* 为什们会有Javascript？

在上世纪90年代，因为绝大多数因特网用户的上网速度仅为28.8kbit/s的“猫”，但网页的大小和复杂性却不断的增加。为完成简单的表单验证而不断的与服务器交换数据。当用户填完一个表单，单击“提交”按钮，然后等待30秒，服务器最终返回消息说有一个字段没有填好……

* Javascrpit创始人

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2个不同的JavaScript版本：Netscape Navigator中的JavaScript、Internet Explorer中的Jscript和ScriptEase中的CEnvi。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，3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。

* ECMAScript的诞生

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 ComputerManufacturers 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.ecmainternational.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 andInternational ElectrotechnicalCommission，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

。

* JavaScript组成部分

　　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。

* 1. 核心（ECMAScript）
  2. 文档对象模型（DOM）
  3. 浏览器对象模型（BOM）
* ECMAScript组成部分

　　由ECMA-262定义的ECMAScript与Web浏览器没有依赖关系。实际上，这门语言本身并不包含输入和输出定义。ECMA-262定义的只是这门语言的基础，而在此基础之上可以构建更完善的脚本语言。我们常见的Web浏览器只是ECMAScript实现可能的宿主环境之一。宿主环境不仅提供基本的ECMAScript实现，同时也会提供该语言的扩展，以便语言与环境之间对接交互。而这些扩展——如DOM，则利用ECMAScript的核心类型和语法提供更多更具体的功能，以便实现针对环境的操作。前面介绍过的Node以及众所周知的AdobeFlash也都是宿主环境。

　　既然ECMA-262标准没有参照Web浏览器，那它都规定了些什么内容呢？大致说来，它规定了这门语言的下列组成部分：

* 1. 语法
  2. 类型
  3. 语句
  4. 关键字
  5. 保留字
  6. 操作符
  7. 对象

　　ECMAScript就是对实现该标准规定的各个方面内容的语言的描述。JavaScript实现了ECMAScript，AdobeActionScript同样也实现了ECMAScript。

* ECMAScript的版本

　　ECMAScript的不同版本又称为版次，以第x版表示（意即描述特定实现的ECMA-262规范的第x个版本）。ECMA-262的最近一版是第5版，发布于2009年。而ECMA-262的第1版本质上与Netscape的JavaScript1.1相同——只不过删除了所有针对浏览器的代码并作了一些较小的改动：ECMA-262要求支持Unicode标准（从而支持多语言开发），而且对象也变成了平台无关的（NetscapeJavaScript 1.1的对象在不同平台中的实现不一样，例如Date对象）。这也是JavaScript1.1和1.2与ECMA-262第1版不一致的主要原因。

　　ECMA-262第2版主要是编辑加工的结果。这一版中内容的更新是为了与ISO/IEC-16262保持严格一致，没有作任何新增、修改或删节处理。因此，一般不使用第2版来衡量ECMAScript实现的兼容性。

　　ECMA-262第3版才是对该标准第一次真正的修改。修改的内容涉及字符串处理、错误定义和数值输出。这一版还新增了对正则表达式、新控制语句、try-catch异常处理的支持，并围绕标准的国际化做出了一些小的修改。从各方面综合来看，第3版标志着ECMAScript成为了一门真正的编程语言。

　　ECMA-262第4版对这门语言进行了一次全面的检核修订。由于JavaScript在Web上日益流行，开发人员纷纷建议修订ECMAScript，以使其能够满足不断增长的Web开发需求。作为回应，ECMATC39重新召集相关人员共同谋划这门语言的未来。结果，出台后的标准几乎在第3版基础上完全定义了一门新语言。第4版不仅包含了强类型变量、新语句和新数据结构、真正的类和经典继承，还定义了与数据交互的新方式。

　　与此同时，TC39下属的一个小组也提出了一个名为ECMAScript3.1的替代性建议，该建议只对这门语言进行了较少的改进。这个小组认为第4版给这门语言带来的跨越太大了。因此，该小组建议对这门语言进行小幅修订，能够在现有JavaScript引擎基础上实现。最终，ES3.1附属委员会获得的支持超过了TC39，ECMAS-262第4版在正式发布前被放弃。

　　ECMAScript3.1成为ECMA-262第5版，并于2009年12月3日正式发布。第5版力求澄清第3版中已知的歧义并增添了新的功能。新功能包括原生JSON对象（用于解析和序列化JSON数据）、继承的方法和高级属性定义，另外还包含一种严格模式，对ECMAScript引擎解释和执行代码进行了补充说明。

* 什么是ECMAScript兼容

ECMA-262给出了ECMAScript兼容的定义。要想成为ECMAScript的实现，则该实现必须做到：

* 1. 支持ECMA-262描述的所有“类型、值、对象、属性、函数以及程序句法和语义”（ECMA-262第1页）；
  2. 支持Unicode字符标准。
  3. 此外，兼容的实现还可以进行下列扩展。
  4. 添加ECMA-262没有描述的“更多类型、值、对象、属性和函数”。ECMA-262所说的这些新增特性，主要是指该标准中没有规定的新对象和对象的新属性。
  5. 支持ECMA-262没有定义的“程序和正则表达式语法”。（也就是说，可以修改和扩展内置的正则表达式语法。）

　　上述要求为兼容实现的开发人员基于ECMAScript开发一门新语言提供了广阔的空间和极大的灵活性，这也从另一个侧面说明了ECMAScript受开发人员欢迎的原因。

* Web浏览器对ECMAScript的支持

　　1996年，Netscape Navigator 3捆绑发布了JavaScript 1.1。而相同的JavaScript1.1设计规范随后作为对新标准（ECMA-262）的建议被提交给Ecma。伴随着JavaScript的迅速走红，Netscape豪情满怀地着手开发JavaScript1.2。然而，问题是Ecma当时还没有接受Netscape的建议。

　　Netscape Navigator 3发布后不久，微软也推出了Internet Explorer3。微软在IE的这一版中捆绑了JScript 1.0，很多人都认为JScript 1.0与JavaScript1.1应该是一样的。但是，由于没有文档依据，加之不适当的特性模仿，JScript 1.0还是很难与JavaScript1.1相提并论。

　　1997年，内置JavaScript 1.2的Netscape Navigator4发布；而到这一年年底，ECMA-262第1版也被接受并实现了标准化。结果，虽然ECMAScript被认为是基于JavaScript1.1制定的，但JavaScript 1.2与ECMAScript的第1版并不兼容。

JScript的升级版是Internet Explorer 4中内置的JScript 3.0（随同微软IIS3.0发布的JScript 2.0从来也没有移植到浏览器中）。微软通过媒体大肆宣传JScript3.0是世界上第一个ECMA兼容的脚本语言，但当时的ECMA-262尚未定稿。于是，JScript 3.0与JavaScript1.2都遭遇了相同的尴尬局面——谁都没有按照最终的ECMAScript标准来实现。

　　Netscape决定更新其JavaScript实现，即在Netscape Navigator 4.06中发布JavaScript1.3，从而做到了与ECMA-262的第一个版本完全兼容。在JavaScript1.3中，Netscape增加了对Unicode标准的支持，并在保留JavaScript1.2新增特性的同时实现了所有对象的平台中立化。

　　在Netscape以Mozilla项目的名义开放其源代码时，预期JavaScript 1.4将随同NetscapeNavigator 5一道发布。然而，一个激进的决定，彻底重新设计Netscape代码，打乱了原有计划。后来，JavaScript1.4只发布了针对Netscape Enterprise Server的服务器版，而没有内置于Web浏览器中。

　　到了2008年，五大主流Web浏览器（IE、Firefox、Safari、Chrome和Opera）全部做到了与ECMA-262兼容。IE8是第一个着手实现ECMA-262第5版的浏览器，并在IE9中提供了完整的支持。Firefox4也紧随其后做到兼容。下表列出了ECMAScript受主流Web浏览器支持的情况。

　　浏 览 器 ECMAScript兼容 浏 览 器 ECMAScript兼容性

　　Netscape Navigator 2 — Opera 6～7.1 第2版

　　Netscape Navigator 3 — Opera 7.2+ 第3版

　　Netscape Navigator 4～4.05 — Safari 1～2.0.x 第3版\*

　　Netscape Navigator 4.06～4.79 第1版 Safari 3.x 第3版

　　Netscape 6+（Mozilla 0.6.0+） 第3版 Safari 4.x～5.x 第5版\*

　　IE3 — Chrome 1+ 第3版

　　IE4 — Firefox 1～2 第3版

　　IE5 第1版 Firefox 3.0.x 第3版

　　IE5.5～IE7 第3版 Firefox 3.5～3.6 第5版\*

　　IE8 第5版\* Firefox 4.0 + 第5版

　　IE9+ 第5版

　　\* 不完全兼容的实现

* 文档对象模型（DOM）

　　文档对象模型（DOM，Document ObjectModel）是针对XML但经过扩展用于HTML的应用程序编程接口（API，Application ProgrammingInterface）。DOM把整个页面映射为一个多层节点结构。HTML或XML页面中的每个组成部分都是某种类型的节点，这些节点又包含着不同类型的数据。

通过DOM创建的这个表示文档的树形图，开发人员获得了控制页面内容和结构的主动权。借助DOM提供的API，开发人员可以轻松自如地删除、添加、替换或修改任何节点。

　　1. 为什么要使用DOM

　　在Internet Explorer 4和Netscape Navigator 4分别支持的不同形式的DHTML（DynamicHTML）基础上，开发人员首次无需重新加载网页，就可以修改其外观和内容了。然而，DHTML在给Web技术发展带来巨大进步的同时，也带来了巨大的问题。由于Netscape和微软在开发DHTML方面各持己见，过去那个只编写一个HTML页面就能够在任何浏览器中运行的时代结束了。

对开发人员而言，如果想继续保持Web跨平台的天性，就必须额外多做一些工作。而人们真正担心的是，如果不对Netscapet和微软加以控制，Web开发领域就会出现技术上两强割据，浏览器互不兼容的局面。此时，负责制定Web通信标准的W3C（WorldWide Web Consortium，万维网联盟）开始着手规划DOM。

　　2. DOM级别

　　DOM1级（DOM Level 1）于1998年10月成为W3C的推荐标准。DOM1级由两个模块组成：DOM核心（DOMCore）和DOM HTML。其中，DOM核心规定的是如何映射基于XML的文档结构，以便简化对文档中任意部分的访问和操作。DOMHTML模块则在DOM核心的基础上加以扩展，添加了针对HTML的对象和方法。

　　请读者注意，DOM并不只是针对JavaScript的，很多别的语言也都实现了DOM。不过，在Web浏览器中，基于ECMAScript实现的DOM的确已经成为JavaScript这门语言的一个重要组成部分。

如果说DOM1级的目标主要是映射文档的结构，那么DOM2级的目标就要宽泛多了。DOM2级在原来DOM的基础上又扩充了（DHTML一直都支持的）鼠标和用户界面事件、范围、遍历（迭代DOM文档的方法）等细分模块，而且通过对象接口增加了对CSS（CascadingStyle Sheets，层叠样式表）的支持。DOM1级中的DOM核心模块也经过扩展开始支持XML命名空间。

DOM2级引入了下列新模块，也给出了众多新类型和新接口的定义：

1. DOM视图（DOM Views）--定义了跟踪不同文档（例如，应用CSS之前和之后的文档）视图的接口；
2. DOM事件（DOM Events）--定义了事件和事件处理的接口；
3. DOM样式（DOM Style）--定义了基于CSS为元素应用样式的接口；
4. DOM遍历和范围（DOM Traversal and Range）--定义了遍历和操作文档树的接口。

DOM3级则进一步扩展了DOM，引入了以统一方式加载和保存文档的方法--在DOM加载和保存（DOM Load and Save）模块中定义；新增了验证文档的方法--在DOM验证（DOM Validation）模块中定义。DOM3级也对DOM核心进行了扩展，开始支持XML 1.0规范，涉及XML Infoset、XPath和XML Base。

在阅读DOM标准的时候，读者可能会看到DOM0级（DOM Level 0）的字眼。实际上，DOM0级标准是不存在的；所谓DOM0级只是DOM历史坐标中的一个参照点而已。具体说来，DOM0级指的是Internet Explorer 4.0和Netscape Navigator 4.0最初支持的DHTML。

3. 其他DOM标准

除了DOM核心和DOM HTML接口之外，另外几种语言还发布了只针对自己的DOM标准。下面列出的语言都是基于XML的，每种语言的DOM标准都添加了与特定语言相关的新方法和新接口：

1. SVG（Scalable Vector Graphic，可伸缩矢量图）1.0；
2. MathML（Mathematical Markup Language，数学标记语言,用于书写书写公式或符号的置标语言）1.0；
3. SMIL（Synchronized Multimedia Integration Language，同步多媒体集成语言，操纵多媒体片断，对多媒体片断的有机的、智能的组合）。

还有一些语言也开发了自己的DOM实现，例如Mozilla的XUL（XML User Interface Language，XML用户界面语言）。但是，只有上面列出的几种语言是W3C的推荐标准。

4. Web浏览器对DOM的支持

在DOM标准出现了一段时间之后，Web浏览器才开始实现它。微软在IE5中首次尝试实现DOM，但直到IE5.5才算是真正支持DOM1级。在随后的IE6和IE7中，微软都没有引入新的DOM功能，而到了IE8才对以前DOM实现中的bug进行了修复。

Netscape直到Netscape 6（Mozilla 0.6.0）才开始支持DOM。在Netscape 7之后，Mozilla把开发重心转向了Firefox浏览器。Firefox 3完全支持DOM1级，几乎完全支持DOM2级，甚至还支持DOM3级的一部分。（Mozilla开发团队的目标是构建与标准100%兼容的浏览器，而他们的努力也得到了回报。）

目前，支持DOM已经成为浏览器开发商的首要目标，主流浏览器每次发布新版本都会改进对DOM的支持。在支持DOM方面，IE已经被另外三个主要浏览器远远地落在了后面--IE仍处于仅部分实现了DOM1级的状态。而Chrome 0.2+、Opera 9和Safari 3不仅完全支持DOM1级，还支持DOM2级的大部分。下表列出了主流浏览器对DOM标准的支持情况：



* 浏览器对象模型（BOM）

Internet Explorer 3和Netscape Navigator 3有一个共同的特色，那就是支持可以访问和操作浏览器窗口的浏览器对象模型（BOM，Browser Object Model）。开发人员使用BOM可以控制浏览器显示的页面以外的部分。而BOM真正与众不同的地方（也是经常会导致问题的地方），还是它作为JavaScript实现的一部分，至今仍没有相关的标准。

从根本上讲，BOM只处理浏览器窗口和框架；但人们习惯上也把所有针对浏览器的JavaScript扩展算作BOM的一部分。下面就是一些这样的扩展：

1. 弹出新浏览器窗口的功能；
2. 移动、缩放和关闭浏览器窗口的功能；
3. 提供浏览器详细信息的navigator对象；
4. 提供浏览器所加载页面的详细信息的location对象；
5. 提供用户显示器分辨率详细信息的screen对象；
6. 对cookies的支持；
7. 像XMLHttpRequest和IE的ActiveXObject这样的自定义对象。

由于没有BOM标准可以遵循，因此每个浏览器都有自己的实现。虽然也存在一些事实标准，例如要有window对象和navigator对象等，但每个浏览器都会为这两个对象乃至其他对象定义自己的属性和方法。

* JavaScript版本

作为Netscape“继承人”的Mozilla公司，是目前唯一还在沿用最初的JavaScript版本编号序列的浏览器开发商。在Netscape将源代码提交给开源的Mozilla项目的时候，JavaScript在浏览器中的最后一个版本号是1.3。（如前所述，1.4版是只针对服务器的实现。）后来，随着Mozilla基金会继续开发JavaScript，添加新的特性、关键字和语法，JavaScript的版本号继续递增。下表列出了Netscape/Mozilla浏览器中JavaScript版本号的递增过程：  
浏 览 器 JavaScript版本 浏览器 JavaScript版本  
Netscape Navigator 2 1.0 Firefox 1.5 1.6  
Netscape Navigator 3 1.1 Firefox 2 1.7  
Netscape Navigator 4 1.2 Firefox 3 1.8  
Netscape Navigator 4.06 1.3 Firefox 3.5 1.8.1  
Netscape 6+（Mozilla 0.6.0+） 1.5 Firefox 3.6 1.8.2  
Firefox 1 1.5  
实际上，上表中的编号方案源自Firefox 4将内置JavaScript 2.0这一共识。因此，2.0版之前每个递增的版本号，表示的是相应实现与JavaScript 2.0开发目标还有多大的距离。虽然原计划是这样，但JavaScript的这种发展速度让这个计划成为不再可行。目前，JavaScript 2.0还没有目标实现。  
请注意，只有Netscape/Mozilla浏览器才遵循这种编号模式。例如，IE的JScript就采用了另一种版本命名方案。换句话说，JScript的版本号与上表中JavaScript的版本号之间不存在任何对应关系。而且，大多数浏览器在提及对JavaScript的支持情况时，一般都以ECMAScript兼容性和对DOM的支持情况为准。

* 小结

JavaScript是一种专为与网页交互而设计的脚本语言，由下列三个不同的部分组成：

ECMAScript，由ECMA-262定义，提供核心语言功能；  
文档对象模型（DOM），提供访问和操作网页内容的方法和接口；  
浏览器对象模型（BOM），提供与浏览器交互的方法和接口。

JavaScript的这三个组成部分，在当前五个主要浏览器（IE、Firefox、Chrome、Safari和Opera）中都得到了不同程度的支持。其中，所有浏览器对ECMAScript第3版的支持大体上都还不错，而对ECMAScript 5的支持程度越来越高，但对DOM的支持则彼此相差比较多。对HTML5已经正式纳入标准的BOM来说，尽管各浏览器都实现了某些众所周知的共同特性，但其他特性还是会因浏览器而异。