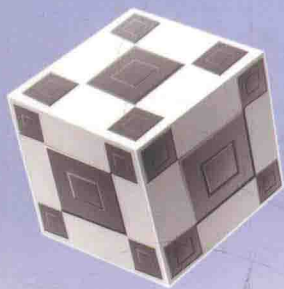


我的 Python 世界

玩《Minecraft 我的世界》 学 Python 编程

程晨 著



本书可以教你

- ✓ Python 语言的基础知识
- ✓ Python 编程入门
- ✓ 在《Minecraft》中制作交互游戏
- ✓ 通过游戏控制外部设备



中国工信出版集团

人民邮电出版社
POSTS & TELECOM PRESS

我的 Python 世界

玩《Minecraft 我的世界》 学 Python 编程

程晨 著



人民邮电出版社
北京

图书在版编目(CIP)数据

我的Python世界：玩《Minecraft我的世界》学
Python编程 / 程晨著. — 北京：人民邮电出版社，
2018.7

(创客教育)

ISBN 978-7-115-48434-5

I. ①我… II. ①程… 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(2018)第097277号

◆ 著 程 晨

责任编辑 周 明

责任印制 彭志环

◆ 人民邮电出版社出版发行

邮编 100164 电子邮件

网址 <http://www.ptpress.com.cn>

北京缤索印刷有限公司印刷

◆ 开本：690×970 1/16

印张：11.5

字数：235千字

2018年7月第1版

2018年7月北京第1次印刷



定价：69.00元

读者服务热线：(010)81055339 印装质量热线：(010)81055316

反盗版热线：(010)81055315

内容提要

Python 是一种解释型、面向对象、动态数据类型的高级程序设计语言，它具有丰富和强大的库，能够把用其他语言（尤其是 C/C++）制作的各种模块很轻松地联结在一起。这两年，随着人们对人工智能的关注越来越多，大家学习 Python 的热情也日益高涨。Python 在 IEEE 发布的 2017 年编程语言排行榜中高居首位。

本书介绍了 Python 编程的基础知识，并以游戏《Minecraft（我的世界）》为载体，通过 Python 编程来与游戏中的玩家或方块互动，先后实现了“剑球”游戏、五子棋游戏以及像素图像扫描仪这几个项目，最后还实现了通过游戏控制 Arduino 等外部设备的功能。希望大家能在玩游戏的过程中轻松地进入 Python 的世界，最终跨越软硬件的鸿沟，初步尝试自动化控制。

本书提供边玩游戏边学编程的全新体验，适合对 Python 编程感兴趣的读者阅读。游戏不再只是用来玩的，你将同时体验超级玩家、设计师和程序员的角色。

FOREWORD

前言

Python 是一种解释型、面向对象、动态数据类型的高级程序设计语言。它具有丰富和强大的库,能够把用其他语言(尤其是 C/C++)制作的各种模块很轻松地联结在一起。这两年,随着人们对人工智能的关注越来越多,大家学习 Python 的热情也日益高涨。Python 在 IEEE 发布的 2017 年编程语言排行榜中高居首位。

我第一次接触 Python 还是在诺基亚的塞班时代,它是为数不多的能够在塞班上编程的语言,当时我的感受就是它比较容易理解,不过我还没有真正学习它,它就被大家遗忘了。经过多年的发展,目前 Python 的功能已经非常强大了,作为一种高级语言,它具有丰富的第三方库,官方库中也有相应的功能模块支持,覆盖了网络、文件、GUI、数据库、文本等大量内容。

Python 可以在多种主流的平台运行,现在有很多领域都采用 Python 进行编程。目前业内绝大多数大中型互联网企业在使用 Python。

我现在也在针对青少年进行一些 Python 编程的教学工作,为了让大家对 Python 学习更感兴趣,我以学生比较喜欢的《Minecraft (我的世界)》游戏为载体,通过 Python 编程来与游戏中的玩家或方块互动,先后实现了“剑球”游戏、五子棋游戏以及像素图像扫描仪这几个项目,最后还实现了通过游戏控制 Arduino 等外部设备的功能。希望大家能够在游戏的过程中更加轻松地进入 Python 的世界。

本书的内容

本书大体上可以分为前后两部分:前面主要是一些 Python 的基础知识,包括基本的程序结构(顺序、选择、循环)、字符串、列表、字典、元祖、对象、类库等,这部分内容有 3 章;后面的部分主要是结合《Minecraft》完成一些交互游戏,包括 Minecraft 代码入门、“剑球”游戏、五子棋游戏、像素图像扫描仪以及通过游戏控制外部设备的一个小例子,这部分内容有 6 章。

本书面向的读者

目前市面上关于 Python 编程的书已经有不少了，不过大都是从编程语言的角度来介绍的。而本书以学生比较喜欢的《Minecraft》游戏为载体，所以应该更加简单易学。本书面向的是所有想学习 Python 语言的人，不过可能需要读者对于《Minecraft》游戏有一定了解，至少应该知道如何操作游戏中的玩家进行探险。

入门之后，大家可以再购买一些专业的 Python 书籍进行阅读，进一步学习与游戏开发、人工智能算法相关的内容。

为了更适合读者阅读，本书采用全彩印刷形式，这样后面的这些例子看起来会更加直观明了。这里要感谢人民邮电出版社的编辑在出版过程中付出的努力，最后还要感谢现在正捧着这本书的您，感谢您肯花费时间和精力阅读本书。由于时间有限，书中难免存在疏漏与错误，诚恳地希望读者批评指正，您的意见和建议将是我巨大的财富。

程晨

2018.3

本书配套程序请扫描二维码获取：



下载地址：box.ptpress.com.cn/y/RC2018000014

CONTENTS

目录

1	Python 基础.....	1
1.1	Python 的历史.....	1
1.2	Python 的发展.....	1
1.3	Python 的优缺点.....	2
1.3.1	Python 的优点.....	2
1.3.2	Python 的缺点.....	2
1.4	Python 的适用领域.....	3
1.5	Python 的安装与使用.....	3
1.5.1	Python 的安装.....	3
1.5.2	Python 的使用.....	5
1.5.3	编辑器.....	7
1.6	关键字.....	9
1.7	数字.....	10
1.8	变量.....	10
1.9	程序的基本结构.....	11
1.9.1	for 循环.....	12
1.9.2	if 选择结构.....	13
1.9.3	比较.....	14
1.9.4	逻辑运算.....	14
1.10	掷骰子.....	16
1.10.1	随机数.....	16
1.10.2	重复掷骰子.....	16

1.10.3	掷两个骰子	17
1.10.4	大小判断	18
1.11	While	20
2	字符串、列表和字典	22
2.1	字符串	22
2.1.1	字符串的定义	22
2.1.2	字符串的方法	22
2.2	列表	24
2.2.1	列表的定义	24
2.2.2	列表的方法	25
2.3	函数	26
2.4	猜词游戏	27
2.4.1	游戏规则	27
2.4.2	创建单词库	27
2.4.3	游戏结构	28
2.4.4	完善函数	29
2.5	字典	34
2.6	元祖	35
2.6.1	元祖的定义	35
2.6.2	多重赋值	35
2.6.3	多返回值	36
2.7	异常	36
2.8	函数汇总	37
2.8.1	数字	37
2.8.2	字符串	38
2.8.3	列表	39
2.8.4	字典	40
2.8.5	类型转换	40
3	类库和方法	42
3.1	库	42
3.1.1	使用 random 库	42
3.1.2	使用 Python 标准库	43

3.2	面向对象	44
3.3	定义类	44
3.4	继承	46
3.5	文件	48
3.5.1	读取文件	48
3.5.2	读取大文件	50
3.5.3	写入文件	51
3.5.4	文件系统操作	51
3.6	侵蚀化	52
3.7	网络	52
4	《Minecraft 我的世界》	54
4.1	《Minecraft》是什么	54
4.2	准备工作	55
4.3	欢迎来到《我的世界》	55
4.4	位置坐标	60
4.5	玩家的位置	60
4.5.1	获取玩家的位置	60
4.5.2	显示玩家的位置	61
4.5.3	设定玩家的位置	62
4.6	弹射区域	62
5	剑球游戏	64
5.1	建造球场	64
5.1.1	开辟空间	64
5.1.2	铺设球场	65
5.1.3	搭建球门	67
5.1.4	球场函数	67
5.2	击打事件处理	68
5.2.1	获取击打事件	68
5.2.2	确认方块的面	70
5.3	球的移动	71
5.3.1	击打方块不同的面	71
5.3.2	移动球的程序	73

5.4 异常情况	74
5.4.1 球多了	74
5.4.2 球没了	75
5.5 出界与进球	77
5.5.1 出界的判定	77
5.5.2 进球了	78
6 剑球进阶	81
6.1 特殊规则	81
6.1.1 规则介绍	81
6.1.2 移动的中线	81
6.1.3 最大偏移量	83
6.2 显示数字	83
6.3 球门上的比分	86
6.3.1 东侧的比分	86
6.3.2 西侧的比分	94
6.4 CSV 文件	99
6.4.1 什么是 CSV 文件	99
6.4.2 新建 CSV 文件	100
6.4.3 读取外部文件	100
6.4.4 优化代码	102
6.5 彩蛋: TNT 来了	106
6.6 显示时间	108
6.6.1 获取系统时间	108
6.6.2 显示形式	109
6.6.3 功能实现	109
7 五子棋	115
7.1 绘制棋盘	115
7.1.1 二维数组	115
7.1.2 代码实现	116
7.2 落子	117
7.2.1 获取击打事件	118
7.2.2 改变方块颜色	118

7.2.3 更改数组	118
7.3 五子相连	118
7.3.1 落子效果	118
7.3.2 重复落子问题	119
7.3.3 获胜判断	121
7.4 重新开始	128
7.4.1 结束后的提示	128
7.4.2 刷新棋盘	129
7.4.3 反复刷新的问题	130
8 图片扫描仪	131
8.1 像素画	131
8.2 项目介绍	132
8.3 操作菜单	133
8.3.1 获取输入信息	133
8.3.2 制作菜单	135
8.3.3 中文输入法	138
8.3.4 扫描仪菜单	138
8.4 扫描操作	142
8.5 复印操作	146
8.6 演示操作	148
8.6.1 原始内容制作	148
8.6.2 扫描	148
8.6.3 复印	152
8.7 完整的代码	154
9 硬件控制	160
9.1 Raspberry Pi 的硬件接口	160
9.1.1 连接 GPIO	160
9.1.2 数字输出	161
9.1.3 模拟输出	163
9.2 串口通信	164
9.2.1 什么是 Arduino	164
9.2.2 在 Raspberry Pi 上安装 Arduino	165

9.2.3 Arduino 与 Python 通信	165
9.3 比特开关	166
9.3.1 功能描述	166
9.3.2 准备工作	167
9.3.3 功能实现	167
附录 方块材质名称对照表	169

Python 基础

Python 是一种解释型、面向对象、动态数据类型的高级程序设计语言。它具有丰富的和强大的库，能够很轻松地把用其他语言（尤其是 C/C++）制作的各种模块联结在一起。这两年，随着对人工智能的关注越来越多，人们对 Python 的学习热情也越来越高涨。在 IEEE 发布的 2017 年编程语言排行榜中，Python 高居首位。本书以《Minecraft（我的世界）》游戏为载体，希望能够引领大家更加轻松（Python 本身就以简单易用著称）地进入 Python 的世界。

1.1 Python 的历史

Python 由 Guido van Rossum 于 1989 年底发明，第一个公开发行人版发行于 1991 年。他对这个叫作 Python 的新语言的定位是：一种介于 C 和 shell 之间，功能全面，易学易用，可扩展的语言。

这门语言之所以叫 Python（巨蟒），是因为 Guido van Rossum 是电视喜剧《巨蟒组的飞行马戏团》（Monty Python's Flying Circus）的狂热爱好者。该剧是英国的喜剧团体巨蟒组（Monty Python）创作的系列超现实主义电视喜剧，1969 年首次以电视短剧的形式在 BBC 电视频道播出，共推出了 4 季共 45 集节目。随后喜剧团体巨蟒组的影响力从电视扩展到舞台剧、电影、音乐专辑、音乐剧等，被外国媒体认为在喜剧上的影响力不亚于披头士乐队在音乐方面的影响。它的 6 位成员都是来自牛津大学和剑桥大学的高材生。除去 Python，以流行文化命名的程序语言还有不少，比如 Frink 语言的名字来自《辛普森一家》中的 Frink 教授。

1.2 Python 的发展

1991 年，第一个 Python 编译器诞生。它基于 C 语言实现，并能够调用 C 语言的库文件。之后历经不断的换代革新，2004 年 Python 到达了一个具有里程碑意义的节点——2.4 版。6 年后 Python 发展到 2.7 版，这是目前为止 2.x 版中使用较为广泛的版本。

2.7 版不同于以往的 2.x 版，它是 2.x 版向 3.x 版过渡的一个桥梁，在最大程度上继承了 3.x 版的特性，同时尽量保持对 2.x 版的兼容性。

在 Python 的发展历程中，3.x 版在 2.7 版之前就已经问世了。从 2008 年的 3.0 版开始，Python 3.x 呈迅猛发展之势，版本更新活跃，一直发展到现在最新的 3.6.4 版。

1.3 Python 的优缺点

1.3.1 Python 的优点

Python 有以下几个优点。

1. 简单优雅

这是 Python 的定位，使得 Python 程序看上去简单易懂，初学者容易入门，学习成本更低。但随着学习的不但深入，Python 同样可以满足复杂场景的开发需求。引用一个说法，Python 的哲学就是简单优雅，尽量写容易看明白的代码，尽量写更少的代码。

2. 开发效率高

Python 作为一种高级语言，具有丰富的第三方库，官方库中也有相应的功能模块支持，覆盖了网络、文件、GUI、数据库、文本等大量内容。因此开发者无需事必躬亲，遇到主流的功能需求时可以直接调用。在基础库的基础上施展拳脚，可以节省你很多时间和精力，大大缩短了开发周期。

3. 无需关注底层细节

Python 作为一种高级开发语言，在编程时无需关注底层细节（如内存管理等）。

4. 功能强大

Python 是一种前端、后端通吃的综合性语言，功能强大。

5. 可移植性

Python 可以在多种主流的平台运行，开发程序时只要绕开对系统平台的依赖性，就可以在无需修改的前提下运行在多种系统平台上。

1.3.2 Python 的缺点

Python 的缺点有以下几点。

1. 代码运行速度慢

因为 Python 是一种高级开发语言，不像 C 语言一样可以深入底层硬件，最大程度上挖掘、榨取硬件的性能，所以用它编写的程序运行速度要远远低于用 C 语言编写的程序的运行速度。另外一个原因是，Python 是解释型语言，代码在执行时会被一行一行地翻译成 CPU 能理解的机器码，这个翻译过程非常耗时，所以很慢。而 C 程序是在运行前直接编译成 CPU 能执行的机器码，所以运行起来非常快。

不过这种慢对于不需要追求硬件高性能的应用场合根本不是问题，因为它们比较的数量级根本不是用户能直观感受到的！

2. 必须公开源代码

因为 Python 是一种解释性语言，没有编译、打包的过程。所以必须公开源代码。

总体来讲，Python 的优点多于缺点，而且缺点在多数情况下不是根本性问题，所以现

在很多领域都采用 Python 进行编程。下面我们就来看看 Python 所适用的领域。

1.4 Python 的适用领域

Python 可应用于众多领域，具体可以分为以下几个方面。

(1) 云计算开发。Python 是云计算领域最火的语言，典型应用代表为 OpenStack。

(2) Web 开发。众多优秀的 Web 框架（如 Youtube、instagrm、豆瓣等）均使用 Python 开发。

(3) 系统运维。各种自动化工具，如 CMDB、监控报警系统、堡垒机、配置管理 & 批量分发工具等的开发均可用 Python 搞定。

(4) 科学计算、人工智能。据说用于人机围棋大战的 AlphaGo 就使用了 Python 进行开发。

(5) 图形 GUI 处理。

(6) 网络爬虫。现在很多网络爬虫，包括谷歌的爬虫都是使用 Python 开发的。

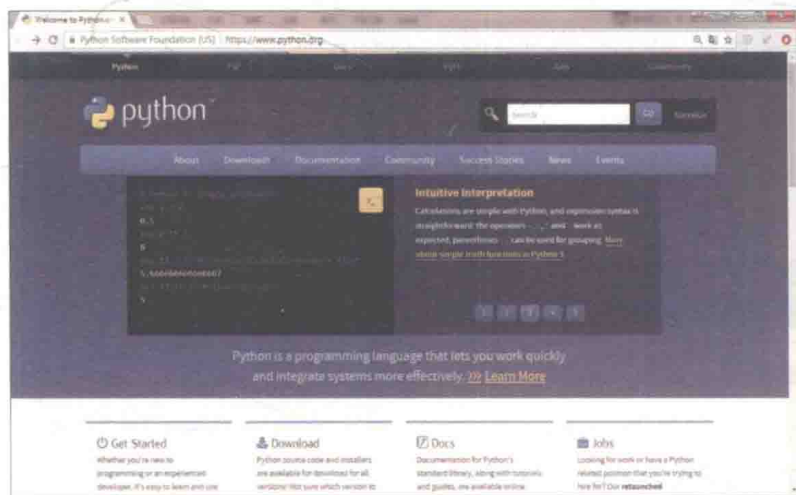
目前业内绝大多数大中型互联网企业在使用 Python。

1.5 Python 的安装与使用

虽然本书主要使用的是 Raspberry Pi（树莓派）上的 Python，而它又是 Raspberry Pi 自带的，不涉及安装问题，但是 Python 并不是只能在 Raspberry Pi 或 Linux 中使用，在 Windows 中也可以使用。本节我们就来介绍一下在 Windows 中如何安装 Python。

1.5.1 Python 的安装

首先，打开 Python 的官网，界面如图 1.1 所示。



■ 图 1.1 Python 官网

这个界面中心有一个黄色的按钮，单击这个按钮能够打开一个在线的控制台，如图 1.2 所示。

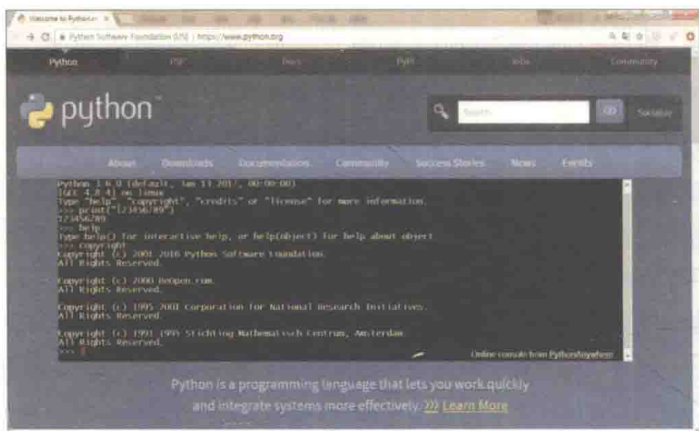


图 1.2 网页中的在线控制台

在这个控制台中，我们就能够初步地感受 Python 的应用，比如在这里输入 copyright 之后，控制台就会显示 Python 的版权信息。

控制台上方有一排选项按钮，将鼠标指针移动到 Downloads 上，就会弹出 Downloads 菜单下的选项，如图 1.3 所示，其中包含各个操作系统版本的 Python 的下载。

这里由于网页检测到现在使用的是 Windows 系统，所以在这些子选项的右侧会自动弹出 Windows 版 Python 的下载。

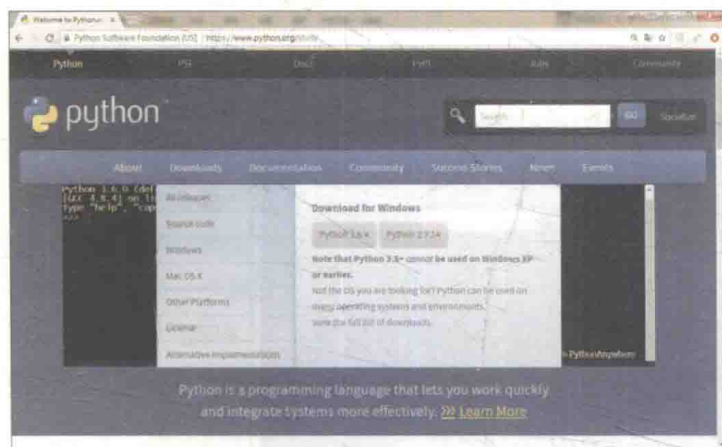


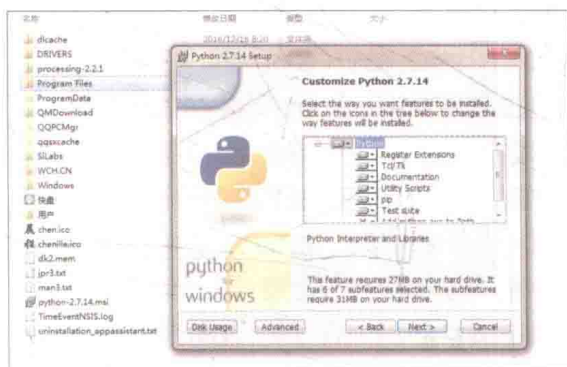
图 1.3 Downloads 选项

这里选择 Python 3.6.4 或 Python 2.7.14 就可以直接下载了。我由于系统的问题下载了 3.x 版却安装不了，如图 1.4 所示，加上之后的程序也没有用到 3.x 版，我安装的是 2.7.14 版。



■ 图 1.4 安装 3.x 版失败

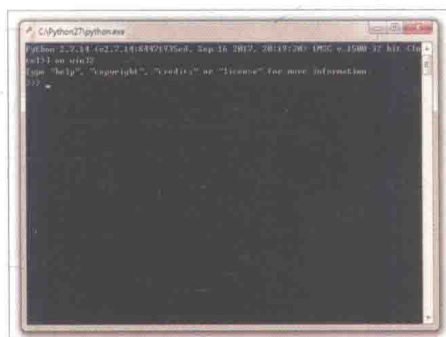
2.7.14 版的安装界面如图 1.5 所示，安装目录默认为 C:\Python27\。



■ 图 1.5 安装 2.7.14 版的 Python

1.5.2 Python 的使用

安装完成后，软件会提供两个工具，一个是命令行形式的 Python(command line)，如图 1.6 所示，另一个是 Python 的集成开发环境 IDLE，如图 1.7 所示。



■ 图 1.6 Python(command line)