

认识 Python

知识目标

- 了解 Python 的发展历史。
- 了解 Python 的特点和运行机制。
- 了解 Python 的学习路径。

能力目标

- 会正确安装 Python 开发工具,并配置环境变量。
- 能利用 pip 命令进行扩展库的安装、升级和卸载。

项目导学

小 T 是一位计算机程序语言的爱好者,立志毕业后进入 IT 领域工作,他想通过学习实现 成为优秀程序员的梦想。如今,他进入了导师组建的学生创新团队,在导师的引领下刻苦钻研 一门强大并且实用的计算机语言——Python。让我们跟随小 T 的学习步伐,一起来学习这门神 奇的语言吧。

- (1) 了解 Python 的基本特性。
- (2) 配置 Python 的开发环境。
- (3) 安装扩展库。

Python 概述 1.1

1.1.1 任务引入

【任务描述】

Python 是现今最流行的程序设计语言之一,不仅具有强大的科学计算能力、数据处理和分析能力,还具有丰富的可视化表现功能和简洁的程序设计能力。只有了解 Python 的起源、认识 Python 的特点才能学好 Python。那现在就与小 T 一起迈出学习 Python 的第一步——了解

Python 的历史、特点以及学习路径。

【任务分析】

Python 是一门非常有趣的语言,我们可以从以下几个方面来进行学习。

- (1) Python 的历史。
- (2) Python 的特点。
- (3) Python 的学习路径。

1.1.2 Python 的历史

1. Python 的产生

Python,本义是指"蟒蛇"。在1989年的圣诞节期间,荷兰人 Guido van Rossum 为了在阿姆斯特丹打发时间,决心开发一个新的脚本解释程序。之所以选中 Python 作为程序的名字,是因为他是 BBC 电视剧——蒙提•派森的《飞行马戏团》的爱好者。

Python 的灵感来自于 ABC 语言——Guido van Rossum 参与开发的一种适用于非专业程序 开发人员的编程语言,但它并不流行,原因是它是非开放性语言,所以 Guido van Rossum 将 Python 定位为开放性语言。

Python 的设计理念是优雅、明确、简单,以至于现在网络上流传着"人生苦短,我用 Python"的说法。例如,完成同一个任务,用 C 语言要写 1000 行代码,用 Java 要写 100 行代码,而用 Python 可能只要写 20 行代码。可见 Python 有着简单易学、开发速度快等特点。

经过多年的发展, Python 已经成为非常流行的程序开发语言。到底有多流行? 让我们看看 知名的 TIOBE 开发语言排行榜。TIOBE 是程序开发语言流行趋势的一个指标,基于互联网中 有经验的程序员、课程及第三方厂商的数量来确定排行,且排名每个月都会更新。图 1-1 所示 为 2022 年 5 月的 TIOBE 程序开发语言排行榜,从该排行榜可以看出, Python 是目前最流行 的编程语言之一。

May 2022	May 2021	Change	Programming Language	Ratings	Change
1	2	~	Python	12.74%	+0.86%
2	1	*	C c	11.59%	-1.80%
3	3		َرْجَّ, Java	10.99%	-0.74%
4	4		C++	8.83%	+1.01%
5	5		С#	6.39%	+1.98%
6	6		VB Visual Basic	5.86%	+1.85%
7	7		JS JavaScript	2.12%	-0.33%
8	8		Assembly language	1.92%	-0.51%
9	10	~	SQL SQL	1.87%	+0.16%
10	9	~	PHP PHP	1.52%	-0.34%

图 1-1 2022 年 5 月的 TIOBE 程序开发语言排行榜

2. Python 的版本

Python 目前有两个版本: Python2 和 Python3, Python3.0 不再向后兼容 Python2.X 的版本, Python2.7 已于 2020 年年底停止支持,现阶段大部分企业使用的是 Python3。

目前,Python 的知识产权由 Python 软件基金会(Python Software Foundation,简称 PSF) 来管理,它是一个独立的非营利组织,拥有 Python 2.1 及以上版本的版权。PSF 的使命是推进 与 Python 相关的开源技术的开发进程,并推广 Python 的使用。

1.1.3 Python 的特点

Python 作为一门较"新"的编程语言备受欢迎,而且能在 C、C++、Java 等老牌编程语言 中夺得一席之地,必然有其优势,接下来根据其特点来阐述这门编程语言的优势。

1. 简单易学

Python 是一种代表简单主义思想的语言,阅读一个良好的 Python 程序就像在读英语一样, 它使用户能够专注于解决问题,而无须理解语言本身。Python 极容易上手,因此可作为学生信 息素养提升的基础编程语言。

2. 开源免费

Python 遵循 GPL 协议,是开源免费的。用户可以自由使用这个语言和代码,而无须支付 任何费用,也不需要担心版权的问题,同时也更利于改进和优化。

3. 模式多样

Python 在语法层面既支持面向对象编程又支持面向过程编程,这一特点使得用户可以灵活选择代码模式,因此它具有将对象属性、方法抽象成类,以及支持多态、重载、继承等面向对象语言的特点。

4. 可移植性

Python 程序可以被移植在许多平台上运行,只要这些平台上安装有 Python 解释器即可。 这些平台包括 Linux、Windows、Macintosh 等。

5. 解释性

用编译性语言(如 C 或 C++)编写的程序源文件需要通过编译器翻译为机器语言形式的目标文件,而用 Python编写的程序不需要编译,可以直接从源文件运行程序,这使 Python的使用更加简单。

6. 生态丰富

Python 提供丰富的标准库,可以帮助处理各种工作,包括数学计算、数据库、网络等;还 支持许多其他高质量的第三方库,可应用于图像处理、网页设计和数据分析等领域。

即学即答:

 下面不属于 Python 特点的是()。

 A. 简单易学
 B. 开源的、免费的

 C. 属于低级语言
 D. 生态丰富

1.1.4 Python 的学习路径

Python 拥有一个强大的生态系统,众多的第三方库使其在数据分析、数据可视化、机器学习和 Web 开发中都可得到应用。

Python 在众多的应用领域中有不同的侧重点。以 Python 应用中最广泛的数据采集、处理、 分析等应用领域为例,简要介绍本书以及后续的学习路径。

从 Python 程序设计基础到数据采集、处理、分析,再到机器学习、数据可视化,这一学 习主线归纳起来可以按照以下几个步骤进行。

第一步: Python 开发环境设置。

Python 开发环境众多且各有所长,又鉴于第三方库及其版本众多,经常需要在不同环境下运行不同的 Python 项目程序,从 IDLE、PyCharm 到 Anaconda,可以在多个开发软件下配置 Python 编程环境,让用户拥有最优的使用效率。

第二步:学习 Python 的基础知识。

学习 Python 的基本语法、各种组合数据类型(字符串、列表、字典、集合等),理解 Python 中的类和对象,能利用 Python 进行文件、数据库的操作,这也是本书的主要内容。

第三步:学习 Python 爬虫。

学习 Python 爬虫相关的第三方库及框架,能通过开发爬虫程序来进行数据采集、存储, 采集的数据一般存入数据库或文件中,这是进行数据分析的数据基础。

第四步:学习 Python 数据分析与应用。

学习 Numpy、Pandas 等与数据科学计算、数据分析相关的扩展库,能进行数据清洗、转换等预处理操作,会运用分组聚合等方法进行数据分析。

第五步:学习 Python 数据可视化。

学习 Matplotlib、Seaborn 等可视化扩展库,将数据分析结果以各类图表,如柱状图、散点 图、雷达图等形式呈现。

第六步:了解 Python 机器学习的相关知识。

了解有监督学习、无监督学习等常用的机器学习模型,如决策树、逻辑回归、卷积神经网络等。了解 Tensorflow、PaddlePaddle、Pytorch 等常用的深度学习框架。

完成以上步骤的学习,并勤加练习,你已基本完成了 Python 数据分析的学习,并掌握了 所需技能。

素养小课堂:

Python 的学习是一个系统的工程。本书着眼于第一步、第二步的 Python 编程思维、基本数据思维的养成。

(1)做好学习规划。根据自身情况,做好后续的 Python 学习规划,在 1~2 年内完成 Python 的系统学习。

(2) 勤于编程实践。在学习中一定要动手实践,从小程序开始,积跬步,至千里。

(3) 养成数据思维。大数据时代需要养成数据思维,并将其应用到工作、学习中,使学习、工作更高效和科学。

任务 1.2 PyCharm 和 Anaconda 联动的开发环境配置

1.2.1 任务引入

【任务描述】

小 T 了解到可进行 Python 编程的集成工具有很多,在使用 Python 进行编程之前,需要安

装和配置 Python 开发环境。在使用过程中,小T发现如果安装多个编程工具,那么其指向的 Python 解释器在默认情况不是同一个,默认工作路径也不相同,这会给后续学习 Python 造成 诸多不便。怎样搭建一个方便快捷的 Python 开发环境呢?让我们和小T一起,通过对本任务 的学习,搭建一个方便快捷的 Python 开发环境吧。

【任务分析】

先进行 Python 不同开发环境的配置,再进行在不同环境下使用同一个 Python 解释器、工作路径的配置。

- (1) Windows 操作系统下载部署 Python IDLE 开发环境。
- (2) Windows 操作系统下载部署 PyCharm 集成开发环境。
- (3) Windows 操作系统下载部署 Anaconda 集成开发环境。
- (4) 配置 PyCharm 和 Anaconda 联动的开发环境。

集成开发环境是一种辅助开发人员进行程序开发的应用软件,它集成了代码编写功能、分析功能、运行功能、调试功能等一体化的开发软件服务套件。在 Python 程序的开发过程中,常用的集成开发环境有 Python 自带的 IDLE、PyCharm、Jupyter Notebook 等,接下来针对几种流行的开发工具与环境进行介绍。

1.2.2 IDLE 的配置与使用

1. 安装文件下载

(1) 访问 Python 官网,选择【Downloads】→【Windows】选项,如图 1-2 所示。

电。	eython™ Python			Do	nate 🧠 Search	Jobs	Community GO Socialize
	About	Downloads		Community	Success Stories	News Even	
	<pre># For loop on a >>> numbers = [>>> product = 1 >>> for number product >>> print('The The product is:</pre>	All releases Source code Windows Mac OS X Other Platforms License Alternative Implement	Down Pythe Note ti or earl Not the many o View th Intations	load for Windows on 3.8.5 hat Python 3.5+ <i>cant</i> ier. t OS you are looking f pperating systems and se full list of downloar	not be used on Windows for? Python can be used o d environments. ds.	ents that of ;e — with s ;XP / tools in on	iher ome
		Python is a and inte	programming la egrate systems r	anguage that more effective	lets you work qu ly. <u>>>></u> <u>Learn Mor</u>	ickly <u>e</u>	

图 1-2 Python 官网

(2) 跳转到 Python 下载界面,如图 1-3 所示,该界面中包含多个版本的安装包,可以根据自身需求下载相应的版本,建议下载与本书一致的 Python 3.8.5 版本。

(3)选择适用于操作系统的安装包,如图 1-4 所示,这里我们选择 Python 3.8.5 版本下的 "Windows x86-64 executable installer"安装包,下载并保存到本地。

这里需要注意的是, Version 列中"Windows x86 …"的安装包适用于 32 位的 Windows 操 作系统,"Windows x86-64 …"的安装包适用于 64 位的 Windows 操作系统。executable 表示 安装包是独立的,包含所有的必需文件; 而 web-based 表示安装包只包含必要的安装引导程

序,需要在安装过程中根据安装选项从网络下载文件。



图 1-4 下载安装包

bc354669bffd81a4ca14f06817222e50

959873b37b74c1508428596b7f9df151

c813e6671f334a269e669d913b1f9b0c

SIG

SIG

SIG

7305731

26777232

1328184

2. 运行安装文件

Windows x86 embeddable zin file

Windows x86 executable installer

Windows x86 web-based installe

Windows

Windows

Windows

(1)双击安装文件,出现图 1-5 所示的安装向导界面。这里一定要勾选【Add Python 3.8 to PATH】复选框。有两种安装方式可供选择,一种是默认方式"Install Now",另一种是自定义 方式"Customize installation",这里选择自定义安装"Customize installation"方式,开始安装。 注意:如果在安装时没有勾选【Add Python 3.8 to PATH】复选框,则在使用 Python 解 释器之前要先手动将 Python 路径添加到环境变量中。环境变量的配置过程如下,右击【此 电脑】,在弹出的快捷菜单中选择【属性】命令,打开【系统】窗口,选择【高级系统设置】 选项,在【系统变量】列表中找到环境变量【Path】并双击,将 Python 的安装路径加入其中 即可。



认识 Python

(而日)

图 1-5 安装向导界面

(2) 配置 Python 的安装路径,点击【Install】按钮开始安装,安装成功后出现图 1-6 所示的安装完成界面。

	Python 3.8.5 (64-bit) Setup	×
		Setup was successful
电子		Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS. New to Python? Start with the <u>online tutorial</u> and <u>documentation</u> . See <u>what's new</u> in this release.
	python windows	Close

图 1-6 安装完成界面

(3) 在 Windows 操作系统中打开命令提示符窗口,输入"python"后显示 Python 的版本 信息,表示安装成功,如图 1-7 所示。

🚥 管理员: 命令提示符 - python	-		×
Microsoft Windows [版本 10.0.15063] (c) 2017 Microsoft Corporation。保留所有权利。			^
C:\Users\mtitan>python Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:57:54) [MSC v.1924 64 bit (AMD6 Type "help", "copyright", "credits" or "license" for more information. >>> _	4)] on	win32	~

图 1-7 命令提示符表示安装成功

3. Python 的运行方式

Python IDLE有两种运行方式:交互式和文件式。交互式是指Python 解释器逐行接收 Python

代码并即时响应运行; 文件式是指先将 Python 代码保存在文件中, 再启动 Python 解释器批量 解释代码。在安装好 Python IDLE 之后,选择【开始】→【Python3.8】→【IDLE...】菜单命令, 进入 Python 编程环境。

1) 交互式

在打开 Python IDLE 进入 Python 编程环境之后,默认进入的是交互式界面。在命令提示符 ">>>" 后面输入代码,按回车键即可运行代码,输出的运行结果如图 1-8 所示。



2) 文件式

在 Python IDLE 的交互式界面下,选择【File】→【New File】菜单命令,创建 Python 文件,在文件中写入代码并保存文件,如图 1-9 所示。可选择【Run】→【Run Module】菜单命 令或直接按 "F5" 键运行。





边学边练:

Windows 操作系统下载部署 Python IDLE 开发环境的过程如下。

(1) 根据计算机实际环境,在 Python 官网下载 Python IDLE。

(2)将其安装至电脑 C:\Python\IDLE 目录下。

(3) 在文件式运行方式下输入代码 "print('Hello World!')" 并运行。

1.2.3 集成开发环境 PyCharm 的配置与使用

PyCharm 是深受开发人员喜爱的 Python 开发工具,操作便捷、功能齐全,具有可调试、 语法高亮、项目管理、代码跳转、单元测试等功能。

1. 安装文件下载

访问 jetbrains 官网,打开下载 PyCharm 工具的界面,并选择所需版本,如图 1-10 所示。 PyCharm 包含 Professional 和 Community 两个版本。其中, Professional 版本支持 Python IDE

8

页目一	认识	Python
-----	----	--------

的所有功能以及 Web 开发,包括 Django、Flask、JavaScript 等各类框架和语言,且支持远程开发、数据库等功能;而 Community 版本只支持轻量级 Python IDE,但它是免费的。这里我们点击 Community 版本下的【Download】按钮,下载 PyCharm 安装包。



图 1-10 选择 PyCharm 版本

2. 运行安装文件

(1)下载成功后,双击"PyCharm Community"安装包,弹出欢迎界面,如图 1-11 所示。



图 1-11 PyCharm Community 欢迎界面

(2)点击【Next】按钮,进入 PyCharm 选择安装路径界面,如图 1-12 所示,建议路径不要太长,且不要包含中文字符。

(3)确定好安装路径后,点击【Next】按钮,进入安装选项界面,如图 1-13 所示。用户可以在该界面中根据需求选择相应功能,建议勾选左边三个复选框。

🖺 PyCharm Con	nmunity Edition Setup – 🗆 🗙	PyCharm Co	mmunity Edition Setup	– 🗆 ×
PC	Choose Install Location Choose the folder in which to install PyCharm Community Edition.	PC	Installation C Configure your	Aptions PyCharm Community Edition installation
Setup will install folder, click Brow Destination Fo	PyCharm Community Edition in the following folder. To install in a different was and select another folder. Click Next to continue.	Create Desk	top Shortcut incher text menu in Folder as Project [®] iciations	Update PATH variable (restart needed)
Space required: Space available:	768.5 MB : 16.7 GB < Back Next > Cancel			< Back Next > Cancel

图 1-12 选择安装路径界面

图 1-13 安装选项界面

(4) 保持默认配置,安装 PyCharm 后会跳转到图 1-14 所示的安装完成界面,提示 "Completing PyCharm Community Edition Setup"信息,点击【Finish】按钮完成安装。

1	PyCharm Community Edition Setup -
	PC Completing PyCharm Community Edition Setup
	PyCharm Community Edition has been installed on your computer.
	Click Finish to close Setup.
	Run PyCharm Community Edition
BFI	水版必元
	< Back Finish Cancel

图 1-14 安装完成界面

3. 使用 PyCharm

在初次使用 PyCharm 时会进入许可协议界面,选择接受许可协议单选按钮并点击确定按钮,跳过 UI 界面设置后,即进入创建新项目的主界面,如图 1-15 所示。



图 1-15 创建新项目主界面

10

认识 Python 项目—

(1) 打开【New Project】窗口,完成新建 Python 项目的相关设置,如图 1-16 所示,自定 义项目存储路径,定义关联 Python 解释器,也可以在选择 1.2.1 节中安装的 Python 解释器(在 Existing interpreter 中设置),点击【Create】按钮。此时弹出提示信息,可以选择在启动时不显 示提示。



图 1-16 新建 Python 项目相关设置

(2)在新建好项目之后创建 Python File,如图 1-17 所示。创建自定义的.py 文件,右击项 目名,在弹出的快捷菜单中选择【New】→【Python File】命令。

×.	Eile	<u>E</u> dit	<u>V</u> iew	<u>N</u> avigate	<u>C</u> ode	<u>R</u> efactor	Run	Iool	s VC	<u>S</u> <u>W</u> indo	w <u>H</u> elp	pythonPro	oject - main.py	- Administrato
ру	thor	Projec	t) 🎼 r											
ect								🐉 m	ain.py					
: Proj	~	pyth	onProj	ect C:\User						This is				
	;		New				► f							
	> II E	∥ E: ₩ E:					× ≝	' New : I Direc						replace it ywhere for
		ំ ំ					/ 🖻			kage				
			Find <u>U</u>	sages		Alt+F	7 🕅	Pytho HTMI						
			Replace				R \$							line below trl+F8 to t
			<u>R</u> efact											
														ter to run
			Add to							name				
			Optim Delete			Ctrl+Alt+C								etbrains.co

图 1-17 创建 Python File

(3) 在弹出的对话框中输入.py 文件名,点击【OK】按钮即可打开该文件。

1.2.4 集成开发环境 Anaconda 的配置与使用

在众多的 Python 开发环境中, Anaconda 因为集成了大量 Python 扩展库,并具有强大的数据处理与分析功能而深受开发人员的喜爱。此外, Anaconda 中的 conda 工具具有强大的扩展 库管理与环境管理功能,允许用户方便地安装不同版本的 Python 解释器和扩展库,并可以进行快速切换。这里以 Windows 操作系统为例,讲述 Anaconda 的安装和使用。

1. 安装文件下载

(1) 访问 Anaconda 官网,选择【Products】→【Individual Edition】选项,进入个人版下 载界面。在个人版下载界面中点击【Download】按钮,进入图 1-18 所示的安装包列表界面。



图 1-18 Anaconda 的安装包列表界面

(2) 选择并下载 Windows 操作系统对应的 Anaconda 安装包。

2. 运行安装文件

(1)下载成功后,双击该安装包会弹出欢迎界面。点击【Next】按钮进入许可协议界面, 点击【IAgree】按钮,选择【Just Me】或【All Users】单选按钮,进入路径设置界面,如图 1-19 所示。注意安装路径不宜太长,且不宜使用中文路径。

	0 Apaconda 2 2020 07 (6	(4. hit) Sotup	
		Choose Install Location Choose the folder in which to install /	Anaconda3 2020.07 (64-bit).
电子丁	Setup will install Anaconda folder, dick Browse and se	3 2020.07 (64-bit) in the following folde lect another folder. Click Next to continu	r, To install in a different .e.
	Destination Folder	nda3	Browse
	Space required: 2.7GB Space available: 14.8GB		
	Anaconda, Inc. ————	< Back	Next > Cancel

图 1-19 路径设置

(2)确定好安装路径后,点击【Next】按钮进入安装选项界面,如图 1-20 所示。在该界面中,如果是第一次安装,则两个复选框均可勾选。其中,第一个复选框是将 Anaconda 下的 Python 环境添加到环境变量中,如果不勾选此复选框,则需手动添加。

(3) 点击【Install】按钮进入安装过程,点击【Finish】按钮完成安装。

3. 使用 Anaconda

Anaconda 安装好后,可以选择【开始】→【Anaconda3(64-bit)】菜单命令启动相关开发环境,如图 1-21 所示。可以看到, Anaconda 下有 Jupyter Notebook、Spyder 等开发环境, Jupyter Notebook 在数据分析领域中用得较多,接下来以 Jupyter Notebook 为例讲述其用法。

 (1)选择【开始】→【Anaconda3(64-bit)】→【Jupyter Notebook(Anaconda3)】菜单命令, 打开 Jupyter Notebook 开发界面,可以发现这是一个网页界面,如图 1-22 所示。Jupyter Notebook

是基于网页的 Python 开发环境。

Anaconda3 2020.07 (64-bit) Setup	□ ×	
Advanced Installation Options Customize how Anaconda integrates with Windows		
Advanced Options Add Anaconda3 to the system PATH environment variable Not recommended. Instead, open Anaconda3 with the Windows Start menu and select "Anaconda (64-bit)". This "add to PATH" option makes Anaconda get found before previously installed software, but may cause problems requiring you to uninstall and reinstall Anaconda.		A Anaconda3 (64-bit) ^ Anaconda Navigator (Anaconda
Register Anaconda3 as the system Python 3.8 This will allow other programs, such as Python Tools for Visual Studio PyCharm, Wing IDE, PyDev, and MSI binary packages, to automatically detect Anaconda as the primary Python 3.8 on the system.		Anaconda Powershell Prompt (Anaconda Prompt (Anaconda3) Jupyter Notebook (Anaconda3)
Anaconda, Inc	Cancel	Reset Spyder Settings (Anacond
图 1-20 安装选项界面		图 1-21 Anaconda 下的开发环境
Jupyter		Quit Logout
me Move D	ス	Upload New - C Notebook:

Files	Running	Ciusters			
Rename	e Move 🛍		Upload	New 🕶	C
	- m/	Name d	Notebook:		
	-		Python 3		
	3D Objects		Other:		
	Contacts		Text File		
	Desktop		Folder		
	Documents		Terminal		
	Downloads		30 分钟前		
0	Favorites		4 个月前		
	🗅 Links		4个月前		
	🗅 Music		1 年前		
	D Nox_share		2年前		
v	D OneDrive		1年前		
	D Pictures		1 年前		
	Saved Games		4个月前		
	Searches		4 个月前		

图 1-22 Jupyter Notebook 开发界面

(2)选择右上角的【New】→【Python3】菜单命令,进入 Python 代码编写的网页界面。 在这个界面的每个 Cell 中输入代码块,点击【运行】按钮即可输出运行结果,如图 1-23 所示。

C Jupyter Untitled1 (未保存改变)	Logout
File Edit View Insert Cell Kernel Widgets Help	可信的 🖋 Python 3 O
□□ + ※ 42 15 + ↓ 月运行 ■ C → 代码 ✓ □□	
In [1]: print('Hello World!')	
Hello World!	
In []:	

图 1-23 Python 代码编写界面

(3) 查看 Anaconda 下已安装的扩展库。可选择【开始】→【Anaconda3(64-bit)】→【Anaconda Prompt(Anaconda3)】菜单命令,运行"conda list"命令,可查询当前环境下安装了哪些扩展库,如图 1-24 所示。

👞 命令提示符			_	\times
k	8.6.10	he774522_0		^
oml	0. 10. 2	pyhd3eb1b0_0		
oolz	0.11.1	pyhd3eb1b0_0		
ornado	6.1	py38h2bbff1b_0		
qdm	4.59.0	pyhd3eb1b0_1		
raitlets	5.0.5	pyhd3eb1b0_0		
/ped-ast	1.4.2	py38h2bbff1b_1		
yping_extensions	3. 7. 4. 3	pyha847dfd_0		
json	4.0.2	py38hd77b12b_0		
nicodecsv	0.14.1	py38_0		
rllib3	1.26.4	pyhd3eb1b0_0		
c	14.2	h21ff451_1		
s2015_runtime	14.27.29016	h5e58377_2		
atchdog	1.0.2	py38haa95532_1		
ewidth	0.2.5	py_0		
ebencodings	0.5.1	py38_1		
erkzeug	1.0.1	pyhd3eb1b0_0		
neel	0.36.2	pyhd3eb1b0_0		
idgetsnbextension	3. 5. 1	py38_0		
in_inet_pton	1.1.0	py38haa95532_0		
ln_unicode_console	0.5	py38_0		
Incertstore	0.2	py38_0		
Inpty	0.4.3			
apt	1.12.1	py38he774522_1		
rd	2.0.1	pyhd3eb1b0_0		
sxwriter	1.3.8	pyhd3eb1b0_0		
wings	0.23.0	py38haa95532_0		
lwt	1.3.0	py38_0		
nltodict	0.12.0	py_0		
2	5.2.5	h62dcd97_0		
am1	0.2.5	he774522_0		
apf	0.31.0	pyhd3eb1b0_0		
eromq	4.3.3	ha925a31_3		
fp	0.5.5	hd77b12b_6		
				\sim

图 1-24 Anaconda 当前的扩展库

边学边练:

Windows 操作系统下载部署 Anaconda 集成开发环境的步骤如下。

(1)根据计算机实际环境,在官网下载 Anaconda。

(2)将 Anaconda 安装至电脑 C:\Python\Anaconda 目录下。

(3) 打开 Jupyter Notebook, 输入代码 "print('Hello World!')"并运行。

1.2.5 任务实现——PyCharm 和 Anaconda 联动的开发环境配置

根据 1.2.2~1.2.4 节边学边练的要求,小 T 已安装好了 Python IDLE、PyCharm 和 Anaconda 3 种 Python 开发环境,每种开发环境都有自己的 Python 解释器,不同解释器下安装有不同的 扩展库,这会对后续学习 Python 带来不便。小 T 对 PyCharm 和 Anaconda 联动的开发环境进 行配置,鉴于 Anaconda 强大的环境管理功能,将 PyCharm 中的 Python 解释器配置为 Anaconda 中的 Python 解释器,并将 D 盘下的 PythonCode 目录作为工作目录,在学习和书写 Python 代 码时使用。

边学边练:

Windows 操作系统下载部署 PyCharm 集成开发环境的步骤如下。

(1) 根据计算机实际环境,在官网下载 PyCharm Community 版本。

(2)将 PyCharm 安装至电脑 C:\Python\PyCharm 目录下。

(3) 打开 PyCharm, 输入代码 "print('Hello World!')"并运行。

【任务分析】

(1) 创建工作路径 D:\PythonCode。

(2)将 PyCharm 中解释器配置为 Anaconda 中的解释器,即 C:\Python\Anaconda3 下的 Python 解释器,并将 PyCharm 的工作路径配置为 D:\PythonCode。

(3) 配置 Anaconda 的工作路径为 D:\PythonCode。

【任务实现】

1. 创建工作路径

在 D 盘下创建目录 PythonCode 并将其作为工作路径。

2. 配置 PyCharm 解释器和工作路径

打开 PyCharm 的项目创建界面,配置【Location】选项为工作路径"D:\PythonCode", "Interpreter"路径为"C:\Python\Anaconda3",如图 1-25 所示。

Location: D:\Python	Code		
 Python Interprete 	r: Python 3.8 (base)	工作改经配置	
	nt using 🔹 Virtualenv 🔹		
	: Yython 3.9 https://www.python.org/ftp/python/3.9.5/pytho		
	gured interpreter		
Interpreter:	Python 3.8 (base) C:\Python\Anaconda3\python.exe		
	welcome script of that provides an entry point to coding in PyCharm.		

图 1-25 配置 PyCharm 工作路径和解释器

配置完成后,新建一个 Python 文件,选择【File】→【Settings】菜单命令,打开【Settings】 界面,通过【Python Interpreter】下拉列表可以看到 Python 解释器已是 Anaconda 中的解释器, 扩展库也与 Anaconda 默认环境下的扩展库一致,如图 1-26 所示。

3. 配置 Anaconda 的工作路径

(1) 配置 Jupyter Notebook 文件路径。选择【开始】→【Anaconda3(64-bit)】→【Anaconda
 Prompt(Anaconda3)】菜单命令,输入"jupyter notebook --generate-config",按回车键。得到 Jupyter Notebook 配置文件路径,如图 1-27 所示。

(2)更改 notebook_dir 目录。根据配置的文件路径,找到对应目录下的配置文件 "jupyter_notebook_config.py",以记事本方式打开该文件,找到"notebook_dir"目录,将其更 改为工作路径"D:\PythonCode"。注意在修改时去掉 c.NotebookApp.notebook_dir 前的"#"及 空格,如图 1-28 所示。

> Appearance & Behavior	Python Interpreter: 🔿 Python 3.8 (t			
Keymap				
> Editor	+ 00			
Plugins =				
> Version Control				
✓ Project: PythonCode				
	anaconda-client		▲ 1.8.0 ▲ 20.4	
> Build, Execution, Deployment	anaconda-navigator anaconda-project	2.0.3	▲ 2.0.4 ▲ 0.10.1	
> Languages & Frameworks	anvio			
> Tools	appdirs			
Advanced Settings				
	astropy	4.2.1	▲ 4.3.1 1.10	
	atomicwrites			
	attrs	20.3.0	▲ 21.2.0	
	autopep8			
	backports.functools_lru_cache	1.6.4	1.6.4	
	图 1-26	【Settings】界		
	<u>д</u> •			
🖬 Anaconda Prompt (Anaconda3)			-	
base) C:\Users\86153>jupy	ter notebookgenerate-config			^
riting default config to:	C:\Users\86153\.jupyter\jupyte	er_notebook_config.py		
base) C:\Users\86153>a				

图 1-27 Jupyter Notebook 配置文件路径

<pre></pre>		
<pre>文件(F) 編輯(E) 施式(C) 查看(V) 帮助(H) # Default: 536870912 # c.NotebookApp.max_buffer_size = 536870912 ## Gets or sets a lower bound on the open file handles process resource limit. # This may need to be increased if you run into an OSError. [Errno 24] Too many # open files. This is not applicable when running on Windows. # Default: 0 # c.NotebookApp.nnin_open_files_limit = 0 ## Dict of Python modules to load as notebook server extensions.Entry values can # be used to enable and disable the loading ofthe extensions. The extensions # will be loaded in alphabetical order. # Default: 0 # c.NotebookApp.nbserver_extensions = {} ## The directory to use for notebooks and kernels. # Default: " :NotebookApp.notebook_dir = 'D'\PythonCode'</pre>) X
<pre># Default: 536870912 # c.NotebookApp.max_buffer_size = 536870912 ## Gets or sets a lower bound on the open file handles process resource limit. # This may need to be increased if you run into an OSError: [Errno 24] Too many # open files. This is not applicable when running on Windows. # Default: 0 # c.NotebookApp.min_open_files_limit = 0 ## Dict of Python modules to load as notebook server extensions.Entry values can # be used to enable and disable the loading of the extensions. The extensions # will be loaded in alphabetical order. # Default: {} # c.NotebookApp.nbserver_extensions = {} ## The directory to use for notebooks and kernels. # Default: " .NotebookApp.notebook_dir = 'D'.PythonCode'</pre>	i著(V) 帮助(H)	
<pre># c.NotebookApp.max_buffer_size = 536870912 ## Gets or sets a lower bound on the open file handles process resource limit. # This may need to be increased if you run into an OSError. [Errno 24] Too many # open files. This is not applicable when running on Windows. # Default: 0 # c.NotebookApp.min_open_files limit = 0 ## Dict of Python modules to load as notebook server extensions.Entry values can # be used to enable and disable the loading ofthe extensions. The extensions # will be loaded in alphabetical order. # C.NotebookApp.nbserver_extensions = {} ## The directory to use for notebooks and kernels. # Default: "</pre>	2	^
<pre>## Gets or sets a lower bound on the open file handles process resource limit. # This may need to be increased if you run into an OSError: [Errno 24] Too many # open files. This is not applicable when running on Windows. # Default: 0 # c.NotebookApp.min_open_files_limit = 0 ## Dict of Python modules to load as notebook server extensions.Entry values can # be used to enable and disable the loading ofthe extensions. The extensions # vill be loaded in alphabetical order. # Default: 0 # c.NotebookApp.notebooks and kernels. # Default: " :NotebookApp.notebook_dir = 'D:\PythonCode'</pre>	3x_buffer_size = 536870912	
<pre>## Dict of Python modules to load as notebook server extensions.Entry values can # be used to enable and disable the loading ofthe extensions. The extensions # will be loaded in alphabetical order. # Default: {} # Default: {} # The directory to use for notebooks and kernels. # Default: " :NotebookApp.notebook_dir = 'D:\PythonCode'</pre>	er bound on the open file handles process resource limit. ee increased if you run into an OSError: [Errno 24] Too many ot applicable when running on Windows. n_open_files_limit = 0	
## The directory to use for notebooks and kernels. # Default: '' :NotebookApp.notebook_dir = 'D:\PythonCode'	vdules to load as notebook server extensions.Entry values can and disable the loading ofthe extensions. The extensions phabetical order. server_extensions = {}	
	ise for notebooks and kernels. book_dir = 'D:\PythonCode'	
## Whether to open in a browser after starting. The specific browser used is # platform dependent and determined by the python standard library `webbrowser` # module, unless it is overridden using thebrowser (NotebookApp.browser)	in a browser after starting. The specific browser used is nt and determined by the python standard library `webbrowser` s overridden using thebrowser (NotebookApp.browser)	~
株 2015 (二)株 44 天日 - 1000(/ - 146 - 1-100 / - 115 - 1		

图 1-28 更改 notebook_dir 目录

(3) 修改 Jupyter Notebook 启动路径。选择【开始】→【Anaconda3(64-bit)】菜单命令,右击【Jupyter Notebook(Anaconda3)】,打开文件位置。右击【Jupyter Notebook】,在弹出的快捷菜单中选择【属性】命令,出现图 1-29 所示的界面,修改【目标】中的参数,即删除"'%USERPROFILE%'",并点击【确定】按钮。

	🚊 Jupyter No	tebook (A	(naconda3)	属性			×	
	兼容性		安全	详	田信息	以前的	的版本	
	常规	央捷方式	选项	字体	布局	颜色	终端	
	N. N.	Jupyter	Notebook	(Anaconda	13)			
	目标类型: 目标位置:	应用 Ana	程序 conda3					
	目标(T):	∣ts\j	upyter-note	ebook-scrij	pt.py "%US	RPROFILE	%"	
	起始位置(S):	%H	OMEPATHS	%				KA
	快捷键(K):	无						
	运行方式(R):	常規	國口			JL	Ĭ	XTI
	备注(O):	Jup	yter Noteb	ook (Anacc	onda3)	H		
	打开文件	听在的位置	1(F)	更改图标(C).		高级(D)		
T		3	11		ど	Ŧ		
1				确定	取消		应用(A)	

图 1-29 修改 Jupyter Notebook 启动路径

(4) 启动 Jupyter Notebook,确认默认工作路径已变更为"D:\PythonCode"。

边学边练:

在完成1.2.3、1.2.4节边学边练的基础上,完成以下任务。

(1)将 PyCharm 中的 Python 解释器设置为 Anaconda 中的解释器。

(2) 在 D 盘下创建 Pythoncode 目录,用来存放本任务的所有 Python 代码。

(3)将 PyCharm、Jupyter Notebook 的默认工作路径设置为(2)所建的路径。

任务 1.3 Python 扩展库的安装

【任务描述】

在后面的学习中,小T会用到 Matplotlib 第三方扩展库绘制图表。导师告诉他,Python 中的标准库可以直接导入使用而无须安装,但第三方扩展库必须在安装后方可使用,并且有不同的安装方法,根据实际情况可选择相应的安装方法。本任务会讲解 Python 扩展库安装的基本方法。

【任务分析】

(1) 了解 Python 扩展库管理的常用方法。

(2) 用不同方法安装 Python 扩展库。

(3) 查看、卸载 Python 扩展库。

在 Python 扩展库的官网中可以搜索和下载各种各样的扩展库。在 Python 中安装扩展库有 3 种常用的方法: pip 命令安装、tar.gz 文件安装和.whl 文件安装。

1.3.1 pip 命令安装

1. pip 包管理工具

pip 是一个安装和管理 Python 的工具,使用起来非常方便,省去了手动搜索、查找版本、 下载、安装等一系列烦琐的步骤,而且能自动解决包依赖的问题。

Python 3.4 以上的版本一般都自带 pip 工具。用户可以通过 cmd 命令行的方式,输入 "pip -V" 命令确认 pip 是否安装,并查看当前的 pip 版本号,如图 1-30 所示。

☞ 管理员: 命令提示符		715	-
Microsoft Windows [版本 10.0.15063] (c) 2017 Microsoft Corporation。保留所有权利。			
C:\Users\mtitan>pip -V pip 20.1.1 from c:\users\mtitan\appdata\local\programs\python\python5	88\lib\site-packages	\pip (pytho	n 3.8)

图 1-30 确认 pip 版本

2. pip 安装命令

在使用 pip 命令安装扩展库时,需要连接互联网下载并安装扩展库。使用 pip 命令安装扩 展库非常简单,这里以 Matplotlib 库为例进行演示。

(1) 安装最新版本的 turtle 库:

```
pip install matplotlib
```

默认获取当前最新版本的安装包进行安装。

(2) 安装指定版本的 turtle 库:

pip install matplotlib==3.1.3

使用"=="指定过去的某个版本,通常是为了在协作开发时与他人或公司的环境保持一致。

(3) 查看当前 matplotlib 库的版本:

pip show matplotlib

使用该命令会显示已安装的 matplotlib 的具体版本和安装路径等信息。

(4) 卸载 matplotlib 库:

pip uninstall matplotlib

只需这一行命令即可轻松地将已安装的库卸载。

(5) 显示安装包 matplotlib 库:

pip show matplotlib

使用该命令可以获得该安装包的名称、版本、作者、安装在本地的地址等信息。

(6) 显示所有已安装的包:



19

pip list

使用该命令可以获得所有已安装的包的名称和版本信息。

1.3.2 tar.gz 文件安装

使用 pip 命令安装扩展库非常方便,但并不是所有的扩展库都能用它来安装,有的扩展库可能只提供了源码压缩包文件,或者安装环境不能上外网,这时就可以用 tar.gz 文件来安装。

可以先在 Python 扩展库官网中搜索要安装的第三方库的库名,然后在找到的扩展库界面中点击【Download files】按钮,即可看到提供的下载文件,如图 1-31 所示。

Navigation	Download files
■ Project description	Download the file for your platform. If you're not sure which to choose, learn more about installing packages 🗹.
3 Release history	
🛓 Download files	Flask-2.2.2.tar.gz 677.4 kB view hashes) Uploaded Aug 9, 2022 source 677.4 kB view hashes)
Project links	
A Homepage	Built Distribution
📜 Changes	Liploaded Aug 9, 2022 pv3
Chat	

图 1-31 扩展库下载文件信息

点击文件名即可下载 tar.gz 压缩包,在本地解压缩之后,进入文件目录,执行下列命令: python3 setup.py install

虽然只有一行命令,但是相比 pip 命令,这种方式是相对复杂的,尤其是在某个扩展库有 很多依赖库时,就更不方便了。

1.3.3 .whl 文件安装

目前有超过一半的库文件有对应的 wheel 文件,其形式是扩展名为.whl 的文件。它本质上 是一个压缩包,专门用于 Python 模块的安装,用户可以通过 pip 工具中内置的 wheel 子命令 来安装。

在图 1-31 中可以看到 Python 用于 Web 开发的 Flask 扩展库文件 Flask-2.2.2-py3-none-any.whl,将它下载并保存到本地,执行下列 pip 命令:

pip install Flask-2.2.2-py3-none-any.whl

边学边练: 请完成下面任务。 (1)使用 pip 命令在线安装 Django 库。 (2)查看安装好的 Django 版本,然后卸载 Django。 (3)下载 Django 的.whl 文件,并将其进行安装(选做)。





D. 执行 Python 程序

- 10. 下列() 不是 Python 的应用领域。
 - A. 科学计算
 B. Web 开发
 - C. 数据分析
- **D**. 操作系统管理

实训任务 2:操作题

- 1. 在电脑上完成本章边学边练的所有任务。
- 2. 以下是由字符组成的"超级玛丽"图形,编写一个简单的 Python 程序打印出此图形。

#####.
####
#########
##*#######
####*****
· · · # * * * · * * * * · * # # # · · · ·
····*******##·····
····**** *****····
####
####### #######

###.#####.######.

###.#####.####.
