

Python 编程及其

在网络科学中的应用

主 编 陈东明
副主编 王冬琦
马 毅



東北大學出版社
Northeastern University Press

Python 编程及其在 网络科学中的应用

主 编 陈东明

副主编 王冬琦 马 毅



东北大学出版社

· 沈 阳 ·

© 陈东明 2018

图书在版编目 (CIP) 数据

Python 编程及其在网络科学中的应用 / 陈东明主编

— 沈阳 : 东北大学出版社, 2018. 10

ISBN 978-7-5517-2044-1

I. ①P… II. ①陈… 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 248242 号

出 版 者: 东北大学出版社

地址: 沈阳市和平区文化路三号巷 11 号

邮编: 110819

电话: 024-83687331(市场部) 83680267(社务部)

传真: 024-83680180(市场部) 83680265(社务部)

网址: <http://www.neupress.com>

E-mail: neuph@neupress.com

印 刷 者: 抚顺光辉彩色广告印刷有限公司

发 行 者: 东北大学出版社

幅面尺寸: 185mm×260mm

印 张: 14.5

字 数: 326 千字

出版时间: 2018 年 11 月第 1 版

印刷时间: 2018 年 11 月第 1 次印刷

策划编辑: 罗 鑫

责任编辑: 郎 坤

责任校对: 一 方

封面设计: 潘正一

ISBN 978-7-5517-2044-1

定 价: 58.00 元

目 录

第 1 章 Python 简介	001
1.1 Python 的由来	001
1.2 Python 能做什么	002
1.2.1 Web 开发	003
1.2.2 人脸识别	004
1.2.3 数据分析	005
1.3 Python 编程环境搭建	006
1.3.1 Windows 环境安装 Python	006
1.3.2 Linux 环境安装 Python	009
1.3.3 Anaconda 管理 Python 环境	011
1.4 Python IDE 的选择	014
1.4.1 PyCharm	015
1.4.2 Sublime Text	015
1.4.3 Eclipse 和 PyDev	016
1.5 小结	023
第 2 章 基本数据结构	024
2.1 字符串	024
2.1.1 字符串截取	024
2.1.2 字符串更新	025
2.1.3 转义字符	025
2.1.4 字符串运算符	026
2.1.5 字符串格式化	027
2.1.6 三引号	028
2.1.7 Unicode 字符串	029
2.2 元组	033
2.2.1 访问元组	033
2.2.2 修改元组	033
2.2.3 删除元组	034
2.2.4 元组运算符	034
2.2.5 元组索引, 截取	035

2.2.6	无关闭分隔符	035
2.2.7	元组内置函数	035
2.3	列表	036
2.3.1	创建一个列表	036
2.3.2	访问列表中的值	036
2.3.3	更新列表	037
2.3.4	删除列表元素	037
2.3.5	列表脚本操作符	037
2.3.6	列表截取	038
2.3.7	列表操作的函数和方法	038
2.4	字典	039
2.4.1	访问字典里的值	039
2.4.2	修改字典	040
2.4.3	字典键的特性	041
2.4.4	字典内置函数和方法	041
2.5	集合	042
2.5.1	创建集合	042
2.5.2	添加、删除集合	043
2.5.3	集合的转换	043
2.5.4	集合常用方法	044
2.6	小结	044
第3章	编程常用语法	045
3.1	Python 数据类型	045
3.1.1	数字	045
3.1.2	字符串	045
3.1.3	列表	047
3.1.4	元组	048
3.1.5	集合	048
3.1.6	字典	049
3.2	判断语句	049
3.3	循环语句	053
3.3.1	while 循环	054
3.3.2	for 循环语句	057
3.4	Python 正则表达式	059
3.4.1	搜索和匹配函数	059
3.4.2	替换函数	060
3.4.3	分割字符串函数	061
3.5	局部变量与全局变量	063
3.6	Python 类和对象	064

3.7 Python lambda 表达式	066
3.8 Python with 语句	068
3.9 Python 多线程	070
3.10 小结	072
第4章 Python 应用实例	073
4.1 数据排序	073
4.1.1 冒泡排序	074
4.1.2 快速排序	074
4.1.3 选择排序	075
4.1.4 插入排序	076
4.1.5 折半插入排序	077
4.1.6 希尔排序	078
4.2 文件批量处理	079
4.3 XML 与 Excel 互转	080
4.3.1 Excel 转 XML	080
4.3.2 XML 转 Excel	088
4.4 网络爬虫	091
4.4.1 用 Python 简单处理 URL	092
4.4.2 用 Python 爬取网络图片	093
4.4.3 Python 网络爬虫实例	095
4.5 Python 图形界面编程	097
4.5.1 tkinter 常用控件	098
4.5.2 控件布局	100
4.6 Python JSON 处理	103
4.7 文件读写操作	104
4.8 小结	105
第5章 Web 应用框架与应用	106
5.1 Django 开发	106
5.1.1 Django 简介	106
5.1.2 安装环境搭建	107
5.1.3 Django 的简单操作	109
5.1.4 Django 模板	115
5.1.5 Django 模型	121
5.1.6 Django 表单	125
5.1.7 Django 搭建论坛系统	132
5.2 Flask 开发	138

5.2.1 安装环境搭建	138
5.2.2 第一个 Flask 项目	139
5.2.3 Flask 模型设计	140
5.2.4 Flask 视图函数 view.py	143
5.2.5 Flask 页面渲染 Jinja2 应用	145
5.3 Flask 开发在线社交网络应用	147
5.3.1 模型	148
5.3.2 页面控制函数	151
5.4 服务器运行与维护	163
5.4.1 服务器环境搭建	163
5.4.2 代码同步	164
5.4.3 常用命令	166
5.5 小结	166
第 6 章 Python 科学计算	167
6.1 Python 科学计算库	167
6.1.1 科学计算库的安装	167
6.1.2 科学计算库案例:函数拟合	168
6.1.3 科学计算库案例:聚类分析	170
6.1.4 科学计算库案例:K-means 算法实现	178
6.2 Networkx 处理图数据	181
6.2.1 创建与访问	181
6.2.2 给图、节点和边添加属性	188
6.2.3 特殊图及其用法	190
6.2.4 图的生成器和图的操作	192
6.3 Python 在复杂网络研究中的应用	195
6.3.1 重要节点识别	195
6.3.2 社团发现	198
6.4 小结	202
第 7 章 Python 在机器学习中的应用	203
7.1 机器学习的概念	203
7.1.1 机器学习的定义	203
7.1.2 机器学习的应用范围	203
7.1.3 机器学习的算法分类	204
7.1.4 机器学习的常用框架	205
7.2 深度学习	205
7.2.1 深度学习基本概念	205
7.2.2 AutoEncoder	206

7.3 Python 机器学习实例	206
7.3.1 Python 实现感知器	206
7.3.2 TensorFlow 项目实例	211
7.4 小结	216
第 8 章 结束语	217
参考文献	218
附录 Python 各版本差异	219

第 1 章 Python 简介

1.1 Python 的由来

Python 是一种面向对象的解释型计算机程序设计语言，由荷兰人 Guido van Rossum 于 1989 年发明，1991 年首次公开发布。Python 的本意是大蟒蛇，但 Python 名字却是来源于英国 20 世纪 70 年代首播的电视剧 *Monty Python's Flying Circus*（《蒙提·派森的飞行马戏团》）。Guido van Rossum 在 1996 年写道：“六年前，在 1989 年 12 月，我在寻找一门‘课余’编程项目来打发圣诞节前后的时间。我的办公室会关门，但我有一台家用电脑，而且没有太多其他东西。我决定为当时我正构思的一个新的脚本语言写一个解释器，它是 ABC 语言的后代，对 UNIX / C 程序员会有吸引力。作为一个略微有些无关想法的人，和一个蒙提·派森的飞行马戏团（*Monty Python's Flying Circus*）的狂热爱好者，我选择了 Python 作为项目的标题。”

Python 是纯粹的自由软件，源代码和解释器 CPython 遵循 GPL（GNU general public license）协议。其语法简洁清晰，特色之一是强制用空白符（white space）作为语句缩进。初学 Python 之时，可能好多熟悉 C 语言或是 Java 的程序员会觉得没有“{ }”确定作用域很不方便。但是，当对 Python 逐渐熟悉之后，就会发现强制缩进的好处——代码更清晰可读。就像大蟒蛇一样，虽然没有脚，但却能够迅速游走。

Python 具有丰富和强大的库。它也常被称为胶水语言，能够把用其他语言制作的各种模块（尤其是 C/C++）很轻松地连接在一起。常见的一种应用情形是，使用 Python 快速生成程序的原型（有时甚至是程序的最终界面），然后对其中有特别要求的部分，用更合适的语言改写，比如 3D 游戏中的图形渲染模块，性能要求特别高，就可以用 C/C++ 重写，而后封装为 Python 可以调用的扩展类库。需要注意的是，在使用扩展类库时可能要考虑平台问题，某些平台可能不提供跨平台操作。

Python 是目前最为活跃的编程语言之一。2017 年 7 月 20 日，IEEE 发布编程语言排行榜：Python 高居首位。见图 1.1。

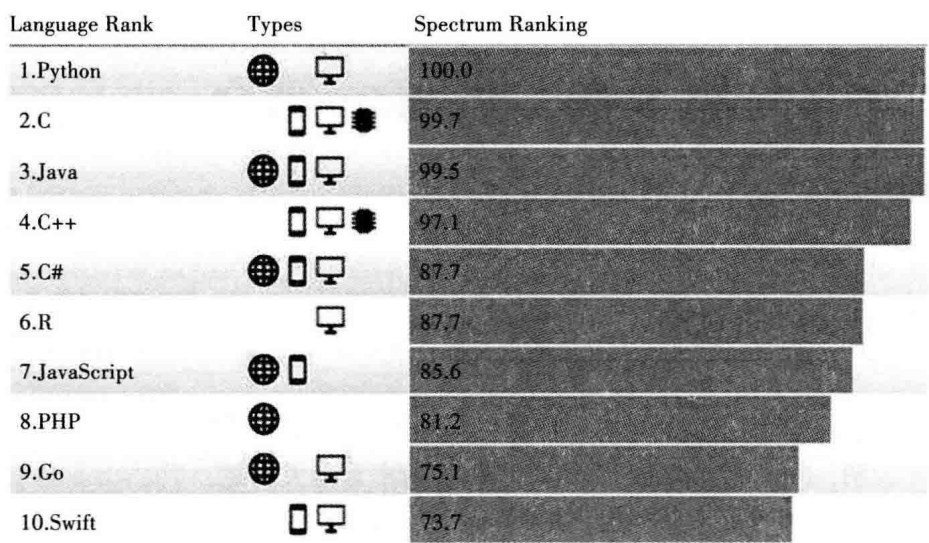


图 1.1 2017 年 IEEE 发布的编程语言排行榜

1.2 Python 能做什么

说了这么多 Python 的由来，想必大家一定很关心 Python 能够做什么？Python 以其简洁的语法、丰富的类库、优秀的多平台支持等特性，在短短的二十几年内迅速占领了 Web 开发、游戏脚本、机器学习、物联网、数据分析、网络信息安全等领域，尤其在机器学习领域稳居首位。此外，一些嵌入式芯片也支持 Python 编程，相信不久的将来，Python 可以凭借其出色表现应用于更多领域。见图 1.2。

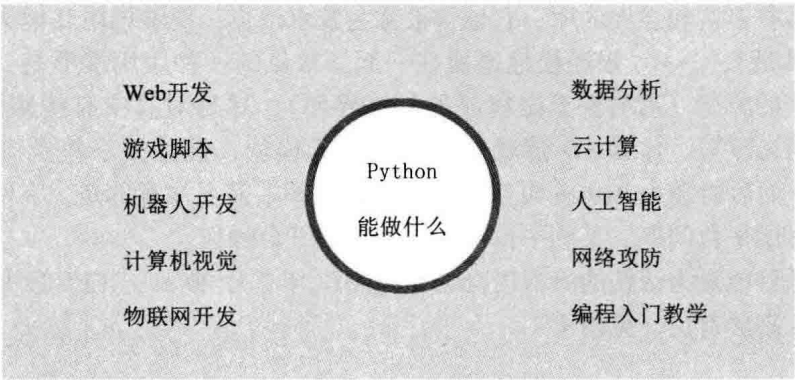


图 1.2 Python 能做什么

Python 语言的应用领域很广，包括运维、Web 开发、应用开发、大数据、数据挖掘、科学计算、机器学习、人工智能、自然语言处理等。举个简单的例子：

```
import time
from selenium import webdriver

def login():
    account_num = input('请输入账号: /n')
    passwd_str = input('请输入密码: /n')
    driver =
webdriver.Chrome(executable_path = '/Users/resolvewang/Documents/program/driver/
chromedriver')
    url = 'http://mail.163.com/'
    driver.get(url)
    time.sleep(5)
    # 163 登陆框是使用 iframe 进行嵌套的, 所以需要先切换到该 iframe
    driver.switch_to.frame('x-URS-iframe')

    account = driver.find_element_by_name('email')
    account.clear()
    account.send_keys(account_num)

    passwd = driver.find_element_by_name('password')
    passwd.clear()
    passwd.send_keys(passwd_str)

    time.sleep(3)
    click_button = driver.find_element_by_id('dologin')
    click_button.click()
    time.sleep(5)
    cur_cookies = driver.get_cookies()[0]
    return cur_cookies

if __name__ == '__main__':
    login()
```

在网络爬虫的应用中, 模拟登陆操作是一个重要的组成部分。只用数行 Python 代码便可以实现网站的模拟登陆, 上述代码是对于 163 邮箱进行模拟登陆的代码。

下面就几个主要应用领域对 Python 的应用进行简要的介绍。

1.2.1 Web 开发

在众多 Python 框架中, 有两个 Web 开发框架使用广泛, 并且提供了非常丰富的插件用于快速开发。本书第 5 章将会详细说明, 此处简单介绍如下:

(1) Flask

Flask 如同官方所描述的那样，是一个 Web 微框架，用几行代码就可以实现一个“Hello World”，非常适合新手入门。

```
from flask import Flask
app = Flask ( _ name_ )

@ app.route ( "/" )
def hello ( ) :
    return "Hello World!"

if _ name_ == "_ main_ " :
    app.run ( )
```

(2) Django

Django 架构的特点是大而全，开箱即用。在对 Flask 基本了解以后，可以着手去了解 Django，因为在生产环境中 Django 使用的更多一点。Django 项目基础目录结构如图 1.3 所示。

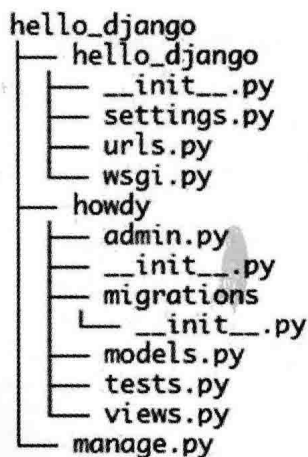


图 1.3 Django 基础目录结构

此外，还有很多简单而强大的 Web 框架，就不一一介绍了。

1.2.2 人脸识别

由于 Python 具有极为丰富的机器学习库，如今利用 Python 进行人脸识别已经非常简单了。Dlib 是基于机器学习的人脸识别的 C++ 库，提供了对 Python 的接口。2017 年圣诞节期间，好多网友发朋友圈“@ 微信官方”，要求在其微信头像上戴一顶圣诞帽，后证明头像上的圣诞帽其实是网友自己 PS 上去的，“@ 微信官方”只是一个恶作剧。不过如果把这件事交给 Python 来做，就没有那么复杂了。笔者搭建了一个社交网络平

台，并设置了一个获取圣诞帽的功能，使用 Dlib 进行人脸识别，效果如图 1.4 所示。



图 1.4 利用 Python 人脸识别获取圣诞帽

图 1.4 展示了利用 Dlib 库实现人脸识别的效果，还是比较理想的。

1.2.3 数据分析

用 Python 做数据分析是非常明智的选择，因为 Python 的库函数非常丰富，同时 Python 能够很方便地实现数据格式之间的转换（更多内容见本书第 4 章）。

随着大数据时代的到来，复杂网络研究一直非常火热，本节以利用空手道数据集使用 GN 算法进行社团检测的研究为例，向读者简单介绍 Python 在数据分析中的应用。例子中使用 Python 编写的 GN 代码进行社团发现，然后将得到的 Json 格式数据通过 Web 页面进行可视化。具体效果如图 1.5 所示。

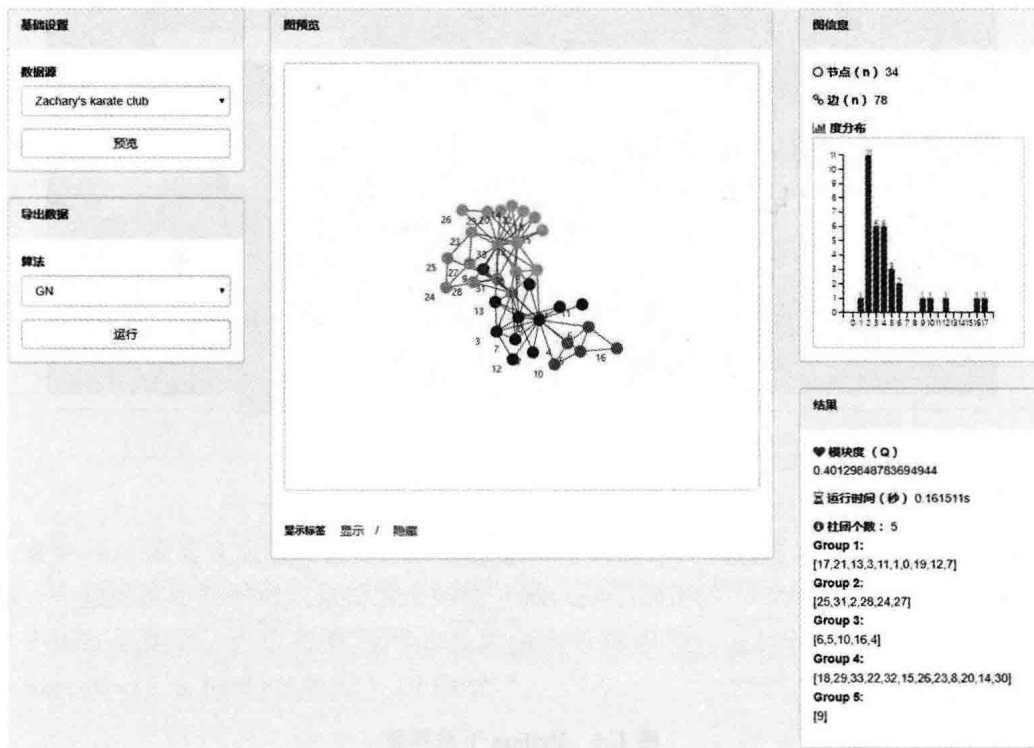


图 1.5 基于 Python 数据分析

在图 1.5 中，页面左侧提供了可供选择的数据集及可用算法，中间给出了社团划分的可视化效果，右侧给出了网络信息，包括网络所包含的节点和边，以及网络的度分布

情况，右下方给出了 GN 算法检测出的社团结构、模块度以及运行时间等。这对于科研人员来说非常方便，在测算相关算法时，可以通过任何操作系统的浏览器获得相关数据，非常方便。看到这里，你是否有些心动了呢，让我们来看看 Python 环境的搭建吧。

1.3 Python 编程环境搭建

Python 是一种跨平台的脚本语言，可以运行在 Windows，Linux，Mac OS 等不同的系统上。如果你使用的是 Windows 操作系统，Python 的编程环境安装非常简单，只需要去 Python 官网下载对应的程序安装即可，不需要太多的硬盘空间，目前，Windows 操作系统 32 位版的 Python 3.6.4 安装包的大小是 29.2MB。接下来让我们一起来安装 Python 编程环境吧！

1.3.1 Windows 环境安装 Python

在 Windows 平台上安装较为简单，步骤如下：

在官网下载安装包。地址：<https://www.python.org/downloads/>。见图 1.6。

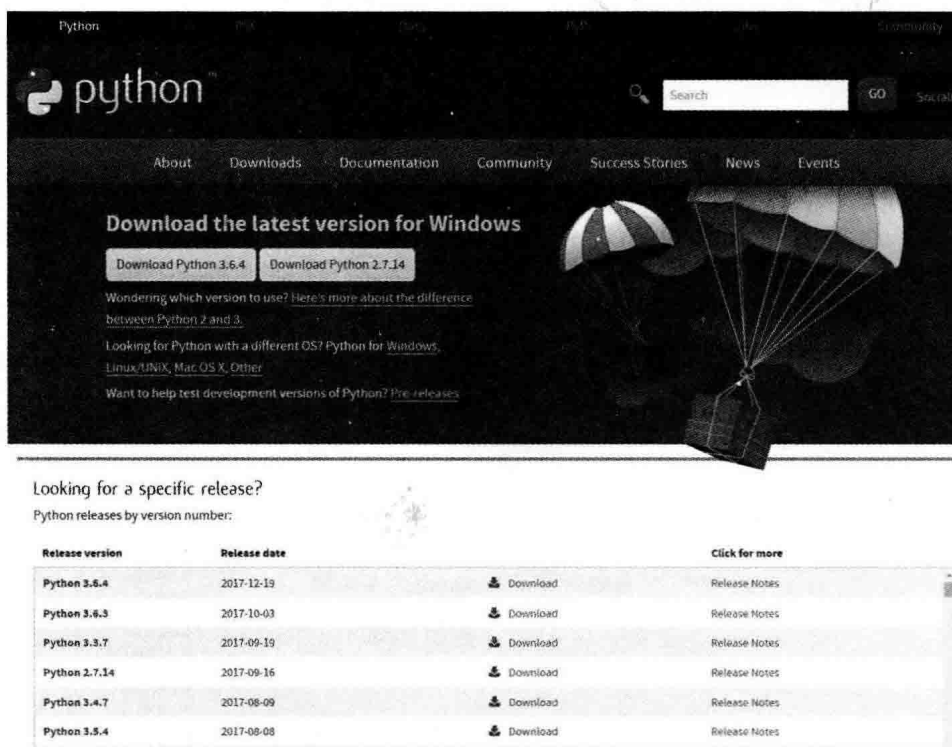


图 1.6 Python 下载界面

将 Python 安装文件下载到本地磁盘位置，然后双击 exe 程序安装。见图 1.7。

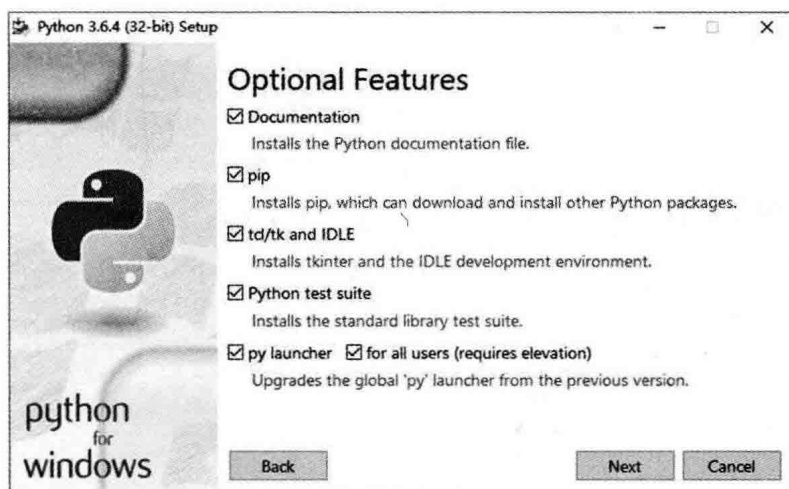


图 1.7 Python 安装选项

需要注意的是 Python 中有一个很重要的工具 pip，是用来管理 Python 相关库的，操作起来非常方便，所以一定要在 pip 前面的复选框中保持勾选，这样在今后的操作中会节省很多时间。Python 安装过程如图 1.8 所示。

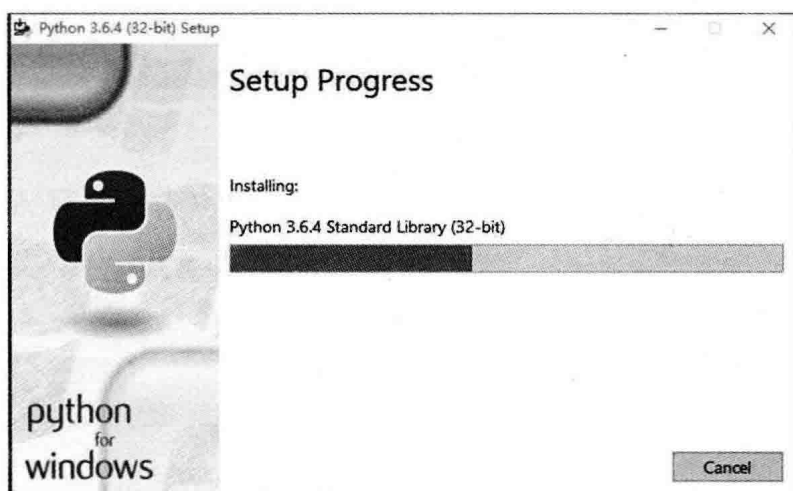


图 1.8 Python 安装过程

在 Python 安装完成后，会提示禁用 Windows 系统路径长度最大值的选项，如图 1.9 所示。一般开发中几乎可以忽略这个问题，默认是不禁用的。

Python 更新后，可以查看官网的最新语法升级说明。（<https://docs.python.org/3.6/whatsnew/3.6.html>），如图 1.10 所示。

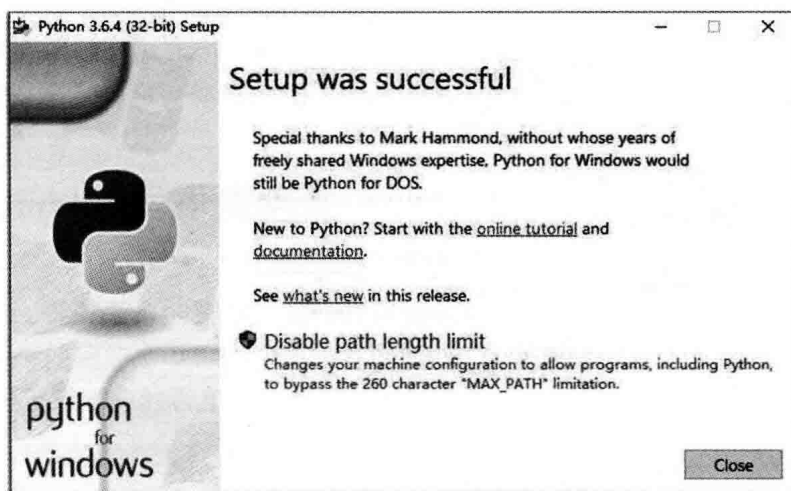


图 1.9 Python 安装完成界面

What's New In Python 3.6

Editors: Elvis Pranskevichus <elvis@magic.io>, Yuri Selivanov <yury@magic.io>

This article explains the new features in Python 3.6, compared to 3.5. Python 3.6 was released on December 23, 2016. For full details, see the changelog.

See also: PEP 494 - Python 3.6 Release Schedule

Summary – Release highlights

New syntax features:

- PEP 498, formatted string literals.
- PEP 515, underscores in numeric literals.
- PEP 526, syntax for variable annotations.
- PEP 525, asynchronous generators.
- PEP 530, asynchronous comprehensions.

New library modules:

- `secrets`: PEP 506 – Adding A Secrets Module To The Standard Library.

CPython implementation improvements:

- The dict type has been reimplemented to use a more compact representation based on a proposal by Raymond Hettinger and similar to the PyPy memory when compared to Python 3.5.
- Customization of class creation has been simplified with the new protocol.
- The class attribute definition order is now preserved.
- The order of elements in `**kwargs` now corresponds to the order in which keyword arguments were passed to the function.
- DTrace and SystemTap probing support has been added.
- The new `PYTHONMALLOC` environment variable can now be used to debug the interpreter memory allocation and access errors.

图 1.10 Python 升级说明

安装完成后，在 Windows 命令行里输入 Python 即可进入 Python 编程环境，如图 1.11 所示。

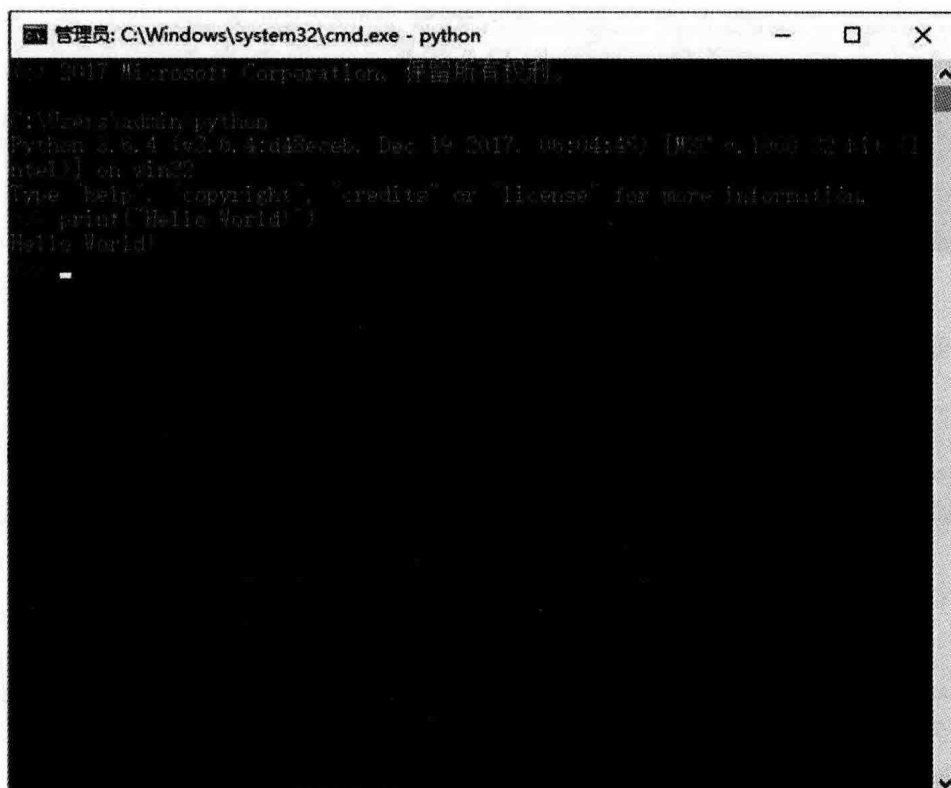


图 1.11 Python 编程环境

1.3.2 Linux 环境安装 Python

通常，Linux 系统默认自带 Python 环境，笔者的一台云服务器系统 Cent OS 系统自带 Python 2.7.5，如图 1.12 所示。

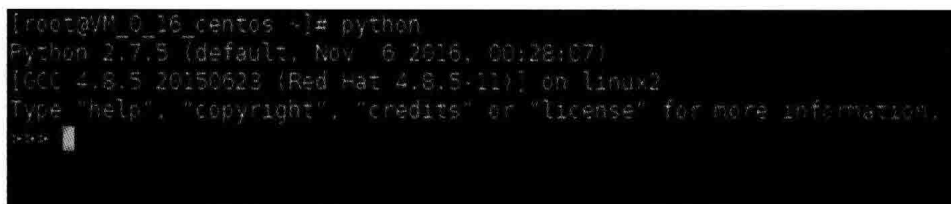


图 1.12 Linux 系统自带 Python 环境

如果我们需要使用 Python 的功能，可以使用 Python 环境管理工具（如 Anaconda 等），也可以直接安装其他版本的 Python。下面介绍 Linux 操作系统升级 Python 方法：

①首先下载 Python 源码，然后解压，如图 1.13 所示，命令如下：

```
# wget https://www.python.org/ftp/python/3.6.4/Python-3.6.4.tgz
# tar zxvf Python-3.6.4.tgz
# cd Python-3.6.4/
# mkdir /usr/local/python3
```