这样除了可以少写一些代码，我认为箭头函数将程序员们经常犯的一个错误给标准化了，也就是混淆了 this 绑定规则和词法作用域规则。

换句话说：为什么要自找麻烦使用 this 风格的代码模式呢？把它和词法作用域结合在一起非常让人头疼。在代码中使用两种风格其中的一种是非常自然的事情，但是不要将两种风格混在一起使用。

另一个导致箭头函数不够理想的原因是它们是匿名而非具名的。具名函数比匿名函数更可取的原因参见第 3 章。

在我看来，解决这个“问题”的另一个更合适的办法是正确使用和包含 this 机制。

```
var obj = {
    count: 0,
    cool: function coolFn() {
        if (this.count < 1) {
            setTimeout( function timer(){
                this.count++; // this 是安全的
                              // 因为 bind(..)
                console.log( "more awesome" );
            }.bind( this ), 100 ); // look, bind()!
        }
    }
};

obj.cool(); // 更酷了。
```

无论你是喜欢箭头函数中 this 词法的新行为模式，还是喜欢更靠得住的 bind()，都需要注意箭头函数不仅仅意味着可以少写代码。

它们之间有意为之的不同行为需要我们理解和掌握，才能正确地使用它们。

现在我们已经完全理解了词法作用域（还有闭包），理解 this 词法是小菜一碟！

附录D

致谢

对于本书和这个系列来说，我有许多想要感谢的人。

首先，我必须感谢我的妻子 Christen Simpson，还要感谢我的两个孩子 Ethan 和 Emily 忍受他们的父亲一直摆弄电脑。即使没有在写书，对于 JavaScript 的痴迷也让我的眼睛一刻不离屏幕，我太不应该这样做了。本书之所以可以非常深入和完整地解释 JavaScript，完全是因为我牺牲了陪伴家人的时间。我亏欠家人的实在太多。

我还要感谢我在 O'Reilly 的编辑 Simon St.Laurent 和 Brian MacDonaldm，以及全体编辑和营销人员。与他们共事真是太棒了，他们在“开源”图书写作、编辑和出版的这次尝试过程中对我无比包容。

感谢在该系列书编写过程提出修改建议和帮忙纠错的朋友，他们的贡献令本书增色不少，这些人包括但不限于 Shelley Powers、Tim Ferro、Evan Borden、Forrest L Norvell、Jennifer Davis、Jesse Harlin。感谢 Shane Hudson 为本书第一部分“作用域和闭包”作序，写得很棒。

感谢社区中的所有人，包括 TC39 委员会的成员，他们分享了许多知识并且一直耐心而且细致地回答我接连不断的问题。这些人包括但不限于 John-David Dalton、Juriy“kangax” Zaytsev、Mathias Bynens、Rick Waldron、Axel Rauschmayer、Nicholas Zakas、Angus Croll、Jordan Harband、Reginald Braithwaite、Dave Herman、Brendan Eich、Allen Wirfs-Brock、Bradley Meck、Domenic Denicola、David Walsh、Tim Disney、Kris Kowal、Peter van der Zee、Andrea Giammarchi、Kit Cambridge 以及其他人，原谅我无法在这里一一列出。

由于本系列脱胎于 Kickstarter，我同样想感谢所有（接近）500 个支持者，没有他们就没有这个系列：Jan Szpila、nokiko、Murali Krishnamoorthy、Ryan Joy、Craig Patchett、pdqtrader、Dale Fukami、ray hatfield、R0drigo Perez [Mx]、Dan Petitt、Jack Franklin、Andrew Berry、Brian Grinstead、Rob Sutherland、Sergi Meseguer、Phillip Gourley、Mark Watson、Jeff Carouth、Alfredo Sumaran、Martin Sachse、Marcio Barrios、Dan、AimelyneM、Matt Sullivan、Delnatte Pierre-Antoine、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、Ægir Þorsteinsson、Eric Pederson、Owain、Nathan Smith、Jeanetteurphy、Alexandre ELISÉ、Chris Peterson、Rik Watson、Luke Matthews、Justin Lowery、Morten Nielsen、Vernon Kesner、Chetan Shenoy、Paul Tregoing、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、Barkóczi Dávid、Kitt Hodsden、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 Pawłowski、nico nuzzaci、Christine Wilks、Hans Bergren、charles montgomery、Ariel בר-לבב Fogel、Ivan Kolev、Daniel Campos、Hugh Wood、Christian Bradford、Frédéric Harper、Ionuț Dan Popa、Jeff Trimble、Rupert Wood、Trey Carrico、Pancho Lopez、Joël kuijten、Tom A Marra、Jeff Jewiss、Jacob Rios、Paolo Di Stefano、Soledad Penades、Chris Gerber、Andrey Dolganov、Wil MooreIII、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、Joël 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 Ziółkowski、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 Bergenhem、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、marzsman、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、João Paulo Dubas、James Taylor、Philipp Kraeutli、Mihai Paun、Sam Gharegozlou、joshjs、Matt Murchison、Eric Windham、Timo Behrmann、Andrew Hall、joshua price、and Théophile Villard。

本系列书的写作是开源的，包括编辑和出版。我们要感谢 GitHub 提供的社区协作功能！

再次感谢，我无法列出所有人的名字，但是非常感谢你们。希望本书可以被所有人“拥有”，能促进大家对于 JavaScript 语言的理解，能帮助现在和未来的社区贡献者们。

第二部分

this和对象原型

[美] Kyle Simpson 著
梁杰 译

序

读这本书准备作序的时候，我不禁想起 15 年前学习 JavaScript 时的情景。过去的这 15 年中，我一直用它进行编程和开发，同时，JavaScript 也在不断发生变化。

15 年前我刚开始使用 JavaScript 时，在网页中使用 CSS 和 JavaScript 等非 HTML 技术被称为 DHTML 或者动态 HTML。在那之后，JavaScript 的用途发生了巨大的变化，印象中主要用于给网页添加动态雪花或者在状态栏中添加动态时钟。说实话，职业生涯的早期我并没有对 JavaScript 给予足够的重视，因为在我看来它主要的功能就是编写一些有趣的小东西。

直到 2005 年我才第一次认识到 JavaScript 是一门真正的编程语言，应当受到我的重视。仔细研究了谷歌地图的第一个测试版本之后，我被它的潜力深深地吸引住了。在那时，谷歌地图是一个史无前例的应用——你可以用鼠标移动和缩放地图，并且可以在不重载页面的情况下发起服务器请求——这些全部用 JavaScript 完成，简直就像魔法一样！

如果某些事情对你来说像魔法一样，那意味着你看到了新生事物的曙光。我的想法是正确的——今天，无论在客户端还是服务端，JavaScript 都已经成为了我的一门主要编程语言，没有其他语言比它更适合完成这些工作。

回顾这 15 年，有一件事我很后悔，那就是没有在 2005 年之前给予 JavaScript 足够的重视。更准确地说，我并没有想到 JavaScript 会像 C++、C#、Java 等语言一样，成为一门非常有用的真正的编程语言。

如果在一开始时就能遇到“你不知道的 JavaScript”系列丛书，我的整个职业生涯都会大不相同。对于这个系列丛书，我非常欣赏的一点是：它可以用有趣并且有效的方式帮助你构建起对于 JavaScript 的理解。

本书第二部分“this 和对象原型”非常不错，它很好地衔接了本书第一部分“作用域和闭包”，进一步介绍了 JavaScript 语言中非常重要的两个部分，this 关键字和原型。这两个部分对于你未来的学习来说非常重要，它们是使用 JavaScript 进行编程的基础。只有掌握了如何创建、关联和扩展对象，你才能用 JavaScript 创建类似谷歌地图这样大型的复杂应用。

在我看来，绝大多数 Web 开发者可能都没有创建过一个 JavaScript 对象，他们只是把 JavaScript 当作按钮和 AJAX 请求之间的事件绑定粘合剂。其实我也曾经是他们中的一员，但是当我学会在 JavaScript 中使用原型和创建对象之后，整个世界都变样了。如果你也只会编写事件绑定代码，那么这本书是非读不可的；如果你只想复习一下的话，这本书也一定是首选。无论如何，你绝对不会失望的，相信我！

——Nick Berardi
nickberardi.com
Twitter：@nberardi

第1章

关于this

this 关键字是 JavaScript 中最复杂的机制之一。它是一个很特别的关键字，被自动定义在所有函数的作用域中。但是即使是非常有经验的 JavaScript 开发者也很难说清它到底指向什么。

> 任何足够先进的技术都和魔法无异。
>
> ——Arthur C. Clarke

实际上，JavaScript 中 this 的机制并没有那么先进，但是开发者往往会把理解过程复杂化，毫无疑问，在缺乏清晰认识的情况下，this 对你来说完全就是一种魔法。

“this”是沟通过程中极其常见的一个代词。所以，在交流过程中很难区分我们到底把“this”当作代词还是当作关键字。清晰起见，我总一直使用 this 表示关键字，使用“this”或者 this 来表示代词。

1.1 为什么要用this

如果对于有经验的 JavaScript 开发者来说 this 都是一种非常复杂的机制，那它到底有用在哪里呢？真的值得我们付出这么大的代价学习吗？的确，在介绍怎么做之前我们需要先明白为什么。

下面我们来解释一下为什么要使用 this：

```
function identify() {
    return this.name.toUpperCase();
}

function speak() {
    var greeting = "Hello, I'm " + identify.call( this );
    console.log( greeting );
}

var me = {
    name: "Kyle"
};

var you = {
    name: "Reader"
};

identify.call( me ); // KYLE
identify.call( you ); // READER

speak.call( me ); // Hello, 我是 KYLE
speak.call( you ); // Hello, 我是 READER
```

看不懂这段代码？不用担心！我们很快就会讲解。现在请暂时抛开这些问题，专注于为什么。

这段代码可以在不同的上下文对象（me 和 you）中重复使用函数 identify() 和 speak()，不用针对每个对象编写不同版本的函数。

如果不使用 this，那就需要给 identify() 和 speak() 显式传入一个上下文对象。

```
function identify(context) {
    return context.name.toUpperCase();
}

function speak(context) {
    var greeting = "Hello, I'm " + identify( context );
    console.log( greeting );
}

identify( you ); // READER
speak( me ); //hello, 我是 KYLE
```

然而，this 提供了一种更优雅的方式来隐式“传递”一个对象引用，因此可以将 API 设计得更加简洁并且易于复用。

随着你的使用模式越来越复杂，显式传递上下文对象会让代码变得越来越混乱，使用 this 则不会这样。当我们介绍对象和原型时，你就会明白函数可以自动引用合适的上下文对象有多重要。