

前言

计算机的概念

- 计算机是根据指令操作数据的设备
- 具备功能性和可编程性两个基本特性
 - 功能性指对数据的操作，表现为数据计算、输入输出处理和结果存储等。
 - 可编程性指它可以根据一系列指令自动地、可预测地、准确地完成操作者的意图
- 计算机硬件所依赖的集成电路规模按照摩尔定律以指数方式增长
- 计算机运行速度也接近几何级数快速增加
- 计算机所能高效支撑的功能不断丰富发展
- 程序设计语言经历了从机器语言到高级语言的发展过程，朝着更接近自然语言的方向发展。

计算机技术发展水平的时代性

第一阶段：1946年到1981年，“计算机系统结构阶段”

——催生了C语言

第二阶段：1982年到2007年，“计算机网络和视窗阶段”

——催生了JAVA\VC\VB语言

第三阶段：2008年至今，“复杂信息系统阶段”。

——Python成为主流语言

第四阶段：约20年后未来的某个时期开始，“人工智能阶段”

——Python辉煌期

程序设计语言概述

- 程序设计语言包括编译执行和解释执行两种方式
 - 程序设计语言是计算机能够理解和识别用户操作意图的一种交互体系，它按照特定规则组织计算机指令，使计算机能够自动进行各种运算处理。按照程序设计语言规则组织起来的一组计算机指令称为计算机程序。
- 机器语言
 - 机器语言是一种二进制语言，它直接使用二进制代码表达指令，是计算机硬件可以直接识别和执行的程序设计语言。
 - 例如：执行数字2和3的加法，16位计算机上的机器指令为：11010010 00111011，不同计算机结构的机器指令不同
- 汇编语言
 - 使用助记符与机器语言中的指令进行一一对应，在计算机发展早期帮助程序员提高编程效率
 - 例如：执行数字2和3的加法，汇编语言指令为：add 2, 3, result，运算结果写入result。

机器语言和汇编语言都直接操作计算机硬件并基于此设计，所以它们统称为低级语言。

- 高级语言

- 高级语言区别于低级语言在于，高级语言是接近自然语言的一种计算机程序设计语言，更容易地描述计算问题并利用计算机解决计算问题
- 例如：执行数字2和3加法的高级语言代码为： $\text{result} = 2 + 3$

高级语言分类

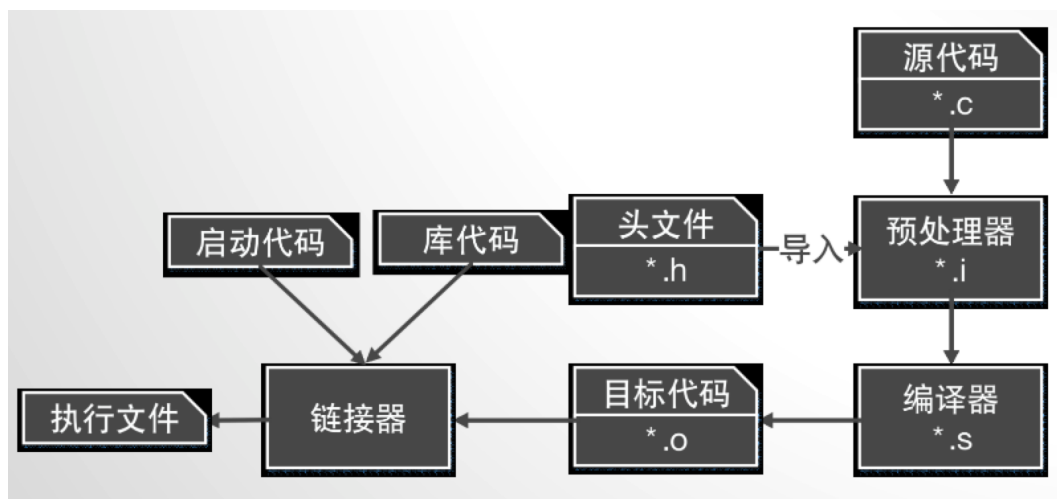
- 编译型语言
C/C++、Fortran、Delphi、Java
- 解释型语言
Ruby、Python、JavaScript



编译型语言

源代码->可执行程序

预处理 -> 编译 -> 汇编 -> 链接



C语言编译过程

```

1  #include <stdio.h>
2
3  int main(int argc, char* argv[])
4  {
5      printf("This is a C Program.\n");
6      return 0;
7  }
  
```

```

1  $ gcc -E -o hello.i hello.c
2  $ gcc -S -o hello.s hello.i
3  $ gcc -c -o hello.o hello.s
4  $ gcc -o hello.bin hello.o
  
```

编译型语言特点

执行前需要进行编译

再次执行无需编译

机器码执行效率高

源代码修改后需重新编译

解释型语言

```

1  #shell 编程实例
2
3  #!/usr/bin/bash
4  echo "This is a Shell Scripy."
  
```

```

1  #Python 编程实例
2
3  #!/opt/anaconda/bin/python
4  print("This is a Python program.\n")
  
```

解释性语言特点

运行时需要解释器

以文本形式存在

程序源码既是最终执行文件

无需编译

每次执行需逐行解释

执行效率低

源代码修改后可直接运行

1. python简介

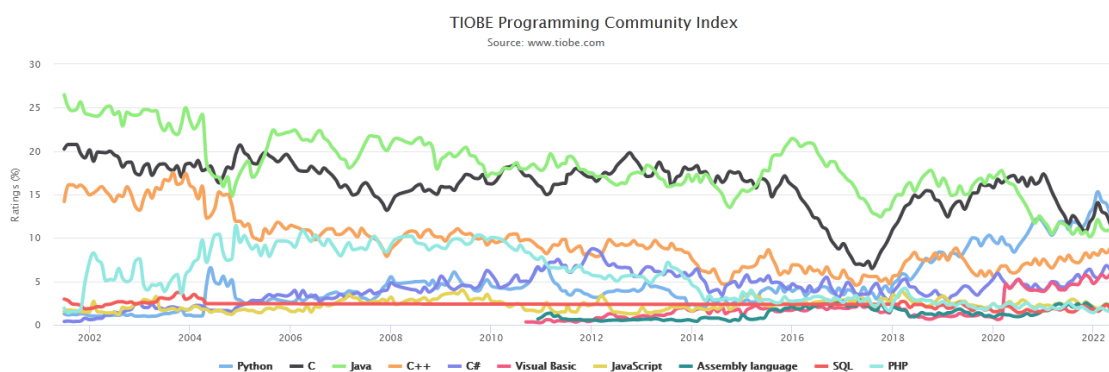
1.1 初识Python

- Python起源
 - 1989年荷兰人Guido van Rossum（吉多·范罗苏姆）发明Python
 - 1991年Python公开发行第一个版本



- 主流版本
 - Python2.X (2002)
 - Python3.X (2008)
 - Python 3.6
 - Python 3.8
- 优点
 - 简单易学
 - 开源免费

- 方便移植：由于它的开源本质，Python已经被移植在许多平台上（经过改动使它能够工作在不同平台上）。
- 开发高效：Python的底层是用C语言写的，很多标准库和第三方库也都是用C写的，运行速度非常快。
- 解释型语言：在计算机内部，Python解释器把源代码转换成为字节码的中间形式，然后再把它翻译成计算机使用的机器语言并运行。这使得使用Python更加简单。也使得Python程序更加易于移植
- 面向对象：Python是完全[面向对象](#)的语言。[函数](#)、模块、数字、[字符串](#)都是对象。并且完全支持继承、重载、派生、多继承，有益于增强源代码的复用性。Python支持重载运算符和动态类型。相对于[Lisp](#)这种传统的函数式编程语言，Python对函数式设计只提供了有限的支持。有两个标准库（functools, itertools）提供了Haskell和Standard ML中久经考验的函数式程序设计工具
- 缺点
 - 运行效率低
 - 代码安全性低
- 编程语言排名
 - TIOBE 编程语言排行榜



- 应用场景
 - 操作系统管理
 - 图形界面编程
 - Web开发
 - 数值计算和科学计算
 - 游戏、图像、机器学习、人工智能
- 平台支持
 - Windows
 - Linux
 - MacOS
- 环境种类



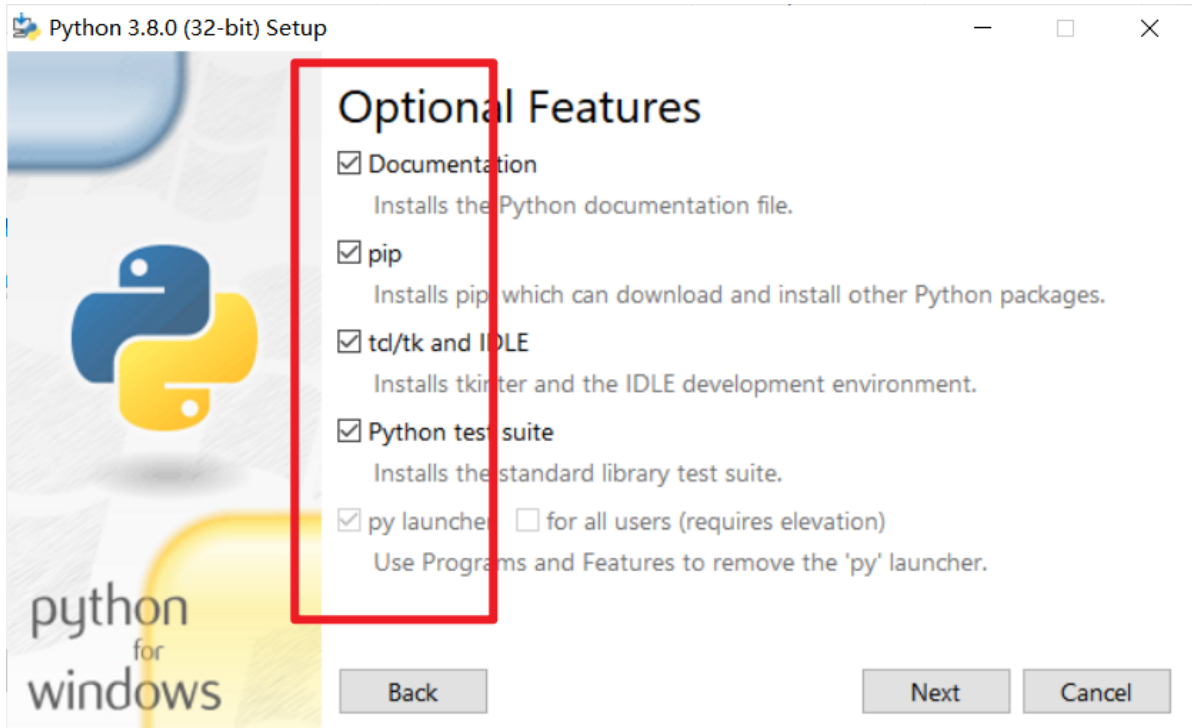
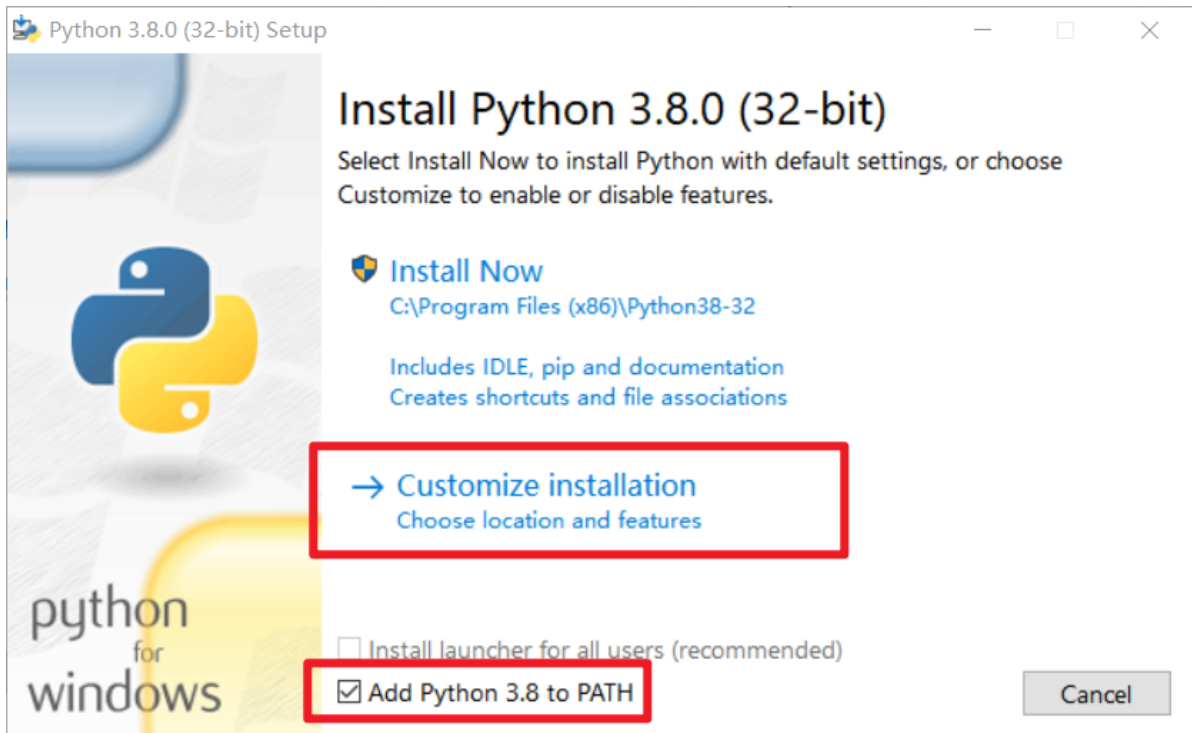
- 纯净Python包
- 发行版工具包
- 集成开发环境
- 常用工具
 - pip install 库名==3.1 -i 源地址
 - 源地址

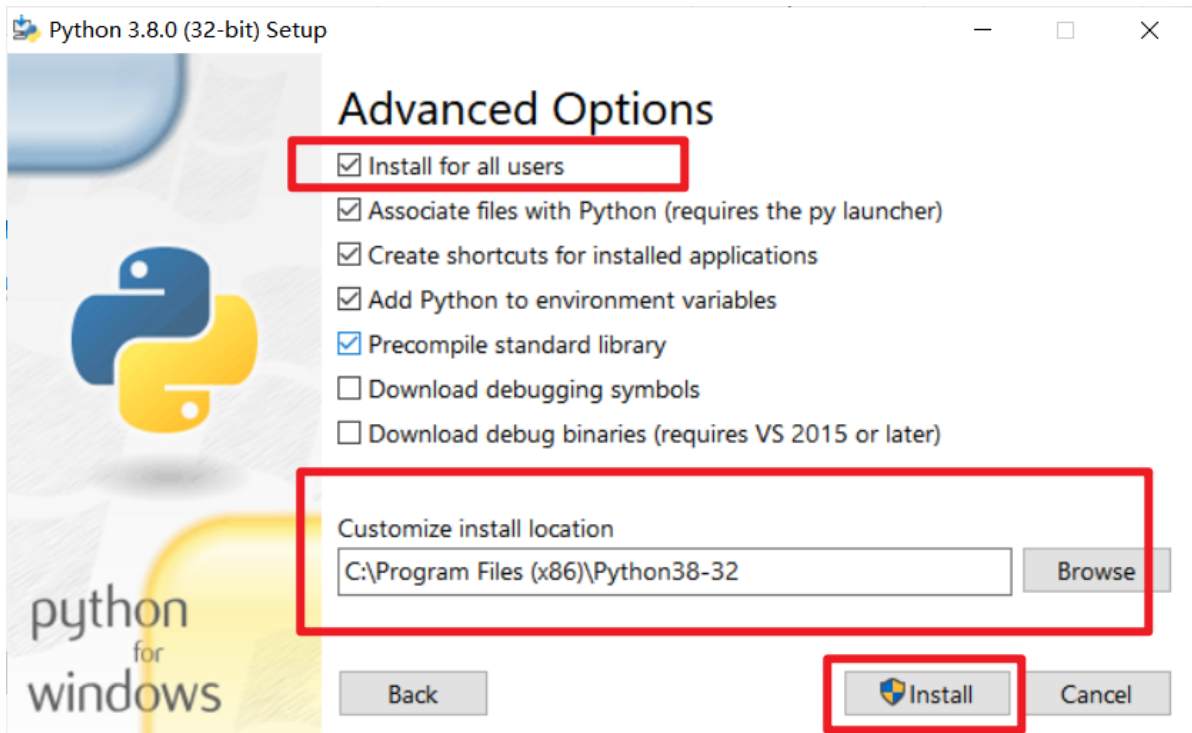
```
1 1 豆瓣(douban) https://pypi.douban.com/simple
2 2 清华大学 https://pypi.tuna.tsinghua.edu.cn/simple
3 3 阿里云 https://mirrors.aliyun.com/pypi/simple
4 4 中国科技大学 https://pypi.mirrors.ustc.edu.cn/simple
```

- IPython: 增强交互式环境 ipython
- jupyter notebook: 写代码和笔记、代码可以运行

2. 安装调试环境

2.1 安装python



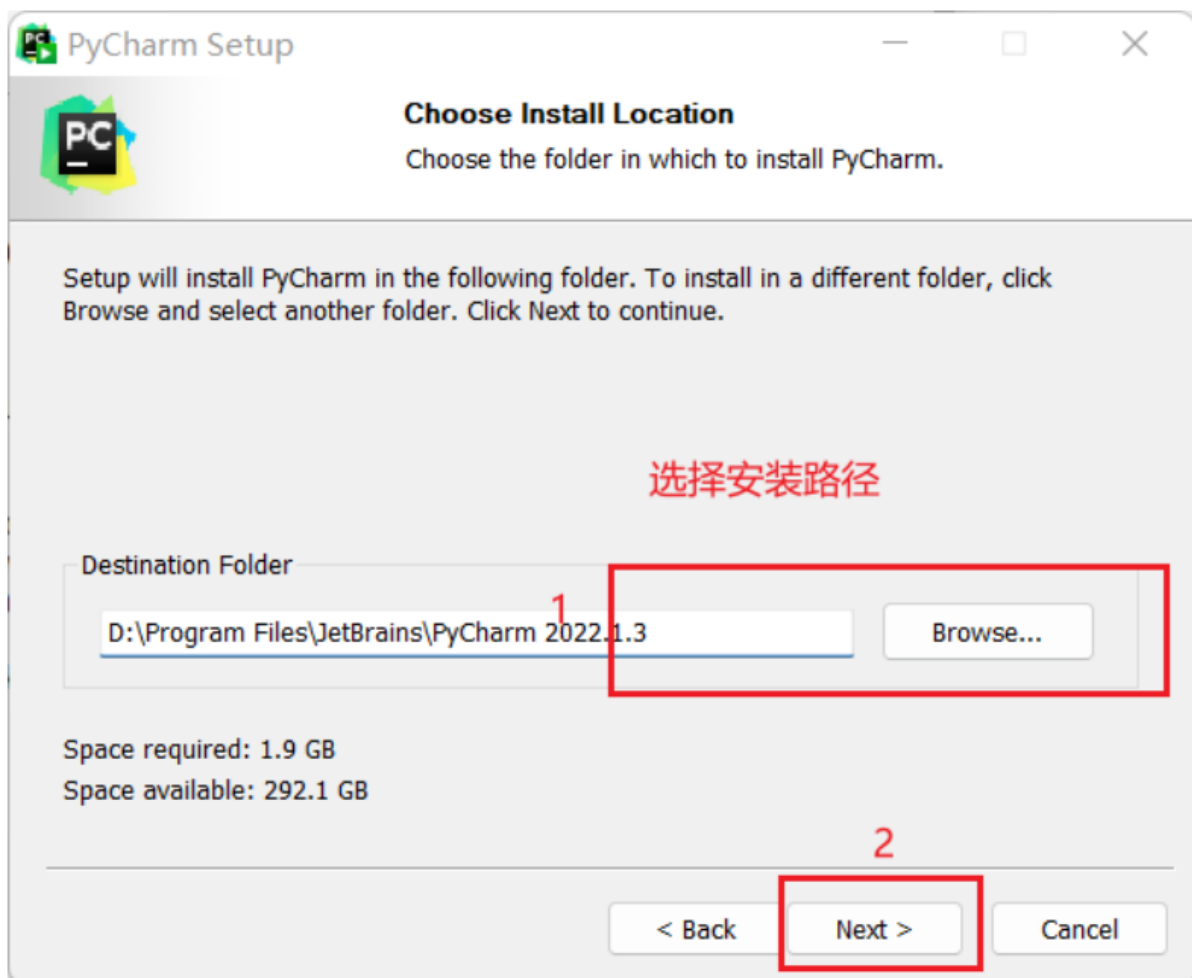
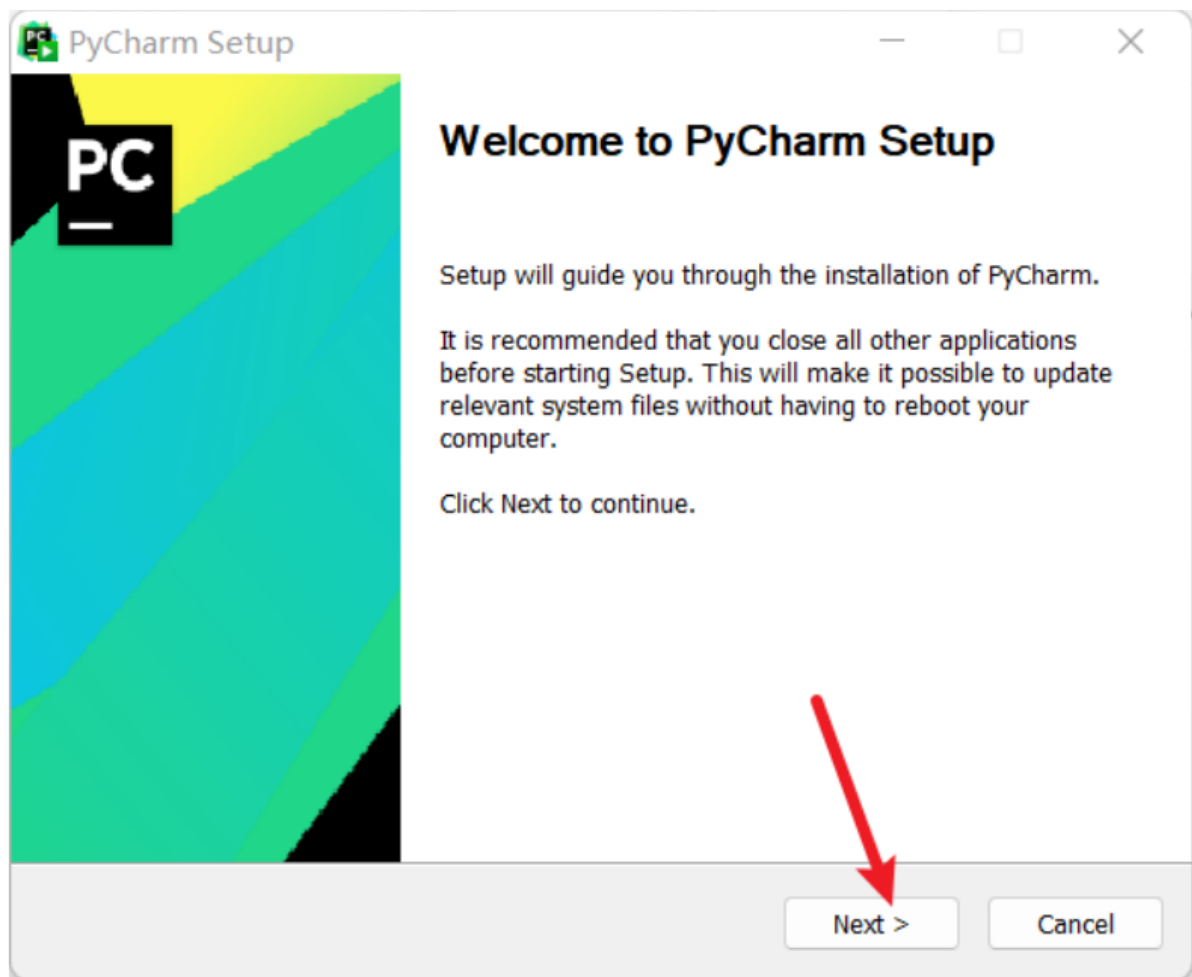


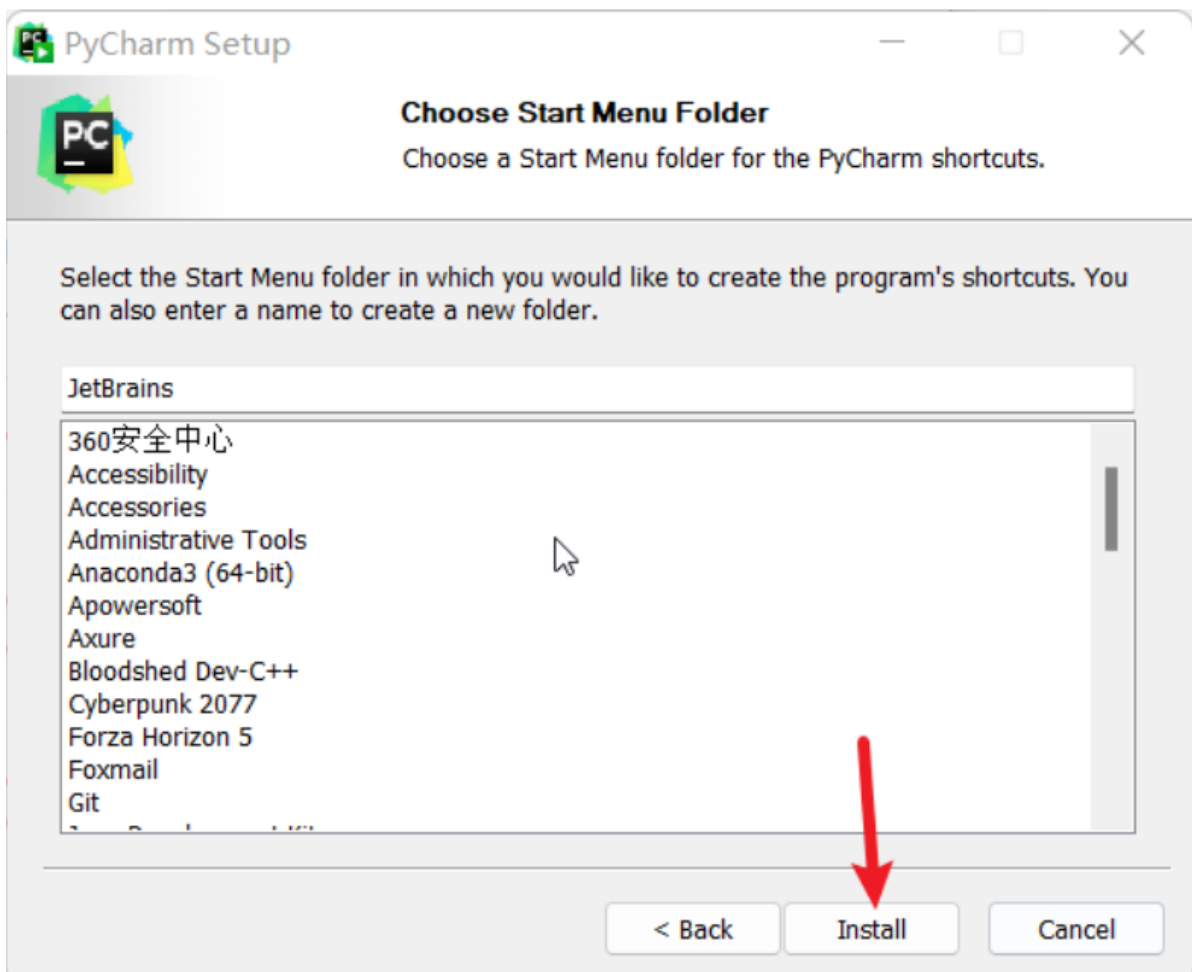
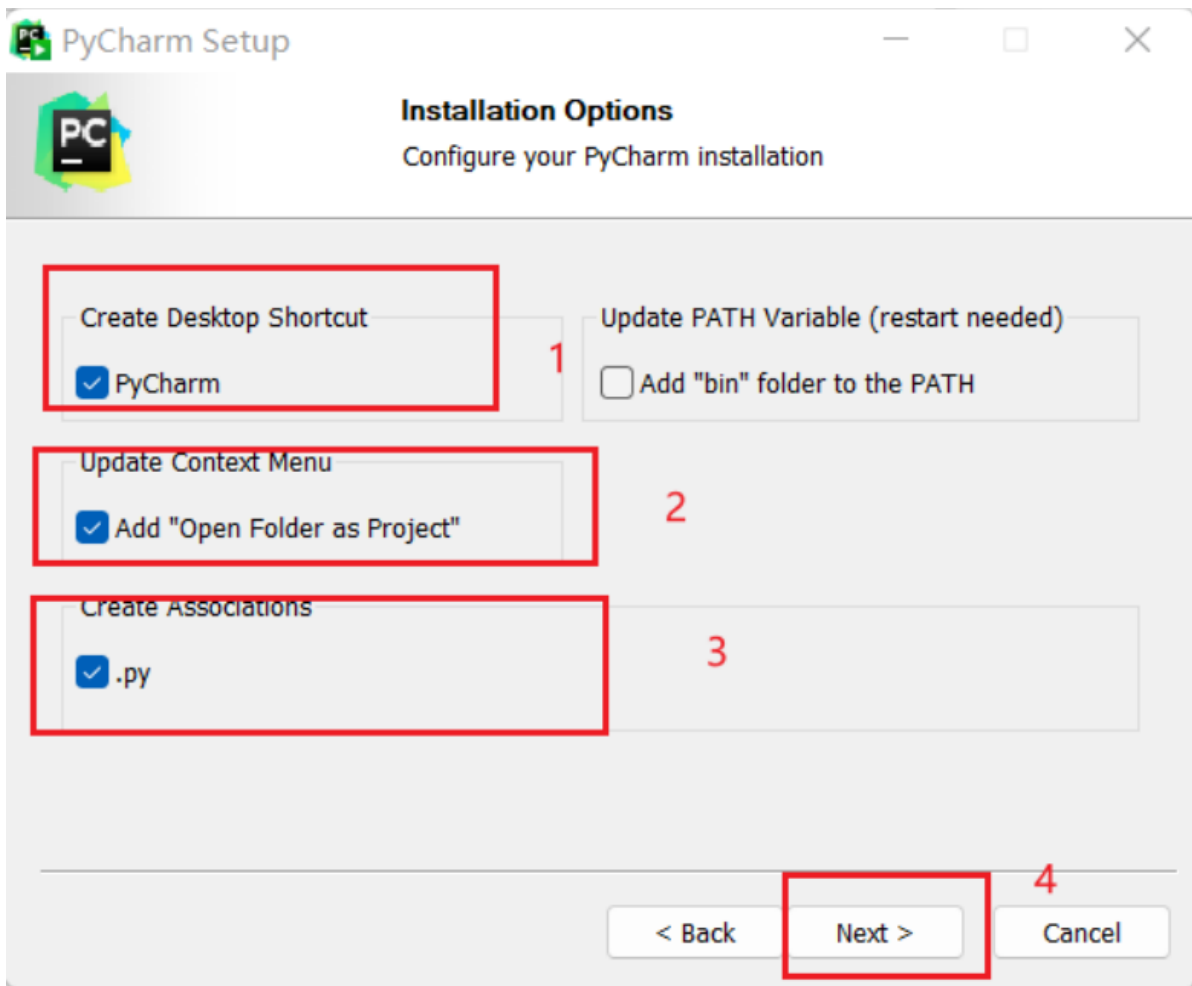
2.2 安装PyCharm

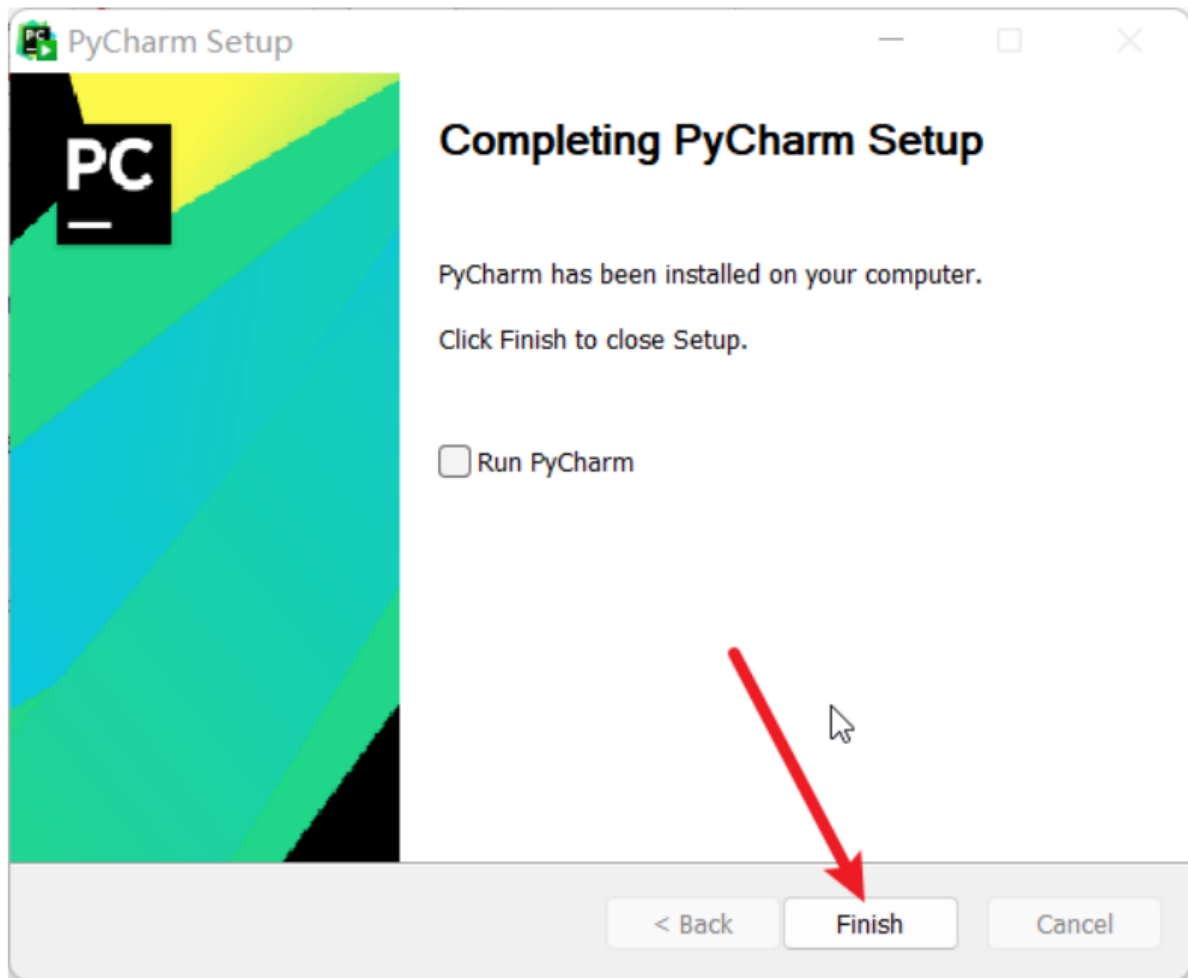
PyCharm是一种Python **IDE**（集成开发环境），带有一整套可以帮助用户在使用Python语言开发时==提高其效率的工具

选择pycharm双击运行

Crack	2022/9/16 14:24	文件夹	
pycharm-2022.1.3.exe	2022/6/28 16:31	应用程序	476,707 KB
破解步骤.txt	2022/9/16 11:08	文本文档	1 KB







双击打开Crack准备破解pycharm

名称	修改日期	类型	大小
Crack	2022/9/16 14:24	文件夹	
pycharm-2022.1.3.exe	2022/6/28 16:31	应用程序	476,707 KB
破解步骤.txt	2022/9/16 11:08	文本文档	1 KB

关闭杀毒软件，双击永久激活，一直点确定

关闭杀毒软件，双击永久激活，一直点确定

名称	修改日期	类型	大小
micool_config	2022/9/16 14:24	文件夹	
jh.cmd	2022/5/5 16:33	Windows 命令脚本	3 KB
备用激活.cmd	2022/3/18 15:25	Windows 命令脚本	2 KB
永久激活.vbs	2022/3/18 14:47	VBScript Script ...	1 KB

系统找不到指定的文件。
系统找不到指定的文件。
开始放置激活所需文件...
复制了 25 个文件
开始逐个激活...
正在激活所有2017版本...
正在激活所有2018版本...
正在激活所有2019版本...
复制了 0 个文件
正在激活所有2020版本...
正在激活所有2021版本...
正在激活所有2022版本...
复制了 198 个文件
“请注意你的杀毒软件提示，一定要允许”

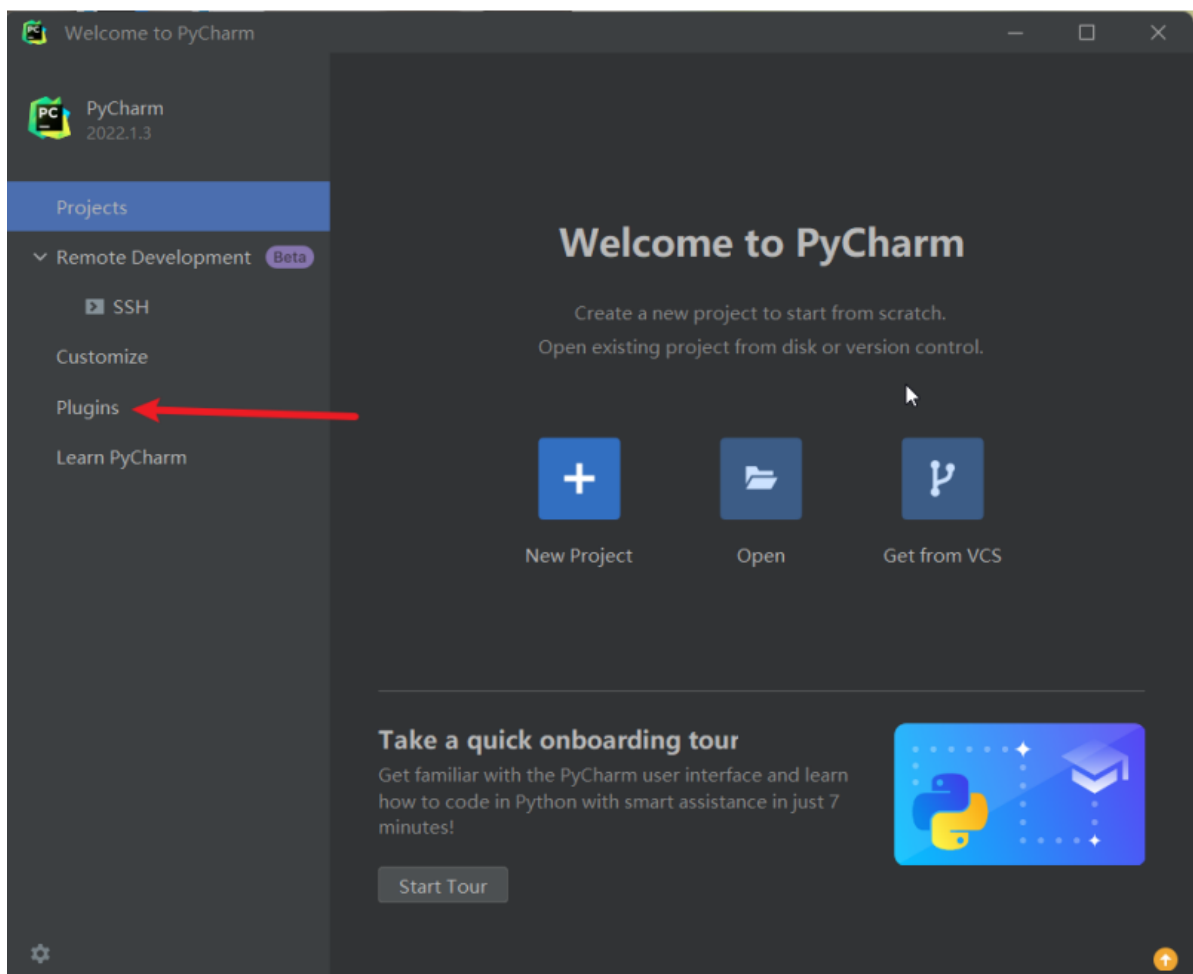
复制了 0 个文件

拒绝访问。
拒绝访问。
拒绝访问。

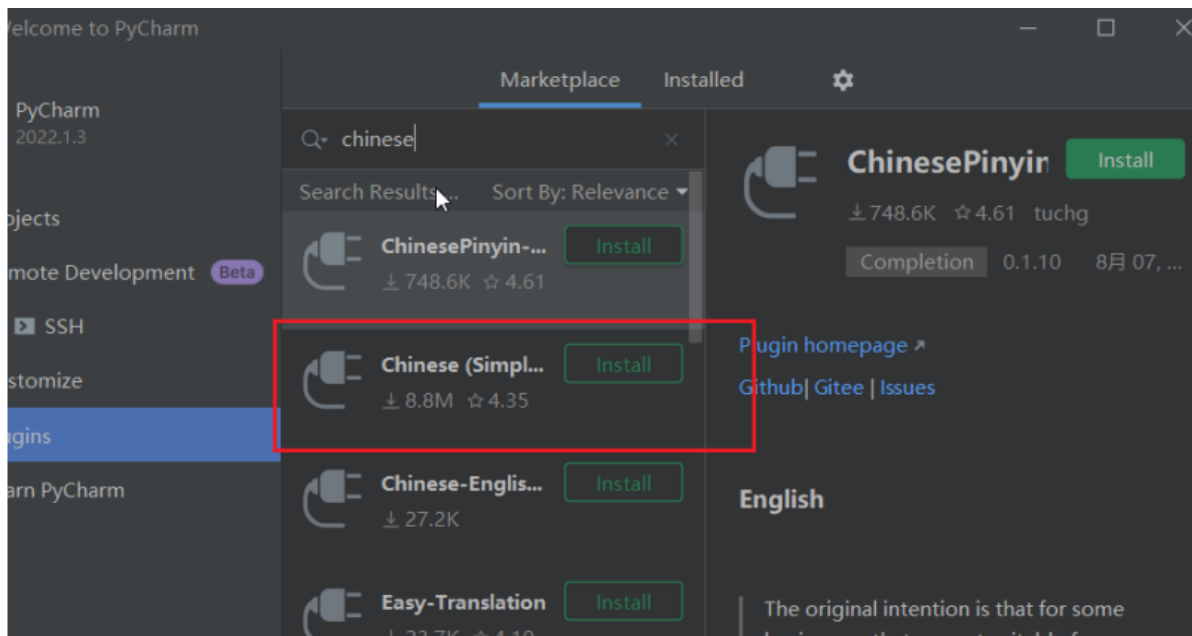
Windows IP 配置

已成功刷新 DNS 解析缓存。
“刷新DNS完成”
分析错误
激活成功! By: jetbrains 全家桶激活 别家买的均为盗版

打开pycharm，汉化软件



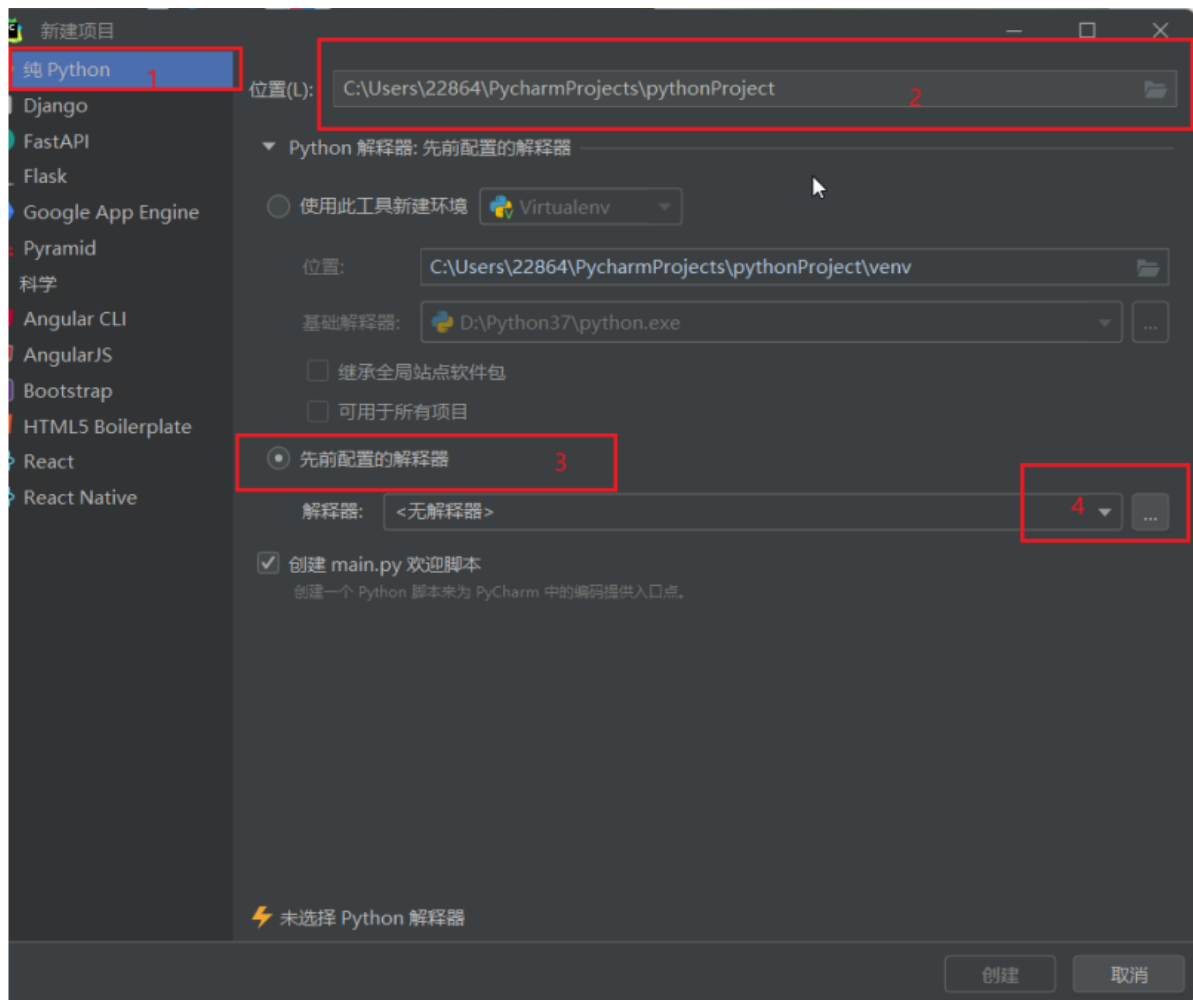
选择框起来的，点击install，等待下载完成后重启，汉化完成



配置python3.8



- 1、选择创建纯python项目
- 2、更改项目保存路径
- 3、使用先前配置的解释器
- 4、选择系统解释器，找到python安装文件夹下的python.exe,点击确定，在点击创建



2.3 Python程序的执行方式

2.3.1 交互式

在命令行输入指令，回车即可得到结果。

- (1) 打开终端
- (2) 进入交互式：python3
- (3) 编写代码：print("hello world")
- (4) 离开交互式：exit()

交互模式的特点

- 交互模式下只能够输入Python命令
- 交互模式下代码执行结果会自动打印
- 交互模式下不需要缩进代码（标准解释器下）
- 交互模式下支持多行语句的编写，复合语句用空行结束(复合语句需要缩进)
- 交互模式下一次只能运行一条语句
- 便于实验Python语句的功能
- 方便代码块功能的测试

2.3.2 文件式

将指令编写到.py文件，可以重复运行程序。

- (1) 编写文件
- (2) 打开终端
- (3) 进入程序所在目录：cd 目录
- (4) 执行程序：python 文件名

2.4. 执行过程

计算机只能识别机器码(1010)，不能识别源代码(如.py)。

(1) 由源代码转变成机器码的过程分成两类：编译和解释。

(2) 编译：在程序运行之前，通过编译器将源代码变成机器码，例如：C语言。

-- 优点：运行速度快

-- 缺点：开发效率低，不能跨平台

(3) 解释：在程序运行之时，通过解释器对程序逐行翻译，然后执行，例如javascript。

-- 优点：开发效率高，可以跨平台

-- 缺点：运行速度慢

(4) python是解释型语言，但为了提高运行速度，使用了一种编译的方法。编译之后得到pyc文件，存储了字节码（特定于Python的表现形式，不是机器码）。

源代码 -- 编译 --> 字节码 -- 解释 --> 机器码

|————1次———|

2.5 解释器类型

(1) CPython (C语言开发)：标准Python的实现方式，底层用C语言实现，运行速度最快、最完整。

(2) Jython (java开发)：目的是为了与JAVA语言集成，底层用JAVA实现，编译过程中形成 JAVA字节码并将得到的字节码映射到JVM上。

(3) IronPython (.net开发)：基于.NET实现，目的是兼容.NET相关平台。