



# Python程序设计

## 第1章 Python概述

授课老师：刘国旭

潍坊科技学院



- 认识Python
- 安装Python解释器
- 常用的Python编辑器
- 开发第一个Python程序
- 模块的安装、导入与使用



# 教师信息



潍坊科技学院  
Weifang University of Science and Technology

- 授课老师：刘国旭

个人主页: <https://pandalgx.github.io>

联系邮箱: [liuguoxu@wfust.edu.cn](mailto:liuguoxu@wfust.edu.cn)

办公地点: 孵化大厦B座4层 农蔬区块链实验室 404



# 课程目标



- 掌握Python语法、数据类型、函数、文件操作、异常、模块、面向对象等内容。
- 通过本课程的学习，掌握Python开发的基础知识，可以独立开发Python简单的项目程序。



# 课程内容



章节	教学内容	章节	教学内容
1	Python 概述	7	文件操作
2	Python 基础	8	面向对象
3	流程控制	9	异常
4	字符串	10	Python 计算生态与常用库
5	组合数据类型	11	项目案例
6	函数		



# 课程学时



- 课时：
  - 计科: 48学时 = 理论 (24学时) + 上机 (24学时)
  - 物联网: 32学时 = 理论 (16学时) + 上机 (16学时)
  - 共16周
- 上课时间：
  - 计科5: 周二 单周: 12节, 双周12、34节
  - 计科6: 周四 单周: 12节, 双周12、34节
  - 物联网: 周二56节
- 理论与上机课交替进行, **上机地点在同一教室, 自带电脑**



# 课程考核



- 考核方式：

- 平日表现 (出勤、课堂作业) ----- 20分
- 作业 (编程题)----- 30分
- 期末考试 ----- 50分
- 总计 ----- 100分

- 按照学校规定 “考勤+课堂作业 (课后练习)” 累计缺席1/3将取消本学期本门课程的成绩。





# 课程主页



潍坊科技学院  
Weifang University of Science and Technology

- **网址:** <https://vlg-wfust.github.io/course/python/index.html>

## 课程简介

《Python程序设计》是面向计算机相关专业的一门专业基础课，涉及Python语法、数据类型、函数、文件操作、异常、模块、面向对象等内容。通过本课程的学习，学生能够掌

## 课程进度计划表

Event Type	Date	Description	Course Materials
Lecture 1	Week1	第1讲 Python概述	[Slide]
Programming	Week2 Section1	Python环境安装和配置	
Lecture 2	Week2 Section2	第2讲 Python基础	
Programming	Week3	Python基础练习	
Lecture 3	Week4 Section1	第3讲 流程控制	
Programming	Week4 Section2	流程控制练习	





# 课程主页

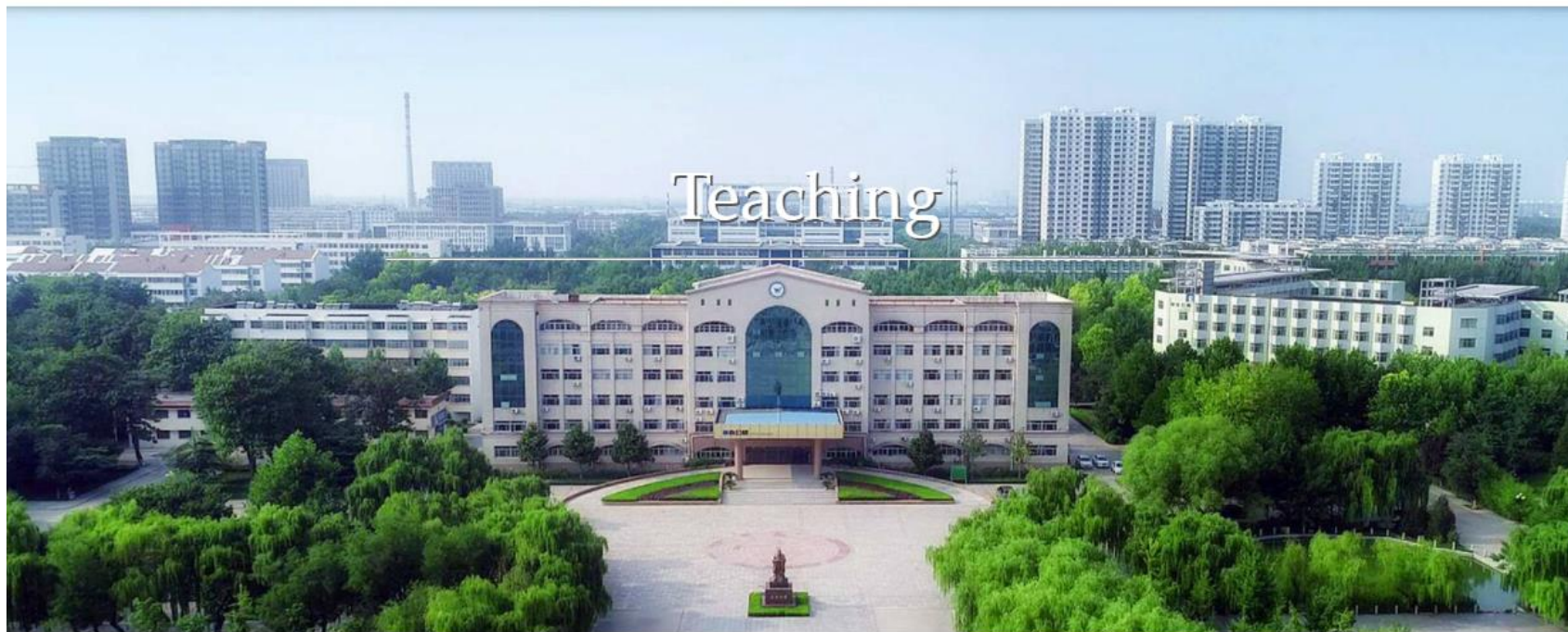


潍坊科技学院  
Weifang University of Science and Technology

<https://vlg-wfust.github.io/teaching.html>

Vision and Learning Group

Home People Research Publications Teaching Contact Us



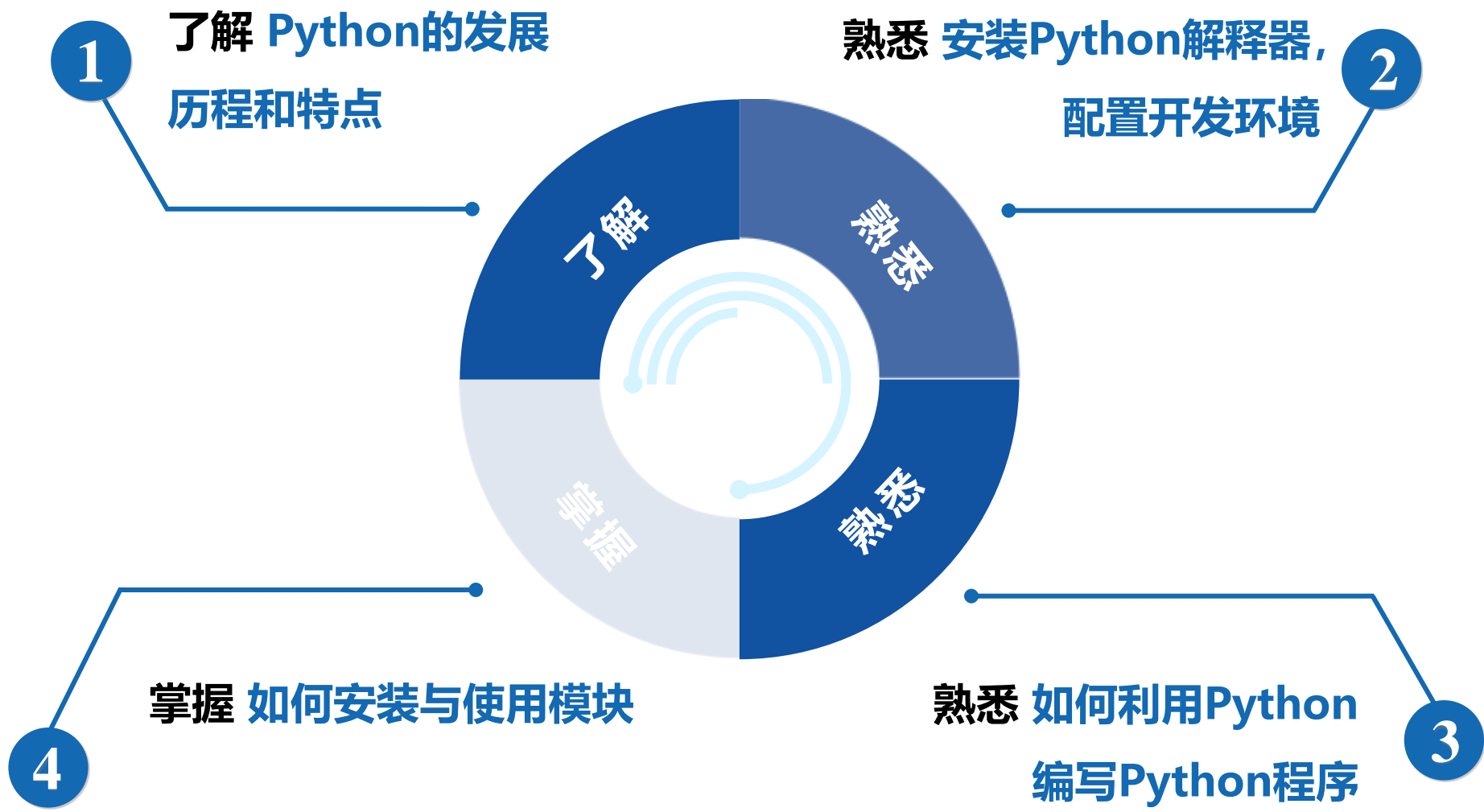
- Autumn 2020 [Digital Image Processing](#)
- Spring 2021 [Python Programming \(Python程序设计\)](#)

Activate Windows





# 学习目标





**01 认识Python**

**02 Python解释器的安装与Python程序的运行**

**03 Python开发工具**

**04 Python模块**



## 01 认识Python

02 Python解释器的安装与Python程序的运行

03 Python开发工具

04 Python模块



## 1.1.1 Python的发展历程



计算机目前已应用在人类日常生活的各个场景，计算机通过程序控制，程序通过编程语言编写，对于初次接触编程的小伙伴而言，Python无疑是最为**简洁、易上手**的编程语言。





## 1.1.1 Python的发展历程



1989年圣诞节期间，阿姆斯特丹的Guido (Guido van Rossum) 自觉假日无趣，想起自己曾参与设计的一种优美与强大并存，但最终惨遭失败的语言ABC，寻思不如开发一个新的脚本解释程序作为ABC语言的继承，于是Python诞生了。



## 1.1.1 Python的发展历程



Python语法很多来自C语言，但又受到ABC语言的强烈影响。自诞生开始，Python已经具有了类（class）、函数（function）、异常处理（exception）、包括列表（list）和字典（dict）在内的核心数据类型，以及以模块为基础的拓展系统。





## 1.1.1 Python的发展历程

Guido于1989年定下目标之后便投身于Python语言的设计之中，但Python的第一个公开版本直到1991年才发行，此版本使用C语言实现，能调用C语言的库文件。

2008年12月，Python 3.0版本发布，并被作为Python语言持续维护的主要系列。

2012年Python 3.3版本发布，2014年Python 3.4版本发布，2015年Python 3.5版本发布，2016年Python 3.6版本发布，2018年6月27日Python 3.7.0发布，2019年10月14日Python 3.8.0发布



2000年10月，Python 2.0发布，Python从基于maillist的开发方式转为完全开源的开发方式。

2010年，Python 2.x系列发布了最后一个版本，其主版本号为2.7，同时，Python的维护者们声称不在2.x系列中继续对主版本号升级，Python 2.x系列慢慢退出历史舞台。

目前Python的最新版本为2021年2月19日发布的3.9.2。





## 1.1.2 Python语言的特点

- **简洁。** Python代码的行数往往只有C、C++、Java代码数量的1/5~1/3。
- **语法优美。** Python语言是高级语言，它的代码接近人类语言，只要掌握由英语单词表示的助记符，就能大致读懂Python代码。
- **简单易学。** Python是一门简单易学的编程语言，它使编程人员更注重解决问题，而非语言本身的语法和结构。
- **开源。** Python是FLOSS（自由/开放源码软件）之一，用户可以自由地下载、拷贝、阅读、修改代码。
- **可移植。** Python语言编写的程序可以不加修改地在任何平台中运行。



## 1.1.2 Python语言的特点

- **扩展性良好。** Python不仅可以引入.py文件，还可以通过接口和库函数调用由其它高级语言（如C语言、C++、Java等）编写的代码。
- **类库丰富。** 世界各地的程序员通过开源社区又贡献了十几万个几乎覆盖各个应用领域的第三方函数库。
- **通用灵活。** Python是一门通用编程语言，可被用于科学计算、数据处理、游戏开发、人工智能、机器学习等各个领域。
- **模式多样。** Python既支持面向对象编程，又支持面向过程编程。
- **良好的中文支持。** Python 3.x解释器采用UTF-8编码表达所有字符信息，编码支持英文、中文、韩文、法文等各类语言。



## 1.1.2 Python语言的特点



### Python 语言的缺点

- ❑ 执行效率不够高，Python程序的效率只有C语言程序的1/10。
- ❑ Python 3.x和Python 2.x不兼容。



## 01 认识Python

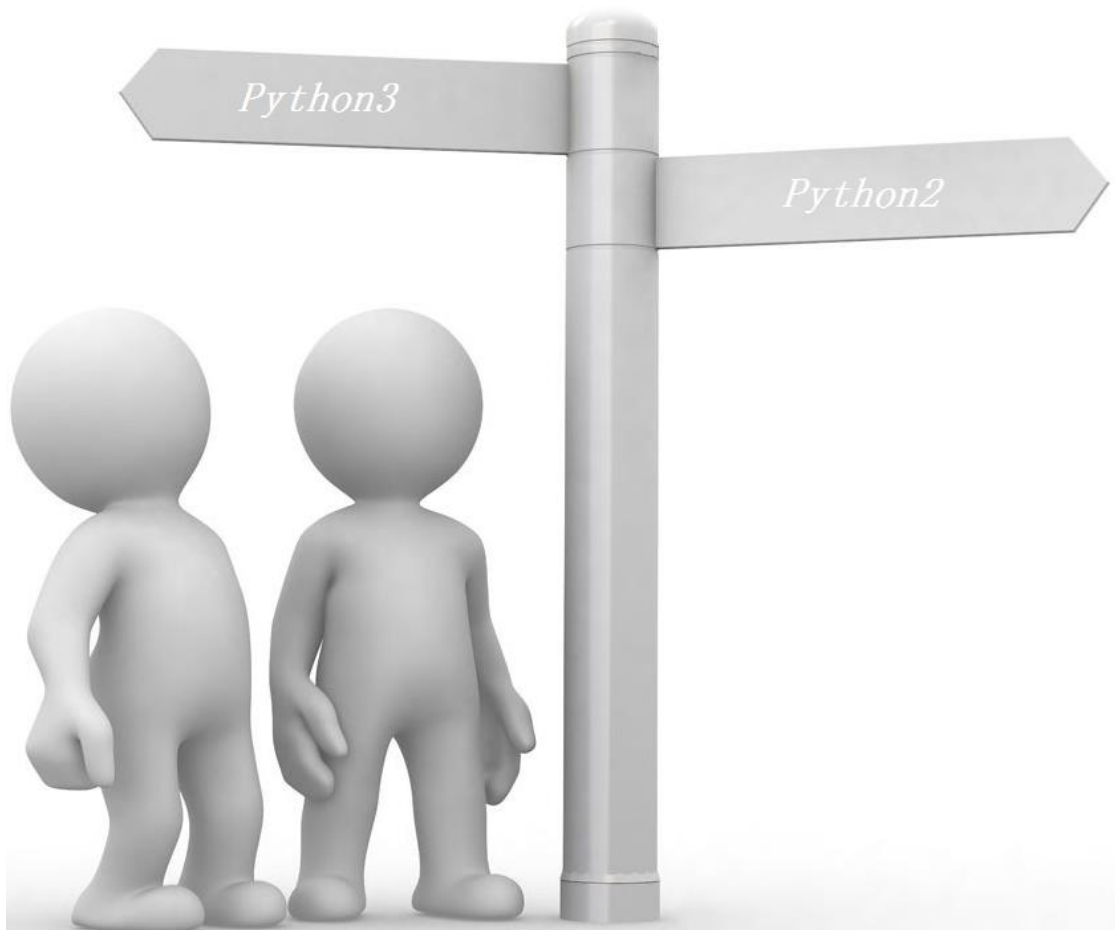
## 02 Python解释器的安装与Python程序的运行

## 03 Python开发工具

## 04 Python模块



## 1.2.1 安装Python解释器



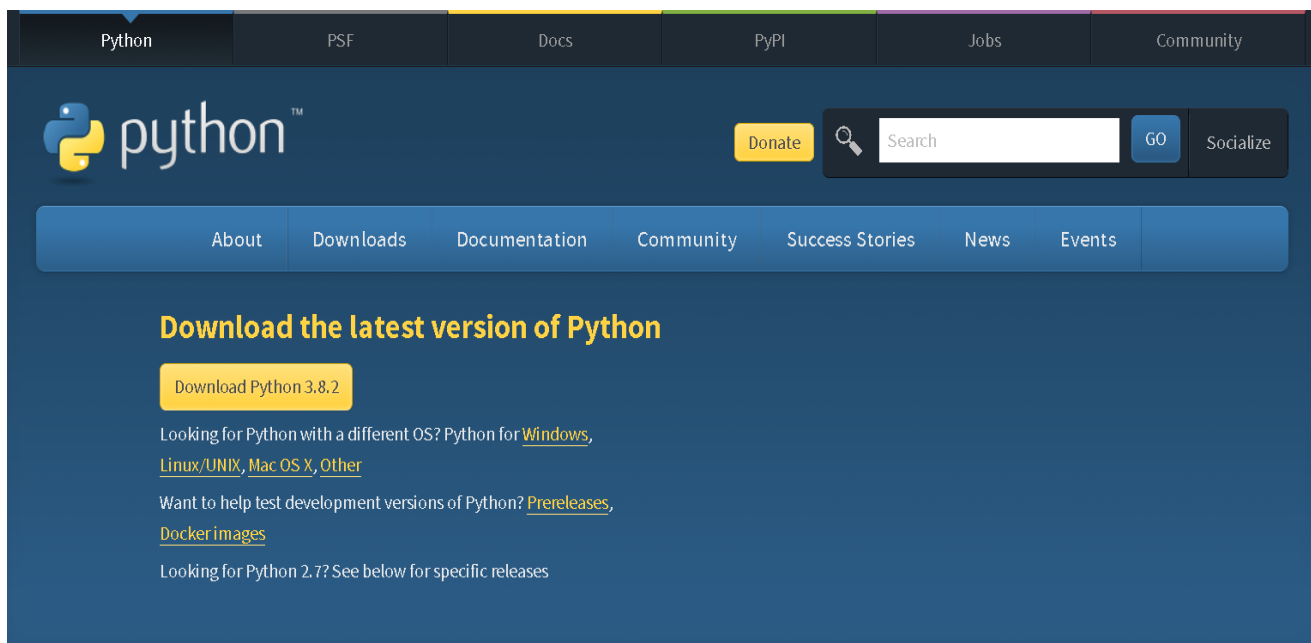
Python解释器有多个版本，考虑到主要的Python标准库更新只针对3.x系列，且当下企业也正从Python 2.x向3.x过渡，因此对于初学Python的读者而言，**Python3.x无疑是明智的选择。**



## 1.2.1 安装Python解释器



1. 访问Python官网的**下载页面**：  
<https://www.python.org/downloads/>。

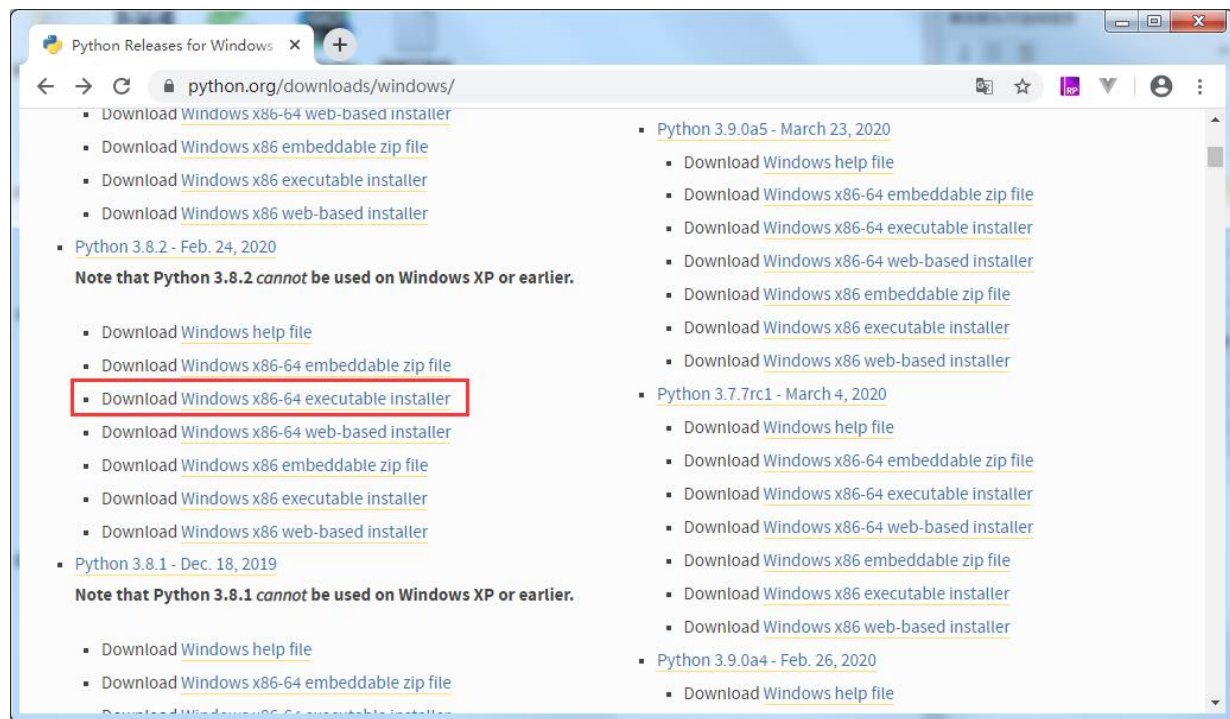




## 1.2.1 安装Python解释器



2. 进入Windows版本软件下载页面，根据操作系统版本选择相应软件包。本教材使用的是Windows 7 64位操作系统，此处选择3.8.2版本、.exe形式的安装包。







## 1.2.1 安装Python解释器



3. 勾选 “Add Python 3.8 to PATH” ，选择 “Install Now” 开始自动安装Python解释器、配置环境变量。片刻后安装完成。





## 1.2.1 安装Python解释器



- 在“开始”菜单栏中搜索“python”，找到并单击打开Python 3.8(64 bit)。

```
Python 3.8 (64-bit)
Python 3.8.2 <tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10> [MSC v.1916 64 bit <AMD64>] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```



## 1.2.1 安装Python解释器

打开控制台窗口，在控制台中执行“python”命令也可以进入Python环境。

```
管理员: Windows 命令处理程序 - python
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Windows\SysWOW64>python
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```



## 1.2.2 Python程序的运行方式



Python程序的运行方式有两种：**交互式**和**文件式**。交互式指Python解释器逐行接收Python代码并即时响应；文件式也称批量式，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。



## 1.2.2 Python程序的运行方式



### 交互式

通过Python解释器或控制台都能用相同的操作以交互方式运行Python程序。以控制台为例，进入Python环境后，在命令提示符“>>>”后输入如下代码：

```
print("hello world")
```

hello world



## 1.2.2 Python程序的运行方式



### 文件式

创建Python文件（后缀为.py的文件），在其中写入Python代码并保存。在该Python文件所在文件夹的空白区域按下“Shift+鼠标右键”，单击选择选项列表中的“在此处打开命令窗口”选项以打开命令窗口，在命令提示符“>”后输入命令“python hello.py”运行Python程序。

```
管理员: C:\Windows\system32\cmd.exe
E:\python_study>python hello.py
hello world
E:\python_study>
```



**01 认识Python**

**02 Python解释器的安装与Python程序的运行**

**03 Python开发工具**

**04 Python模块**





## 1.3.1 常用的开发工具



Python解释器捆绑了Python的官方开发工具——**IDLE**，IDLE具备**集成开发环境**（Integrated Development Environment，简称IDE）的基本功能，但开发人员一般还是会根据自己的需求或喜好选择使用其他的开发工具。



## 1.3.1 常用的开发工具

### 常用的开发工具

#### Sublime Text

Sublime Text是一个编辑器，它功能丰富、支持多种语言、有自己的包管理器，开发者可通过包管理器安装组件、插件和额外的样式，以提升编码体验。Sublime Text在开发者群体中非常受欢迎。

#### Eclipse+PyDev

Eclipse是古老且流行的程序开发工具，支持多种编程语言；PyDev是Eclipse中用于开发Python程序的IDE。Eclipse+PyDev通常被用于创建和开发交互式的Web应用。



## 1.3.1 常用的开发工具

### Vim

Vim是Linux系统中自带的高级文本编辑器，也是Linux程序员广泛使用的编辑器，它具有**代码补全**、**编译及错误跳转**等功能，并支持以插件形式进行扩展，实现更丰富的功能。

### Jupyter Notebook

Jupyter Notebook（简称Jupyter）支持**实时代码**，便于创建和共享文档，它本质上是一个Web应用程序，常被应用于数据分析领域。

### PyCharm

PyCharm具备一般IDE的功能，如**调试**、**语法高亮**、**Project管理**、**代码跳转**、**智能提示**、**单元测试**、**版本控制**等，使用PyCharm可以实现程序编写、运行、测试的一体化。

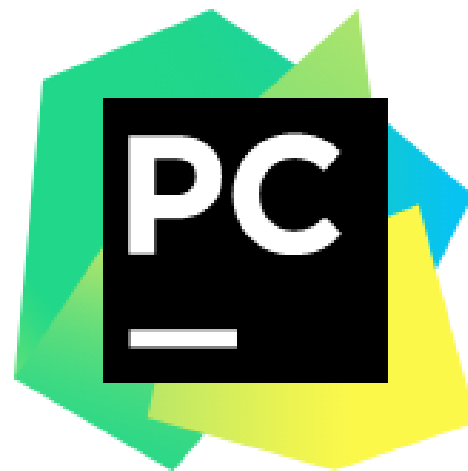


## 1.3.2 Python IDE——PyCharm 的下载与安装



潍坊科技学院  
Weifang University of Science and Technology

**PyCharm**操作简捷、功能齐全，既适用于新手，也可满足开发人员的专业开发需求。





# 1.3.2 Python IDE——PyCharm 的下载与安装

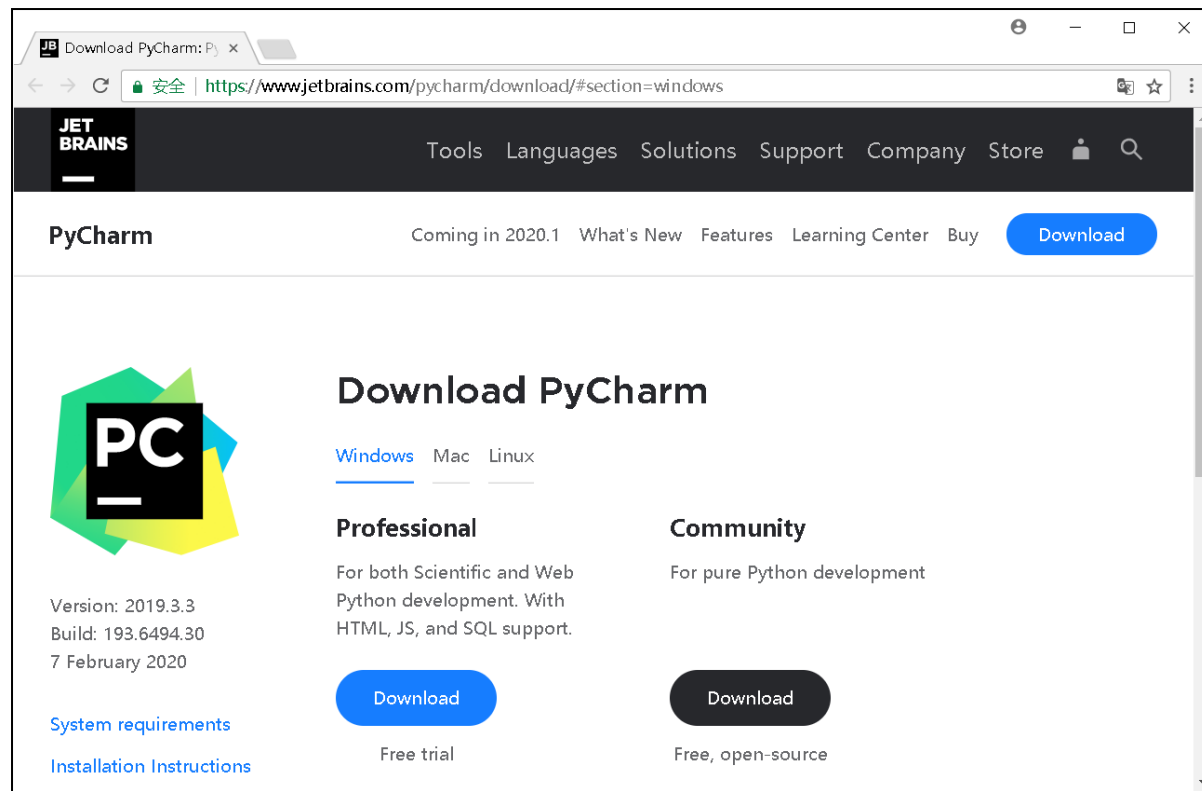


潍坊科技学院  
Weifang University of Science and Technology

## 1. 下载PyCharm

访问PyCharm官方网址

<http://www.jetbrains.com/pycharm/download/>，进入PyCharm的下载页面。





## 1.3.2 Python IDE——PyCharm 的下载与安装



潍坊科技学院  
Weifang University of Science and Technology

### 1. 下载PyCharm

Professional和Community是PyCharm的两个版本。

#### Professional版本特点

- ✓ 提供Python IDE的所有功能，支持Web开发。
- ✓ 支持JavaScript、CoffeeScriptTypeScript、CSS和Cython等。
- ✓ 支持远程开发、Python分析器、数据库和SQL语句。

#### Community版本特点

- ✓ 轻量级的Python IDE，只支持Python开发。
- ✓ 免费、开源、集成Apache2的许可证。
- ✓ 智能编辑器、调试器、支持重构和错误检查，集成VCS版本控制。



## 1.3.2 Python IDE——PyCharm 的下载与安装



潍坊科技学院  
Weifang University of Science and Technology

### 2. 安装PyCharm

(1) 双击下载好的安装包 (pycharm-community-2020.1.1.exe) 打开PyCharm安装向导, 可看到 “Welcome to PyCharm Community Edition Setup” 界面, 如右图所示。







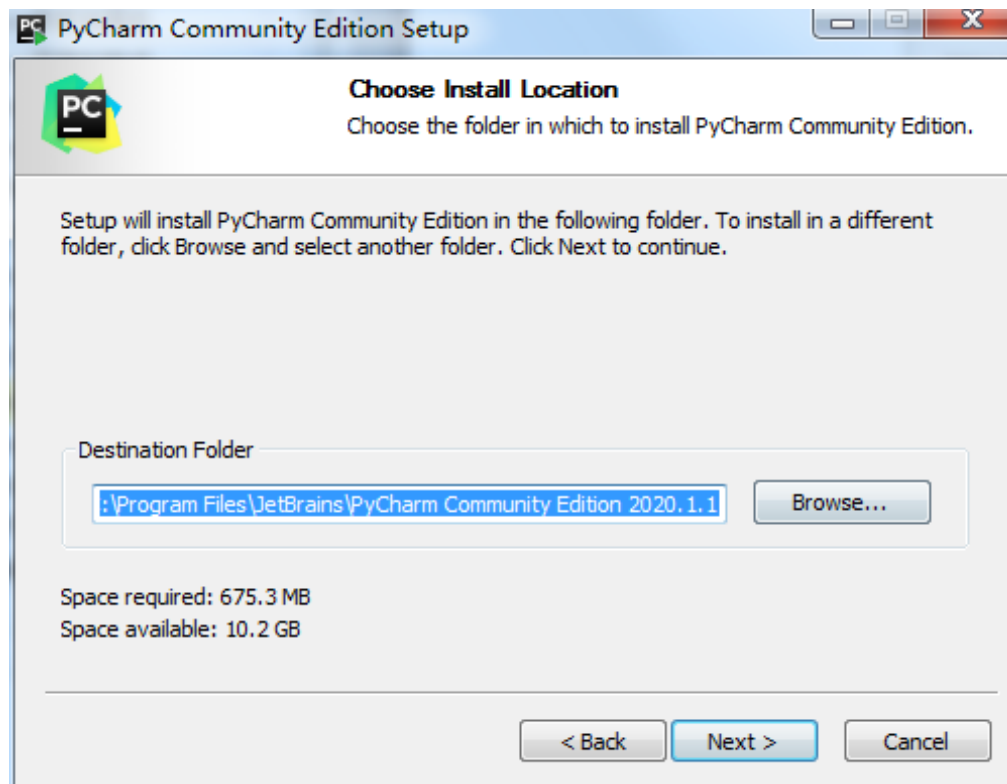
## 1.3.2 Python IDE——PyCharm 的下载与安装



潍坊科技学院  
Weifang University of Science and Technology

### 2. 安装PyCharm

(2) 单击【Next >】按钮进入“Choose Install Location”界面，用户可在此界面设置PyCharm的安装路径。此处使用默认路径，如右图所示。





## 1.3.2 Python IDE——PyCharm 的下载与安装

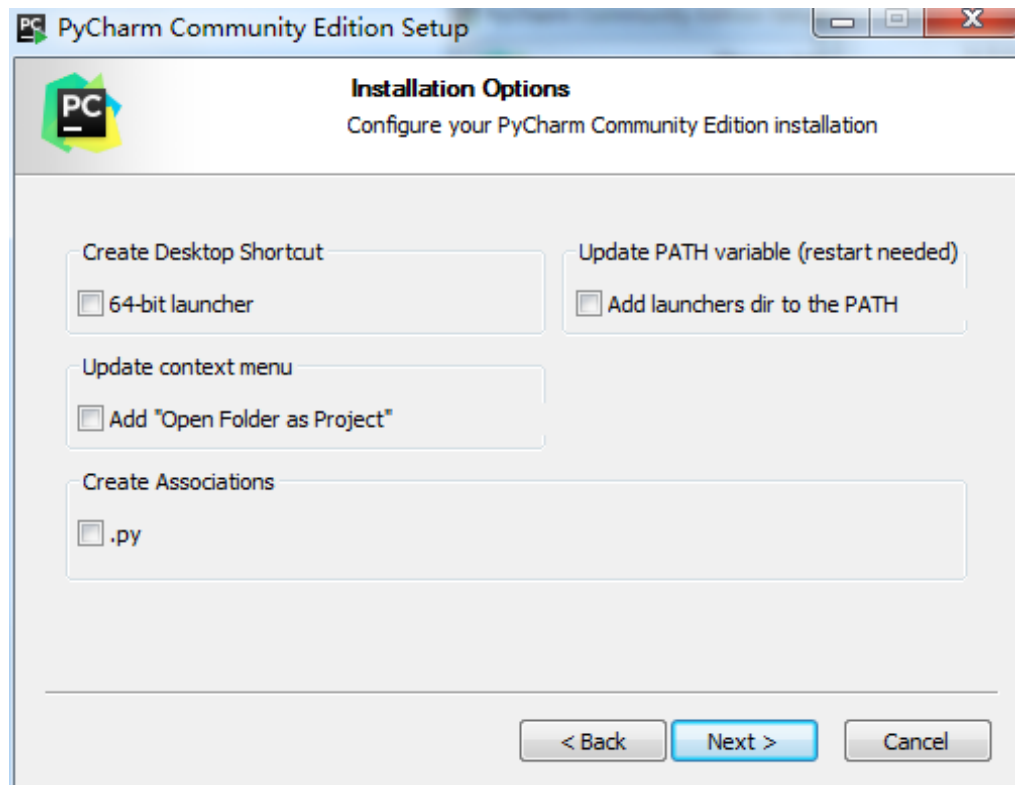


潍坊科技学院  
Weifang University of Science and Technology

## 2. 安装PyCharm

(3) 单击【Next >】按钮进入

“Installation Options” 的界面，在该界面可配置PyCharm的选项，如右图所示。





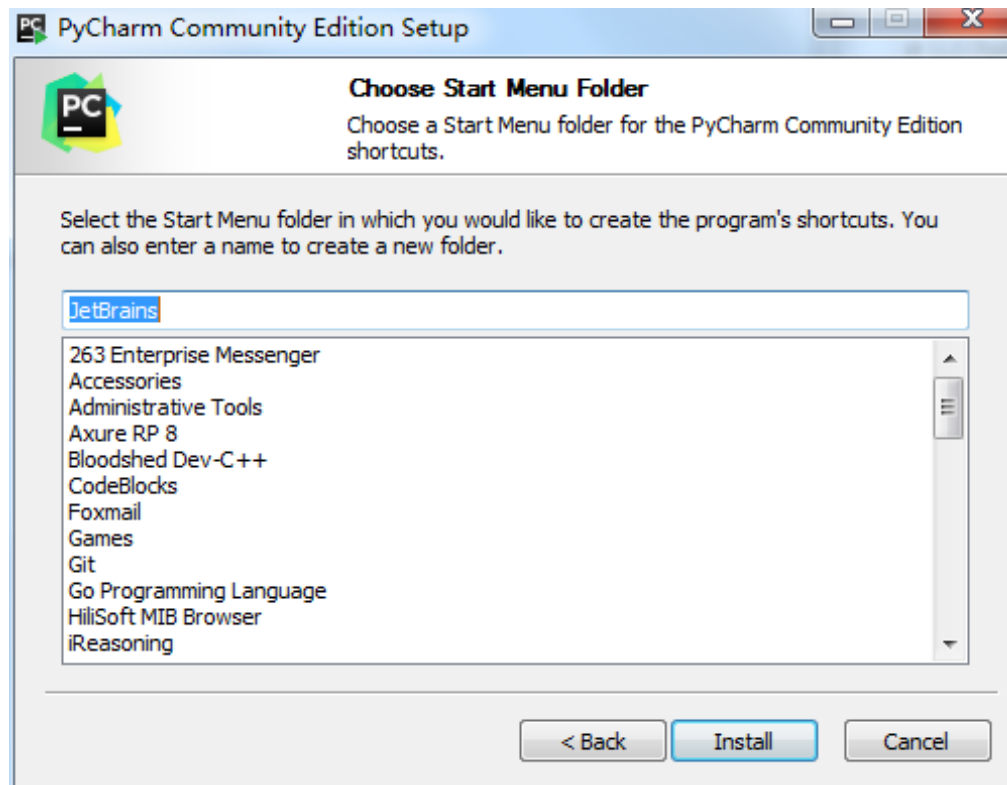
## 1.3.2 Python IDE——PyCharm 的下载与安装



潍坊科技学院  
Weifang University of Science and Technology

### 2. 安装PyCharm

(4) 勾选前图所示界面中的所有选项，单击【Next >】按钮进入“Choose Start Menu Folder”界面，如右图所示。





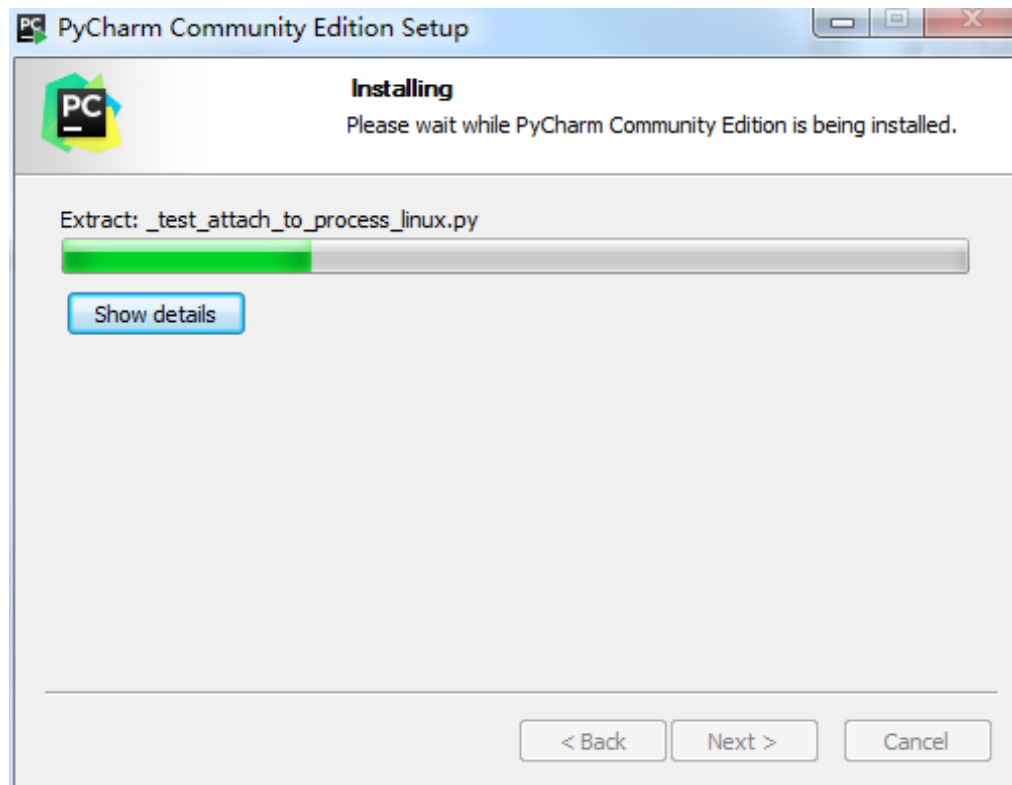
## 1.3.2 Python IDE——PyCharm 的下载与安装



潍坊科技学院  
Weifang University of Science and Technology

## 2. 安装PyCharm

(5) 单击上图中的【Install】按钮**安装PyCharm**，如右图所示。





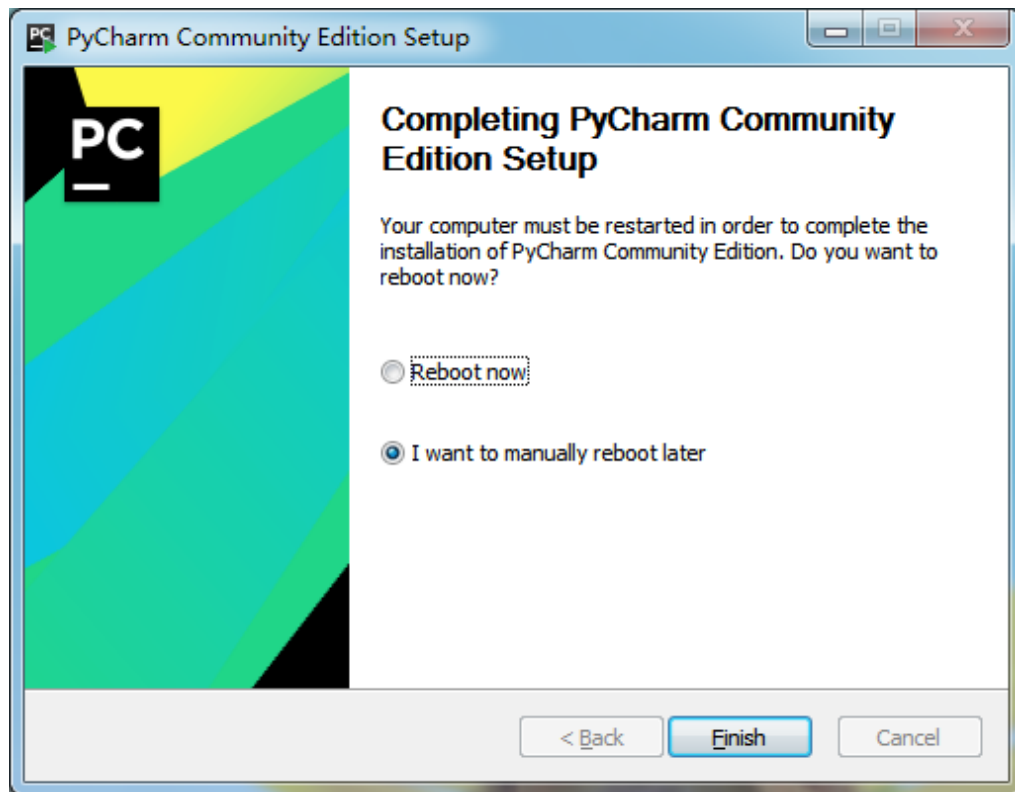
## 1.3.2 Python IDE——PyCharm 的下载与安装



潍坊科技学院  
Weifang University of Science and Technology

### 2. 安装PyCharm

(6) 片刻后PyCharm安装完成，界面如右图所示。单击【Finish】按钮可结束安装。

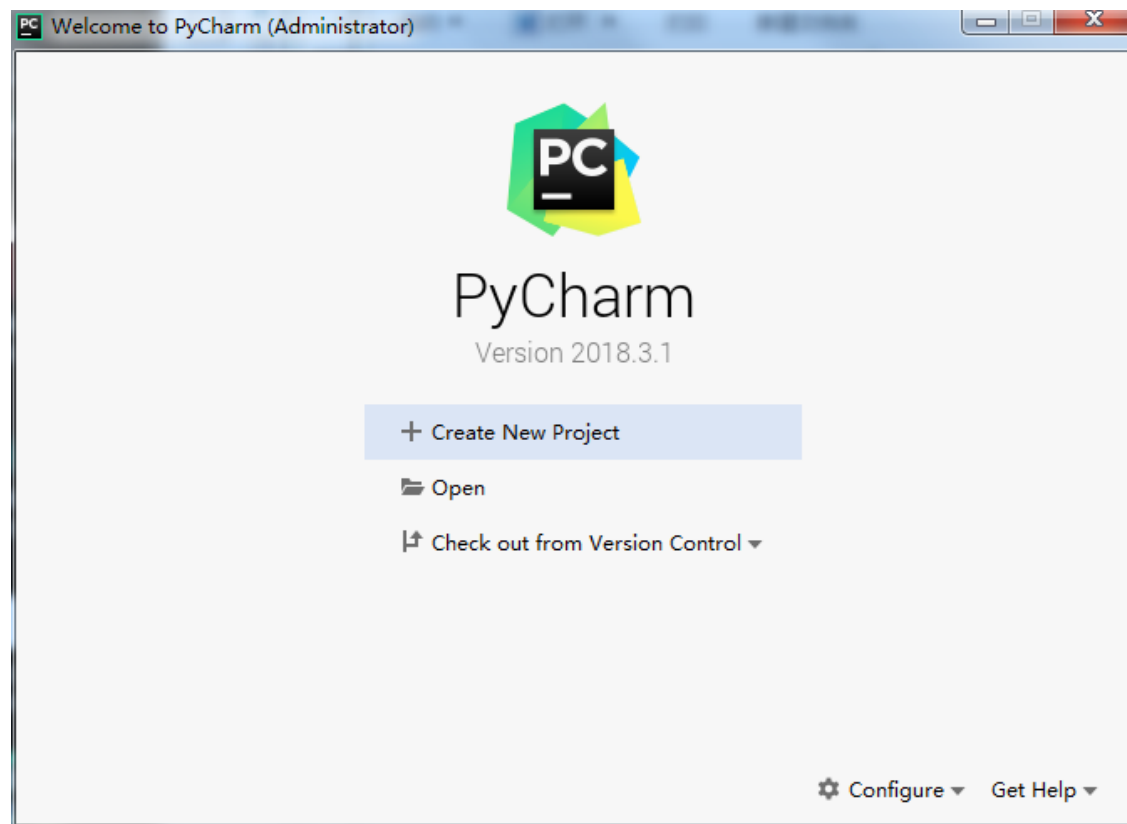




## 1.3.3 使用PyCharm编写Python程序



1. 单击桌面上PyCharm的快捷方式打开PyCharm，初次打开PyCharm时会弹出JetBrains Privacy Policy窗口，用户需在该窗口中勾选同意用户协议；之后会进入PyCharm的主题选择窗口，在该窗口中选择PyCharm的主题后方可启动PyCharm，进入**PyCharm的欢迎窗口**。

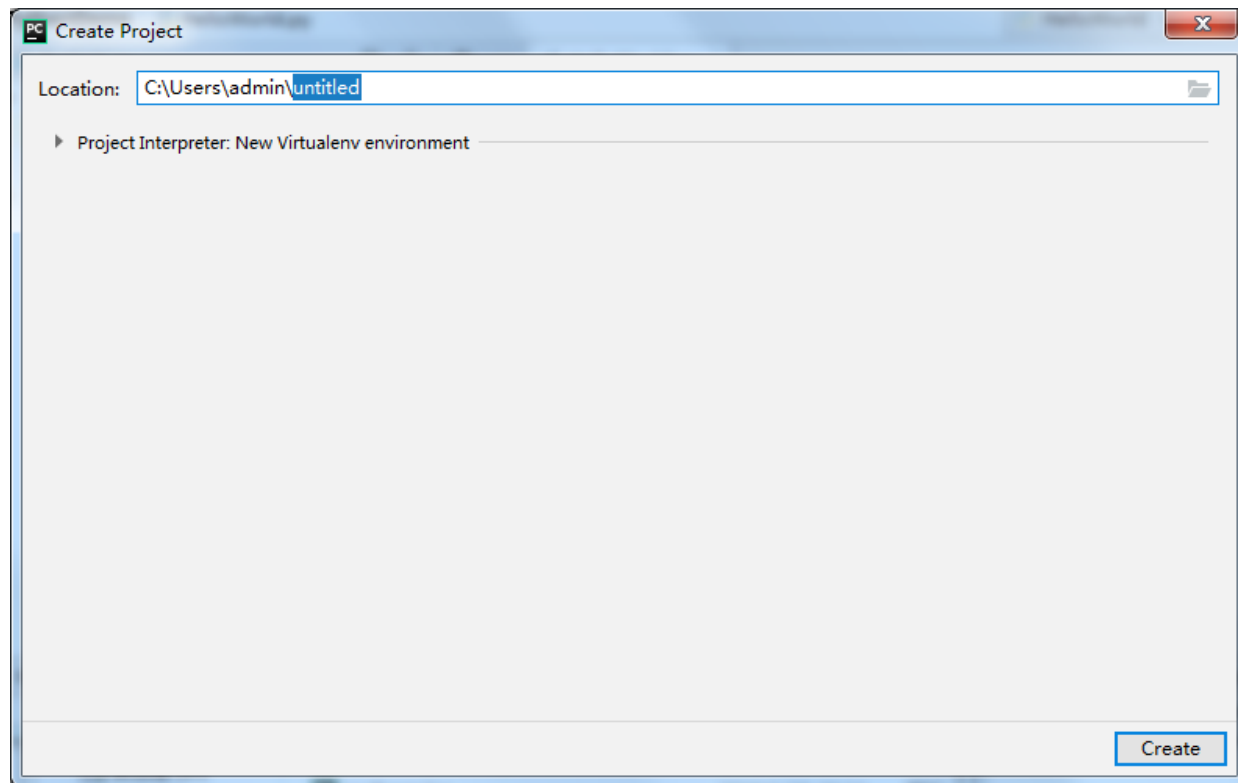




## 1.3.3 使用PyCharm编写Python程序



2. 单击“Create New Project”  
进入Create Project窗口。



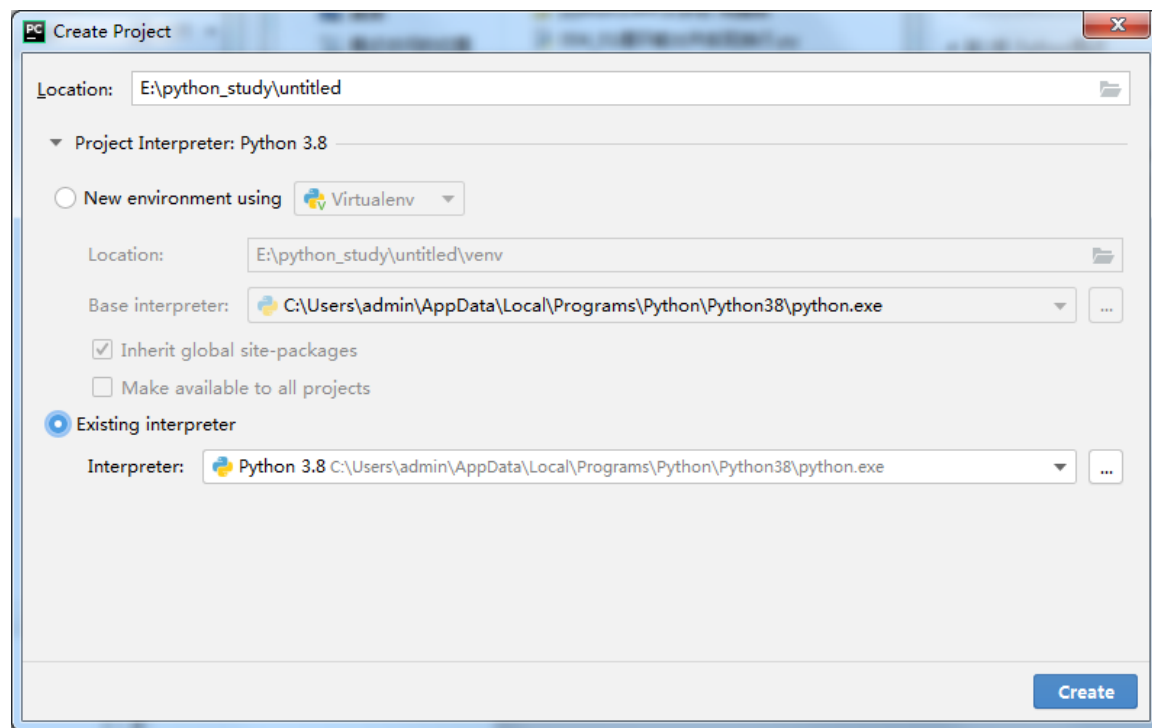




### 1.3.3 使用PyCharm编写Python程序



3. 在路径E:\python\_study\first\_proj下**创建项目first\_proj**，选择Existing interpreter并**配置Python解释器**，具体如下图所示。

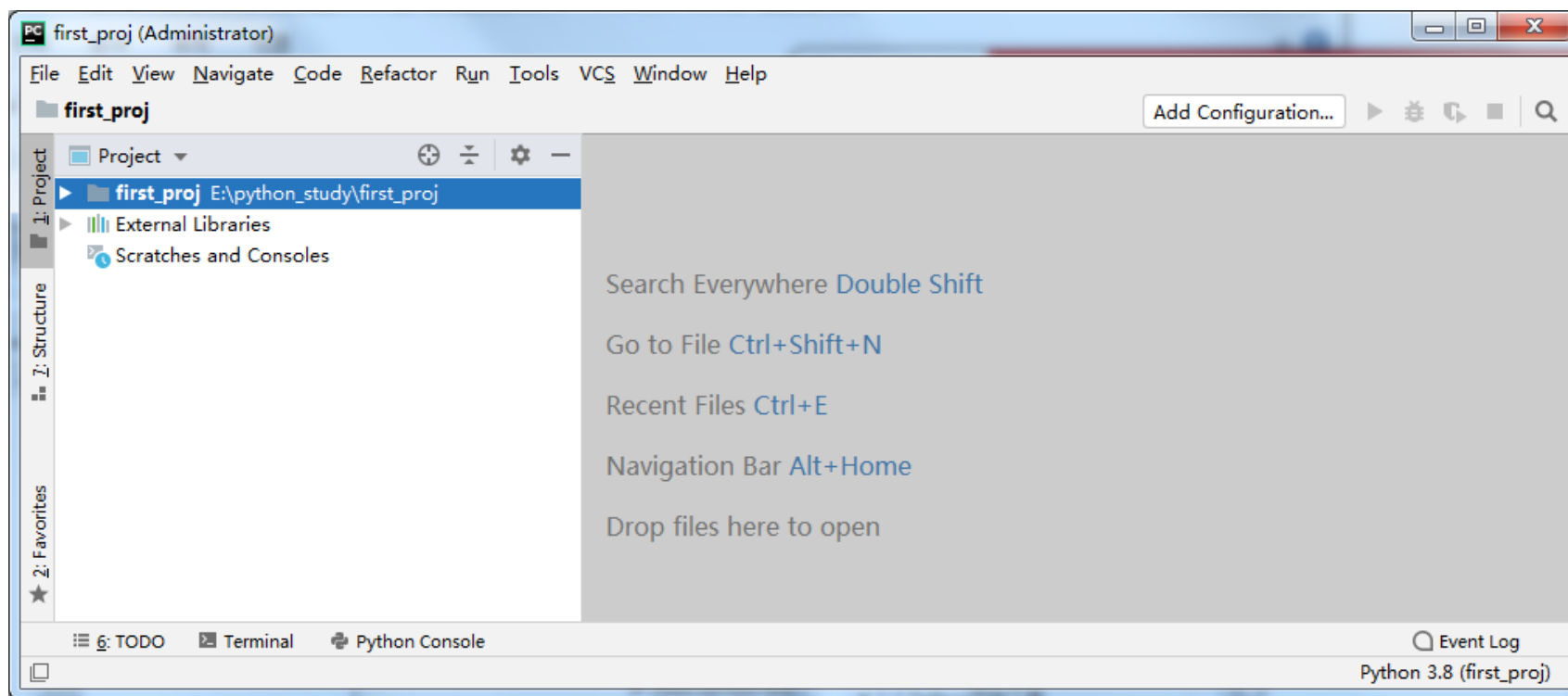




## 1.3.3 使用PyCharm编写Python程序



4.单击【Create】按钮完成项目创建并进入**项目管理界面**，如下图所示。

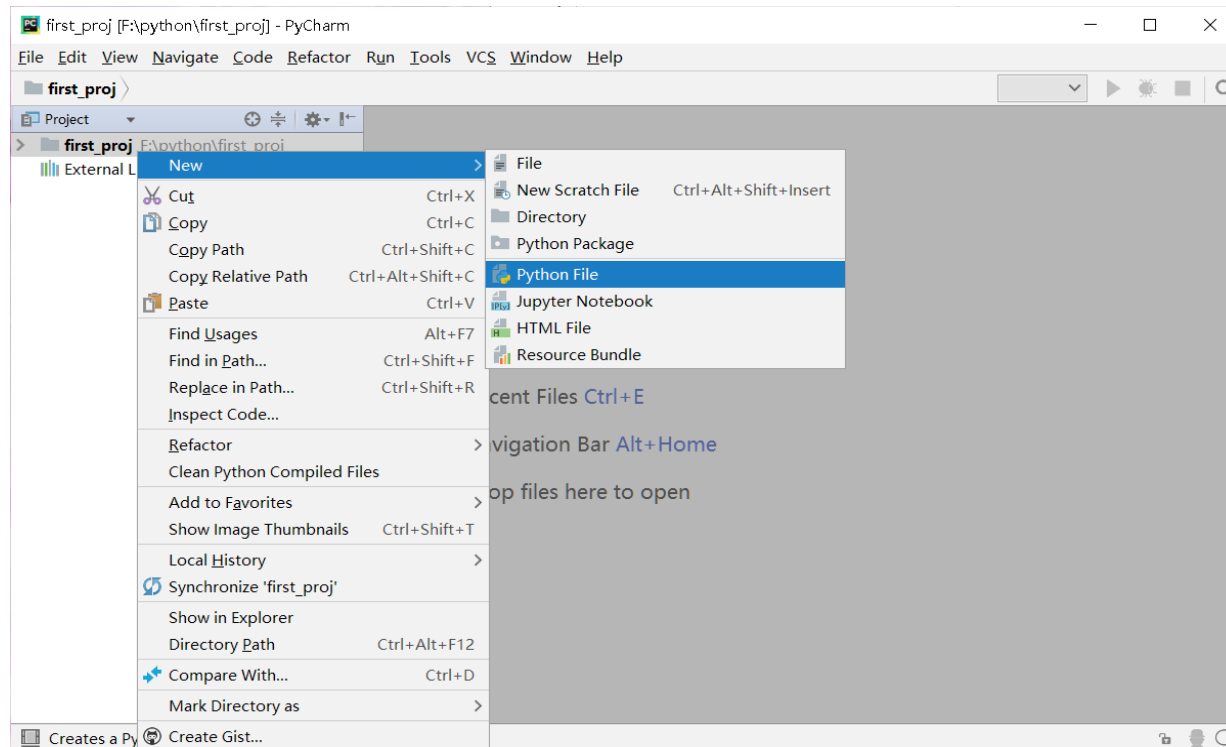




## 1.3.3 使用PyCharm编写Python程序



5. 经以上操作后我们创建了一个空Python项目，之后还需要在该项目中**添加Python文件**。右击项目名称，在弹出的下拉菜单中选择【New】→【Python File】。

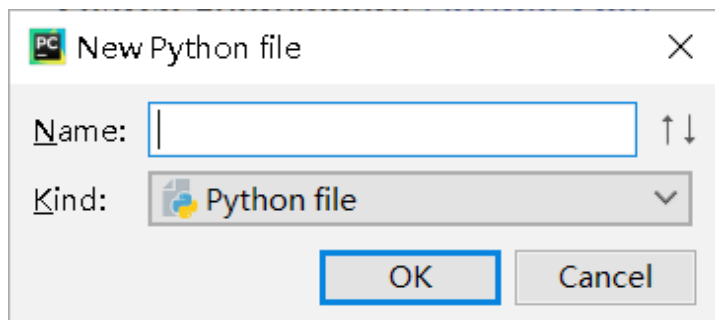




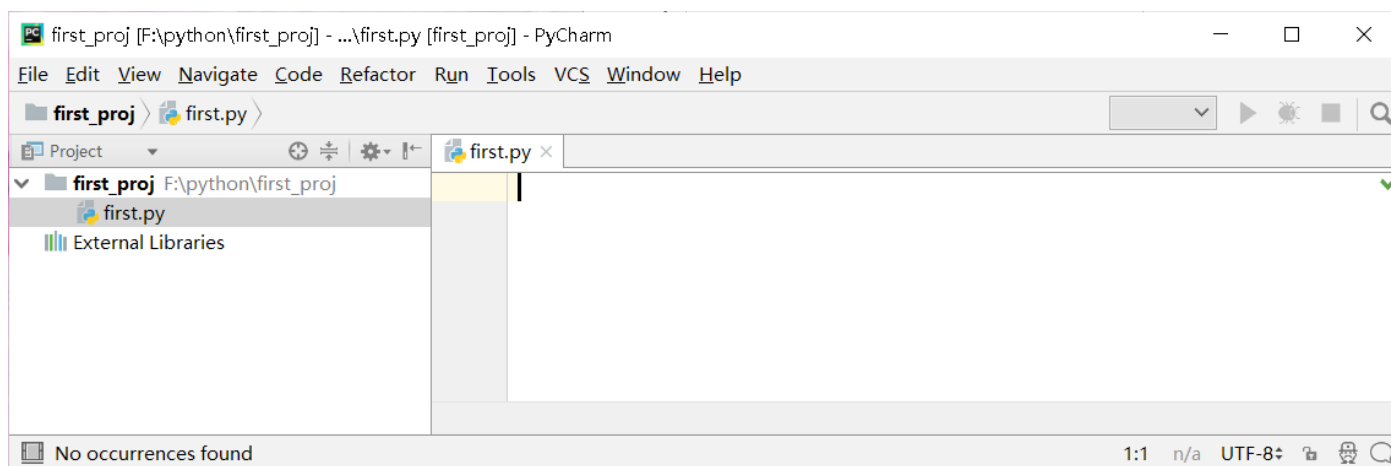
## 1.3.3 使用PyCharm编写Python程序



6. 单击下拉列表中的“Python File”将弹出“New Python file”窗口。



7. 这里添加的文件为“first.py”，文件添加完成后的PyCharm窗口如下图所示。





## 1.3.3 使用PyCharm编写Python程序



8. 在first.py文件中输入代码：`print( "Hello World!" )`，选中要执行的文件first.py，在右键下拉列表中选择“Run 'first' ”命令可执行该文件。文件执行结果将窗口下方显示，如下图所示。

```
Run: first x
E:\python\first_proj\venv\Scripts\python.exe E:/python/first_proj/first.py
Hello World!
Process finished with exit code 0
```

观察左图可看到程序的运行结果“**hello world**”，说明程序成功执行。



**01 认识Python**

**02 Python解释器的安装与Python程序的运行**

**03 Python开发工具**

**04 Python模块**



## 1.4.1 模块的安装



1.3节编写的Python程序只有极少的代码，实现的功能非常简单。随着程序复杂度的提高代码量会同步增长，这时若还是在一个文件中编写代码，代码的维护就会越来越困难。为了保证代码的可维护性，开发人员通常将一些功能性代码放在其他文件中，这样用于存放功能性代码的文件就是**模块**。





## 1.4.1 模块的安装



作为一种强大且便捷的编程语言，Python自然支持以模块的形式组织代码。Python内置了一些**标准模块**，Python的使用者也贡献了丰富且强大的**第三方模块**；标准模块可以直接导入与使用，第三方模块则需先行安装。



## 1.4.1 模块的安装

利用Python内置的**pip工具**（安装Python3.8时会自动安装该工具）可以非常方便地**安装Python第三方模块**，该工具可在命令行中使用，语法格式如下：

```
pip install 模块名
```

### 注意

pip是在线工具，pip命令执行后，它需要联网获取模块资源，若没有网络或网络不佳，pip将无法顺利安装第三方模块。



## 1.4.2 模块的导入与使用



在使用模块中定义的内容之前，需先将模块导入到当前程序。Python使用**import**关键字导入模块，其语法格式如下：

```
import 模块1,模块2,...
```

模块导入后，可通过点字符“.”调用模块中的内容，其语法格式如下：

- ✓ 模块.函数
- ✓ 模块.变量



## 1.4.2 模块的导入与使用

通过点字符调用模块中的内容可避免多个模块中存在同名函数时代码产生歧义，但若不存在同名函数，可使用`from...import...`语句直接将模块的指定内容导入程序，并在程序中直接使用模块中的内容。

### 示例

```
from pygame import init  
init()
```



## 1.4.2 模块的导入与使用



使用**from...import...语句**也可将指定模块的全部内容导入当前程序，此时可使用“\*”指代模块中的全部内容。

示例

```
from pygame import *
```

虽然**from...import \***可以方便地导入一个模块中的所有内容，但考虑到代码的可维护性，此种方式不应被过多地使用。



## 多学一招：代码的组织方式——模块、包与库

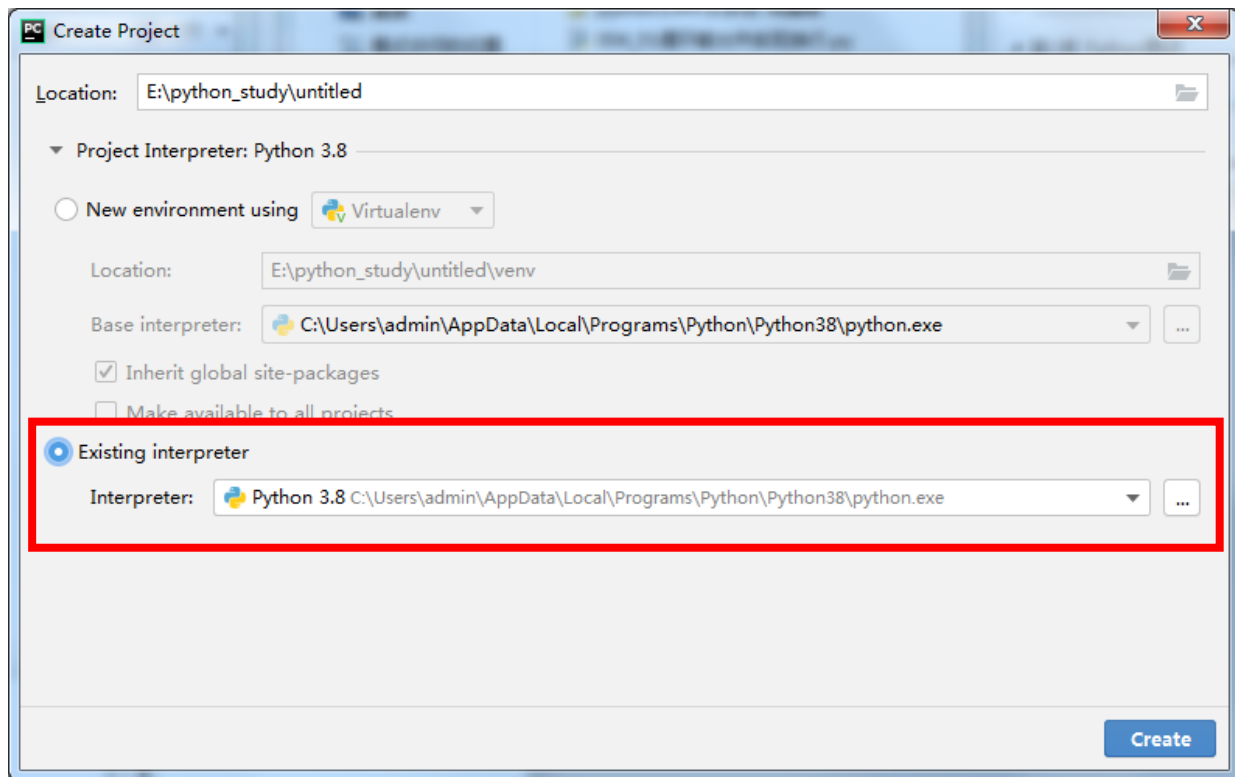


**模块** (module)、**包** (package) 和**库** (lib) 是Python组织代码的三种方式。

- ❑ 模块是最基础的代码组织方式，每个包含有组织的代码片段的.py文件都是一个模块，**文件名就是模块名**。
- ❑ 包以类似目录的结构组织模块文件或子包，简单来说，一个包含 `__init__.py` 文件的目录就是一个包。**包中必有 `__init__.py` 文件**，可以有多个模块或子包。
- ❑ 库是一个**抽象概念**，它是指具有相关功能的模块的集合。



## 附：Anaconda的安装

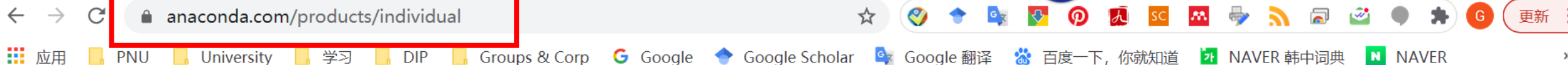


Anaconda是一个开源的Python发行版本，其包含了conda、python以及一大堆安装好的工具包，比如常用的numpy、scipy、pandas等





# 附：Anaconda的安装



## Anaconda Installers

### Windows

Python 3.8

64-Bit Graphical Installer (457 MB)

32-Bit Graphical Installer (403 MB)

### MacOS

Python 3.8

64-Bit Graphical Installer (435 MB)

64-Bit Command Line Installer (428 MB)

### Linux

Python 3.8

64-Bit (x86) Installer (529 MB)

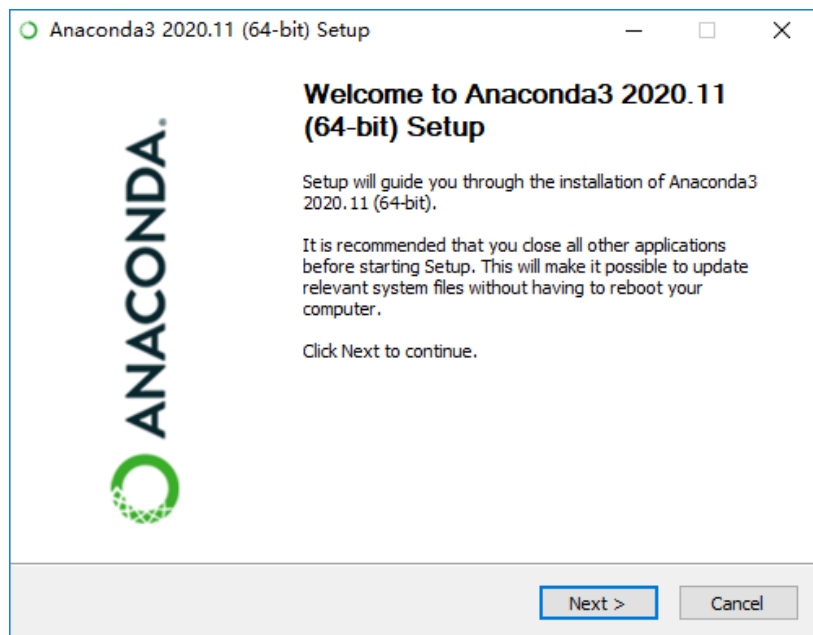
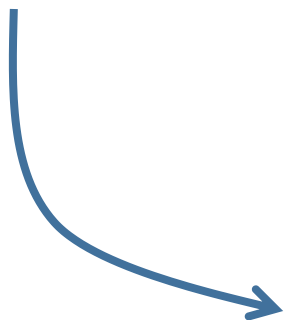
64-Bit (Power8 and Power9) Installer (279 MB)



# 附：Anaconda的安装



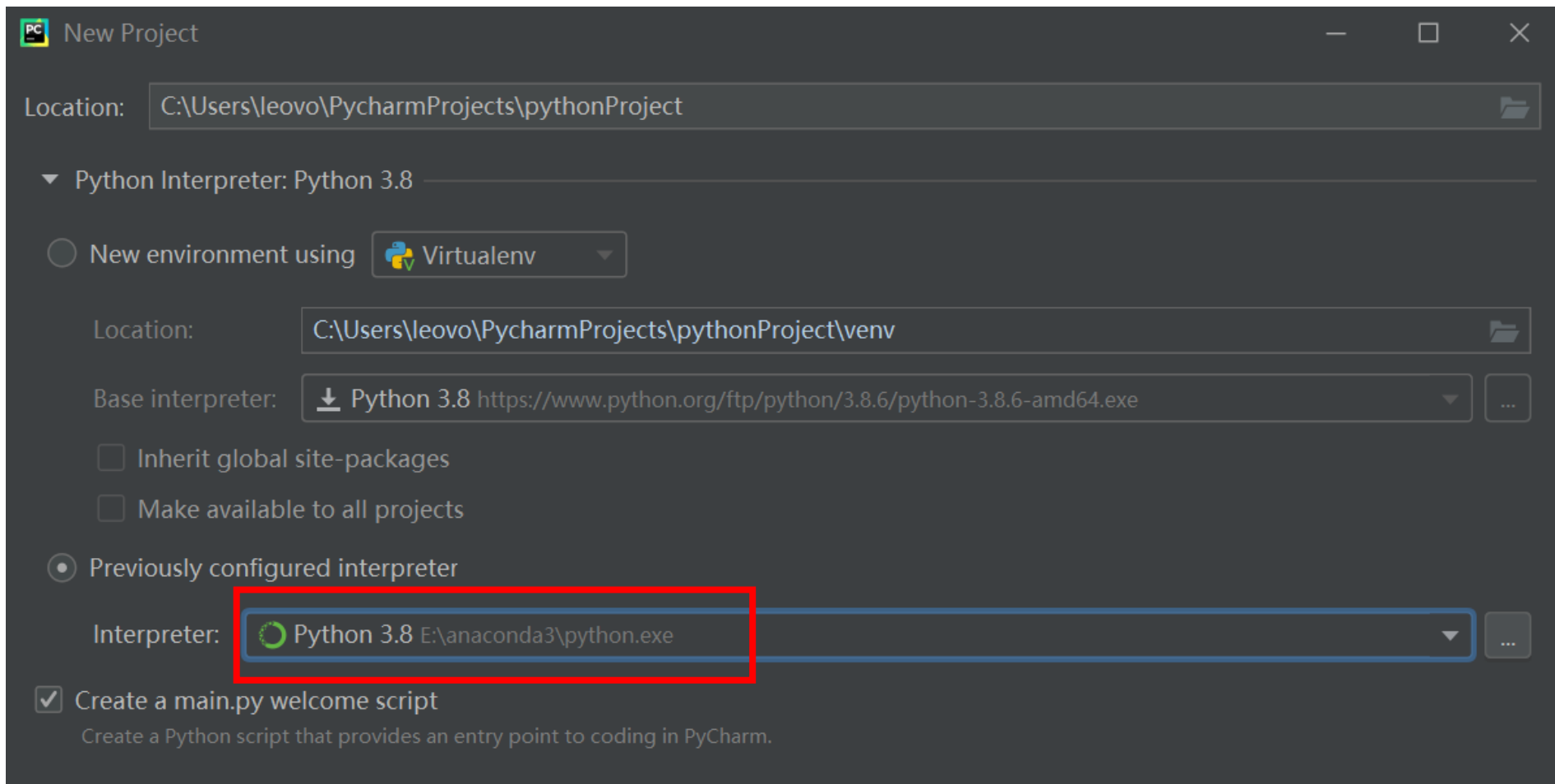
- Anaconda3-2020.11-Windows-x86\_64.exe
- AnySign\_Installer.exe
- BaiduNetdisk-5.7.3.exe



<input checked="" type="checkbox"/>	anaconda3	2020/12,
<input type="checkbox"/>	Captura	2020/12,
<input type="checkbox"/>	Installations	2021/3/3
<input type="checkbox"/>	MailMasterData	2020/12,
<input type="checkbox"/>	Program Files	2020/12,
<input type="checkbox"/>	Program Files (x86)	2020/12,



## 附：Anaconda的安装





## 1.6本章小结



本章首先通过Python的**发展史**、**语言特点**两方面简单介绍了Python，然后介绍了**如何安装Python解释器**，之后介绍了**PyCharm**、**Jupyter Notebook**两种常用的Python编辑器，以及**如何编写Python程序**，最后介绍了**Python模块的安装、导入与使用**。

通过本章的讲解，希望读者能对Python语言有个简单的认识，并能熟练搭建Python开发环境，了解Python编辑器的使用方式，以及模块的安装与使用。