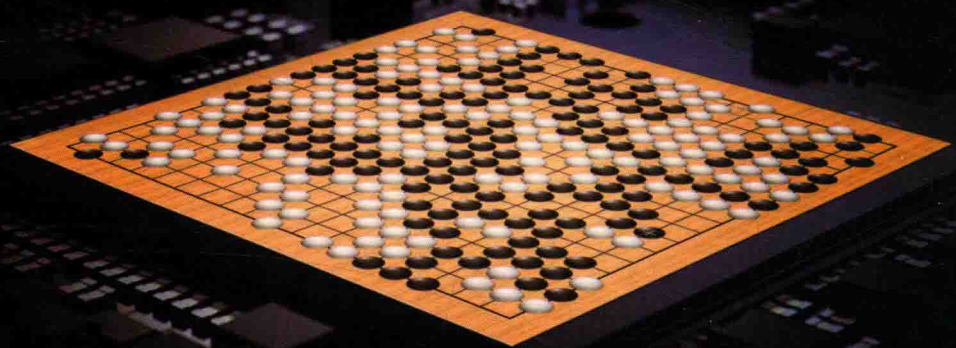




JavaScript

网页游戏制作轻松学

程 晨 著



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

JavaScript

网页游戏制作轻松学

程晨 著



人民邮电出版社
北京

图书在版编目 (C I P) 数据

JavaScript网页游戏制作轻松学 / 程晨著. — 北京:
人民邮电出版社, 2017. 8
(i创客)
ISBN 978-7-115-46238-1

I. ①J… II. ①程… III. ①JAVA语言—程序设计
IV. ①TP312.8

中国版本图书馆CIP数据核字(2017)第149746号

内 容 提 要

JavaScript是一种直译式脚本语言, 它的一个很大的优势就是绝大多数浏览器都包含了JavaScript的解释器, 这样我们在学习JavaScript时不需要安装任何开发软件, 只要写一个网页文件, 嵌入JavaScript代码就能在浏览器中进行代码的调试, 非常方便。对于硬件爱好者来说, 学习JavaScript还有一个好处就是能让自己的硬件作品更贴近IoT(物联网)应用。了解了JavaScript, 我们能够在显示的页面上实现更多的效果, 同时, 目前有很多硬件也支持用JavaScript来编程。

本书以制作国际象棋、五子棋、弹球、贪吃蛇、拼图、打飞机等经典游戏为例, 带你从零起步, 学习JavaScript的编程方法, 适合对游戏和编程感兴趣的初学者阅读。而且本书融入了创客的项目式思路, 在一个个实例的制作过程中学习知识, 语言通俗易懂, 图文并茂, 会带给你与以往内容枯燥的编程教程完全不同的全新体验。

◆ 著 程 晨

责任编辑 周 明

责任印制 周昇亮

◆ 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>

北京缤索印刷有限公司印刷

◆ 开本: 690×970 1/16

印张: 9.75

2017 年 8 月第 1 版

字数: 208 千字

2017 年 8 月北京第 1 次印刷

定价: 49.00 元

读者服务热线: (010)81055339 印装质量热线: (010)81055316

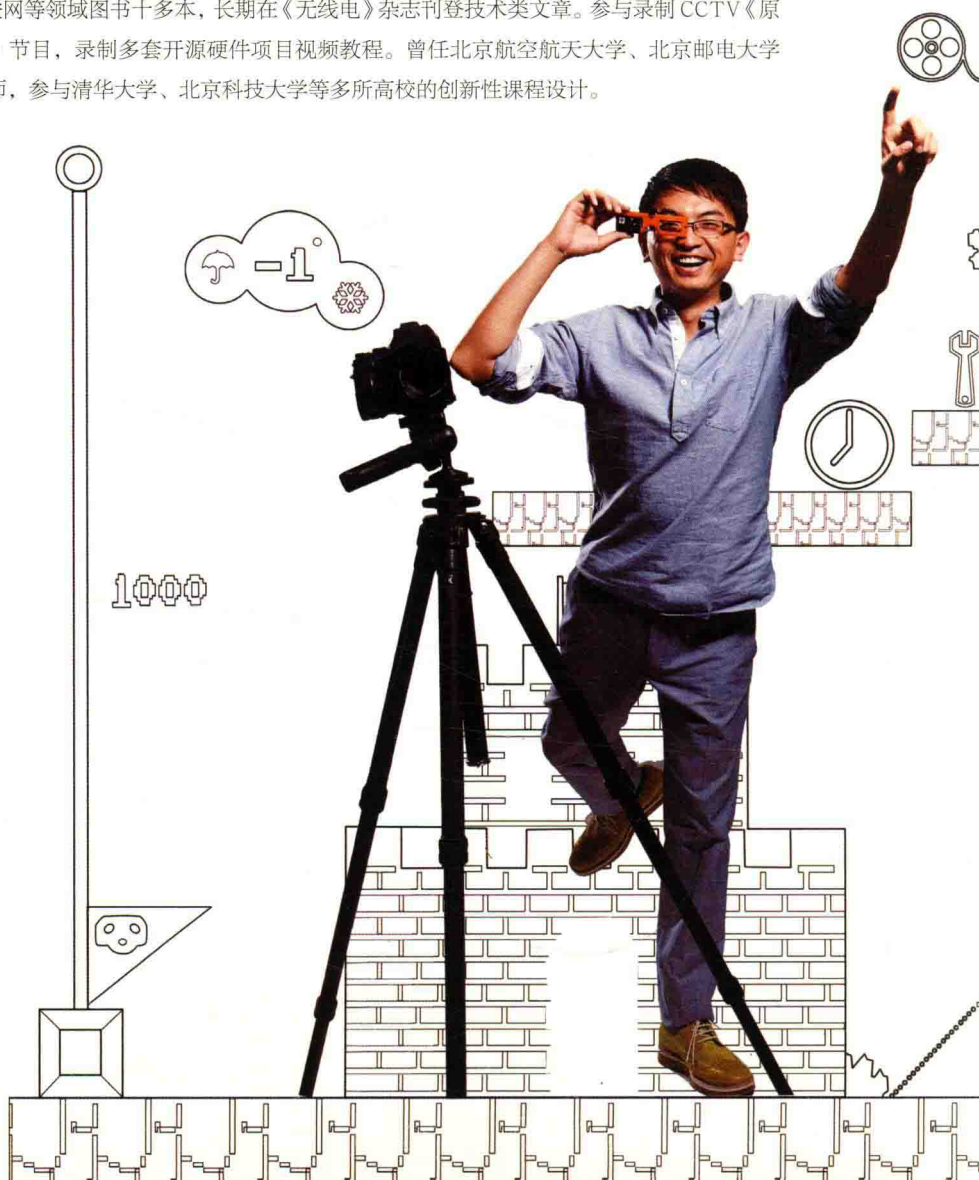
反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字 20170147 号

作者简介

程晨

创客布道师、创客教育践行者，拥有十多年嵌入式开发经验，技术方面涉猎广泛，先后主持、参与十多项军民科研产品的研发，拥有多项发明专利。少年创学院联合创始人兼院长、砾喜教育联合创始人兼技术总监，兼任 Intel 软件创新大使、中国电子学会全国青少年机器人技术等级考试标准工作组副组长、中国教育技术协会数字教育资源专业委员会专家等职务。第一本 Arduino 中文书作者，编著、翻译开源硬件、3D 打印、机器人、智能硬件、物联网等领域图书十多本，长期在《无线电》杂志刊登技术类文章。参与录制 CCTV《原来如此》节目，录制多套开源硬件项目视频教程。曾任北京航空航天大学、北京邮电大学特聘讲师，参与清华大学、北京科技大学等多所高校的创新性课程设计。



前言

JavaScript 是一种直译式脚本语言，是一种动态类型、弱类型、基于原型的语言。我认为它的一个很大优势就是绝大部分浏览器都包含了 JavaScript 的解释器，我们在学习 JavaScript 时不需要安装任何开发软件，只要写一个网页文件，嵌入 JavaScript 代码就能在浏览器中进行代码的调试，非常方便。

学习 JavaScript 是一直以来停留在我脑海中的一个想法，但一直未能实施，真正让我开始接触 JavaScript，还得缘于前段时间我撰写《米思齐实战手册——Arduino 图形化编程指南》这本书。因为米思齐是基于 Blockly 的，很多内容都靠 JavaScript 来完成，为了完成那本书的内容，我才开始沉下心来较为系统地学习 JavaScript 的内容。虽然在那本书中涉及 JavaScript 的内容较少，但这件事让我迈进了 JavaScript 的大门。

入门之后，我发现 JavaScript 中的 canvas（画布）应用这块内容非常有意思，于是就在少年创学院开了一期 JavaScript 培训班，主要内容就是在 canvas 中实现简单的平面游戏。在教学的过程中，我一方面对自己学习成果进行检验，一方面温故而知新。实际上，在这个过程中，这些小学员确实让我对之前学习的内容的理解更加深刻了，这里要对沈加明、朱明帅、邵盈涵这几位同学表示感谢。在教学过程中，我还和他们一起在 codecombat 上应用 JavaScript 来用代码完成游戏的闯关，这个经历也让我意识到现在学习编写代码也可以是一件很有意思的事。

教学过程中的笔记，或者说备课写的文档，我在整理之后作为稿件投给了《无线电》杂志，最后也是杂志的周明编辑鼓励我将这些内容再整理成图书出版的。

对于我来讲，学习 JavaScript 还有一个好处就是让我的电子作品离 IoT（物联网）更近了。在编写《Intel Edison 智能硬件开发指南——基于 Yocto Project》以及翻译《Raspberry Pi 编程指南——基于 Python 的游戏编程与机器人制作（第 2 版）》时，介绍到最后的网络应用时，我总有一种心有余而力不足的感觉，因为当赋予电子产品网购功能之后，应该能实现很多有意思的想法，但在网页交互方面，我只有一些 HTML 开发的经验，



这就让我完成的内容只能是一个功能上的简单说明。而了解了 JavaScript 之后，我能够在显示的页面上实现很多动态效果，同时，我还发现目前有很多的硬件也能够用 JavaScript 来进行编程，HTML5 以及微信小程序的新闻更激发了我学习 JavaScript 的热情。

本书面向的读者就是那些像我一样想了解 JavaScript 的人，不管有没有软硬件开发的基础，本书都能够让你对 JavaScript 有一个直观的了解。最后要感谢现在正捧着这本书的您，感谢您肯花费时间和精力阅读本书，由于本人水平有限，书中难免存在疏漏与错误，诚恳地希望您批评指正，您的意见和建议将是我巨大的财富。

程晨

2017.5.8

CONTENTS

目录

01	认识 JavaScript	1
1.1	缘起	1
1.2	历史	1
1.3	特性	2
02	类型和变量	4
2.1	浏览器中的控制台	4
2.2	数字和运算符	5
2.3	变量	6
2.4	变量命名规则	7
2.5	字符串	8
2.5.1	连接字符串	9
2.5.2	获取字符串的长度	10
2.5.3	获取单个字符	10
2.5.4	截取字符串	11
2.5.5	大小写转化	11
2.6	逻辑运算	12
2.6.1	布尔类型	12
2.6.2	逻辑运算符	12
2.6.3	关系运算符	14
2.7	undefined 和 null	16
2.8	原始类型	16
2.8.1	鉴别原始类型	16



2.8.2 原始方法	17
03 数组	18
3.1 创建数组	18
3.2 访问数组元素	18
3.3 修改数组元素	19
3.4 应用数组	20
3.4.1 查看数组长度	20
3.4.2 添加数组元素	20
3.4.3 删除数组元素	21
3.4.4 数组相加	22
3.4.5 查找元素的序号	22
3.4.6 将数组转换成字符串	23
04 对话框	24
4.1 HTML 基础	24
4.2 嵌入 JavaScript 代码	25
4.3 对话框	27
4.3.1 输入对话框	27
4.3.2 确定对话框	29
4.3.3 if 语句	31
4.3.4 警告对话框	32
4.4 DOM	33
4.4.1 用 id 标识元素	33
4.4.2 选中元素	34
4.4.3 替换文本内容	34
05 绘制国际象棋棋盘	37
5.1 canvas 元素	37
5.2 绘制方块	38
5.3 绘制方框	39
5.4 绘制棋盘	41
5.5 循环结构	42
5.5.1 while 语句	42
5.5.2 do-while 语句	43

5.5.3 for 循环	44
06 五子棋	48
6.1 基本绘图方式	48
6.1.1 绘制线条	48
6.1.2 填充路径	49
6.1.3 绘制圆弧和圆	50
6.2 绘制棋盘	51
6.3 鼠标交互	52
6.4 绘制棋子	54
6.5 黑子和白子	56
6.6 获胜判断	59
6.7 函数	63
6.7.1 函数的形式	63
6.7.2 落子函数	64
6.7.3 获胜检测函数	65
07 弹球游戏	69
7.1 对象	69
7.2 对象的方法	70
7.3 定时功能	71
7.4 边界判断	72
7.5 增加挡板	73
7.6 键盘交互	74
08 贪吃蛇	78
8.1 构造函数	78
8.2 原型方法	78
8.3 贪吃蛇游戏的框架	79
8.4 贪吃蛇的移动	80
8.5 贪吃蛇的控制	82
8.6 添加苹果	83
8.7 吃苹果	84
8.8 生成新苹果	86
8.9 显示分数	87

8.9.1	设置文本基线	90
8.9.2	设置文本的大小和字体	91
8.10	游戏结束	93
8.11	碰撞检测	93
09	拼图游戏	100
9.1	显示图片	100
9.2	图形的变化	102
9.3	拼图游戏说明	104
9.4	显示局部图片	105
9.5	图片编号	107
9.6	移动拼图	108
9.7	打乱拼图	112
9.8	拼图状态判断	115
10	飞行射击游戏	121
10.1	图片处理	121
10.2	绘制飞机	122
10.3	子弹射击	123
11	游戏引擎	128
11.1	什么是游戏引擎	128
11.2	Quintus 游戏引擎	128
11.3	网页初始化	129
11.4	场景和精灵	130
11.5	精灵控制	132
11.6	调试功能	133
11.7	GMP 游戏引擎	135
11.8	js 文件下载	136
11.9	真·射击游戏	137
11.10	游戏优化	144



认识 JavaScript

1.1 缘起

前一段时间微信爆出“小程序”的消息让很多人开始关注 JavaScript，本人也是其中之一。很多人都说“小程序”会让 JavaScript 工程师越来越受到人们的欢迎，而对于我来说，这正是一个很好的学习理由。

我对 JavaScript 的学习意愿是在制作基于 OpenWRT 的无线监控小车时产生的，当时小车的监控画面是通过路由器传输的，需要在网页中插入一段脚本，因为能力问题，我没有选择 JavaScript，而是采用了 Lua。

后来由于时间问题，我也就一直没能认真学习 JavaScript，不过在我的心中一直在惦记这件事。2015 年，我们创办了一个叫“少年创学院”的创客培训机构，主要是在编程、电子制作、机械结构搭建等方面培养孩子的创新意识和科技素养，当时就有很多家长过来咨询我们有没有 JavaScript 的课程，当时因为精力有限，我们没有开发相应的课程。现在随着“小程序”的出现，咨询的人越来越多了，我觉得这是将这件事落实下来的一个很好契机，所以就在 2016 年底开了一个 JavaScript 的培训班，通过边学边教的过程迫使自己每周抽出一定时间来学习 JavaScript。而所有教学中整理下来的笔记和资料就形成了您手中的这本书。

1.2 历史

JavaScript 是一种高级编程语言，是一种解释型的直译语言。它已经由 ECMA（欧洲计算机制造商协会）通过 ECMAScript 实现语言的标准化。

在计算机领域，编程语言很多，JavaScript 并不是唯一的一种，甚至不是最高效的一种，不过学习 JavaScript 最大的优势是我们不需要安装任何额外的开发软件（不过安装一个文本编辑软件能有效地提高编程效率），只要你的计算机上有浏览器，就可以完成 JavaScript 的程序编写。

JavaScript 被世界上绝大多数的浏览器，包括 Chrome、IE、FireFox、Safari、Opera 支持。它是一种多范式的语言，支持面向对象编程、命令式编程以及函数式编程。它提供语法来操控文本、数组、日期以及正则表达式等，不支持 I/O，比如网络、存储和图形等，但这些都可以通过它的宿主环境提供支持。

JavaScript 的诞生最早应该追溯到 1995 年网景 (Netscape) 公司的脚本语言 LiveScript, 当时这个语言是为 Netscape Navigator 2.0 浏览器开发的。后来网景公司与 Sun 公司组成的开发联盟为了让这门语言搭上 Java 这个编程语言“热词”, 将其临时改名为“JavaScript”。

JavaScript 推出后在浏览器上大获成功, 于是微软公司接着为 Internet Explorer 3.0 浏览器推出了 JScript, 希望能够与处于市场领导地位的网景产品同台竞争。JScript 也是一种 JavaScript 实现, 这两个 JavaScript 语言版本在浏览器端共存意味着语言标准化的缺失, 这门语言的标准化也就被提上了日程。

1997 年, 由网景、Sun、微软、宝蓝等公司组织及个人组成的技术委员会在 ECMA 确定定义了一种名叫 ECMAScript 的新脚本语言标准, 规范名为 ECMA-262。JavaScript 即成为 ECMAScript 的实现之一。

一个完整的 JavaScript 实现应该包含 3 个部分, 即 ECMAScript (语言核心)、DOM (文档对象模型) 和 BOM (浏览器对象模型)。

虽然 JavaScript 与 Java 这门语言不管是在名字上, 还是在语法上都有很多相似性, 但这两门编程语言从设计之初就有很大的不同。JavaScript 的语言设计主要受到了 Self (一种基于原型的编程语言) 和 Scheme (一种函数式编程语言) 的影响, 在语法结构上它又与 C 语言有很多相似性 (例如 if 条件语句、while 循环、switch 语句、do-while 循环等)。

1.3 特性

不同于服务器端脚本语言, JavaScript 主要被作为客户端脚本语言, 在用户的浏览器上运行, 不需要服务器的支持。所以在早期程序员比较青睐于使用 JavaScript 以减少对服务器的负担, 而与此同时也带来了安全性的问题。随着服务器的强壮, 虽然现在的程序员更喜欢运行于服务端的脚本以保证安全, 但 JavaScript 仍然以其跨平台、容易上手等优势得以大行其道。同时, 有些特殊功能 (如 AJAX) 必须依赖 JavaScript 在客户端进行支持。随着引擎 (如 V8) 和框架 (如 Node.js) 的发展, 及其事件驱动、异步 I/O 等特性, JavaScript 逐渐被用来编写服务器端程序。在近几年中, Node.js 的出世让 JavaScript 也具有了一定的服务器功能, 而且在某些方面比 PHP 的效果更为显著。

JavaScript 的基本特点如下。

- 它是一种解释性脚本语言, 即代码不需要进行预编译。
- 它主要用来向 HTML 页面添加交互行为。
- 它可以直接嵌入 HTML 页面, 但写成单独的 js 文件有利于结构和行为的分离。

JavaScript 常用来完成以下任务。

- 在 HTML 页面中嵌入动态文本。
- 对浏览器事件做出响应。

- 读写 HTML 元素。
- 在数据被提交到服务器之前验证数据。
- 控制 cookies，包括创建和修改等。

每一个浏览器都带有一个 JavaScript 解释器，使得我们的浏览器能够直接运行 JavaScript 程序。其实在生活中，很多网站都嵌入了 JavaScript 代码，而我们在浏览这些网站时不用安装额外的软件就能看到程序运行的效果。

02

类型和变量

2.1 浏览器中的控制台

前面说过，我们要学习 JavaScript，不用安装其他软件，使用浏览器就可以了，下面我们就利用浏览器来学习一下 JavaScript 中的基本概念。

说明：本书使用的是 Google 的 Chrome 浏览器，操作系统是 Windows。

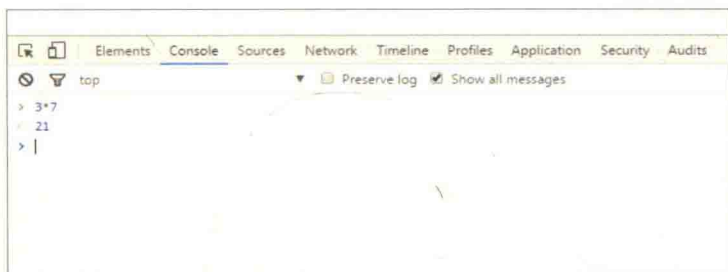
打开 Chrome 浏览器，按下 Ctrl+Shift+j 或 F12 来打开浏览器的开发人员工具，这是程序员对网页进行测试的一个窗口，如图 2.1 所示。



■ 图 2.1 Chrome 中的调试窗口

调试工作主要是在控制台 (Console) 完成的，如果你的窗口中选中的不是这个标签项，那就要用鼠标单击这个选项卡切换到控制台，然后在下方空白区域的左上方应该有一个向右的蓝色箭头，箭头后面就是我们可以简单地编写 JavaScript 代码的位置。

我们可以尝试在这里进行简单的数学运算，输入“3*7”，则会在下方显示出 JavaScript 解释器输出的结构，从显示上能够看到输出的内容前面是一个向左的箭头。这里在箭头后面是 3 乘以 7 的运算结果 21，如图 2.2 所示。



■ 图 2.2 简单的运算

2.2 数字和运算符

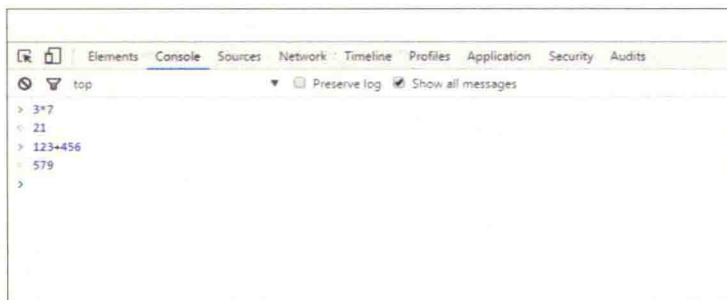
在执行完简单的运算之后，我再来说说 JavaScript 中的数字和运算符。

JavaScript 中有 3 种基本的数据类型：数字类型、字符串类型和布尔类型。

我们之前做乘法运算时输入的数字就属于数字类型，生活中很多的内容都可以通过数字来表示，比如物体的长宽高、我们的年龄和体重等。

数字的基本运算除了之前用到的乘“*”之外，还有加、减和除，对应的操作符分别是+、-、/。另外在计算机领域还有一个取余的运算符“%”，通过这个运算符，我们能够得到一个数整除另外一个数之后余下的值。

我们先在控制台中做一个加法运算。输入“123+456”，回车后就能得到运算的结果，如图 2.3 所示。



■ 图 2.3 加法运算

说明：考虑到篇幅和显示的问题，之后在控制台中显示的内容，我们直接列出对应的文本内容，不再以图片的形式显示。

接着我们再完成减法和除法的运算。

```
>89-23
<66
```

```
>45/4
<11.25
```

说明：在上面列出的文本内容中，前面的符号“>”和“<”都是浏览器的控制台中自动输出的，不是我们输入的内容。实际上只有符号“>”后面的内容才是我们输入的。

JavaScript 中是不区分数字是整型还是浮点型的，所以我们在进行除法运算时，如果除不尽，就会得到一个小数。

我们再来看看取余的操作。如果 10 整除 5，因为 10 是 5 的整数倍，所以余数为 0；而如果 11 整除 5，则会剩下余数 1；而如果 12 整除 5，则会剩下余数 2。对应的运算如下。

```
>10%5
<0
>11%5
<1
>12%5
<2
```

最后，我们再来完成一个复杂的混合运算。

```
>12*4+56/7
<56
```

在 JavaScript 中也是遵循先乘除后加减的运算法则的。比如上面的内容，是先运算 12 乘以 4 和 56 除以 7，分别得到 48 和 8，然后再将 48 和 8 加起来得到最后的 56。如果我们希望能先运算加减法，就需要使用小括号。假设这里先运算 4 加 56，则内容就要更改为以下的样子。

```
>12*(4+56)/7
<102.85714285714286
```

这样 JavaScript 就会先计算 4 加 56 得到 60，然后 60 乘以 12 得到 720，最后 720 除以 7 即得到最终的结果。

2.3 变量

在代码中，我们通常会给一个具体表示一定意义的数据起个名字，这就是变量。变量的值会根据程序的运行不断变化。我们可以把变量想象成一个有名字的盒子，盒子里放着对应的数据，程序可以不断地将新的数据放进去，也可以只是看看盒子里的数据是多少。不过放入数据的情况还需要说明一下，当我们放入新数据时，旧的数据就被覆盖了，所以盒子里始

终只有一个数据。

在 JavaScript 中创建一个变量要用到关键字 `var`，关键字之后就是变量的名称。关键字是编程语言保留的特殊标识符，它们具有固定的名称和含义。假设我们输入关键字 `var`，当解释器看到 `var` 时，就知道后面跟着的是变量的名字。

现在我们要定义一个变量 `name`，则在控制台中输入和反馈的信息如下。

```
> var name  
< undefined
```

这里要注意，因为 `var` 是关键字，所以在控制台中它的颜色是紫色。另外，此时的反馈值是 `undefined`，这不是一个错误，只是表示定义变量的这条语句没有返回值。在之前我们进行运算时，其返回值就是运算结果，而在创建变量时是没有返回值的，所以解释器输出 `undefined`。

给变量这个盒子中放入数据的过程叫赋值，要给变量赋值需要使用等号，示例如下。

```
> var age = 36;  
< undefined
```

这里我们定义了一个变量，变量名为 `age`，同时使用等号将变量赋值为 36。这条语句是新建一个变量，所以依然没有返回值。

定义了变量之后，就能在控制台中使用它了。现在我们在控制台中输入变量名，控制台就会返回这个变量的值。另外，我们还能将变量直接代入运算当中。示例如下。

```
> age  
< 36  
> age + 12  
< 48
```

定义变量只是定义了一个存放数据的盒子，盒子中的数据是通过赋值完成的，这个操作可以反复进行，假设现在我们希望改变变量的值，则只需要再次使用等号，如下所示。

```
> age = 24;  
< 24
```

这次没有使用关键字 `var`，这是因为现在这个变量已经存在了，我们只是重新赋值而已，而这次的操作是有返回值的，返回值就是变量新的数据值。

2.4 变量命名规则

在程序中，变量的名字必须前后一致，对于程序来说，哪怕是一个字母的大小写不同，JavaScript 也会认为这是两个不同的变量。如果一个变量你没有定义，那么解释器就会返