21世纪高等教育计算机规划教材

COMPUTER

JavaScript 程序设计基础教程

JavaScript Programming Tutorial

李源 编著

- 从基础到进阶，从理伦到实战，循序渐进
- 实例代码中注释明晰，通俗易懂
- 程序案例采用了分步骤实现方

中国工信出版集团

人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等教育计算机规划教材

COMPUTER

JavaScript 程序设计基础教程

JavaScript Programming Tutorial

李源 编著

人民邮电出版社

北京

图书在版编目（CIP）数据

JavaScript程序设计基础教程 / 李源编著. -- 北京：人民邮电出版社，2017.4
21世纪高等教育计算机规划教材
ISBN 978-7-115-44327-4

Ⅰ. ①J… Ⅱ. ①李… Ⅲ. ①JAVA语言－程序设计－高等学校－教材 Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2016)第297311号

内 容 提 要

JavaScript 是目前最流行的网页前端开发技术之一。本书由浅入深、循序渐进地介绍了使用 JavaScript 开发网页前端应用的基础知识和技术技能。

全书分为 3 篇。第 1 篇是 JavaScript 语法基础，包括 JavaScript 简介、基本语法、数据类型、控制语句、函数与数组等。第 2 篇是 JavaScript 面向对象基础，包括 JavaScript 面向对象编程、屏幕和浏览器对象、文档对象、窗口对象、历史地址与 cookie 对象以及表单和 DOM 对象。第 3 篇是 JavaScript 进阶与实战，包括 JavaScript 中正则表达式的使用、jQuery 框架的使用以及一个接元宝游戏实例。通过进阶技术的学习与综合实例的练习，读者能真正感受到 JavaScript 的魅力。

本书语言通俗，内容精练，重点突出，实例丰富，是广大 Web 开发人员、计算机编程爱好者、网站管理维护人员必备的参考书，也非常适合大中专院校师生学习阅读，并可作为高等院校计算机及相关专业教材使用。

◆ 编　　著　李　源
责任编辑　吴　婷
责任印制　杨林杰

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路 11 号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京天宇星印刷厂印刷

◆ 开本：787×1092　1/16
印张：16.75　　2017 年 4 月第 1 版
字数：440 千字　　2017 年 4 月北京第 1 次印刷

定价：45.00 元

读者服务热线：(010)81055256　印装质量热线：(010)81055316
反盗版热线：(010)81055315

前言

未来的互联网是桌面平台与移动平台平分秋色的世界，而作为联系二者的纽带，Web 应用将在未来的跨平台应用中占据重要的地位。所以作为 Web 应用的流行技术 JavaScript，也受到越来越多的程序员的重视与喜爱。通过 JavaScript 可以构建功能完备、界面友好的 Web 应用前台，而好的前端会帮助网站留住更多的客户。因此，学好 JavaScript 也成为各类应用开发，特别是跨平台的 Web 应用开发所必备的条件之一。

为了方便广大读者学习，作者结合自己多年的 JavaScript 开发和培训经验编写了本书。本书比较全面地介绍了 JavaScript 基础、JavaScript 面向对象、JavaScript 中正则表达式的使用以及 jQuery 框架的使用等。通过本书的学习，读者可以对 JavaScript 有深入的了解，并且可以实现各类 Web 前台使用 JavaScript 所要求的功能。

本书的特点

1. 语言简练，通俗易懂

本书尽量使用通俗易懂的语言来组织内容，避免使用艰深晦涩的专业术语，让初学者更容易接受，从而为学好、用好 JavaScript 打好基础。

2. 内容丰富，知识全面

本书共分 3 篇 15 章，采用从易到难、循序渐进的方式进行讲解。内容几乎涉及了 JavaScript 程序开发的各个方面。

3. 循序渐进，由浅入深

为了方便读者学习，本书首先让读者了解什么是 JavaScript，然后介绍最常用的语法基础内容，如运算表达式、控制语句等。读者在掌握语法基础之后，逐渐学习 JavaScript 的面向对象编程，了解各种对象的使用方法。接着通过对正则表达式与 jQuery 的学习实现知识的进阶，最后通过一个综合实例，使读者学以致用，掌握 JavaScirpt 的开发技巧。

4. 格式统一，讲解规范

书中每个例程都采用了分步骤实现的方法。这样读者可以很清晰地知道每个技术的具体实现步骤，从而提高学习的效率。

5. 实例丰富，注释明晰

本书通过实例来说明重要知识点的用法，而且实例代码中都有清晰明了的注释，读者即使对代码的运行不太理解，也可以根据注释了解代码所实现的功能，从而对读者学习该知识点有着很好的引导作用。

本书的内容安排

本书分为 3 篇，共 15 章，主要章节规划如下。

第 1 篇（第 1 章 ~ 第 6 章）为 JavaScript 语法基础，内容包括 JavaScript 简介、数据类型、运算符与表达式、控制语句、函数和数组、JavaScript 的调试与优化等基础知识。

第 2 篇（第 7 章 ~ 第 12 章）为 JavaScript 面向对象基础，讲述了 JavaScript 面向对象编程，屏幕和浏览器对象，文档对象，窗口对象，历史、地址和 cookie 对象，

表单对象与 DOM 对象等。

第 3 篇（第 13 章～第 15 章）为 JavaScript 进阶与实战，讲述了 JavaScript 中的正则表达式技术、jQuery 框架使用技术以及一个综合实例——接元宝网页小游戏。

本书由浅入深，由理论到实践，尤其适合初级读者逐步学习和完善自己的知识结构。

适合阅读本书的读者

- 希望进入 Web 开发领域的新手。
- JavaScript 学习人员。
- 从事 Web 前端开发的人员。
- 想使用 JavaScript 开发网络应用的人员。
- 各类网站管理人员。
- 想自学制作网站的网络爱好者。
- 大中专院校的学生。

本书由解放军第三〇二医院计算机信息中心的李源主编，其他参与编写的还有梁静、黄艳娇、任耀庚、刘海琛、刘涛、蒲玉平、李晓朦、张鑫卿、李阳、陈诺、张宇微、李光明、庞国威、史帅、何志朋、贾倩楠、曾源、胡萍凤、杨罡、郝召远。

编　者

2016 年 11 月

目 录

第 1 篇 JavaScript 语法基础

第 2 篇 JavaScript 面向对象基础

第 3 篇 JavaScript 进阶与实战

第1篇 JavaScript语法基础

第1章 认识JavaScript

“千里之行，始于足下”。这句千古遗训蕴含着深刻的道理，在计划安排好之后需要落实行动。只有从现在的脚下开始出发，才能达千里之外的目的地。学习 JavaScript 最好从了解它的起源开始，了解其产生的背景，从而知道其主要应用场合，对今后的学习和目标的建立有莫大的帮助。本章将向读者讲解 JavaScript 的背景和现在的状况，以及未来可能的发展方向。通过本章的学习，读者将学会编写一个最简单的 JavaScript 程序并知道如何运行。

- 了解 JavaScript 产生的背景。
- 了解 JavaScript 和其他脚本语言的异同。
- 了解如何编写一个 JavaScript 程序并运行它。
- 牢记编写 JavaScript 程序的注意事项。

以上几点是对读者所提出的基本要求，也是本章希望达到的目的。读者在学习本章内容时可以将其作为学习的参照。

1.1 脚本语言 JavaScript

JavaScript 是世界上使用人数最多的程序语言之一，几乎每一个普通用户的电脑上都存在 JavaScript 程序的影子。然而绝大多数用户不知道它的起源以及如何发展至今。JavaScript 程序设计语言在 Web 领域的应用越来越火，未来它将会怎样发展，本节将对这部分内容分别进行讲述。

1.1.1 脚本语言的分类

如今成熟的脚本语言非常多，根据使用方式的不同分成嵌入式和非嵌入式两类。嵌入式脚本语言通常为了应用程序的扩展而开发出来。解释器通常嵌入在被扩展的应用程序中，成为宿主程序的一部分。例如，Lua 语言、Python 语言的嵌入性比较好，如今这两者在游戏开发领域应用较多，通常作为游戏软件的脚本系统或配置文件。根据笔者的经验，Lua 语言无论在嵌入性和运行效率上都远超过其他语言，将 Python 语言纳入嵌入式语言分类中有些勉强，因为它更像其他独立运行的语言。

非嵌入式脚本语言无须嵌入其他程序中，如本书所讲的 JavaScript 语言。这些语言主要应用不是作为系统扩展，而是实现一般的任务控制。

【提示】将语言分类比较勉强，因为其在开发的时候大都针对某一类应用而不先考虑属于某一类。

1.1.2 JavaScript 的标准与历史

众多 Web 浏览器对 JavaScript 的支持很不一致，相同的语言特性在不同的浏览器中会有所差异，这种差异对开发者影响极大，开发者在开发时不得不为不同的浏览器编写不同的代码，这种难堪的局面一直持续到 JavaScript 标准的制定。1997 年发布了 ECMA-262 语言规范，将 JavaScript 语言标准化并重命名为 ECMAScript，现在各种浏览器都以该规范作为标准。

【提示】语言和系统接口标准化后可以大大减轻开发人员的负担，不用为不同的语言特性或接口编写不同的代码，这也增强了软件的可移植性。

在互联网形成的初期，Web 技术远远没有像今天这样丰富，这样让人难以选择。当时，在 Web 客户端进行最基本的数据有效性验证都非常麻烦，浏览器端的用户体验效果非常单调，几乎没有交互性。今天所看到的全动态 Flash、SilverLight、JavaScript 等精彩应用在当时都没有，有的只是纯 HTML 静态页。

基于这样的状况，Netscape 公司在它的 Navigator Web 浏览器中增加了脚本功能，以简单的方式实现在浏览器中的数据验证，该脚本名为 LiveScript。与此同时，Java 技术也逐渐红火，其特点也正好能弥补 Web 客户端交互性方面的不足。Netscape 公司在其 Navigator 浏览器中支持 JavaApplet 时，考虑 JavaApplet 与 LiveScript 目标的相似性，将 LiveScript 更名为 JavaScript，可以理解为其欲借 Java 之势以求发展。

JavaScript 语言刚推出就在市场上获得巨大的成功，这表现在 Navigator 浏览器的用户量上。当 JavaScript 语言的使用形成一种大趋势之后，微软公司的 IE 浏览器也增加对 JavaScript 语言的支持，这加快了 JavaScript 语言发展的速度。

微软公司的 IE 浏览器搭乘 Windows 操作系统这艘巨舰在市场上获得了空前的成就，同时微软也实现了一门兼容 JavaScript 的脚本语言，命名为 JScript。如今对 JavaScript 的支持已经成为 Web 浏览器中不可缺少的技术。

【提示】很多有名的编程语言起初都是由个人或小团体创造出来，逐步完善并发展壮大的。

随着 Ajax 的技术大潮，JavaScript 重新受到 Web 开发者的重视。在此之前，JavaScript 主要应用还是在客户端实现一些数据验证等简单工作，多媒体交互应用被类似 Flash 的技术抢占了市场。正当 JavaScript 处于低潮的时候，Ajax 技术被开发出来了，简单地说，就是利用 JavaScript 的异步更新机制实现 Web 页的局部刷新。当一个页面不需要全部重新加载，只要加载部分数据即可的时候，互联网的运行速度便大大加快了。因此，JavaScript 在 Web 开发中站在了一个更加重要的位置。JavaScript 在浏览器中的层次结构如图 1-1 所示。

很多开发者开始挖掘 JavaScript 在其他方面的潜力，打算发现类似 Ajax 那样令人吃惊的东西。结合 W3C 现行的 DOM 规范，JavaScript 表现出了惊人的魅力，涌现出很多基于 Web 的应用程序，这是在 Web 客户端方面。在服务器端技术中，微软公司也将 JavaScript 纳入了.NET 语言的范畴，使其成了 ASP.NET 的语言工具，开发者不必重新学习语言即可运用 ASP.NET 技术。如今基于 JavaScript 的应用不胜枚举，大家可通过网络了解更多的信息。

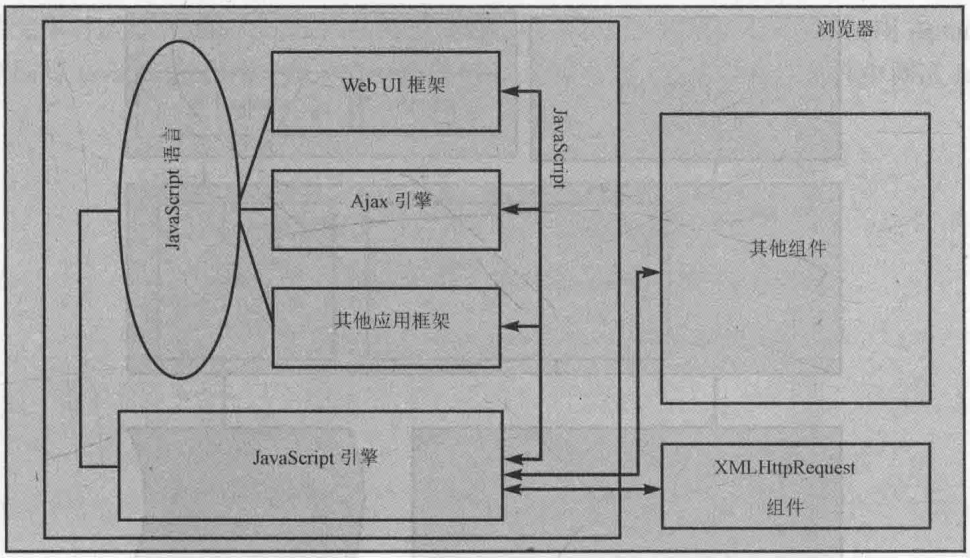

图 1-1　JavaScript 在浏览器中的层次结构

【提示】自从 Ajax 技术出现之后，人们重新重视了 JavaScript 的价值，如今不少开发者使用 JavaScript 开发出极具价值的通用程序框架，如一些流行的 Web UI 库。

1.1.3　JavaScript 在网页中的应用

JavaScript 的脚本包括在 HTML 中，它成为 HTML 文档的一部分。与 HTML 标志相结合，构成了一门功能强大的 Internet 网上编程语言。使用特定的标记可以直接将 JavaScript 脚本加入文档。

```
01  <html>                                          <!--文档开始-->
02  <head></head>                                   <!--文档头--->
03  <body>                                          <!--文档体--->
04  <script Language ="JavaScript">
05  JavaScript 语言代码; JavaScript 语言代码; ....
06  </script>
07  </body>                                         <!--文档体结束-->
08  </html>                                         <!--文档结束--->
```

以上代码中 4～6 行为 JavaScript 代码，其余部分为 HTML 内容。通过以上代码可以看出，在 HTML 中使用 JavaScript 要使用<script>标记，并在其中指定所要使用的脚本语言为 JavaScript，即：<script Language ="JavaScript">。

1.1.4　JavaScript 的发展趋势

语言永远被当作工具，这一点从来都没有被改变过，以后也不会。例如，在 Windows 平台上，使用 ADODB 组件可以让 JavaScript 能处理支持 SQL 的数据库中的数据，使用 FSO 组件可以实现本地文件 IO 功能。这些说明了 JavaScript 位于应用开发的最顶端，与低层技术的实现无关，JavaScript 在系统中的位置如图 1-2 所示。

尽管平台技术不断发生变化，JavaScript 仍将以不变的形式去使用平台提供的能力从而适应新的需求。未来的一段时间内，Web 开发将是开发者众聚之地，也是 JavaScript 变得紫红的时代。

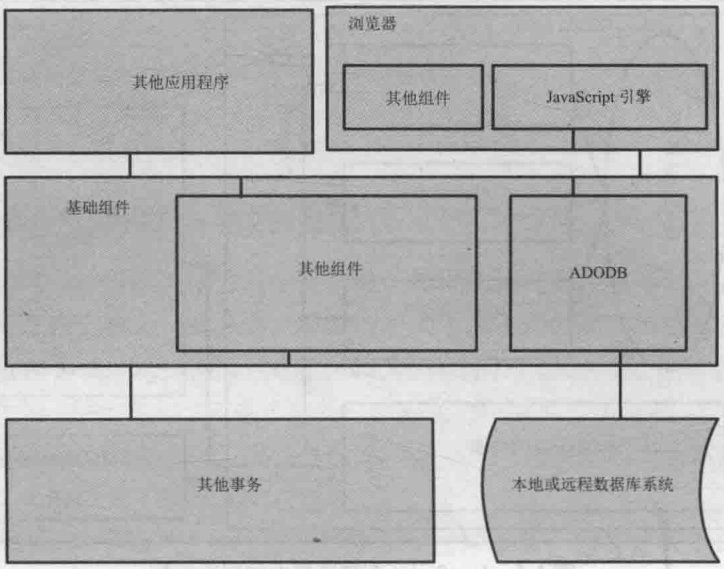

图 1-2　JavaScript 在系统中的位置

1.2　第一个 JavaScript 程序

学习一门新语言，大致了解了它的背景之后，最想做的莫过于先写一个最简单的程序并成功运行。如果最初连续几个程序都无法成功编译或运行，初学者学习的信心多少会受些打击，这是正常现象。本节将带领读者对 JavaScript 进行第一次实践尝试，用它编写一个最简单且流行了几十年的“Hello World”程序。

1.2.1　选择 JavaScript 编辑器

JavaScript 源程序是文本文件，因此可以使用任何文本编辑器来编写程序源代码，如 Windows 操作系统里的“记事本”程序。为了更快速地编写程序并且降低出错的概率，通常会选择一些专业的代码编辑工具。专业的代码编辑器有代码提示和自动完成功能，笔者推荐使用 Aptana Studio，它是一款很不错的 JavaScript 代码编辑器，其安装初始界面如图 1-3 所示。

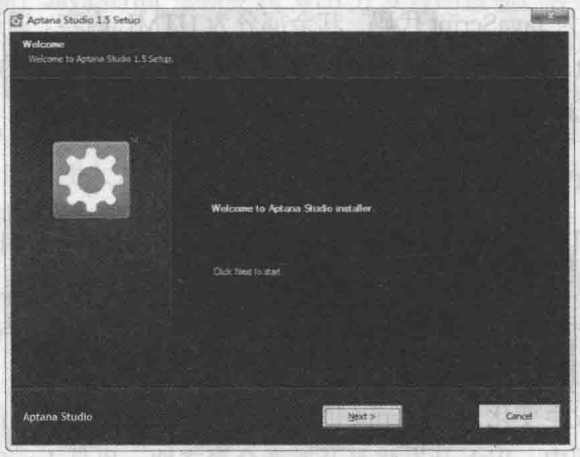

图 1-3　开始安装 Aptana Studio

安装完毕后运行 Aptana Studio，即可进入程序的主界面，如图 1-4 所示。使用 Aptana Studio 可以快速编写 JavaScript 程序。如果使用的是 Firefox 浏览器，还可以在该软件中调试 JavaScript 程序。

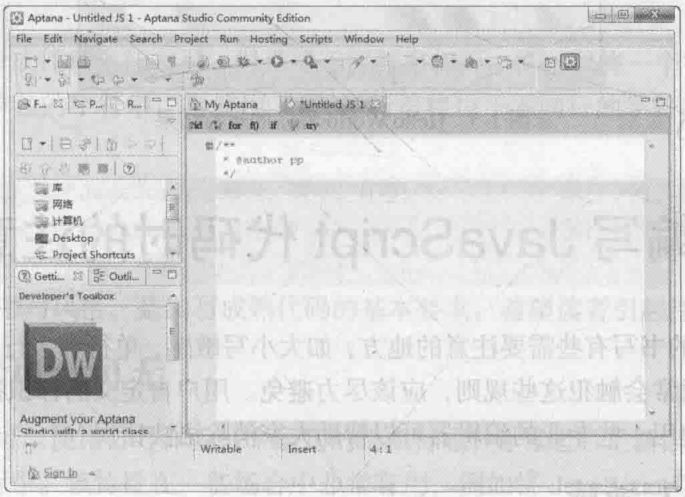

图 1-4　Aptana Studio 主界面

1.2.2　编写 Hello World 程序

下面正式开始编写“Hello World”程序，推荐使用记事本或 1.2.1 小节介绍的 Aptana Studio。为简单起见，这里使用记事本编写程序。

【实例 1-1】编写并运行最经典的入门程序，输出“Hello World!”。打开记事本，输入如下代码并将文件另存为网页文件“helloworld.htm”。

```
<html>                          <!---------HTML 文档开始----->
<body>                          <!--------文档体开始-------->
<script language="JavaScript"><!---------脚本程序--------->
    document.write("Hello World!");// 输出经典的 Hello World!
</script>                       <!---------脚本结束--------->
</body>                         <!---------文档体结束-------->
</html>                         <!---------HTML 文档结束----->
```

编写完毕，将以上代码保存为范例 1-1.html 以备后用。

【代码解析】第 4 行是 JavaScript 程序代码，第 3、5 行是标准 HTML 标签，该标签用于在 HTML 文档中插入脚本程序。其中的“language”属性指明了“<script>”标签对间的代码是 JavaScript 程序。第 4 行调用 document 对象的 write 方法将字符串“Hello World!”输出到 HTML 文本流中。

【提示】嵌入 JavaScript 脚本时也可以使用标签“<script type="text/JavaScript"> </script>”。

1.2.3　运行程序

运行 JavaScript 程序最简单的方法就是使用浏览器打开包含 JavaScript 代码的网页文件，通常网页文件的扩展名为 htm 或者 html。使用系统自带的浏览器即可。双击网页文件运行程序，其结果如图 1-5 所示。

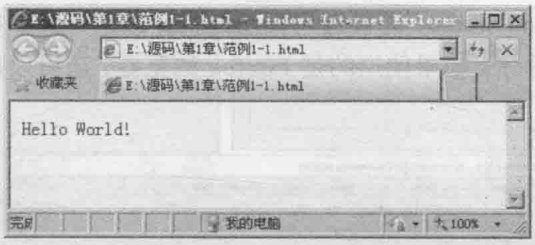

图 1-5　Hello World 程序的运行结果

1.3　编写 JavaScript 代码时的注意事项

JavaScript 程序的书写有些需要注意的地方，如大小写敏感、单行和多行、分号的运用等。初学者在编写程序时通常会触犯这些规则，应该尽力避免。用户自定义的标识符不能与语言保留的关键字同名，通过使用一些专业的编辑器可以帮助大家消除语法错误。

1.3.1　大小写敏感

JavaScript 代码是大小写敏感的，Name 和 name 是不同的标识符，编码时应当予以注意。同一个词如果各个字母间大小写不同，系统将当作不同的标识符来处理，相互之间没有任何联系。现举例说明，代码如下所示。

```
01  Name = "sunsir";                    // 大写字母开头
02  name = "foxsir";                    // 小写字母开头
```

此时 Name 的值仍然是“sunsir”，对 name 进行操作并不影响到变量 Name，它们是不同的变量，因为在 JavaScript 中所有的代码都区分大小写。

1.3.2　空格与换行

代码中多余的空格会被忽略，同一个标识符的所有字母必须连续。一行代码可以分成多行书写，以下代码的书写都正确。单行书写如下。

```
if(1==1 && 6>3 ){alert("return true");}else{alert( "return false" );}
// 代码写于一行中，用分号作为语句结束标志分成多行、规范的书写如下：
01  if( 1==1 && 6>3 )                   // 如果 1 等于 1，且 6 大于 3，则
02  {
03      alert("return true" );           // 输出“return true”
04  }
05  else                                 // 否则
06  {
07      alert( "return false" );         // 输出“return false”
08  }
```

也可以在代码中的标识符间任意添加空格，多余的空格会被忽略，如以下代码效果与上述代码完全一样。

```
01  if     (          1                 // 一个语句分多行书写
02  ==1                                 // 将一行代码分成多行
03  && 6>        3                      // 将一行代码分成多行
04  )                                   // 将一行代码分成多行
```

```
{ alert(                                    // 将一行代码分成多行
"return true"); }else                       // 将一行代码分成多行
{                                           // 将一行代码分成多行
    alert( "return false" );                // 将一行代码分成多行
}                                           // 将一行代码分成多行
```

虽然代码可以分成任意多行去写，但是对于字符串却不一样。要将一个字符串分成多行，须将每一行作为一个单独的字符串，再使用“+”运行符将位于不同行的字符串连接起来。代码如下所示。

```
var Message = "JavaScript 编程，简单，有趣！";        // 单行中的字符串
var message = "JavaScript 编程，" +                  // 多行中的字符串
              "简单，有趣！";
```

【提示】规范的书写风格，是编写成熟代码的基本要求，希望读者引起注意。

1.3.3　分号可有可无

JavaScript 程序可以使用分号作为一个语句的结束标志，分号之后是新语句的开始，这样可以将多个语句放在一行中。该特性在一些场合中非常有用，例如将 JavaScript 程序写在一个字符串中以构造函数对象。当一行只有一个程序语句时，结尾可以不使用分号。反之，当不使用分号时，一行被认为是一个程序语句，代码如下所示。

```
<script language="JavaScript">                  // 脚本开始
    var name = "Sunsir"                         // 名字
    var age = 25                                // 年龄
    alert( "Sunsir's age:" + age )              // 输出信息
</script>                                       // 脚本结束
```

1.3.4　注释形式

作为一种流行的编程语言，JavaScript 也支持对代码进行注释。使用注释一方面可以提高代码的可读性，另一方面也可以在作者对代码进行升级时能够清楚地理解各部分代码的作用。

在 JavaScript 代码中有单行注释与多行注释两种形式。其中单行注释的符号为“//”。在本章前几节的代码中已经出现。

多行注释的符号为：/**/，下面的代码就使用了多行注释。

```
<script language="JavaScript">                  // 脚本开始
    /*注释开始
    这里是注释内容
    这里是注释内容
    ……
    注释结束*/
</script>                                       // 脚本结束
```

以上代码中使用了多行注释，其中以“/*”开始注释，中间所有的内容均为注释内容，均不会被解释执行，最后以“*/”结束注释。

1.4 小　　结

本章介绍了 JavaScript 语言产生的背景、发展的过程及使用方法。现行的 JavaScript 是以 ECMAScript 为语言标准的，常见的浏览器基本上都实现了 ECMA-262 语言规范。对于不同浏览器间的一些微小的差别读者仍需注意，可以在程序中判断当前浏览器并编写与之适应的代码。JavaScript 程序以文本的形式嵌入或链接到 HTML 文档中，其代码标识符大小写敏感。一个程序语句可以分成多行书写，可以使用分号作为语句的结束标志。

1.5 习　　题

一、简答题

1. 简述 JavaScript 的发展史以及它的未来。
2. 简述 JavaScript 语言的特点。

二、练习题

编写程序，在浏览器中显示用户的名字。

【提示】对 Hello World 程序稍加修改即可实现，差别只是输出不同的字符串。参考代码如下。

```
<script language="JavaScript">                    // 脚本开始
      name = "Sunsir";                            // 名字
      document.write( name );                     // 在浏览器中输出
</script>                                         // 脚本结束
```

【运行结果】打开网页运行程序，结果如图 1-6 所示。

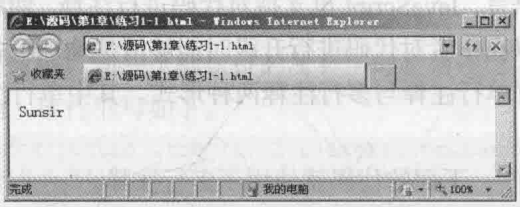

图 1-6　输出字符串

【提示】本书假定读者具有基本的 HTML 语言知识，非必要情况，HTML 部分代码将不多做解释。