

第1章 JavaScript概述

JavaScript，常常被称之为Web语言。这么称呼不是没有原因的，因为最早JavaScript就是被用来制作网页特效和表单验证的。可以这么说，JavaScript从一开始就是为我们Web所服务的。同时，JavaScript有着学习简单，容易上手的特点。比起其他像Java，C#这种传统编程语言，JavaScript可算是相当容易学习的。甚至不需要搭载什么复杂的运行环境，只需要安装一个浏览器，再用一个记事本就可以开始写JavaScript代码了。

但是，JavaScript往往又是一门棘手的语言。因为它的简单，为它本身带来了诸多的问题。还有各种各样的版本，令人疑惑的名称，常常也是让初学者摸不清头脑。甚至在JavaScript里所存在的一些特殊现象，即使是经验丰富的JavaScript开发人员，如果没有认真学习过的话也无法真正理解。要解释清楚JavaScript为什么是一门棘手的语言，我们首先不得不从它的发展史讲起。

本章中我们将学习如下的内容：

- JavaScript发展史
- JavaScript版本介绍
- 搭建JavaScript开发环境
- 书写第一个JavaScript程序
- NPM介绍

JavaScript发展史

这里，我想从4个阶段来回顾JavaScript的发展史，分别是

第一阶段：JavaScript起源

第二阶段：第一次浏览器大战

第三阶段：第二次浏览器大战

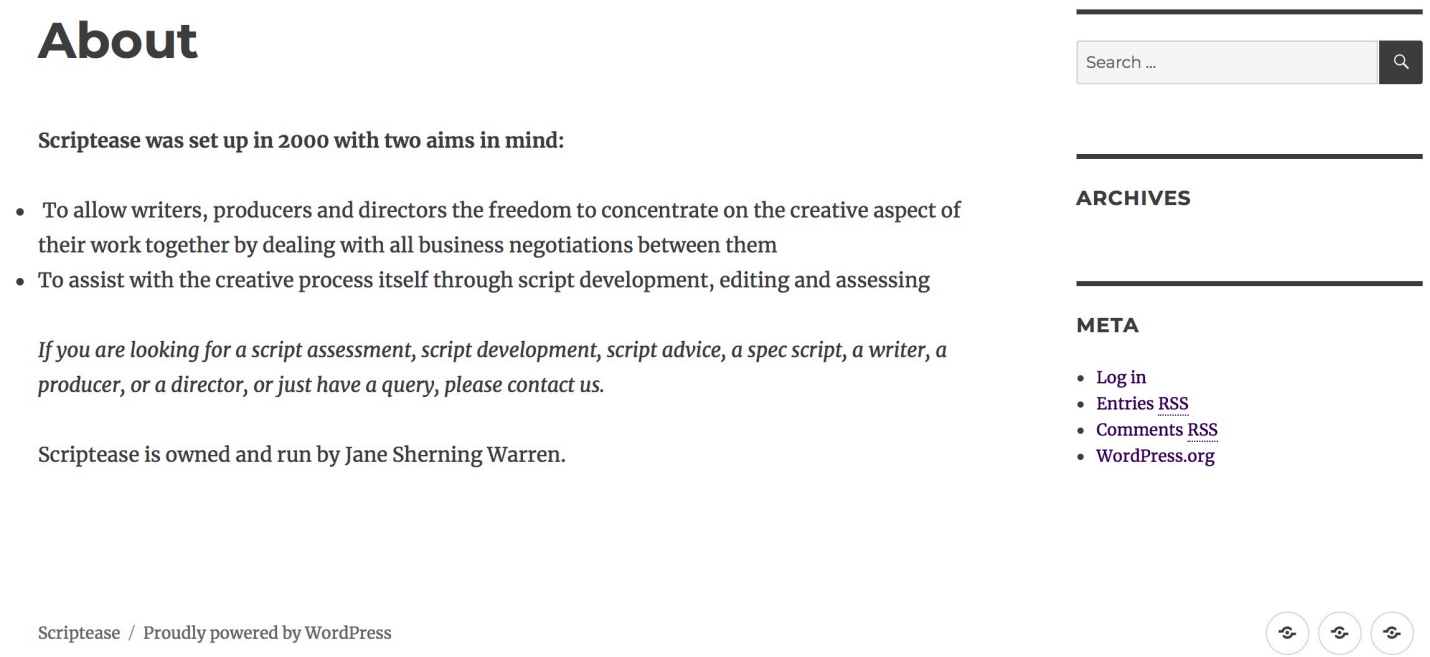
第四阶段：JavaScript蓬勃发展

第一阶段：JavaScript起源

JavaScript首次出现是在1995年，当年JavaScript的诞生绝不是偶然的。在1992年，一个叫做Nombas的公司开发了一门叫做"C减减"(C minus minus, Cmm)的语言，后来改名为ScriptEase。ScriptEase最初的设计是将一种微型脚本语言与一个叫做Espresso Page的工具配合，使脚本能够在浏览器中运行，因此ScriptEase成为了第一个客户端脚本语言。到目前为止，我们仍然可以在该语言的官方网站

<http://scriptease.info>

上面找到关于这门语言的介绍，如下图：



(图为ScriptEase官网截图)

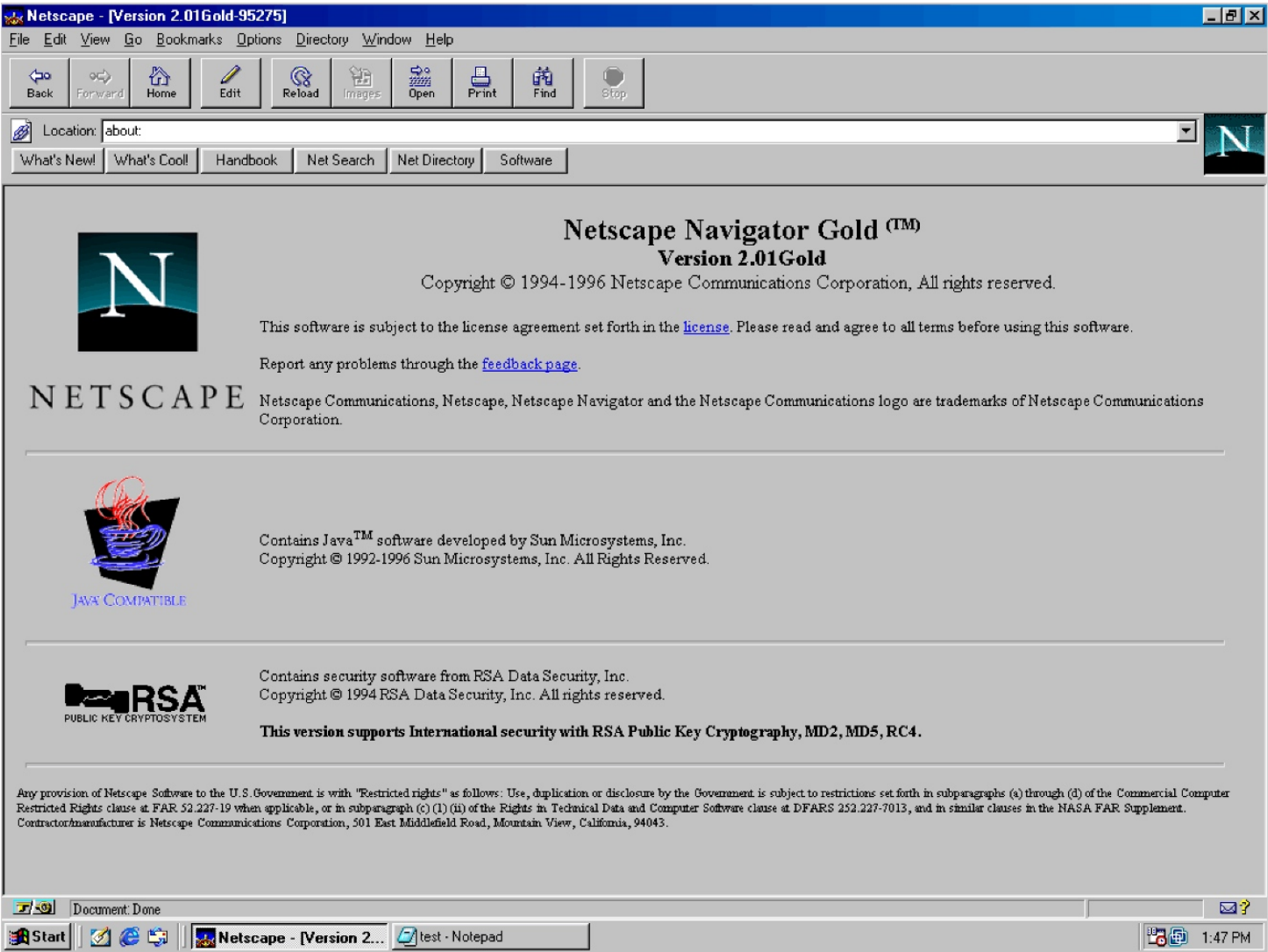
在ScriptEase出现以后，有一家叫做网景的公司，该公司也是世界上第一个将浏览器商用的公司。它们也想独立开发一种与ScriptEase相似的客户端脚本语言。于是，该公司的一个名叫布兰登·艾奇的人接受了这个任务，开始开发这门脚本语言。该语言的语法借鉴了Java，Perl，C，Self，Scheme等编程语言。布兰登·艾奇仅仅花了10天时间，就写出来了这门语言。起初，这个语言的目标是为了非专业的开发人员(如网站设计者)，提供一个方便的工具。因为大多数网站设计者都没有任何的编程背景，所以这个语言应该尽可能简单，易学，最终一个弱类型的动态解释语言LiveWire就此诞生。LiveWire没过多久就改名为了LiveScript了，直到现在，在一些古老的Web页面中还能看到这个名字。10天就写出来的东西，里面必然是存在很多问题的。这里面所存在的问题，有些遗留到至今都没有解决。有些问题现在反而还成了这门语言的一个特点。

在LiveScript诞生之前，Java applet曾经被热炒。之前Sun公司一直在不遗余力地推广Java，宣称Java applet将会改变人们浏览网页的方式。然而市场并没有像Sun公司预期的那样好，这很大程度上是因为Java applet速度慢而且操作不便。网景公司的市场部门抓住了这个机遇，与Sun合作完成了LiveScript在浏览器端的实现，并在网景的Navigator2.0发布前，将LiveScript更名为了JavaScript。网景公司为了取得Sun公司的支持，把JavaScript称之为Java applet和HTML的补充工具。

网景公司决不会预料到当年那个市场策略带来的副作用有多大。多年来，到处都有人混淆Java和JavaScript这两个不相干的语言。两者除了名字相似和历史渊源之外，几乎没有任何关系。布兰登·艾奇为此也一直抱憾不已，他后来在一个名为"JavaScript at Ten Years"(JavaScript这10年)的演讲稿中写道："Dont't let marketing name your language"(不要让营销来决定语言的名称)。

最初的JavaScript版本为1.0版本。JavaScript1.0出现之后，让原本只能在服务器端进行的表单验

证，在客户端就能够实现，这样大大增强了用户体验，并且还让网页上面存在能够动起来的东
西，也给当时的网页添加了一定的活力。所以JavaScript在诞生之初就获得了巨大的成功。网景
公司的Netscape浏览器也因此在整个浏览器市场上占据了超高的份额。



(图为浏览器Netscape Navigator 2.01)

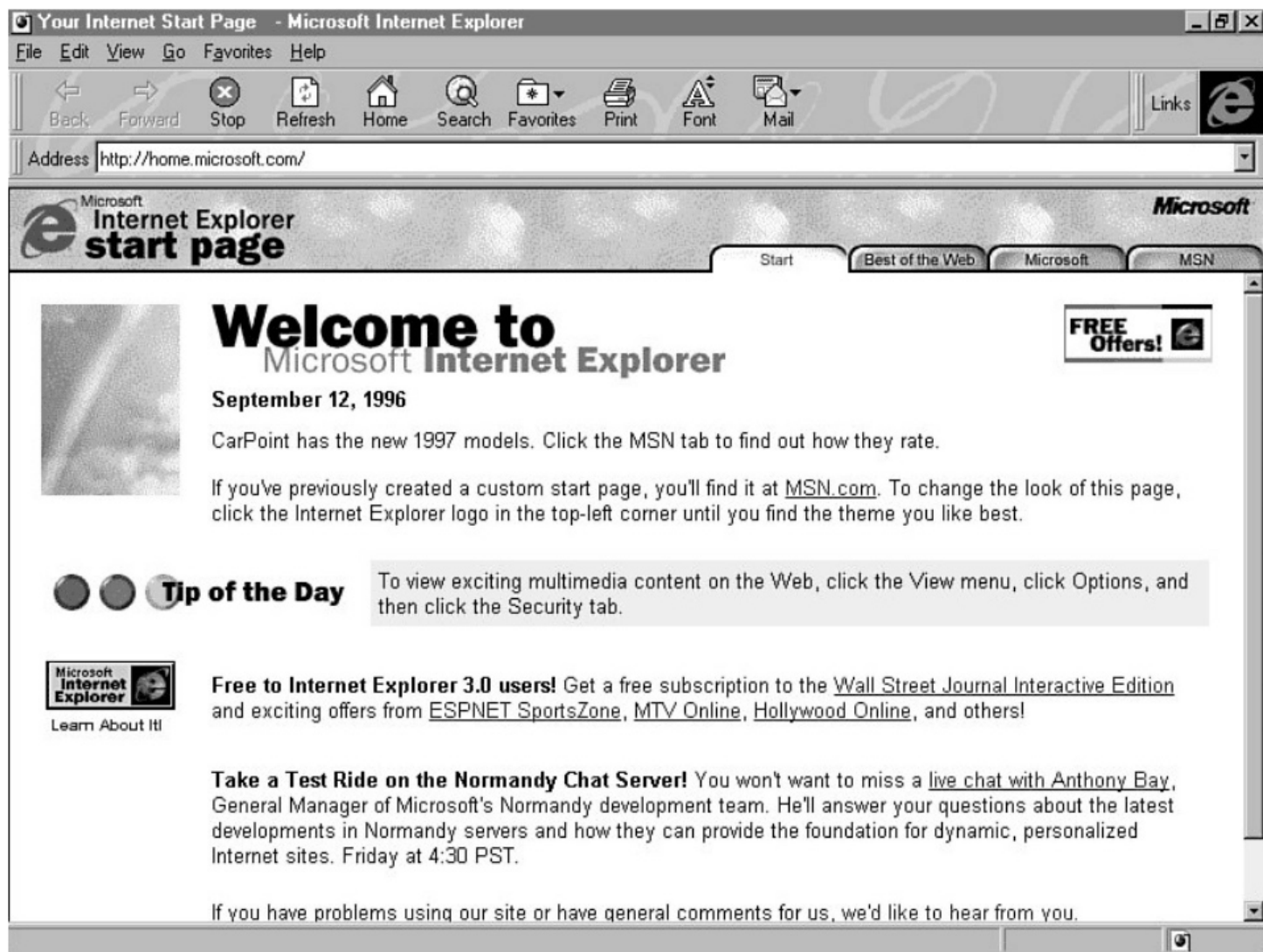
看到了网景公司的成功，当时的IT界霸主微软公司就想出了要收购网景公司的策略。但是网景公
司当时在整个Web领域有着相当高的地位，所以并不接受微软的收购协议。

于是，微软一怒之下就对JavaScript进行了反编译，也实现了类似于JavaScript的客户端脚本语
言JScript。并将其投放于自家的浏览器Internet Explorer供其使用。考虑到当时全球范围内有大量的VB程序员，所以还顺带弄了一个VBScript出来。JScript的诞生成为JavaScript发展的一个重要里程碑，标志了动态网页时代的全面到来。

至此，网景公司和微软公司之间开始了长达数年的浏览器以及客户端脚本语言的市场份额争夺之
战。

第二阶段：第一次浏览器大战

就在网景公司和微软公司进行激烈的浏览器市场份额之争的时候，微软的Internet Explorer3随Windows 95 OSR2捆绑销售的策略堪称一颗重磅炸弹。这一举动轻松击败了强劲的对手：网景公司的Navigator。尽管这个做法致使微软后来声名狼藉(以及一系列的反垄断诉讼)，但Internet Explorer3的成功却有目共睹，其成功不仅仅在于市场营销策略，也源于产品本身。Internet Explorer3是一个划时代的产品。



(图为Windows95上的Internet Explorer3)

最终，由于微软将Internet Explorer3浏览器嵌入到自家的操作系统里面的策略，对于普通用户来讲，也就没有必要去重新下载一款浏览器了。所以导致网景公司的浏览器市场份额骤降，使得微软赢得了第一次浏览器大战的胜利。

虽然两家公司为争夺浏览器市场份额而进行的数年之久的浏览器之战，最终以微软的胜利告终，但是这场"战争"也给当时的程序员带来不小的痛苦。因为在最初，JavaScript并没有一个标准。网景公司和微软公司各自为了争夺浏览器的市场份额，都在自己的脚本语言里面加入新的东西。然而这些新的东西大多数都是效果相同，只是实现的方法不同。这就造成了一个很尴尬的事情，那就是苦逼的开发人员往往要为了兼容性书写两份不同的代码。

对于这种语言没有一个标准，两家公司各自为战的情况，来自开发人员的怨言越来越重。于是，

针对JavaScript的标准化问题就被提上了日程，对JavaScript的标准化势在必行。

1997年，以JavaScript1.1为蓝本的建议被提交给了欧洲计算机制造商协会(ECMA)。该协会指定39号技术委员会(TC39)负责提供JavaScript的标准化，定出来的标准称之为ECMA-262，也就是ECMAScript。紧接着国际标准化组织也采纳了ECMAScript标准，定为ISO-16262。



不过，尽管有ECMAScript作为JavaScript的语法和语言特性的标准，但是关于JavaScript其他方面的规范还是不明确，加上当时两家公司也并没有重视标准的重要性，各自在自家的浏览器中又加入了各自特有的对象和函数。这也是为什么这么多年来同样的JavaScript代码会在不同的浏览器中呈现不同的效果，甚至在一个浏览器中可以执行，而在另一个浏览器中却不可以。这直接导致了当时的JavaScript处于一个非常混乱的局面。

并且，微软在获得浏览器大战的胜利以后，认为已经没有什么东西可以影响它霸主的地位了，一度停止了对JavaScript以及Internet Explorer的继续开发。JavaScript就一直处于一个只是给网页添加特效的时期。甚至一度被民众给厌恶，因为一些开发人员对JavaScript的滥用，导致网页上充斥着大量的弹窗和漂浮不定还很难关闭的广告，这都导致了人们对JavaScript的好感度骤降，JavaScript就这样一直被当作是一门玩具语言。

第三阶段：第二次浏览器大战

网景公司在第一次浏览器大战中战败以后，公司也面临着破产。破产以后，开发Navigator浏览器的那一波人成立了一个叫做Mozilla的基金会，并且将之前的Navigator浏览器的代码给开源了。这一下子，有大批的爱好者自发的为这款浏览器添加补丁和维护。

2002年，Mozilla在之前Navigator浏览器的基础上，推出了火狐浏览器。这一举动，标志着当年的网景又回来了。火狐浏览器也成为了当时浏览速度最快的浏览器。但是这个时候，并没有引起微软的重视，微软认为，还可以依靠之前的策略来打败这款浏览器，毕竟在当时，Internet Explorer6在世界范围内有着超高的市场份额。

接下来，我们回头来说一下JavaScript。虽说火狐浏览器推出了，但是JavaScript依旧没有改变它是用来做动画的命运，仍然被当作是玩具语言。直到2005年，这一年，JavaScript的命运彻底被改变。新兴IT公司Google推出了Ajax技术，并将其应用到了自家的Gmail上面。Ajax的出现，大大的改变了用户的上网体验，可以无刷新的改变页面内容。而Ajax的主体就是JavaScript。此时此刻，人们对这门以前的玩具语言，有了新的认识。



(2005年，Ajax技术诞生)

如果说第一次浏览器大战只有微软和网景两家公司打的火热，那么第二次浏览器大战远比第一次要激烈得多。2008年，Google推出了Google浏览器，苹果也推出了自家的Safari浏览器，还有以小巧著名的opera浏览器，都参与到了第二次浏览器大战里面。

比起第一次浏览器大战各自在自家的浏览器里面添加独特功能的情况，第二次浏览器大战一定程度上来讲可以看作是标准之战。因为这个时候，大家能意识到了遵循标准的重要性，谁的浏览器能够更接近标准，谁就能在市场上得到更高的认可。

第四阶段：JavaScript蓬勃发展

随着第二次浏览器大战所引发的Web标准之战，逐渐的升级到了浏览器的引擎之战。什么意思呢？JavaScript代码是需要JavaScript引擎来进行解析的，所以，在同样都满足标准的情况下，谁的浏览器能够更快的解析JavaScript代码，谁就能够更加获得青睐。

第一款JavaScript引擎是由布兰登·艾奇在网景的Navigator中开发的，它的名字叫做SpiderMonkey。SpiderMonkey在这之后还用做Mozilla Firefox1.0~3.0版本的引擎，而从Firefox3.5开始换为了TraceMonkey，4.0版本以后有换为了JaegerMoney。

微软从Internet Explorer9开始使用的JavaScript引擎为Chakra，而苹果的Safari浏览器使用的JavaScript引擎为Nitro。Google公司的Google Chrome的JavaScript引擎为V8引擎。

在整个引擎之战中，要属Google公司的V8引擎最为瞩目，该引擎成为了解析JS代码最快的引擎。甚至其速度能够接近于C语言的速度。V8引擎的出现，给JavaScript带来了一系列的革命。2009年，RyanDahl在V8引擎的基础上发布了Node.js。至此，JavaScript不再局限于客户端的开发，而是将触角延伸到了服务器端，可以进行服务器端的开发了。



(2009年，Node.js横空出世)

时至今日，各个JavaScript引擎的效率已经不相上下，通过不同引擎根据不同测试基准测得的结果各有千秋。但是，更有趣的是，JavaScript的效率在不知不觉中已经超越了其他所有传统的脚本语言，并带动了解释器的革新运动。JavaScript已经成为了当今速度最快的脚本语言之一，昔日的"丑小鸭"终于成了惊艳绝俗的"白天鹅"。

随着现在JavaScript的蓬勃发展，我们可以在很多领域都能见到JavaScript的身影。例如前面所说的Web开发还有服务器端开发，桌面应用开发，移动端开发，HTML5游戏开发等。

Jeff Atwood曾经说过：凡是能够用JavaScript来写的应用，最终都必将用JavaScript来写。也许未来，就会成为JavaScript的天下，我们有理由相信JavaScript会变得越来越好。

JavaScript版本介绍

说到JavaScript的版本，大致可以分为两种，一种是JavaScript版本，另一种是ECMAScript版本。我们知道，ECMAScript就是JavaScript的一个标准，而JavaScript是ECMAScript的一种实现。还有一点要明白的是，实现了ECMAScript的可不仅仅只有JavaScript，还有诸如ADOBE公司旗下的Flash所用到的ActionScript，也可以看作是ECMAScript的一种实现。

关于JavaScript版本，目前只有Mozilla公司，还在继续沿用最初的JavaScript版本编号序列对应的版本号如下表：

浏览器	JavaScript版本
Netscape Navigator2	1.0
Netscape Navigator3	1.1
Netscape Navigator4	1.2
Netscape Navigator4.06	1.3
Netscape6+	1.5
Firefox1	1.5

Firefox1.5	1.6
Firefox2	1.7
Firefox3	1.8
Firefox3.5	1.8.1
Firefox3.6	1.8.2

另一个是ECMAScript版本，从1997年发布的ES1到现在最新的ES6，发展路线也是挺曲折的，如下：

- 第1版：1997年6月发布，本质上与JavaScript 1.1 相同
- 第2版：1998年6月发布，主要是编辑加工的结果。
- 第3版：1999年12月发布，是对ECMAScript标准第一次真正的修改。
- 第4版：ES4由于太过激进，未被采用，与此同时TC39下属的一个小组提出了ES3.1过渡版本
- 第5版：ES3.1最终成为ES5，于2009年12月发布，该版本力求澄清第3版中的歧义
- 第6版：2015年6月17日发布。第6版标志着ECMAScript成为了一门真正的通用编程语言。

搭建JavaScript开发环境

搭建环境，无外乎就是书写代码的环境以及运行代码的环境。

对于书写代码的环境，前面我们也有提过，书写JavaScript代码使用记事本都能够书写，但是，使用工具的话可以让我们书写代码的速度更快，事半功倍。

书写代码的工具可以分为两大类：文本型代码编辑器以及集成开发环境(IDE)

常见的文本型代码编辑器如下：

- 记事本
- Sublime
- Atom
- Brackets
- VScode

常见的集成开发环境如下：

- Dreamwaver
- Webstrom

接下来是代码运行环境。最早的时候，JavaScript一般都是在浏览器上面运行，但是自从Node.js

出现以后，JavaScript已经有了服务器端的运行环境，并且随着ES6的发布，JavaScript已经成为了一种通用编程语言，所以刚开始的时候，我们完全可以在Node里面运行了的JS代码。

安装过程Node.js

在Windows和Mac上面都可以很轻松的安装Node.js，只需要访问

<https://nodejs.org/en/>

点击Download链接，下载相应的安装包，然后进行安装即可。

安装好以后，可以在控制台里面通过node -v命令查看当前安装的node的版本

```
Jie-Xie:~ Jie$ node -v  
v8.7.0
```

还有一种方式是使用在线的运行环境，这里推荐

ES6console.com

www.ES6console.com

该网站可以让我们在线的书写JavaScript代码，并且在线的运行立即看到效果。

第一个JavaScript程序

书写代码的方式非常简单，可以通过控制台下面，输入 node 进入REPL(Read Eval Print Loop)模式。即输入-求值-输出循环的模式。在该模式下可以进入一个即时求值的运行环境，对于测试简单的代码这个环境是非常有用的，示例如下：

```
Jie-Xie:~ Jie$ node  
> console.log("Hello World");  
Hello World  
undefined
```

无论是Windows系统还是Mac系统，我们都可以通过 Ctrl+d 键的组合来退出REPL模式。

除了上面所介绍的REPL，我们也可以自己新建一个文本文件，以js为后缀，书写好JS代码以后在node里面运行。这里我在我的桌面上新建了一个1.js的文件，书写如下的代码：

```
console.log("Hello World!!!");
```

在node里面运行，首先cd切换到desktop路径下面，然后输入 `node 文件名` 即可运行：

```
> Jie-Xie:~ Jie$ cd desktop
Jie-Xie:desktop Jie$ node 1
Hello World
```

NPM介绍

NPM全称Node Package Manager，是一个由Node.JS官方提供的包管理和分发工具，上面有很多可供我们使用的包，可以为我们书写代码带来便捷。

在现在的Node.JS版本中(Node.JS 0.6开始)，npm已经包含在安装包里面了，我们在Windows或者Mac上安装Node.JS时会自动安装NPM。

如果你安装的Node.JS没有自动安装NPM，那么需要你自己手动来进行安装，安装方法如下：

1.安装NPM指令：(外网)淘宝镜像：在命令行中敲入

```
npm install -g cnpm --registry=https://registry.npm.taobao.org
```

2.(学校局域网)朗沃服务器镜像：在命令行中敲入

```
npm config set registry=http://192.168.8.10:7001
```

3.安装好以后我们可以对它进行初始化，初始化指令如下：

```
npm init: npm初始化设置
```

4.安装好npm后，我们来安装一个常用的输入模块，指令如下：

```
npm install readline-sync: 安装输入模块
```

接下来在书写代码的时候，就可以引入这个输入模块，从而在控制台进行输入了

```
let readline = require("readline-sync");
console.log("请输入一点东西:");
let i = readline.question(""); // 接收用户输入的内容
console.log("你输入的是: ", i);
```

效果：

```
Jie-Xie:desktop Jie$ node 1
```

请输入一点东西：

```
this is a test
```

你输入的是： **this** is a test

总结

1. JavaScript的发展史大致可以分为4个阶段，分别是JavaScript起源，第一次浏览器大战，第二次浏览器大战和JavaScript蓬勃发展
2. Java和JavaScript是两个不相干的语言。两者除了名字相似和历史渊源之外，几乎没有任何关系。
3. 1997年，以JavaScript1.1为蓝本的建议被提交给了欧洲计算机制造商协会(ECMA)。该协会指定39号技术委员会(TC39)负责提供JavaScript的标准化，定出来的标准称之为ECMA-262，也就是ECMAScript。
4. 2002年，Mozilla在Navigator浏览器的基础上，推出了火狐浏览器。
5. 2005年，Google公司推出了Ajax技术，并将其应用到了自家的Gmail上面。Ajax的出现，大大的改变了用户的上网体验，可以无刷新的改变页面内容。而Ajax的主体就是JavaScript。
6. 2009年，RyanDahl在V8引擎的基础上发布了Node.js。至此，JavaScript不再局限于客户端的开发，而是可以进行服务器端的开发了。
7. JavaScript的版本，大致可以分为两个部分，一个是JavaScript版本，另一个是ECMAScript版本。
8. 搭建JavaScript开发环境，可以分别书写代码的环境以及运行代码的环境。
9. REPL(Read Eval Print Loop)模式。即输入-求值-输出循环的模式。在该模式下可以进入一个即时求值的运行环境，对于测试简单的代码这个环境是非常有用的。
10. NPM全称Node Package Manager，是一个由Node.JS官方提供的包管理和分发工具，上面有很多可供我们使用的包，可以为我们书写代码带来便捷。