速推动了这些库的采用。2010 年出版的 DOM Scripting, 2nd (《JavaScript DOM 脚本编程艺术 (第 2 版)》) 就是其中之一。这本书专门有一个附录，介绍当时流行的库，主推 jQuery。

当时 jQuery 官网的自我描述是这样的：

“a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development. jQuery is designed to change the way that you write JavaScript.”

不错，正是这样。jQuery 凭借其选择器方法、面向集合的链式调用、对事件和 Ajax 的完善支持、简单而强大的动效等独有优势，迅速风靡，成为几乎所有网站开发必备的 JavaScript 库，乃至开创了 jQuery 编程风格。

开发者选择 jQuery 的理由简单直接：

跨浏览器
简单高效
稳定可靠
插件丰富

2011 年新版的“犀牛书”第 6 版——JavaScript: The Definitive Guide, 6th 甚至拿出第 19 章整整 64 页篇幅隆重讲解了 jQuery（“Chapter 19. The jQuery Library”）。

jQuery 从此走向鼎盛和辉煌。后来，随着前端交互越来越重和移动应用的普及，jQuery UI、jQuery Mobile 相继面世。

时至今日，jQuery 仍然在支撑着数以千万计各种规模网站的运作——尽管聚光灯下已经不常看到她的身影。

历史的选择

最近 10 年，是“前端行业”有史以来发展最快的 10 年。

移动社交时代的到来不仅没有让桌面 Web 失色，反倒刺激了 Web 标准的迅猛改进。HTML5 不仅带来了极大的向后兼容性，也带来了更丰富的原生 DOM API。CSS 从 CSS3 开始走上模块化的快车道，文本样式、排版布局、媒体查询，各种新模块让人目不暇接。

各大主流浏览器也在快速跟进，Firefox、Chrome、Opera、Safari、IE 乃至 Edge，都在积极重构甚至重写内核，争做支持 Web 标准的“楷模”。在这个大背景下，各大互联网公司不断调高兼容的 IE 版本号，从 8 到 9 到 10，再到 11。

当然，还有 ECMAScript 语言标准。自从划时代的 ES6（ECMAScript 2015）发布之后，JavaScript 终于真正开始摆脱“玩具”语言的尴尬境地。更重要的，从 ES6 起，ECMAScript 也进入了快速迭代、每年发一版的节奏。ES7、ES8，以及 ES9，每次都会给这门语言注入更强大的语言特性。

与此同时，Node.js 和 Babel 等服务端运行时及转译工具的出现，也让前端工程化，以及向传统工业级软件开发最佳实践靠拢的速度日益加快。

2012 年，笔者在图灵社区翻译过一篇文章“JavaScript 宝座：七大框架论剑”（<http://www.ituring.com.cn/article/8108>）。当时的“七大框架”是：

Backbone
Knockout
Spine
CanJS
Ember
AngularJS
Batman
Meteor

后来，谷歌主打 SPA (Single Page Application, 单页应用) 的 Angular 终于一枝独秀。不久，脸书推出的“在 JS 里写 HTML 一样优雅”的 React 则一路高歌猛进。最终，集各家所长且简单易用的 Vue 横空出世。

前端开发已经从后“刀耕火种”时代的“农业文明”，逐渐进化为以大规模、可扩展、规范化、自动化为特征的准“工业文明”。

俗话说：“皮之不存，毛将焉附。”随着时代变迁、技术进步，jQuery 赖以存在的环境正逐渐消失。如前所述，新的环境催生了一批框架新秀。曾经辉煌的 jQuery 终于走到了可以华丽谢幕的时刻。

几句忠告

8 年前，DOM Scripting, 2nd 在推荐开发者使用前端库时，语重心长地说过下面这段话：

“开发中使用库当然没问题，但前提是不要仅仅只知道如何使用，更要知道它的工作原理。如果不能超越这些库，那在这些库变成你的拐杖之时，你也会随之“残废”。在使用某个库之前，一定要花时间学习掌握 JavaScript 和 DOM。我们从一开头就强调“知其所以然”的重要性，告诉大家不能满足于仅仅“知其然”。如果使用了库，却不晓得其背后原理，无论对自己还是对产品都将是有害的。”

2012 年，从后端转前端不久的 Ray Nicholus 接手了一个跨浏览器文件上传库 (<https://github.com/FineUploader>) 的维护和开发工作。他的第一反应就是用 jQuery 重写这个库。然而，用户反对引入任何对其他库的依赖。结果他只能使用原生浏览器的 API。那又怎样呢？他发现，原来放弃 jQuery 的写法比自己想象的要容易。

2014 年，Ray 发表了：“You Don't Need jQuery!” 系列博客

(<https://blog.garstasio.com/you-dont-need-jquery/>)。2016 年，他的著作 Beyond jQuery (Apress, <https://www.amazon.com/dp/1484222342/>) 出版。Ray 建议，学习前端开发，本来应该是这个顺序：

学习 JavaScript

学习 Web API

学习 jQuery (或其他框架/库)

但很多人实际上是从 #3 开始的，然后过了很久才开始 #1 和 #2 (甚至从来没想到 #1 和 #2)。结果就是“只知其然，而不知其所以然”。

前端的核心是 js，外围打酱油的技术太多，不一一列举。

css 就像一瓶酒，得品。

html, css 总共就那些标签跟选择器属性什么的，但是要写一个有扩展性，健壮性或维护性的页面不容易。现在写页面基本条件反射，不是如何快速的完成，而是思考如果有前端界面需求修改，怎么在修改代码最少的情况下完成整体需求任务。

js 就像一把剑，得磨。

js 刚开始只是为了较验，随便技术社会的发展，承担的角色越来越重，刚开始玩玩 jQuery 感觉已经会 js 了，其实只是冰山一角。随着对 js 的了解越来越多，

他即变态又可爱，即好玩又难控，即有很多兼容问题，但解决兼容是我们基本生存之道。

人生就是一场梦，得作。

技术只是生活的一部分，曾经雄心斗志，如今低头写码。改变能改变的，接受不能改变的。人生有限，兄争朝夕啊。

最后说，什么时候能阅读 jQuery 源码，感觉其码是纯 js 编码一年以上，js 基础知识没有盲点。阅读源码，确实能学到很多东西。不只是技术，而是一种生活的态度。一种把一件事情做到极致的态度(糖饼语)。

如今，时移世易，jQuery 即将谢幕，新一代组件化开发框架粉墨登场。然而，“太阳底下没有新鲜事。”不管是使用 Angular，还是 React，抑或是 Vue，还是要真正理解它们背后的工作原理。

尾声

jQuery 作为第一代前端库的优秀代表，成就了今天 Web 的繁荣，也成就了一代 Web 开发者。

