第 1 章

什么是 JavaScript

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

1995 年，JavaScript 问世。当时，它的主要用途是代替 Perl 等服务器端语言处理输入验证。在此之前，要验证某个必填字段是否已填写，或者某个输入的值是否有效，需要与服务器的一次往返通信。网景公司希望通过在其 Navigator 浏览器中加入 JavaScript 来改变这个局面。在那个普遍通过电话拨号上网的年代，由客户端处理某些基本的验证是让人兴奋的新功能。缓慢的网速让页面每次刷新都考验着人们的耐心。

从那时起，JavaScript 逐渐成为市面上所有主流浏览器的标配。如今，JavaScript 的应用也不再局限于数据验证，而是渗透到浏览器窗口及其内容的方方面面。JavaScript 已被公认为主流的编程语言，能够实现复杂的计算与交互，包括闭包、匿名（lambda）函数，甚至元编程等特性。不仅是桌面浏览器，手机浏览器和屏幕阅读器也支持 JavaScript，其重要性可见一斑。就连拥有自家客户端脚本语言 VBScript 的微软公司，也在其 Internet Explorer（以下简称 IE）浏览器最初的版本中包含了自己的 JavaScript 实现。

从简单的输入验证脚本到强大的编程语言，JavaScript 的崛起没有任何人预测到。它很简单，学会用只要几分钟；它又很复杂，掌握它要很多年。要真正学好用好 JavaScript，理解其本质、历史及局限性是非常重要的。

1.1 简短的历史回顾

随着 Web 日益流行，对客户端脚本语言的需求也越来越强烈。当时，大多数用户使用 28.8kbit/s 的调制解调器上网，但网页变得越来越大、越来越复杂。为验证简单的表单而需要大量与服务器的往返通信成为用户的痛点。想象一下，你填写完表单，单击“提交”按钮，等 30 秒处理，然后看到一条消息，告诉你有一个必填字段没填。网景在当时是引领技术革新的公司，它将开发一个客户端脚本语言来处理这种简单的数据验证提上了日程。

1995 年，网景公司一位名叫 Brendan Eich 的工程师，开始为即将发布的 Netscape Navigator 2 开发一个叫 Mocha（后来改名为 LiveScript）的脚本语言。当时的计划是在客户端和服务器端都使用它，它在服务器端叫 LiveWire。

为了赶上发布时间，网景与 Sun 公司结为开发联盟，共同完成 LiveScript 的开发。就在 Netscape Navigator 2 正式发布前，网景把 LiveScript 改名为 JavaScript，以便搭上媒体当时热烈炒作 Java 的顺风车。

由于 JavaScript 1.0 很成功，网景又在 Netscape Navigator 3 中发布了 1.1 版本。尚未成熟的 Web 的受欢迎程度达到了历史新高，而网景则稳居市场领导者的位置。这时候，微软决定向 IE 投入更多资源。就在 Netscape Navigator 3 发布后不久，微软发布了 IE3，其中包含自己名为 JScript（叫这个名字是为了避免与网景发生许可纠纷）的 JavaScript 实现。1996 年 8 月，微软重磅进入 Web 浏览器领域，这是网景永远的痛，但它代表 JavaScript 作为一门语言向前迈进了一大步。

微软的 JavaScript 实现意味着出现了两个版本的 JavaScript：Netscape Navigator 中的 JavaScript，以及 IE 中的 JScript。与 C 语言以及很多其他编程语言不同，JavaScript 还没有规范其语法或特性的标准，两个版本并存让这个问题更加突出了。随着业界担忧日甚，JavaScript 终于踏上了标准化的征程。

1997 年，JavaScript 1.1 作为提案被提交给欧洲计算机制造商协会（Ecma）。第 39 技术委员会（TC39）承担了“标准化一门通用、跨平台、厂商中立的脚本语言的语法和语义”的任务（参见 TC39-ECMAScript）。TC39 委员会由来自网景、Sun、微软、Borland、Nombas 和其他对这门脚本语言有兴趣的公司的工程师组成。他们花了数月时间打造出 ECMA-262，也就是 ECMAScript（发音为“ek-ma-script”）这个新的脚本语言标准。

1998 年，国际标准化组织（ISO）和国际电工委员会（IEC）也将 ECMAScript 采纳为标准（ISO/IEC-16262）。自此以后，各家浏览器均以 ECMAScript 作为自己 JavaScript 实现的依据，虽然具体实现各有不同。

1.2　JavaScript 实现

虽然 JavaScript 和 ECMAScript 基本上是同义词，但 JavaScript 远远不限于 ECMA-262 所定义的那样。没错，完整的 JavaScript 实现包含以下几个部分（见图 1-1）：

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

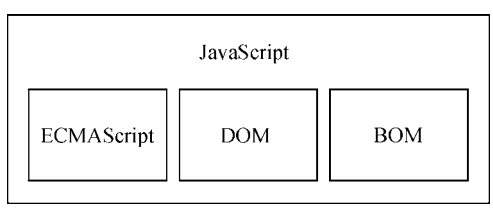

图　1-1

1.2.1　ECMAScript

ECMAScript，即 ECMA-262 定义的语言，并不局限于 Web 浏览器。事实上，这门语言没有输入和输出之类的方法。ECMA-262 将这门语言作为一个基准来定义，以便在它之上再构建更稳健的脚本语言。Web 浏览器只是 ECMAScript 实现可能存在的一种**宿主环境**（host environment）。宿主环境提供 ECMAScript 的基准实现和与环境自身交互必需的扩展。扩展（比如 DOM）使用 ECMAScript 核心类型和语法，提供特定于环境的额外功能。其他宿主环境还有服务器端 JavaScript 平台 Node.js 和即将被淘汰的 Adobe Flash。

如果不涉及浏览器的话，ECMA-262 到底定义了什么？在基本的层面，它描述这门语言的如下部分：

- 语法
- 类型
- 语句
- 关键字
- 保留字
- 操作符
- 全局对象

ECMAScript 只是对实现这个规范描述的所有方面的一门语言的称呼。JavaScript 实现了 ECMAScript，而 Adobe ActionScript 同样也实现了 ECMAScript。

1. ECMAScript 版本

ECMAScript 不同的版本以“edition”表示（也就是描述特定实现的 ECMA-262 的版本）。ECMA-262 最近的版本是第 10 版，发布于 2019 年 6 月。ECMA-262 的第 1 版本质上跟网景的 JavaScript 1.1 相同，只不过删除了所有浏览器特定的代码，外加少量细微的修改。ECMA-262 要求支持 Unicode 标准（以支持多语言），而且对象要与平台无关（Netscape JavaScript 1.1 的对象不是这样，比如它的 `Date` 对象就依赖平台）。这也是 JavaScript 1.1 和 JavaScript 1.2 不符合 ECMA-262 第 1 版要求的原因。

ECMA-262 第 2 版只是做了一些编校工作，主要是为了更新之后严格符合 ISO/IEC-16262 的要求，并没有增减或改变任何特性。ECMAScript 实现通常不使用第 2 版来衡量符合性（conformance）。

ECMA-262 第 3 版第一次真正对这个标准进行更新，更新了字符串处理、错误定义和数值输出。此外还增加了对正则表达式、新的控制语句、`try/catch` 异常处理的支持，以及为了更好地让标准国际化所做的少量修改。对很多人来说，这标志着 ECMAScript 作为一门真正的编程语言的时代终于到来了。

ECMA-262 第 4 版是对这门语言的一次彻底修订。作为对 JavaScript 在 Web 上日益成功的回应，开发者开始修订 ECMAScript 以满足全球 Web 开发日益增长的需求。为此，Ecma T39 再次被召集起来，以决定这门语言的未来。结果，他们制定的规范几乎在第 3 版基础上完全定义了一门新语言。第 4 版包括强类型变量、新语句和数据结构、真正的类和经典的继承，以及操作数据的新手段。

与此同时，TC39 委员会的一个子委员会也提出了另外一份提案，叫作“ECMAScript 3.1”，只对这门语言进行了较少的改进。这个子委员会的人认为第 4 版对这门语言来说跳跃太大了。因此，他们提出了一个改动较小的提案，只要在现有 JavaScript 引擎基础上做一些增改就可以实现。最终，ES3.1 子委员会赢得了 TC39 委员会的支持，ECMA-262 第 4 版在正式发布之前被放弃。

ECMAScript 3.1 变成了 ECMA-262 的第 5 版，于 2009 年 12 月 3 日正式发布。第 5 版致力于厘清第 3 版存在的歧义，也增加了新功能。新功能包括原生的解析和序列化 JSON 数据的 `JSON` 对象、方便继承和高级属性定义的方法，以及新的增强 ECMAScript 引擎解释和执行代码能力的严格模式。第 5 版在 2011 年 6 月发布了一个维护性修订版，这个修订版只更正了规范中的错误，并未增加任何新的语言或库特性。

ECMA-262 第 6 版，俗称 ES6、ES2015 或 ES Harmony（和谐版），于 2015 年 6 月发布。这一版包含了大概这个规范有史以来最重要的一批增强特性。ES6 正式支持了类、模块、迭代器、生成器、箭头函数、期约、反射、代理和众多新的数据类型。

ECMA-262 第 7 版，也称为 ES7 或 ES2016，于 2016 年 6 月发布。这次修订只包含少量语法层面的增强，如 `Array.prototype.includes` 和指数操作符。

ECMA-262 第 8 版，也称为 ES8、ES2017，完成于 2017 年 6 月。这一版主要增加了异步函数（async/await）、SharedArrayBuffer 及 Atomics API，以及 Object.values()/Object.entries()/Object.getOwnPropertyDescriptors()和字符串填充方法，另外明确支持对象字面量最后的逗号。

ECMA-262 第 9 版，也称为 ES9、ES2018，发布于 2018 年 6 月。这次修订包括异步迭代、剩余和扩展属性、一组新的正则表达式特性、Promise finally()，以及模板字面量修订。

ECMA-262 第 10 版，也称为 ES10、ES2019，发布于 2019 年 6 月。这次修订增加了 Array.prototype.flat()/flatMap()、String.prototype.trimStart()/trimEnd()、Object.fromEntries()方法，以及 Symbol.prototype.description 属性，明确定义了 Function.prototype.toString()的返回值并固定了 Array.prototype.sort()的顺序。另外，这次修订解决了与 JSON 字符串兼容的问题，并定义了 catch 子句的可选绑定。

2. ECMAScript 符合性是什么意思

ECMA-262 阐述了什么是 ECMAScript 符合性。要成为 ECMAScript 实现，必须满足下列条件：

- 支持 ECMA-262 中描述的所有“类型、值、对象、属性、函数，以及程序语法与语义”；
- 支持 Unicode 字符标准。

此外，符合性实现还可以满足下列要求。

- 增加 ECMA-262 中未提及的“额外的类型、值、对象、属性和函数”。ECMA-262 所说的这些额外内容主要指规范中未给出的新对象或对象的新属性。
- 支持 ECMA-262 中没有定义的“程序和正则表达式语法”（意思是允许修改和扩展内置的正则表达式特性）。

以上条件为实现开发者基于 ECMAScript 开发语言提供了极大的权限和灵活度，也是其广受欢迎的原因之一。

3. 浏览器对 ECMAScript 的支持

1996 年，Netscape Navigator 3 发布时包含了 JavaScript 1.1。JavaScript 1.1 规范随后被提交给 Ecma，作为对新的 ECMA-262 标准的建议。随着 JavaScript 迅速走红，网景非常愿意开发 1.2 版。可是有个问题：Ecma 尚未接受网景的建议。

Netscape Navigator 3 发布后不久，微软推出了 IE3。IE 的这个版本包含了 JScript 1.0，本意是提供与 JavaScript 1.1 相同的功能。不过，由于缺少很多文档，而且还有不少重复性功能，JScript 1.0 远远没有 JavaScript 1.1 那么强大。

JScript 的再次更新出现在 IE4 中的 JScript 3.0（2.0 版是在 Microsoft Internet Information Server 3.0 中发布的，但从未包含在浏览器中）。微软发新闻稿称 JScript 3.0 是世界上第一门真正兼容 Ecma 标准的脚本语言。当时 ECMA-262 还没制定完成，因此 JScript 3.0 遵受了与 JavaScript 1.2 同样的命运，它同样没有遵守最终的 ECMAScript 标准。

网景又在 Netscape Navigator 4.06 中将其 JavaScript 版本升级到 1.3，因此做到了与 ECMA-262 第 1 版完全兼容。JavaScript 1.3 增加了对 Unicode 标准的支持，并做到了所有对象都与平台无关，同时保留了 JavaScript 1.2 所有的特性。

后来，当网景以 Mozilla 项目的名义向公众发布其源代码时，人们都期待 Netscape Navigator 5 中会包含 JavaScript 1.4。可是，一个完全重新设计网景代码的激进决定导致了人们的希望落空。JavaScript 1.4 只在 Netscape Enterprise Server 中作为服务器端语言发布了，从来就没有进入浏览器。

到了 2008 年，五大浏览器（IE、Firefox、Safari、Chrome 和 Opera）全部兼容 ECMA-262 第 3 版。

IE8 率先实现 ECMA-262 第 5 版，并在 IE9 中完整支持。Firefox 4 很快也做到了。下表列出了主要的浏览器版本对 ECMAScript 的支持情况。

浏 览 器	ECMAScript 符合性
Netscape Navigator 2	—
Netscape Navigator 3	—
Netscape Navigator 4~4.05	—
Netscape Navigator 4.06~4.79	第 1 版
Netscape 6+（Mozilla 0.6.0+）	第 3 版
IE3	—
IE4	—
IE5	第 1 版
IE5.5~8	第 3 版
IE9	第 5 版（部分）
IE10~11	第 5 版
Edge 12+	第 6 版
Opera 6~7.1	第 2 版
Opera 7.2+	第 3 版
Opera 15~28	第 5 版
Opera 29~35	第 6 版（部分）
Opera 36+	第 6 版
Safari 1~2.0.*x*	第 3 版（部分）
Safari 3.1~5.1	第 5 版（部分）
Safari 6~8	第 5 版
Safari 9+	第 6 版
iOS Safari 3.2~5.1	第 5 版（部分）
iOS Safari 6~8.4	第 5 版
iOS Safari 9.2+	第 6 版
Chrome 1~3	第 3 版
Chrome 4~22	第 5 版（部分）
Chrome 23+	第 5 版
Chrome 42~48	第 6 版（部分）
Chrome 49+	第 6 版
Firefox 1~2	第 3 版
Firefox 3.0.*x*~20	第 5 版（部分）
Firefox 21~44	第 5 版
Firefox 45+	第 6 版