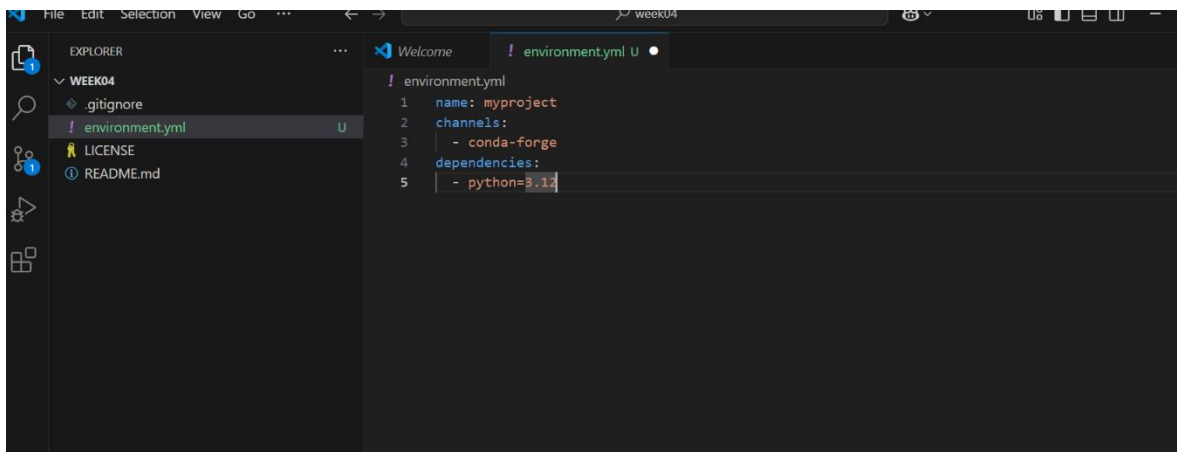


《金融编程与计算》第四周打卡

将 week04 克隆至本地

```
$ cd repo
(base)
zhangyunfei@LAPTOP-NU6BCKG MINGW64 ~/repo
$ git clone git@gitcode.com:zhangyunfei-2001/week04.git
Cloning into 'week04'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 5 (delta 0), reused 5 (delta 0), pack-reused 0 (f
Receiving objects: 100% (5/5), 8.45 KiB | 4.22 MiB/s, done.
(base)
zhangyunfei@LAPTOP-NU6BCKG MINGW64 ~/repo
$ |
```

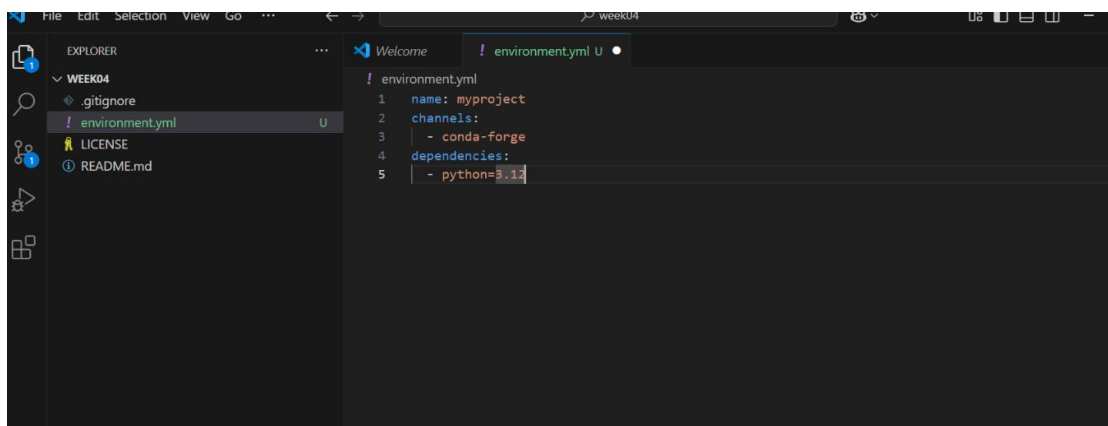
配置环境文件



cd week04 git remote show origin 查看远程仓库

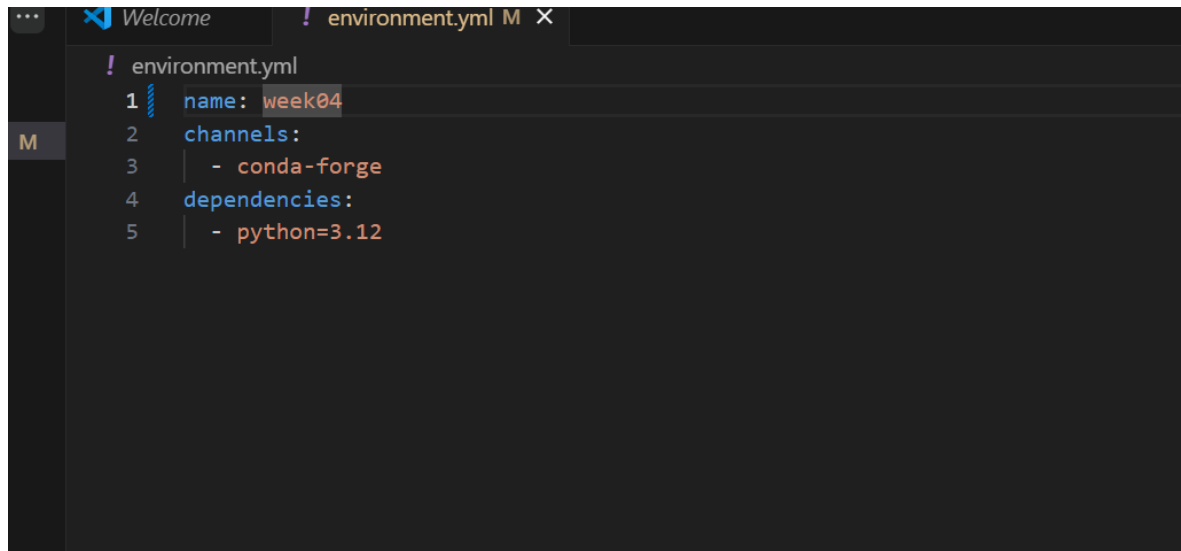
conda env create 创建 conda 环境 (cd 到 week04 文件夹之后)

初次创建, 创建失败, 经过检查, 原因是复制 environment 文件时 name 未修改



此时运行创建环境命令，创建的环境名称仍为 my project，由于之前已经创建过该环境，此时运行报错。

修改环境名称后，再次创建，创建成功。

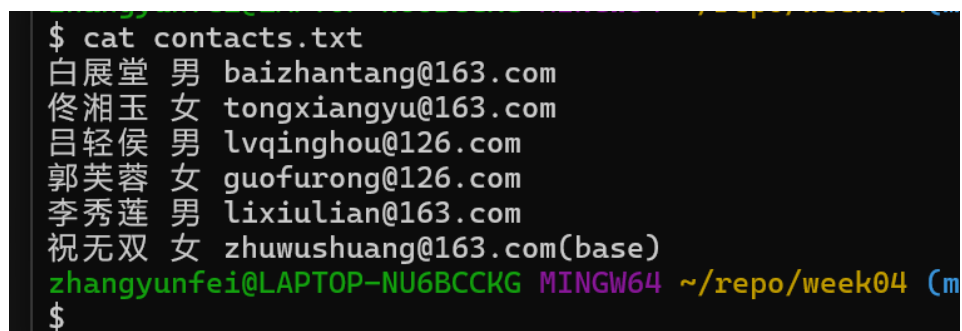


```
! environment.yml
1  name: week04
2  channels:
3    - conda-forge
4  dependencies:
5    - python=3.12
```

```
zhangyunfei@LAPTOP-NU6BCCKG MINGW64 ~/repo/week04 (main)
$ conda env list
# conda environments:
#
base                  *  D:\python
myproject             D:\python\envs\myproject
prj1                  D:\python\envs\prj1
prj2                  D:\python\envs\prj2
week04                D:\python\envs\week04

(base)
zhangyunfei@LAPTOP-NU6BCCKG MINGW64 ~/repo/week04 (main)
$ |
```

新建纯文本文件，内容为联系人信息 cat 文件名 查看文件内容



```
$ cat contacts.txt
白展堂 男 baizhantang@163.com
佟湘玉 女 tongxiangyu@163.com
吕轻侯 男 lvqinghou@126.com
郭芙蓉 女 guofurong@126.com
李秀莲 男 lixiulian@163.com
祝无双 女 zhuwushuang@163.com
zhangyunfei@LAPTOP-NU6BCCKG MINGW64 ~/repo/week04 (m
$
```

根据程序要求，AI 生成 python 程序

新建一个 contacts.txt 文件，每行写一个联系人，每个联系人都包含姓名、性别、邮箱三个字段，用空格分隔，例如

白展堂 男 baizhantang@163.com
佟湘玉 女 tongxiangyu@163.com
吕轻侯 男 lvqinghou@126.com
郭芙蓉 女 guofurong@126.com
李秀莲 男 lixiulian@163.com
祝无双 女 zhuwushuang@163.com

新建一个 main.py 文件，里面写 Python 代码，要求读取 contacts.txt 文件的内容，进行数据处理后，输出一个 emails.txt 文件，例如

to: <guofurong@126.com>
尊敬的郭芙蓉女士，您的会员资格即将到期，请及时续费。

to: <lvqinghou@126.com>
尊敬的吕轻侯先生，您的会员资格即将到期，请及时续费。

to: <baizhantang@163.com>
尊敬的白展堂先生，您的会员资格即将到期，请及时续费。

Files

- contacts.txt
- main.py

main.py

```
1 def process_contacts():
2     try:
3         with open('contacts.txt', 'r', encoding='utf-8') as file:
4             contacts = []
5             for line in file:
6                 name, gender, email = line.strip().split()
7                 contacts.append((name, gender, email))
8
9             contacts.sort(key=lambda x: (x[2].split('@')[1], x[2].split('@')[0]))
10
11         with open('emails.txt', 'w', encoding='utf-8') as output_file:
12             for name, gender, email in contacts:
13                 title = '先生' if gender == '男' else '女士'
14                 output_file.write(f'to: <{email}>\n')
15                 output_file.write(f'尊敬的{name}{title}，您的会员资格即将到期，请及时续费：\n')
16                 output_file.write('---\n')
17     except FileNotFoundError:
18         print("错误: contacts.txt 文件未找到!")
19     except Exception as e:
20         print(f"错误: 发生了一个未知错误: {e}")
21
22
23 if __name__ == "__main__":
24     process_contacts()
25
```

将程序复制到 main.py 运行成功。

```
zhangyundfei@LAPTOP-NU6BCKG MINGW64 ~/repo/week04 (main)
$ python main.py
(week04)
zhangyundfei@LAPTOP-NU6BCKG MINGW64 ~/repo/week04 (main)
$ cat emails.txt
to: <guofurong@126.com>
尊敬的郭芙蓉女士，您的会员资格即将到期，请及时续费。
---
to: <lvqinghou@126.com>
尊敬的吕轻侯先生，您的会员资格即将到期，请及时续费。
---
to: <baizhantang@163.com>
尊敬的白展堂先生，您的会员资格即将到期，请及时续费。
---
to: <lixiulian@163.com>
尊敬的李秀莲先生，您的会员资格即将到期，请及时续费。
---
to: <tongxiangyu@163.com>
尊敬的佟湘玉女士，您的会员资格即将到期，请及时续费。
---
to: <zhuwushuang@163.com>
尊敬的祝无双女士，您的会员资格即将到期，请及时续费。
---
```

ls -l 查看当前目录下的文件列表 rm email.txt 删除文件

```

$ ls -l
total 31
-rw-r--r-- 1 zhangyunfei 197121 202 3月 27 14:03 contacts.txt
-rw-r--r-- 1 zhangyunfei 197121 666 3月 27 14:28 emails.txt
-rw-r--r-- 1 zhangyunfei 197121 72 3月 27 13:57 environment.yml
-rw-r--r-- 1 zhangyunfei 197121 18805 3月 26 16:29 LICENSE
-rw-r--r-- 1 zhangyunfei 197121 1036 3月 27 14:23 main.py
-rw-r--r-- 1 zhangyunfei 197121 2239 3月 26 16:29 README.md
(base)
zhangyunfei@LAPTOP-NU6BCKG MINGW64 ~/repo/week04 (main)
$ rm emails.txt
(base)
zhangyunfei@LAPTOP-NU6BCKG MINGW64 ~/repo/week04 (main)
$ ls -l
total 30
-rw-r--r-- 1 zhangyunfei 197121 202 3月 27 14:03 contacts.txt
-rw-r--r-- 1 zhangyunfei 197121 72 3月 27 13:57 environment.yml
-rw-r--r-- 1 zhangyunfei 197121 18805 3月 26 16:29 LICENSE
-rw-r--r-- 1 zhangyunfei 197121 1036 3月 27 14:23 main.py
-rw-r--r-- 1 zhangyunfei 197121 2239 3月 26 16:29 README.md
(base)

```

python -m pdb main.py 启用调试模式

```

zhangyunfei@LAPTOP-NU6BCKG MINGW64 ~/repo/week04 (main)
$ python -m pdb main.py
> c:\users\zhangyunfei\repo\week04\main.py(1)<module>()
-> def process_contacts():
(Pdb) |

```

pdb 下的命令含义 1 显示程序代码（箭头表示即将运行的代码）

```

(Pdb) 1
1 -> def process_contacts():
2     try:
3         with open("contacts.txt", "r", encoding="utf-8") as file:
4             contacts = []
5             for line in file:
6                 name, gender, email = line.strip().split()
7                 contacts.append((name, gender, email))
8
9             contacts.sort(key=lambda x: (x[2].split("@")[1], x[2].split("@")[0]))
10
11         with open("emails.txt", "w", encoding="utf-8") as output_file:
(Pdb) |

```

n 执行当前行

```

11         with open("emails.txt", "w", encoding="utf-8") as output_file:
(Pdb) n
> c:\users\zhangyunfei\repo\week04\main.py(25)<module>()
-> if __name__ == "__main__":
(Pdb) 1
20     print("错误: contacts.txt 文件未找到!")
21 except Exception as e:
22     print(f"错误: 发生了一个未知错误: {e}")
23
24
25 -> if __name__ == "__main__":
26     process_contacts()
[EOF]
(Pdb) |

```

p 打印表达式

```
[EOF]
(Pdb) p file
*** NameError: name 'file' is not defined
(Pdb) p contacts
*** NameError: name 'contacts' is not defined
(Pdb) p line
*** NameError: name 'line' is not defined
(Pdb) n
> c:\users\zhangyunfei\repo\week04\main.py(26)<module>()
-> process_contacts()
(Pdb) l .
21         except Exception as e:
22             print(f"错误: 发生了一个未知错误: {e}")
23
24
25     if __name__ == "__main__":
26         process_contacts()
[EOF]
(Pdb) p __name__
'__main__'
(Pdb) |
```

s 步入调用 函数

l. 显示上下五行的代码 11,5 显示 1 到 5 行的代码

pp 美观打印

r 返回

c 继续将程序运行到结束

```
(Pdb) c
The program finished and will be restarted
> c:\users\zhangyunfei\repo\week04\main.py(1)<module>()
-> def process_contacts():
```

安装 wat-inspector 软件包

```
(base)
zhangyunfei@LAPTOP-NU6BCCKG MINGW64 ~/repo/week04 (main)
done
#
# To activate this environment, use
#
#     $ conda activate week04
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(week04)
zhangyunfei@LAPTOP-NU6BCCKG MINGW64 ~/repo/week04 (main)
$ |
```

Python 保留字在 vscode 中显示为红色字体

理解形参和实参的区别

定义层面

- **形参**：在定义函数时使用，它是函数定义语句中括号内声明的变量，用来规定函数期望接收的数据。可以把形参看作是函数内部的一个“占位符”，它代表了函数运行时会用到的数据，但在函数未被调用时，形参并没有实际的值。
- **实参**：在调用函数时使用，是调用函数时实际传递给函数的数据。实参是具体存在的值，它可以是常量、变量、表达式或者其他函数的返回值等。

代码示例

```
python ^  
  
# 定义一个函数，这里的 name 和 age 是形参  
def introduce(name, age):  
    print(f"我叫 {name}，今年 {age} 岁。")  
  
# 调用函数，'Bob' 和 30 是实参  
introduce('Bob', 30)
```

在上述代码中，`introduce` 函数定义时的 `name` 和 `age` 是形参，而调用函数时传入的 `'Bob'` 和 `30` 就是实参。

Wat-inspector 软件包的使用方法

1. 简单对象检查

以下示例展示了如何对一个简单的整数对象进行检查：

```
python ^  
  
import wat  
  
num = 10  
wat.inspect(num)
```

在上述代码里，先是导入了 `wat` 库，接着定义了一个整数变量 `num`，最后调用 `wat.inspect` 函数对 `num` 进行检查。执行此代码后，`wat` 会输出 `num` 的详细信息，比如类型、值等。

2. 自定义类对象检查

对于自定义类的对象，同样可以使用 `wat` 进行检查，示例如下：

```
python ^  
  
import wat  
  
class Person:  
    def __init__(self, name, age):  
        self.name = name  
        self.age = age
```

```
$ python -m pdb main.py
> c:\users\zhangyunfei\repo\week04\main.py(1)<module>()
-> def process_contacts():
(Pdb) import wat
(Pdb) wat.globals
Global variables:
  __builtins__: dict = {...
  __file__: pdb._ScriptTarget = 'C:\Users\zhangyunfei\repo\week04\main.py'
  __name__: str = '__main__'
  __pdb_convenience_variables: dict = {...
  __spec__: NoneType = None
  wat: wat.inspection.inspection.Wat = <WAT Inspector object>
(Pdb) |
```