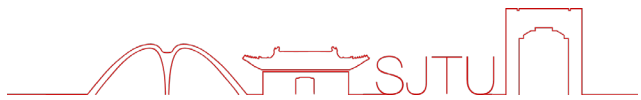




上海交通大学
SHANGHAI JIAO TONG UNIVERSITY



Python实践

晏轶超

2025年3月

饮水思源 · 爱国荣校

目 录

1

Python基础

2

软件安装与使用

3

Python实践



1. Python基础 — Python简介



- Python是一个高层次的结合了解释性、互动性和面向对象的脚本语言
- Python 是由吉多·范罗苏姆在八十年代末和九十年代初，在荷兰国家数学和计算机科学研究所设计出来的



1. Python基础 — Python在AI中的应用

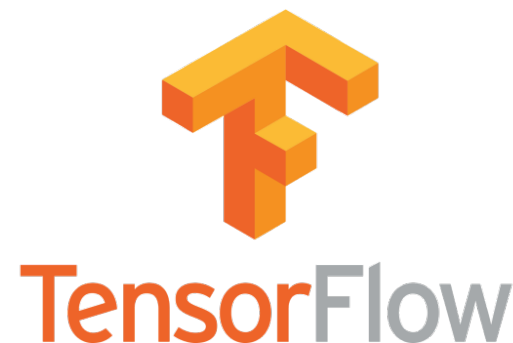


- **深度学习**

- Python的框架TensorFlow、Keras、PyTorch等能够快速构建神经网络模型，实现图像识别、自然语言处理、语音识别、推荐系统等应用场景。

- **计算机视觉**

- Python的库OpenCV、Dlib、Pillow等提供了图像处理和分析的基础工具，能够进行图像分类、目标检测、人脸识别、光流分析等任务。



1. Python基础 — Python字符串



- 字符串 (String)

- Python的一种常用数据类型，可使用引号（ ' 或 " ）创建字符串

- 字符串的索引

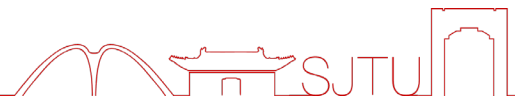
```
>>> var = 'Hello World'
>>> var[0:5]
'Hello'
```

- 字符串的加法

```
>>> 'Artificial' + "Intelligence"
'ArtificialIntelligence'
```

- 字符串的内置函数

```
>>> 'artificial'.upper()
'ARTIFICIAL'
>>> len('artificial')
10
```



1. Python基础 — Python列表



- 列表 (List)

- Python的一种基本数据结构，用方括号 “[]” 表示

- 列表的索引

```
>>> fruits = ['apple', 'orange', 'pear', 'banana']
>>> fruits[0]
'apple'
>>> fruits[-1]
'banana'
>>> fruits[1:3]
['orange', 'pear']
```

- 列表的加法

```
>>> otherFruits = ['kiwi', 'strewberry']
>>> fruits + otherFruits
['apple', 'orange', 'pear', 'banana', 'kiwi', 'strewberry']
```



1. Python基础 — Python列表



- 列表内置函数

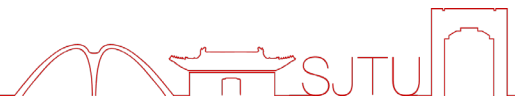
```
>>> fruits.pop()
'banana'
>>> fruits
['apple', 'orange', 'pear']
>>> fruits.append('pineapple')
>>> fruits
['apple', 'orange', 'pear', 'pineapple']
```

- 列表可以包含重复的元素

```
>>> lst = [1, 1, 1]
>>> lst
[1, 1, 1]
```

- 列表元素的删除

```
>>> lst = [1, 2, 3, 4]
>>> del lst[1]
>>> lst
[1, 3, 4]
```



1. Python基础 — Python元组



- 元组 (Tuple)
 - Python的一种基本数据结构，用圆括号 “ () ” 表示
- 元组的用法与列表很相似，区别在于**无法对元组进行修改**

```
>>> pair = (3, 5)
>>> pair[0]
3
>>> x, y = pair
>>> x
3
>>> y
5
>>> pair[1] = 6
Traceback (most recent call last):
  File "<pyshell#36>", line 1, in <module>
    pair[1] = 6
TypeError: 'tuple' object does not support item assignment
```



1. Python基础 — Python字典



▪ 字典 (Dictionary)

- Python的一种基本数据结构，用于存储键值对 (key-value pair)

```
>>> studentIds = {'knuth': 42.0, 'turing': 56.0, 'nash': 92.0}
>>> studentIds['turing']
56.0
>>> studentIds['nash'] = 'ninety-two'
>>> studentIds
{'knuth': 42.0, 'turing': 56.0, 'nash': 'ninety-two'}
>>> del studentIds['knuth']
>>> studentIds
{'turing': 56.0, 'nash': 'ninety-two'}
>>> studentIds['knuth'] = [42.0, 'forty-two']
>>> studentIds
{'turing': 56.0, 'nash': 'ninety-two', 'knuth': [42.0, 'forty-two']}
>>> studentIds.keys()
dict_keys(['turing', 'nash', 'knuth'])
>>> studentIds.values()
dict_values([56.0, 'ninety-two', [42.0, 'forty-two']])
>>> studentIds.items()
dict_items([('turing', 56.0), ('nash', 'ninety-two'), ('knuth', [42.0, 'forty-two'])])
>>> len(studentIds)
3
```



1. Python基础 — Python函数



▪ Python中，我们可以自定义函数

```
fruitPrices = {'apples': 2.00, 'oranges': 1.50, 'pears': 1.75}
```

```
def buyFruit(fruit, numPounds):  
    if fruit not in fruitPrices:  
        print("Sorry we don't have %s" % fruit)  
    else:  
        cost = fruitPrices[fruit] * numPounds  
        print("That'll be %f please" % cost)
```

```
# Main Function
```

```
if __name__ == '__main__':  
    buyFruit('apples', 2.4)  
    buyFruit('coconuts', 2)
```



1. Python基础 — Python类



- Python的类（class）定义了一个数据类型，包含一个或多个属性和方法

```
class FruitShop:
```

```
def __init__(self, name, fruitPrices):  
    """  
        name: Name of the fruit shop  
        fruitPrices: Dictionary with keys as fruit  
        strings and prices for values e.g.  
        {'apples':2.00, 'oranges': 1.50, 'pears': 1.75}  
    """  
    self.fruitPrices = fruitPrices  
    self.name = name  
    print('Welcome to %s fruit shop' % (name))
```

```
def getCostPerPound(self, fruit):  
    """  
        fruit: Fruit string  
        Returns cost of 'fruit', assuming 'fruit'  
        is in our inventory or None otherwise  
    """  
    if fruit not in self.fruitPrices:  
        print("Sorry we don't have %s" % (fruit))  
        return None  
    return self.fruitPrices[fruit]
```



1. Python基础 — Python类



▪ Python类的使用

```
from shop import FruitShop

if __name__ == '__main__':
    shopName = 'the Berkeley Bowl'
    fruitPrices = {'apples': 1.00, 'oranges': 1.50, 'pears': 1.75}
    berkeleyShop = FruitShop(shopName, fruitPrices)
    applePrice = berkeleyShop.getCostPerPound('apples')
    print('Apples cost $%.2f at %s.' % (applePrice, berkeleyShop.getName()))

    otherName = 'the Stanford Mall'
    otherFruitPrices = {'kiwis': 6.00, 'apples': 4.50, 'peaches': 8.75}
    otherFruitShop = FruitShop(otherName, otherFruitPrices)
    otherPrice = otherFruitShop.getCostPerPound('apples')
    print('Apples cost $%.2f at %s.' % (otherPrice, otherFruitShop.getName()))
```

```
Run - P0
Run: shopTest x
D:\conda\python.exe C:/Users/lenovo/Desktop
Welcome to the Berkeley Bowl fruit shop
Apples cost $1.00 at the Berkeley Bowl.
Welcome to the Stanford Mall fruit shop
Apples cost $4.50 at the Stanford Mall.

Process finished with exit code 0
```



目 录

1

Python基础

2

软件安装与使用

3

Python实践



2. 软件安装与使用 — Python安装

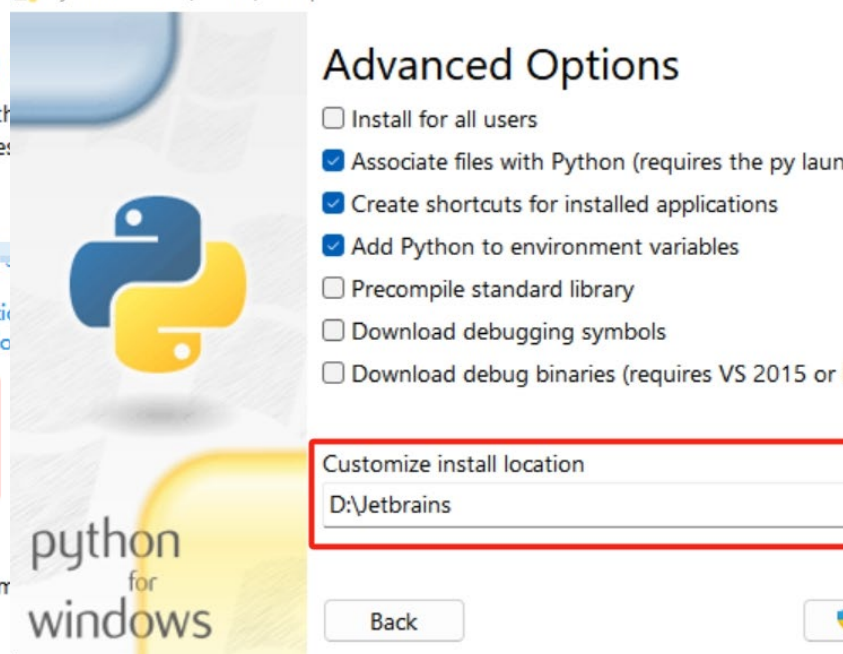


- 访问 <https://www.python.org/downloads/windows/>
- 使用Python安装向导安装，安装过程中需要将Python加入环境变量
- 创建D:\Jetbrains文件夹，将Python安装在此文件夹中 (3.8.10)

Python 3.8.10 (64-bit) Setup



Python 3.8.10 (64-bit) Setup



Python 3.9.5 - May 3, 2021

Note that Python 3.9.5 cannot be used on Windows 7 or earlier.

- Download [Windows embeddable package \(32-bit\)](#)
- Download [Windows embeddable package \(64-bit\)](#)
- Download [Windows help file](#)
- Download [Windows installer \(32-bit\)](#)
- Download [Windows installer \(64-bit\)](#)

Python 3.8.10 - May 3, 2021

Note that Python 3.8.10 cannot be used on Windows XP or earlier.

- Download [Windows embeddable package \(32-bit\)](#)
- Download [Windows embeddable package \(64-bit\)](#)
- Download [Windows help file](#)
- Download [Windows installer \(32-bit\)](#)
- Download [Windows installer \(64-bit\)](#)

Python 3.9.4 - April 4, 2021

Note that Python 3.9.4 cannot be used on Windows 7 or earlier.

- Download [Windows embeddable package \(32-bit\)](#)
- Download [Windows embeddable package \(64-bit\)](#)
- Download [Windows help file](#)



2. 软件安装与使用 — PyCharm安装与使用

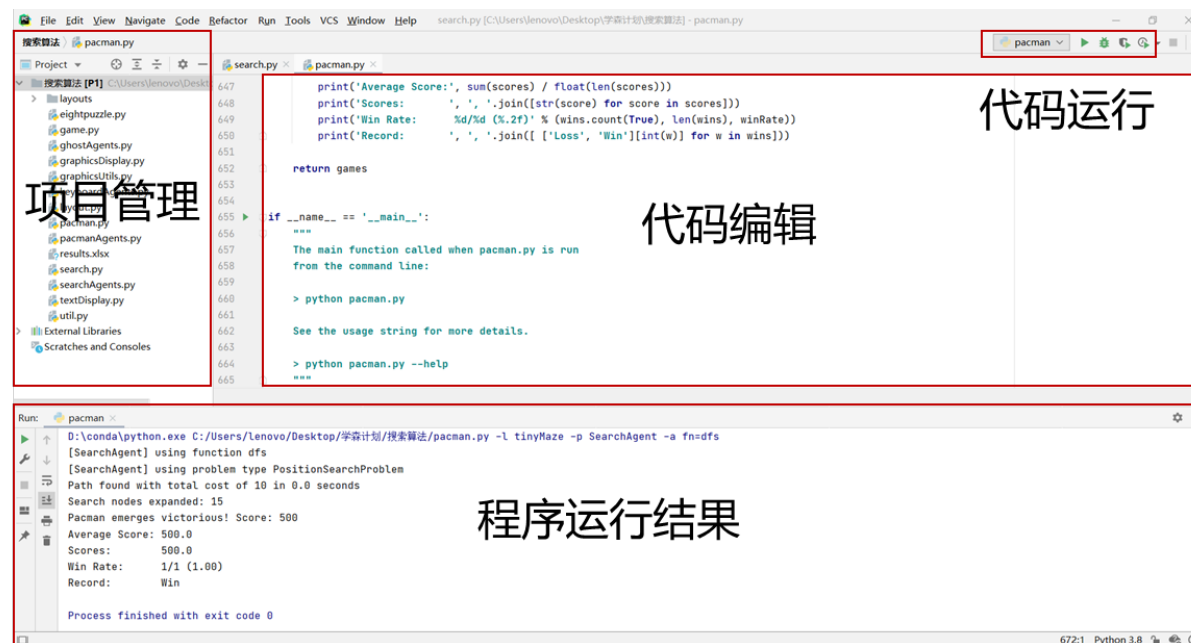


▪ PyCharm功能

- 调试、语法高亮、Project管理、代码跳转、智能提示等

▪ PyCharm下载

- <https://www.jetbrains.com.cn/pycharm/>, PyCharm官网下载



2. 软件安装与使用 — PyCharm安装与使用



- 在PyCharm官网上选择社区版进行下载

PyCharm

用于数据科学和 Web 开发的 Python IDE

下载

成熟的 Professional Edition 或免费的
Community Edition



PyCharm Community Edition

适用于纯 Python 开发的 IDE

下载

.exe ▼

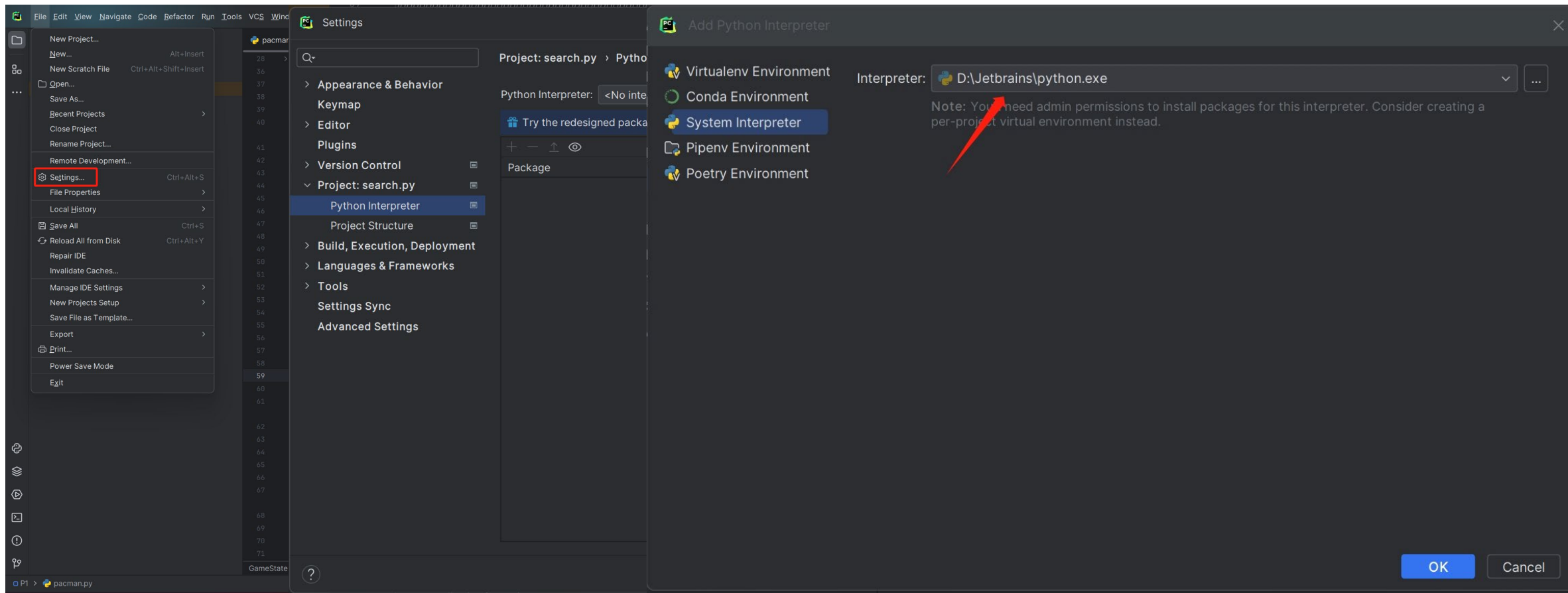
免费，开源构建



2. 软件安装与使用 — PyCharm安装与使用



- 安装后打开文件夹，点击右上角设置，选择安装的python环境



目 录

1

Python基础

2

软件安装与使用

3

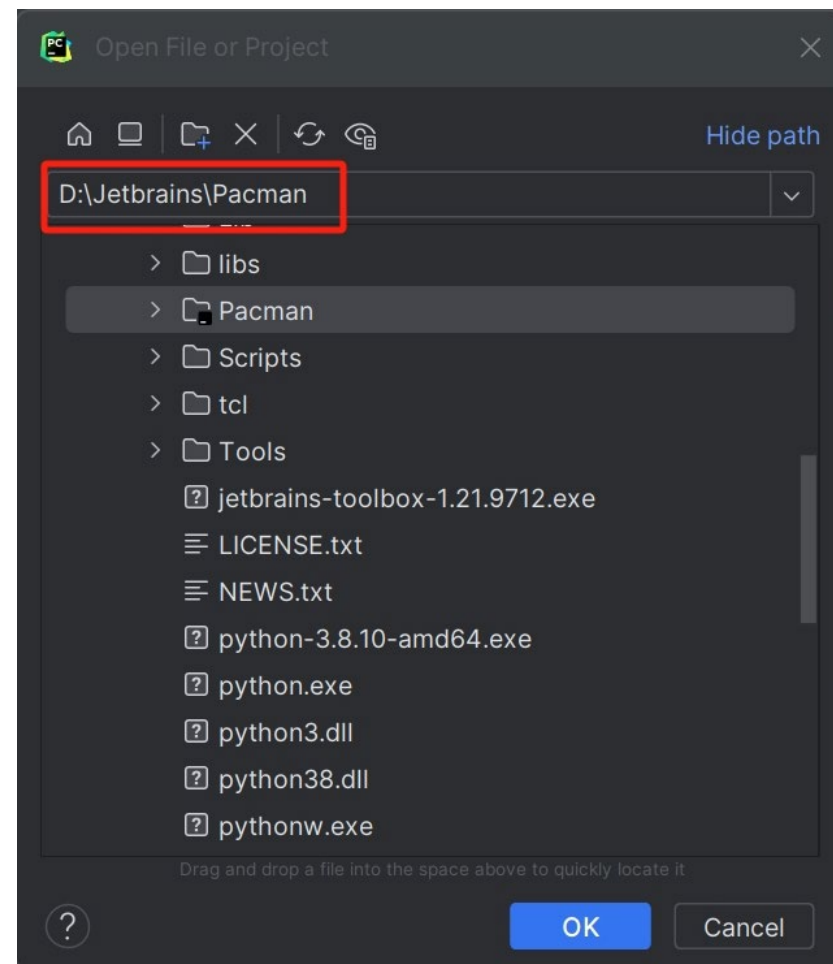
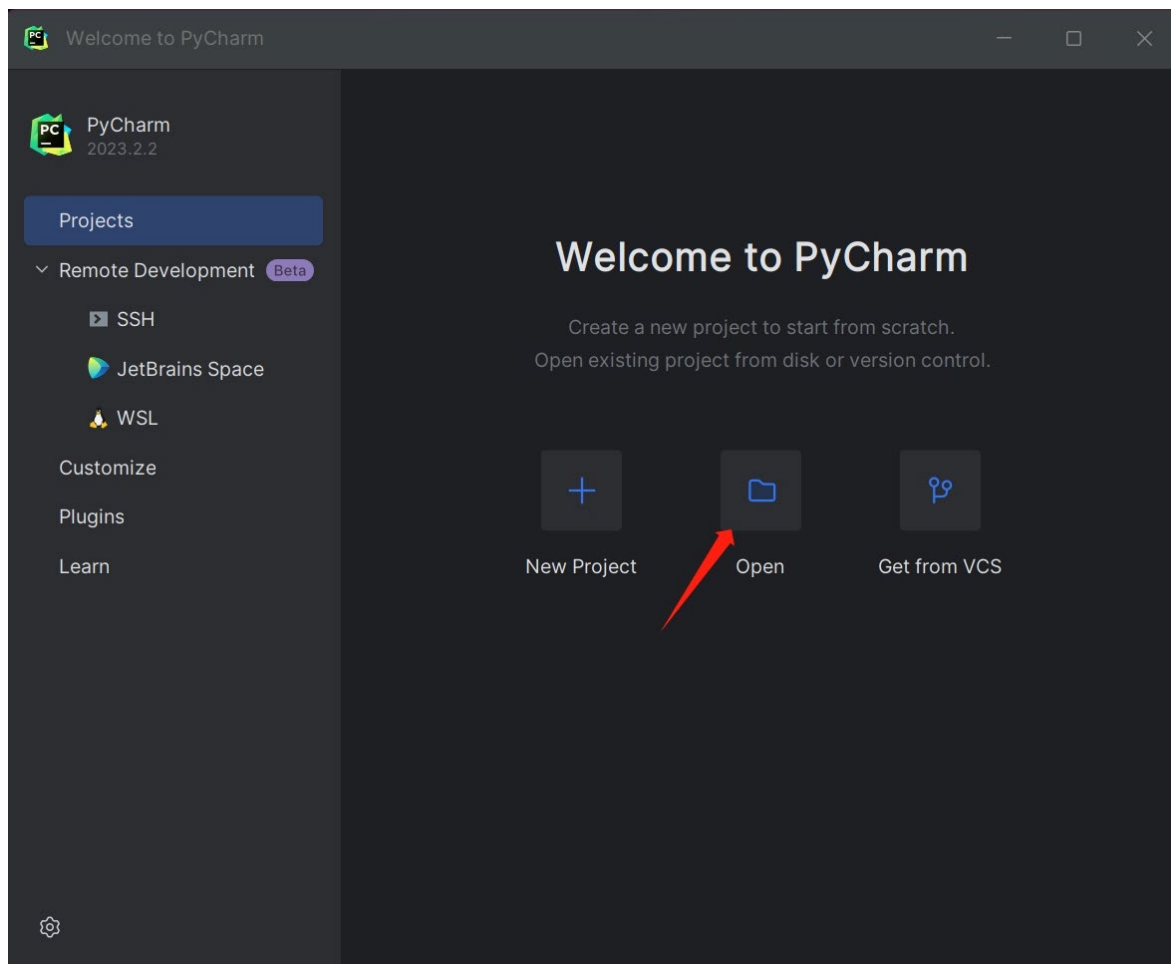
Python实践



3. Python实践 — Python在吃豆人中的应用



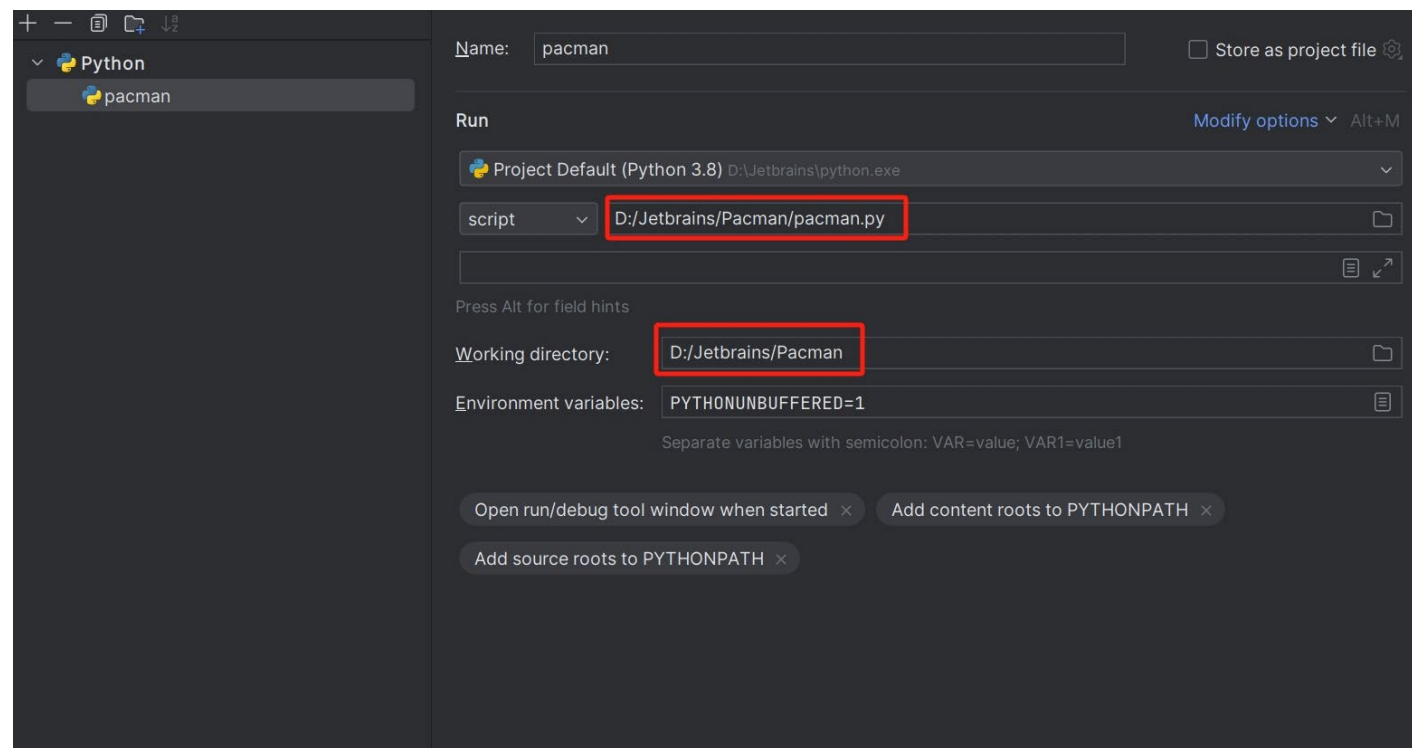
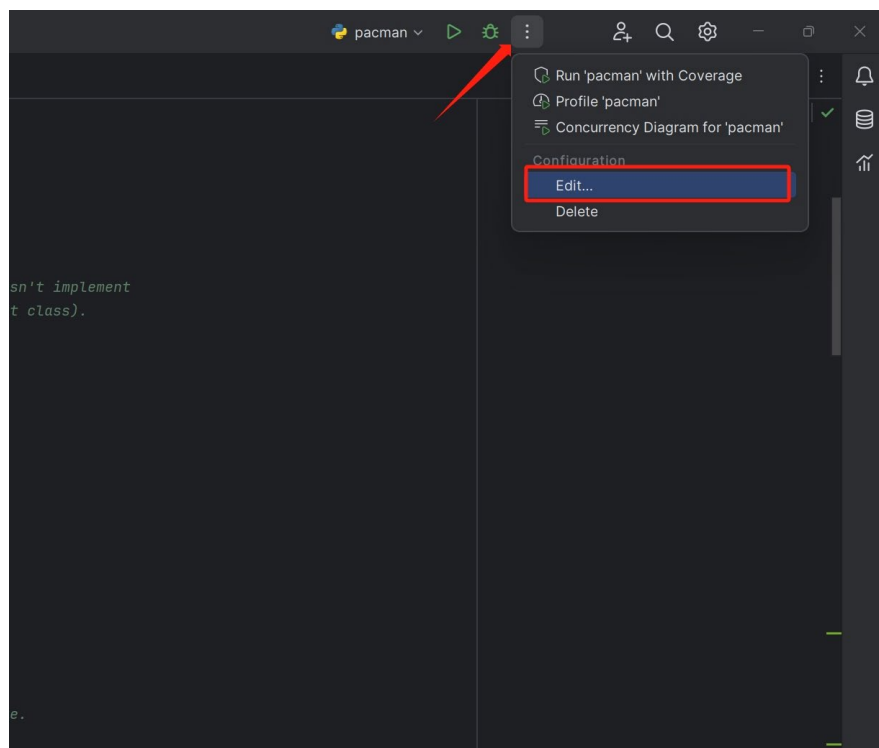
- 下载代码库，解压，放到 D:\Jetbrains\Pacman 路径
- 使用PyCharm打开D:\Jetbrains\Pacman文件夹，并打开pacman.py文件



3. Python实践 — Python在吃豆人中的应用



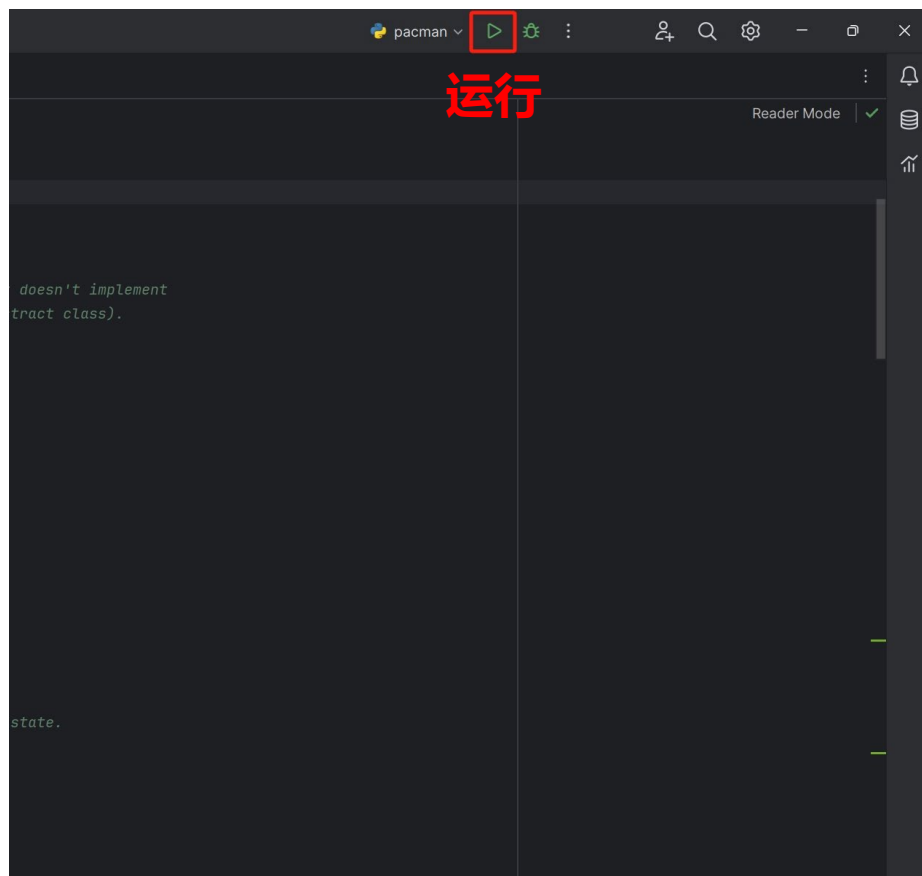
- 点击Run – Edit Configurations – Add new – Python
- 在Script path处写入pacman.py的存储地址
- 在Working directory处写入D:\Jetbrains\Pacman文件夹的地址



3. Python实践 — Python在吃豆人中的应用



- 点击Apply, 然后运行程序, 会弹出吃豆人窗口





上海交通大学

SHANGHAI JIAO TONG UNIVERSITY



人工智能研究院

Artificial Intelligence Institute

谢谢!

饮水思源 爱国荣校