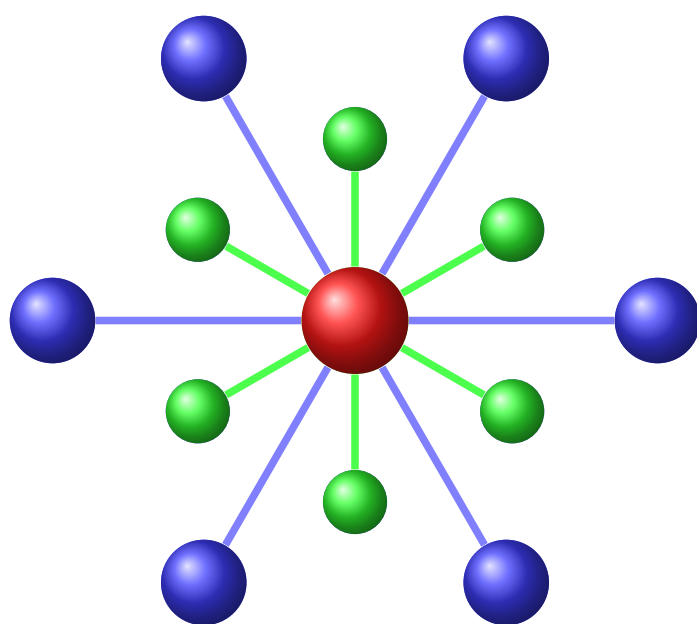


Python 3 与数据分析基础

—数据科学家的分析利器



夏天 @IRM-Renmin University of China

February 8, 2016

序言

Python 是一门简单易学、功能强大的编程语言。它具有高效的高级数据结构和简单而有效的面向对象编程的特性。Python 优雅的语法和动态类型、以及其解释性的性质，使它在许多领域和大多数平台成为脚本编写和快速应用程序开发的理想语言。

从 Python 网站 <https://www.python.org/> 可以免费获得所有主要平台的源代码或二进制形式的 Python 解释器和广泛的标准库，并且可以自由地分发。网站还包含许多免费的第三方 Python 模块、程序、工具以及附加文档的发布包和链接。

Python 解释器可以用 C 或 C++（或调用 C 的其他语言）中轻松的对新的函数和数据类型进行扩展 Python 也适合作为可定制应用程序的一种扩展语言。

本教程通俗地向读者介绍 Python 语言及其体系的基本概念和功能。随手使用 Python 解释器来亲自动手实践是很有帮助的，并且由于所有示例都是自成体系的，所以本教程也可以离线阅读。

有关标准对象和模块的说明，请参阅 Python 标准库。Python 语言参考给出了 Python 语言的更正式的定义。要编写 C 或 C++ 的扩展，请阅读扩展和嵌入 Python 解释器与 Python/C API 参考手册。也有几本书深度地介绍了 Python。

本教程不会尝试全面地涵盖每一个单独特性，甚至即使它是常用的特性。相反，它介绍了许多 Python 的值得注意的特性，从而能让你很好的把握这门语言的特性。经过学习，你将能够阅读和编写 Python 的模块和程序，并可以更好的学会 Python 标准库中描述的各种 Python 库模块。

图书网站和相关资源

本书的网站和附带资源可以通过以下网址获得：

<http://dmml.asu.edu/smm>

网站同时提供了配套课件、练习题和示范工程，同时给出了与社会媒体挖掘相关的公共资源入口。

教师须知

本书按照高年级本科生或研究生一个学期的学习课程进行设计，主要面向拥有计算机科学背景的学生，但对于掌握概率论、统计和线性代数基础知识的读者，也很容易理解本书内容。本书部分章节用于回顾一些基础知识，学生已掌握该部分内容时可忽略这些章节。例如，如果学生已经学过了数据挖掘或机器学习课程，可略过第 5 章。如果时间受限，第 6 章至第 8 章应深入讨论，但第 9 章和第

10 章可以简要讨论或者作为阅读材料部分。

Reza Zafarani

Mohammad Ali Abbasi

Huan Liu

2013 年 8 月于美国亚利桑那州坦佩

目录

1 引言	1
1.1 什么是 Python	1
1.2 Python 的安装	2
1.2.1 Linux 用户	2
1.2.2 Mac 用户	2
1.2.3 Windows 用户	2
1.3 如何运行 Python 程序	3
1.3.1 使用带提示符的解释器	3
1.3.2 运行 Python 源代码文件	3
1.4 工作环境的配置	3
1.4.1 源代码编辑器的选择	3
1.4.2 利用 iPython 增强 Python 的交互操作	4

1.1 什么是 Python

如果你打算让计算机做更多的事情，而不仅仅局限于使用他人已经编好的可运行程序，例如，以某种你希望的方式对照片文件重新命名，或者信手写一个数独游戏以消磨时间，甚至是在科学兴趣探索中，进行一些复杂的数字运算，那么 Python 将是一种非常适合你需求的高级编程语言，其语法简单，安装测试方便，功能又异常强大，并拥有高效率的高层数据结构和面向对象特征。付出时间成本投资于 Python 语言，你一定会取得满意的回报。

Python 语言是一种少有的既简单又强大的高级编程语言，注重问题的解决而非编程语言的语法和结构，正如其官方所介绍：

1. Python is a programming language that lets you work more quickly and integrate your systems more effectively.

(Python 是一种能够让你工作更便捷、系统集成更高效的编程语言)

2. Python is a clear and powerful object-oriented programming language, comparable to Perl, Ruby, Scheme, or Java.

(相比于 Perl、Ruby、Scheme 或 Java，Python 是一种简单、强大的面向对象的编程语言)

顺便说一句，Python 语言的作者 Guido van Rossum 是根据英国广播公司的节目“Monty Python’s Flying Circus”(蟒蛇飞行马戏) 给这个语言命名的，并非他本人特别喜欢蛇缠起它们的长身躯碾死动物觅食。

目前，Python 正处于有 2.x 版本到 3.x 版本过渡的过程之中，Python 3 又称为 Python 3000 或 Py3K，作为新一代 Python 语言，Python 3 以上版本去除了之前版本中积累的一些老问题，并使语言更为清晰，但另一方面，由于 Python2.x 是如此的成功，在大量系统之中得到了广泛应用，出于移植成本考虑，短时间之内还无法将所有 Python 代码都升级为 Python 3 版本，因此，Python 2 和 Python 3 还将长时间并存¹，考虑到未来发展，本教程将采用 Python 3 的语言规范进行讲解²。

¹Python 2.7 之后，Python 2.x 版本将不会再有重大更新，新项目建议使用 Python 3.x，到底选择 Python2 还是 Python3，可阅读：<https://wiki.python.org/moin/Python2orPython3>


²如果你已有大量的 Python 2.x 代码，也可以使用工具将 2.x 转换成 3.x 的源代码，参考<http://docs.python.org/3.5/library/2to3.html>

1.2 Python 的安装

如果你已经安装了 Python 2.x , 那么就没必要卸载后再安装 Python 3.x , 实际上, 可以将它们同时安装在电脑里。


1.2.1 Linux 用户

如果你正在使用一个 Linux 的发行版, 比如 Ubuntu 或 Fedora, 默认情况下, 系统里面已经安装了 Python, 要测试系统中是否已经安装了 Python 以及 Python 的版本, 可以打开一个 shell 程序, 输入如下命令:

 命令
\$ python -V
Python 2.7.10

注意, 参数 V 为大写形式, 如果写成了小写的 v, 则代表以详细追踪方式显示 Python 的启动时的执行内容, 而不是显示 Python 的版本。

2016 年以来发布的 Linux 版本, 通常在装有 python 2 的同时, 也安装有 python3, 以 Ubuntu 为例, 在命令行上运行 python3 -V, 可以运行 python3, 并得到其版本信息, 类似于如下:

 命令
\$ python3 -V
Python 3.4.3+ (default, Oct 14 2015, 16:03:50)

1.2.2 Mac 用户

Mac OS X 用户会发现已经在系统中安装了 Python 。打开 Terminal.app 运行 python -V , 接着参考上面关于 Linux 部分的建议。

1.2.3 Windows 用户

访问 <http://www.python.org> 网站下载最新版, 在写本书的时候是 3.0 beta 1 。仅有 12.8MB , 与大多其它的语言或软件相比, 是非常紧凑的。安装与其它 Windows 软件一样。

有趣的是, 大多的 Python 下载是来自 Windows 用户的。当然, 这并不能说明问题, 因为几乎所有的 Linux 用户已经在安装系统的时候默认安装了 Python 。

1.3 如何运行 Python 程序

我们将看一下如何用 Python 编写运行一个传统的 “Hello World” 程序。通过它, 你将学会如何编写、保存和运行 Python 程序。

有两种使用 Python 运行你的程序的方式——使用交互式的带提示符的解释器或使用源文件。我们将学习这两种方法。

1.3.1 使用带提示符的解释器

在 shell 提示符下, 键入 `python` 命令启动解释器。对 Windows 用户, 如果你已经配置好了 `PATH` 变量, 那么就可在命令行中启动解释器。

如果使用 IDLE, 点击开始 → 程序 → Python 3.0 → IDLE (Python GUI)。键入 `print('Hello World')`, 按回车键。将会看到 Hello World 字样的输出。

注意, Python 会在下一行立即给出你输出! 对于刚键入的 Python 语句 `print('Hello World')`, 我们使用 `print`(不要惊讶, 这里的打印默认是 “打印” 到显示器上, 即在显示器上输出结果) 来打印你供给它的值, 这里提供的文本是 “Hello World”, 它被立即打印在屏幕上。

至于如何退出解释器提示符, 如果你使用的是 Linux/BSD shell, 那么按 “Ctrl-d” 退出提示符, 如果是在 Windows 命令行中, 则按 “Ctrl-z”, 再按回车键退出。

1.3.2 运行 Python 源代码文件

1.4 工作环境的配置

工欲善其事必先利其器, 搭建一个好的工作环境, 对于提高生产效率是十分重要的。下面, 我们从源代码编辑器和增强命令行交互工具两个方面对 Python 工作环境进行介绍。

1.4.1 源代码编辑器的选择

支持 Python 的编辑器有很多, 在一个由多个工具组成的环境中, 最好只关注与 Python 代码编写相关的编辑器。实际上, 简单的代码编辑器和集成开发环境之间的边界并不明显, 一些简单的编辑器也会提供与系统交互和扩展的机制。对于代码编写来说, 一个配置齐全的源代码编辑器可以降低一些重复性工作, 有效辅助代码的编写。

多年以来, Python 源代码编辑器的最佳选择是 vim 或 Emacs, 初次接触这两个编辑器会觉得其操作并不友好, 甚至感觉完全无法使用, 这是因为这两个编辑器都引入了大量键盘快捷键, 只有经过

一段时间的熟悉之后，才会体会到它们的强大之处，如果你觉得自己还年轻，那么投入精力去掌握它们，你的付出会让你终生受用。

Vim 在多数 Linux 的发行版和苹果的笔记本操作系统中，默认都已经安装³，可以直接使用，而 Emacs 通常需要单独安装。相比于 Vim 或 Emacs，新潮的开发人员或初学者，可能更喜欢以鼠标操作为主的编辑器，其中，Sublime Text(<http://www.sublimetext.com/>) 是目前非常流行的编辑器，对 Python 的语法支持也很好。

综上，初学者可以先需用 Sublime Text 作为 Python 的源代码编辑器，并逐步学习使用 Vim 或 Emacs，提高熟练程度。

1.4.2 利用 iPython 增强 Python 的交互操作

³部分系统默认安装的是 vi，而 vim 是 vi 的升级版本，兼容 vi 的所有指令，并拥有大量新特性，如果系统默认安装的是 vi，建议读者进一步安装 vim。