第1章 Python 概述

学习目标

- □ 了解 Python 的应用领域、特点、程序运行机制。
- □ 了解 Python 的版本,熟悉 Python 2.x 和 Python 3.x 的区别。
- □ 掌握如何搭建 Python 开发环境。
- □ 掌握 Python 程序的编写方法。
- □ 掌握如何调试 Python 程序。
- □ 熟悉扩展库的安装方法。

任务导入

当前,不仅高校开设了 Python 课程,很多中学也开设了 Python 课程,甚至连小学生 也在学习 Python。为何 Python 这么热门?因为 Python 简单易学,初学者很容易就能完成 自己的第一个程序,并输出结果;使用 Python 能实现网络爬虫、可为批量图片添加水印……

那么有如下几个问题。

- (1) Python 有哪些应用领域?
- (2) Python 环境如何搭建?
- (3) Python 如何使用?

随着大数据和人工智能的兴起, Python 这门"古老"的语言在这两个领域大放异彩, 这使得 Python 语言变得非常流行。实际上, Python 一直是一门优秀的编程语言,不仅简洁、 易用,而且功能强大。它既可用于开发桌面应用,也可用于网络编程,还可用于开发 Web 应用等多个领域。

本章重点介绍如何搭建 Python 的开发环境。

1.1 Python 语言简介

Python 是一种面向对象编程语言,它是一种既简单又功能强大的编程语言。Python 注 重如何解决问题而不是编程语言的语法和结构。使用 Python 语言编写的程序是跨平台的。

1.1.1 什么是 Python

Python 是一种面向对象的解释型编程语言。

Python 英文原意为"蟒蛇"。1989 年,荷兰人 Guido van Rossum(简称 Guido,音译 一般为吉多•范罗苏姆)发明了一种面向对象的解释型编程语言,并将其命名为 Python。

Python 语言是基于 ABC 语言的,而 ABC 这种语言非常强大,是专门为非专业开发者 设计的。但 ABC 语言并没有获得广泛的应用,Guido 认为是非开放造成的。

基于这个考虑, Guido 在开发 Python 时不仅为其添加了很多 ABC 没有的功能,还为其 设计了各种丰富而强大的库。利用这些 Python 库,开发者可以把使用其他语言(尤其是 C 语言和 C++)制作的各种模块轻松地集成在一起。所以, Python 常被称为"胶水"语言。

1991 年发行第一个公开发行版。目前, Python 的最新发行版是 Python 3.10.0 (2021 年 10 月 4 日发布)。

1.1.2 Python 的应用领域

Python 是一种跨平台的编程语言。理论上,Python 可以在任何操作系统平台上运行。 目前,最常用的操作系统平台是 Windows、Mac OS X 和 Linux。

Python 简单易学、有众多第三方程序库、运行速度快,这些特性让 Python 的应用领域 十分广泛。

Python 的应用领域主要有如下 10 个。

- □ Web 程序开发。
- □ 大数据处理。
- □ Linux/UNIX 运维。
- □ 云计算。
- □ 人工智能。
- □ 网络爬虫。
- □ 服务器端程序开发。
- □ 命令行程序开发。
- □ 移动 APP 开发。
- □ GUI 程序开发。

尽管这里没有列出 Python 的所有应用领域,但也已包含了绝大多数的开发场景。

在 Mac OS X 和 Linux 这两个操作系统中,已经内置了 Python 开发环境;也就是说, Python 程序可以在 Mac OS X 和 Linux 上直接运行。所以,很多运维工程师都习惯使用 Python 来完成自动化操作。

Python 在操作网络和文本方面的能力也尤为突出。Google 搜索引擎的第一个版本就是用 Python 编写的。而且 Python 也已成为深度学习的第一语言。

因此,从各个角度来看,无论是学生、开发者,还是数据分析师或科学家,都离不开 Python。Python 已然成为编程语言领域的世界语。

1.1.3 Python 的特点

(1) 语法简单。Python 是一种代表极简主义的编程语言,阅读一段排版优美的 Python 代码就像在阅读一个英文段落;所以,人们常说 Python 是一种具有伪代码特质的编程语言。

(2)易于学习。Python 有相对较少的关键字,结构简单,还有一个明确定义的语法, 学习起来比较简单。

(3)免费、开源。用户使用 Python 进行开发或者发布自己的程序,不需要支付任何费用,也不用担心版权问题;即使作为商业用途,Python 也是免费的。开源即开放源代码,意思是所有用户都可以看到源代码。开发者使用 Python 编写的代码是开源的;Python 解释器和模块也是开源的。

(4) 自动内存管理。Python 会自动管理内存,需要时自动分配,不需要时自动释放。

(5) 解释型。Python 程序可以直接使用源代码运行,不需要编译成二进制代码。在计算机内部,Python 解释器把源代码编写转换成字节码的中间形式,然后再把它翻译成计算机使用的机器语言并运行。

(6) 可移植性。基于其开放源代码的特性, Python 已经被移植(即工作)到许多平台。

(7)可扩展性。如果需要一段运行很快的关键代码,或者是想要编写一些不愿开放的算法,可以使用 C 或 C++完成编写,然后在 Python 程序中调用那部分程序。

(8)面向对象。Python 既支持面向过程的编程,也支持面向对象的编程。在面向过程的语言中,程序是由过程或仅仅是可重用代码的函数构建起来的。在面向对象的语言中,程序是由数据和功能组合而成的对象构建起来的。与 C++和 Java 相比, Python 以一种更加简单的方式实现了面向对象编程。

(9) 广泛的标准库。Python 的标准库非常庞大,可以处理各种工作,包括正则表达式、 文档生成、单元测试、线程、数据库、网页浏览器、CGI、FTP、电子邮件、XML、XML-RPC、 HTML、WAV 文件、密码系统、GUI(图形用户界面)、Tk 和其他与操作系统有关的操作。

(10) 规范的代码。Python 采用强制缩进的方式,所以代码具有极佳的可读性。

1.1.4 Python 程序运行机制

Python 是一门解释型的编程语言,因此它具有解释型语言的运行机制。

计算机是不能理解高级语言的,更不能直接执行高级语言,它只能直接理解机器语言。 所以,使用任何高级语言编写的程序若想被计算机运行,都必须将其转换成计算机语言, 也就是机器码。

高级语言的转换方式有两种:编译和解释。因此,高级语言分为编译型语言和解释型 语言。

1. 编译型语言

编译型语言使用专门的编译器,针对特定平台(操作系统),将某种高级语言的源代 码一次性"翻译"成可被该平台硬件执行的机器码(包括机器指令和操作数),并包装成 该平台所能识别的可执行程序的格式。

编译型语言的特点:编写的程序在执行之前,需要一个专门的编译过程,把源代码编译成机器语言的文件,如.exe 格式的文件。以后,需要再运行时,直接运行.exe 文件即可。因为只需编译一次以后运行时就不再需要编译,所以编译型语言执行效率高。

编译型语言的执行方式,如图 1-1 所示。



图 1-1 编译型语言的执行方式

编译型语言总结如下。

- (1) 一次性编译成平台相关的机器语言文件,可脱离开发环境独立运行,效率较高。
- (2) 与特定平台相关,一般无法移植到其他平台。
- (3) 现有的 C、C++、Objective Pascal 等高级语言都属于编译型语言。

2. 解释型语言

解释型语言使用专门的解释器将源程序逐行解释成特定平台的机器码并立即执行。 解释型语言的特点:不需要事先编译,直接将源代码解释成机器码并立即执行,所以 只要平台提供了相应的解释器即可运行该程序。解释型语言的执行方式,如图 1-2 所示。



图 1-2 解释型语言的执行方式

解释型语言总结如下。

- (1)每次运行都需要将源代码解释成机器码并执行,效率较低。
- (2)只要平台提供相应的解释器就可以运行源代码,所以可以方便地实现源程序的移植。

1.1.5 如何学好 Python

如何学好 Python 语言,是所有初学者共同面对的问题。不管是学习哪种编程语言,学 习方法都大同小异。下面给出几点建议如下。

(1)对 Python 语言有一个宏观的认识。也就是要了解 Python 能做什么,擅长做什么。

(2) 搭建开发环境,先弄出一个"Hello World!"(可以复制现有的代码)。以增强 自己继续学习 Python 语言的信心。

(3)不要深究设计模式。因为设计模式是给拥有多年经验的开发者准备的。所以,在 开始写程序时,只管写就好了。

(4)模仿书中的代码,自己亲手写代码。刚开始为了运行结果,可以复制代码。但后 来一定要每一个字符自己写一遍,代码可以与例子的不同,只要能实现相同的功能即可。

(5) 对于初学者,应充分利用教材给出的代码多做练习。多写代码,多输入代码。

- (6) 多做练习题。
- (7) 养成经常总结的好习惯。
- (8) 多阅读源代码。

1.2 Python 版本简介

Python 自发布以来,主要有以下3个版本。

- (1) 1994 年发布的 Python 1.0 版本(已过时)。
- (2) 2000 年发布的 Python 2.0 版本(Python 2.7 于 2020 年后终止支持)。

(3) 2008 年发布的 Python 3.0 版本(2021 年 10 月已更新到 3.10.0)。

Python 版本更迭的历史如下。

- □ 1994 年 1 月, Python 1.0。
- □ 2000 年 10 月 16 日, Python 2.0。
- □ 2001 年 11 月 21 日, Python 2.2。
- □ 2003 年 7 月 29 日, Python 2.3。
- □ 2004 年 11 月 30 日, Python 2.4。
- □ 2006 年 9 月 19 日, Python 2.5。
- □ 2008 年 10 月 2 日, Python 2.6。
- □ 2008 年 12 月 3 日, Python 3.0。
- □ 2009 年 6 月 26 日, Python 3.1。
- □ 2010 年 7 月 3 日, Python 2.7。
- □ 2011 年 2 月 20 日, Python 3.2。
- □ 2012 年 9 月 29 日, Python 3.3。
- □ 2014 年 3 月 17 日, Python 3.4。
- □ 2015 年 9 月 13 日, Python 3.5。
- □ 2016 年 12 月 23 日, Python 3.6。
- □ 2018 年 3 月 10 日, Guido 在邮件列表上宣布 Python 2.7 将于 2020 年 1 月 1 日终止支持,并且不会推出 2.8 版本,希望用户尽快迁移至 3.4 以上的版本。
- □ 2018 年 6 月 27 日, Python 3.7。
- □ 2019 年 10 月 14 日, Python 3.8。
- □ 2020 年 10 月 5 日, Python 3.9, 但不支持 Windows 7 及更早版本的操作系统。
- □ 2021 年 10 月 4 日, Python 3.10, 但不支持 Windows 7 及更早版本的操作系统。

1.2.1 初学者应该选择哪个版本

基于 Python 创始人宣布将 Python 2.7 支持时间只延长至 2020 年。建议初学者选择 Python 3.x 版本,理由如下。

- (1) 目前, 使用 Python 3.x 是大势所趋。
- (2) Python 3.x 在 Python 2.x 基础上做了功能升级。

Python 3.x 对 Python 2.x 的标准库进行了重新拆分和整合。在字符编码方面比 Python 2.x

更容易理解。而且, Python 3.x 对中文字符的支持性能更好, 能正确显示中文。

(3) Python 3.x 和 Python 2.x 思想基本是共通的。

Python 3.x 和 Python 2.x 思想基本是共通的,只有少量的语法差别。学会了 Python 3.x,只要稍稍花点时间学习 Python 2.x 的语法,就可以灵活运用两个版本了。

当然,选择 Python 3.x 也有缺点,那就是很多扩展库的发行总是滞后于 Python 的发行 版本,甚至目前还有很多库不支持 Python 3.x。

1.2.2 Python 2.x 的代码转换成 Python 3.x 的代码

Python 2.x 与 Python 3.x 的差别很大, Python 2.x 的多数代码不能直接在 Python 3.x 环境下运行。在网上查找到的 Python 2.x 代码,需要修改后才能在 Python 3.x 环境下运行。 针对此问题, Python 官方提供了 Python 2 自动转换为 Python 3 的方法,就是小工具 2to3.py。 使用该工具可以将大部分 Python 2.x 代码转换为 Python 3.x 代码。

2to3.py工具的具体操作如下。

(1) 找到 2to3.py 文件。2to3.py 文件的位置在 Python 安装路径下的 Tools\scripts 文件 夹下。例如,安装的是 anaconda3; 所以,目录是 Anaconda3\Tools\scripts,如图 1-3 所示。

중 ♥ ₩ ≪ 本地磁盘	(C:) ▶ P	rogramData 🕨 Anaconda3 🕨 Tools	► scripts ► -	— □ X 4 搜索 sc ₽
文件(E) 编辑(E) 查看(V)	工具(I	帮助(<u>H</u>)		
组织 ▼ 📓 打开 ▼	刻录	新建文件夹	8=	- 🗌 🔞
🚖 收藏夹	^	名称	修改日期	类型 🔺
■ 桌面	=	L_pycache_	2020/1/16 16:43	文件夹
🐉 最近访问的位置		📓 2to3.py	2019/3/26 5:21	PY 文件
🔰 下载		abitype.py	2019/3/26 5:21	PY 文件
		analyze_dxp.py	2019/3/26 5:21	PY 文件
■ 桌面		byext.py	2019/3/26 5:21	PY 文件
[]] 库		byteyears.py	2019/3/26 5:21	PY 文件 🚽
📢 家庭组				Þ.
2to3.py 修改 PY 文件	日期: 20 大小: 96	19/3/26 5:21 创建日期: 2019/: 字节	3/26 5:21	

图 1-3 2to3.py 工具的位置

(2) 复制 2to3.py 文件到要转换代码所在的目录下。

(3) 打开"命令提示符"窗口。

方法一:选择"开始"命令,然后选择"所有程序",接着选择"附件",最后选择 "命令提示符"命令。

方法二:选择"开始"命令,在"搜索程序和文件"文本框中输入 cmd 命令,最后按 Enter 键。

(4) 进入转换代码文件所在的目录。例如,转换代码文件在 D:\temp 文件夹下。在"命 令提示符"窗口输入 d:按 Enter 键,再输入 cd temp 按 Enter 键(注意: cd 和 temp 之间有 一个空格符),如图 1-4 所示。

(5) 调用 2to3.py 工具转换代码。例如,要转换的文件名为 test.py,则在图 1-4 的 "命 令提示符"窗口输入命令 python 2to3.py -w test.py (注意: python、2to3.py、-w 和 test.py

之间都有一个空格符),如图 1-5 所示。



图 1-5 调用 2to3.py 工具转换代码

- 【小贴士】上面的代码执行后,会在 D:\temp 文件夹下创建一个名为 test.py.bak 的备份文件, 同时,原 test.py 文件的内容被转换为 Python 3.x 对应的代码。 注意,尽量不要把想要转换的代码保存在 C 盘。因为可能会因为权限问题导致 转换不能正常完成。
- 【小贴士】本书将以 Python 3.x 来介绍 Python 编程,本章也会简单对比 Python 2.x 和 Python 3.x 的语法差异。

1.2.3 Python 2.x 和 Python 3.x 的区别

Python 3.x 自带的 2to3.py 工具可以将 Python 2.x 程序源文件作为输入,然后自动转换 成 Python 3.x,但并不是所有内容都可以自动转换。

关于 Python 2.x 与 Python 3.x 的比较,可参考链接文件如下。 https://www.cnblogs.com/meng-wei-zhi/articles/8194849.html 以下列举几项 Python 2.x 和 Python 3.x 的区别。

1. print

Python 2.x 中 print 是一个语句, 要输出的内容直接放到 print 关键字后面即可。Python 3.x

里, print()是一个函数,像其他函数一样, print()需要将输出的内容作为参数传递给它。 Python 2.x 和 Python 3.x 中 print 用法的差别,如表 1-1 所示。

Python 2.x	Python 3.x	备注
print	print()	输出一个空白行, Python 3.x 需要调用不带参数的 print()
print 1	print(1)	输出一个值,将值传入 print()函数
print 1,2	print(1,2)	输出使用空格分割的两个值,使用两个参数调用 print()
print 1,2,	print(1,2,end=' ')	Python 2.x 中如果使用一个","作为 print 结尾,将会用空格分割输出的结果;然后再输出一个尾随的空格,而不输出回车符号。Python 3.x 里,把 end=''作为一个关键字传给 print()可以实现同样的效果,end 默认值为'\n',所以通过重新指定 end 参数的值,可以取消在末尾输出回车符号

表 1-1 print 用法比较

2. 字符串类型

Python 2.x 中有两种字符串类型: Unicode 字符串和非 Unicode 字符串。Python 3.x 中 只有一种类型: Unicode 字符串。

3. 数据类型

Python 2.x 为非浮点数准备了 int 和 long 两种类型。int 类型最大值不能超过 sys.maxint, 这个最大值与平台相关。可以通过在数字的末尾附上一个 L 来定义长整型, long 类型比 int 类型表示的数字范围更大。在 Python 3.x 里, 只有 int 类型一种; 大多数情况下, 和 Python 2.x 中的长整型类似。

4. 不等于运算符

Python 2.x 中 " <> " 和 "!=" 意思相同,都可以使用。但 Python 3.x 中只能用 "!="。

5. 编码

Python 2.x 的默认编码是 ASCII。在使用 Python 2.x 的过程中经常会遇到编码问题,当时由于 Python 语言还没使用 Unicode (UTF-8),所以使用 ASCII 作为默认编码。Python 3.x 默认使用的编码是 Unicode,可以更好地支持中文或其他字符,也就不需要在文件头部写 # coding=utf-8。

6. 除法运算符

Python 中有两个除法运算符: "/"和"//"。

其中, "//"运算符在 Python 2.x 和 Python 3.x 中是一致的,都是 floor 除法。floor 除 法对结果做向下取整处理。(向下取整:向-∞方向取最接近精确值的整数。在这种取整方 式下, 5 // 3 = 1, -5 // -3 = 1, -5 // 3 = -2, 5 // -3 = -2)

但是"/"运算符在 Python 2.x 和 Python 3.x 中的结果不同。在 Python 2.x 中,整数相除的结果是整数,有浮点数参与运算结果是浮点数。而在 Python 3.x 中,结果都是浮点数,包括两个整数相除后,结果也是浮点数。Python 2.x 和 Python 3.x 中除法运算的差别,如

表 1-2 所示。

Python 2.x	Python 3.x	备注
1/2=0	1/2=0.5	Duther 2 中 日 西 方 涇 占 粉 会 上 运 管
1.0/2=0.5	1.0/2=0.5	任 Fyllion 2.X 中, 六安有行点数参与, 运异 结用計旦浮占粉
4/2.0=2.0	4/2.0=2.0	年 Puthon3 x 由 结果都县浮占粉
9/2=4	4/2=2.0	任 Fyllion5.x 中, 纪禾郁龙行点数
5//2=2	5//2=2	
5.0//2=2.0	5.0//2=2.0	"//"运算中不管两者出现任何数,都以整
5//2.0=2.0	5//2.0=2.0	除结果为准,对结果做向下取整处理
-15//10=-2	-15//10=-2	

表 1-2 除法运算符比较

7. 输入函数

Python 2.x 中有两个全局函数,用在命令行请求用户输入。第一个是 input(),它等待用 户输入一个 Python 表达式(然后返回结果)。第二个是 raw_input(),获取用户的原始输入,即用户输入什么它就返回什么。

而 Python 3.x 中,只有 input()函数。并且 Python 3.x 中的 input()与 Python 2.x 中的 raw_input()功能一样。

Python 2.x 会自动识别类型,如 str、int、float;而 Python 3.x 只会将输入默认为 str 类型。Python 2.x 和 Python 3.x 中输入函数用法的差别,如表 1-3 所示。

	Pytho	on 2.x	Python 3.x
代码	a=input() print(a)	a=raw_input() print(a)	a=input() print(a)
输入	1+2	1+2	1+2
显示结果	3	1+2	1+2
类型	int (整型)	str (字符串)	str (字符串)

表 1-3 输入函数比较

1.3 Python 开发环境安装与配置

在开发 Python 程序之前,必须先完成一些准备工作,那就是在计算机上搭建好 Python 的开发环境。

Python 程序既可以直接使用记事本开发,也可以使用 IDE (integrated development environment,集成开发环境)开发。但大多数项目都使用 IDE 开发。因为 IDE 支持代码高 亮、智能提示、可视化等功能,可以大大提升开发效率。本节会介绍其中一个最常用的 Python IDE 工具——Pycharm 的安装和使用。

Python 是跨平台的,可以在很多平台上工作,但目前最常用的三大平台是 Windows、 Mac OS X 和 Linux。Python 最常用工作平台,如表 1-4 所示。 表 1-4 Python 最常用工作平台

操作系统	说 明
Windows	推荐使用 Windows 7 或以上版本。Windows XP 系统不支持安装 Python 3.5 及以上版本
Mac OS X	从 Mac OS X 10.3(Panther)开始,系统已包含 Python
Linux	推荐 Ubuntu 版本,因为最新版的 Ubuntu 会自带 Python 2.x 和 Python 3.x

本书主要介绍 Windows 平台中 Python 的运行环境搭建和程序的开发。

可以直接在 Python 的官网下载各操作系统平台的 Python 安装包及相关文件。

(1) Python 安装包: https://www.python.org/downloads/。

(2) Python 文档: https://www.python.org/doc/。

目前,虽然 Python 3.9 系列及新版本已发布, Python 3.8 系列也在不断发布新版本,但因为 3.8 系列和 3.9 系列仍处于 bugfix (错误修复)中。而 Python 3.6 系列和 Python 3.7 系列却处于 security (安全)中。所以本章以 3.7 系列为例来讲解 Python 环境搭建。喜欢体验新版本的读者,可在官网下载并安装最新版本。

1.3.1 Windows 上安装 Python 开发环境

本节的 Python 开发环境的安装版本是 3.7.9(2020 年 8 月 17 日发布)。以下将介绍在 Windows 平台安装 Python 开发环境的步骤。

(1) 访问地址 https://www.python.org/downloads/, 在菜单中选择 Windows 平台的安装 包, 如图 1-6 所示。



图 1-6 Python 官网首页

(2)选择如图 1-7 所示的 Python 3.7.9 右侧的 Download 命令,进入如图 1-8 所示的下载 列表界面,接着选择自己需要的版本。因为笔者的 Windows 操作系统是 64 位的,所以选择 Windows x86-64 executable installer 进行下载,下载完成后的文件名为 Python-3.7.9-amd64.exe。

第1章 Python 概述

C 🔒 python.	org/downloads/				\$	-
Python version	Maintenance status	First released	End of support	Release schedule		
3.10	bugfix	2021-10-04	2026-10	PEP 619		
3.9	bugfix	2020-10-05	2025-10	PEP 596		
3.8	security	2019-10-14	2024-10	PEP 569		
3.7	security	2018-06-27	2023-06-27	PEP 537		
3.6	security	2016-12-23	2021-12-23	PEP 494		
2.7	end-of-life	2010-07-03	2020-01-01	PEP 373		
ooking for a s	pecific release?					_
ooking for a s ython releases by w Release version	Decific release? ersion number: Release date			Click for more		_
ooking for a s ython releases by v Release version Python 3.8.7	pecific release? ersion number: Release date Dec. 21, 2020		t Download F	Click for more Release Notes	•	_
ooking for a s ython releases by w Release version Python 3.8.7 Python 3.9.1	Decific release? rrsion number: Release date Dec. 21, 2020 Dec. 7, 2020	4	s Download F is Download F	Click for more Release Hotes Release Hotes	•	_
ooking for a s ython releases by v Release version Python 3.8.7 Python 3.9.1 Python 3.9.0	Decific release? rrsion number: Release date Dec. 21, 2020 Dec. 7, 2020 Oct. 5, 2020	d d d	Download F Download F Download F	Click for more Release Notes Release Notes Release Notes	^	_
Ooking for a sy ython releases by w Release version Python 3.8.7 Python 3.9.1 Python 3.9.0 Python 3.8.6	Decific release? rrsion number: Release date Dec. 21, 2020 Dec. 7, 2020 Oct. 5, 2020 Sept. 24, 2020	4	6 Download F 6 Download F 6 Download F 6 Download F 6 Download F	Click for more Release Notes Release Notes Release Notes Release Notes		_
ooking for a sy ython releases by w Release version Python 3.8.7 Python 3.9.1 Python 3.9.0 Python 3.8.6 Python 3.5.10	Decific release? rrsion number: Release date Dec. 21, 2020 Dec. 7, 2020 Oct. 5, 2020 Sept. 24, 2020 Sept. 5, 2020	4 4 4 4 4	6 Download 6 6 Download 7 7 Download 7 7 Download 7 8 Download 7 8 Download 8	Click for more Release Notes Release Notes Release Notes Release Notes Release Notes		_
cooking for a sp ython releases by w Release version Python 3.8.7 Python 3.9.1 Python 3.9.0 Python 3.8.6 Python 3.5.10 Python 3.7.9	Decific release? rrsion number: Release date Dec. 21, 2020 Dec. 7, 2020 Oct. 5, 2020 Sept. 24, 2020 Sept. 5, 2020 Aug. 17, 2020	4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	Download F Download F Download F Download F Download F Download F	Click for more Release Notes Release Notes Release Notes Release Notes Release Notes Release Notes	•	_

图 1-7 选择 Python 版本

ull Changelog	icasc/pytion 5757				A
iles					
Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		bcd9f22cf531efc6f06ca6b9b2919bd4	23277790	SIG
XZ compressed source tarball	Source release		389d3ed26b4d97c741d9e5423da1f43b	17389636	SIG
macOS 64-bit installer	macOS	for OS X 10.9 and later	4b544fc0ac8c3cffdb67dede23ddb79e	29305353	SIG
Windows help file	Windows		1094c8d9438ad1adc263ca57ceb3b927	8186795	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	60f77740b30030b22699dbd14883a4a3	7502379	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	7083fed513c3c9a4ea655211df9ade27	26940592	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	da0b17ae84d6579f8df3eb24927fd825	1348904	SIG
Windows x86 embeddable zip file	Windows		97c6558d479dc53bf448580b66ad7c1e	6659999	SIG
Windows x86 executable installer	Windows		1e6d31c98c68c723541f0821b3c15d52	25875560	SIG
Mindows x86 web bacad installor	Windows		22f68f09e533c4940fc006e035f08aa2	1319904	SIG

图 1-8 Python 下载列表

【小贴士】

- □ X86 和 X86-64 的区别:操作系统是 32bit (位)的,还是 64bit (位)的。
- □ web-based、executable、embeddable zipfile 区别。
 - ▶ web-based:通过网络安装。即执行安装后才通过网络下载 Python。
 - ▶ executable:可执行文件。即把 Python 全部下载到本机后再安装。
 - > embeddable zipfile: zip 压缩文件。即把 Python 打包成 zip 压缩包。

(3) 双击 Python-3.7.9-amd64.exe 文档,进入 Python 安装向导对话框,如图 1-9 所示。 选择 Customize installation,即自定义安装方式,就可以自行选择软件的安装路径。还要选 中 Add Python 3.7 to PATH 复选框,在安装过程中会自动添加 Python 环境变量,否则需要 手动配置环境变量。

(4)选择自定义安装后,界面如图 1-10 所示。单击 Next 按钮,进入如图 1-11 所示界面。选中复选框中的前五项或采用默认设置,并将安装路径改为 D:\Python\Python379(读者可自行设置路径)。单击 Install 按钮。

Python 3.7.9 (64-b	bit) Setup	Python 3.7.9 (64-bit) 9	Setup
	Install Python 3.7.9 (64-bit) Select Instal Now to install Python with default settings, or choose Customize to enable or disable features.		Optional Features
9	 Install Now C:\Users\jj\AppData\Local\Programs\Python\Python37 Includes IDLE, pip and documentation Croster charterity and file secolations 	2	pip Installs pip, which can download and install other Python packages. ft/tk and JDLE Installs tkinter and the IDLE development environment. ft Python test suite
1	 Customize installation Choose location and features 		Installs the standard library test suite. y py launcher of for all users (requires elevation) Installs the global 'py' launcher to make it easier to start Python.
python windows	✓ Install Jauncher for all users (recommended) ✓ Add Eython 3.7 to PATH Cancel	python windows	Bark Next Cancel

图 1-9 Python 安装向导

图 1-10 安装选项界面

(5) Python 的安装速度很快,安装成功后的界面如图 1-12 所示。然后,即可单击 Close 按钮,关闭安装对话框。





图 1-12 安装完成界面

(6) 验证 Python 是否安装成功。打开"命令提示符"窗口,输入 Python,再按 Enter 键,如果显示内容如图 1-13 所示,则说明本机已成功安装 Python 3.7.9 版本。

西 管理员: C:\Windows\system32\cmd.exe - python	X)
Microsoft Windows [版本 6.1.7601] 版权所有 (c) 2009 Microsoft Corporation。保留所有权利。	
C:\Users\jj>python Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1 (AMD64)] on win32	900 64 bit
Type "help", "copyright", "credits" or "license" for more information >>>	
	v

图 1-13 Python 安装成功界面

• 12 •

【小贴士】在安装的第一个界面,如果未选中 Add Python 3.7 to PATH 复选框,安装完成后, 在"命令提示符"窗口输入 python 命令后,会显示:"'python'不是内部或外部 命令,也不是可运行的程序或批处理文件。"需要手动配置 PATH 环境变量。

1.3.2 配置 PATH 环境变量

安装好 Python 后,在"命令提示符"窗口输入 python 命令时,会显示:"'python'不 是内部或外部命令,也不是可运行的程序或批处理文件。"这是因为在安装向导界面,没 有选中 Add Python 3.7 to PATH 复选框。需要手动配置 PATH 环境变量。建议将 Python 安 装目录添加到 PATH 环境变量中。

Windows 操作系统配置 PATH 环境变量的步骤如下。

(1) 打开"控制面板"界面,单击"系统"按钮,再单击"系统"选项卡左上方的"高级系统设置"按钮,打开"系统属性"对话框,如图 1-14 所示。

(2)单击"系统属性"对话框中的"环境变量"按钮,弹出"环境变量"对话框。选中"系统变量"栏中的 PATH 变量,再单击"编辑"按钮,如图 1-15 所示。

沇/垂1生			-	1.00	
计算机名 硬件	高级 系	统保护	程		
要进行大多数更正	收,您必须作	为管理员	登录。		
111ffi 视觉效果,处理	器计划,内	存使用,じ	【及虚拟	勺存	
					设置(<u>s</u>)
用户配置文件					
与您登录有关的]桌面设置				
					设置(匹)
启动和故障恢复					
系统启动、系统	失败和调试	信息			
					设置(<u>I</u>)
			(环境	变量(11)
		确定		取消	○ 肉用(4

图 1-14 "系统属性"对话框

入里	值
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp
	新建(N) 编辑(E) 删除(D)
统变量(<u>S</u>)	
统变量(<u>s</u>) 变量	<u>1. 选中PATH变量</u> 值
统变量(<u>S</u>) 变量 NUMBER_OF_PR	<u>1.选中PATH变量</u> ₄
統変量(<u>S</u>) 変量 NUMBER_OF_PR OS	值 1. 选中PATH变量 4 Vindows NT ↓
統变量(<u>S</u>) 变量 NUMBER_OF_PR OS Path	1.选中PATH变量 4 ¥indows NT D:\Python\Python379\Scripts\;D:
統変量(<u>S</u>) 变量 NUMBER_OF_PR OS Path PATHEXT	值 值 4 Vindows NT D:\Python\Python379\Scripts\;D: .COM;.EXE;.EAT;.CMD;.VES;.VEE;

图 1-15 "环境变量"对话框

(3) 在弹出的"编辑系统变量"对话框中,在原变量值的最前端添加 D:\Python\Python379;D:\Python\Python379\Scripts;变量值(注意:最后的";"不能少,它是用于分割不同的变量值的;再有,D盘为笔者安装 Python的路径,读者须按自己的实际情况进行修改),再单击"确定"按钮完成环境变量的设置,如图 1-16 所示。

编辑系统变量	X
添加变量值D:\Pyth	on\Python379;D:\Python\Python379\Scripts;
变量名(<u>N</u>):	Path
变量值(型):	D:\Python\Python379\Scripts\;D:\Pyth

图 1-16 "编辑系统变量"对话框

【小贴士】"编辑系统变量"对话框中的"变量值"中不能删除系统变量 PATH 中的原有 变量值;而且添加的分号为英文半角符号,否则会出错。

(4)设置完成后,再打开"命令提示符"窗口,输入 python 命令,就可以进入 Python 解释器了。

1.3.3 编写第一个 Python 程序

作为一名初学 Python 的程序开发人员,能输出自己的第一个 Python 程序一定非常开心, 并且,对 Python 开发也会更有兴趣!下面通过两种方法来实现第一个 Python 程序的输出。

1. 在"命令提示符"窗口中启动 Python 解释器实现

在"命令提示符"窗口中输出"Hello,world!人生苦短,我用 Python!",步骤如下。 (1)打开"命令提示符"窗口。选择操作系统的"开始"命令,在"搜索程序和文件" 文本框中输入 cmd 命令,最后按 Enter 键。或者选择操作系统的"开始"→"所有程序"→ "附件"→"命令提示符"命令。

- (2) 在命令提示符后输入 python,并按 Enter 键,进入 Python 解释器。
- (3) 在 Python 的提示符 ">>>" 后输入以下代码,并按 Enter 键。

print("Hello,world!人生苦短,我用 Python! ")

【小贴士】在上面的代码中,小括号和双引号都要在英文半角状态下输入,且 print 所有字母都是小写,因为 Python 中区分大小写字母。

运行结果,如图 1-17 所示。



图 1-17 在"命令提示符"窗口输出第一个程序

2. 在 Python 自带的 IDLE 中实现

在"命令提示符"窗口的 Python 解释器中,编写的代码是纯色的,不方便阅读。在安装 Python 时,自动安装了一个开发工具 IDLE。通过 IDLE 编写的 Python 代码,会用不同的颜色显示,更容易阅读。

下面用 IDLE 工具,完成"Hello world!人生苦短,我用 Python!"的输出,步骤如下。

(1) 打开 IDLE 窗口。选择操作系统的"开始"→"所有程序"→Python 3.7→IDLE
 (Python 3.7 64-bit)命令。

(2) 在 Python 的提示符 ">>>" 后输入以下代码,并按 Enter 键。

print("Hello world! 人生苦短, 我用 Python! ")

运行结果,如图 1-18 所示。



图 1-18 IDLE 窗口

1.3.4 PyCharm 安装和使用

除了 Python 自带的 IDLE 外,还有很多能做 Python 编程的开发工具。本小节对第三方 开发工具 PyCharm 进行简要介绍。

PyCharm 是由 JetBrains 公司开发的一款 Python 程序开发工具。它具有调试、语法高亮、 Project (项目)管理、代码跳转、智能提示、自动完成、单元测试、版本控制等功能。另 外, PyCharm 还支持在 Django (Python 的 Web 开发框架)框架下的 Web 开发。

PyCharm 有两个版本:专业版(Professional)和社区版(Community)。其中,专业版免费试用,可用于科学和 Web Python 开发,支持 HTML、JavaScript 和 SQL。而社区版免费、开源,只用于 Python 开发。

官网可下载 PyCharm 安装文件,地址为 https://www.jetbrains.com/pycharm。

打开 PyCharm 官网,单击右上方的 Download 按钮,进入 PyCharm 下载页面,选择好版本后按 Download 按钮,下载 PyCharm,如图 1-19 所示。



图 1-19 PyCharm 下载页面

读者可根据自己计算机的平台类型,按需要下载 PyCharm 的版本。目前,最新版本是 2021.1.1 (2021 年 4 月 22 日发布)。

对于本教材编写的案例,免费、开源的社区版已足够使用。所以,本书以 Windows 操 作系统下的 pycharm-community-2019.2.1 为例来讲解 PyCharm 的安装、配置和使用。

1. 安装 PyCharm

(1)下载完 PyCharm 后,双击 pycharm-community-2019.2.1.exe 文档,进入 PyCharm 安装界面,然后单击 Next 按钮,如图 1-20 所示。

(2) 进入选择 PyCharm 安装路径界面,修改安装路径或选择默认安装路径,单击 Next 按钮,如图 1-21 所示。



图 1-20 PyCharm 安装界面

图 1-21 选择 PyCharm 安装路径界面

(3) 进入文件配置界面,选中如图 1-22 所示的复选框或根据自己喜好勾选需要的选项,然后单击 Next 按钮。

R PyCharm Community Edition Setup				
PC	Configure your PyCharm Community Edition installation			
Create Desktop Shortcut 64-bit launcher Update context menu Add "Open Folder as F Create Associations V:pp	Update PATH variable (restart needed) Image: Construction of the path o			
	< Back Next > Cancel			

图 1-22 文件配置界面

- (4) 进入启动菜单界面,单击 Install 按钮。
- (5) 进入安装 PyCharm 界面。
- (6) 安装完成后,进入完成安装界面,选择 Reboot now (现在重启) 命令或 I want to

manually reboot later(我稍后手动重启)命令,最后单击 Finish 按钮。

【小贴士】大家如果想把 PyCharm 转为中文界面,可在网上搜索汉化包。PyCharm 的汉化 过程很简单。

2. 使用 PyCharm

(1)运行 PyCharm 程序(选择"开始"→"所有程序"→JetBrains→JetBrains PyCharm Community Edition 2019.2.1 命令),第一次运行 PyCharm,会显示欢迎界面,单击 Create New Project 按钮即可建立 Python 项目界面。

(2) 在 Create Project 对话框中,项目名字直接在 Location 文本框中输入。如果要选择不同的 Python 运行环境,可以单击 Project Interpreter: Python 3.7 右三角按钮,此时, Create Project 对话框下方就会显示 Python 运行环境选择界面。如果已经配置好了 PyCharm 中的 Python 运行环境,可直接从 Interpreter 列表中选择。如果还没有配置 PyCharm,则需 要单击 Interpreter 下拉列表框右侧的按钮,如图 1-23 所示。



图 1-23 Create Project 对话框

(3) 弹出 Add Python Interpreter 对话框, 按如图 1-24 所示的步骤进行操作。

Add Python Interpreter	1 minute 2.7	X
Virtualenv Environment	Interpreter: <no interpreter=""></no>	~
Conet Ingronment	Select Python Interpreter ★ Image: Big Start Control Hide path C ± > D > D > D > D > D > D	第2步
第3步,选者	¥Anaconda或其它Python解释器 Dreg and drep a file into the space space to paths locate it in the tree	第5步
		Cancel

图 1-24 添加&选择 Python 解释器

(4)回到如图 1-23 所示的 Create Project 对话框,选择好刚才指定的 Python 运行环境, 单击 Create 按钮,即可创建 Python 项目。

(5) 创建好项目后,新建 Python 文件,步骤如图 1-25 所示。



图 1-25 在项目中新建 Python 文件

(6) 在弹出的新建窗口为新建的 Python 文件命名,如 hello_world,文件类型已默认为 py,按 Enter 键。

(7) 在新建好的 Python 文件中输入以下代码,如图 1-26 所示。

```
print("Hello,world! ")
print("I am a student! ")
```

1	🖬 untitled2 [C:\Users\jj\PycharmProjects\untitled2]\hello_world.py - PyCharm (Administrator)							
<u>F</u> ile	<u>Eile Edit View Navigate Code Refactor Run Iools VCS Window H</u> elp							
	untitled2 > 📸 hello_world.py		i hello_world V 🕨 🛔 🔳 🔍					
lect	🔲 Project 🔻 😌 😤 🗢	🖟 hello_w	orld.py ×					
Pro	untitled2 C:\Users\jj\PycharmProjects\untitled2	1	<pre>print("Hello,world!")</pre>					
-	🐉 hello_world.py	2	print("I am a student!")					
-	V IIII External Libraries	3						
	Python 3.7 > C:\ProgramData\Anaconda3\p	4						
	- Scratches and consoles	5						
		6						
		7						
		8	-					
		9						
Ictur		10						
Str		11						
2		12						
		13						
nites		14						
Favo		15						
ii t		10						
	≣ <u>6</u> : TODO I Terminal	- D-+	Event Log					
	Updating skeletons for C:\Program	nDat	3:1 CKLF UTF-8 4 spaces Python 3.7 1					

图 1-26 第一个 Python 程序

(8) 右击 hello_world.py, 在弹出的快捷菜单中选择 Run'hello_world'命令,运行结果 如图 1-27 所示。

第1章 Python 概述

🖬 untitled2 [C:\Users\jj\PycharmProjects\untitled2]\hello	world.py - PyCharm (Administrator)					
Eile Edit View Navigate Code Refactor Run Iools VCS Window Help						
untitled2 > 🛃 hello_world.py	ello_world ∨ ▶ ♣ ■ Q					
및 Project ▼ ③ 😤 \$\$\$ -	🛃 hello_world.py ×					
Y IIII Untitled2 (AlserAji/PycharmProjects\untitled2	<pre>1 print("Hello,world ! ") 2 print("I am a student ! ") 3 4 5 6 7 8 9 10</pre>					
Run: hello_world ×	\$ −					
Image: A state of the stat						
▶ <u>4</u> ; Run <u>6</u> ; TODO 🖾 Terminal 🔮 Python Console	Event Log 3:1 CRLF UTF-8 4 spaces Python 3.7 🍙 👼					

图 1-27 程序运行结果

1.4 Python 编程规范

"不以规矩,不能成方圆"。任何一种语言都有一些约定俗成的编码规范。当然, Python 也不例外。Python 非常重视代码的可读性,对代码布局和排版有更加严格的要求。本节主 要介绍 Python 代码编写的一些共同的要求、规范和常用代码的优化建议,最好从编写第一 段代码开始就养成遵循规范的好习惯。

1.4.1 语句

Python 中,通常每个语句应该独占一行。但如果语句过长,可以使用反斜杠(\)来实现多行语句,代码如下。

```
cj_sum = cj_yu+cj_yw+cj_sx+\
cj wl
```

但是,在包含[]、{}或()的语句需分多行时,不需要使用反斜杠(\),代码如下。

```
cj_sum = [cj_yu,cj_yw,cj_sx,
cj_wl]
```

1.4.2 注释

为程序添加注释可以用来解释程序某些部分的作用和功能,提高程序的可读性。除此 之外,注释也是调试程序的重要方式。如果不希望执行程序中的某些代码,就可以将这些 代码注释掉。Python 解释器会忽略所有的注释语句,即注释语句不影响程序的执行。当然, 添加注释的目的还是提高程序的可读性!

Python 允许在任何地方插入空字符或注释,但不能在标识符和字符串中间插入注释。

Python 中的注释有单行注释和多行注释两种。

单行注释以"#"开始,跟在"#"后面直到这行结束为止的代码都将被解释器忽略。 多行注释可以一次性注释多行代码,使用 3 个单引号(")或 3 个双引号开头(""")将注 释的内容括起来。

程序 1-1: 注释使用。

```
#这是我的第一个 Python 程序
"""
这是我的第一个 Python 程序
请多多指教!
"""
print("Hello,world!")
print("China!")
'''
这是我的第一个 Python 程序
请多多指教!
'''
```

运行结果:

Hello,world! China!

1.4.3 缩进

在 Python 中,对于类定义、函数定义、流程控制语句、异常处理语句等,行尾的冒号 和下一行的缩进,表示的是下一个代码块的开始,而缩进的结束则表示此代码块的结束。 Python 对代码的缩进要求非常严格,同一个级别代码块的缩进量必须一样。

程序 1-2: 缩进举例。

对于 Python 缩进规则,初学者可以这样理解: Python 要求属于同一作用域中的各行代码的缩进量必须一致,但具体缩进量为多少,并不做硬性规定。

【小贴士】Python 中代码的缩进,可以使用空格或者 Tab 键来实现。但无论是手动敲空格, 还是使用 Tab 键,通常情况下都是采用 4 个空格长度作为一个缩进量(默认情 况下,一个 Tab 键就表示 4 个空格)。 【小贴士】为了增加 Python 中代码的可读性,可以适当使用空行和空格。通常在顶级定义 (如函数或类的定义)之间空两行,而方法定义之间空一行。另外,在用于分 隔某些功能的位置也可以空一行。而在运算符两侧、函数参数之间以及逗号两 侧,都建议使用空格进行分隔。

1.5 扩展库安装方法

随着 Python 的广泛使用, Python 的扩展库(也称为模块,或扩展包)也越来越多,如 果能够熟练使用各种扩展库,必将提升工作效率。

安装扩展库之前,需要先安装好 Python 环境。

安装 Python 扩展库主要有两种方法。第一种方法:使用 pip 命令行工具在线下载需要的第三方库。第二种方法:手动下载第三方库,再使用 pip 命令安装。

Python 扩展库(第三方库)下载地址: https://pypi.python.org/pypi。

【小贴士】PyPI(Python Package Index)包索引是 Python 官方的第三方库的仓库。

1.5.1 pip 命令安装

pip 是 Python 包管理工具,该工具提供了对 Python 包的查找、下载、安装、卸载等功能。Python 2.7.9 +或 Python 3.4+以上版本都自带 pip 工具,无须再安装 pip。如果没有安装 pip,可以去官网下载,地址为 https://pypi.org/project/pip/。

一般情况, pip 对应的是 Python 2.7, pip3 对应的是 Python 3.x。本教材安装的是 Python 3.7.9, 所以本节讲解 Python 3.x 中的 pip 工具。

在"命令提示符"窗口中输入 pip3,可显示 pip3 的用法包括命令及选项,如图 1-28 所示。读者可根据自己的需要,进一步学习 pip3 命令。

🔤 管理员: C:\Windows\system32\cmd.exe				
Microsoft Windows [版本 6.1.7 版权所有 (c) 2009 Microsoft C	601] ▲ A Comporation。保留所有权利。			
C:\Users\jj>pip3				
Usage: pip3 <command/> [options]				
Commands:				
install	Install packages.			
download	Download packages.			
uninstall	Uninstall packages.			
freeze	Output installed packages in requirements format.			
list	List installed packages.			
show	Show information about installed packages.			
check	Verify installed packages have compatible dependen			
cies.				
config	Manage local and global configuration.			
search	Search PyPI for packages.			
cache	Inspect and manage pip's wheel cache.			
wheel	Build wheels from your requirements.			
nasn 	Compute nasnes of package archives.			
completion	H helper command used for command completion.			
debug	Show information useful for debugging.			
netb	show help for commands.			
General Options:				
-h,help	Show help.			

图 1-28 pip3 命令

在"命令提示符"窗口中输入 pip3 list 或 pip list,可以查看当前已安装的 Python 扩展库。

接着,以 Django 库为例,讲解如何使用 pip3 命令安装最新版本扩展库、安装指定版本扩展库、查看当前安装的扩展库的版本、卸载扩展库(在"命令提示符"窗口完成)。

1. 安装最新版本的 Django 扩展库

C:\Users\jj> pip3 install django

【小贴士】如果安装的 Python 版本跟最新的 Django 版本不是对应关系,会显示错误。

2. 安装指定版本的 Django 扩展库

C:\Users\jj> pip3 install django==3.0.11

3. 查看当前安装的 Django 库的版本

C:\Users\jj> pip3 show django

4. 卸载 Django 库

C:\Users\jj> pip3 uninstall django

【小贴士】Django 是 Python 编程语言驱动的一个开源模型-视图-控制器(MVC)风格的 Web 应用程序框架。使用 Django, 在几分钟之内就可以创建高品质、易维护、 数据库驱动的应用程序。Django 的下载地址: https://www.djangoproject.com/ download/。

1.5.2 手动下载第三方库,再使用 pip 命令安装

这种方法既然要通过 pip 命令安装, 就需要在 pip 环境下才能操作。

1. tar.gz 文件安装

pip 非常方便,但是并不是所有的扩展库都能用 pip 来安装,有的扩展库可能只提供了 源码的压缩包文件,或者要求安装环境不能连接外网,这时就可以直接使用 tar.gz 文件进 行安装。

下载 tar.gz 文件的步骤,如图 1-29 所示。

- (1) 打开 Python 扩展库官网地址,在搜索栏中输入 tar.gz,按 Enter 键。
- (2) 在扩展库页面中单击想要的库文件。
- (3)页面中单击 Download files 按钮,再单击 tar.gz 文件,下载。

(4) 在本地解压下载的 tar.gz 文件,在"命令提示符"窗口,进入 setup.py 文件所在 的目录,执行下列命令即可:

python setup.py install

【小贴士】如果输入的命令为 python3 setup.py install, 会显示 "'python3'不是内部或外部命令, 也不是可运行的程序或批处理文件。"



图 1-29 下载 tar.gz 文件

虽然只有一行命令,但对比 pip,这种方式略显烦琐。

2. .whl 文件安装

使用.whl 扩展名的文件是 wheel 文件,专门用于 Python 模块的安装。

.whl 文件的下载方法与 tar.gz 文件类似,只是步骤中搜索时输入改为.whl。下载页面, 如图 1-30 所示。

← → C	d9/#files	≋ ☆ 😝 :
mod9 0.0.1		Latest version Released: Oct 16, 2020
Python wrappers over Mod9 ASR E	ingine TCP Server.	
Navigation	Download files	
Project description Project description	Download the file for your platform. If you're not sure which to choose, lear packages 💆.	n more about <u>installing</u>
Lownload files	Filename, size 下载. Wh1 定件 Python version	Upload date Hashes
	mod9-0.0.1-py3-none-any.whl][88.6 kB) Wheel py3	Oct 16, 2020 View
mod9-0.0.1- ^		全部显示 ×

图 1-30 .whl 文件的下载

下载后,在"命令提示符"窗口,进入 whl 文件所在的目录,执行下列命令即可:

```
pip3 install mod9-0.0.1-py3-none-any.whl
```

综上所述,自行安装 Python 扩展库时,除无法安装外,还是使用 pip 命令安装更加方便、快捷。

本章小结

本章首先简要介绍了 Python 语言的发展历史、特点、应用领域。然后详细介绍了如何 安装、配置和使用 Python 开发环境以及如何运行 Python 程序,包括使用命令行工具和使 用 Python 自带的 IDLE 工具两种实现方法。还介绍了 PyCharm 的安装、配置和使用。最后, 介绍了 Python 语言的编程规范和扩展库的安装方法。

习 题

一、填空题

1. Python 安装扩展库最常用的是____工具。

2. Python 中单行注释以_____开头。

3. Python 中多行注释用______或____。

4. PyCharm 是 JetBrains 公司开发的 Python____。

6. 使用 pip 工具查看当前已安装的 Python 扩展库的完整命令是____。

7. Python 是面向_____的编程语言。

8. Python 官方提供了_____工具,使用该工具可以将大部分 Python 2.x 代码转换为 Python 3.x 代码。

9. Python 3.x 中使用_____(符号)表示不等于运算符。

10. Python 3.x 默认使用的编码是 编码。

二、判断题

1. Python 是一种跨平台、开源、免费的高级动态编程语言。 ()

2. Python 3.x 完全兼容 Python 2.x。()

3. Python 3.x 和 Python 2.x 唯一的区别就是: print 在 Python 2.x 中是输出语句,而在 Python 3.x 中是输出函数。 ()

4. 在 Windows 平台上编写的 Python 程序无法在 UNIX 平台运行。()

5. 不可以在同一台计算机上安装多个 Python 版本。()

6. Python 采用的是基于值的自动内存管理方式。()

7. Python 属于编译型语言。 ()

8. "//"运算符在 Python 2.x 和 Python 3.x 中是一致的,都是 floor 除法。 ()

- **9.** Python 语言可移植性差。()
- 10. Python 3.x 中, print()是一个函数。 ()

三、选择题

print(1+2)的输出是()。
 A. 1+2
 B. 1
 C. 2
 D. 3
 Python 程序源文件的扩展名是()。
 A. Python
 B. pyc
 C. pp
 D. py
 Python 语言属于()。
 A. 机器语言
 B. 汇编语言
 C. 高级语言
 D. 科学计算语言

四、简答题

- 1. 按照程序的转换方式,高级语言分为哪两种?
- 2. 简述 Python 的应用领域(至少3个)。

五、编程题

- 1. 编写第一个 Python 程序,运行该程序并输出 Hello everyone! I am a student.。
- 2. 编写 Python 程序,输出如下图形效果。

* * * * * * * * *