首先 搜集关于JS的书籍中有关历史的论述

<https://weread.qq.com/web/reader/56d32bd05b230f56d1828f0>

**你不知道的JavaScript序言**

我相信你注意到，但是这系列丛书标题中的“JS”并不是一个用来诅咒JavaScript的缩写词，尽管我们可能经常咒骂这门语言的奇怪特性。

在互联网发展的早期，JavaScript就已经成为了支撑网页内容交互体验的基础技术。那时JavaScript的作用可能仅仅是生成一些闪烁的鼠标轨迹或者烦人的弹出窗口，但是经过了大约20年的发展，JavaScript的技术和能力都发生了天翻地覆的变化，现在的JavaScript毫无疑问已经成为了世界上使用范围最广的软件平台————互联网————的核心技术。

但是作为一个语言来说，它总是成为大家批评的对象，部分原因是它有很多历史遗留问题，但主要原因是它的设计哲学有问题。就像Brendan Eich曾经说过的，JavaScript甚至连名字都给人一种“蠢弟弟”的感觉，就像是它更成熟的大哥Java的不完整版本。不过名字只不过是营销策略上的一个意外，这两个语言有许多本质上的区别。JavaScript和Java的关系，就像Carnival（嘉年华）和Car（汽车）的关系一样，八杆子打不着。

JavaScript借鉴了许多语言的概念和语法，比如C风格的过程式编程以及不太明显的Scheme/Lisp风格的函数式编程，因此吸引了许多开发者，甚至是那些不会编程的新手。用JavaScript来编写“Hello World”是非常简单的，因此这门语言很有吸引力并且很好上手。

虽然JavaScript可能是最早出现的语言之一，但是由于其本身的特殊性，相比其他语言，能真正掌握JavaScript的人比较少。如果想用C、C++这样的语言编写功能全面的程序，那需要对语言有很深的了解。但是对于JavaScript来说，编写功能全面的程序仅仅是冰山一角。

JavaScript语言本质上有许多复杂的概念，但是却用一种看起来比较简单的方式体现出来，比如回调函数，因此JavaScript开发者通常只是简单地使用这些特性，并不会关心语言内部的实现原理。

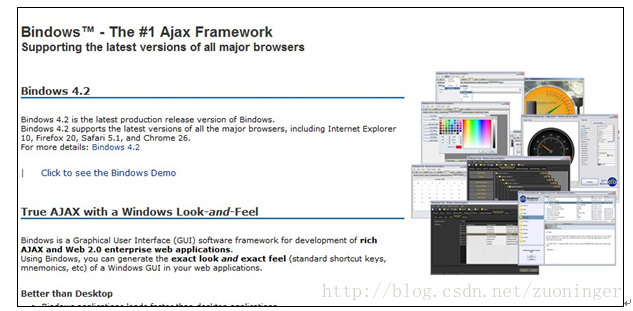
JavaScript既是一门充满吸引力、简单易用的语言，又是一门具有许多复杂微妙技术的语言，即使是经验丰富的JavaScript开发者，如果没有认真学习的话也无法真正理解它们。

这就是JavaScript的矛盾之处，也是这门语言的阿喀琉斯之踵。由于JavaScript不必理解就可以使用，因此通常来说很难真正理解语言本身，这就是我们面临的挑战。

**2．JavaScript年发展历程**

第一个年代，洪荒时代。从1995年到2005年，就是从JavaScript发明到Ajax概念[1]的提出。其间打了第一场浏览器战争，IE VS Netscape。这两者的DOM API出入很大，前端开发人员被迫改进技术，为了不想兼容某一个浏览器，发明UA（navigator.userAgent）嗅探技术。

这个时期的杰出代表是Bindows[2]，2003年发布，它提供了一个完整的Windows桌面系统，支持能在EXT看到的各种控件，如菜单、树、表格、滑动条、切换卡、弹出层、测量仪表（使用VML实现，现在又支持SVG）。现在版本号是4.x，如下图所示。



其他比较著名的还有，Dojo（2004年）、Sarissa（2003年）、JavaScript Remote Scripting（2000年）。

Dojo有IBM做后台，有庞大的开发团队在做，质量有保证，被广泛整合到各大Java框架内（ struct2、Tapestry、Eclipse ATF、MyFaces）。特点是功能无所不包，主要分为Core、Dijit、DojoX三大块。Core提供Ajax、events、packaging、CSS-based querying、animations、JSON等相关操作API。Dijit是一个可更换皮肤，基于模板的Web UI控件库。DojoX包括一些新颖的代码和控件，如DateGrid、charts、离线应用和跨浏览器矢量绘图等，如下图所示。



JavaScriptRemote Scripting是较经典的远程脚本访问组件，支持将客户端数据通过服务器做代理进行远程的数据/操作交互。

Sarissa封装了在浏览器端独立调用XML的功能。

第2时期，Prototype“王朝”，2005年～2008年。其间打了第2次浏览器“战争”，交战双方是IE6、IE7、IE8 VS Firefox 1、Firefox 2、Firefox 3，最后Firefox3大胜。浏览器“战争”中，Prototype积极进行升级，加入诸多DOM功能，因此，JSer比之前好过多了。加之，有rails、script.aculo.us（一流的特效库）、Rico等助阵，迅速占领了市场。

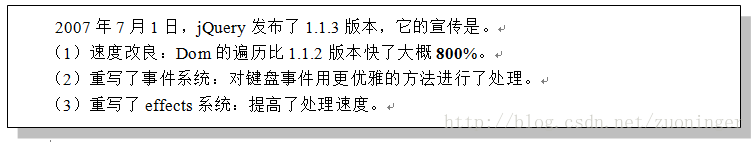
Prototype时期，面向对象技术发展到极致，许多组件成套推出。DOM特征发掘也有序进行，再也不依靠浏览器嗅探去刻意屏蔽某一个浏览器了。无侵入式JavaScript开发得到推崇，所有JavaScript代码都抽离到JavaScript文件，不在标签内“兴风作浪”了。

Prototype的发展无可限量，直到1.5版本对DOM进行处理，这是一个错误[1]。比如它一个很好用的API-getElementsByClassName，由于W3C的标准化，Prototype升级慢了，它对DOM的扩展成为了它的“地雷”。

第3时代，jQuery纪元，2008年到现在（如下图所示）。



jQuery则以包裹方式来处理DOM，而且配合它的选择器引擎，若一下子选到N个元素，那么就处理N个元素，是集化操作，与主流的方式完全不一样。此外，它的方法名都起得很特别，人们一时很难接受。



停滞不前的Prototype已经跟不上时代的节奏，jQuery在1.3x版本时更换Sizzle，更纯净的CSS选择器引擎，易用性与性能大大提高，被程序员一致看好的mouseenter、mouseleave及事件代理，也被整合进去，jQuery就占据了市场。