是什么让我们爱上JavascriptJava认证考试 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/644/2021_2022__E6_98_AF_E4_BB_80_E4_B9_88_E8_c104_644473.htm 过去，人们对于Javascript一直报着轻视的态度，人们认为它又慢又容易出错，而且在不同浏览器中解释也不一样，但是现在，Javascript 确确实实的在改变我们的网络，越来越多的网络和APP应用开始使用Javascript.今天，我们就来讲讲我们为什么会有这种观念，回顾 Javascript的历史来诊断他的现状，同时通过一些片段来表明Javascript同其他开发语言的本质不同，唯有如此我们才能明白为何 Javascript这么重要。 早期的迷惑 从某种程度上来讲，Javascript是Java的恶魔兄弟（Javas evil twin）。他们同岁，都于1995年以Beta的版本出现，并且都在次年1996年推出了1.0版本。在语法上他们也很相近，从名称上来看他们就像是一家人一样。 我们第一次听说Javascript是在1995年，当时Netscape推出了Navigator 2.0的Beta版本，这个版本中没有包含Javascript，但是有对于Java Applet的支持。同时，Netscape 宣布页面内嵌的语言LiveScript.这个声明并没有引起太多的注意，在当时Java Applet是一个热门的技术，当时对于为什么Netscape要在浏览器中内嵌两种语言也不是十分清楚，难道有什么事情LiveScript能做而 Java做不了的么？ 2个月以后，LiveScript看到了曙光，在Navigator 2.0B3中，它被重新命名为JavaScript.这次改动吸引了众多人的关注。“ Oh，他肯定不能体现他的价值。”、“ 为什么Sun要让这么个玩具来搭Java的车？ ”我至今仍人为这次改名是一个失败的主意，因为他引起了无尽的困惑，很多非编程人员从来都没有明白

过Java和Javascript是两个不同的东西。 改名字的另外一个影响是细微的，但是我认为改善了这个语言的接受程度。与Java的关系使这门语言看起来并不是一个全新的东西。事实上，当时Java本身都还不是十分成熟，也还没有到1.0，但是外界对于Java成熟状况的认知要比本身高出很多。例如，在1995年，时代杂志将Java选为当年的十大新产品之一。同时，市面上已经有很多关于Java的书籍。所以，当Netscape将这项技术命名为JavaScript，其意图很明显就是表明当时的 Java开发团体是使用和评价JavaScript的最佳人选。 在怀疑的浪潮中，凭借JavaScript与Java Applet的通信和控制能力，JavaScript开始出现在一些应用中。所以，那些对Java感兴趣的人开始觉得有了学习JavaScript的必要。 Javascript就像一个玩具，它是面向对象的，所有的东西都是公有的，没有封装。另外，你无法创建真正的子类，在实例变量和方法之间并没有真正的区别，这就像一个灾难。你可以在任何时候将一个变量变成一个方法。在这方面我并不是孤独的，虽然对于面向对象的细节每个人的理解都有所不同，但是Java程序员几乎无一例外的认为，JavaScript并不是一门严肃的编程语言，我们也不像用它来做些什么。 附加的伤害 在早期的时候，JavaScript并没有什么真正的技术问题，它是缓慢又充满风险的，并且只在Netscape下工作。后来IE也支持了，但并不是完全兼容。人们吹捧他用来编写页面的applets，但是它并不能检测applet的失败。它从来没有被广泛的使用过，很快他就成为了一个分母，人们仅仅用它来完成那些可靠的任务，并且避免在其他的地方使用。 什么是JavaScript能够可靠的支持的？闪动的、基于图像的视觉效果。人们开始制造一些炫目的效果。Netscape引导了

这个趋势，在页面中使用令人生厌的图层干扰了人们对正常内容的浏览。那些各种各样闪动、滚动的效果让我们认为这个标签简直就是网络的原罪。到处都充斥着闪动的Banner、滚动的Banner、状态栏里跳动的文字、随时可能弹出的对话框、……各种各样的滥用还在不断继续。Miscorsoft也开始谈论DHTML，但是不久 DHTML就变成了Buggy、Slow 、Unmaintable网站的代名词。 对了，不要忘记安全问题。一般来讲，当你构建一个图灵完备（Turing-Complete）的编程语言时，你就需要特别小心安全问题。Netscape做的不够好，早期的Javascript中存在漏洞，而且花了很长时间才完全解决掉。 结果是什么？很多的有技术头脑的用户关闭了浏览器的Javascript支持。要知道，在1996年的时候可以被称为有“ 技术头脑（Technically Savvy） ”的人员站的比重相当的高。所以，这些人便不再依赖于JavaScript，因为这会导致用户不使用你的网站。 那些使用JavaScript的用户做的也不够好。那些严谨的程序员完全抛弃了JavaScript，但是设计师和页面人员补充了进来。早期的书籍中都提供了糟糕的JavaScript实例，但是很少去从语义上对它进行分析、阐述。（David Flanagan是个例外）让事情更糟糕的是，有些人（也许是大多数人）根据就没从书中学习过JavaScript，他们只是从一个网站现有的代码中拷贝，修改，然后粘贴到自己的网站上让它工作，到最后都不知道为何这些代码能够正常运行。 即使大家都是从例子开始学习JavaScript，只要保证两件事情这也许会成为一个很好的策略。第一个就是大家都学习好的例子，但显然JavaScript并不属于这种情况。第二种情况就是这种语言应该和学习它的人所知道的另外一种语言相似，但是很不幸，虽

然JavaScript从语义上看来和Java非常相近，但实际上JavaScript来自一个和Java完全不同的编程语言家族。JavaScript是NewtonScript、 Self、Smalltalk……和Lisp语言的一个直系后裔。Waldemar Horwat，一位在JavaScript早期产生重大影响的工程人员说，我更愿意把JavaScript认为是Common Lisp的另一种形式。这个说法有些夸张，但是如果你知道这两种语言，你就会很容易发现两者之间确实存在很多相似之处。 所以状况就变的越来越不好，一门与之前广泛传播的语言有很大不同的新语言，被一些没有经验的编程者推广，其他的编程人员Follow了他们的例子。更多的用户基于安全考虑关闭了JavaScript的支持，还有有经验的程序员推荐大家避免使用JavaScript.上面的这些还不够，浏览器大战似乎一触即发，儿JavaScript被选作了一种武器。战争中的双方都不断的发展JavaScript，有时候甚至故意引入了很多不兼容特性。JavaScript此时已经变得更庞大，但同时也更容易引起问题。也难怪没有人会喜欢她。 帝国的重建 在那段时间，我还是不断听到更多关于JavaScript的消息。关于基于prototype和面向对象的特性的介绍总是让我很好奇，我曾经看到过的教材里从来没有提到过这些，也许是作者本人也没有必要的语言背景去将这些概念联系起来。同时，人们开始指出由于浏览器造成了多少错误，而不是 JavaScript本身。 后来，IE赢了，Netscape投降了，这一阶段的浏览器大战结束了。Miscrosoft开始将他们的bug光芒覆盖到CSS领域（任何处理过IE6下页面开发的工程师都应该深有感受），而Mozila的团队则开始认真的考虑兼容性的问题。除了一些根深蒂固的区别，想要修改他们必须大量的修改源码，这个小组修复大多

数的不同，于是IE和Mozilla/Firefox下的JavaScript变得更可控了。其他的开发者也更加的有迹可循。 JavaScript和浏览器的内置支持开始进入了一段稳固的发展时期。在我们大家都忙着写自己的JavaScript代码的时候，几乎没有人注意到一个具有传奇色彩的特性引入-XMLHttpRequest.最重要的DHTML特性，Ajax的重要部分，悄悄的加入到了IE中来。Microsoft的Outlook小组引入了这个ActiveX控件来使Outlook支持网络访问。200年的Ajax Experience的大会上，Dion Almaer和Ben Galbraith做了很多关于XMLHttpRequest的工作，但是在Mozilla第一次引入XMLHttpRequest支持的时候的官方文件中，他的特性根本就没有提到。 那段时间有很多JavaScript的活动，在JavaScript的支持下，Macromedia为Flash提供了脚本语言支持。Adobe也在不断努力使他们的很多程序可以使用JavaScript来控制和扩展。Apple将JavaScript嵌入了它的Sherlock应用中。当然，Mozilla项目也做出了一个重要决定，他们大部分的浏览器都可以支持JavaScript.很明显，JavaScript已经不再仅仅是一个网页语言，而成为了程序中嵌入的动态的、运行时扩展的编程语言的首选。其中的原因也许是当时大家或多或少的都会一些JavaScript，还有当时存在两个高质量的嵌入扩展（SpiderMonkey c语言编写，Rhino Java语言编写）。 在2000年的时候，一些人（著名的Brent Ashley、Alex Russell、Douglas Crockford）开始注意到JavaScript究竟能做什么。Brend开始研究JavaScript与服务器通信的方法，提出了我们现在公认的Ajax Style.而且在当时浏览器对于XMLHttpRequest支持不是很完善的时候，他增加或者形成了很多完成这项处理的很多巧妙的方法。他建立了一个网站“ Remote Scripts

Resources ”，并且编写了一个JavaScript Remote Scripting（JSRS）库来支持不同浏览器的兼容性。 Alex Russel建立了一个netWindow项目，意图建立一个支持富UI、可编程、图形环境的网页，同时带有完全可拖动的窗口和其他小组件，netWindows变成了nWigets，并且最终催生了Dojo Project，一个当前重要的Ajax框架。 Douglas Crockford一直在研究JavaScript语言的丰富性，最后他发布了一系列有建设性的文章，例如：The Worlds Most Misunderstood Programming Language，在这些文章中，他指出了JavaScript在使用中的一些技巧和一些缺点。 Brent、Alex和Doug当时的声音都很薄弱，就像旷野中的呼声，但是当最后其他的Web开发社区准备将注意力投向JavaScript的时候，他们所作的这些工作让我们的生活变得更加简单了。 我在2000年到2003年中段的时候，对于JavaScript的关注都只是一些零星的片段，我确实没有投入太多的关注，我沉浸在Java的乐土中，闲暇的时候研究一下Ruby，而正是这段经历为我再一次的认识JavaScript提供了准备。 在2003年的6月，我的朋友给我演示了netWindows，当我从地板上捡起我的下巴来以后（After i pick my jaw up off the floor，作者还真是幽默），我朋友告诉我说Alex这个人不是疯子，我如梦方醒开始认真的研究JavaScript，现在看来，当时Google也有一帮人在做同样的事情。 伟大的革命 每个看这篇文章的人都应该知道一些Ajax第一次引爆屏幕的情景，今天我来简短的说一些完整的情节。 第一个引起我注意的Ajax应用是Gmail.很明显，他先是下载了很多的JavaScript，然后在后台不断地与服务器进行通信，避免页面的刷新。他确实给人非常深的印象，同时也给其他的技术团队带来不知所措的

感觉。我当时在想Google一定是有一个非常成熟的定制化工具来开发如此复杂的客户/服务器端管理。 真正让我明白的是Google的Google Suggest.现在的人一定很熟悉，当我们输入的时候，通过与服务器的交互将我们可能需要的词条显示出来供我们选择。 Google Suggest在两个方面让我印象深刻，第一是他是如此简单以至于大多数的技术人员都能够明白他的工作原理并且应用到自己的网络项目中；另外一个是 Google Suggest针对用户的每次keyup事件进行相应和通信，让我看到同服务器通信是如此的简单，我完全被折服了。 不久以后又袭来了Google Maps，Jess James Garret形成了Ajax这个术语，从此以后讨论变得更加方便。 从此以后，Ajax变成了我们处理网络应用的的标准方法。而且现在我们有了更多的工具和第三方库选择，但我们又陷入另一个难题：选择太多了。 停止担心，爱上DOM 我已经阐明了为什么JavaScript有这个名声的众多原因，其中有好的也有坏的，在这众多的原因中，我只想重复一条，他是如此重要，能够解释为什么很多有经验的程序员经给被自己的经验所蒙蔽。 这一部分作者以一个例子来说明了JavaScript的编写方式和Java是多么的不同，最后提出了一些有意义的总结：Functions are first-class objects. Methods are just functions attached to objects. You can add methods to classes at any time （even after instances have been created）。 Individual objects can have their own methods. ``Class constructors are just functions. Functions， being objects， can have their own properties. You can call functions with fewer （or more） arguments than the function is declared to take. If no value is passed for a function argument， it gets the value undefined. 总结最后让

我们每一位JavaScript开发者来正确的认识JavaScript，用它来完成他应当完成的任务。 Ajax is a gateway drug for JavaScript. [参考资料] 1、Learn To love Javascript http://vanderburg.org/Writing/LearningToLoveJavaScript/ 2、NewtonScript http://en.wikipedia.org/wiki/NewtonScript 3、Java In Best Top Ten of 1995 http://www.time.com/time/magazine/article/0，9171，983903，00.html 编辑特别推荐: 指点一下：到底该不该去考JAVA认证? Java认证权威问答精华集 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com