

第三章

编程语言 Python

当提到 Python 时，人们可能首先想到一条危险的蟒蛇。尽管名称听起来像，但是 Python 与此无关。Python 是现在可用于计算机编程的众多语言之一。在本章中，你将发现为什么 Python 是编程领域中的入门语言。

机器语言——处理器的母语

计算机除执行给定命令外不执行其他操作。一个接一个被执行的命令的列表被称为“程序”。处理器中的各个命令非常简单，例如从存储单元 34567 中获取一个字节到处理器中，或者将处理器内存中当前字节的值加倍，或者将字节发送到图形单元的数据线路上。这些命令本身不是以字词的形式提供给处理器的，而是采用由位和字节组成的数字代码。处理器知道哪个位组合代表哪条指令，然后执行该指令。处理器的这种内部“母语”被称为机器语言。迄今为止，计算机处理器基本上只懂机器语言。

但是，除非在非常罕见的特殊情况下，现在几乎没有人会使用机器语言编程。在这些特殊情况中，必须控制简单、快速的设备或必须扩展操作系统。如果你是普通使用者，希望使用机器语言进行编程，即使仅仅编写在屏幕上写入“Hello”一词的程序，也必须将数百条密码一样的数字命令组合在一起。

因此，在很早以前就已经为开发人员研发出所谓的“高级编程语言”。这些语言比机器语言更容易使用。它们包含一些直接表达计算机应当执行哪些操作的命令（例如在屏幕上写“Hello”，只有一行），然后一条内部程序将这个命令翻译成幕后的上百行，甚至更多行机器语言，以便机器按照程序员想象的那样执行。使用高级编程语言

进行编程时，始终涉及两个层级：我们在电脑上输入的更高级的语言在一个层级上，而将命令翻译成机器语言以供电脑操作的解释器在另一个层级上。

解释器和编译器

有两种高级编程语言类型，即解释型语言和编译型语言。

编译器总是首先将整个完成的程序完全翻译成机器语言（称为编译），然后计算机执行整个程序。

著名的编译型语言有 C 和 C++。它们的优点是可以用来编写运行速度非常快的高性能程序。但是它们的缺点是编程要求更高，通常比较无趣。因为程序员必须事先考虑很多具体的数据格式，然后进行计算机的内部处理和数据管理，以便最终创建功能全面的机器程序。每次更改程序时，都必须重新编译整个程序，这使得快速测试变得费力，并且如果程序中有错误，则整个计算机很容易“崩溃”。

编译型语言对于时间敏感的程序、游戏、控制和操作系统的专业编程非常重要。如今，许多专业软件都是用 C++ 编写的。

解释型语言也是非常强大的，并且在许多专业领域中也得到使用。对于解释型语言，程序不必在执行之前进行翻译；相反，管理程序和翻译程序始终在后台同时运行（解释器 = 翻译器或程序引擎），轮到该命令时，翻译并执行该命令，然后继续执行下一条命令。该程序引擎自动负责在后台进行内存分配，或对其进行合理管理、识别、拦截和防止错误。这使程序员的工作更加轻松，并且让他可以专注于基础内容和非常快速、轻松地测试程序。与此同时，这些程序的运行速度稍慢，因为后台始终有一个程序引擎在运行、检查和翻译。但是以当今计算机的速度，在大多数情况下，这几乎无足轻重。

解释型语言更易于学习和使用。因此，它们非常适合作为编程基础进行学习。典型的解释型语言有 BASIC（过去非常流行）、PHP、JavaScript，以及 Python。



介于解释型和编译型之间的语言

此外，还有一些“折中的”语言，它们使用编译器进行预翻译，与此同时程序引擎在后台运行。由此，可以在更轻松使用和更快速执行之间实现折中。Java 和 C# 就属于这类语言。尽管进行了简化，但它们还是更适合高级程序员，纯解释型语言无论如何还是更适合用于学习编程。

Python——简单通用

你已经决定学习如何使用 Python 编程。这是一个很好的选择。无论你想专业地学习编程还是为了娱乐而学习，Python 绝对是正确的入门工具。

- **Python 是通用的**——这意味着，如果你掌握了该语言及其应用程序，你就可以使用 Python 在所有可能的领域中进行编程，无论是学习程序、图形程序还是在线软件、游戏或机器人控件。而 JavaScript 这种解释型语言只能为网络浏览器编写应用程序，或者使用 PHP 之类的语言创建网络服务器程序，而 Python 几乎可以在任何领域中使用。
- **Python 是简单明了的**——了解一些 Python 的基本命令和结构后，就可以在各种复杂环境中不断使用。Python 中的程序一目了然，易于阅读和理解，因为它们使用缩进而不是易造成混乱的括号。同时，Python 使用了其他专业编程语言中出现的所有复杂原理。但是使用 Python 时，从一开始理解这些内容就十分容易。正确理解 Python 之后，可以根据需要轻松切换到另一种语言。
- **Python 是新式的**——这种编程语言比某些传统编程语言年轻多了。它由荷兰计算机程序员吉多·范·罗苏姆（Guido van Rossum）于 1994 年发明，他的意图十分明确，就是希望开发一种全新的、简单易懂的编程语言。Python 极尽清晰明了，它使用尽可能少的命令，并尽可能易于使用。使用 Python，无需过多技术知识你就能学习编程。



喜剧团体

顺便提一句，“Python”并没有像许多人所想的那样与蟒蛇同名，它源自英国喜剧团体“Monty Python”，因为开发者是这一团体的忠实拥护者。如果需要，你可以通过搜索引擎搜索相关信息。Monty Python 在二十世纪七十年代颇具传奇色彩，他们主要为电视节目“Monty Python’s Flying Circus”（蒙蒂·派森的飞翔马戏团）制作了非常新颖有趣的短剧。由 Monty Python 主演的电影《布莱恩的一生》（又译《万世魔星》）和《脱线一箩筐》（又译《人生七部曲》）也是广受欢迎的影片。若感兴趣想观看，最好有良好的英语水平，因为影片语言为英语。

现在主要有两个 Python 版本：Python 2 和 Python 3。如今，这两个版本的使用量几乎相同。即使在 Python 3 中对某些内容进行了内部优化，并且一些命令的工作方式也有所不同，但这两个版本的基本原理是相同的，并且大多数程序的工作方式完全相同（有时会有微小变化）。因此，Python 2 仍然被广泛用于许多应用程序中。Python 2 和 Python 3 一样适合学习。

Jython——这是什么？

在本课程中学习 Python 时使用的 Python 引擎被称为 Jython。这是什么意思？

Jython 是由“Java”和“Python”组成的人造词。具体而言，这意味着程序引擎（后台的翻译和管理程序）是用 Java 编写的，而所使用的语言是纯 Python。这有几个优点：由于 Java 几乎可以在所有计算机系统（无论是 Mac、Windows 还是 Linux 系统）上运行，因此 Jython 引擎可以在任何地方使用而不会出现任何问题；我们将要编写的 Python 程序也可以使用 Java 领域中的特殊库和函数。正如我们将看到的那样，这是非常实用的。你不需要额外拥有 Java 知识，因为 Java 仅在后台运行，以执行 Python 命令。

Jython 中使用的语言就是真正的 Python（2.7 版本），这就是我们要学习的东西。无论使用哪种平台，Jython 都可以帮助我们确保从一开始就无需复杂地工作。

TigerJython——你的学习环境

TigerJython 是一个完整的程序包，其中包括 Jython 引擎和我们需要的所有库（你将了解有关库的更多信息），甚至可以将图形和声音用到你自己的程序中。因此，这是一个“包含所有的包裹”，就像已经插入电池的玩具一样。

TigerJython 是由贾卡·阿诺德（Jarka Arnold）、托比亚斯·科恩（Tobias Kohn）、埃吉迪乌斯·普吕斯（Aegidius Plüss）在瑞士开发的，特别适合初学者学习 Python。

TigerJython 可以在几分钟内准备就绪，你可以立即开始。

这就是你将在下一章中所做的！TigerJython 几乎可以在所有常规计算机上运行，无论是台式机还是笔记本电脑，无论是使用 Windows、Mac 还是 Linux 系统的计算机。TigerJython 甚至可以安装在 Raspberry Pi 微型计算机上。



在平板电脑上如何安装 TigerJython？

本书中使用的完整版本 TigerJython 不能在装有 iOS 或 Android 的移动设备上运行。这挺合理的，因为我们强烈建议使用带有真实键盘的“真正的计算机”，尤其是在学习编程时，程序员必须在工作中大量打字。

但是，如果你仍然想在平板电脑等移动设备上使用 TigerJython，有一个名为“WebTigerJython”的新项目正在开发中。这意味着 TigerJython 的受限版本可以在每个网络浏览器中运行，无须安装。本书中直到第十一章的示例已经在该系统上完美运行。

你可以在网站上找到有关 WebTigerJython 的最新信息。