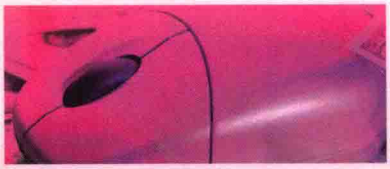

# 数据馆员的Python简明手册




非外借

顾立平　田鹏伟　**编著**

科学技术文献出版社
SCIENTIFIC AND TECHNICAL DOCUMENTATION PRESS

# 数据馆员的 Python 简明手册

>>> 顾立平 田鹏伟 编著

科学技术文献出版社
SCIENTIFIC AND TECHNICAL DOCUMENTATION PRESS

·北 京·

**图书在版编目(CIP)数据**

数据馆员的Python简明手册 / 顾立平, 田鹏伟编著. —北京：科学技术文献出版社，2017. 10

ISBN 978-7-5189-3014-2

Ⅰ. ①数… Ⅱ. ①顾… ②田… Ⅲ. ①软件工具—程序设计 Ⅳ. ① TP311.561

中国版本图书馆 CIP 数据核字（2017）第 161157 号

**数据馆员的Python简明手册**

策划编辑：崔灵菲 责任编辑：王瑞瑞 责任校对：张吲哚 责任出版：张志平

出 版 者 科学技术文献出版社
地 址 北京市复兴路15号 邮编 100038
编 务 部 （010）58882938，58882087（传真）
发 行 部 （010）58882868，58882874（传真）
邮 购 部 （010）58882873
官方网址 www.stdp.com.cn
发 行 者 科学技术文献出版社发行 全国各地新华书店经销
印 刷 者 虎彩印艺股份有限公司
版 次 2017 年 10 月第 1 版 2017 年 10 月第 1 次印刷
开 本 850 × 1168 1/32
字 数 96千
印 张 5.75
书 号 ISBN 978-7-5189-3014-2
定 价 58.00元

# Preface 前言

本手册旨在协助初级数据馆员们能够迅速了解 Python 方面的知识、用途及整体概貌，作为进一步实践操作层面之前的入门基础读物。

数据馆员是能够充分实现开放科学政策、措施、服务的一群新型信息管理人员，他们熟悉数据处理、数据分析、数据权益、数据政策，且具有知识产权与开放获取的知识和经验。

Python 是一种简单易学、功能强大的编程语言，它具有高效率的高层数据结构，可以简单而有效地实现面向对象编程。它语法简洁，支持动态输入，适用于快速的应用程序开发。

简单易懂，其实往往比故作玄虚或者令人费解来的更有价值，这也是 Python 的核心价值之一。本手册旨在系统地介绍 Python 的主要核心知识，由于这门语言还在不断产生新的应用，并且还在不断发展当中，因此人们从各种角度，出版了面向不同程度和需求的 Python 介绍或者专著，本手册是少数简明扼要、有

系统性、可操作性强的专著。

本手册包括 10 个部分。第 1 章概述 Python 是什么。第 2 章概述 Python 的流程控制，挑选了一部分重要的计算机语言的内容。第 3 章是 Python 的函数及数据结构，这也是介绍一门计算机语言所不可少的部分。第 4 章是 Python 的异常处理，这是工程实践上重要，但是一般数据分析内容常常忽略的部分。第 5 章是 Python 的数据处理与计算，是学习这门计算机语言的基础。第 6 章是数据描述与分析。第 7 章是绘图与可视化。第 8 章概述数据挖掘。这些章节里都给了一些简单案例，能够操作和直观理解。第 9 章概述 Django 与 Twisted 两个工程应用上的重要内容。附录以截图方式一步步带着完全没有计算机编程经验或者没有数据分析软件操作经验的读者搭建环境和进行初步练习。

我们相信通过本手册读者都能够利用计算机程序高效完成信息分析的任务，可以让你直接掌握绝大部分必须知道的知识点，并且通过上机操作的方式理解它，你会越来越喜欢上并且会越来越擅长这门简单易懂却又千变万化的语言。

编著者

2017 年初春于中关村

Contents >>>>>>>>> 目录

>>>>>> 第1章

# 关于 Python

## 1.1 Python 发展历史

### 1.1.1 什么是 Python

“python”在英语单词中是蟒蛇的意思（图 1–1）。


图 1–1 蟒蛇图标

Python 语言是少有的一种可以称得上既简单易学又功能强大的编程语言。你将惊喜地发现 Python 语言是多么的简单，它注重的是如何解决问题而不是编程语言的语法和结构。

Python 是一种简单易学、功能强大的编程语言，它有高效率的高层数据结构，简单而有效地实现面向对象编程。

Python 简洁的语法和对动态输入的支持，再加上解释性语言的本质，使得它在大多数平台上的许多领域都是一个理想的脚本语言，特别适用于快速的应用程序开发，具体介绍见 Python 官方网站 https：//www.python.org/（图 1–2）。

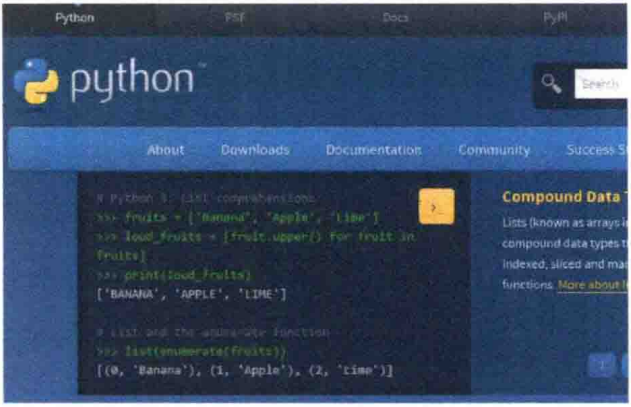


图 1–2　Python 官网界面

图 1–3　Guido

### 1.1.2　Python 的作者

Python 的作者是吉多・范罗苏姆（Guido von Rossum）（图 1–3），荷兰人。1982 年，Guido 从阿姆斯特丹大学获得了数学和计算机硕士学位。虽然他算得上是一位数学家，但他更加享受计算机带来的乐趣。用他的话说，虽然拥有数学和计算机双料资质，但他更趋向于做计算机相关的工作，并热衷于做任何和编程相关的事情。

### 1.1.3　Python 的历史

Python 语言诞生于 1989 年。在阿姆斯特丹，圣诞节 Guido 在家中正为 ABC 语言编写一个插件。ABC 语言是由荷兰的数学与计算机研究所（图 1–4）开发的，专为方便数学家、物理学家使用。Guido 在该研究所工作，并参与到 ABC 语言的开发。

Guido 希望有一种语言能够像 C 语言那样，全面调用计算机

的功能接口，同时又可以轻松地编程。ABC 语言让 Guido 看到希望。ABC 语言以教学为目的，目标是“让用户感觉更好”，希望让语言变得容易阅读，容易使用，容易记忆，容易学习，并以此来激发人们学习编程的兴趣。

图 1-4　数学与计算机研究所

在 1989 年圣诞节假期，Guido 开发的这个插件实现了一个个脚本语言，且功能强大。Guido 以自己的名义发布了这门语言，且命名为 Python。

因为 Guido 是天空马戏团忠实的粉丝，所以他用一个大蟒蛇飞行马戏团（图 1-5）的名字中的一个单词“Python”作为这门新语言的名字。

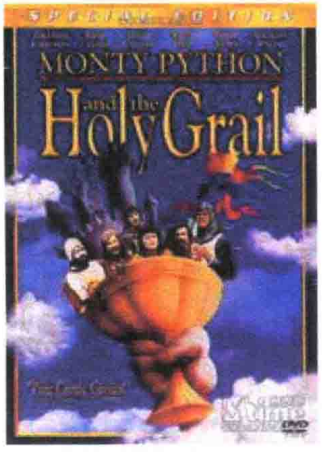


图 1-5　巨蟒与圣杯海报

### 1.1.4　Python 的发展阶段

CNRI 时期：CNRI 是 Python 发展初期的重要资助单位，

Python 1.5 前的主要成果大部分在此时期完成。

BeOpen 时期：Guido von Rossum 与 BeOpen 公司合作，Python 1.6 与 Python 2.0 基本上同时推出，但原则上分别维护。Python 2.0 的许多功能与 Python 1.6 不同。

DC 时期：Guido 离开 BeOpen 公司，将开发团队带到 Digital Creations（DC）公司，该公司以发展 Zope 系统闻名，由于 Guido 的加入，这个项目也颇受关注。

Python 3.0：Python 2.X 和 Python 3.X 差异很大，前后不兼容，虽然有 2 to 3 的工具可以转换，但不能解决所有的问题。Python 3.X 尚未完全普及，很多第三方的库都没有官方支持 Python 3.X。考虑到前后版本的这个不兼容性，这会让一些人对采用 Python 开发项目产生顾忌。

里程碑：Python 因在 2010 年获得较大市场份额的增长（1.81%，增长速度最快）获年度 Tiobe 编程语言大奖，参见 http://www.iteye.com/news/19455，最新的排名参见 http://www.tiobe.com/tiobe-index/（图 1-6）。

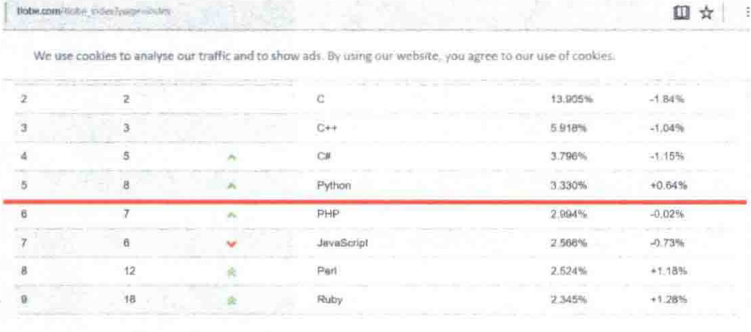


图 1-6 各语言所占市场份额

## 1.2 Python 版本

### 1.2.1 版本分类

Python 的版本主要集中在 2.0 和 3.0 之间，且主要版本如图 1-7

所示。

| Release version | Release date | | Click for more |
|---|---|---|---|
| Python 3.4.4 | 2015-12-21 | Download | Release Notes |
| Python 3.5.1 | 2015-12-07 | Download | Release Notes |
| Python 2.7.11 | 2015-12-05 | Download | Release Notes |
| Python 3.5.0 | 2015-09-13 | Download | Release Notes |
| Python 2.7.10 | 2015-05-23 | Download | Release Notes |
| Python 3.4.3 | 2015-02-25 | Download | Release Notes |
| Python 2.7.9 | 2014-12-10 | Download | Release Notes |

图 1–7　Python 主要版本

2.0 版本，也称为 old 版本，目前稳定的版本是 2.7.11 版。

3.0 版本，是相对而言比较新的版本，目前稳定的版本是 3.5.0 版。

### 1.2.2　版本对比

Python 是开源软件，其版本在不断更新，且 2.0 系列和 3.0 系列相互不兼容，所以在选择具体版本时需要定夺，但是具体该选择哪一个版本，笔者建议选择 2.7.11 版，原因在于 Python 之所以能被大众接受并流行，除了其开源之外，就是它强大的第三方包支持，而目前来说许多第三方的包更多支持 2.7.11 版。因此建议选择 2.7.11 版。

关于用 Mac 还是 Windows 开发，笔者认为 Mac 较优，原因还是第三方包支持的作用。当然，一般的开发和数据分析在哪类操作系统上都是可以的。

## 1.3　Python 基本语法

Python 的设计目标之一是让代码具备高度的可阅读性。它设计时尽量使用其他语言经常使用的标点符号和英语单词，让代码看起来整洁美观。它不像其他的静态语言如 C、Pascal 那样需要

重复书写声明语句，也不像它们的语法那样经常有特殊情况。

### 1.3.1 Python 缩进

Python 开发者有意让违反了缩进规则的程序不能通过编译，以此来强制程序员养成良好的编程习惯。并且 Python 语言利用缩进表示语句块的开始和退出（Off-side 规则），而非使用花括号或者某种关键字。增加缩进表示语句块的开始，而减少缩进则表示语句块的退出。缩进成了语法的一部分。例如，if 语句：

```
if age<21:
    print" 你不能买酒。"
    print" 不过你能买口香糖。"
print" 这句话处于 if 语句块的外面。"
```

注：上述例子为 Python 2.0 版本的代码，3.0 版本需要添加括号，如：print(" 你不能买酒。")。

根据 PEP 的规定，必须使用 4 个空格来表示每级缩进（如果不清楚 4 个空格是如何规定的，在实际编写中可以自定义空格数，但是要满足每级缩进间空格数相等）。使用 Tab 字符和其他数目的空格虽然都可以编译通过，但不符合编码规范。支持 Tab 字符和其他数目的空格仅仅是为了兼容很旧的 Python 程序和某些有问题的编辑程序。

### 1.3.2 控制语句

if 语句。当条件成立时运行的语句块。经常与 else、elif( 相当于 else if) 配合使用。

for 语句。遍历列表、字符串、字典、集合等迭代器，依次处理迭代器中的每个元素。