

# 编程小白的第一本 Python 入门书

作者：侯爵

版权信息

书名：编程小白的第一本 Python 入门书

作者：侯爵

本书由北京图灵文化发展有限公司发行数字版。版权所有，侵权必究。

---

您购买的图灵电子书仅供您个人使用，未经授权，不得以任何方式复制和传播本书内容。

我们愿意相信读者具有这样的良知和觉悟，与我们共同保护知识产权。

如果购买者有侵权行为，我们可能对该用户实施包括但不限于关闭该帐号等维权措施，并可能追究法律责任。

---

图灵社区会员 人民邮电出版社 (zhanghaichuan@ptpress.com.cn) 专享 尊重版权

[写在前面：你需要这本书的原因](#)

[第一章 为什么选择 Python?](#)

[1.1 Python 能做什么?](#)

[第二章 现在就开始](#)

[2.1 安装 Python 环境](#)

[在 Windows 上安装 Python](#)

[在 Mac 上安装 Python](#)

[在 Linux 上安装 Python](#)

[2.2 使用 IDE 工具](#)

[第三章 变量与字符串](#)

[3.1 开始学习编程](#)

[3.2 变量](#)

[3.3 print\(\)](#)

[3.4 字符串](#)

[字符串是什么](#)

[字符串的基本用法](#)

[字符串的分片与索引](#)

[字符串的方法](#)

[字符串格式化符](#)

[第四章 函数的魔法](#)

[4.1 重新认识函数](#)

[4.2 开始创建函数](#)

[练习题](#)

[4.3 传递参数与参数类型](#)

[4.4 设计自己的函数](#)

[第五章 循环与判断](#)

[5.1 逻辑控制与循环](#)

[逻辑判断——True & False](#)

[比较运算 \(Comparison\)](#)

[比较运算的一些小问题](#)

[成员运算符与身份运算符 \(Membership & Identity Operators\)](#)

[布尔运算符 \(Boolean Operators\)](#)

[5.2 条件控制](#)

[5.3 循环 \(Loop\)](#)

[for 循环](#)

[嵌套循环](#)

[While 循环](#)

[练习题](#)

[5.4 综合练习](#)

[练习题](#)

[第六章 数据结构](#)

[6.1 数据结构 \(Data Structure\)](#)

[6.2 列表 \(list\)](#)

[列表的增删改查](#)

[6.3 字典 \(Dictionary\)](#)

[字典的增删改查](#)

[6.4 元组 \(Tuple\)](#)

[6.5 集合 \(Set\)](#)

[6.6 数据结构的一些技巧](#)

[多重循环](#)

[推导式](#)

[循环列表时获取元素的索引](#)

[6.7 综合项目](#)

[第七章 类与可口可乐](#)

[7.1 定义一个类](#)

[7.2 类的实例化](#)

[7.3 类属性引用](#)

[7.4 实例属性](#)

[7.5 畅爽开怀，实例方法](#)

[self?](#)

[7.6 更多参数](#)

[7.7 魔术方法](#)

[7.8 类的继承](#)

[令人困惑的类属性与实例属性](#)

[类的扩展理解](#)

[7.9 类的实践](#)

[似乎这并没有解决什么问题?](#)

[为什么?](#)

[安装自己的库](#)

[练习题](#)

[第八章 开始使用第三方库](#)

[8.1 令人惊叹的第三方库](#)

[8.2 安装第三方库](#)

[最简单的方式：在 PyCharm 中安装](#)

[最直接的方式：在终端/命令行中安装](#)

[最原始的方式：手动安装](#)

[8.3 使用第三方库](#)

[必读：给编程小白的学习资源](#)

[练手项目](#)

[资料库参考](#)

[基础教程与书籍](#)

写在前面：你需要这本书的原因

有没有哪一个瞬间，让你想要放弃学习编程？

在我决心开始学编程的时候，我为自己制定了一个每天编程1小时的计划，那时候工作很忙，我只能等到晚上9点，同事都下班之后，独自留在办公室编程。在翻遍了我能找到的几十本国内外的 Python 编程教程之后，我还是似懂非懂。那些教程里面到处都是抽象的概念、复杂的逻辑，对于专业开发者这些再平常不过，而对于我这样一个学设计出身的编程小白，没有被视觉化的东西是无法被理解的。

而且，这些书大多着重于一步步构建一个完整体系，但事实上，现实生活中没有哪个技能是这么习得的。难道要练习1年切菜才能给自己做一顿饭么？难道要到体校学习3年才能开始晨跑么？难道要苦练5年基本功才能开始拿起吉他弹第1首曲子么？

做任何事情一定有人在短期内简单可行的方法。学习不应该是苦差事，而应该是快乐的，重要的是找到适合自己的学习方法。

既然笨办法不能让我学会 Python，那么我决定用一种聪明方法来学，为自己创造学习的捷径。这种高效学习法的核心在于：

- 1、精简：学习最核心的关键知识；
- 2、理解：运用类比、视觉化的方法来理解这些核心知识；
- 3、实践：构建自己的知识体系之后，再通过实践去逐渐完善知识体系。

实际上，如果你听说过《如何高效学习》中的整体学习法，你会发现它和我的高效学习法很相似，作者斯科特·杨用一年的时间学完了麻省理工四年的课程。既然这种高效学习法可以用来学习经济学、数学、物理，那么当然也可以用来学编程。

运用了高效学习法之后，我的编程学习速度突飞猛进，不到一个月时间就完成了爬虫加上 Django 的网站。为了追求最大程度的精简，我借用了许多强大的库和框架，这我不需要重新发明轮子，而是专注于最核心的功能。在一次项目合作中，我惊讶的发现，我用70行代码实现的功能，一名工程师竟然用了800行代码来实现。在运动场上，第一名和最后一名的差距也许只有几十秒。然而在编程这个竞技场上，不同方法的效率竟然会有10倍的差距，这也正是追求高效学习的价值所在。

为了能让更多的编程小白轻松地入门编程，我把高效学习法结合 Python 中的核心知识，写成了这本书。随意翻上几页，你就会发现这本书和其他编程书的不同，其中有大量的可视化场景帮你理解概念，穿插了若干有趣的小项目，最重要的是，这本书是为零基础小白而设计的。

。

考虑到很多书都标榜是针对零基础读者的，所以有必要说明一下这一本有哪些独到之处。

我不写字典式的知识体系，有些编程书像字典一样，各种细枝末节的知识都涵盖在内。但我认为，正如不应该让小孩拿着字典学汉语，也不应该让初学者拿着这样的厚书学编程。在汉语中，掌握常见的1500-2000个字就能看懂80%的文字。在编程中，同样有着最核心的关键知识。先用这些关键的知识构建你的知识体系会让学习效率加速，这是比一开始就钻到各种细枝末节里面更好的学习方式。这是精简的意义。

我不去对比各种语言的特点，许多程序员背景的作者喜欢去对比 Python 和其他语言有什么异同，或者试图让你通过理解 C 语言从而理解 Python，但我不会这么做。我知道对于大多数读者，Python 很可能是将要学习的第一门编程语言，所以我不会用一个陌生概念讲解另一个陌生概念，反过来，我会运用类比和视觉化的方法讲解 Python 中的抽象概念，把复杂的东西简单的讲清楚。这是理解的捷径。

我不追求让你达到精通的程度，事实上我也很怀疑有哪本书能真正做到21天从入门到精通。精通一门语言，需要在实际项目开发中踩过许多坑，需要熟悉计算机运作的底层原理。我是一名实用主义的开发者，我相信你也一样，学习编程是为了真正做出点东西来，也许你想爬取大量的数据和信息，方便用来分析与决策。也许你想快速搭建一个网站，展示自己的产品。也许你对量化交易感兴趣，想试着把自己的投资策略程序化。对于实用主义的开发者来说，更应该追求的是“达成”而不是“精通”。先掌握项目所需的最少必要知识，然后把热情和精力投入到搭建真实项目中，而不是死磕半年的基础知识，直到把所有兴趣都耗竭了也没做出什么像样的东西。在实践中，你自然会逐渐完善知识体系。在这本书里面，会穿插一些真实项目的片段，让你知道学了这个基础知识能用在哪儿，并且完成一些小型项目。这是让你最有成就感的实践。

说了这么多，就是让你能放下疑虑。这不是一本让你中途放弃的编程书，这是一本黏着你看完的编程书。大多数读者都能在一周内读完，其中有35岁才开始学编程的中年男子、有工作非常忙碌的女性创业者、还有对编程感兴趣的高中生。所以，相信你也可以跟着这本书一起从零到一。

放轻松，如果你准备好了，那就翻开下一页吧。

注：除了在这里在线阅读，你还可以到「随书下载」中免费下载 pdf 版本的电子书或是推送 mobi 版。

。

作者：侯爵

麻瓜编程创始人。网易云课堂上最畅销的课程《Python实战》系列课程讲师，目前已有超过4万名学员。

设计专业背景，拥有设计与编程跨界思维，善于找到学习技能的最佳路径，擅长把复杂的东西简单的讲清楚。

初学编程时，发现市面上很难找到适合小白的学习资料，于是开始用生动易懂、视觉化的方式来写这本教程。

## 第一章 为什么选择 Python?

### 1.1 Python 能做什么?

那些最好的程序员不是为了得到更高的薪水或者得到公众的仰慕而编程，他们只是觉得这是一件有趣的事情。——Linux 之父 Linus Torvalds

作为一个实用主义的学习者，最关心的问题一定是“我为什么要选择学 Python，学会之后我可以用来做什么？”

首先，对于初学者来说，比起其他编程语言，Python 更容易上手。

Python 的设计哲学是优雅、明确、简单。在官方的 The Zen of Python (《Python 之禅》) 中，有这样一句话，

There should be one-- and preferably only one --obvious way to do it.

Python 追求的是找到最好的解决方案。相比之下，其他语言追求的是多种解决方案。

如果你试着读一段写的不错的 Python 代码，会发现像是在读英语一样。这也是 Python 的最大优点，它使你能够专注于解决问题而不是去搞明白语言本身。

注：漫画《口渴的 Python 开发者》，形容了 Python 开发者是多么轻松，来自 Pycot 网站

其次，Python 功能强大，很多你本来应该操心的事情，Python 都替你考虑到了。当你用 Python 语言编写程序的时候，你不需要考虑如何管理你的程序使用的内存之类的底层细节。并且，Python 有很丰富的库，其中有官方的，也有第三方开发的，你想做的功能模块很有可能已经有人写好了，你只需要调用，不需要重新发明轮子。这就像是拥有了智能手机，可以任意安装需要的 app。

漫画 Python, 作者 xkcd

这幅漫画形容了 Python 的库有多强大，导入一个反重力库就可以飞起来了。

第三，Python 能做的事情有许多。

在职场中，使用 Python 工作的主要是这样几类人：

- 网站后端程序员：使用 Python 搭建网站、后台服务会比较容易维护，当需要增加新功能，用 Python 可以比较容易的实现。不少知名网站都使用了 Python 开发，比如：
- 自动化运维：越来越多的运维开始倾向于自动化，批量处理大量的运维任务。Python 在系统管理上的优势在于强大的开发能力和完整的工具链。
- 数据分析师：Python 能快速开发的特性可以让你迅速验证你的想法，而不是把时间浪费在程序本身上，并且有丰富的第三方库的支持，也能帮你节省时间。
- 游戏开发者：一般是作为游戏脚本内嵌在游戏中，这样做的好处是即可以利用游戏引擎的高性能，又可以受益于脚本化开发的优点。只需要修改脚本内容就可以调整游戏内容，不需要重新编译游戏，特别方便。
- 自动化测试：对于测试来说，要掌握 Script 的特性，会在设计脚本中，有更好的效果。Python 是目前比较流行的 Script。

如果你是一名业余开发者，只是想在资源少的情况下快速做出自己想要的东西、自动化的解决生活中的问题，那么 Python 可以帮你做到这几类事情：

- 网站的开发  
借助功能丰富的网站框架 django、flask 等等，你可以快速搭建自己的网站，还可以做到移动端自适应。  
注：Python 全栈实战课程的项目：十分钟短视频平台
- 用爬虫爬取或处理大量信息  
当你需要获取大批量数据或是批量处理的时候，Python 爬虫可以快速做到这些，从而节省你的重复劳动时间。比如：微博私信机器人、批量下载美剧、运行投资策略、刷折扣机票、爬合适房源、系统管理员的脚本任务等等。  
注：Python 爬虫实战课程的项目：二手行情网站
- 再包装其他语言的程序  
Python 又叫做胶水语言，因为它可以用混合编译的方式使用 c/c++/java 等等语言的库。
- 智能硬件  
树莓派作为微型电脑，也使用了 Python 作为主要开发语言。  
用红外线遥控器控制树莓派，作者八宝粥

最后，附一张选择编程语言的小测试，你可以根据你的需要，选择学习哪种语言。

The lord of the rings, 作者 carlkeo

## 第二章 现在就开始

### 2.1 安装 Python 环境

在你开始学习 Python 之前最重要的是——对，你要安装 Python 环境。许多初学者会纠结应该选择 2.x 版本还是 3.x 版本的问题，在我看来，世界变化的速度在变得更快，语言的更新速度亦然。没有什么理由让我们只停留在过去而不往前看。对于越来越普及、同时拥有诸多炫酷新特性的 Python 3.x，我们真的没有什么理由拒绝它。如果你理解了 *life is short, you need Python* 的衷衷，就更应该去选择这种「面向未来」的开发模式。

所以，我们的教材将以最新的 Python 3.x 版本为基础，请确保电脑上有对应版本。

在 Windows 上安装 Python

第一步

根据你的 Windows 版本（64位还是32位），从 Python 的官方网站下载对应的 Python 3.5，另外，Windows 8.1 需要选择 Python 3.4，地址如下：

- [Python 3.5 64位安装程序下载地址](#)
- [Python 3.5 32位安装程序下载地址](#)
- [Python 3.4 64位安装程序下载地址](#)
- [Python 3.4 32位安装程序下载地址](#)
- 网速慢的同学请移步[国内镜像](#)

然后,运行下载的EXE安装包:

□

特别要注意勾选 Add Python 3.5 to PATH，然后点 Install Now 可完成安装。默认会安装到 C:\Python35 目录下。

第二步

打开命令提示符窗口（方法是点击“开始”-“运行”-输入：“cmd”），敲入 Python 后，会出现两种情况：

- 情况一：

□

看到上面的画面，就说明 Python 安装成功！

你看到提示符 >>> 就表示我们已经在 Python 交互式环境中了，可以输入任何 Python 代码，回车后会立刻得到执行结果。现在，关掉命令行窗口，就可以退出 Python 交互式环境。

- 情况二：得到一个错误：

Python 不是内部或外部命令，也不是可运行的程序或批处理文件。

□

这是因为 Windows 会根据一个 Path 的环境变量设定的路径去查找 Python.exe，如果没找到，就会报错。如果在安装时漏掉了勾选 Add Python 3.5 to PATH，那就要手动把 Python.exe 所在的路径添加到 Path 中。

如果你不知道怎么修改环境变量，建议把 Python 安装程序重新运行一遍，务必记得勾选 Add Python 3.5 to PATH。

在 Mac 上安装 Python

□

如果你正在使用 Mac，系统是 OS X 10.8~10.10，那么系统自带的 Python 版本是 2.7，需要安装最新的 Python 3.5。

第一步：

- 方法一：下载安装

- 从 Python 官网下载 [Python 3.5 安装程序](#)
- 网速慢的同学请移步[国内镜像](#)

Mac 的安装比 Windows 要简单，只需要一直点击继续就可以安装成功了。

- 方法二：Homebrew 安装

如果安装了 Homebrew，直接通过命令 `brew install Python3` 安装即可。

第二步：

如果不放心，可以再检查一下。操作方法是打开终端，输入 `Python3`（不是输入 `Python 3`，也不是 `Python`）

□

得到这样的结果，就说明安装成功了。

在 Linux 上安装 Python

一个好消息是，大多数 Linux 系统都内置了 Python 环境，比如 Ubuntu 从 13.04 版本之后，已经内置了 Python 2 和 Python 3 两个环境，完全够用，你不需要再折腾安装了。

如果你想检查一下 Python 版本，打开终端，输入：

```
python3 --version
```

就可以查看 Python3 是什么版本的了。

如果你需要安装某个特定版本的 Python，在终端输入这一行就可以：

```
sudo apt-get install python3.5
```

其中的 3.5 可以换成你需要的版本，目前 Python 最新是 3.5 版。

## 2.2 使用 IDE 工具

安装好环境之后，还需要配置一个程序员专属工具。正如设计师使用 Photoshop 做图、产品经理使用 Axure 做原型，程序员也有编程的工具，叫做：IDE。

在这里推荐公认最智能最好用的 Python IDE，叫做 PyCharm，同时支持 Windows 和 Mac 用户，本教程使用的版本是目前最新的 3.4 版本。

这里是 [PyCharm 的官网下载链接](#)。

社区版是免费的，专业版是付费的。对于初学者来说，两者的差异微乎其微，使用社区版就够用了。

。

到这里，Python 开发的环境和工具就搭建好了，我们可以开始安心编程了。

如果你是第一次上手编程，可能会对 IDE 感到很陌生，甚至不知道怎样创建一个新文件。在这里推荐一些容易上手的 PyCharm 学习视频：

- [快速上手的中文视频](#)，简单介绍了如何安装、如何创建文件、如何设置皮肤。新手先掌握这些就够用了。
- [PyCharm 官方的快速上手视频](#)，第一节视频就让你快速掌握这个工具的基本使用方法，如果你想继续深入了解，可以继续看后面8节短视频，每个在3-5分钟，全面的介绍了如何更有效率的使用 PyCharm。
- [如何高效使用 PyCharm 的系列文档](#)，图文并茂的介绍了许多高效的技巧，比如快捷键设置等等，可以在上手之后持续学习。

可能有些同学会有疑问，在这里解答一下。

为什么不需要安装 Python 解释器？

因为在 Python 官方网站下载了 Python 3.5 之后，就自带了官方版本的解释器，所以不需要再次安装。

为什么不使用文本编辑器，比如 Sublime？

因为文本编辑器是相对轻量级的，和 IDE 相比功能太弱了，尤其在 debug 的时候会遇到很多问题。

能不能不安装 IDE，直接在命令行或者终端里编程？

可以。但是在命令行中保存完整代码很麻烦，最重要的是编辑器是交互式的，不小心手滑写错的代码无法修改，要重新敲一遍。珍惜时间，善用工具。

### 第三章 变量与字符串

#### 3.1 开始学习编程

初学者经常会遇到的困惑是，看书上或是听课都懂，但还是不明白要怎么编程。这是因为缺乏足够多的实践。

正如我们在婴儿时期学习说话的时候，最初是模仿父母的发音，逐渐才能学会表达自己的想法。学习编程也是一样，在你阅读这本教程的时候，需要模仿着示例敲一遍代码，不要怕麻烦、不要嫌简单，当你动手敲代码的时候，就会发现很多眼睛会忽略的细节：小到中文标点还是英文标点、大到语句之间的逻辑关系。当然，在你发现亲手写出的程序运行成功之后，你也会感受到无比的喜悦，你能用程序计算数学题了！你能实现小功能了！我会带着你循序渐进地完成一个个实践，直到你有能力脱离模仿、开始创造。

所以，你需要在阅读时打开 IDE，边看边敲代码。如果安装环境这一步遇到了问题，请回到上一章阅读。

如果你准备好了，跟我走吧。

欢迎访问：电子书学习和下载网站 (<https://www.shgis.cn>)

文档名称：《编程小白的第一本 Python 入门书》侯爵 著.epub

请登录 <https://shgis.cn/post/326.html> 下载完整文档。

手机端请扫码查看：

