走进Python的世界

实验指导手册

版本:1.5

华为技术有限公司

商标声明


和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编：518129

网址： https://edu.huaweicloud.com/

目录

背景介绍

Python 是一个高层次的结合了解释性、编译性、互动性和面向对象的脚本语言，由于其简单的语法、强大的工具库等优势被很多人青睐，应用于各个行业之中。

实验目的

本实验指导主要实现 Python 环境搭建和了解 Python 基础语法的使用。通过本实验，您将能够：

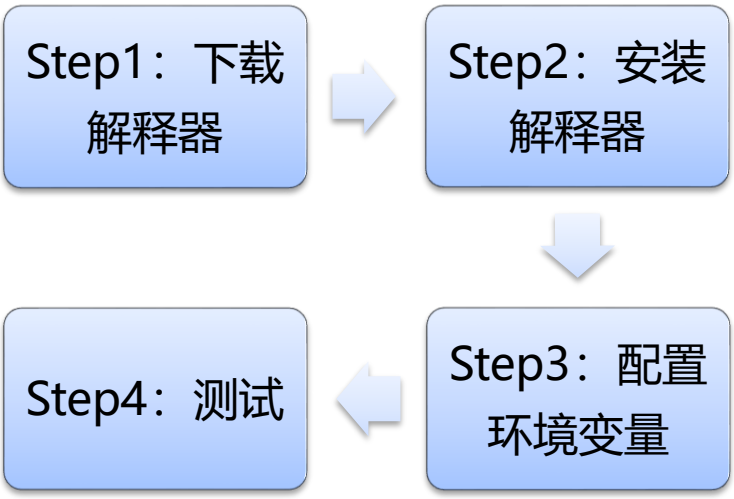
- 搭建自己的 Python 开发环境
- 了解 Python 基础语法规则

1 Python 环境安装

1.1 实验介绍

通过如下步骤下载Python解释器并进行安装。

1.2 实验流程



1.3 资源准备

下载Python解释器（此处Windows为例：https://www.python.org/downloads/windows/）。

- Python 3.7.9 - Aug. 17, 2020

  Note that Python 3.7.9 *cannot* be used on Windows XP or earlier.

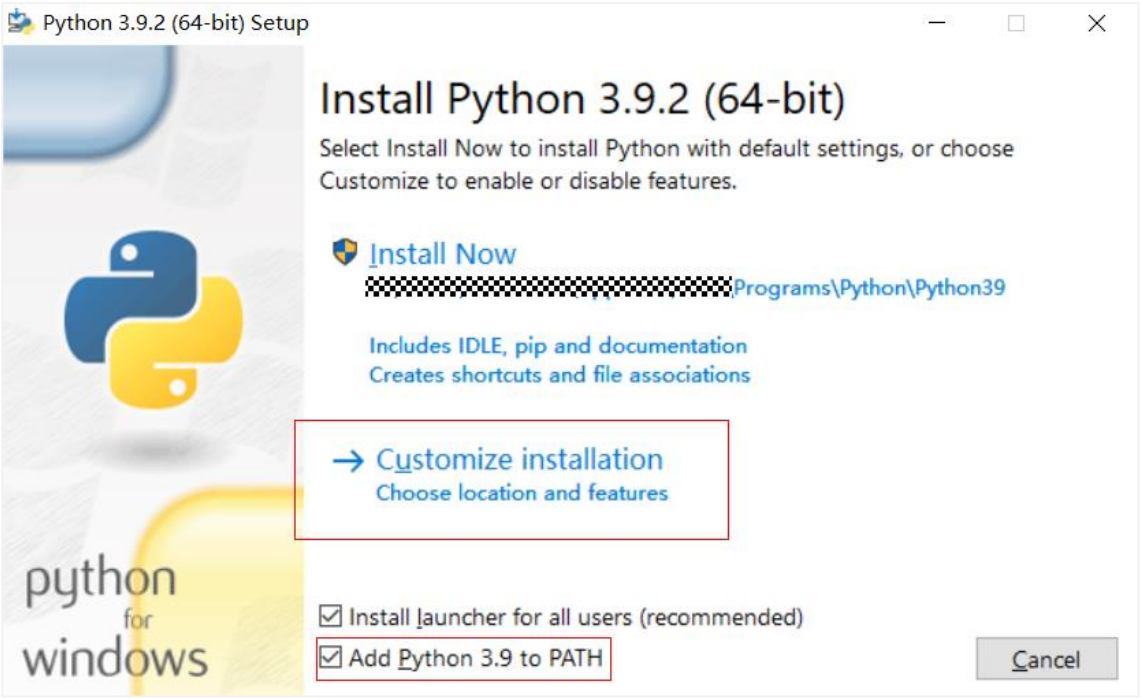
  - Download Windows help file
  - Download Windows x86-64 embeddable zip file
  - Download Windows x86-64 executable installer
  - Download Windows x86-64 web-based installer
  - Download Windows x86 embeddable zip file
  - Download Windows x86 executable installer
  - Download Windows x86 web-based installer

此处根据自己电脑选择下载可执行文件：

- Download Windows x86-64 executable installer

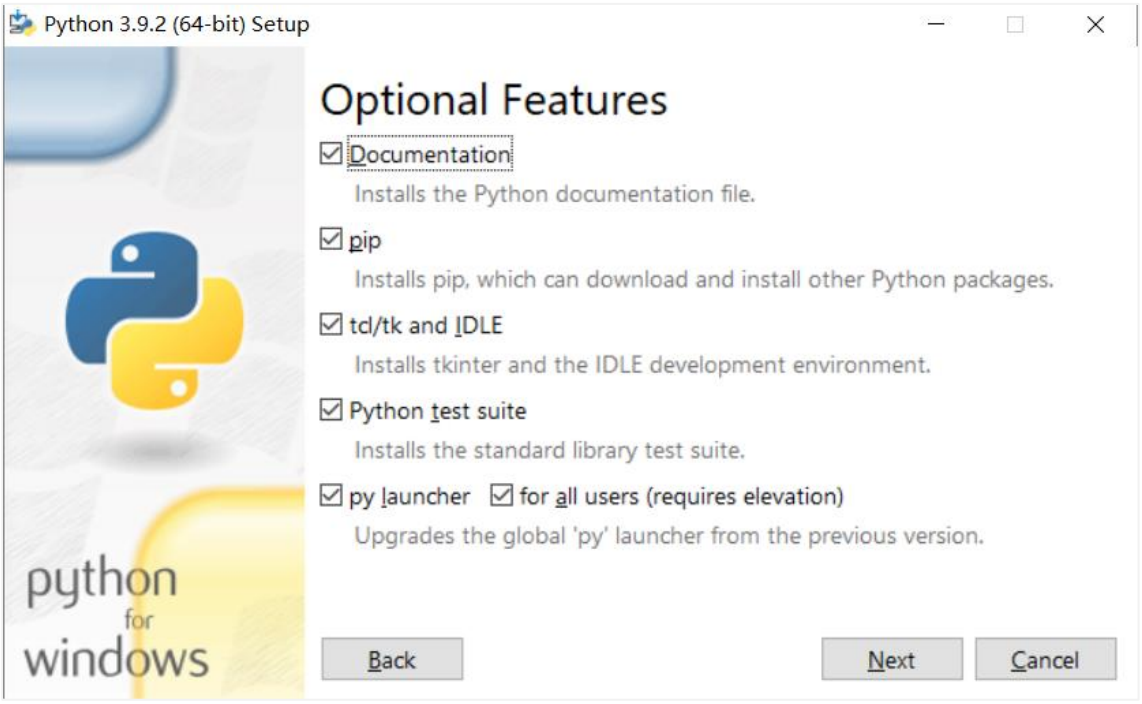
想安装 Python 解释器

步骤 1　点击安装文件进行安装。

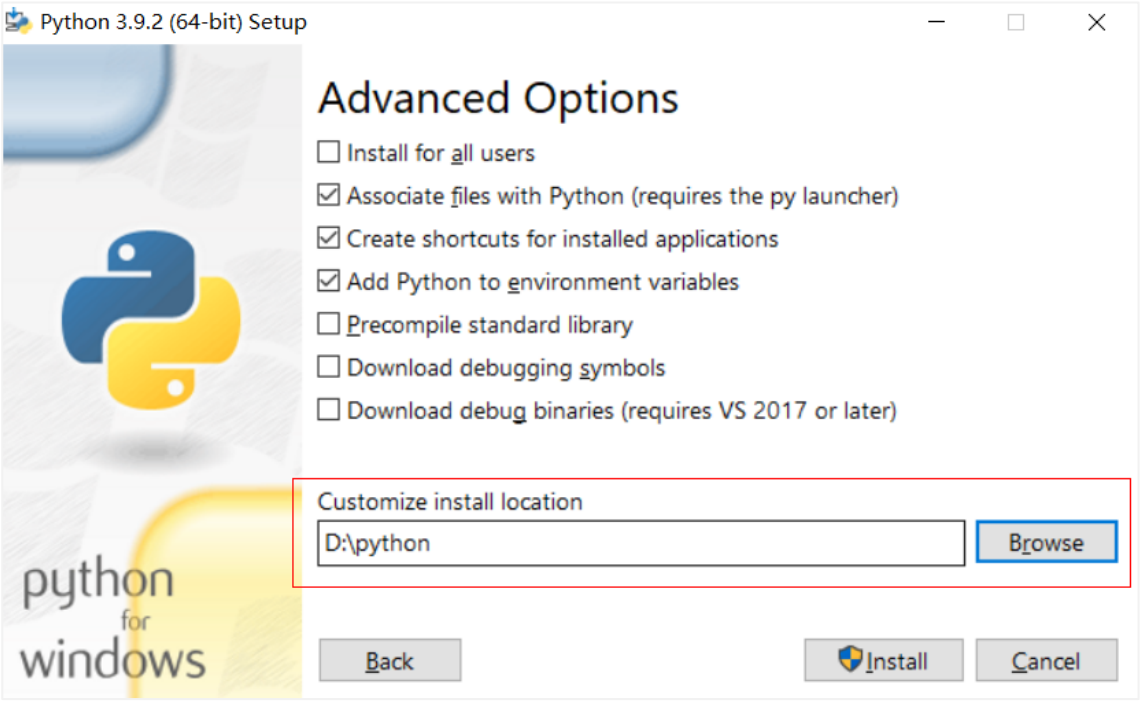


勾选添加路径。

选择自定义安装：



选择安装路径：

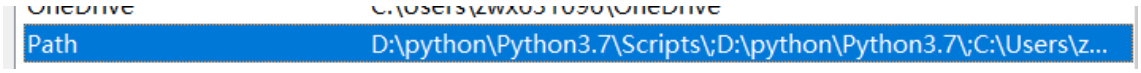


点击安装。

步骤 2　配置环境变量（如果安装时勾选，则忽略此步骤）

右击我的电脑，选择属性，点击高级系统设置：

选中变量中的 path：



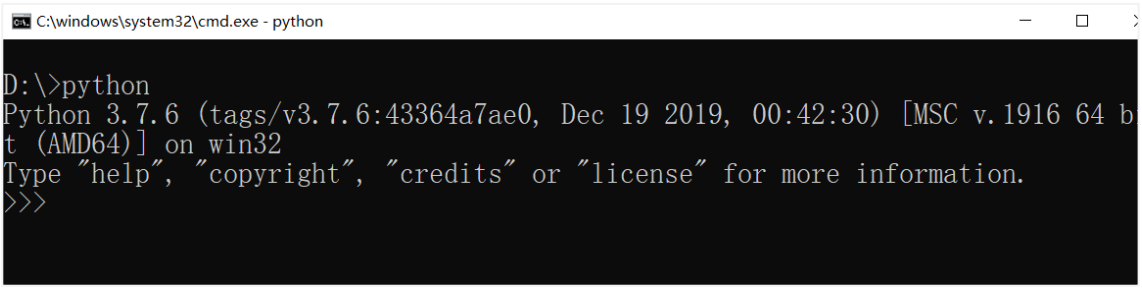
添加变量：

```
D:\python\Python3.9\Scripts\
D:\python\Python3.9\
```

此处路径为 Python 的安装路径和下面的 script 文件夹（pip）的路径。

步骤 3　测试是否安装成功

打开 CMD 命令行，输入 Python：



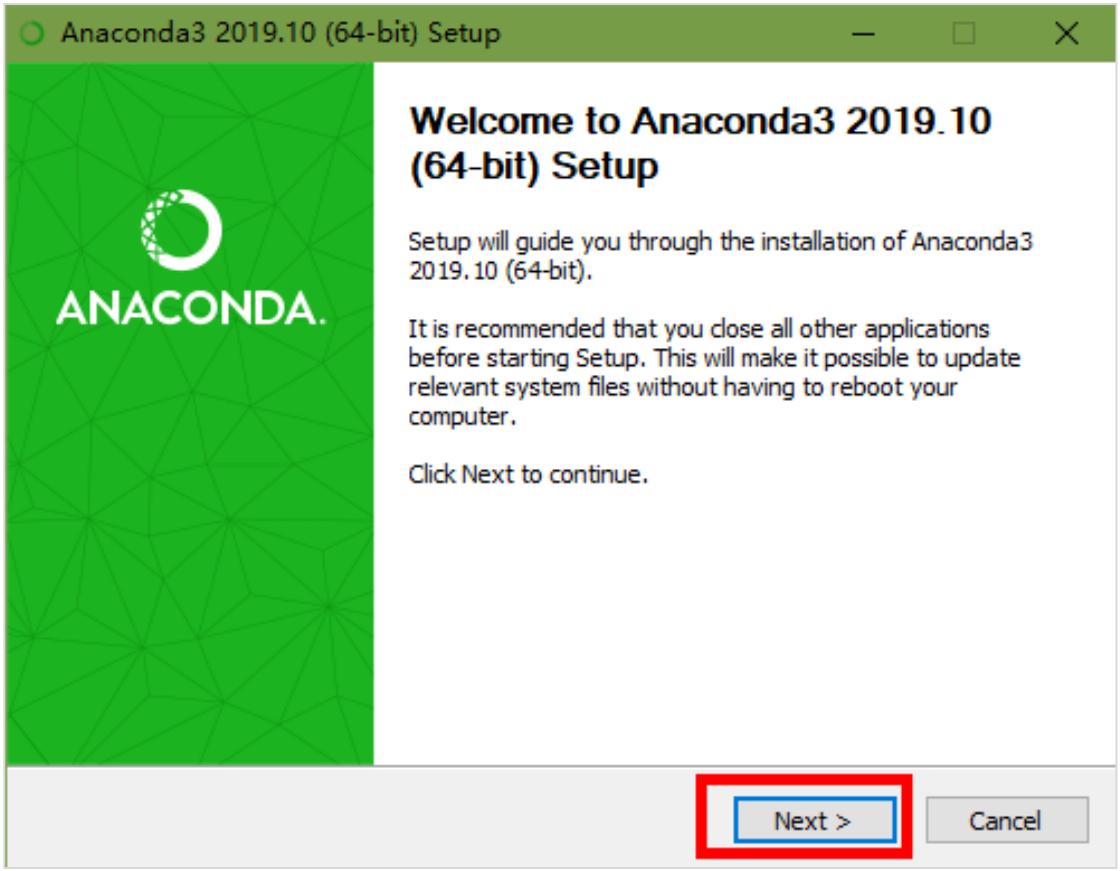
进入 Python 环境则安装成功。

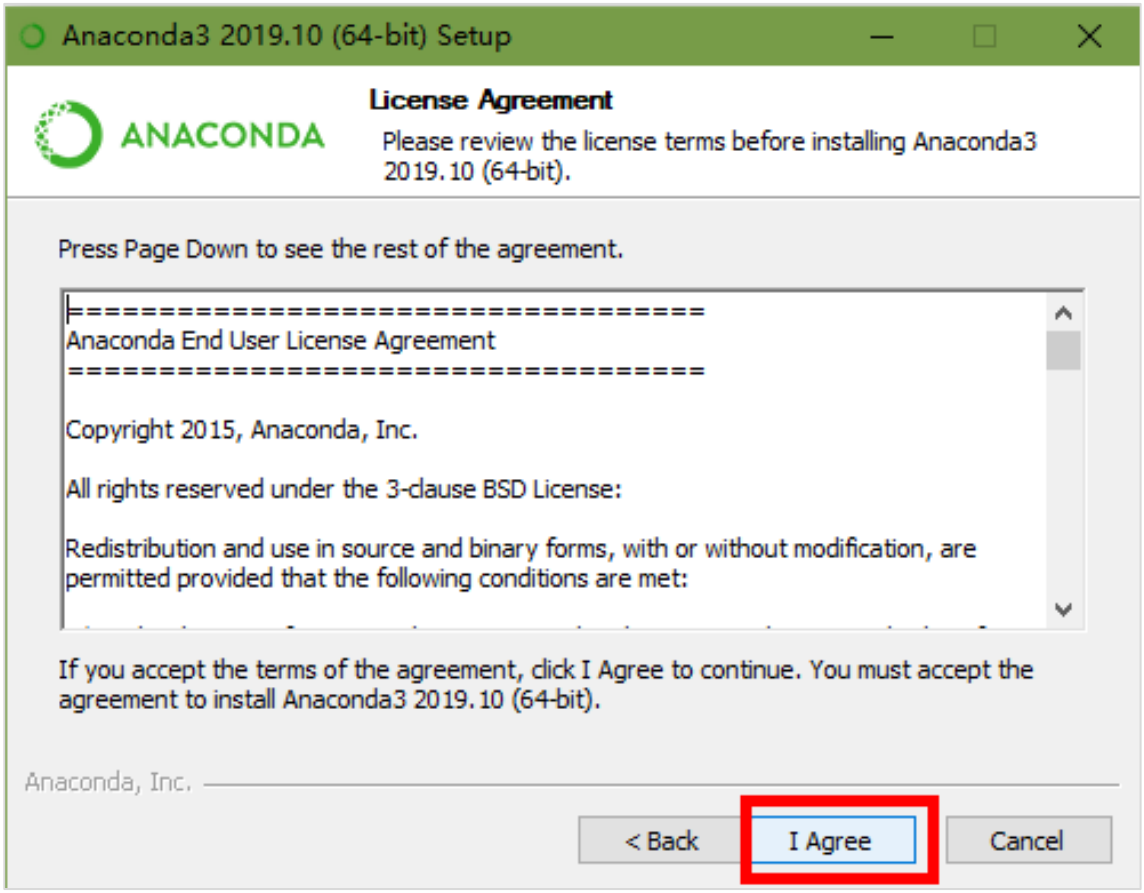
（如果安装多个版本的 Python，会根据环境变量的位置进入，可以进入不同版本的 Python 文件中，将 Python.exe 文件，创建快捷方式后重命名，如，改为：python3.7，后续可输入 Python3.7 进入环境。）

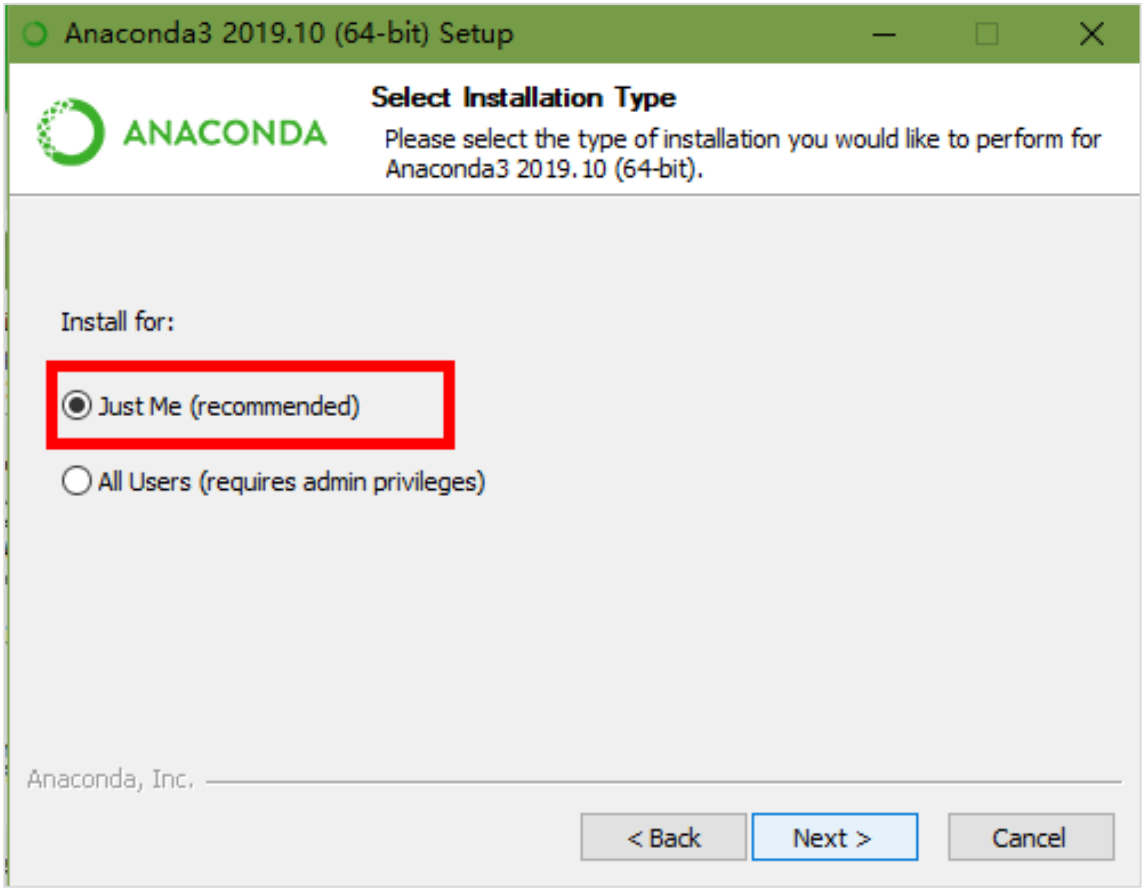
1.4 安装 Anaconda

步骤 1　打开浏览器，输入网址 https://www.anaconda.com/products/individual 根据您的系统位数下载 32 位/64 位的安装包。

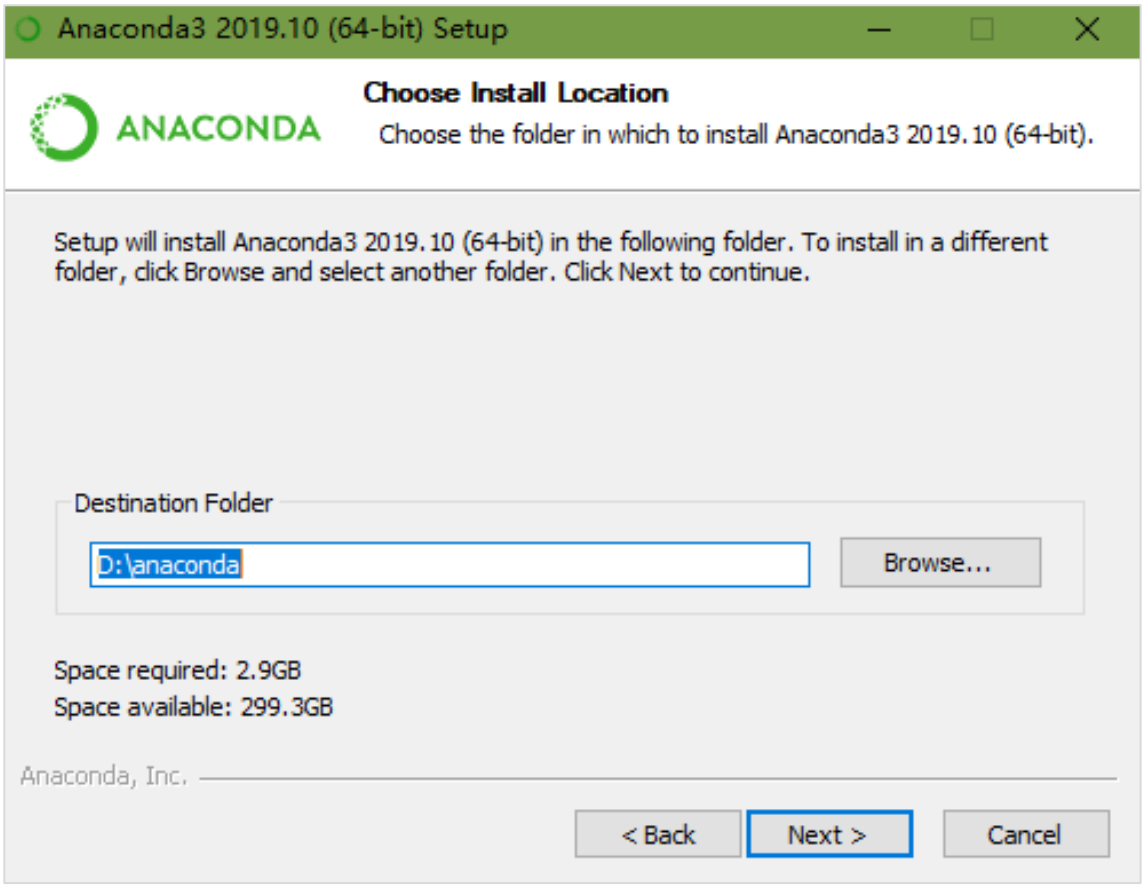
步骤 2　安装 Anaconda 软件。


Anaconda3 2019.10 (64-bit) Setup
ANACONDA.
Welcome to Anaconda3 2019.10 (64-bit) Setup
Setup will guide you through the installation of Anaconda3 2019.10 (64-bit).
It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.
Click Next to continue.
Next >
Cancel

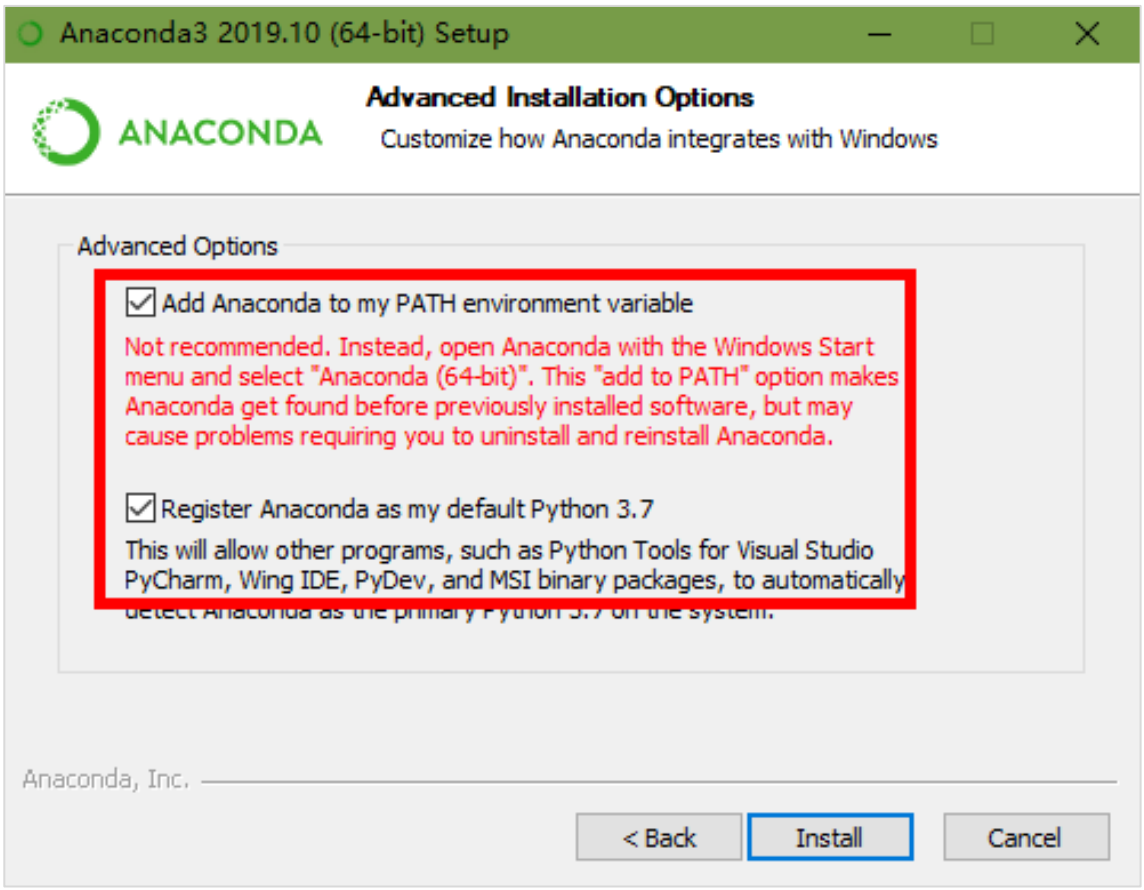

Anaconda3 2019.10 (64-bit) Setup
ANACONDA
License Agreement
Please review the license terms before installing Anaconda3 2019.10 (64-bit).
Press Page Down to see the rest of the agreement.
Anaconda End User License Agreement
Copyright 2015, Anaconda, Inc.
All rights reserved under the 3-clause BSD License:
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Anaconda3 2019.10 (64-bit).
Anaconda, Inc.
< Back
I Agree
Cancel


Anaconda3 2019.10 (64-bit) Setup
ANACONDA
Select Installation Type
Please select the type of installation you would like to perform for Anaconda3 2019.10 (64-bit).
Install for:
Just Me (recommended)
All Users (requires admin privileges)
Anaconda, Inc.
< Back
Next >
Cancel

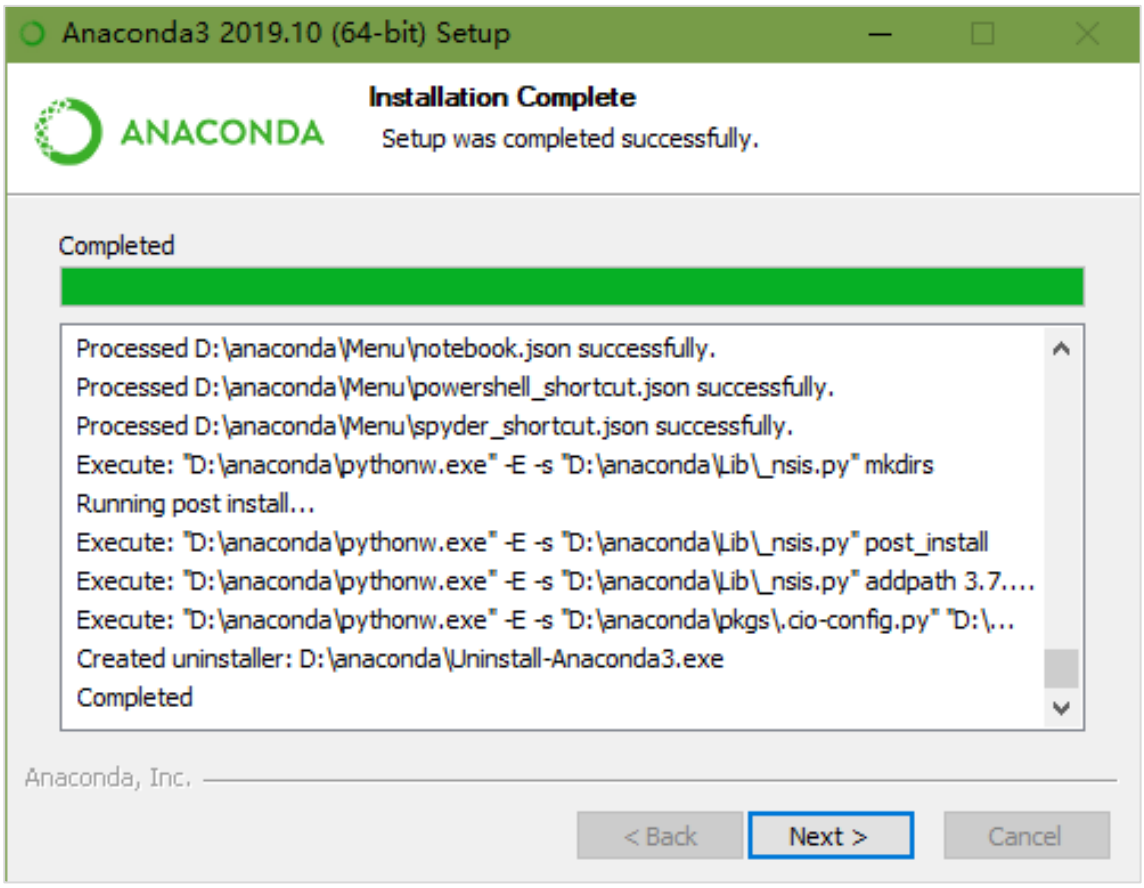
步骤 3　选择安装路径（建立纯英文路径）。



步骤 4　Options 选项中全部选择添加（确保软件环境的完整）。


Anaconda3 2019.10 (64-bit) Setup
ANACONDA
Advanced Installation Options
Customize how Anaconda integrates with Windows
Advanced Options
Add Anaconda to my PATH environment variable
Not recommended. Instead, open Anaconda 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.
Register Anaconda as my default Python 3.7
This will allow other programs, such as Python Tools for Visual Studio PyCharm, Wing IDE, PyDev, and MSI binary packages, to automatically
Anaconda, Inc.
< Back
Install
Cancel

步骤 5　完成安装。



步骤 6　安装完成后查看安装结果

点击此图标可打开 anaconda：

点击此图标，打开 jupyter 环境。

点击此图标，打开命令行安装工具：

1.5 虚拟环境

步骤 1　在 Python 解释器中安装工具包

在 CMD 命令行中输入：

```
pip install virtualenv
pip install virtualenvwrapper-win
```

(注：在 windows 安装 virtualenvwrapper-win，在 linux 则为 virtualenvwrapper)

安装完成后可以通过命令创建虚拟环境：

```
mkvirtualenv env
```

也可以指定虚拟机的 python 版本：

```
mkvirtualenv env -p 已安装的 Python 解释器的位置
```

启动虚拟环境：

```
workon env
```

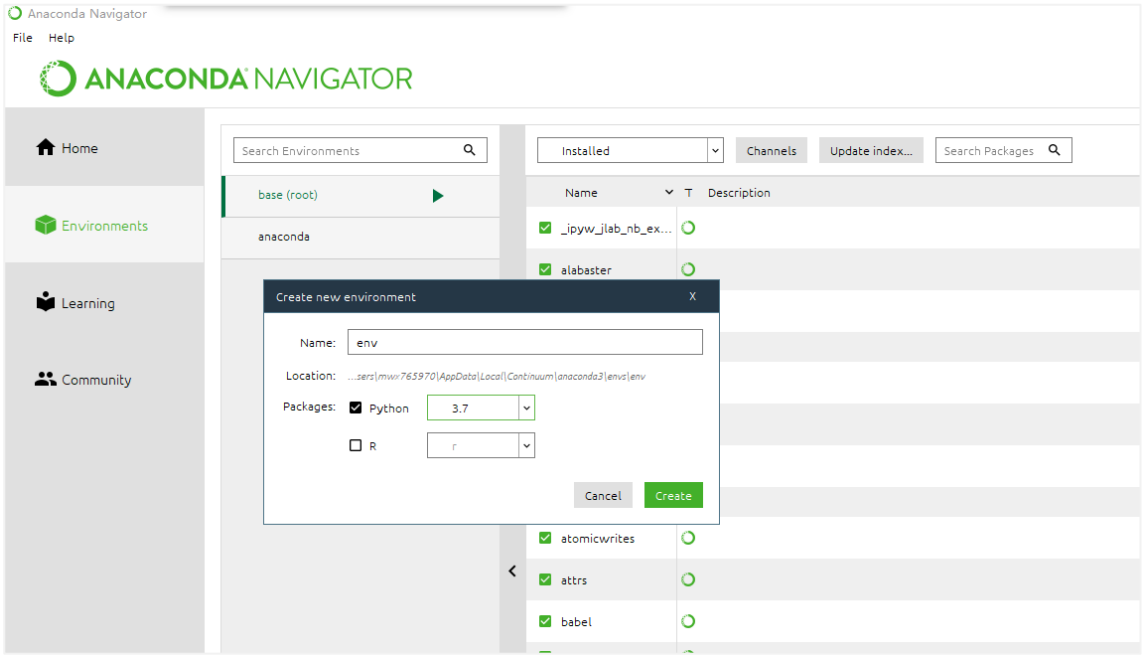
退出虚拟环境：

```
Deactivate
```

删除虚拟环境：

```
rmvirtualenv env
```

步骤 2　Anaconda 中的虚拟环境



进入 anaconda 后，点击 Environments，点击 create，选择 Python 版本和虚拟环境的名称。

1.6 开发工具安装

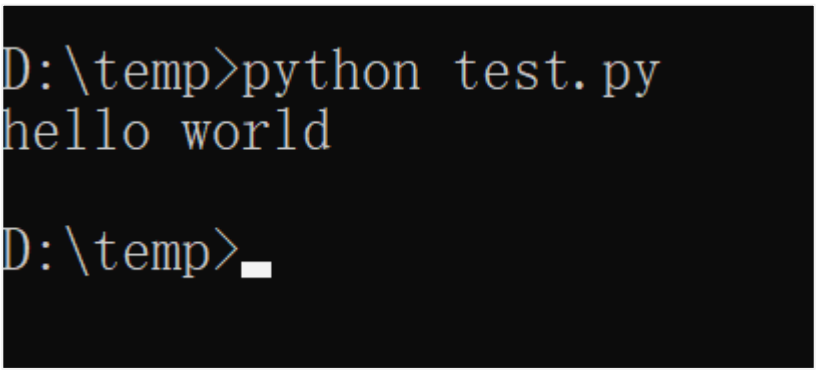
步骤 1　记事本工具

新建文本文件，更改后缀名称为.py，test.py，在文件中输入代码：

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

打开 test.py 所在文件夹，输入 CMD 进入命令行：

输入 python test.py（虚拟环境需要先进入对应环境）：

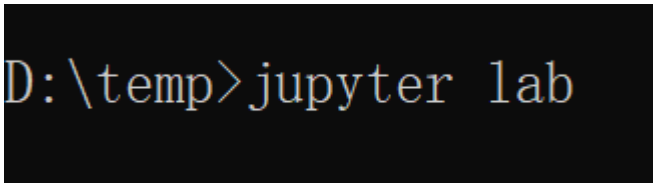


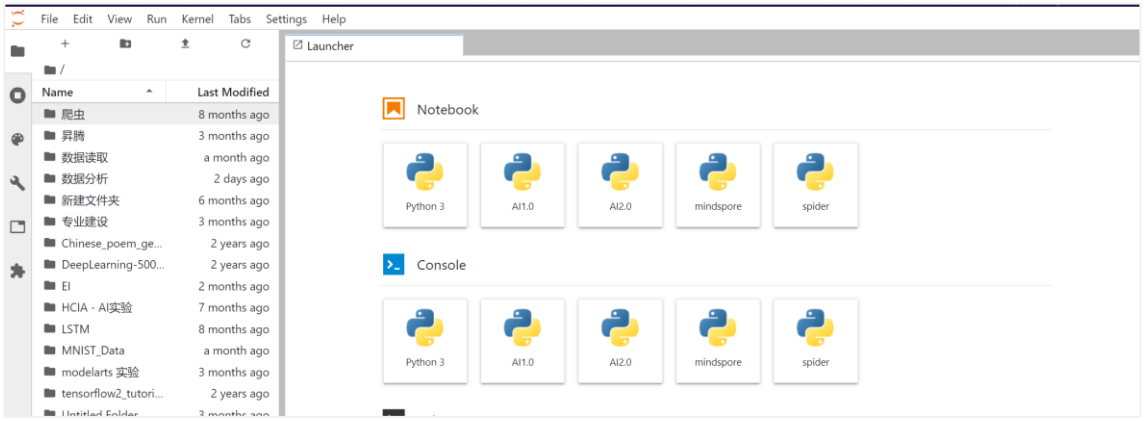
步骤 2　Jupyter lab

安装 jupyter lab：在 anaconda 中可以直接在 HOME 中点击对应图标启动；在 Python 解释器中需要先进行安装，打开命令行，使用 pip 命令安装。

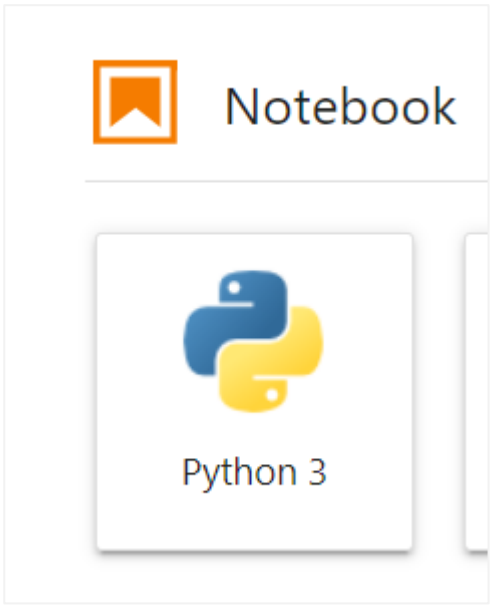
```
pip install jupyterlab
```

安装完成后，在命令行中输入 jupyter lab 即可启动。





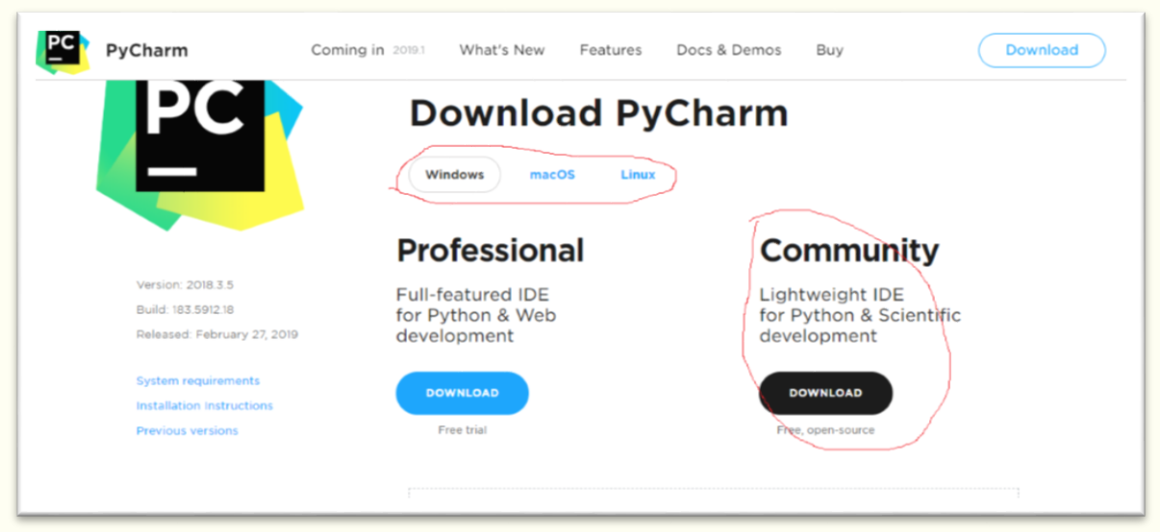
点击图标即可创建对应环境的 notebook 文件：



创建完成后即可编写代码。

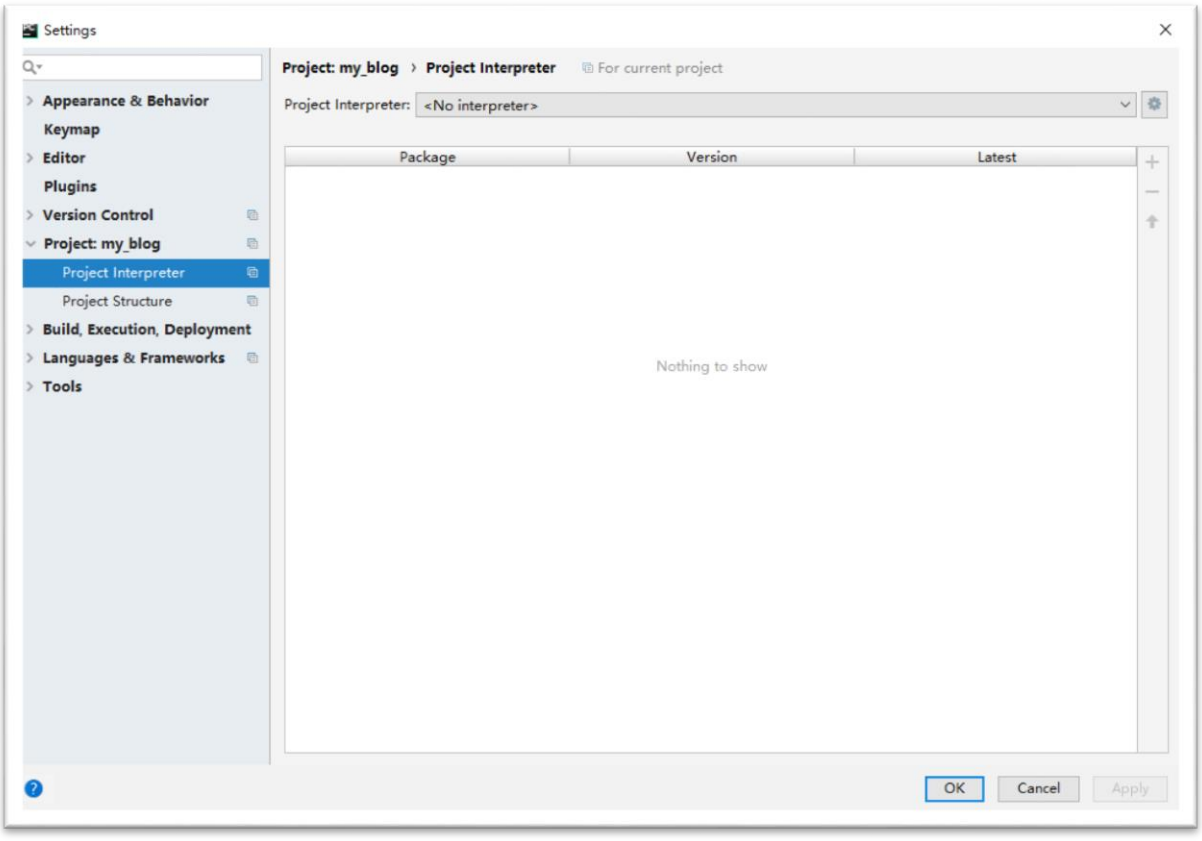
步骤 3　Pycharm

下载 pycharm 官网地址：https://www.jetbrains.com/pycharm/download



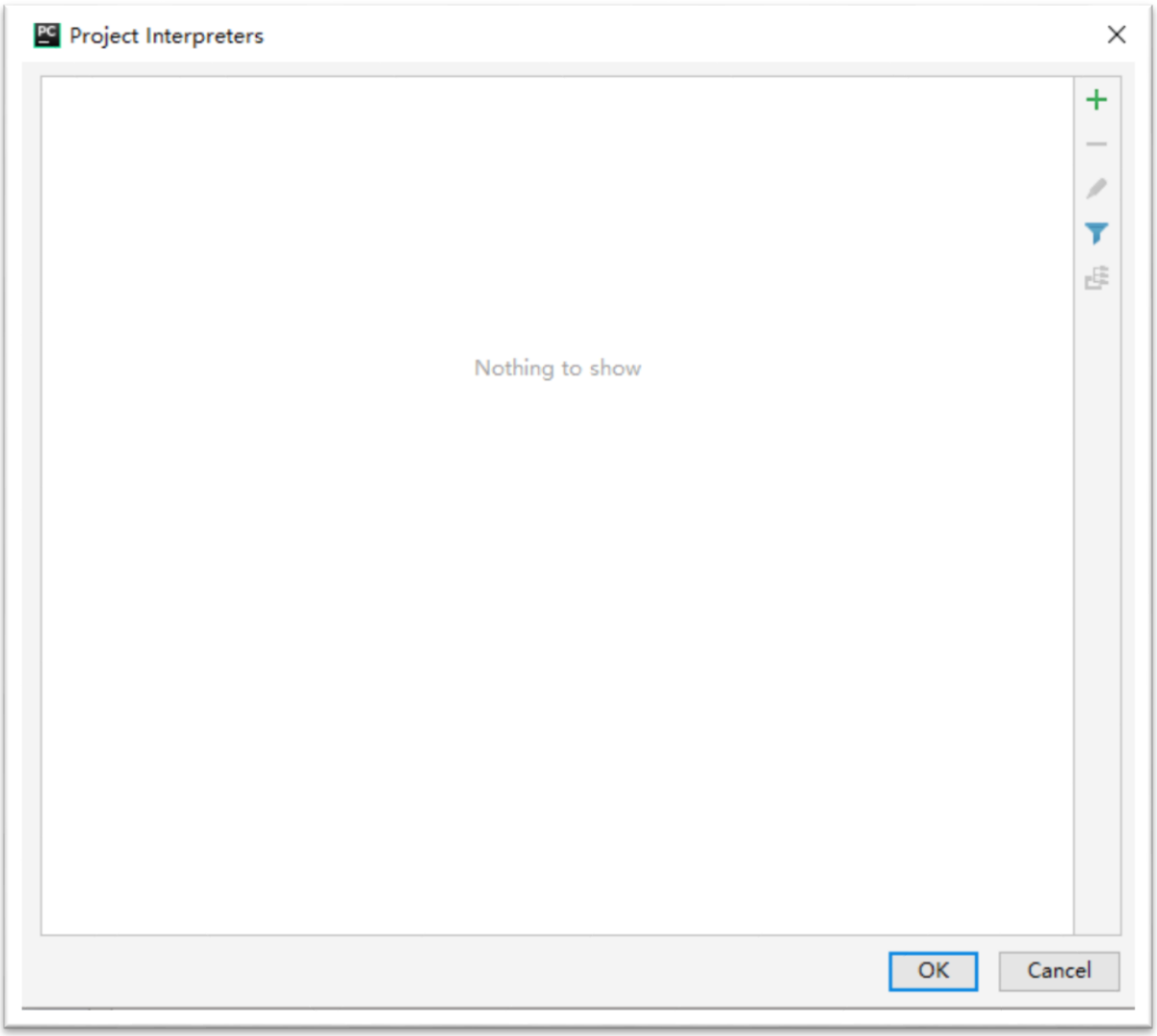
双击 exe 文件进行安装。

打开工程文件夹后点击左上角的 file，选择 settings 选项。

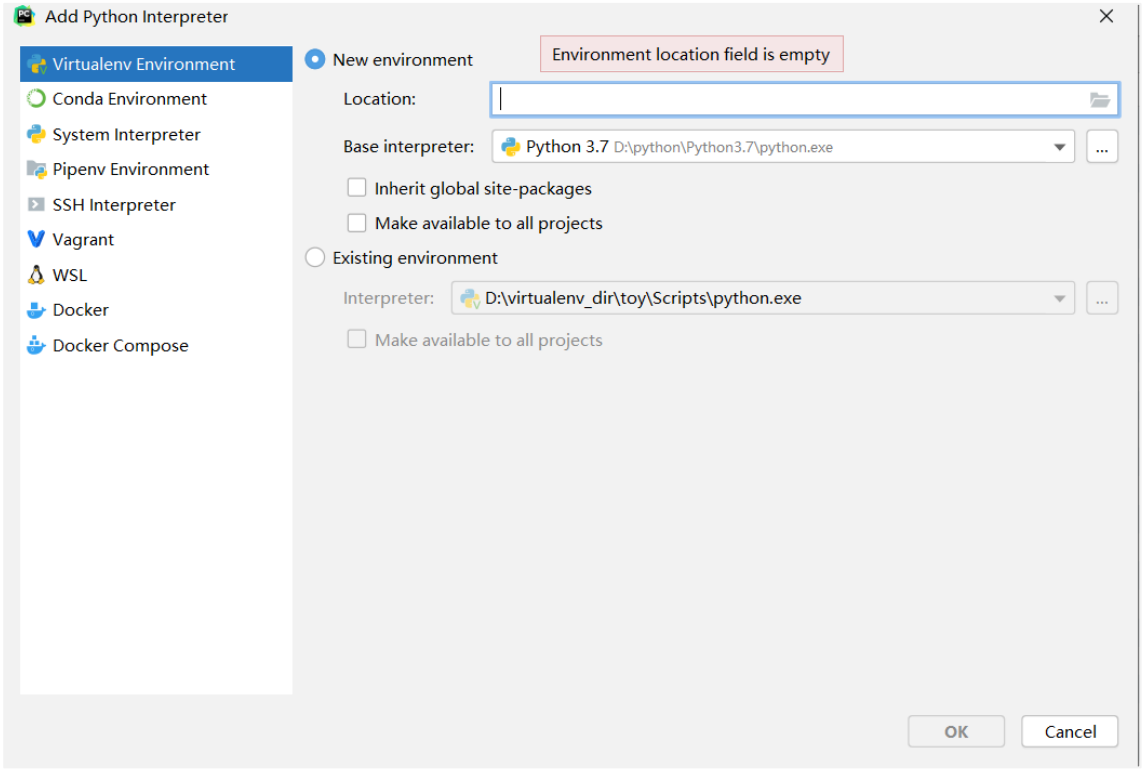


点击 project interpretr。

选择 show all：

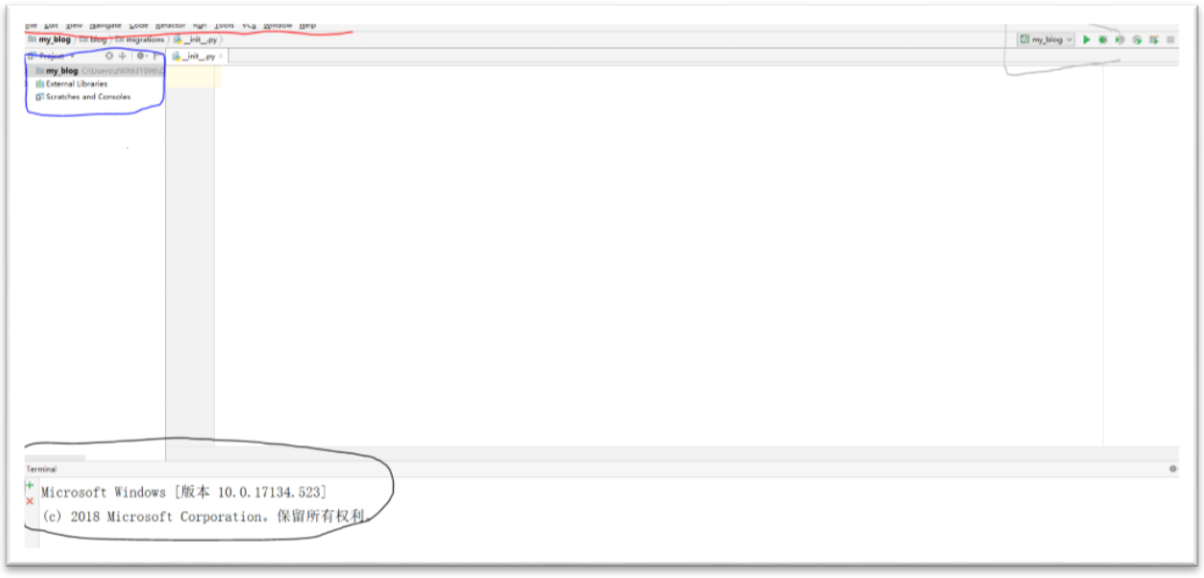


点击右边的+。



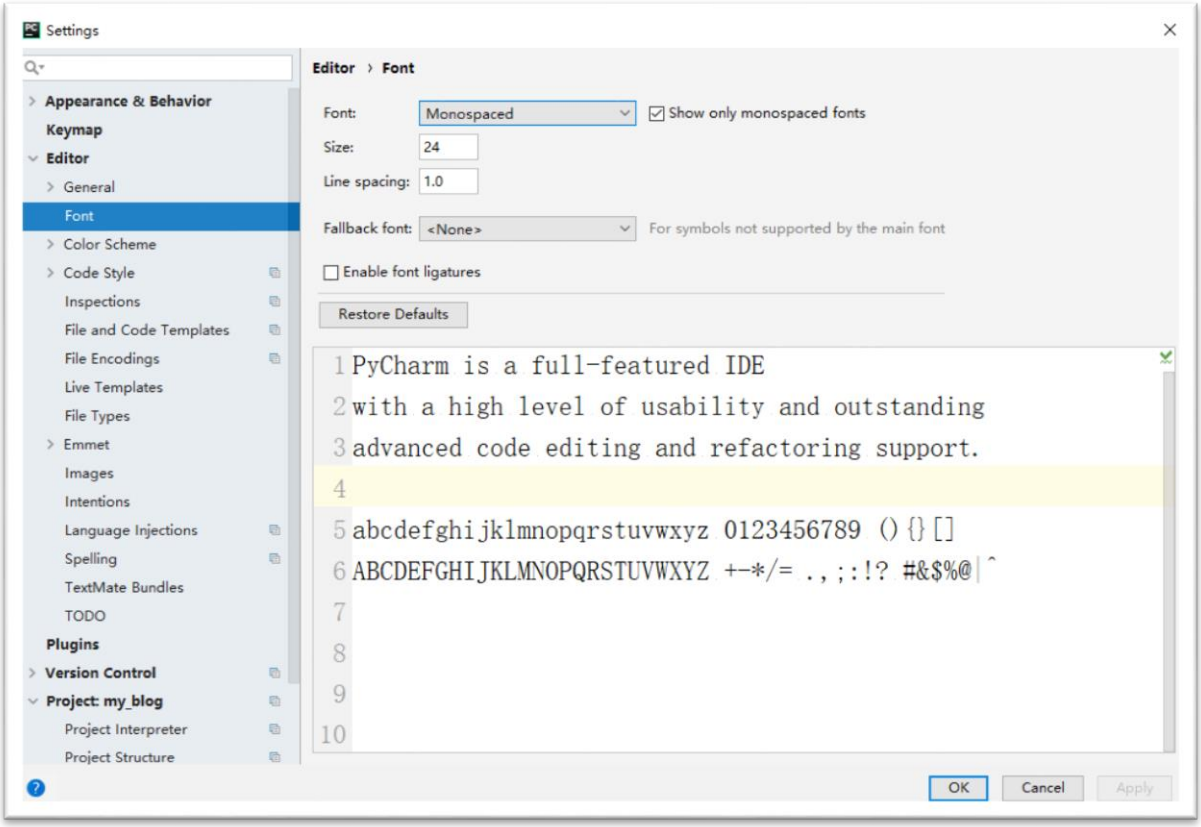
选择本机对应的环境即可。

Pycharm 的结构：



红色划线部分是工具栏，蓝色部分是项目的结构，下面黑色部分是控制台，右面灰色部分是程序的运行和调试按钮。

设置字体：



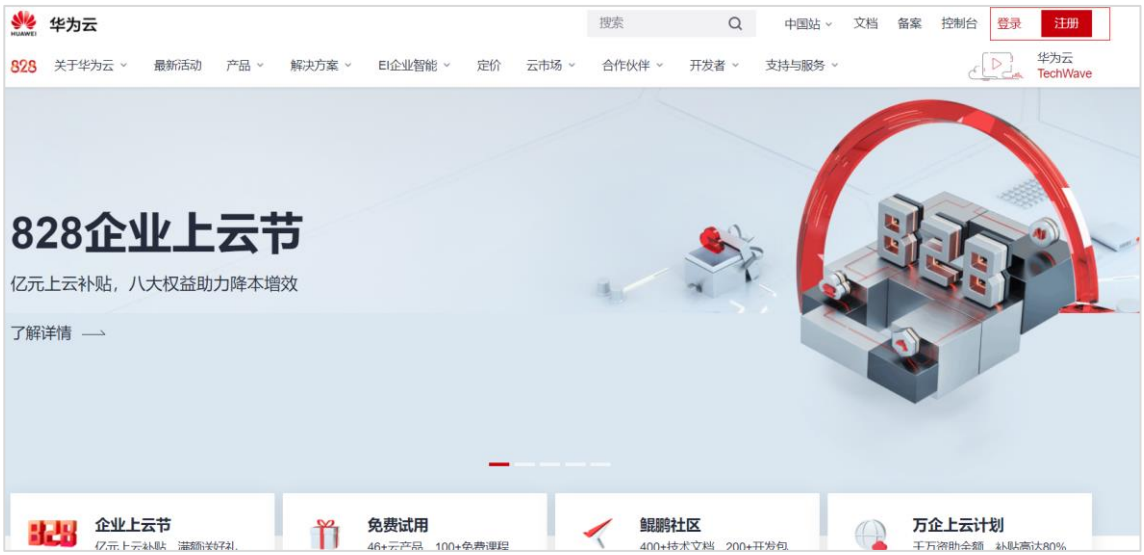
1.7 云上环境的使用（选做实验）

（此实验需要实现注册华为云账号，实验内容不涉及收费）

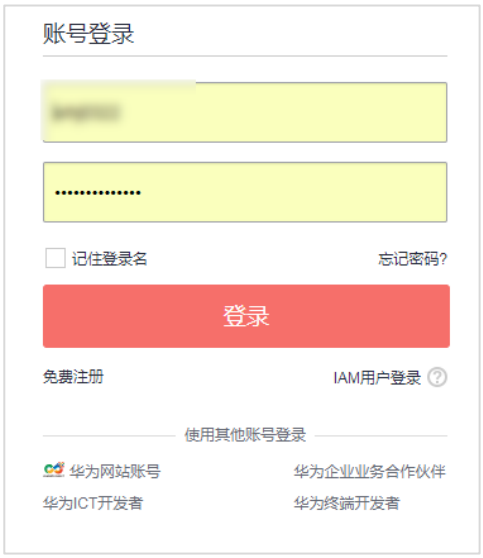
由于 Python 的一些应用开发（如，训练 AI 模型、部署 web 应用等），对于硬件设备有所要，所以云上环境成为了一些企业和程序员的选择。以下以华为云为例，了解云上开发平台 ModelArts 的使用。

步骤 1　登陆华为云，进入 ModelArts 开发平台。

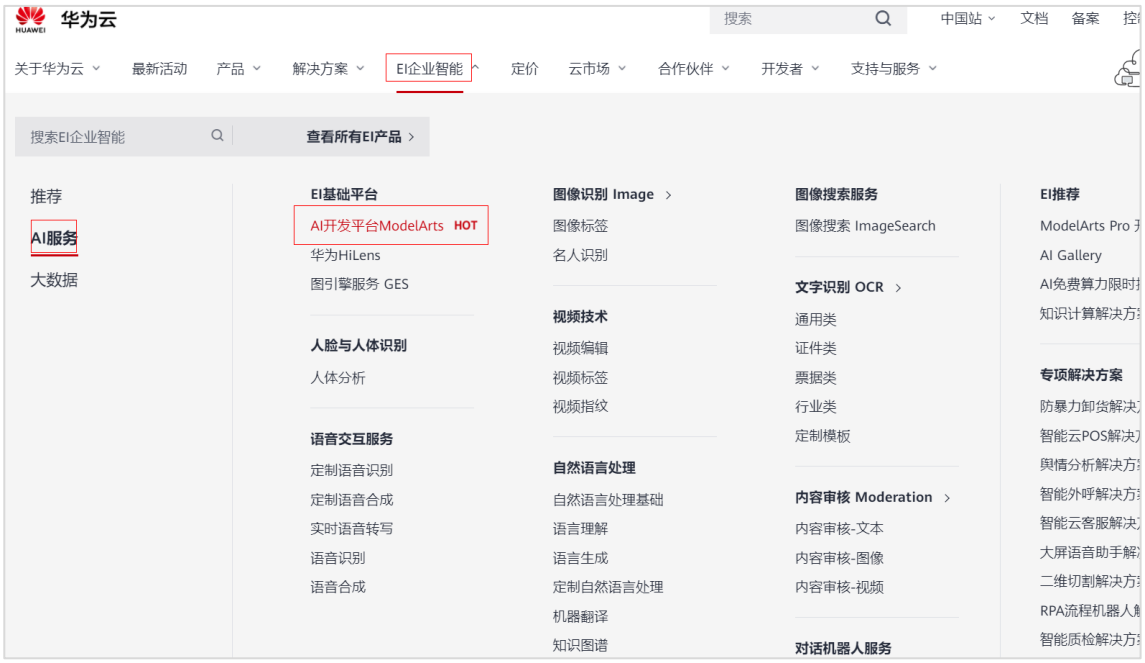
打开浏览器，输入华为云网站地址 https://www.huaweicloud.com/，在上方菜单栏中点击“登录”。

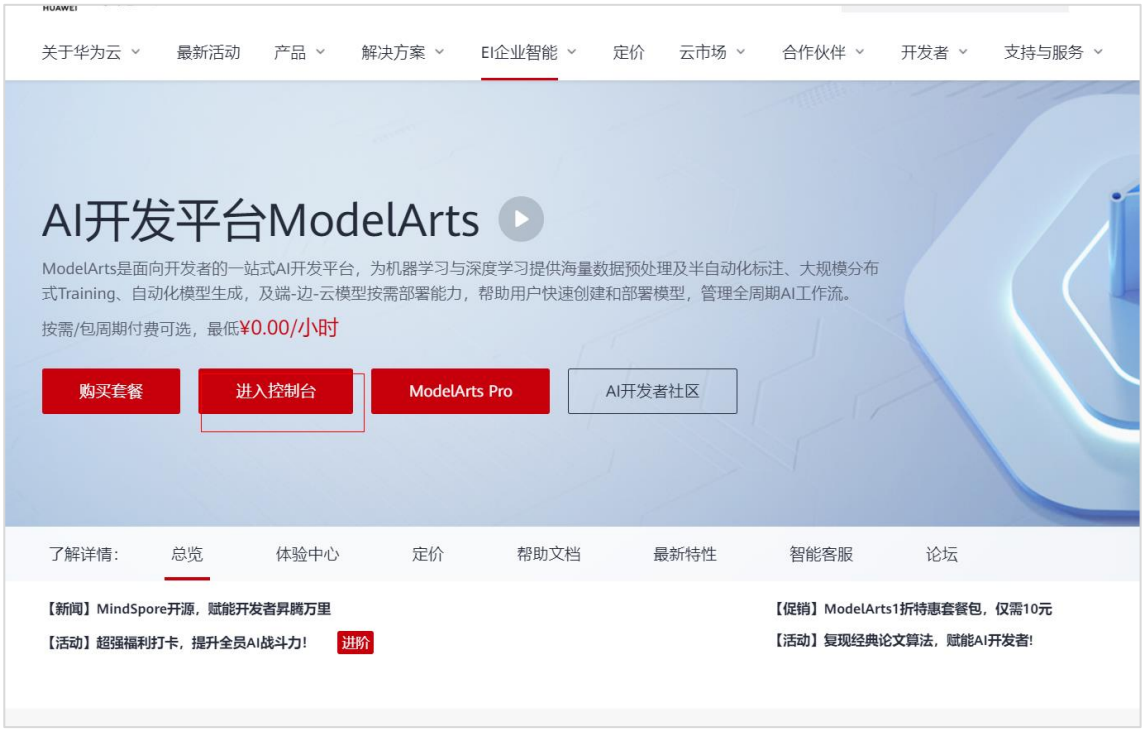


在账号登录界面输入账户名、密码，点击“登录”。

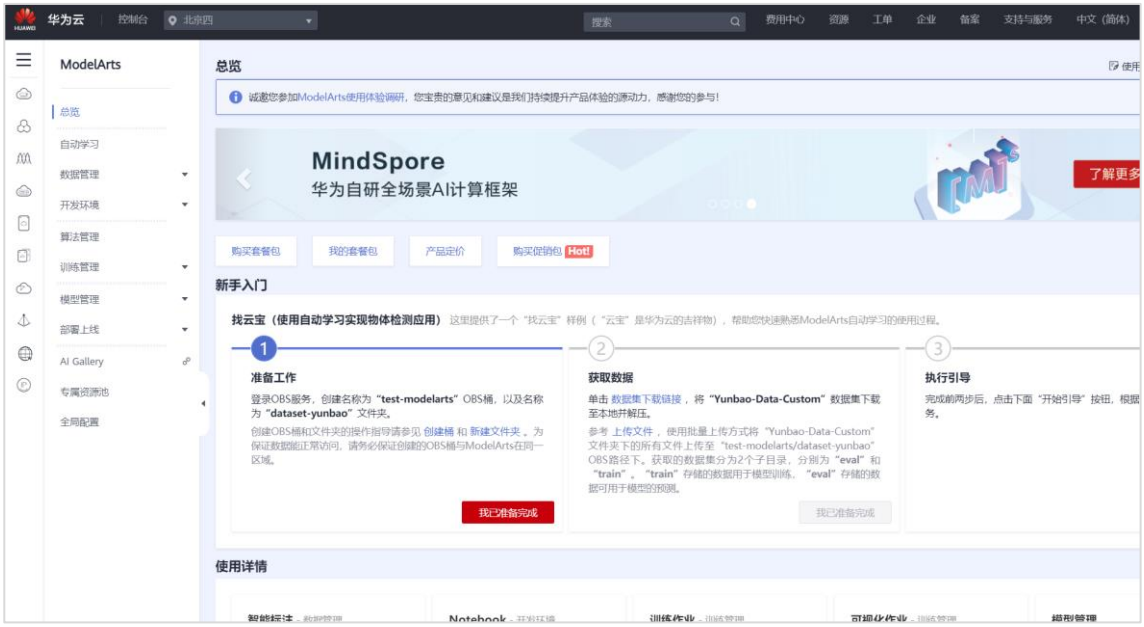


选择 EI 企业智能 -> AI 开发平台 ModelArts


华为云
搜索
中国站
文档
备案
关于华为云
最新活动
产品
解决方案
EI企业智能
定价
云市场
合作伙伴
开发者
支持与服务
搜索EI企业智能
查看所有EI产品
推荐
AI服务
大数据
EI基础平台
AI开发平台ModelArts HOT
华为HiLens
图引擎服务 GES
人脸与人体识别
人体分析
语音交互服务
定制语音识别
定制语音合成
实时语音转写
语音识别
语音合成
图像识别 Image
图像标签
名人识别
视频技术
视频编辑
视频标签
视频指纹
自然语言处理
自然语言处理基础
语言理解
语言生成
定制自然语言处理
机器翻译
知识图谱
图像搜索服务
图像搜索 ImageSearch
文字识别 OCR
通用类
证件类
票据类
行业类
定制模板
内容审核 Moderation
内容审核-文本
内容审核-图像
内容审核-视频
对话机器人服务
EI推荐
ModelArts Pro
AI Gallery
专项解决方案

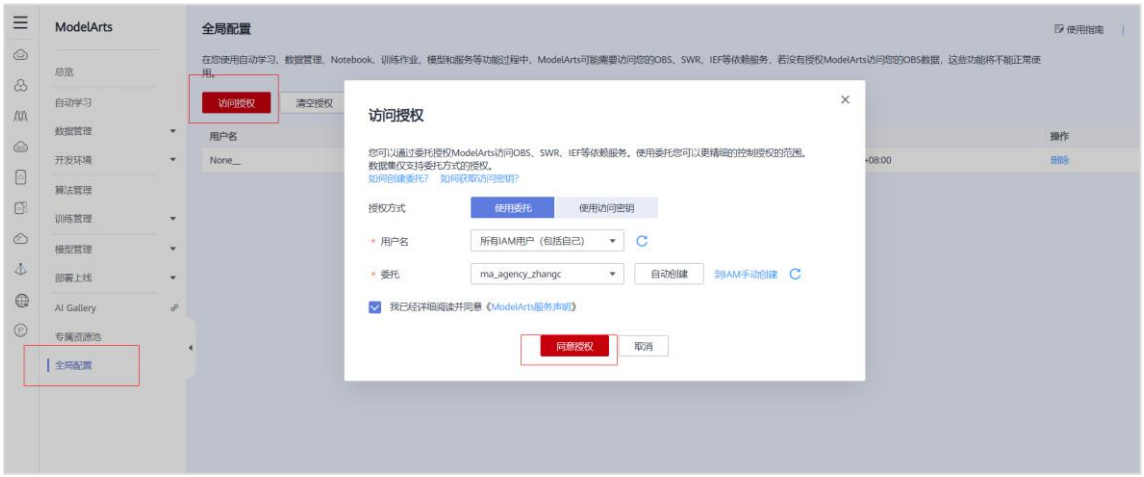

关于华为云
最新活动
产品
解决方案
EI企业智能
定价
云市场
合作伙伴
开发者
支持与服务
AI开发平台ModelArts
ModelArts是面向开发者的一站式AI开发平台，为机器学习与深度学习提供海量数据预处理及半自动化标注、大规模分布式Training、自动化模型生成，及端-边-云模型按需部署能力，帮助用户快速创建和部署模型，管理全周期AI工作流。
按需/包周期付费可选，最低¥0.00/小时
购买套餐
进入控制台
ModelArts Pro
AI开发者社区
了解详情:
总览
体验中心
定价
帮助文档
最新特性
智能客服
论坛
【新闻】MindSpore开源，赋能开发者昇腾万里
【活动】超强福利打卡，提升全员AI战斗力!
进阶
【促销】ModelArts1折特惠套餐包，仅需10元
【活动】复现经典论文算法，赋能AI开发者!

点击进入控制台。



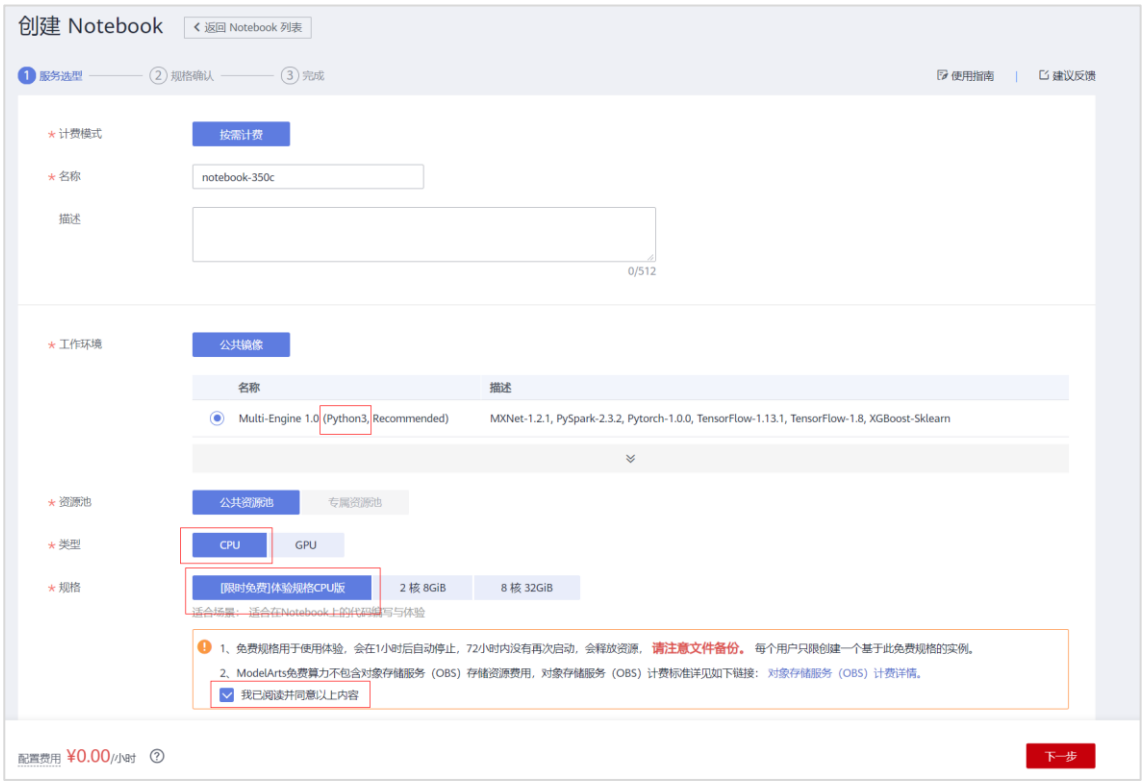
步骤 2　进入云上 notebook 开发环境。

点击全局配置，进行服务授权。



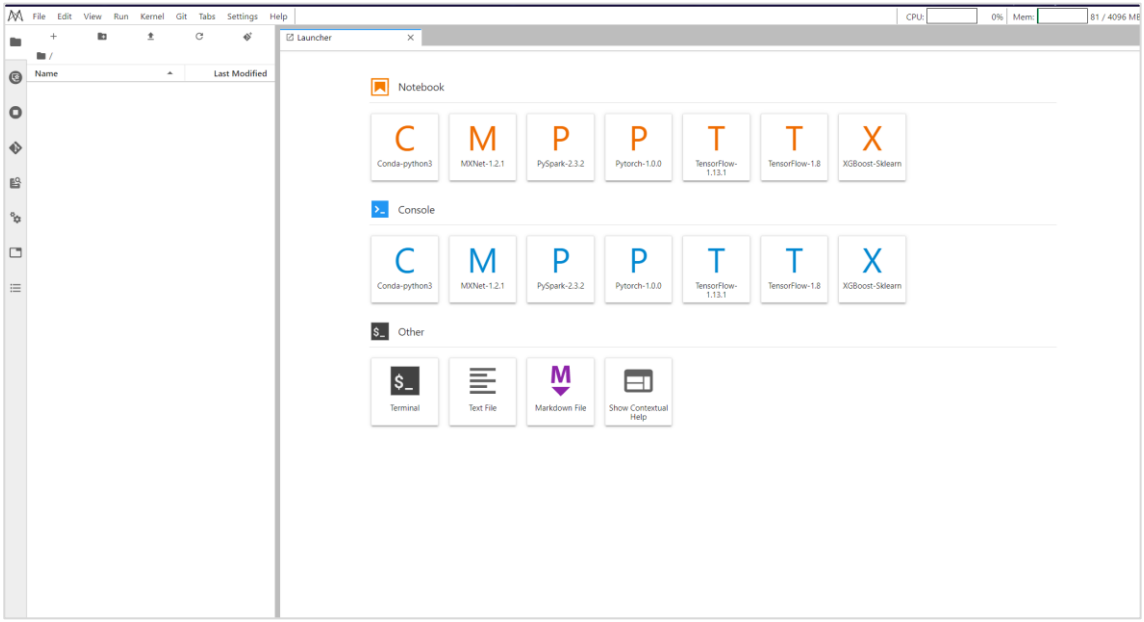
点击开发环境，创建 notebook。

在界面中选择如下配置。

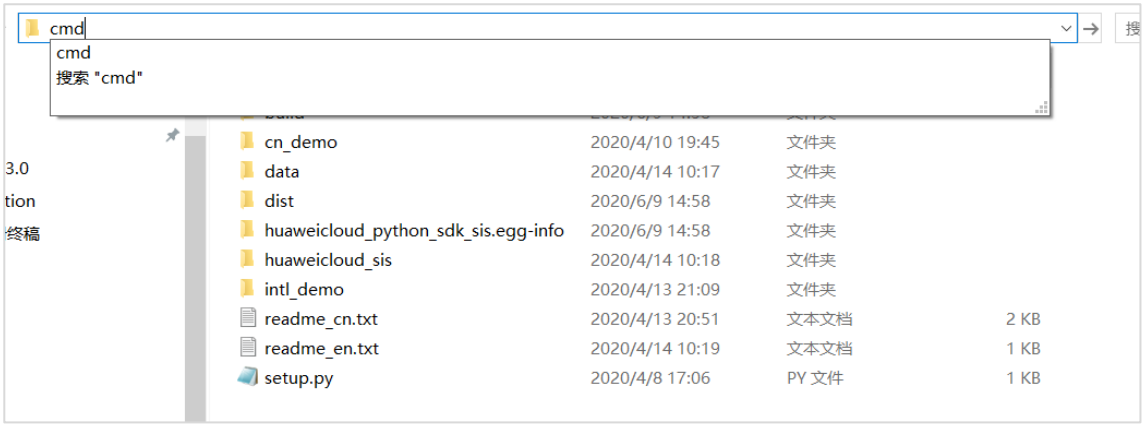


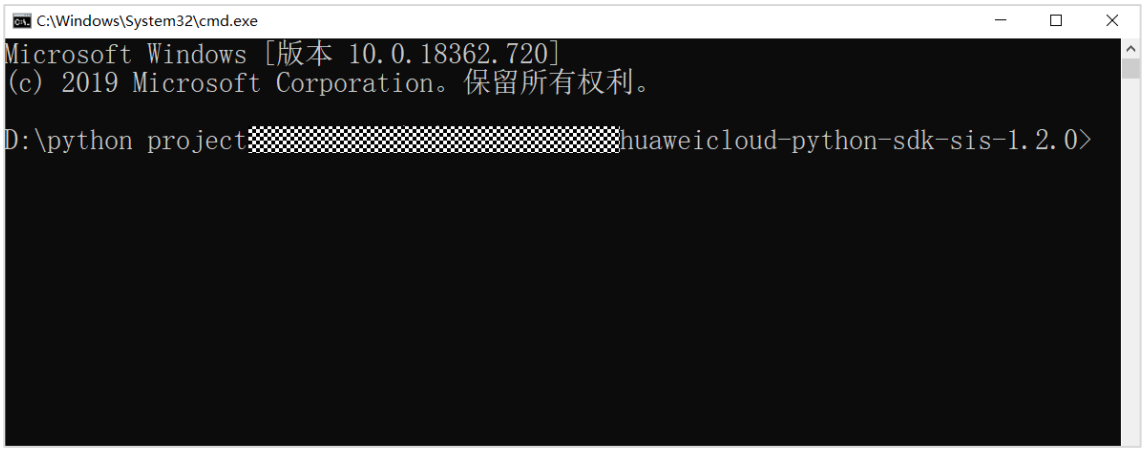
点击下一步并提交。

选择打开 jupyter lab。



此时和本地开发环境的操作就变得一样了。





2 Python 基础语法

2.1 实验介绍

实验介绍了Python（Python3）语言的基础，帮助读者快速掌握Python编程语言的基本语法和Python中的基础函数的使用。

2.2 实验目的

通过以下小的实验可以帮助我们掌握Python这门编程语言。

2.3 资源准备

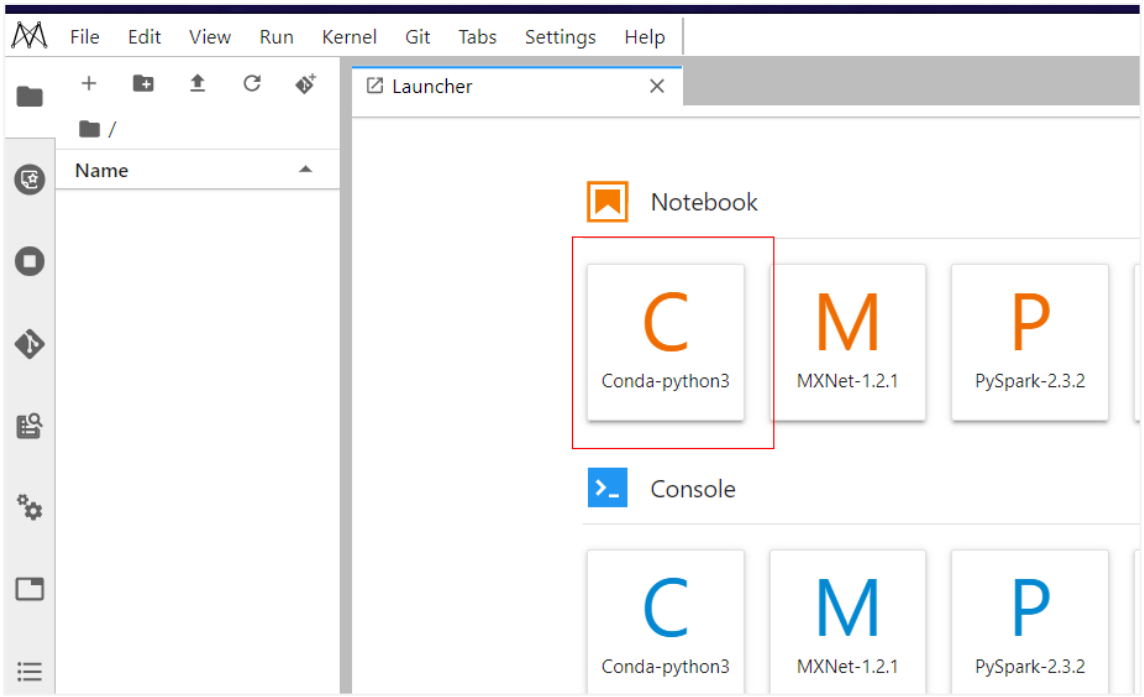
安装 Python 解释器/anaconda。

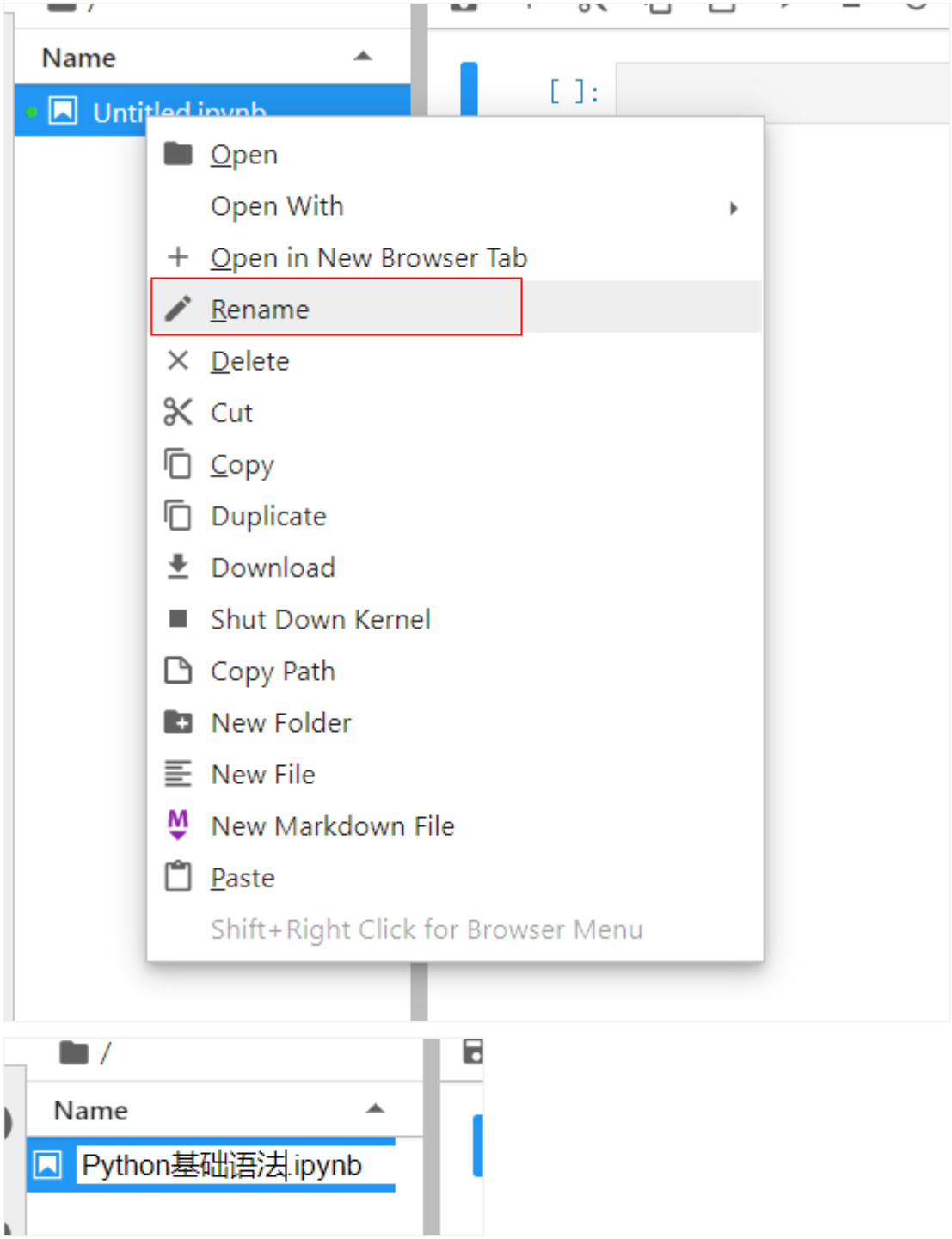
安装好 jupyter lab。

(方便起见，实验所使用的环境为云上环境 modelarts)

2.4 实验步骤

步骤 1　新建 notebook（python3）文件。并修改名称为 Python 基础语法


File
Edit
View
Run
Kernel
Git
Tabs
Settings
Help
Launcher
/
Name
Notebook
C
Conda-python3
M
MXNet-1.2.1
P
PySpark-2.3.2
Console
C
Conda-python3
M
MXNet-1.2.1
P
PySpark-2.3.2



步骤 2　第一个 Python 程序，打印 hello world

```
print('hello world') #打印出：hello world
print("hello world") #打印出：hello world。单双引号输出相同。
```

步骤 3　使用不同的方式导入工具包

通过 import 导入工具包。

```
import os # 导入 os 模块
os.getcwd() # 查看当前路径
```

输出：

```
'/home/ma-user/work'
```

使用 from...import...的方式导入工具包：

```
from os import getcwd
getcwd()
```

输出：

```
'/home/ma-user/work'
```

给工具包起个别名:

```
from os import getcwd as gt
gt()
```

输出：

```
'/home/ma-user/work'
```

步骤 4　变量的使用和命名

变量使用时需要一个名字-变量名（标识符），这个名字由字母数字下划线组成，并且数字不能开头，不能和关键字重名（已经被定义好拥有特定功能的标识符）。

查看关键字。

```
import keyword
print(keyword.kwlist)
```

输出：

```
['False', 'None', 'True', 'and', 'as', 'assert', 'break', 'class', 'continue',
'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global',
'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise',
'return', 'try', 'while', 'with', 'yield']
```

声明变量：

```
a = 1 # "="的作用为赋值
a1_ = 1
_1 = 1
```

错误的变量名：

```
1a_ = 1
if = 1
1&2 = 1 # 这些变量会使得程序报错
```

局部变量和全局变量：

```
a = 1 # 全局变量

def func(): # 定义一个函数
```

```
    b = 3 # 在函数内部定义局部变量

c = 2

print(a+c)
print(a+b)
```

输出：

```
3
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-3-ad3ad6c8508d> in <module>
      7
      8 print(a+c)
----> 9 print(a+b)

NameError: name 'b' is not defined
```

步骤 5　在程序中使用注释

使用单行注释：

```
s = "python" # 这是一个字符串
print(s)  # print("hello")
```

输出：

```
python
```

实现多行注释：

```
"""
使用三引号
实现多行注释
"""
print("hello world")
```

输出：

```
hello world
```

步骤 6　Python 程序的执行顺序

Python 语句自上向下执行，变量需要先定义在使用：

```
a = 0
print(a+b)
b = 1
```

输出：

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-5-3e1a61996624> in <module>
      1 a = 0
----> 2 print(a+b)
      3 b = 1

NameError: name 'b' is not defined
```

步骤 7　Python 中区分语句块的缩进

Python 中的缩进可以是任意多个空格：

```
def func1():
  a = 1 # 此处是三个空格
def func2():
  b = 2
```

步骤 8　基础函数的使用

使用 help 函数查看对象的帮助信息：

```
help(print) # help 函数查看帮助信息
```

```
Help on built-in function print in module builtins:

print(...)
    print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)

    Prints the values to a stream, or to sys.stdout by default.
    Optional keyword arguments:
    file:  a file-like object (stream); defaults to the current sys.stdout.
    sep:   string inserted between values, default a space.
    end:   string appended after the last value, default a newline.
    flush: whether to forcibly flush the stream.
```

使用 dir 查看对象的属性和具有的方法：

```
dir(print) # 查看函数的属性和具有的方法
```

```
['__call__',
 '__class__',
 '__delattr__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__le__',
 '__lt__',
 '__module__',
 '__name__',
 '__ne__',
 '__new__',
 '__qualname__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__self__',
 '__setattr__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 '__text_signature__']
```

使用 id 函数查看对象的内存地址：

```
a = 1
print(id(a)) # id 查看内存地址
```

输出：

```
140734273581328
```

使用 type 函数查看对象的类型：

```
type(a) # 查看数据类型
```

输出：

```
int
```

输入和输出：

```
name = input("请输入您的姓名：")
```

```
age = input("请输入您的年龄：")
print(name,end="  ")
print(age)
```

输出：

```
请输入您的姓名：  张三
请输入您的年龄：  18
张三  18
```

使用 del 方法删除内存中的对象：

```
b = 10
del(b)
print(b)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-14-dd192c9b1278> in <module>()
      1 b = 10
      2 del(b)
----> 3 print(b)

NameError: name 'b' is not defined
```

使用 len 函数查看数据的长度：

```
len("hello") # 查看对象长度
```

输出：

```
5
```

使用 range 函数生成序列：

```
# range(start, stop[, step])
# 从 start 开始。默认是从 0 开始
# 到 stop 结束，但不包括 stop。
# 步长，默认为 1
for i in range(5):
    print(i)
```

输出：

```
0
1
2
3
4
```

2.5 小结任务

发挥你的想象力写一段连续的代码（10 行以上），可以正常运行。