

Python

程序设计基础实战教程

韦玮 / 著



Abc



- 系统性学习、由简入难、通俗易懂
- 赠送视频课程、项目丰富





Python

程序设计基础实战教程

韦玮 / 著

清华大学出版社

北京

内 容 简 介

本书内容由浅入深,覆盖了绝大部分 Python 基础方面的知识,体系性较强,每个章节都基于各知识点编写了相应的 Python 程序实例,注重读者编程能力的培养。

这是一本定位于 Python 3 入门的书籍,适合没有 Python 编程基础,但是又想学习 Python 的读者使用。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Python 程序设计基础实战教程/韦玮著. —北京:清华大学出版社,2018(2018.5重印)
(清华科技大讲堂)
ISBN 978-7-302-48626-8

I. ①P… II. ①韦… III. ①软件工具—程序设计 IV. ①TP311.561

中国版本图书馆 CIP 数据核字(2017)第 261158 号

责任编辑:贾 斌 薛 阳

封面设计:刘 健

责任校对:焦丽丽

责任印制:沈 露

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:三河市君旺印务有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:17 字 数:408 千字

版 次:2018 年 1 月第 1 版 印 次:2018 年 5 月第 2 次印刷

印 数:2001~4000

定 价:45.00 元

产品编号:073118-01

1. 关于本书

Python 是一门非常简洁优美的编程语言,不管读者是否有编程基础,都可以很快地入门 Python。

同时,Python 还是一门近乎“全能”的编程语言,比如,我们可以使用 Python 进行数据采集,也可以使用 Python 进行 Web 开发,还可以使用 Python 进行数据分析与挖掘,进行量化投资分析,进行自动化运维等。

所以,总的来说,Python 是一门非常容易入门,并且功能非常强大的编程语言。我们可能会听到“人生苦短,我用 Python”之类的说法,这样的说法也是不无道理的,因为我们使用 Python 进行编程,不管是从学习的角度,还是从项目开发的角度来说,都可以节约很多时间。

千里之行,始于足下。

如果要使用 Python 进行常规项目的开发,或者应用到各个不同的领域(比如数据采集、Web 开发、数据挖掘等),必须首先掌握好 Python 编程的基础,只有扎实地掌握好 Python 编程基础之后,才能够更灵活地将 Python 运用于各方面。

正如本书的名字一样,这本书只讲 Python 的基础编程方面的知识,关于 Python 在各领域更多的应用方面的知识,将在本系列图书的后面几本书中分别详细介绍。

如果对 Python 有些了解的朋友,会知道 Python 目前有 Python 2. x 和 Python 3. x 的版本。并且 Python 2. x 与 Python 3. x 的编程规则在很多地方都有变动(这一点跟其他编程语言不太一样),也就是说 Python 2. x 与 Python 3. x 版本的承接性不是太好,考虑到 Python 2. x 比较稳定,Python 3. x 比较新并且越来越成熟,各有各的优势,在笔者综合考虑之后,本书一律采用 Python 3. x 进行写作。

本书的主要特点是:系统化、实战化。

笔者一直坚信,其实学习任何知识都不难,关键是要集中一段时间沉下心去系统地学习相关的知识,如果零散地学习各知识点,事实上会让你越学越感到迷茫,如果系统地学习,构建好自己的知识体系,会让你事半功倍。所以,建议你拿到一本书的时候,首先要做的事情是熟悉目录,因为相关的知识点基本上在目录中就有体现,熟悉目录的目的,是让你在心中初步建立一套知识体系,再学习的时候至少知道学到哪了,接下来会学什么。同时,当以后你遇到新的知识点,而本书没有讲到的时候,你完全可以将相关知识点添加到你的知识体系中的某个合适位置,这样,非常有利于对整个知识系统进行全局的把控。如果坚持建立知识体系的习惯,就会逐渐培养出全局意识出来,同时也会发现掌握知识会快很多,此外还有很多好处大家都会逐渐感受到。

其次,这本书每章都会结合具体的编程实例进行讲解,并尽量对编程实例的安排把握由

浅入深、层层递进的原则,让大家可以更好地接受,建议一定要把相关的代码自己动手敲一遍,并且如果基础不算太好,最好能够合上书,在理解的基础上默写敲一遍,这样,可以让你以后运用代码能力更强,说白了就是将现实世界的需求转化为代码的编程能力更强。

总之,系统化、实战化这两点也希望大家能够运用在其他各种知识的学习上,持之以恒,一定可以让你的学习能力变得更强。

综上,本书是一本定位于 Python 初学者,主要对 Python 基础知识进行实战讲解的书籍,如果你想零基础入门 Python,系统掌握 Python 基础编程的知识,为后续将 Python 运用在各领域的开发打下基础,那么,本书将适合你。

2. 本书目标读者

- Python 初学者;
- 高校计算机专业学生;
- 编程爱好者;
- 其他对 Python 感兴趣的人员。

3. 如何阅读本书

第 1~3 章主要介绍 Python 基本概述与基础编程方面的内容,包括 Python 基本介绍、Python 开发环境搭建、Python 基础语法、数据类型与运算符方面的内容。

第 4 章主要介绍 Python 的几种典型控制结构,事实上,控制结构在编程中非常重要,对于这一部分内容建议重点掌握,要求掌握得非常熟练,尤其是循环结构部分。

第 5 章和第 6 章主要介绍 Python 中稍微复杂一些的基础知识,包括迭代与生成、函数、模块等基础知识。

第 7 章和第 8 章主要介绍 Python 面向对象编程方面的知识,对于这一部分的知识尽量用了比较通俗的案例进行讲解,希望大家可以更好地掌握,因为后续如果想做一些大型的项目,常常会用面向对象的编程思想去编程。

第 9~12 章主要介绍 Python 基础中的一些提升部分的知识,主要包括正则表达式、数据库操作、文件操作、异常处理等,这一部分的知识事实上我们在实际项目中会常常遇到,用得非常多,是基础提升的关键部分。

第 13 章主要为大家介绍一个火车票查询与自动订票的项目,主要目的是希望读者可以运用之前学过的基础知识完成这个项目,将基础知识运用于项目开发实践。

第 14 章主要介绍了一个 2048 小游戏项目,主要目的是希望读者可以通过此 2048 小游戏项目,熟练掌握 Python 的基础知识,将 Python 基础知识融会贯通,并完成一个好玩的小游戏项目,培养综合运用知识的能力。

通过这 14 章的学习,目的是希望读者可以对 Python 基础有一个全面的掌握,同时,书中涉及的代码,希望读者可以自己手动输入一遍,这样可以更好地掌握相关知识。

4. 勘误与支持

由于作者水平有限,书中难免有一些疏漏或不准确的地方,恳请各位读者不吝指正。

相关建议可以通过微博@韦玮 pig 或微信公众平台正版韦玮(可以直接扫描下方二维码

二维码添加)进行反馈,也可以直接向邮箱 ceo@iqianyue.com 发送邮件(标题请注明一下:勘误反馈一书名),期待能够收到各位读者的意见和建议,欢迎来信。

5. 致谢

感谢清华大学出版社魏江江主任与编辑贾斌老师,是他们的鼓励与支持,才让我有了将这本书坚持写下去的毅力。

感谢 CSDN、51CTO 与天善智能,因为有他们,让我在这个领域获得了更多的学员与支持。

感谢很久以来一直支持我的学员们,平时公司的工作也比较忙,如果没有他们一直以来的支持,在业余时间去完成这么多课程的录制以及书籍的写作,确实太难,是他们的支持与包容,给予了我在这个领域一直走下去的动力,非常感谢大家!

特别感谢我的女友,因为编写这本书,少了很多陪她的时间,感谢她的不离不弃与理解,同时,也感谢她帮我完成书稿的校对工作,谢谢她的付出与支持。

特别感谢远方的父母、叔叔、姐姐、爷爷,也特别感谢所有支持我的朋友们,谢谢!

6. 配套视频与代码下载

所有赠送的视频课程与配套源代码可以分别扫描下列二维码观看或下载。



赠送视频扫描
上方二维码获取



配套源代码从上方
“资源下载”栏目获取

第 1 章 Python 概述	1
1.1 Python 的诞生	1
1.2 Python 的特点	2
1.3 Python 的功能	2
1.3.1 Python 常规应用	2
1.3.2 Python 在大数据时代下的应用	3
1.4 Python 的安装与配置	4
1.4.1 在 Windows 中搭建 Python 开发环境	4
1.4.2 在 MAC 中搭建 Python 开发环境	7
1.4.3 在 Linux 中搭建 Python 开发环境	8
1.5 编辑器的选用	11
1.5.1 常见的编辑器	12
1.5.2 Python 编辑器选用技巧	12
1.6 第一个 Python 程序	13
1.7 注释	14
1.8 小结	15
习题 1	16
第 2 章 基础语法	17
2.1 标识符	17
2.1.1 标识符的概念	17
2.1.2 标识符的命名规则	17
2.2 变量	18
2.2.1 变量的定义	18
2.2.2 变量的应用实践	18
2.3 保留字	19
2.4 行与缩进	20
2.4.1 行	20
2.4.2 缩进规律详解	20
2.5 小结	21
习题 2	21

第3章 数据类型与运算符	23
3.1 数字	23
3.2 字符串	24
3.3 列表	26
3.3.1 列表的定义	26
3.3.2 列表使用详解	27
3.4 元组	28
3.4.1 元组的定义	28
3.4.2 元组使用详解	28
3.4.3 列表与元组的区别	29
3.5 字典	30
3.6 运算符实践	31
3.7 运算符优先级规律与使用技巧	34
3.7.1 运算符优先级规律	34
3.7.2 运算符使用技巧	35
3.8 小结	35
习题3	36
第4章 条件控制与循环结构	37
4.1 程序执行流程概述	37
4.2 if 语句详解	37
4.2.1 几种常见的 if 语句格式及使用	38
4.2.2 if 语句的嵌套使用	39
4.3 while 语句详解	40
4.4 for 语句详解	41
4.5 循环的中断	42
4.5.1 break 语句	42
4.5.2 continue 语句	43
4.6 小结	44
习题4	45
第5章 迭代与生成	46
5.1 迭代器概述	46
5.2 迭代器常见使用	47
5.3 可迭代对象	48
5.4 自定义迭代器类	49
5.5 生成器概述与工作流程	51
5.6 生成器表达式	53

5.7 小结	54
习题 5	54
第 6 章 函数与模块	56
6.1 函数概述	56
6.2 函数的定义与调用	56
6.2.1 函数的定义	56
6.2.2 函数的调用	57
6.3 函数参数的传递与使用	58
6.3.1 形参与实参	58
6.3.2 参数的传递	60
6.4 函数返回值	61
6.5 变量作用域与变量类型	62
6.6 匿名函数	63
6.7 模块概述	64
6.8 Python 自带模块	65
6.9 自定义模块详解	67
6.10 小结	70
习题 6	70
第 7 章 类与对象	71
7.1 面向对象编程概述	71
7.1.1 面向过程编程与面向对象编程	71
7.1.2 面向对象编程的特点	72
7.2 类	72
7.2.1 类的概念	72
7.2.2 类的创建	73
7.3 对象	74
7.3.1 对象的概念	74
7.3.2 对象的创建	74
7.4 构造方法与析构方法	75
7.4.1 构造方法详解	76
7.4.2 析构方法详解	78
7.5 小结	80
习题 7	80
第 8 章 继承	82
8.1 子类与父类	82
8.2 单继承	82

8.3 多继承	84
8.4 小结	88
习题 8	88
第 9 章 正则表达式	90
9.1 正则表达式概述	90
9.2 原子	91
9.3 元字符	94
9.4 贪婪模式与懒惰模式	100
9.5 模式修正符	101
9.6 正则表达式函数	103
9.7 小结	104
习题 9	105
第 10 章 数据库操作实践	106
10.1 数据库操作概述	106
10.2 MySQL 数据库与 SQL 语句基础	107
10.2.1 MySQL 数据库服务器的安装	107
10.2.2 SQL 语句基础	111
10.3 Python 操作 MySQL 数据库实践	130
10.3.1 数据库的连接	130
10.3.2 使用 Python 执行 SQL 语句	132
10.4 Python 操作 SQLite3 数据库实践	137
10.5 小结	142
习题 10	143
第 11 章 文件操作	145
11.1 文件操作概述	145
11.2 目录操作实践	145
11.3 如何读取文件	148
11.4 如何写入文件	154
11.5 如何删除文件	157
11.6 小结	158
习题 11	159
第 12 章 异常处理技巧	160
12.1 Python 异常概述	160
12.2 如何抛出一个异常	161
12.3 自定义异常	163

12.4	异常处理及技巧	165
12.5	小结	170
	习题 12	170
第 13 章	12306 火车票查询与自动订票项目实践	172
13.1	火车票查询与自动订票项目功能分析	172
13.2	火车票查询与自动订票项目实现思路	173
13.3	火车票余票自动查询功能的实现	174
13.4	Cookie 处理实践	180
13.5	自动登录 12306 及验证码处理实践	181
13.6	自动获取个人中心页面信息实践	189
13.7	自动订票功能的实现——订单自动提交实践	191
13.8	自动订票功能的实现——订单自动确认实践	199
13.9	完整代码	206
13.10	调试与运行	214
13.11	小结	215
	思考与扩展	216
第 14 章	2048 小游戏项目实战	217
14.1	2048 小游戏项目介绍	217
14.2	2048 小游戏项目开发思路	219
14.3	实战编写 2048 小游戏项目基本代码结构	220
14.4	编写初始化方法与数字随机生成功能	222
14.5	棋盘与棋盘数据输出功能的实现	224
14.6	左滑与左滑合并功能的实现	226
14.7	右滑与右滑合并功能的实现	230
14.8	上滑与上滑合并功能的实现	233
14.9	下滑与下滑合并功能的实现	237
14.10	游戏按键监听功能的实现	240
14.11	编写主控程序	243
14.12	完善输赢判定与得分输出功能	245
14.13	完整代码	246
14.14	2048 小游戏的调试与运行	252
14.15	小结	255
14.16	思考与扩展	255

Python 是一门非常流行的语言,使用 Python 可以实现很多功能。目前 Python 主要有 Python 2.x 和 Python 3.x 两个版本系列,并且这两个系列的兼容性并不是太好。相对来说,Python 2.x 比较稳定,但经过多年的发展,Python 3.x 也逐渐变得越来越成熟,前景会更好,所以本书选择 Python 3.x 进行程序的编写。本章将主要介绍 Python 的一些基本情况并带领大家配置好基本的 Python 开发环境。

1.1 Python 的诞生

Python 于 1989 年发明,1991 年公开发行了第一个版本,创始人为 Guido van Rossum (吉多·范罗苏姆)。

Python 语言的设计参照了 C 语言、ABC 语言与 Modula-3。所以,如果读者有其他语言的编程基础,在学习 Python 的时候,会发现总有一种似曾相识的感觉,因为 Python 的一些基本语法相对来说还是沿袭了 C 语言的,所以,对很多程序员来说,会感觉 Python 的语法非常容易掌握。

虽然 Python 在很多语法上沿袭了 C 语言,但是 Python 语言的语法会比 C 语言更加简洁,又由于其参照了 ABC 语言与 Modula-3(两种非常强大而优美的语言),所以,Python 的设计也非常优美并强大。

同时,Python 是开源的,所谓开源,即开放源代码。这非常有利于 Python 的传播与使用,后续 Python 之所以会流行,也是与此分不开的。

Python 在设计之初就具有了比较完备的功能,比如:面向对象、各种常用的数据类型、函数、异常处理等。

20 世纪 90 年代的时候,计算机进入网络时代,并且计算机开始进入成千上万的家庭,用户增长非常快。Python 正好在这个时候出现,也自然就有了一个非常好的发展时机,由于 Python 的设计简洁优美、功能强大,所以受到大量程序员的喜爱与拥护,故而迅速地培养了一批忠实的粉丝用户。

人们在使用 Python 的时候,如果遇到问题,可以直接修改对应的 Python 语言的源代码(因为 Python 是开源的),同时,如果人们有较好的主意,也可以开发出对应的程序,然后提交给吉多·范罗苏姆,吉多·范罗苏姆可以选择采用这些程序加入到 Python 中,若被采用,这对程序员来说是莫大的荣耀。

在 2016 年 3 月的 TIOBE 编程语言排行榜上,Python 已经上升进入前 5 名。到今天为止,越来越多的人都在使用 Python,并且,由于 Python 在人工智能、大数据领域应用得非常好,再加上现在大数据发展速度非常快,所以 Python 的发展也非常迅速。

1.2 Python 的特点

上面已经介绍了 Python 的诞生,那么,Python 有哪些显著的特点呢?

总的来说,Python 的特点主要有:

- 简洁优美;
- 功能强大;
- 支持面向对象。

首先,对于 Python 语言简洁优美的特点,大家在未来学习本书的过程中就能够体会到,使用 Python 写程序,写出来的程序会非常简洁,并且,由于 Python 具有强制缩进的要求,所以写出来的 Python 程序也非常美观,可读性非常强。

其次,Python 虽然简洁,但是其功能是非常强大的。比如,Python 在人工智能、系统编程、Web 开发、网络爬虫等领域都有非常好的应用;Python 的可扩展性非常强,第三方库也非常丰富,使用 Python 可以做非常多的事情。

再者,Python 是一门支持面向对象的编程语言,这一点在开发大型项目的时候我们可以深刻感觉到它的优势。

总之,Python 是一门简洁优美、功能强大,支持面向对象的编程语言,好处非常多,同时也非常容易学习,相对于其他编程语言,用户可以更轻松学会 Python。

1.3 Python 的功能

前面已经介绍了 Python 的优势,这一节中重点介绍 Python 能够做一什么事情,将分为 Python 在常规时的应用与在大数据时代下的应用两方面进行介绍。

1.3.1 Python 常规应用

通常情况下,Python 可以做以下事情:

- (1) 进行简单脚本编程;
- (2) 进行系统编程;
- (3) 开发网络爬虫;
- (4) 进行 Web 开发;
- (5) 进行自动化运维;
- (6) 进行网络编程;
- (7) 进行数据挖掘、机器学习等大数据与人工智能领域方面的程序开发。

.....

由此可以看出,Python 能够做的事情是非常多的,小到开发一些简单的脚本程序,大到

机器学习等领域的应用。

在以上应用领域中,值得指出的是,Python 在网络爬虫、自动化运维、数据挖掘与机器学习等领域的应用尤为广泛。

比如,如果读者想做一个网络爬虫进行信息的自动采集,此时可以选择 Python 的 Urllib 库或者第三方爬虫库,如 Scrapy 等就可以很快地做出一个爬虫,然后使用该爬虫就可以进行信息的自动收集了,在搜集了对应的信息之后可以直接使用 Python 的正则表达式(re 模块)或者其他的数据筛选表达式实现数据的自动筛选,即可以大大地减轻人力劳动。

再比如,如果读者是做 Linux 运维方向的,平常只能依靠一些管理工具或者人力去进行服务器的运维,此时可以学习 Python,之后可以开发一些自动化运维的脚本或者程序去实现对服务器的自动运维与管理,同样可以大大减轻自己的负担与劳动。

除此之外,如果读者在做大数据、数据挖掘、机器学习等方向的工作也可以学习 Python,在掌握了 Python 之后,可以研究一些相应的算法,然后,在 Python 中,可以很方便地实现这些算法,同样,也可以很方便地解决各种业务场景中的问题,比如,如果需要对现有客户价值进行分类或分析,可以在学习了 Python 之后,使用 Python 实现相关的聚类算法,比如 K-Means 算法等,随后根据相应的算法对数据进行处理与分析,实现相应的需求的功能。

所以,可以看到,Python 在这些常见的领域中的应用是非常多的。除了这里所介绍的 Python 的应用外,Python 在其他领域中的应用也是非常广泛的,因为 Python 的库非常丰富,所以到后面我们会发现,用 Python 来实现各种各样的功能非常方便。

1.3.2 Python 在大数据时代下的应用

大数据时代下,云计算、数据分析与挖掘、人工智能等领域得到了极速的发展,而在这些领域中,我们可以使用各种语言实现所需要的功能,比如可以选择 Java 实现,也可以选择 C++ 实现,当然还可以选择 Python 来实现。

那么,这么多的语言应该如何选择呢?

影响我们做选择的因素主要有:

- (1) 你熟悉哪种语言?
- (2) 哪种语言实现起来相对来说比较方便简单?
- (3) 各语言的实现效率怎么样?

综合多种因素,我们会发现,如果我们已经学会了 Python,并且同时掌握了一些其他语言,此时,使用 Python 实现机器学习等方面的应用会相对来说简单很多。并且 Python 语言的执行效率虽不及 C++ 等更接近底层的编程语言,但是 Python 的执行效率也并不低,此外,如果任务量非常大,也可以使用多进程、多线程,以及分布式等技术对任务进行切分,此时并行处理这些任务,我们的速度也可以是非常快的。

同时,Python 关于数据挖掘、机器学习等相关算法等库也非常丰富,所以用户可以非常方便地实现这些相关的算法,甚至有些算法比较难,由于有了丰富的第三方模块,所以在 Python 中应用这些算法也是极为方便的。当然,在用户学到一定深度的时候,需要尽量尝试着编写一些新算法实现程序,或者也可以自己思考一些新的算法并实现。

现在 Python 无疑已经成为人工智能(AI)时代的首选语言。

Python 之所以能成为 AI 时代的首选语言,跟前面我们所分析的原因也是分不开的,在 AI 时代,如果有一种语言可以让我们去选择学习,那么 Python 必将是需要重点考虑的语言。

1.4 Python 的安装与配置

前面我们已经对 Python 进行了简单的介绍,接下来将为大家介绍如何安装好 Python 的开发环境。考虑到用户的计算机操作系统可能有所不同,所以本节中会分别为大家介绍 Windows、MAC、Linux 系统中 Python 开发环境的搭建。

1.4.1 在 Windows 中搭建 Python 开发环境

首先,我们需要选择一个 Python 版本下载对应版本的 Python,例如选择的 Python 版本为 Python 3.5.2,此时可以打开以下链接进行 Python 的下载:

<https://www.python.org/downloads/release/python-352rc1/>

打开了该页面之后,会发现此时有如图 1-1 所示可以下载的内容。

Files

Version	Operating System	Description
Gzipped source tarball	Source release	
XZ compressed source tarball	Source release	
Mac OS X 32-bit i386/PPC installer	Mac OS X	for Mac OS X 10.5 and later
Mac OS X 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later
Windows help file	Windows	
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64, not Itanium processors
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64, not Itanium processors
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64, not Itanium processors
Windows x86 embeddable zip file	Windows	
Windows x86 executable installer	Windows	
Windows x86 web-based installer	Windows	

图 1-1 可以选择下载的文件

在图 1-1 中可以看到,此时有非常多个文件。在这里,只需要关注以 executable installer 结尾的文件即可,可以看到以 executable installer 结尾的文件(以该字样结尾的文件意思是其为可执行文件安装包)主要有两个:

- (1) Windows x86-64 executable installer
- (2) Windows x86 executable installer

如果读者的计算机版本是 64 位的,可以下载使用安装包(1),如果读者的计算机版本是

32 位的,可以下载使用安装包(2)。

由于作者的计算机版本是 64 位的,所以下载安装包(1) Windows x86-64 executable installer,下载之后,只需要双击即可打开安装页面,当然,如果双击打开权限不够,可以右键,然后选择以管理员身份运行以打开安装界面。

打开安装页面后可以发现,会出现如图 1-2 所示的界面。

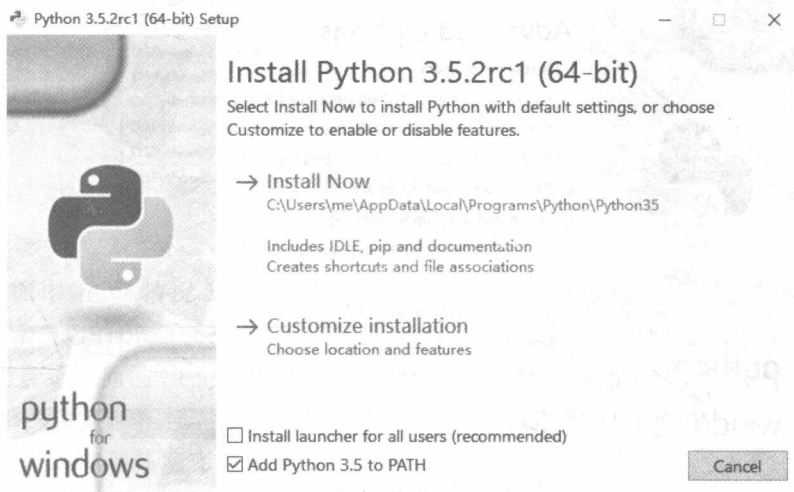


图 1-2 Python 安装步骤 1

在图 1-2 所示界面中,单击 Install Now 直接进行快速安装,当然也可以选择 Customize installation 进行用户自定义安装,我们推荐使用 Customize installation 选项进行安装,因为这样在后续可以自定义的地方会多一些,同时,建议勾选下方的 Add Python 3.5 to PATH,该勾选选项之后会自动将环境变量添加好,勾选好之后,单击 Customize installation 即可进入下一步。

进入下一步之后,随后会出现如图 1-3 所示的界面。

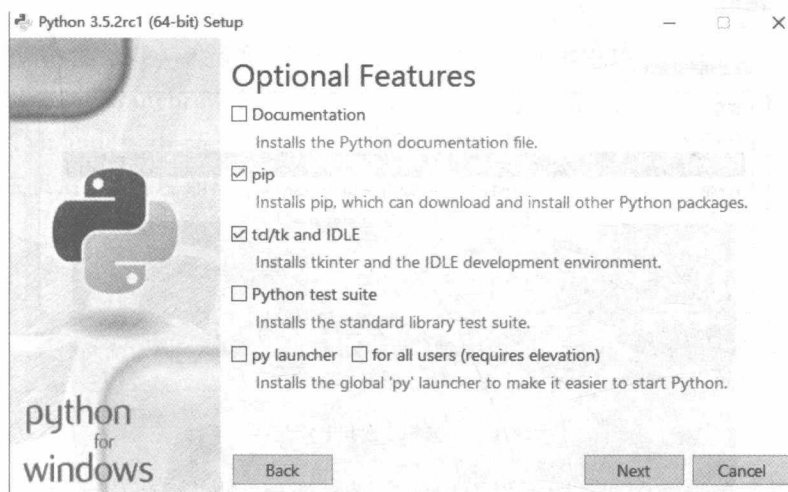


图 1-3 Python 安装步骤 2

在如图 1-3 所示的界面中,建议勾选 pip 与 tcl/tk and IDLE,因为这样就可以将这两者安装上,pip 在后续安装模块的时候用得非常多,而 IDLE 则是 Python 的自带编辑器,我们用得也会比较多。

勾选好这两项之后,单击 Next 按钮即可进入下一步,这时会出现如图 1-4 所示的界面。

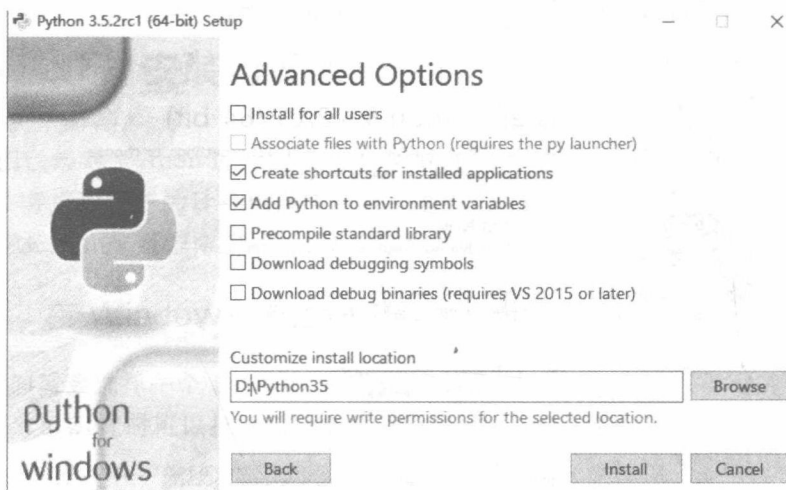


图 1-4 Python 安装步骤 3

在如图 1-4 所示的界面中可以设置安装目录,比如此时希望将 Python 安装到 D 盘下面的 Python35 目录中,可以将该路径设置为“D:\Python35”,然后,单击 Install 按钮即可进行安装,安装好之后就可以看到安装成功提示页面,只需要关闭该页面即可。

此时,系统已经安装好 Python 了。

有的时候,环境变量可能会由于系统或其他原因自动添加不上,此时,最好检查一遍环境变量的配置,如有问题,手动进行修改。

打开环境变量的配置界面后选择 PATH 变量,如图 1-5 所示。

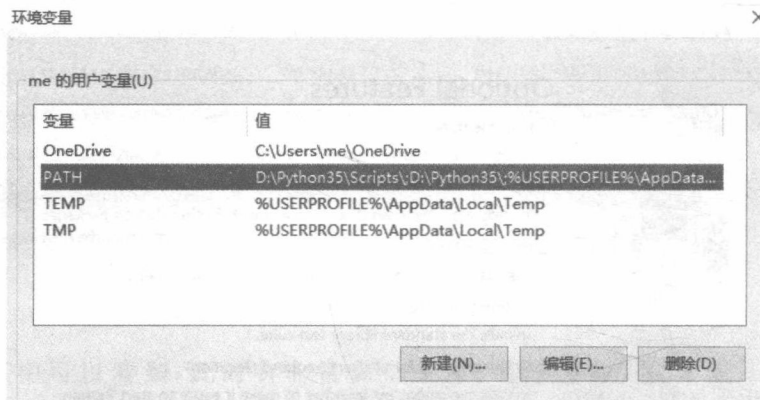


图 1-5 选择 PATH 变量

然后,只需要单击如图 1-5 所示界面中的“编辑”选项,即可编辑该环境变量,随后会打开如图 1-6 所示的界面。