

Grinstead, Rob Sutherland, Sergi Meseguer, Phillip Gourley, Mark Watson, Jeff Carouth, Alfredo Sumaran, Martin Sachse, Marcio Barrios, Dan, AimelyneM, Matt Sullivan, Delnatte Pierre-Antoin, Jake Smith, Eugen Tudorancea, Iris, David Trinh, simonstl, Ray Daly, Uros Gruber, Justin Myers, Shai Zonis, Mom & Dad, Devin Clark, Dennis Palmer, Brian Panahi Johnson, Josh Marshall, Marshall, Dennis Kerr, Matt Steele, Erik Slagter, Sacah, Justin Rainbow, Christian Nilsson, Delapouite, D.Pereira, Nicolas Hoizey, George V. Reilly, Dan Reeves, Bruno Laturner, Chad Jennings, Shane King, Jeremiah Lee Cohick, od3n, Stan Yamane, Marko Vucinic, Jim B, Stephen Collins, Egir Porsteinsson, Eric Pederson, Owain, Nathan Smith, Jeanetteurphy, Alexandre ELISé, Chris Peterson, Rik Watson, Luke Matthews, Justin Lowery, Morten Nielsen, Vernon Kesner, Chetan Shenoy, Paul Tregoin, Marc Grabanski, Dion Almaer, Andrew Sullivan, Keith Elsass, Tom Burke, Brian Ashenfelter, David Stuart, Karl Swedberg, Graeme, Brandon Hays, John Christopher, Gior, manoj reddy, Chad Smith, Jared Harbour, Minoru TODA, Chris Wigley, Daniel Mee, Mike, Handyface, Alex Jahraus, Carl Furrow, Rob Foulkrod, Max Shishkin, Leigh Penny Jr., Robert Ferguson, Mike van Hoenselaar, Hasse Schougaard, rajan venkataguru, Jeff Adams, Trae Robbins, Rolf Langenhuijzen, Jorge Antunes, Alex Koloskov, Hugh Greenish, Tim Jones, Jose Ochoa, Michael Brennan-White, Naga Harish Muvva, Barkóczy Dávid, Kitt Hodsdon, Paul McGraw, Sascha Goldhofer, Andrew Metcalf, Markus Krogh, Michael Mathews, Matt Jared, Juanfran, Georgie Kirschner, Kenny Lee, Ted Zhang, Amit Pahwa, Inbal Sinai, Dan Raine, Schabse Laks, Michael Tervoort, Alexandre Abreu, Alan Joseph Williams, NicolasD, Cindy Wong, Reg Braithwaite, LocalPCGuy, Jon Friskics, Chris Merriman, John Pena, Jacob Katz, Sue Lockwood, Magnus Johansson, Jeremy Crapsey, Grzegorz Pawlowski, nico nuzzaci, Christine Wilks, Hans Bergren, charles montgomery, Ariel-Fogel, Ivan Kolev, Daniel Campos, Hugh Wood, Christian Bradford, Frédéric Harper, Ionut Dan Popa, Jeff Trimble, Rupert Wood, Trey Carrico, Pancho Lopez, Joel kuyten, Tom A Marra, Jeff Jewiss, Jacob Rios, Paolo Di Stefano, Soledad Penades, Chris Gerber, Andrey Dolganov, Wil Moore III, Thomas Martineau, Kareem, Ben Thouret, Udi Nir, Morgan Laupies, jory carson-burson, Nathan L Smith, Eric Damon Walters, Derry Lozano-Hoyland, Geoffrey Wiseman, mkeehner, KatieK, Scott MacFarlane, Brian LaShomb, Adrien Mas, christopher ross, Ian Littman, Dan Atkinson, Elliot Jobe, Nick Dozier, Peter Wooley, John Hoover, dan, Martin A. Jackson, Héctor Fernando Hurtado, andy ennamorato, Paul Seltmann, Melissa Gore, Dave Pollard, Jack Smith, Philip Da Silva, Guy Israeli, @megalithic, Damian Crawford, Felix Gliesche, April Carter Grant, Heidi, jim tierney, Andrea Giammarchi, Nico Vignola, Don Jones, Chris Hartjes, Alex Howes, john gibbon, David J. Groom, BBox, Yu Dilys Sun, Nate Steiner, Brandon Satrom, Brian Wyant, Wesley Hales, Ian Pouncey, Timothy Kevin Oxley, George Terezakis, sanjay raj, Jordan Harband, Marko McLion, Wolfgang Kaufmann, Pascal

Peuckert, Dave Nugent, Markus Liebelt, Welling Guzman, Nick Cooley, Daniel Mesquita, Robert Syvarth, Chris Coyier, Rémy Bach, Adam Dougal, Alistair Duggin, David Loidolt, Ed Richer, Brian Chenault, GoldFire Studios, Carles Andrés, Carlos Cabo, Yuya Saito, roberto ricardo, Barnett Klane, Mike Moore, Kevin Marx, Justin Love, Joe Taylor, Paul Dijou, Michael Kohler, Rob Cassie, Mike Tierney, Cody Leroy Lindley, tofuji, Shimon Schwartz, Raymond, Luc De Brouwer, David Hayes, Rhys Brett-Bowen, Dmitry, Aziz Khoury, Dean, Scott Tolinski-Level Up, Clement Boirie, Djordje Lukic, Anton Kotenko, Rafael Corral, Philip Hurwitz, Jonathan Pidgeon, Jason Campbell, Joseph C., SwiftOne, Jan Hohner, Derick Bailey, getify, Daniel Cousineau, Chris Charlton, Eric Turner, David Turner, Joel Galeran, Dharma Vagabond, adam, Dirk van Bergen, dave ♡🎵★ furf, Vedran Zakanj, Ryan McAllen, Natalie Patrice Tucker, Eric J. Bivona, Adam Spooner, Aaron Cavano, Kelly Packer, Eric J, Martin Drenovac, Emilis, Michael Pelikan, Scott F. Walter, Josh Freeman, Brandon Hudgeons, vijay chennupati, Bill Glennon, Robin R., Troy Forster, otaku_coder, Brad, Scott, Frederick Ostrander, Adam Brill, Seb Flippence, Michael Anderson, Jacob, Adam Randlett, Standard, Joshua Clanton, Sebastian Kouba, Chris Deck, SwordFire, Hannes Papenberg, Richard Woeber, hnzz, Rob Crowther, Jedidiah Broadbent, Sergey Chernyshev, Jay-Ar Jamon, Ben Combee, luciano bonachela, Mark Tomlinson, Kit Cambridge, Michael Melgares, Jacob Adams, Adrian Bruinhout, Bev Wieber, Scott Puleo, Thomas Herzog, April Leone, Daniel Mizieliński, Kees van Ginkel, Jon Abrams, Erwin Heiser, Avi Laviad, David newell, Jean-Francois Turcot, Niko Roberts, Erik Dana, Charles Neill, Aaron Holmes, Grzegorz Ziolkowski, Nathan Youngman, Timothy, Jacob Mather, Michael Allan, Mohit Seth, Ryan Ewing, Benjamin Van Treese, Marcelo Santos, Denis Wolf, Phil Keys, Chris Yung, Timo Tijhof, Martin Lekvall, Agendine, Greg Whitworth, Helen Humphrey, Dougal Campbell, Johannes Harth, Bruno Girin, Brian Hough, Darren Newton, Craig McPheat, Olivier Tille, Dennis Roethig, Mathias Bynens, Brendan Stromberger, sundeep, John Meyer, Ron Male, John F Croston III, gigante, Carl Bergenheim, B.J. May, Rebekah Tyler, Ted Foxberry, Jordan Reese, Terry Suitor, afeliz, Tom Kiefer, Darragh Duffy, Kevin Vanderbeken, Andy Pearson, Simon Mac Donald, Abid Din, Chris Joel, Tomas Theunissen, David Dick, Paul Grock, Brandon Wood, John Weis, dgrebb, Nick Jenkins, Chuck Lane, Johnny Megahan, marzsmn, Tatu Tamminen, Geoffrey Knauth, Alexander Tarmolov, Jeremy Tymes, Chad Auld, Sean Parmelee, Rob Staenke, Dan Bender, Yannick derwa, Joshua Jones, Geert Plaisier, Tom LeZotte, Christen Simpson, Stefan Bruvik, Justin Falcone, Carlos Santana, Michael Weiss, Pablo Villoslada, Peter deHaan, Dimitris Iliopoulos, seyDoggy, Adam Jordens, Noah Kantrowitz, Amol M, Matthew Winnard, Dirk Ginader, Phinam Bui, David Rapson, Andrew Baxter, Florian Bougel, Michael George, Alban Escalier, Daniel Sellers, Sasha Rudan, John Green, Robert Kowalski, David I. Teixeira (@ditma), Charles Carpenter,

Justin Yost、Sam S、Denis Ciccale、Kevin Sheurs、Yannick Croissant、Pau Fracés、Stephen McGowan、Shawn Searcy、Chris Ruppel、Kevin Lamping、Jessica Campbell、Christopher Schmitt、Sablons、Jonathan Reisdorf、Bunni Gek、Teddy Huff、Michael Mullany、Michael Fürstenberg、Carl Henderson、Rick Yoesting、Scott Nichols、Hernán Ciudad、Andrew Maier、Mike Stapp、Jesse Shawl、Sérgio Lopes、jsulak、Shawn Price、Joel Clermont、Chris Ridmann、Sean Timm、Jason Finch、Aiden Montgomery、Elijah Manor、Derek Gathright、Jesse Harlin、Dillon Curry、Courtney Myers、Diego Cadenas、Arne de Bree、Joao Paulo Dubas、James Taylor、Philipp Kraeutli、Mihai Paun、Sam Gharegozlou、joshjs、Matt Murchison、Eric Windham、Timo Behrmann、Andrew Hall、joshua price、Théophile Villard

这套系列丛书的写作、编辑和制作都是以开源的方式进行的。我要感谢 GitHub 让这一切成为可能！

再次向我没能提及的支持者们表示感谢。这套系列丛书属于我们每一个人，希望它能够帮助更多的人更好地了解 JavaScript，让当前和未来的社区贡献者受益。

第一部分

起步上路

序

你最近新学的技能是什么？

可能是一门外语，比如意大利语或德语。可能是一个图像编辑工具，如 Photoshop。也可能是某种厨艺或者木工活，又或是某种健身项目。请回忆一下你最终掌握这项技能的那个时刻，那应该就是你突然顿悟的一刻。当你学会操控台锯或者理解了法语中阳性名词和阴性名词之间的区别时，事情就从模糊变得明朗清晰起来。那时感觉如何？是不是非常不可思议？

现在，再往前回忆一下你掌握这项新技能之前的情形。那时你是什么感觉呢？可能有点恐惧又有点心慌，是不是？在某一刻，我们还不了解自己现在已经掌握的知识，这完全没有任何问题；每个人都是从某个起点开始学习的。学习新技能是一场令人激动的探险，特别是当你想要高效学习某个主题时。

我教授过很多初级的编程课程。跟着我学习的学生通常之前已经试着通过博客或复制、粘贴代码来自学过 HTML 或者 JavaScript 这样的主题，但是他们并没有能够真正掌握编码知识来实现想要的目标。而且，由于并没有真正掌握某些编程主题的细节，他们便无法编写功能强大的代码或调试自己的作品，因为他们并没有真正理解所发生的一切。

我一直坚信要以正确的方法授课，这意味着我会讲授 Web 标准、语义标记、注释良好的代码以及其他的最佳实践。我会详细讲解涉及的主题，解释如何做以及这么做的原因，而不仅仅是扔出代码以供复制、粘贴。努力理解自己的代码后，你就可以更好地完成任务，同时自己也会得到进步。此时代码不再仅仅是你的工作，更是你的作品。这就是我喜欢本书内容的原因。Kyle 带领我们深入了解语法和术语，对 JavaScript 这个语言进行了出色、全面的介绍。本书并没有流于表面，而是确实有助于我们真正理解概念。

就像只学会如何在 Photoshop 中打开、关闭和保存文档是不够的，只能够将 jQuery 代码片段复制到你的网站上也是不够的。的确，只需要学习一些与编程相关的基础知识，我就能

够编写并共享自己的设计。但如果没有对工具及其背后原理的正确理解，我怎么能够定义一个网格或创建一个明晰的类型系统呢？又如何才能优化 Web 上的图像呢？对 JavaScript 来说也是一样的。如果不清楚循环的工作模式、变量的定义以及作用域的含义，那么我们就无法编写自己所能实现的最佳代码。我们不能接受退而求其次的作品，毕竟这是我们自己的作品。

你对 JavaScript 的探索越深入，它就会变得越清晰。或许你对闭包、对象和方法这样的词汇目前来说还不是很熟悉，但本书将帮助你明晰这些术语。我希望你在开始学习本书时记住自己学习某样东西前后的感受。这可能很艰巨，但为了要开始一段磨砺你的知识宝剑的精彩之旅，你已经翻阅至此。本书第一部分是我们理解编程的起点。享受你顿悟的时刻吧！

——Jenn Lukas (<http://jennlukas.com>, @jennlukas), 前端顾问

第 1 章

深入编程

欢迎来到“你不知道的 JavaScript”系列。

本部分介绍了编程中的一系列基本概念，并有助于你更好地理解本系列其余几本书的内容。当然，本部分的内容会更偏向于 JavaScript（常简写为 JS）。如果你是刚开始学习编程或 JavaScript，那么本部分将会简单探讨你起步和继续学习所需要了解的概念。

本部分一开始会从很高的层次来介绍编程的基本原则。基本上假设你在阅读“你不知道的 JavaScript”系列图书时几乎没有编程经历，并想要通过学习这几本书从 JavaScript 这个视角开始理解编程。

第 1 章总结了深入学习和实践编程所需要的知识，此外还有许多其他介绍编程的优秀资源，这些资源可以帮助你更深入地探索这些主题。除了第 1 章的知识，我建议你还要利用这些资料来更深入地学习。

在熟悉了常用的基础编程概念后，第 2 章将帮助你熟悉 JavaScript 的编程风格。第 2 章介绍了 JavaScript 的含义，但再次声明，这并不是完整的指南——这是“你不知道的 JavaScript”系列其余几本图书的主旨！

如果你已经对 JavaScript 有了一定的了解，那么可以先查看第 3 章来了解“你不知道的 JavaScript”系列图书的简介，然后直接阅读你所感兴趣的章节！

1.1 代码

让我们从头开始。

程序常被称为**源码或代码**，它是一组特定的指令，用来指示计算机要执行哪些任务。虽然对 JavaScript 来说可以直接在浏览器的开发者终端中输入代码，但代码通常会被保存在文本文件中，我们将在后文中对此进行简单介绍。

指令的格式和组合规则被称为**计算机语言**，有时也被称为**语法**，这非常类似于英语中告诉你如何拼写单词以及如何使用单词和标点符号来构造有效的句子。

语句

在计算机语言中，执行特定任务的一组单词、数字和运算符被称为**语句**。在 JavaScript 中，一条语句可能如下所示：

```
a = b * 2;
```

其中的字符 **a** 和 **b** 称为**变量**（参见 1.7 节），它们就好比是可以存放东西的小盒子。在程序中，变量保存程序要使用的值（比如数字 42）。你可以将它们想象成值本身的替代符。

相比之下，**2** 本身就是一个值，称为**字面值**，因为它独立存在而没有保存在变量之中。

其中的字符 **=** 和 ***** 是**运算符**（参见 1.4 节），它们对值和变量执行动作，如赋值和进行乘法运算。

JavaScript 的多数语句都是以分号（**;**）结尾的。

粗略地说，语句 **a = b * 2;** 告诉计算机获取变量 **b** 的当前值，然后将这个值乘以 2，再将计算结果保存到另一个名为 **a** 的变量中。

程序就是多个这样语句的集合，它们合起来描述了程序要执行的所有步骤。

1.2 表达式

语句由一个或多个**表达式**组成。一个表达式是对一个变量或值的引用，或者是一组值和变量与运算符的组合。

举例来说，**a = b * 2;** 这个语句中有四个表达式。

- **2** 是一个**字面值表达式**。
- **b** 是一个**变量表达式**，表示获取它的当前值。
- **b * 2** 是一个**算术表达式**，表示进行乘法运算。
- **a = b * 2** 是一个**赋值表达式**，意思是将表达式 **b * 2** 的结果赋值给变量 **a**（我们将在后文中深入介绍赋值）。

一个独立的表达式也可以称为**表达式语句**，如下所示：


```
b * 2;
```

这种表达式语句不是很常用，或者说不是很有用，因为它通常不会对程序的运行起到任何作用，它只是取得 `b` 的值并乘以 2，但是却没有对结果有任何影响。

更常用的表达式语句是**调用表达式语句**（参见 1.11 节），因为整个语句本身就是一个函数调用表达式：

```
alert( a );
```

执行程序

这些编程语句的集合是如何通知计算机来执行任务的呢？程序需要**被执行**，我们也将这一过程称为**运行程序**。

`a = b * 2` 这样的语句便于开发者读写，但实际上计算机并不能直接理解这种形式。因此，需要通过计算机上一个专门的工具（**解释器或编译器**）将你编写的代码翻译成计算机可以理解的命令。

对某些计算机语言来说，在程序被执行时，对命令的翻译通常是自上而下逐行执行的，这通常被称为**代码解释**。

对另外一些语言来说，这种翻译是预先进行的，被称为**代码编译**，这样一来，当**执行程序**时，实际上运行的是已经编译好的、可以执行的计算机指令。

基本上可以说 JavaScript 是**解释型**的，因为每次执行 JavaScript 源码时都需要进行处理。但这么说并不完全精确。JavaScript 引擎实际上是**动态编译**程序，然后立即执行编译后的代码。



有关 JavaScript 编译的更多信息，参见本系列《你不知道的 JavaScript（上卷）》¹ 第一部分中的前两章。

1.3 实践

本章将通过简单的代码片段来介绍每个编程概念，这些代码（当然）是用 JavaScript 编写的。

非常重要的一点是，在阅读本章时，你应该通过亲自编写代码来实践每个概念，并且你可能需要花一点时间反复阅读本章。最简单的方法是，使用最方便的浏览器（Firefox、Chrome、IE 等）的开发者工具来实践。

注 1：此书已由人民邮电出版社出版。——编者注



一般来说，你可以通过菜单项或者快捷键来打开开发者终端。有关在你喜欢的浏览器中打开和使用终端的更多详细信息，参见“掌握开发者工具终端”(<http://blog.teamtreehouse.com/mastering-developer-tools-console>)。

如果要在终端中一次输入多行，那么可以使用 `<shift>+<enter>` 组合键来另起一行。一旦点击 `<enter>` 键，终端会立即执行已输入的所有代码。

我们来熟悉一下在终端中运行代码的流程。首先，建议你在浏览器中打开一个空白的标签页。我更喜欢在地址栏中输入 `about:blank` 来实现这一点。然后，确保你的开发者终端是开启状态，就像我们之前提到的那样。

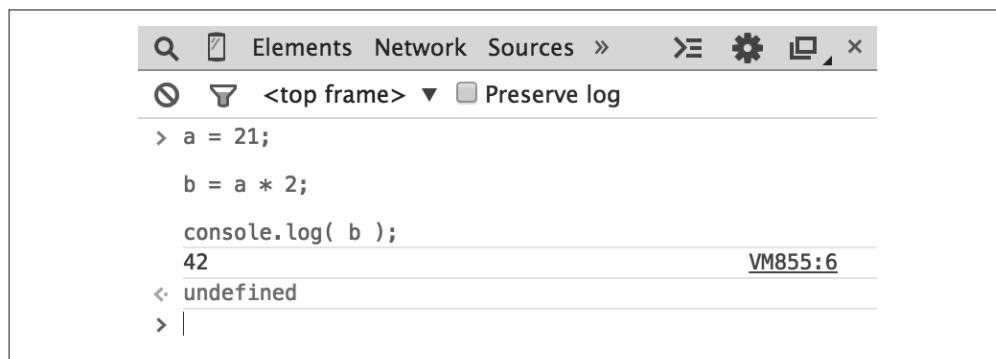
现在，输入如下代码，并观察代码的执行：

```
a = 21;

b = a * 2;

console.log( b );
```

在 Chrome 浏览器的终端中输入前面的代码将会产生如下所示的输出：



你可以自己试一下。学习编程的最好方法就是编写代码！

1.3.1 输出

在前面的代码片段中，我们使用了 `console.log(...)`。现在我们来简单了解一下这行代码做了些什么。

可能你已经猜到了，这就是我们在开发者终端打印文本（即向用户输出）的方法。我们应该对这个语句的两点解释一下。

首先，`log(b)` 这一部分是一个函数调用（参见 1.11 节）。我们将变量 `b` 传给这个函数，请求它将 `b` 的值打印到终端中。