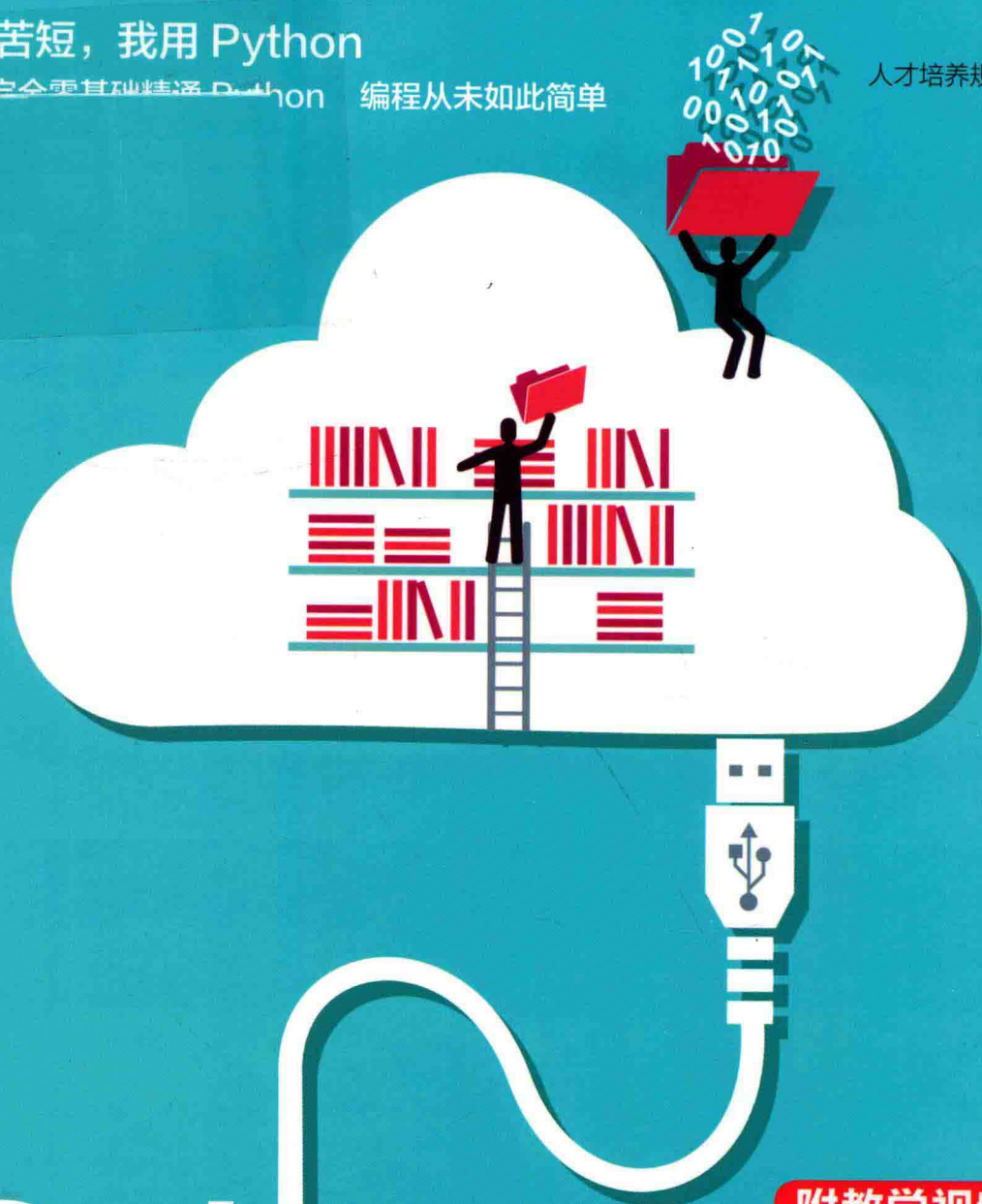


人生苦短，我用 Python

15 天零基础基础精通 Python 编程从未如此简单

大数据

人才培养规划教材



Python 基础教程

第一门编程语言

附教学视频

吕云翔 / 主编 姜峤 孔子乔 / 副主编

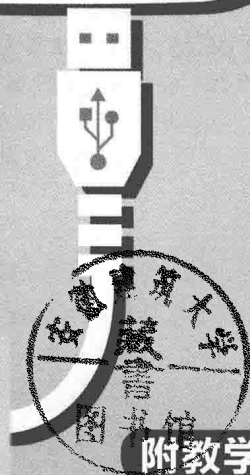
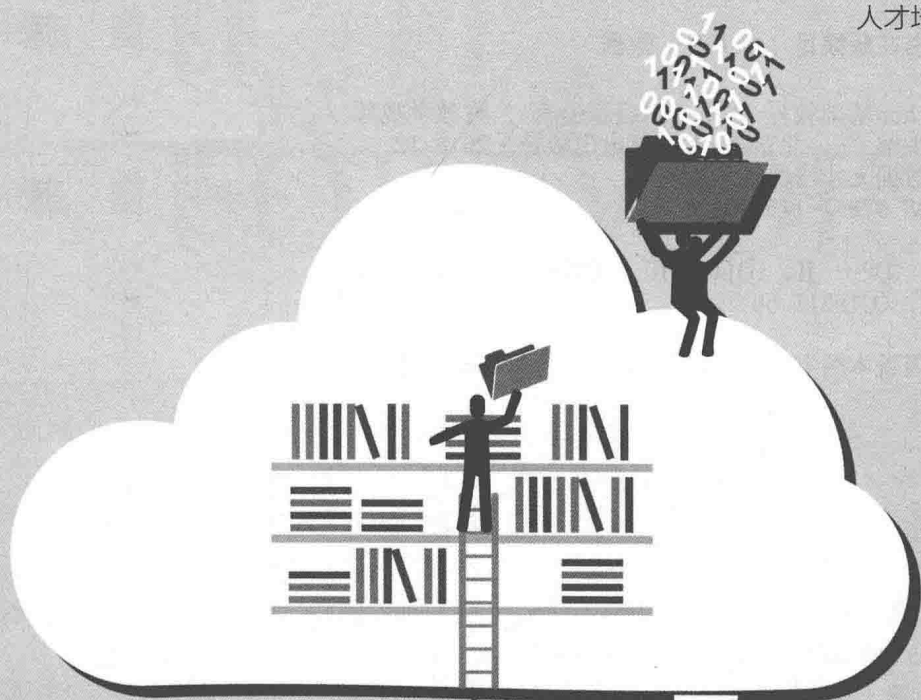


中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

大数据
人才培养规划教材



附教学视频

Python 基础教程

第一门编程语言

吕云翔 / 主编 姜峤 孔子乔 / 副主编

人民邮电出版社
北京

图书在版编目(CIP)数据

Python基础教程：第一门编程语言：附教学视频 /
吕云翔主编. — 北京：人民邮电出版社，2018.12
大数据人才培养规划教材
ISBN 978-7-115-49126-8

I. ①P… II. ①吕… III. ①软件工具—程序设计—
教材 IV. ①TP311.56

中国版本图书馆CIP数据核字(2018)第185253号

内 容 提 要

本书结合了 Python 3.6 的新特性，完全为零基础的初学者量身定做。书中例举大量实例，介绍了 Python 的基本语法、编码规范和一些编程思想。

本书第 1~8 章为 Python 语言基础，主要介绍 Python 的基本用法；第 9 章为一个实战，帮助读者理解前 8 章的知识；第 10~17 章为 Python 的进阶使用，包含面向对象编程、函数式编程入门、文件读写、异常处理、模块和包几个部分；第 18 章为第 2 个实战，帮助读者融会贯通前 17 章的知识，同时抛砖引玉，引起读者探索的兴趣。

本书既可以作为高等院校计算机与软件相关专业的教材，也可以作为软件从业人员、计算机爱好者的学习指导用书。

-
- ◆ 主 编 吕云翔
副 主 编 姜 峤 孔子乔
责任编辑 刘 佳
责任印制 马振武
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
涿州市京南印刷厂印刷
 - ◆ 开本：787×1092 1/16
印张：13.25 2018 年 12 月第 1 版
字数：302 千字 2018 年 12 月河北第 1 次印刷
-

定价：42.00 元

读者服务热线：(010)81055256 印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广登字 20170147 号



前言

FOREWORD

在 阅读这本书前，先问自己一个问题：我是出于什么目的学习 Python？如果一时答不上来这个问题，不妨把它看成选择题，是好玩，还是有用？

如果学 Python 是为了好玩，那么千万不要三天热度。

学习的过程永远不可能一帆风顺，伴随乐趣的同时必然会有坎坷。如果想从中发现源源不断的乐趣，并且不被其中的坎坷所绊倒，那就需要给自己制定一些目标，比如发现一个好玩的研究方向，写一个自己认为有用的小程序等。真正认为学编程好玩的人，能在这条路上走得最远。希望各位都能发现属于自己的乐趣。

如果学 Python 是为有用，为了学习编程，那么不妨想一想自己了解编程是为了什么。

如果是计算机方向从业人员，那么对编程的能力要求会比较高。虽然 Python 不可能包含所有的编程知识，但是学习 Python 可以打开一扇编程之门，后面的道路通往四面八方，读者需要找到自己的位置，明白自己距离目的地还有多远。比如想成为后端工程师，除了掌握一门适合的后端语言，还需要掌握数据库等多项技能。

非计算机方向的从业人员，学习 Python 的时候不要有太多负担，因为学习使用 Python 是为了解编程和使用 Python 这个工具，并不需要达到计算机从业人员的要求。Python 能够带来便捷，这就足够了，在遇到困难的时候不要有过多的纠结，找准自己的需求，放心大胆去学去用。

对于刚刚开始学习编程的初学者，他们有着非常多优秀的语言可以选择，但是笔者推荐选择 Python 作为第一门语言，结合本书边实践边学习。不推荐直接开始学习 C++、Java 等高级语言，原因主要有两条。

第一，Python 实在太先进了。这并不是说 C++、Java 等语言落后于时代，而是说它们对于初学者而言，抽象的概念太多，太深，理解难度大，容易令人丧失兴趣。相比而言，Python 上手要简单许多，因为 Python 本身是一门现代化的语言，它没有历史的包袱，非常人性化，而且是符合人类直觉的。通过学习本书精选的内容，所有初学者都可以顺利学会 Python。

第二，兴趣才是最好的老师。初学编程一定要注意找到自己的兴趣所在。Python 这门语言可以让初学者轻松做到很多有趣的事情而不需要考虑复杂的实现过程，所以它并不会打击学习的积极性，反而能带来很大成就感。本书的内容考虑到了读者的接受能力，仔细按照内容的依赖关系和难度进行了排序。如果 Python 是你学的第一门语言，那么本书再适合不过了，只要从头开始学即可由浅及深，系统地学习 Python 的内

容。如果你有一定的基础，那么这本书的简洁性会让你印象深刻。

本书的目标非常明确，就是为初学者量身定做一套以兴趣为导向的 Python 教程，其中包含以下特点。

1. 非常适合初学者：本书针对的是没有学过编程的初学者，内容不但简单明了，而且会将概念的说明减少至最少，从而专注于通过实践去理解。

2. 基于 Python 3.6：要学就学最新的，本书全部基于 Python 3 而且会加入 Python 3.6 才有的一些特性，让读者体会最新版 Python 带来的便捷。

3. 基于实践的理论学习：据笔者了解，很多人学习编程的时候存在一个误区，就是认为书看完了就懂了，结果一动手就抓瞎。正如 Linux 的创始人 Linus Torvalds 所说的：“Talk is cheap, show me the code!” 在本书的讲解中实践贯穿始终，迫使初学者去动手练习，在书写代码中掌握知识。

4. 习题设计：小练习和两个实战可以帮助初学者将所学的知识融会贯通，并且激发其探索编程领域中其他知识的欲望。

5. 趣味性：本书不保证处处有趣，但是至少不会处处无聊。

本书的作者为吕云翔、姜峤、孔子乔，曾洪立、吕彼佳、姜彦华参与了部分内容的编写并进行了素材整理及配套资源制作等。

由于编者水平有限，本书难免会有内容的疏漏，恳请广大读者给予批评指正，也希望各位能与我们交流实践过程中获得的经验和心得（yunxianglu@hotmail.com）。

编 者

2018.6.1

目 录 CONTENTS

第 1 章 欢迎来到 Python 的世界..... 1	第 3 章 强大的包管理器 pip..... 26
1.1 Python 是什么..... 1	3.1 包..... 26
1.1.1 Python 是一门语言..... 1	3.2 包管理器..... 26
1.1.2 Python 是一个工具..... 1	3.3 pip..... 27
1.1.3 Python 是一瓶胶水..... 1	3.3.1 pip3 search..... 27
1.2 获取 Python..... 2	3.3.2 pip3 list..... 28
1.2.1 Windows..... 2	3.3.3 pip3 check..... 28
1.2.2 Linux..... 4	3.3.4 pip3 download..... 29
1.2.3 macOS..... 4	3.3.5 pip3 install..... 29
1.2.4 Android..... 7	3.3.6 pip3 freeze..... 29
1.2.5 iOS..... 8	3.3.7 pip3 uninstall..... 30
1.3 初试 Python..... 9	小结..... 30
1.4 Python 的优点..... 10	习题..... 30
1.4.1 简洁优美..... 10	第 4 章 基本计算..... 31
1.4.2 上手简单..... 11	4.1 四则运算..... 31
1.4.3 应用广泛..... 11	4.2 数值类型..... 31
1.4.4 平台独立..... 11	4.2.1 整型 (integer)..... 32
1.5 Python 的应用..... 11	4.2.2 浮点型 (float)..... 33
小结..... 12	4.2.3 复数类型 (complex)..... 34
习题..... 13	4.2.4 布尔型 (bool)..... 34
第 2 章 写 Python 代码的工具..... 14	4.3 数值类型转换..... 34
2.1 交互式解释器..... 14	4.4 变量..... 35
2.2 IPython..... 14	4.4.1 什么是变量..... 35
2.3 IDLE..... 16	4.4.2 声明变量..... 36
2.4 PyCharm..... 19	4.4.3 动态类型..... 36
2.4.1 安装 PyCharm..... 20	4.4.4 命名规则..... 36
2.4.2 初始化 PyCharm..... 21	4.5 运算符..... 36
2.4.3 创建第一个项目..... 21	4.5.1 算术运算符..... 37
小结..... 25	4.5.2 比较运算符和逻辑运算符..... 37
习题..... 25	4.5.3 赋值运算符..... 38

4.5.4 复合赋值运算符	39	7.2.2 修改	69
4.5.5 位运算符	40	7.2.3 遍历	70
4.5.6 运算符优先级	46	7.2.4 查找	70
小结	47	7.2.5 内置函数	70
习题	47	7.3 List（列表）	71
第 5 章 控制语句	48	7.3.1 添加	71
5.1 执行结构	48	7.3.2 删除	71
5.2 控制语句	49	7.3.3 修改	72
5.2.1 顺序结构	49	7.3.4 排序和翻转	73
5.2.2 选择结构	50	7.3.5 推导式	74
5.2.3 循环结构	52	7.4 Dict（字典）	74
5.2.4 Break, Continue, Pass	54	7.4.1 访问	75
小结	55	7.4.2 修改	76
习题	55	7.4.3 添加	76
第 6 章 字符串与输入	56	7.4.4 删除	77
6.1 字符串表示	56	7.4.5 遍历	77
6.2 转义字符	57	7.5 嵌套	78
6.3 格式化字符串	58	7.6 字符串与 Tuple	79
6.4 字符串输入	61	7.6.1 访问	79
6.5 字符串运算	61	7.6.2 遍历	79
6.6 字符串内建方法	62	小结	79
6.6.1 count(sub[, start[, end]])	63	习题	80
6.6.2 find(sub[, start[, end]])	63	第 8 章 函数	81
6.6.3 isalpha() 和 isdigit()	63	8.1 认识 Python 的函数	81
6.6.4 join(iterable)	63	8.2 函数的定义和调用	81
6.6.5 lstrip([chars]), rstrip([chars]) 和 strip([chars])	63	8.2.1 定义	81
6.6.6 split(sep=None, maxsplit=-1)	64	8.2.2 调用	82
6.6.7 startswith(prefix[, start[, end]])	64	8.3 函数的参数	82
6.6.8 zfill(width)	64	8.3.1 默认参数	83
小结	64	8.3.2 任意位置参数	84
习题	65	8.3.3 任意键值参数	84
第 7 章 Tuple, List, Dict	66	8.3.4 组合使用	84
7.1 什么是数据结构	66	8.3.5 修改传入的参数	86
7.2 Tuple（元组）	67	8.4 函数的返回值	88
7.2.1 切片	67	8.4.1 返回一个值	88
		8.4.2 什么都不返回	88
		8.4.3 返回多个值	88
		8.5 函数的嵌套	89

8.5.1 作用域	89	10.2 PEP 484	120
8.5.2 global 和 nonlocal	90	10.2.1 类型提示	120
8.6 使用轮子	91	10.2.2 PyCharm 中的类型提示	121
8.6.1 随机库	91	10.2.3 扩展的类型提示	122
8.6.2 日期库	91	10.3 PEP 526	122
8.6.3 数学库	92	小结	123
8.6.4 操作系统库	92	习题	123
8.6.5 第三方库	92	第 11 章 面向对象编程	124
8.6.6 文档	92	11.1 面向对象	124
小结	93	11.1.1 类	124
习题	93	11.1.2 对象	125
第 9 章 实战 1: 2048 小游戏	94	11.1.3 类和对象的关系	125
9.1 规则简介	94	11.1.4 面向过程还是对象	126
9.2 创建项目	96	11.2 类的定义	127
9.3 代码设计	97	11.2.1 类名	127
9.3.1 棋盘存储	97	11.2.2 属性	127
9.3.2 棋盘移动	97	11.2.3 方法	128
9.3.3 用户交互	99	11.3 特殊的实例方法	131
9.4 代码实现	101	11.3.1 初始化和析构	131
9.4.1 棋盘定义	101	11.3.2 转字符串	132
9.4.2 棋盘初始化	101	11.3.3 实现运算符	133
9.4.3 棋盘移动——向下	102	小结	134
9.4.4 生成新数字	104	习题	134
9.4.5 其他方向判断和移动	104	第 12 章 封装、继承和多态	135
9.4.6 检查所有方向移动	107	12.1 封装	135
9.4.7 用户界面	108	12.1.1 使用 slots	135
9.4.8 用户操作处理	109	12.1.2 访问控制	135
9.4.9 处理用户的操作	110	12.2 继承	137
9.4.10 游戏主体逻辑	111	12.2.1 父类与子类	137
9.5 提升游戏体验	111	12.2.2 私有属性和方法	138
小结	113	12.2.3 单继承和多继承	140
习题	113	12.2.4 覆盖父类方法	140
第 10 章 Python 编码规范	114	12.2.5 调用父类方法	141
10.1 PEP 8	114	12.3 多态	142
10.1.1 代码布局	114	小结	143
10.1.2 命名规范	118	习题	143
10.1.3 自动检查调整	119		

第 13 章 生成器与迭代器	145	15.6.1 Pickle	169
13.1 初探迭代器	145	15.6.2 JSON	170
13.2 生成器	146	15.7 文件系统操作	172
13.3 迭代器	151	15.7.1 os.listdir(path='.')	172
13.3.1 可迭代对象	151	15.7.2 os.mkdir(path, mode=0o777)	172
13.3.2 迭代器	152	15.7.3 os.makedirs(name, mode=0o777, exist_ok=False)	172
13.4 生成器推导式	152	15.7.4 os.remove(path)	173
小结	153	15.7.5 os.rmdir(path)	173
习题	153	15.7.6 os.removedirs(name)	173
第 14 章 函数式编程	154	15.7.7 os.rename(src, dst)	173
14.1 匿名函数	154	15.7.8 os.path.exists(path)	173
14.2 高阶函数	154	15.7.9 os.path.isfile(path)	173
14.2.1 map	154	15.7.10 os.path.join(path, paths)	173
14.2.2 filter	155	15.7.11 os.path.split(path)	174
14.2.3 reduce	156	小结	174
14.3 闭包	156	习题	174
14.4 装饰器	158	第 16 章 异常处理	175
小结	162	16.1 什么是异常	175
习题	162	16.1.1 错误 (Bug)	175
第 15 章 文件读写	163	16.1.2 异常 (Exception)	175
15.1 打开文件	163	16.2 捕获异常	176
15.1.1 file	163	16.3 抛出异常	178
15.1.2 mode	164	16.3.1 raise	178
15.1.3 encoding	165	16.3.2 调用栈	179
15.2 关闭文件	165	16.3.3 内建异常	179
15.3 读文件	166	16.3.4 传递异常	180
15.3.1 read	166	16.4 断言	181
15.3.2 readline	166	16.5 实现自定义异常	182
15.3.3 readlines	167	16.6 with 关键字	183
15.3.4 迭代	167	16.6.1 上下文管理器	183
15.4 写文件	168	16.6.2 上下文管理协议	183
15.4.1 write 和 writelines	168	16.6.3 一个例子：计时器	184
15.4.2 flush	168	小结	184
15.5 定位读写	168	习题	185
15.5.1 tell	168	第 17 章 模块和包	186
15.5.2 seek	168	17.1 模块	186
15.6 数据序列化	169		

17.1.1 为什么需要模块	186	应用场景	194
17.1.2 模块的导入	186	基本用法	194
17.2 编写模块	187	创建机器人	194
17.2.1 第一个模块	187	给好友发送消息	195
17.2.2 <code>__name__</code>	188	保持运行	195
17.2.3 重载模块	189	试一试	195
17.3 包	189	决定“吃啥”	197
17.3.1 为什么需要包	189	关键词回复	198
17.3.2 第一个包	189	智能聊天	198
17.3.3 打包与导入	191	完整代码	200
小结	193	小结	201
习题	193	习题	201
第 18 章 实战 2：微信聊天机器人	194	参考文献	202
wxpy	194		



第 1 章 欢迎来到 Python 的世界

在这个信息化的时代，时常能听到这样的问题：

“我不是学编程的，我想入门该从什么学起？”

“我有一些重复性的工作想让机器自动完成，我用什么工具比较好？”

“我听说 Python 很简单，我该从何学起呢？”

.....



01 扫码看视频

对于这些问题，并没有唯一的答案，但是选择 Python 一定是一个正确的选择。接下来的介绍文字不是很长，也不是很深奥，如果你拿起这本书的时候也有相同或相似的问题，那么相信读完本章后你心里就会有一个明确的学习方向。

1.1 Python 是什么

1.1.1 Python 是一门语言

Python 是一门语言，但是这门语言跟现在印在书上的中文、英文这些自然语言不太一样，它是为了跟计算机“对话”而设计的，所以相对来说，Python 作为一门语言更加结构化，表意更加清晰简洁。

但是别忘了，在异国的时候，需要一名翻译员来把你的语言翻译成当地语言才能沟通的，想在计算机的国度里用 Python 和系统沟通，也需要一个“解释器”来充当翻译员的角色，在后面的章节中我们就可以看到怎么“请来”这个翻译员。

1.1.2 Python 是一个工具

工具是让完成某件特定的工作更加简单高效的一类东西，比如中性笔可以让书写更加简单，鼠标可以让计算机操作更加高效。Python 也是一种工具，它可以帮助我们完成计算机日常操作中繁杂重复的工作，比如把文件批量按照特定需求重命名，再比如去掉手机通讯录中重复的联系人，或者把工作中的数据统一计算一下等，Python 都可以把我们从无聊重复的操作中解放出来。

1.1.3 Python 是一瓶胶水

胶水是用来把两种物质粘连起来的東西，但是胶水本身并不关注这两种物质是什么。Python 也是一瓶这样的“胶水”。比如现在有数据在一个文件 A 中，但是需要上传到服务器 B 处理，最后存到数据库 C，这个过程就可以用 Python 轻松完成，（别忘了 Python 是一个工具！）而且我们并不需要关注这些过程背后系统做了多少工作，有什么指令被 CPU 执行——这一切都被放在了一个黑盒子中，只要把想实现的逻辑告诉 Python 就够了。

1.2 获取 Python

不同的设备与不同的系统都可快速获得 Python，即使只是手机，也可以体验 Python！

1.2.1 Windows

本节假设使用的是 Windows 10 系统，事实上对于 Windows XP 以后的所有 Windows 系统，操作是完全相同的。

首先打开浏览器，访问 Python 的官方网站 <https://www.python.org/>，如图 1-1 所示。将鼠标指针移动到 Downloads 上就会出现一个下拉框，选择 Python 3.x.x 下载即可，本书以 Python 3 为基础。



图 1-1 Python 官方网站

如果需要其他版本的话，也可以去 <https://www.python.org/downloads/windows/> 选择自己喜欢的版本，但是一般来说我们只要选择最新的版本就可以了。接下来就以 Python 3.6.4 的安装为例。

图 1-2 所示的是 Python 安装器启动后的界面。



图 1-2 安装界面

这里请勾选上“Add Python 3.6 to PATH”，方便之后直接在命令行使用，然后进入如

图 1-3 所示的 Python 的安装过程。

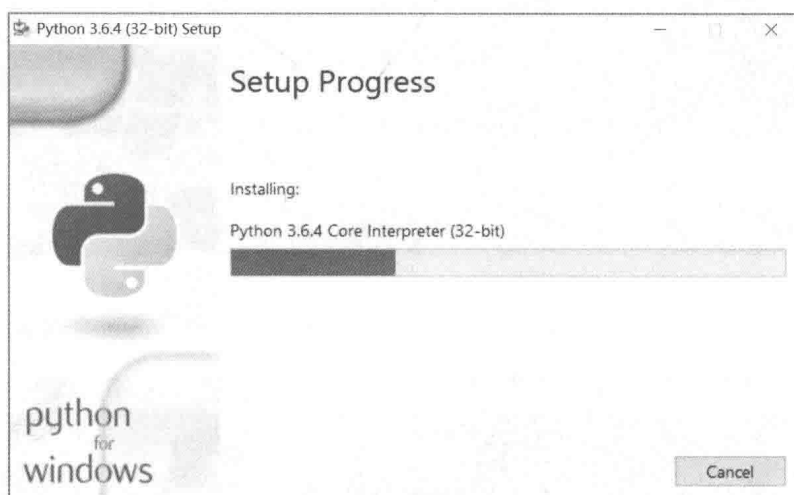


图 1-3 安装过程

直到出现图 1-4 所示的界面就说明安装完成了。

Python 的“翻译员”与其他软件有些区别，为了启动这个“翻译员”我们需要先启动一个命令提示符，这里方便起见我们直接按下组合键 Win+R 调出运行，然后输入“cmd”来启动它，如图 1-5 所示。



图 1-4 安装完成

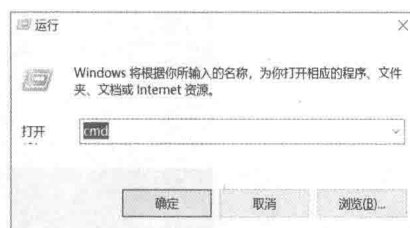


图 1-5 运行启动界面

选择“确定”后就可以看到命令提示符了，如图 1-6 所示。



图 1-6 命令提示符的启动页面

接下来就是请出这位“翻译员”了，只要输入“py”即可，启动后结果如图 1-7 所示。



图 1-7 启动 Python

如果看到这些输出就说明“翻译员”安装完成了。

1.2.2 Linux

实际上绝大多数 Linux 发行版已经自带了 Python，比如在 Ubuntu 中启动终端后只要输入“python3”就可以启动 python 解释器，如图 1-8 所示。

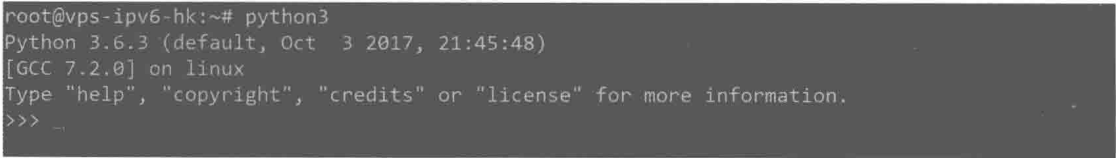


图 1-8 Ubuntu 下的 Python

如果系统中没有安装 Python 的话，可以用相应的包管理器来安装，比如 Debian 系列 Linux 发行版使用 `apt-get install python3`，Fedora 系列 Linux 发行版使用 `yum install python3`，Arch 系列 Linux 发行版使用 `Pacman -S python3` 即可。

1.2.3 macOS

在 macOS 下最好不要直接使用官网的 Installer，而是使用一款名为 Homebrew 的包管理器。对于所有 mac 用户来说，Homebrew 可以大大提升 macOS 的使用体验。

首先我们需要获取 Homebrew，打开 Homebrew 的官网可以看到如图 1-9 所示的内容。



图 1-9 Homebrew 官方网站

按照官网提示的操作，先复制中间的一行指令。

然后我们使用 Spotlight 打开终端，只要按下 Command + Space 组合键就可以看到如图 1-10 所示样式的搜索框。



图 1-10 Spotlight 搜索框

接下来在这个输入框中输入“Terminal”，如图 1-11 所示，可以看到第一个选项就是终端。

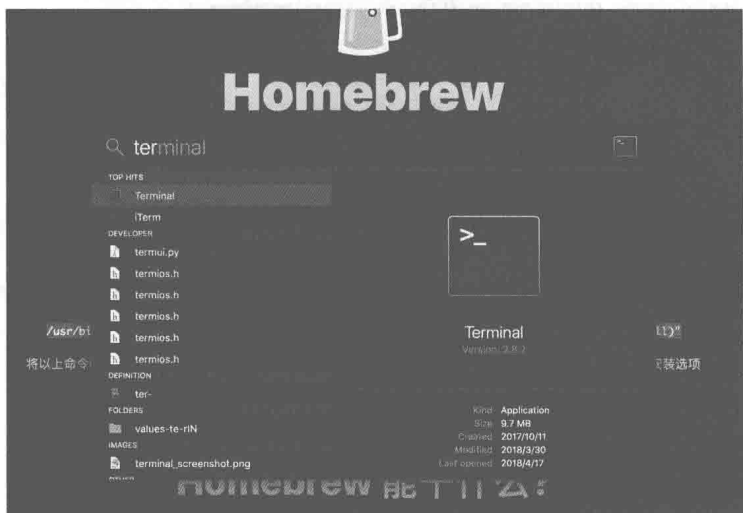


图 1-11 Spotlight 搜索 Terminal

按下回车键，终端就会启动，可以看到图 1-12 所示的白色窗口。

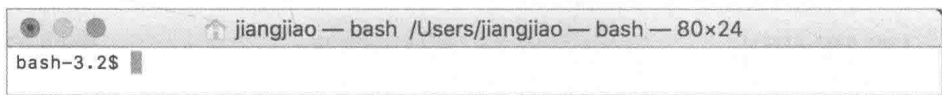


图 1-12 终端窗口

把刚才复制的指令粘贴进去回车执行，然后会暂停在图 1-13 所示的这一步。

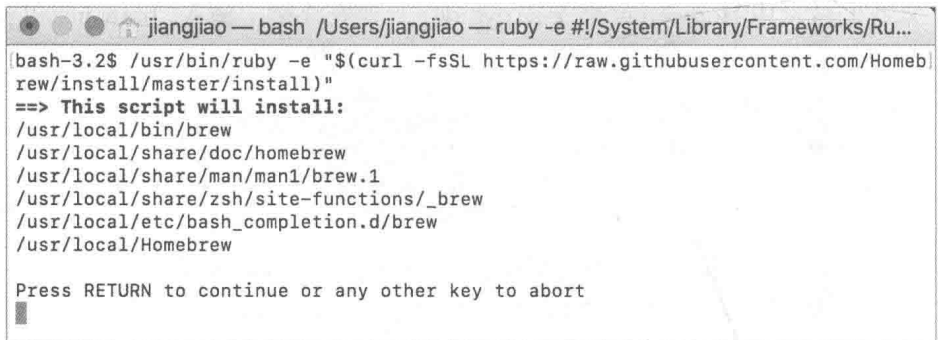


图 1-13 安装 Homebrew

这里的意思是 Homebrew 会在这些目录下进行安装，如果继续安装就按下回车键，否则按下其他任何键取消安装。这里只要正常按下回车键就可以了，然后程序会第 2 次暂停，因为需要输入用户的密码，如图 1-14 所示。

```
jiangjiao — bash /Users/jiangjiao — sudo • ruby -e #!/System/Library/Frameworks/
bash-3.2$ /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
==> This script will install:
/usr/local/bin/brew
/usr/local/share/doc/homebrew
/usr/local/share/man/man1/brew.1
/usr/local/share/zsh/site-functions/_brew
/usr/local/etc/bash_completion.d/brew
/usr/local/Homebrew

Press RETURN to continue or any other key to abort
==> /usr/bin/sudo /bin/mkdir -p /Library/Caches/Homebrew
Password: [ ]
```

图 1-14 输入密码

输入密码后再按回车，稍等一段时间后可以看见 “Installation successful!” 的字样，表示安装完成，如图 1-15 所示。

```
jiangjiao — bash /Users/jiangjiao — bash — 80x24
==> Cleaning up /Library/Caches/Homebrew...
==> Migrating /Library/Caches/Homebrew to /Users/jiangjiao/Library/Caches/Homebr
==> Deleting /Library/Caches/Homebrew...
==> New Formulae
brewsci/bio/andi                                brewsci/bio/centrifuge
brewsci/bio/cd-hit                               brewsci/bio/idba
==> Updated Formulae
boost ✓      diff-pdf      kompose      mysql@5.6      scala@2.11
abcmidi      freetds       libuv        osquery        shml
akamai       gpgame        lolcat       pdftoedn       xxhash
ansible      gradle        monero       pdftoipe
atlassian-cli heroku        mysql        phpunit
caddy        juju          mysql@5.5    poppler
==> Installation successful!

==> Homebrew has enabled anonymous aggregate user behaviour analytics.
Read the analytics documentation (and how to opt-out) here:
https://docs.brew.sh/Analytics.html

==> Next steps:
- Run `brew help` to get started
- Further documentation:
https://docs.brew.sh
bash-3.2$
```

图 1-15 安装成功

接下来我们就利用 Homebrew 来安装 python，只要在终端中输入 “brew install python3” 就可以开始安装 python 了，如图 1-16 所示。

```
jiangjiao — bash /Users/jiangjiao — bash — 80x24
bash-3.2$ brew install python3
```

图 1-16 安装 Python

稍等一段时间后看到图 1-17 所示的 Summary，就说明 Python 安装完成了。

然后我们继续在终端中输入 “python3” 就可以启动 python 了，它的输出和 Windows 上基本一致，如图 1-18 所示。

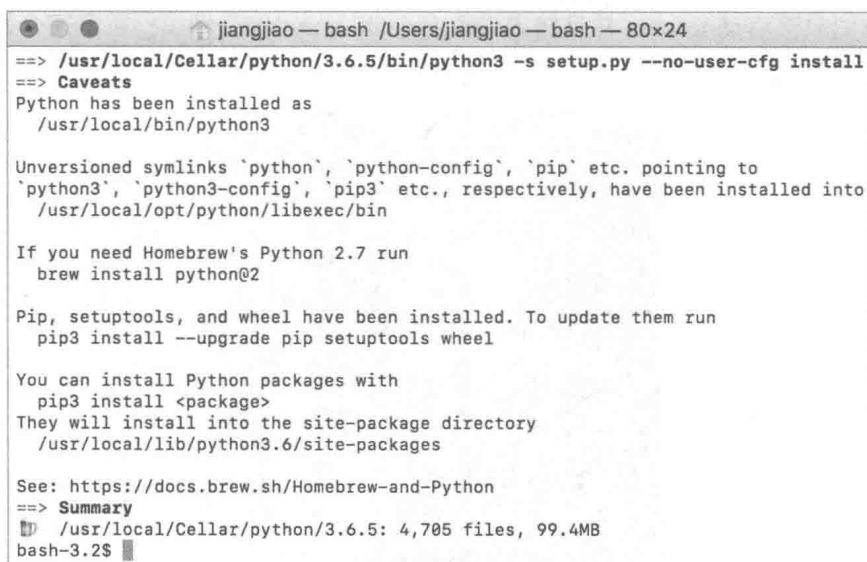


图 1-17 安装完成

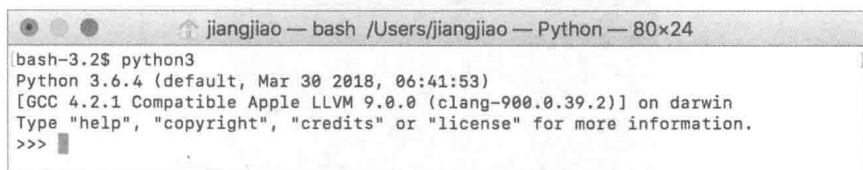


图 1-18 启动 Python

看到这个界面即表示安装成功。

1.2.4 Android

在 Android（安卓）上用到的是一款终端模拟软件 termux，获得这个 App 的方法很多，推荐从酷安等应用市场快速获取，如图 1-19 所示。



图 1-19 酷安 termux 页面