

## 第四周学习报告

1. 用 VS Code 打开项目目录，新建一个 environment.yml 文件，指定安装 Python 3.12，然后运行 conda env create 命令创建 Conda 环境

```
MINGW64:/c/Users/86153/rep  ×  +  ▾

(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~
$ cd repo

(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~/repo
$ ls -l
total 17
drwxr-xr-x 1 86153 197609  0  3月 20 20:34 prj1/
-rw-r--r-- 1 86153 197609 221 3月 15 23:07 tkconch-script.py
drwxr-xr-x 1 86153 197609  0  3月  9 11:57 week01/
drwxr-xr-x 1 86153 197609  0  3月  9 12:00 week01_3492/
drwxr-xr-x 1 86153 197609  0  3月 16 13:28 week02/
drwxr-xr-x 1 86153 197609  0  3月 20 20:57 week03/

(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~/repo
$ git clone https://gitcode.com/gossamer/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 (from 0)
Unpacking objects: 100% (5/5), 8.43 KiB | 431.00 KiB/s, done.

(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~/repo
$ cd week04/

(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week04 (main)
$ pwd
/c/Users/86153/repo/week04
```

```
MINGW64:/c/Users/86153/rep  ×  +  ▾

$ git clone https://gitcode.com/gossamer/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 (from 0)
Unpacking objects: 100% (5/5), 8.43 KiB | 431.00 KiB/s, done.

(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~/repo
$ cd week04/

(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week04 (main)
$ pwd
/c/Users/86153/repo/week04

(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week04 (main)
$ git remote show origin
* remote origin
  Fetch URL: https://gitcode.com/gossamer/week04.git
  Push URL: https://gitcode.com/gossamer/week04.git
  HEAD branch: main
  Remote branch:
    main tracked
  Local branch configured for 'git pull':
    main merges with remote main
  Local ref configured for 'git push':
    main pushes to main (up to date)

(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week04 (main)
```

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

2. 新建一个 contacts.txt 文件新建一个 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。

```
MINGW64:/c/Users/86153/rep X + v
(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week04 (main)
$ cat ../myproject/environment.yml
cat: ../myproject/environment.yml: No such file or directory

(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week04 (main)
$ cat environment.yml
cat: environment.yml: No such file or directory

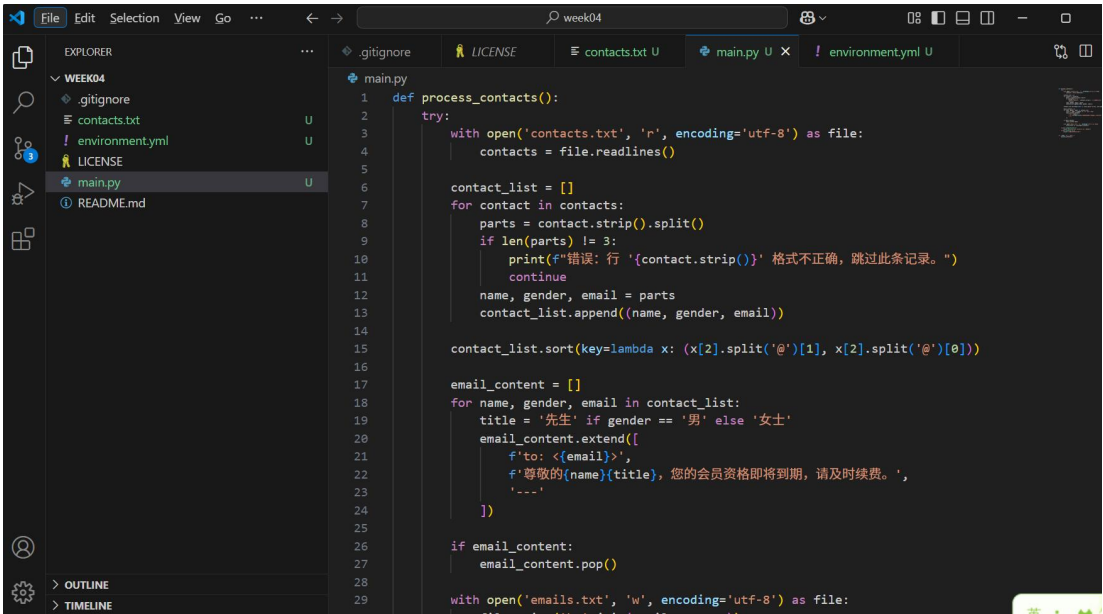
(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week04 (main)
$ cat contacts.txt

(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week04 (main)
$ ls -l
total 25
-rw-r--r-- 1 86153 197609 0 3月 28 10:58 contacts.txt
-rw-r--r-- 1 86153 197609 72 3月 28 10:59 environment.yml
-rw-r--r-- 1 86153 197609 18805 3月 28 10:36 LICENSE
-rw-r--r-- 1 86153 197609 2239 3月 28 10:36 README.md

(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week04 (main)
$ cat contacts.txt

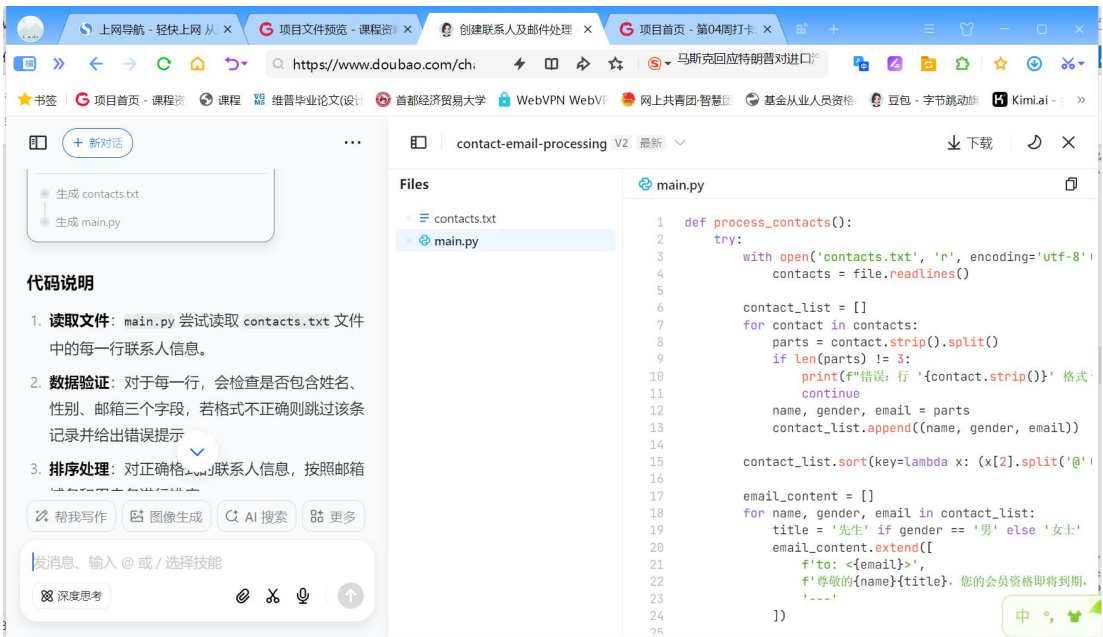
(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week04 (main)
$ cat contacts.txt
白展堂 男 baizhantang@163.com
佟湘玉 女 tongxiangyu@163.com
吕轻侯 男 lvqinghou@126.com
郭芙蓉 女 guofurong@126.com
李秀莲 男 lixiulian@163.com
```

3. 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 文件要求输出是先按邮箱域名排序 (126.com 排在 163.com 之前), 然后再按邮箱用户名排序 (guofurong 排在 lvqinghou 之前)



```
1 def process_contacts():
2     try:
3         with open('contacts.txt', 'r', encoding='utf-8') as file:
4             contacts = file.readlines()
5
6         contact_list = []
7         for contact in contacts:
8             parts = contact.strip().split()
9             if len(parts) != 3:
10                 print(f"错误: 行 '{contact.strip()}' 格式不正确, 跳过此条记录。")
11                 continue
12             name, gender, email = parts
13             contact_list.append((name, gender, email))
14
15         contact_list.sort(key=lambda x: (x[2].split('@')[1], x[2].split('@')[0]))
16
17         email_content = []
18         for name, gender, email in contact_list:
19             title = '先生' if gender == '男' else '女士'
20             email_content.extend([
21                 f'to: <{email}>',
22                 f'尊敬的{name}{title}, 您的会员资格即将到期, 请及时续费。',
23                 '---'
24             ])
25
26         if email_content:
27             email_content.pop()
28
29         with open('emails.txt', 'w', encoding='utf-8') as file:
```

4. 可以将以上 “任务要求” 的文本, 复制粘贴到大模型 (比如豆包、DeepSeek) 里, 请 AI 来帮助编写程序初稿



**代码说明**

- 读取文件:** main.py 尝试读取 contacts.txt 文件中的每一行联系人信息。
- 数据验证:** 对于每一行, 会检查是否包含姓名、性别、邮箱三个字段, 若格式不正确则跳过该条记录并给出错误提示
- 排序处理:** 对正确格式的联系人信息, 按照邮箱域名排序 (126.com 排在 163.com 之前), 然后再按邮箱用户名排序 (guofurong 排在 lvqinghou 之前)

帮我写作 图像生成 AI 搜索 更多

发消息、输入 @ 或 / 选择技能

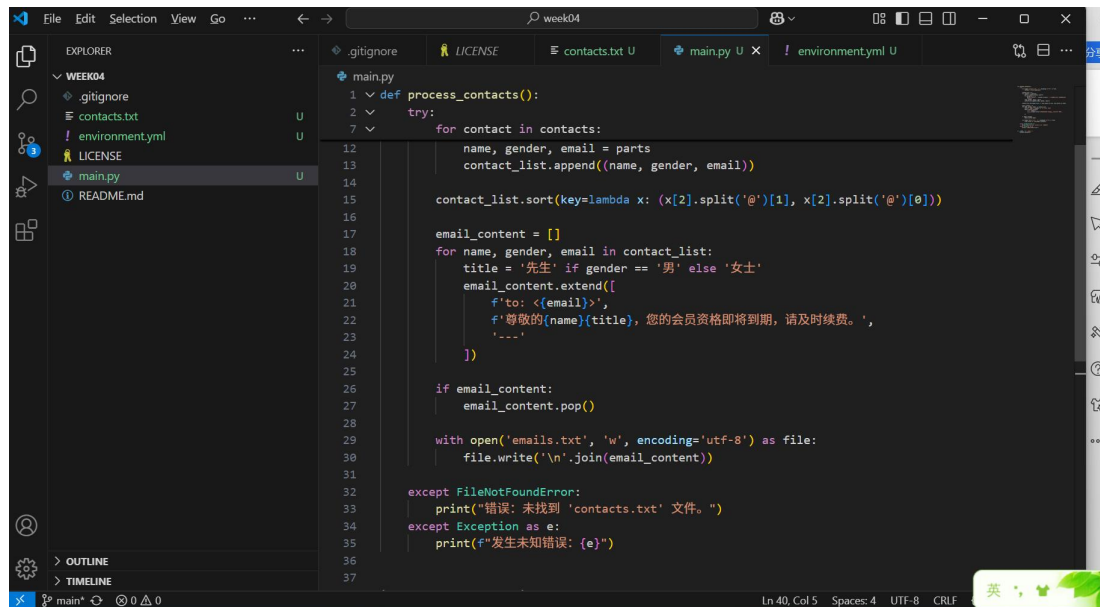
深度思考

**Files**

- contacts.txt
- main.py

```
1 def process_contacts():
2     try:
3         with open('contacts.txt', 'r', encoding='utf-8') as file:
4             contacts = file.readlines()
5
6         contact_list = []
7         for contact in contacts:
8             parts = contact.strip().split()
9             if len(parts) != 3:
10                 print(f"错误: 行 '{contact.strip()}' 格式不正确, 跳过此条记录。")
11                 continue
12             name, gender, email = parts
13             contact_list.append((name, gender, email))
14
15         contact_list.sort(key=lambda x: (x[2].split('@')[1], x[2].split('@')[0]))
16
17         email_content = []
18         for name, gender, email in contact_list:
19             title = '先生' if gender == '男' else '女士'
20             email_content.extend([
21                 f'to: <{email}>',
22                 f'尊敬的{name}{title}, 您的会员资格即将到期, 请及时续费。',
23                 '---'
24             ])
25
26         if email_content:
27             email_content.pop()
28
29         with open('emails.txt', 'w', encoding='utf-8') as file:
```

5. AI 回复的只是静态代码，而且可能含有错误，所以我们必须在 Conda 环境里运行代码，逐行调试，检查每一行代码的运行都符合我们的期望

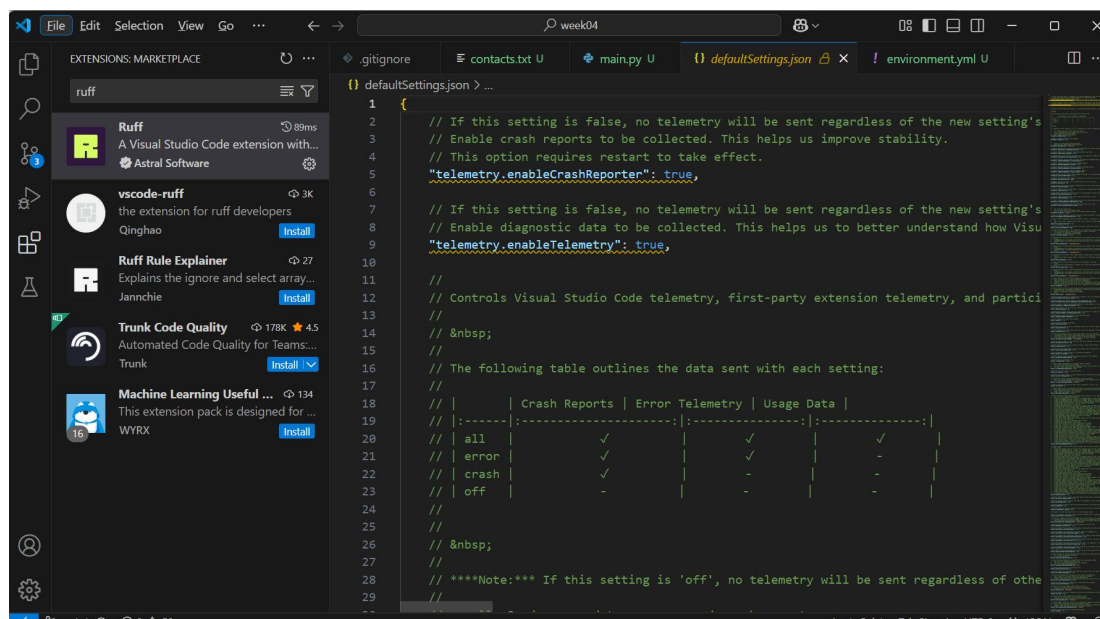


```
1 def process_contacts():
2     try:
3         for contact in contacts:
4             name, gender, email = parts
5             contact_list.append((name, gender, email))
6
7             contact_list.sort(key=lambda x: (x[2].split('@')[1], x[2].split('@')[0]))
8
9             email_content = []
10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

6. 将大模型提供的代码复制粘贴进 main.py 文件，记得保存

```
.gitignore LICENSE contacts.txt U main.py U ! environment.yml U
main.py
1 def process_contacts():
2     try:
3         for contact in contacts:
4             if len(parts) != 3:
5                 print(f"错误: 行 '{contact.strip()}' 格式不正确, 跳过此条记录。")
6                 continue
7             name, gender, email = parts
8             contact_list.append((name, gender, email))
9
10            contact_list.sort(key=lambda x: (x[2].split('@')[1], x[2].split('@')[0]))
11
12            email_content = []
13            for name, gender, email in contact_list:
14                title = '先生' if gender == '男' else '女士'
15                email_content.extend([
16                    f'to: <{email}>',
17                    f'尊敬的{name}{title}, 您的会员资格即将到期, 请及时续费。',
18                    '---'
19                ])
20
21            if email_content:
22                email_content.pop()
23
24            with open('emails.txt', 'w', encoding='utf-8') as file:
25                file.write('\n'.join(email_content))
26
27            except FileNotFoundError:
28                print("错误: 未找到 'contacts.txt' 文件。")
29            except Exception as e:
30                print(f"发生未知错误: {e}")
```

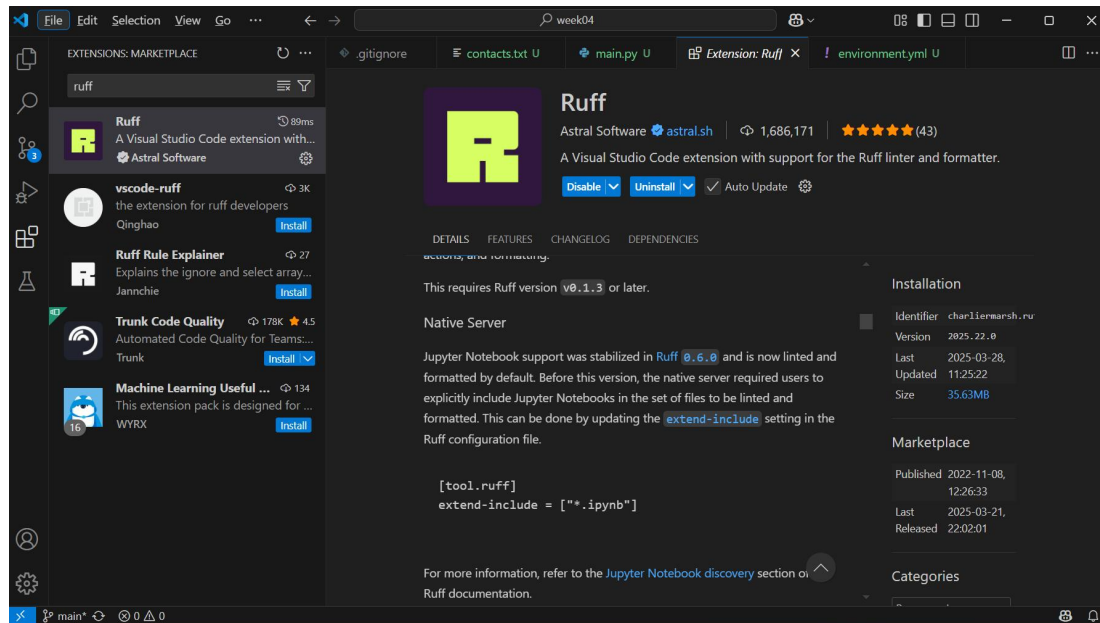
7. 在 VS Code 扩展商店里安装 Python 扩展, 使得在编写 py 文件时能够显示和选择 Python 解释器 (需要绕过防火墙)



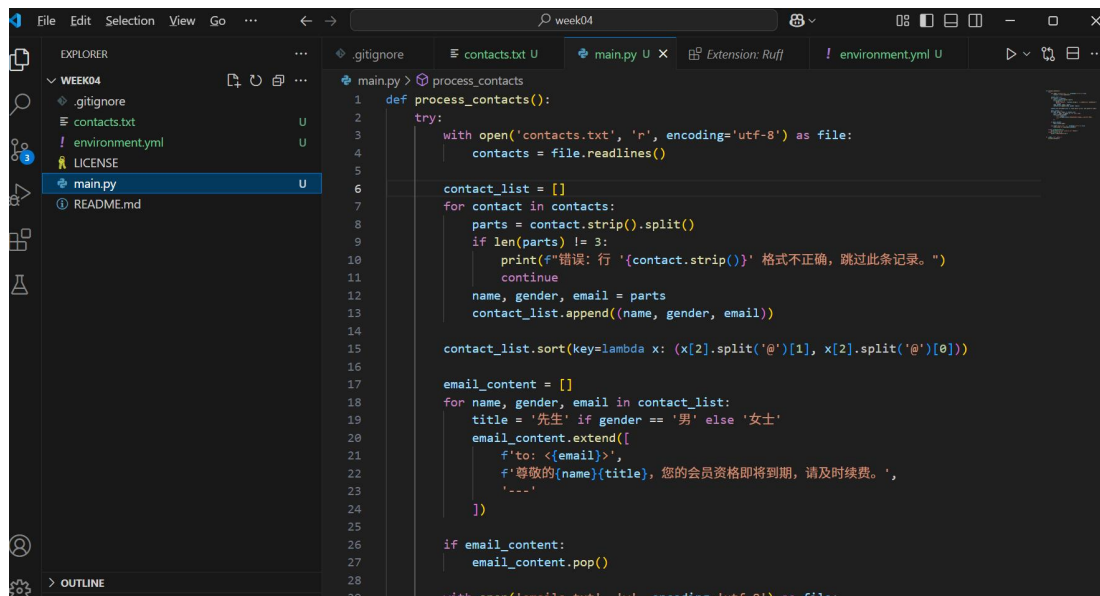
8. 在 VS Code 扩展商店里安装 Ruff 扩展, 按照文档配置 Ruff, 实现



在保存 py 文件时能够自动规范化 Python 代码



9. 运行 `python main.py` 命令 (作用是启动 Python 解释器，执行 `main.py` 里的代码直至结束 (EOF) 或报错 (Exception))，检查运行结果是否符合预期



10. 运行 `python -m pdb main.py` 命令 (作用是以调试模式 (debug mode) 启动 Python 解释器，准备执行 `main.py` 里的代码)

```
MINGW64:/c/Users/86153/rep  ×  +  v
-rw-r--r-- 1 86153 197609    72  3月 28 10:59 environment.yml
-rw-r--r-- 1 86153 197609 18805  3月 28 10:36 LICENSE
-rw-r--r-- 1 86153 197609  1349  3月 28 11:14 main.py
-rw-r--r-- 1 86153 197609  2239  3月 28 10:36 README.md

(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week04 (main)
$ cat email.txt
cat: email.txt: No such file or directory

(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week04 (main)
$ cat emails.txt
cat: emails.txt: No such file or directory

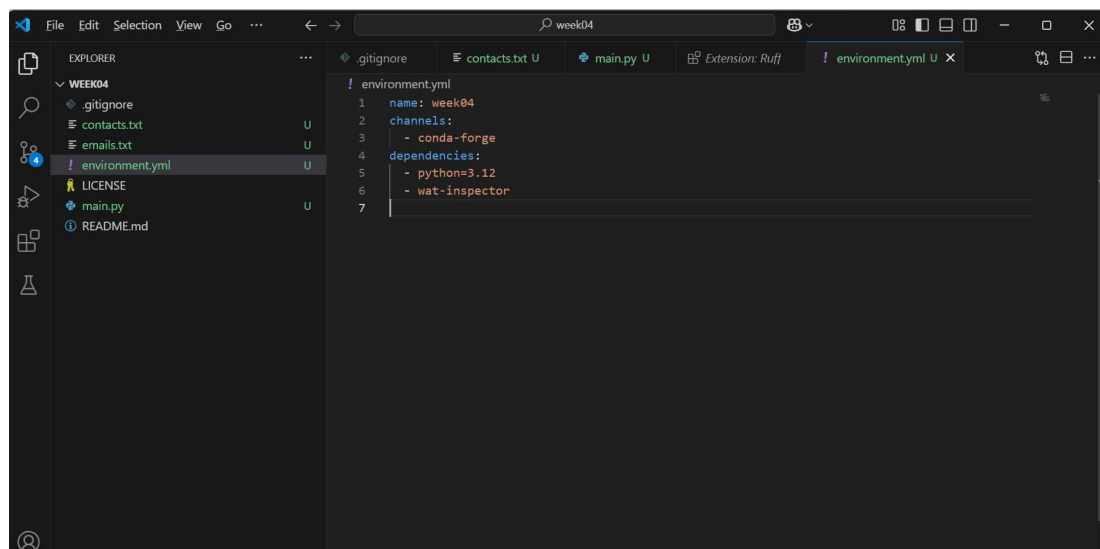
(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week04 (main)
$ python -m pdb main.py
> c:\users\86153\repo\week04\main.py(1)<module>()
-> def process_contacts():
(Pdb) l
1  -> def process_contacts():
2      try:
3          with open('contacts.txt', 'r', encoding='utf-8') as file:
4              contacts = file.readlines()
5
6          contact_list = []
7          for contact in contacts:
8              parts = contact.strip().split()
9              if len(parts) != 3:
10                 print(f"错误: 行 '{contact.strip()}' 格式不正确, 跳过此条记录。")
11                 continue
(Pdb) |
```

11. 在(pdb)提示符下练习使用 l(显示代码)、n(执行当前行)、p(打印表达式)、s 步入调用)、pp(美观打印)、c(继续执行) 等命令 (参考文档)

```
MINGW64:/c/Users/86153/rep  ×  +  v
> c:\users\86153\repo\week04\main.py(1)<module>()
-> def process_contacts():
(Pdb) l
1  -> def process_contacts():
2      try:
3          with open('contacts.txt', 'r', encoding='utf-8') as file:
4              contacts = file.readlines()
5
6          contact_list = []
7          for contact in contacts:
8              parts = contact.strip().split()
9              if len(parts) != 3:
10                 print(f"错误: 行 '{contact.strip()}' 格式不正确, 跳过此条记录。")
11                 continue
(Pdb) n
> c:\users\86153\repo\week04\main.py(38)<module>()
-> if __name__ == "__main__":
(Pdb) l
33     print("错误: 未找到 'contacts.txt' 文件。")
34     except Exception as e:
35         print(f"发生未知错误: {e}")
36
37
38 -> if __name__ == "__main__":
39     process_contacts()
40
[EOF]
(Pdb) ll
1     def process_contacts():
2         try:
```

```
MINGW64/c/Users/86153/rep x + v
23     '---'
24     })
25
26     if email_content:
27         email_content.pop()
28
29     with open('emails.txt', 'w', encoding='utf-8') as file:
30         file.write('\n'.join(email_content))
31
32     except FileNotFoundError:
33         print("错误: 未找到 'contacts.txt' 文件。")
34     except Exception as e:
35         print(f"发生未知错误: {e}")
36
37
38 -> if __name__ == "__main__":
39     process_contacts()
40
(Pdb) ps
*** NameError: name 'ps' is not defined
(Pdb) s
> c:\users\86153\repo\week04\main.py(39)<module>()
-> process_contacts()
(Pdb) pp
*** SyntaxError: invalid syntax
(Pdb) c
The program finished and will be restarted
> c:\users\86153\repo\week04\main.py(1)<module>()
-> def process_contacts():
(Pdb) |
```

12. 在调试过程中，利用 wat-inspector(第三方软件包，需要安装) 检查 (inspect) 各种对象 (参考文档)



13. 在调试过程中，观察代码逐步运行的效果，学习理解以下 Python 基本概念, Python 语法保留字 (reserved key words); 语句 (statement) 和表达式 (expression); 缩进 (indent); 局部变量 (local variable) vs. 全局变量 (global variable); 函数 (function) 的定义 (define) 和调用 (call); 字面值 (literal) (字符串 (str)、整数 (int)、列表 (list)、字典 (dict)、



元组 (tuple)); 运算符 (operator); 形参 (parameter)、实参 (argument)、返回值 (return value); 对象 (object)、类型 (type)、属性 (attribute)、方法 (method)

```
MINGW64/c/Users/86153/rep  X + v
drwxr-xr-x 1 86153 197609 0 3月 16 13:28 week02/
drwxr-xr-x 1 86153 197609 0 3月 20 20:57 week03/
drwxr-xr-x 1 86153 197609 0 3月 28 11:53 week04/

(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~/repo
$ cd week04/

(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week04 (main)
$ python -m pdb main.py
> c:\users\86153\repo\week04\main.py(1)<module>()
-> def process_contacts():
(Pdb) l
1  -> def process_contacts():
2      try:
3          with open('contacts.txt', 'r', encoding='utf-8') as file:
4              contacts = file.readlines()
5
6              contact_list = []
7              for contact in contacts:
8                  parts = contact.strip().split()
9                  if len(parts) != 3:
10                     print(f"错误: 行 '{contact.strip()}' 格式不正确, 跳过此条记录。")
11                     continue
(Pdb) wat
*** NameError: name 'wat' is not defined
(Pdb) import wat
*** ModuleNotFoundError: No module named 'wat'
(Pdb) wat()
```

```
MINGW64/c/Users/86153/rep  X + v
39      process_contacts()
40
[EOF]
(Pdb) l
[EOF]
(Pdb) n
> c:\users\86153\repo\week04\main.py(39)<module>()
-> process_contacts()
(Pdb) l .
34      except Exception as e:
35          print(f"发生未知错误: {e}")
36
37
38      if __name__ == "__main__":
39  ->      process_contacts()
40
[EOF]
(Pdb) n
--Return--
> c:\users\86153\repo\week04\main.py(39)<module>()->None
-> process_contacts()
(Pdb) l
34      except Exception as e:
35          print(f"发生未知错误: {e}")
36
37
38      if __name__ == "__main__":
39  ->      process_contacts()
40
[EOF]
```