

Python 3.x

全栈开发 从入门到精通

张云河 刘友祝 王硕◎著



北京大学出版社
PEKING UNIVERSITY PRESS



Python 3.x

全栈开发从入门到精通

张云河 刘友祝 王硕◎著



北京大学出版社
PEKING UNIVERSITY PRESS

内 容 简 介

全栈工程师,也称为全端工程师(同时具备前端和后台能力),是掌握多种技能,并能利用多种技能独立完成产品的人。Python全栈工程师,是指在精通Python编程语言的前提下,对于其他上下游的技术也有足够的了解和掌握。本书上下游的技术涉及数据传输、数据存储、数据分析和数据可视化等方面的知识。本书以实践的方式,将一系列的领域及理论知识结合到一起,帮助读者构建全栈开发的知识体系,并辅以精益及敏捷的思想,来一步步开发Web应用,让读者不只学会编码,还在学完后具备真实项目的经验。

图书在版编目(CIP)数据

Python 3.x全栈开发从入门到精通 / 张云河,刘友祝,王硕著. — 北京:北京大学出版社, 2019.5

ISBN 978-7-301-30308-5

I. ①P… II. ①张… ②刘… ③王… 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(2019)第034847号

书 名 Python 3.x全栈开发从入门到精通

PYTHON 3.X QUANZHAN KAIFA CONG RUMEN DAO JINGTONG

著作责任者 张云河 刘友祝 王 硕 著

责任编辑 吴晓月 王蒙蒙

标准书号 ISBN 978-7-301-30308-5

出版发行 北京大学出版社

地 址 北京市海淀区成府路205号 100871

网 址 <http://www.pup.cn> 新浪微博: @北京大学出版社

电子信箱 pup7@pup.cn

电 话 邮购部 010-62752015 发行部 010-62750672 编辑部 010-62570390

印刷者 北京大学印刷厂

经 销 者 新华书店

787毫米×1092毫米 16开本 35印张 893千字

2019年5月第1版 2019年5月第1次印刷

印 数 1-4000册

定 价 99.00 元

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究

举报电话:010-62752024 电子信箱:fd@pup.pku.edu.cn

图书如有印装质量问题,请与出版部联系。电话:010-62756370

前言



Python 是一门非常优秀的编程语言，其语法简洁，易学易用，越来越受到编程人员的喜爱；Python 也是一门非常“人性化”的编程语言，其各种语法规则的设计符合人们的思维方式，开发人员可以用最简单的方式实现自己的编程目的，降低时间成本。同时 Python 也是一门非常强大的编程语言，其在编程的各个领域都有非常不错的表现，如在网页开发、GUI 程序、网络爬虫、科学计算、数据可视化等领域，所有企业软件都可以使用 Python 进行一站式开发。本书中 Python 全栈开发涉及数据分析、数据存储、数据可视化、系统网络运维、Web 项目和量化交易等，对各部分的知识点都进行了详细介绍，深入浅出，帮助读者快速掌握全栈开发。



为什么要写一本这样的书

本书作者均来自开发和教育第一线，具备丰富的实际研发和培训经验。在对学校和企业的培训中，针对学校和企业的实际开发需要，定制了全套的 Python 全栈开发的课程。在实际的授课过程中根据企业和学生的反馈，不断调整 Python 全栈开发的内容。为了让更多的读者系统地学习 Python 全栈开发，本书作者决定抽出大量的时间来完成本书的创作，这就是本书的写作背景。

本书从项目开发经验入手，结合理论知识进行讲解。本书的目标就是让初学者快速成为一个合格的 Python 全栈开发工程师，并拥有项目开发技能，在未来的职场中有一个较高的起点。



本书结构

第 1 章是本书的快速入门部分，介绍 Python 3.x（为了更简洁，本书在描述时，Python 3.x 统称为 Python 3）的 Windows 和 Linux 安装环境，对 Python 全栈做了简要的介绍。

第 2 章是 Python 3 的基础知识部分，对 Python 的基本数据类型、函数、模块和面向对象编程做了介绍。

第 3 章是 Python 3 的高级知识部分，对 Python 的高级函数对象、多线程、多进程、正则表达式和发送邮件等做了介绍。

第 4 章是使用 Python 3 操作数据库部分，主要介绍了 MySQL、MongoDB 和 Redis 三种数据库的基本用法和使用 Python 3 对数据库进行操作。

第 5 章是 Python 3 网络编程部分，主要介绍了网络编程的基础知识和 Python 3 对 TCP 和 UDP 程序的设计。

第 6 章是 Python 3 的运维部分，主要介绍了 WebLogic 服务器的安装和使用，使用 Python 3 对配置服务器的 JNDI 和使用脚本自动部署 Java Web 的应用。在 Linux 操作系统下，使用 Python 脚本来监控 Linux 系统和 MySQL 数据库，进行日常的运维操作。

第 7 章是数据可视化部分，主要介绍了 NumPy、Pandas 和 Matplotlib 模块的使用和金融数据的可视化。

第 8 章是 Web 开发框架部分，主要介绍了 Flask Web 框架的使用并结合 ECharts 绘制基本图形。

第 9 章是 Python 在量化交易中的应用部分，主要讲解量化交易的基本概念和 Python 在量化交易中的作用。

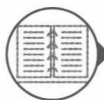
对于本书章节的内容，读者可以根据自身的实际情况有选择地阅读，毕竟 Python 全栈开发所涉及的范围比较广泛，读者可以挑选自己感兴趣的章节单独阅读。



读者对象

- 没有任何 Python 基础的初学者。
- 有一定 Python 基础，想精通 Python 全栈开发的人员。
- 有一定 Python 基础，没有项目经验的人员。

- 大专院校及培训学校的教师和学生。



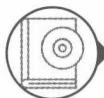
本书特色

零基础也能入门

无论您是否接触过 Python，是否使用 Python 开发过项目，都能从本书开启学习之旅。

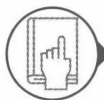
案例丰富

本书中的案例逐一讲解 Python 全栈的各种知识和技术，步骤详细、分析到位，能为读者入手真实项目打下良好的基础，轻松拥有项目经验。



配套资源

本书的所有代码都保存在 GitHub 上，后续代码更新也会以 GitHub 地址为准，网址是 <https://github.com/cxinping/PythonFullStack>，读者可自行下载，也可扫描下方二维码进行查看。另外为了方便读者交流，学习 Python 全栈开发，笔者建立了 Python 3 全栈开发高级群（QQ 群号：701141872）



致谢

作为资深软件架构师，能够编写一本技术性和实践性非常强的 Python 全栈开发图书，我感到非常荣幸。在此向所有给我提供指导、支持和鼓励的朋友表示衷心的感谢。

感谢我的亲人对我的关心与照顾，我现在取得的成果离不开你们对我的付出。

感谢北京大学出版社的魏雪萍和王蒙蒙两位编辑，是他们不辞辛苦，仔细严谨的审阅和校对工作为本书的顺利出版提供了有力保障。

感谢王登高（网名：量华林），他对量化系统的探索和执着体现在自研的程序化交易开

源框架 QuickLib。QuickLib 是一款优秀的开源框架，融入了他对量化交易的独到见解，它的功能强大，操作简单。在本书的编写过程中王登高对作者提供了无私的帮助，在此特别表示感谢。本书在编写过程中，得到了潭州教育的多位老师的帮助和鼓励，在此表示感谢。

与读者相识于 Python 全栈开发是一种缘分，能够看到本书说明读者对 Python 全栈开发是感兴趣的，感谢读者愿意花费时间阅读本书。希望每一位读者都能够通过阅读本书有所收获，真心祝愿你们能够学习顺利，事业有成。

目录

CONTENTS



第 1 章 初识 Python 语言 1

1.1 初识 Python..... 2

1.2 Python 2/Python 3..... 3

1.3 Python 全栈的兴起..... 4

1.4 安装 Python 3..... 6

1.4.1 在 Windows 下安装 Python 3 6

1.4.2 在 CentOS 下安装 Python 3 12

1.5 搭建 Python 3 开发环境 14

1.5.1 使用 PyCharm 新建项目 15

1.5.2 配置 PyCharm 17

第 2 章 Python 基础篇 20

2.1 变量 21

2.2 标准数据类型 22

2.2.1 数字 (Number) 22

2.2.2 字符串 (String)..... 24

2.2.3 列表 (List) 28

2.2.4 元组 (Tuple) 31

2.2.5 字典 (Dictionary) 33

2.2.6 集合 (Set) 35

2.3 标识符..... 36

2.4 行和缩进 37

2.5 运算符和表达式 37

2.5.1 算术运算符 38

2.5.2 比较运算符 38

2.5.3 逻辑运算符 39

2.5.4 成员运算符 40

2.6 条件控制 41

2.7 循环语句 42

2.7.1 while 循环 42

2.7.2 for 语句 43

2.7.3 使用枚举遍历序列 44

2.8 函数 44

2.8.1 函数的定义和调用 45

2.8.2	函数中的文档	46	2.10.6	可调用对象	69
2.8.3	默认参数	47	2.11	错误和异常	70
2.8.4	不定长参数	47	2.11.1	错误	70
2.8.5	range() 函数	48	2.11.2	异常	70
2.8.6	函数作为参数传递	50	2.11.3	处理异常	72
2.9	文件	51	2.11.4	打印异常信息	73
2.9.1	操作文件	51	2.11.5	自定义异常	74
2.9.2	使用 with 语句	56	2.12	模块	75
2.9.3	电子表格	57	2.12.1	导入模块	75
2.10	面向对象	59	2.12.2	模块的 <code>__name__</code> 属性	76
2.10.1	类与对象的定义	59	2.12.3	模块路径	78
2.10.2	私有属性和方法	61	2.12.4	包	80
2.10.3	继承	63	2.13	常用模块	81
2.10.4	静态方法	67	2.13.1	os 模块	81
2.10.5	魔法方法和特殊属性	67	2.13.2	time 模块	84

第3章 Python 高级篇 87

3.1	高级函数对象	88	3.3.1	Linux 平台下的多进程	118
3.1.1	lambda 函数	88	3.3.2	跨平台的多进程	120
3.1.2	map() 函数	89	3.3.3	跨平台的多进程间通信	127
3.1.3	reduce() 函数	90	3.3.4	分布式进程	129
3.1.4	迭代器 (Iterator)	92	3.4	正则表达式	135
3.1.5	生成器 (Generator)	95	3.4.1	re.match 函数	137
3.1.6	装饰器 (Decorator)	99	3.4.2	re.search 函数	140
3.2	多线程	104	3.4.3	re.findall 函数	141
3.2.1	多线程介绍	104	3.5	JSON 数据解析	142
3.2.2	线程模块	104	3.5.1	JSON 简介	142
3.2.3	守护线程	107	3.5.2	Python 处理 JSON 数据	144
3.2.4	优雅地停止线程	109	3.5.3	自定义对象的序列化	146
3.2.5	多线程的锁机制	112	3.6	存储对象序列化	148
3.2.6	本地线程变量	116	3.6.1	序列化对象	148
3.3	多进程	117	3.6.2	反序列化对象	149

3.7 发送 E-mail	150	3.7.2 发送 HTML 格式的邮件	154
3.7.1 发送简单邮件	152	3.7.3 发送带附件的邮件	155

第 4 章 使用 Python 操作数据库 158

4.1 操作 MySQL 数据库	159	4.3 操作 Redis 数据库	211
4.1.1 MySQL 简介	159	4.3.1 Redis 简介	211
4.1.2 在 Windows 下安装 MySQL	160	4.3.2 安装 Redis	211
4.1.3 在 Linux 下安装 MySQL	169	4.3.3 Redis 开启远程访问	217
4.1.4 MySQL 可视化工具	174	4.3.4 Redis 可视化工具	217
4.1.5 MySQL 基础知识	177	4.3.5 Redis 数据类型与操作	218
4.1.6 Python 操作 MySQL	182	4.3.6 使用 Python Redis 模块	223
4.2 操作 MongoDB 数据库	192	4.3.7 连接 Redis 服务器	224
4.2.1 MongoDB 简介	192	4.3.8 操作 string 类型	225
4.2.2 安装 MongoDB	193	4.3.9 操作 hash 类型	229
4.2.3 MongoDB 基本操作	199	4.3.10 操作 list 类型	231
4.2.4 MongoDB 的集合	200	4.3.11 操作 set 类型	233
4.2.5 MongoDB 的文档	202	4.3.12 操作 sorted set 类型	235
4.2.6 使用 Python 操作 MongoDB	206	4.3.13 其他操作	237

第 5 章 Python 网络编程 238

5.1 网络编程的基本概念	239	5.2.1 Socket() 函数	248
5.1.1 网络基础知识	239	5.2.2 TCP 程序设计	250
5.1.2 网络基本概念	240	5.2.3 UDP 程序设计	254
5.1.3 网络传输协议	245	5.2.4 Socket 实现文件传输	255
5.2 Python 3 网络编程	248	5.2.5 多线程与网络编程	261

第 6 章 Python 自动化运维 264

6.1 自动化运维简介	265	6.2 WebLogic 简介	266
--------------------------	-----	------------------------------	-----

6.3 安装 WebLogic.....	266	6.5.1 统计磁盘使用情况.....	311
6.3.1 安装 WebLoigc Server.....	267	6.5.2 统计内存使用情况.....	312
6.3.2 配置域 (Domain)	271	6.5.3 读取 passwd 文件中的用户名和 shell 信息.....	313
6.3.3 启动 WebLogic Server.....	276	6.5.4 统计 Linux 系统的平均负载.....	314
6.3.4 配置 JDK 环境变量.....	278	6.5.5 查看 CPU 信息.....	316
6.4 WebLogic 部署和配置.....	280	6.5.6 查看 MySQL 的慢日志 (slow-query-log).....	319
6.4.1 启动 WebLogic 脚本工具 (WLST)	280	6.5.7 监控 MySQL 的状态.....	321
6.4.2 通过网页部署 Java Web 应用.....	286	6.6 psutil 的使用.....	324
6.4.3 通过命令行部署工程.....	290	6.6.1 获取 CPU 信息.....	325
6.4.4 通过脚本部署 Java Web 应用.....	292	6.6.2 获取内存信息.....	326
6.4.5 通过网页配置 JNDI 数据源.....	297	6.6.3 获取磁盘信息.....	327
6.4.6 通过脚本配置 JNDI 数据源.....	307		
6.5 Python 在 Linux 运维中的常见应用... 310			

第 7 章 数据分析与可视化 329

7.1 NumPy.....	330	7.3 Matplotlib.....	363
7.1.1 安装 NumPy.....	330	7.3.1 安装 Matplotlib.....	364
7.1.2 创建矩阵.....	331	7.3.2 散点图.....	365
7.1.3 ndarray 对象属性.....	333	7.3.3 折线图.....	370
7.1.4 矩阵的截取.....	334	7.3.4 柱状图.....	374
7.1.5 矩阵的合并.....	336	7.3.5 直方图.....	379
7.1.6 通过函数创建矩阵.....	336	7.3.6 饼状图.....	383
7.1.7 矩阵的运算.....	339	7.3.7 Matplotlib 常用设置.....	386
7.1.8 保存和加载数据.....	342	7.3.8 子图 subplot.....	390
7.2 Pandas.....	345	7.3.9 多张图像 Figure.....	392
7.2.1 安装 Pandas.....	345	7.3.10 显示网格.....	394
7.2.2 Series	346	7.3.11 图例 legend.....	395
7.2.3 DataFrame	351	7.3.12 坐标轴范围.....	396
7.2.4 常用操作.....	356	7.3.13 坐标轴刻度.....	398
7.2.5 Pandas 操作 CSV 文件.....	358	7.3.14 调整坐标中日期刻度的显示.....	398
7.2.6 SQLAlchemy 操作数据库.....	360	7.4 金融绘图.....	400

7.4.1 获得股票数据源	400	7.4.2 显示股票历史数据	401
---------------------	-----	----------------------	-----

第 8 章 ▶ Python Web 开发框架.....406

8.1 Flask 简介	407	8.2.6 配置 Flask 应用集群	439
8.1.1 安装 Flask	408	8.3 ECharts 简介	444
8.1.2 最简单的 Web 应用	409	8.3.1 ECharts 轻松上手	446
8.1.3 路由	411	8.3.2 Flask 与 ECharts	450
8.1.4 HTTP 方法	414	8.3.3 柱状图	450
8.1.5 静态文件	416	8.3.4 折线图	452
8.1.6 模板渲染	416	8.3.5 饼状图	455
8.1.7 Request 对象	419	8.3.6 仪表盘	458
8.1.8 Session	424	8.3.7 可实时刷新的饼状图	460
8.1.9 保存 Session 到数据库	427	8.4 案例 1: 系统监控	463
8.2 Flask 应用集群	430	8.4.1 环境准备	464
8.2.1 分布式 Session	431	8.4.2 存储器	465
8.2.2 使用 jQuery	432	8.4.3 监控器	466
8.2.3 实验环境	434	8.4.4 路由器	467
8.2.4 配置 Redis	435	8.4.5 页面	468
8.2.5 配置 Nginx	435	8.5 案例 2: 动态显示销量	472

第 9 章 ▶ Python 在量化交易中的应用.....476

9.1 量化交易介绍	477	9.2 量化交易方案	489
9.1.1 量化交易的背景	477	9.2.1 期货量化交易环境介绍	489
9.1.2 可实现量化交易的市场比较	477	9.2.2 CTP 量化交易方案介绍	491
9.1.3 量化交易软件、平台、框架的 特点	479	9.2.3 行情数据采集	493
9.1.4 量化交易从哪个市场做起	481	9.2.4 期货 CTP 账户资金曲线监控和 绘制	496
9.1.5 量化交易策略类型	482	9.2.5 Quicklib CTP Python 框架	500
9.1.6 CTA 策略程序化交易指南	483	9.2.6 QuicklibTrade Python 接口	505
9.1.7 量化资源网站介绍	486	9.2.7 量化交易使用资管系统的好处和	

必要性	519	9.2.10 程序化实盘交易需要注意的	
9.2.8 高频交易	523	问题	538
9.2.9 算法交易	525		

附录 541

附录 A 使用 Postman 测试网络请求 542

附录 B 配置 Centos..... 547

1.1 初识 Python



Python 是一种面向对象、函数式编程的解释型程序设计语言，需要 Python 解释器进行解释运行。它可以运行在 Windows、Mac 和各种 Linux 系统上，Python 的语法简洁、清晰和灵活。Python 是一种脚本语言，也就是说，Python 程序需要在一个解释器中运行，这个解释器把程序翻译成计算机可执行的二进制代码，Python 的官方解释器称为 CPython。

Python 的编译速度超快，从诞生到现在已经有二十多年了。它具有丰富和强大的库，常被称为“胶水语言”，能够把用其他语言编写的各种模块（尤其是 C/C++）很轻松地联结在一起。其特点在于运用灵活，因为拥有大量第三方库，所以开发人员不必重复造轮子，就像搭积木一样，只要擅于利用这些库就可以完成绝大部分工作。

Python 是由荷兰计算机程序员 Guido van Rossum（吉多·范罗苏姆）创建的高级通用编程语言。Guido van Rossum 说过著名的一句话就是“Life is short, you need Python”（人生苦短，我用 Python），一直到现在在介绍 Python 语言时都会提及。他设计 Python 的目标如下。

（1）一门简单、直观的语言，并与主流开发语言一样强大。

（2）开源，以便任何人都可以为它做贡献。

（3）代码像阅读英文那样容易理解。

（4）适用于短期开发的日常任务。

（5）Python 的设计哲学是“优雅、明确、简单”。它的语言方式与自然语言很接近，具有很好的可读性，是适合初学者使用的编程语言。使用 Python 开发效率高、程序调试排错方便和跨平台方便。其具有以下特点。

（1）入门简单。任何熟悉 JavaScript 脚本、C 或 Java 的用户都能在短时间内熟练掌握 Python。

（2）功能强大。海量级的 Python 模块库提供了 IT 行业最前沿的开发功能。

- 数据分析：NumPy、Pandas 库在商业上已经逐步超越 R 语言。
- 机器学习：Scikit-Learn、TensorFlow 是国际上热门的机器学习平台。
- 数据库：无缝连接主流数据库 MySQL、Oracle 和 SQLServer。
- 游戏开发：Pygame 提供了图像、音频、视频、手柄、AI 等全套游戏开发模块库。
- 电脑设计：Maya、3DMax 都内置或扩展了 Python 语言支持。

（上面提到的 Pandas、Scikit-Learn、TensorFlow 是 Python 模块库）

Python 有丰富的第三方 Python 库、活跃的开源社区和完善的模块文档，第三方 Python 库都托管在 GitHub 上。GitHub 是一个面向开源及私有软件项目的托管平台，因为只支持 Git 作为唯一的版本库格式进行托管，故名 GitHub。

<https://github.com/>

Python 是一种动态语言，支持交互式编程、面向对象编程和函数式编程，具有类、函数、异常处理、列表 (List)、字典 (Dictionary) 和元组 (Tuple) 等数据类型。从 TIOBE 编程社区 2018 年 9 月的编程语言排行榜 TOP20 榜单来看，Python 编程语言排名第 3，前 2 名分别是 Java 和 C，如图 1-1 所示。

Sep 2018	Sep 2017	Change	Programming Language	Ratings	Change
1	1		Java	17.436%	+4.75%
2	2		C	15.447%	+8.06%
3	5	▲	Python	7.653%	+4.67%
4	3	▼	C++	7.394%	+1.83%
5	8	▲	Visual Basic .NET	5.308%	+3.33%
6	4	▼	C#	3.295%	-1.48%
7	6	▼	PHP	2.775%	+0.57%
8	7	▼	JavaScript	2.131%	+0.11%
9	-	▲	SQL	2.062%	+2.06%
10	18	▲	Objective-C	1.509%	+0.00%
11	12	▲	Delphi/Object Pascal	1.292%	-0.49%
12	10	▼	Ruby	1.201%	-0.64%
13	16	▲	MATLAB	1.276%	-0.35%
14	15	▲	Assembly language	1.232%	-0.41%
15	13	▼	Swift	1.223%	-0.54%
16	17	▲	Go	1.081%	-0.48%
17	9	▼	Perl	1.073%	-0.88%
18	11	▼	R	1.016%	-0.80%
19	19		PL/SQL	0.850%	-0.63%
20	14	▼	Visual Basic	0.682%	-1.07%

图 1-1 2018 年 9 月 TIOBE 编程语言排行榜 TOP20 榜单

注意

TIOBE 编程语言排行榜是编程语言流行趋势的一个指标，每月更新一次。这个排行榜的排名是基于互联网上有经验的程序员、课程和第三方厂商的数量，排名是使用著名的搜索引擎（如 Google、MSN、Yahoo!、Wikipedia、YouTube 及 Baidu 等）进行计算的。注意，这个排行榜只是反映某种编程语言的热门程度，并不能说明它好或不好，或者它所编写的代码数量多少。

1.2 Python 2/Python 3

Python 2 与 Python 3 虽然语法结构有些类似，但是却不能完全兼容。本书主要介绍的语言开发

环境是 Python 3，原因如下。

(1) 目前，Python 2 的绝大部分开源框架都提供了对 Python 3 的支持，并且一些新开源框架如 TensorFlow 等只提供了对 Python 3 的支持。

(2) 对于 Python 2，官方只支持到 2020 年，而 Python 的新开源框架往往不会对即将被淘汰的语言提供太多的支持，而且从 Python 2 到 Python 3 做了大量性能上的改进，符合语言发展规律，Python 3 会是以后的主流。Python 的新开源框架的模块库已经宣布了新功能不支持 Python 2，原有功能兼容 Python 2，如图 1-2 所示。



图 1-2 Python 提供的新开源框架模块库

(3) Python 3 默认使用 UTF-8 编码，对中文字符串无缝兼容。不用考虑中文编码的问题，可以节省大量的时间和精力，代码看起来也更加清晰。

```
# -*- coding: utf-8 -*-
```

现在使用 Python 3 是最好的选择，所以本书使用的 Python 开发环境为 Python 3.x，确切地说是 3.6.4 版本。

1.3 Python 全栈的兴起

全栈工程师，也称为全端工程师（同时具备前端和后台能力），是指掌握多种技能，并能利用多种技能独立完成产品的人，英文为 Full Stack Engineer。Python 全栈工程师是指在精通 Python 编程语言的前提下，对于其他上下游的技术也有足够的了解和掌握。之所以说 Python 适合全栈开发，

是因为 Python 遵循了“长板理论”。

“木桶理论”讲的是一只水桶能装多少水取决于它最短的那块木板，“木桶理论”对个人而言是要思考和补齐自己的短板，才能成功。到了互联网时代，根据“木桶理论”发展出了“长板理论”。“长板理论”是将木桶放在一个平面上，而且是把木桶放置在一个斜面上，木桶倾斜方向的木板越长，木桶内装的水就越多，长板理论如图 1-3 所示。互联网时代，知识爆炸、信息拥堵，对个人而言只有发挥自己的长处才能成功，而短板只需要找别人补齐就可以了。

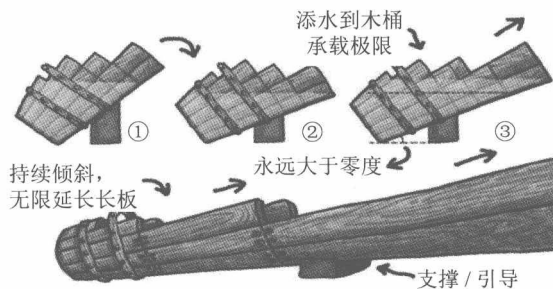


图 1-3 长板理论

Python 遵循了“长板理论”也是一样的道理。Python 的长处是语法简洁优美，功能强大，有丰富的第三方开源库可以使用，应用领域非常广泛，本书主要介绍以下方面的技术。

1. Web 项目开发

Python 在 Web 开发领域有成熟的 Web 架构来开发网站，如 Django、Flask 框架。可以快速完成一个网站的开发和 Web 服务，如国内的知乎、豆瓣等。

2. 数据处理

Python 被广泛地用于科学和数字计算中，如大数据处理、图像可视化分析等。

3. 系统网络运维

在运维的工作中，有大量重复性的工作需要做管理系统、监控系统、发布系统。使用 Python 编写脚本可以代替人工，在 Linux 下自动运维、监控、配置和部署等，提高工作效率。

4. 数据库存储

Python 标准数据库接口为 PythonDB-API，Python 数据库接口支持非常多的数据，如主流数据库 Oracle、MySQL 和 SQL Server 等。

以上这些都是本书重点讲解的内容，这些内容编写的思维导图如图 1-4 所示。