#II. A TREE-WALK INTERPRETER

> With this part, we begin jlox, the first of our two interpreters. Programming languages are a huge topic with piles of concepts and terminology to cram into your brain all at once. Programming language theory requires a level of mental rigor that you probably haven't had to summon since your last calculus final. (Fortunately there isn't too much theory in this book.)

在这部分中，我们开始学习jlox，这是我们两个解释器中的第一个。编程语言是一个巨大的话题，其中有大量的概念和术语需要一下子塞进你的大脑。编程语言理论需要一定程度的脑力投入，你可能自上次微积分期末考试后就没这么投入过了。(幸运的是，这本书没有太多的理论。）

> Implementing an interpreter uses a few architectural tricks and design patterns uncommon in other kinds of applications, so we'll be getting used to the engineering side of things too. Given all of that, we'll keep the code we have to write as simple and plain as possible.

实现一个解释器需要一些架构技巧和设计模式，这在其他类型的应用程序中是不常见的，所以我们也要习惯于工程方面的东西。考虑到这些，我们会尽可能地让代码简单明了。

> In less than two thousand lines of clean Java code, we'll build a complete interpreter for Lox that implements every single feature of the language, exactly as we've specified. The first few chapters work front-to-back through the phases of the interpreter—scanning, parsing, and evaluating code. After that, we add language features one at a time, growing a simple calculator into a full-fledged scripting language.

在不到2000行简洁的Java代码中，我们将为Lox构建一个完整的解释器，完全按照我们指定的方式实现该语言的每一个功能。前几章从头到尾介绍解释器的各个阶段——扫描、解析和计算代码。之后，我们逐步添加语言特性，将一个简单的计算器发展成一种成熟的脚本语言。

PREV TITLE / CONTENTS NEXT