目　录

图灵社区会员 StinkBC(StinkBC@gmail.com) 专享 尊重版权

图灵社区会员 StinkBC(StinkBC@gmail.com) 专享 尊重版权

图灵社区会员 StinkBC(StinkBC@gmail.com) 专享 尊重版权

图灵社区会员 StinkBC(StinkBC@gmail.com) 专享 尊重版权

图灵社区会员 StinkBC(StinkBC@gmail.com) 专享 尊重版权

图灵社区会员 StinkBC(StinkBC@gmail.com) 专享 尊重版权

图灵社区会员 StinkBC(StinkBC@gmail.com) 专享 尊重版权

第 1 章

JavaScript 简介

本章内容

- JavaScript 历史回顾
- JavaScript 是什么
- JavaScript 与 ECMAScript 的关系
- JavaScript 的不同版本

JavaScript 诞生于 1995 年。当时，它的主要目的是处理以前由服务器端语言（如 Perl）负责的一些输入验证操作。在 JavaScript 问世之前，必须把表单数据发送到服务器端才能确定用户是否没有填写某个必填域，是否输入了无效的值。Netscape Navigator 希望通过 JavaScript 来解决这个问题。在人们普遍使用电话拨号上网的年代，能够在客户端完成一些基本的验证任务绝对是令人兴奋的。毕竟，拨号上网的速度之慢，导致了与服务器的每一次数据交换事实上都成了对人们耐心的一次考验。

自此以后，JavaScript 逐渐成为市面上常见浏览器必备的一项特色功能。如今，JavaScript 的用途早已不再局限于简单的数据验证，而是具备了与浏览器窗口及其内容等几乎所有方面交互的能力。今天的 JavaScript 已经成为一门功能全面的编程语言，能够处理复杂的计算和交互，拥有了闭包、匿名（lamda，拉姆达）函数，甚至元编程等特性。作为 Web 的一个重要组成部分，JavaScript 的重要性是不言而喻的，就连手机浏览器，甚至那些专为残障人士设计的浏览器等非常规浏览器都支持它。当然，微软的例子更为典型。虽然有自己的客户端脚本语言 VBScript，但微软仍然在 Internet Explorer 的早期版本中加入了自己的 JavaScript 实现[①]。

JavaScript 从一个简单的输入验证器发展成为一门强大的编程语言，完全出乎人们的意料。应该说，它既是一门非常简单的语言，又是一门非常复杂的语言。说它简单，是因为学会使用它只需片刻功夫；而说它复杂，是因为要真正掌握它则需要数年时间。要想全面理解和掌握 JavaScript，关键在于弄清楚它的本质、历史和局限性。

1.1 JavaScript 简史

在 Web 日益流行的同时，人们对客户端脚本语言的需求也越来越强烈。那个时候，绝大多数因特网用户都使用速度仅为 28.8kbit/s 的“猫”（调制解调器）上网，但网页的大小和复杂性却不断增加。为完成简单的表单验证而频繁地与服务器交换数据只会加重用户的负担。想象一下：用户填写完一个表单，单击“提交”按钮，然后等待 30 秒钟，最终服务器返回消息说有一个必填字段没有

① 对 IE 而言，当我们提到 JavaScript 时，实际上就是指 IE 对 JavaScript（ECMAScript）的实现——JScript。最早的 JScript 基于 Netscape JavaScript 1.0 开发，于 1996 年 8 月随同 Internet Explorer 3.0 发布。

图灵社区会员 StinkBC(StinkBC@gmail.com) 专享 尊重版权

填好……当时走在技术革新最前沿的 Netscape 公司，决定着手开发一种客户端语言，用来处理这种简单的验证。

当时就职于 Netscape 公司的布兰登・艾奇（Brendan Eich），开始着手为计划于 1995 年 2 月发布的 Netscape Navigator 2 开发一种名为 LiveScript 的脚本语言——该语言将同时在浏览器和服务器中使用（它在服务器上的名字叫 LiveWire）。为了赶在发布日期前完成 LiveScript 的开发，Netscape 与 Sun 公司建立了一个开发联盟。在 Netscape Navigator 2 正式发布前夕，Netscape 为了搭上媒体热炒 Java 的顺风车，临时把 LiveScript 改名为 JavaScript。

由于 JavaScript 1.0 获得了巨大成功，Netscape 随即在 Netscape Navigator 3 中又发布了 JavaScript 1.1。Web 虽然羽翼未丰，但用户关注度却屡创新高。在这样的背景下，Netscape 把自己定位为市场领袖型公司。与此同时，微软决定向与 Navigator 竞争的自家产品 Internet Explorer 浏览器投入更多资源。Netscape Navigator 3 发布后不久，微软就在其 Internet Explorer 3 中加入了名为 JScript 的 JavaScript 实现（命名为 JScript 是为了避开与 Netscape 有关的授权问题）。以现在的眼光来看，微软 1996 年 8 月为进入 Web 浏览器领域而实施的这个重大举措，是导致 Netscape 日后蒙羞的一个标志性事件。然而，这个重大举措同时也标志着 JavaScript 作为一门语言，其开发向前迈进了一大步。

微软推出其 JavaScript 实现意味着有了两个不同的 JavaScript 版本：Netscape Navigator 中的 JavaScript、Internet Explorer 中的 JScript。与 C 及其他编程语言不同，当时还没有标准规定 JavaScript 的语法和特性，两个不同版本并存的局面已经完全暴露了这个问题。随着业界担心的日益加剧，JavaScript 的标准化问题被提上了议事日程。

1997 年，以 JavaScript 1.1 为蓝本的建议被提交给了欧洲计算机制造商协会（ECMA，European Computer Manufacturers Association）。该协会指定 39 号技术委员会（TC39，Technical Committee #39）负责“标准化一种通用、跨平台、供应商中立的脚本语言的语法和语义”（http://www.ecma international.org/memento/TC39.htm）。TC39 由来自 Netscape、Sun、微软、Borland 及其他关注脚本语言发展的公司的程序员组成，他们经过数月的努力完成了 ECMA-262——定义一种名为 ECMAScript（发音为“ek-ma-script”）的新脚本语言的标准。

第二年，ISO/IEC（International Organization for Standardization and International Electrotechnical Commission，国标标准化组织和国际电工委员会）也采用了 ECMAScript 作为标准（即 ISO/IEC-16262）。自此以后，浏览器开发商就开始致力于将 ECMAScript 作为各自 JavaScript 实现的基础，也在不同程度上取得了成功。

1.2　JavaScript 实现

虽然 JavaScript 和 ECMAScript 通常都被人们用来表达相同的含义，但 JavaScript 的含义却比 ECMA-262 中规定的要多得多。没错，一个完整的 JavaScript 实现应该由下列三个不同的部分组成（见图 1-1）。

- 核心（ECMAScript）
- 文档对象模型（DOM）
- 浏览器对象模型（BOM）

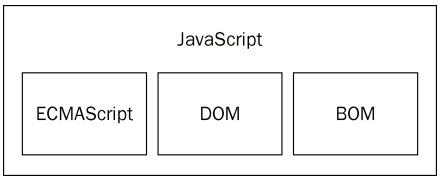

图　1-1

图灵社区会员 StinkBC(StinkBC@gmail.com) 专享 尊重版权