**计算机视觉课程设计-华为内容一**

**目录**

实验1-1

实验1-2

实验1-3

实验1-4

选做1

选做2

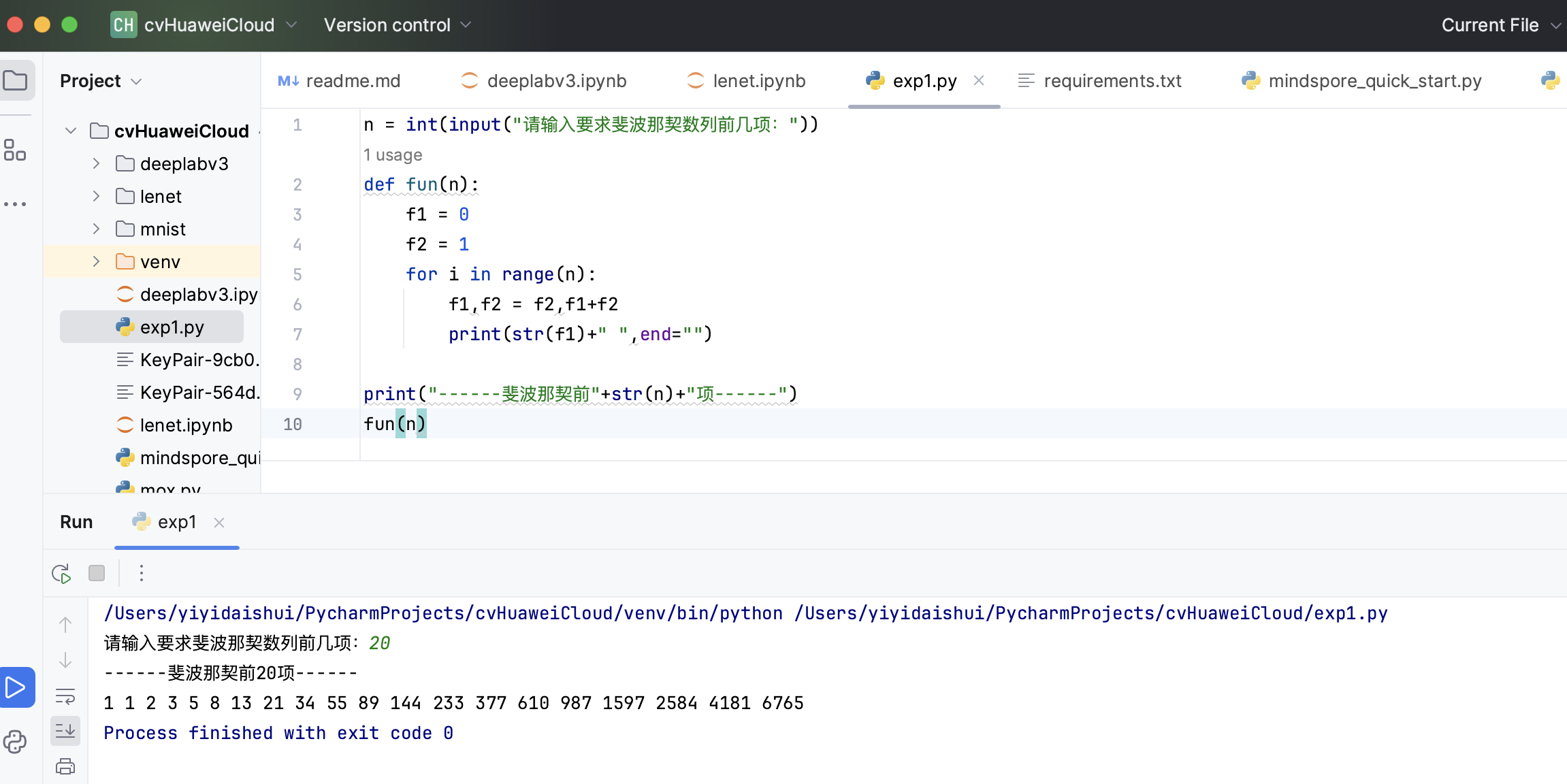
**实验1-1**

**实验要求：**

用Python实现斐波那契序列。

**代码及结果如下：**

*# exp1-1*n = int(input("请输入要求斐波那契数列前几项："))  
def fun(n):  
 f1 = 0  
 f2 = 1  
 for i in range(n):  
 f1,f2 = f2,f1+f2  
 print(str(f1)+" ",end="")  
  
print("------斐波那契前"+str(n)+"项------")  
fun(n)



**实验1-2**

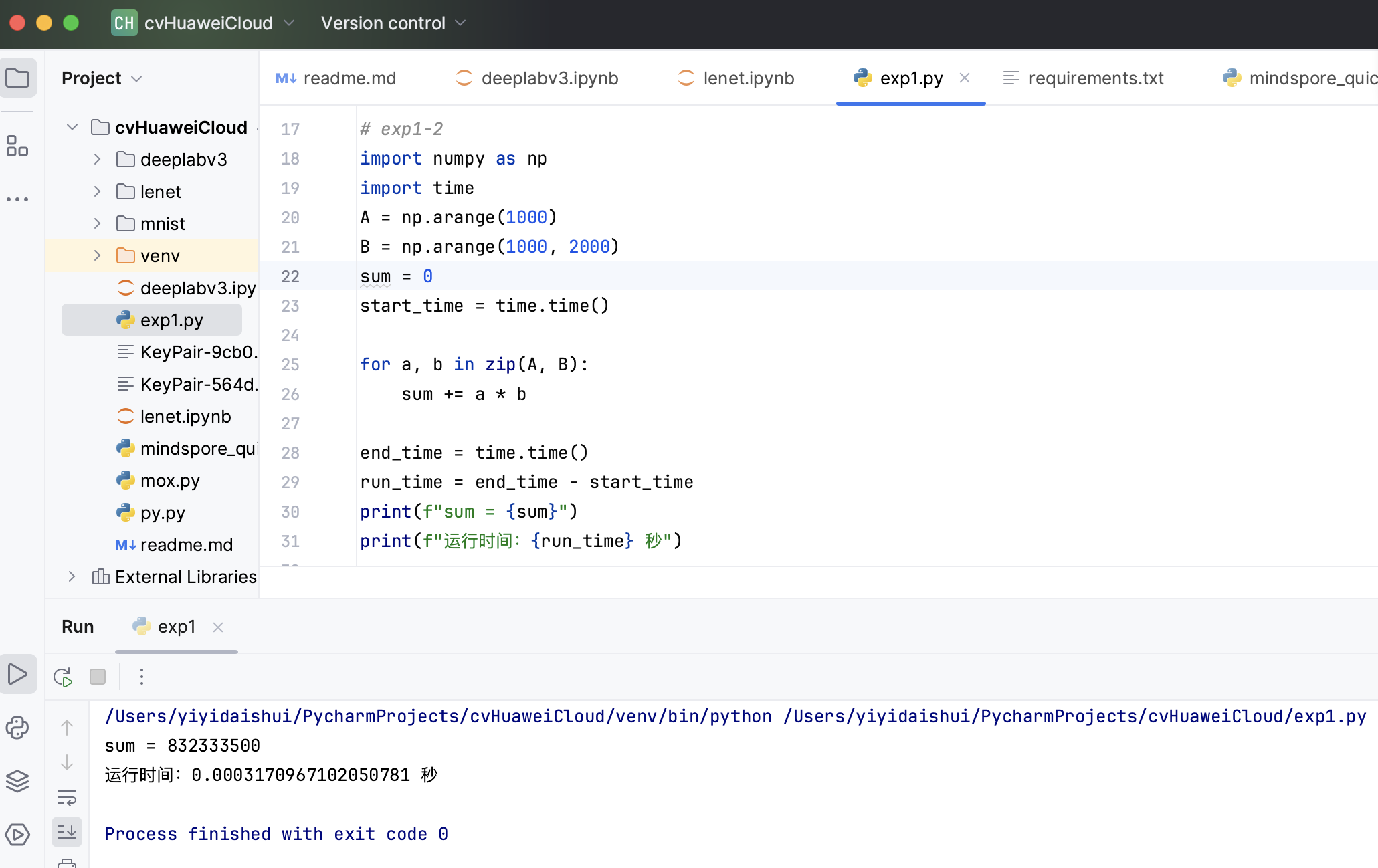
**实验要求：**

用Python实现两个数组中对应元素相乘并累加，即点积运算。用for循环实现，给出计算结果，并给出这段程序的运行时长。

，，，

**代码及结果如下：**

*# exp1-2*import numpy as np  
import time  
A = np.arange(1000)  
B = np.arange(1000, 2000)  
sum = 0  
start\_time = time.time()  
  
for a, b in zip(A, B):  
 sum += a \* b  
  
end\_time = time.time()  
run\_time = end\_time - start\_time  
print(f"sum = {sum}")  
print(f"运行时间：{run\_time} 秒")



**实验1-3**

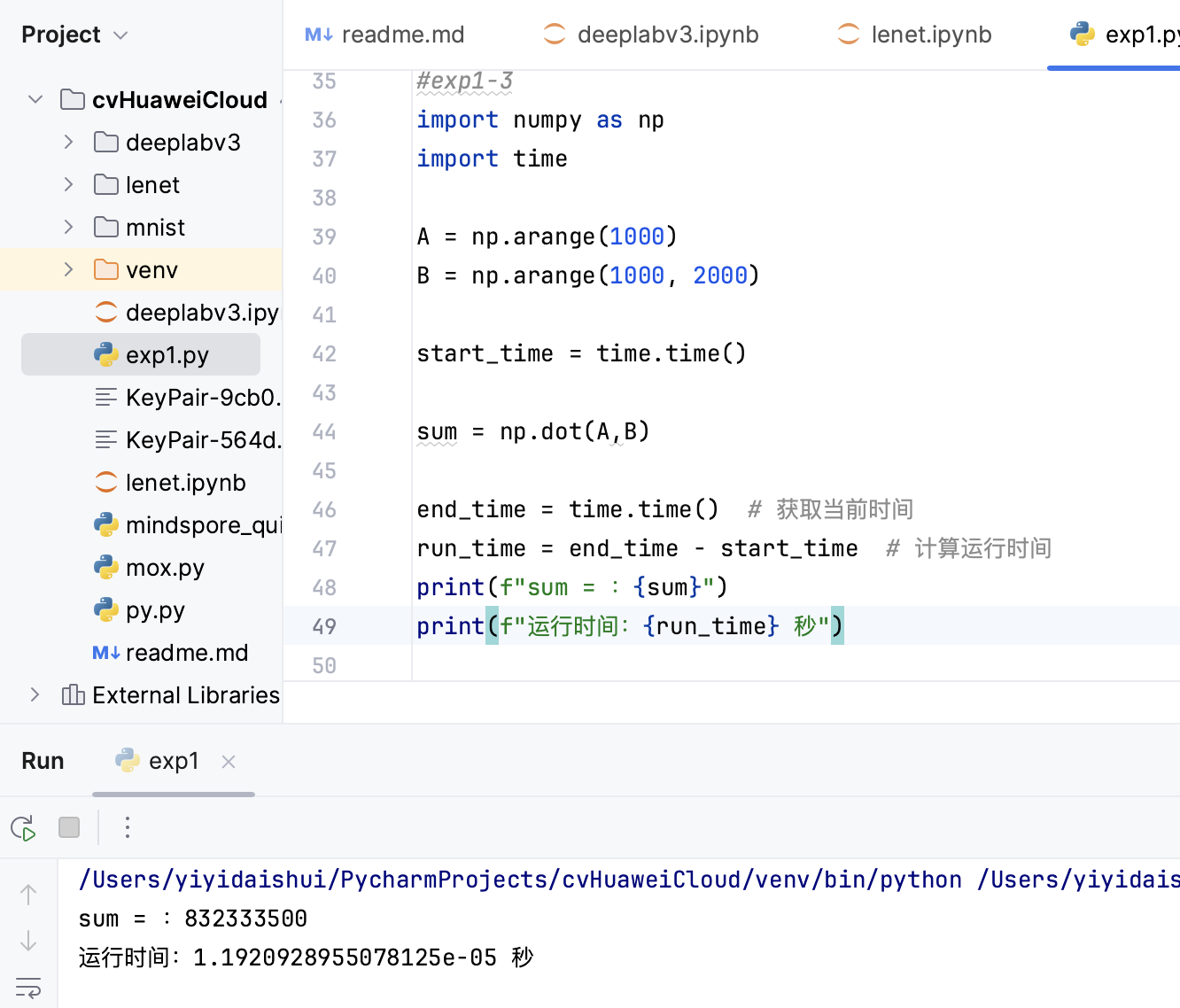
**实验要求：**

实现实验1-2中的点积运算。用NumPy库实现，给出计算结果，并给出这段程序的运行时长。

，，，

**代码及结果如下：**

*#exp1-3*import numpy as np  
import time  
  
A = np.arange(1000)  
B = np.arange(1000, 2000)  
  
start\_time = time.time()  
sum = np.dot(A,B)  
end\_time = time.time() *# 获取当前时间*run\_time = end\_time - start\_time *# 计算运行时间*print(f"sum = ：{sum}")  
print(f"运行时间：{run\_time} 秒")



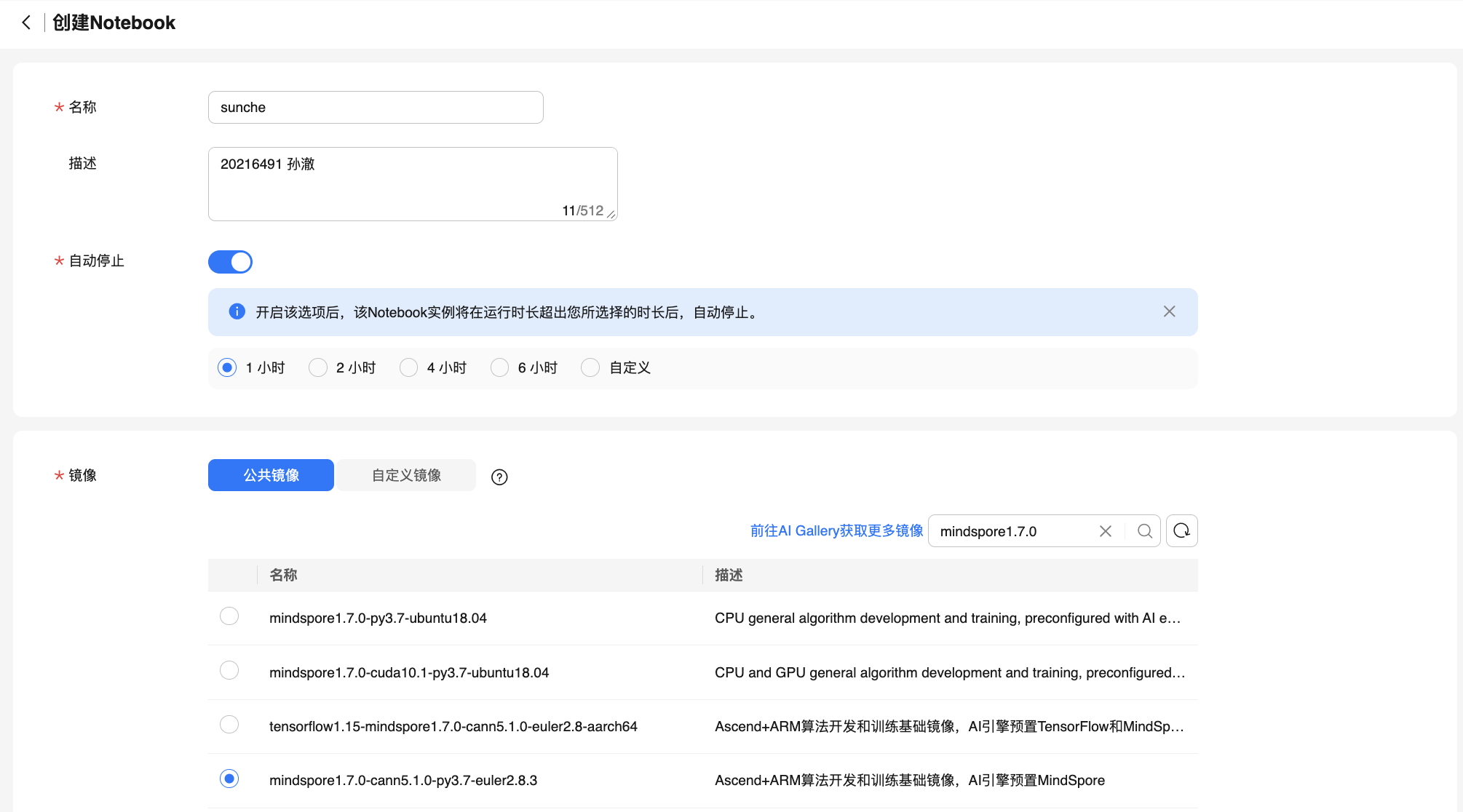
**实验1-4**

**实验要求：**

使用opencv库实现图像的 1. 旋转 2. 镜像(左右、上下) 3.增亮 4. 增加噪音 。将原图和新图保存下来。在modelarts上实现。命名方法：原名\_XX.png

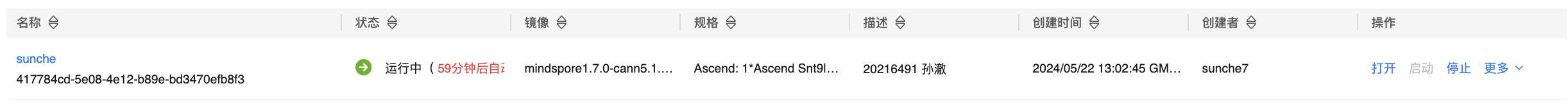
**步骤如下：**

创建notebook，名称为sunche



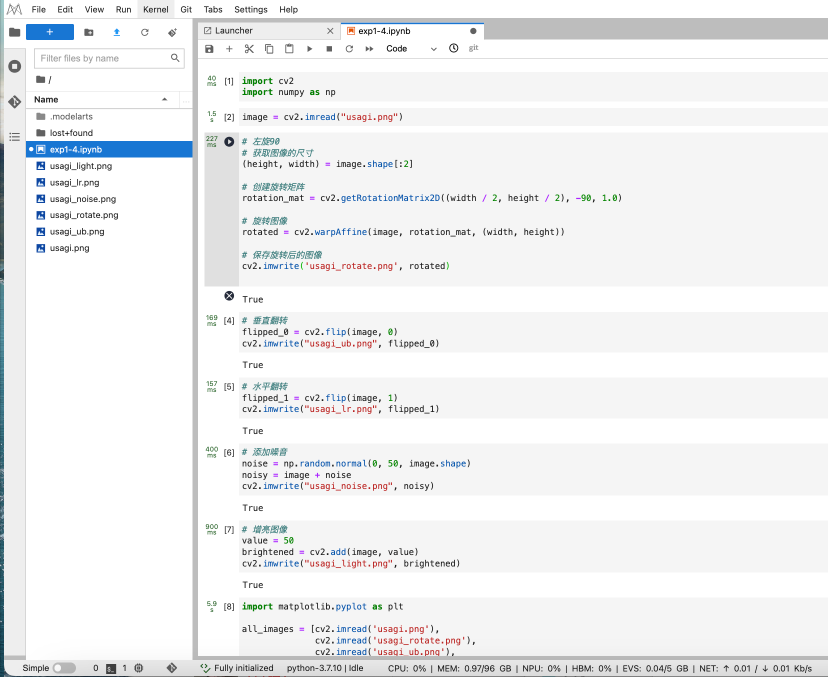


创建成功：



**代码如下：**

*#exp1-4*import cv2  
import numpy as np  
image = cv2.imread("usagi.png")  
*# 左旋90  
# 获取图像的尺寸*(height, width) = image.shape[:2]  
  
*# 创建旋转矩阵*rotation\_mat = cv2.getRotationMatrix2D((width / 2, height / 2), -90, 1.0)  
  
*# 旋转图像*rotated = cv2.warpAffine(image, rotation\_mat, (width, height))  
  
*# 保存旋转后的图像*cv2.imwrite('usagi\_rotate.png', rotated)  
  
*# 垂直翻转*flipped\_0 = cv2.flip(image, 0)  
cv2.imwrite("usagi\_ub.png", flipped\_0)  
*# 水平翻转*flipped\_1 = cv2.flip(image, 1)  
cv2.imwrite("usagi\_lr.png", flipped\_1)  
*# 添加噪音*noise = np.random.normal(0, 50, image.shape)  
noisy = image + noise  
cv2.imwrite("usagi\_noise.png", noisy)  
*# 增亮图像*value = 50  
brightened = cv2.add(image, value)  
cv2.imwrite("usagi\_light.png", brightened)  
import matplotlib.pyplot as plt  
  
all\_images = [cv2.imread('usagi.png'),  
 cv2.imread('usagi\_rotate.png'),  
 cv2.imread('usagi\_ub.png'),  
 cv2.imread('usagi\_lr.png'),  
 cv2.imread('usagi\_noise.png'),  
 cv2.imread('usagi\_light.png')]  
  
file\_names = ['usagi.png',  
 'usagi\_rotate.png',  
 'usagi\_ub.png',  
 'usagi\_lr.png',  
 'usagi\_noise.png',  
 'usagi\_light.png']  
  
  
fig, axes = plt.subplots(2, 3, figsize=(15, 10))  
  
*# 将每张图像添加到子图中*for i, ax in enumerate(axes.flat):  
 *# 将 BGR 图像转换为 RGB* image\_rgb = cv2.cvtColor(all\_images[i], cv2.COLOR\_BGR2RGB)  
 ax.imshow(image\_rgb)  
 *# 在图片下方添加文件名* ax.set\_title(file\_names[i], fontsize=12, pad=10)  
 ax.axis('off') *# 隐藏坐标轴  
  
# 调整子图间的间距*plt.subplots\_adjust(wspace=0.2, hspace=0.3)  
  
*# 显示图像*plt.show()



**结果如下：**



**选做1**

**实验要求：**

输入一个整数(高度)，输出对应高度的等腰三角形。

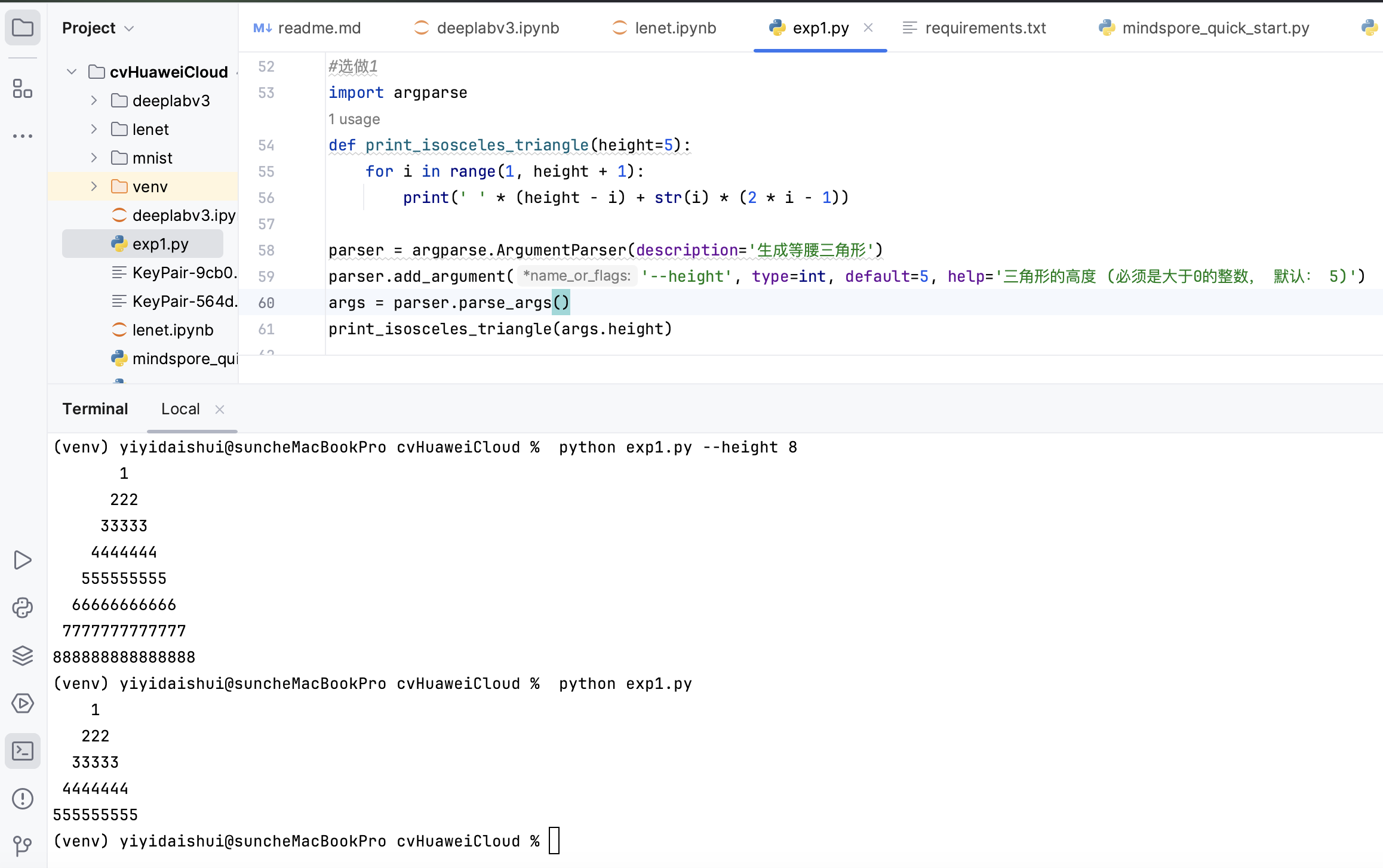
使用argparse来输入参数，不能使用input()函数

高度小于10，默认是5

**代码及结果如下：**

*#选做1*import argparse  
def print\_isosceles\_triangle(height=5):  
 for i in range(1, height + 1):  
 print(' ' \* (height - i) + str(i) \* (2 \* i - 1))  
  
parser = argparse.ArgumentParser(description='生成等腰三角形')  
parser.add\_argument('--height', type=int, default=5, help='三角形的高度 (必须是大于0的整数， 默认： 5)')  
  
args = parser.parse\_args()  
  
print\_isosceles\_triangle(args.height)

在终端输入python exp1.py --height 8和python exp1.py运行：



**选做2**

**实验要求：**

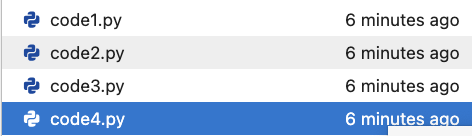
使用shell脚本在modelarts上跑通实验1-1, 1-2, 1-3, 1-4

一个shell实现4个代码

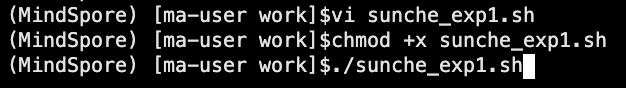
运行结果用“-------------”

**步骤如下：**

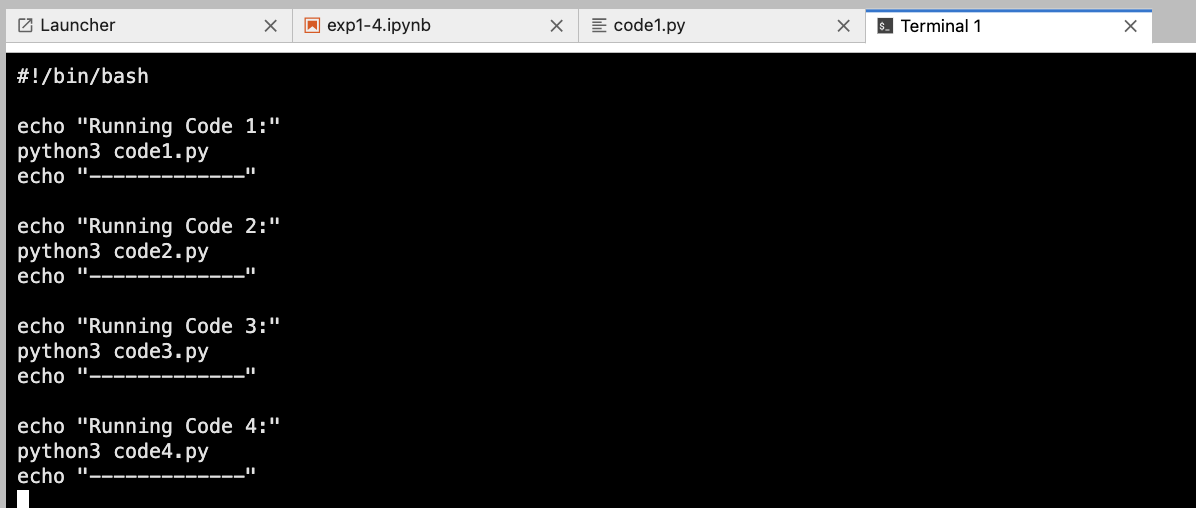
首先用code1.py、code2.py、code3.py 和 code4.py 四个 Python 文件保存实验1-1至1-4的代码。



在终端中新建脚本文件：



**内容如下：**



**运行结果如下：**

其中，实验1-4结果保存在文件中。

