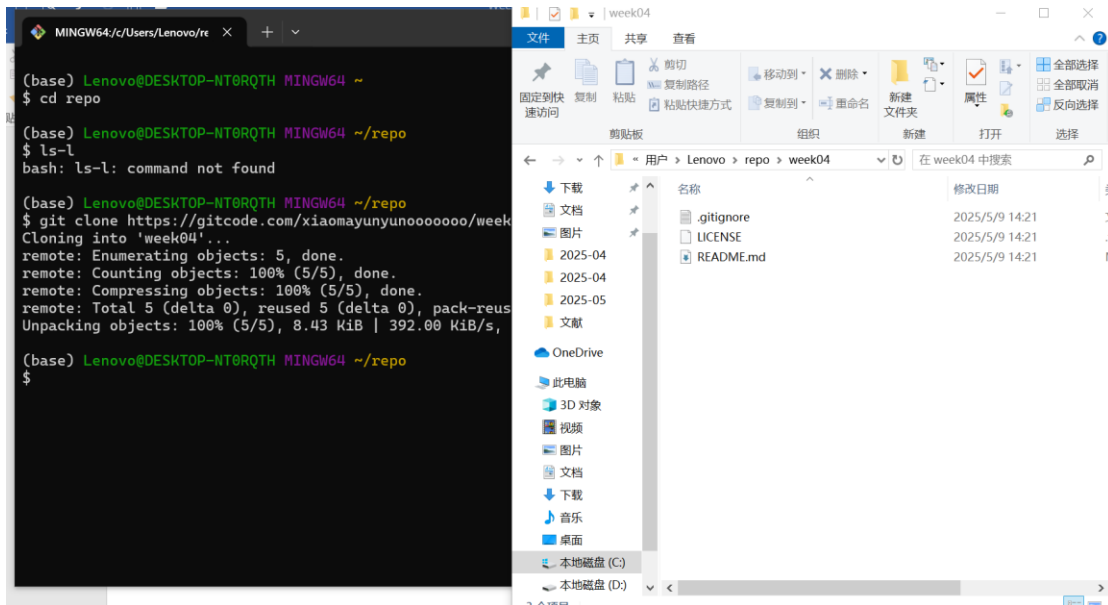
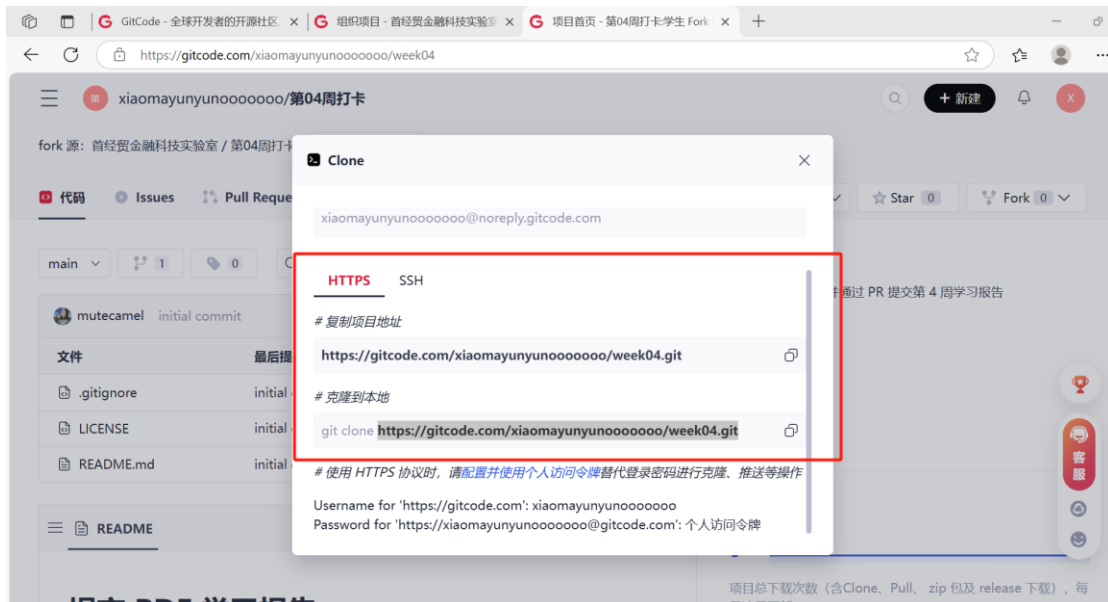


## Week04 学习笔记

Fork 第 04 周打卡 仓库至你的名下，然后将你名下的这个仓库 Clone 到你的本地计算机：



查看远程仓库地址：

```
MINGW64/Users/Lenovo/re > $ git clone https://gitcode.com/xiaomayunyunooooooo/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 | 392.00 KiB/s, done.

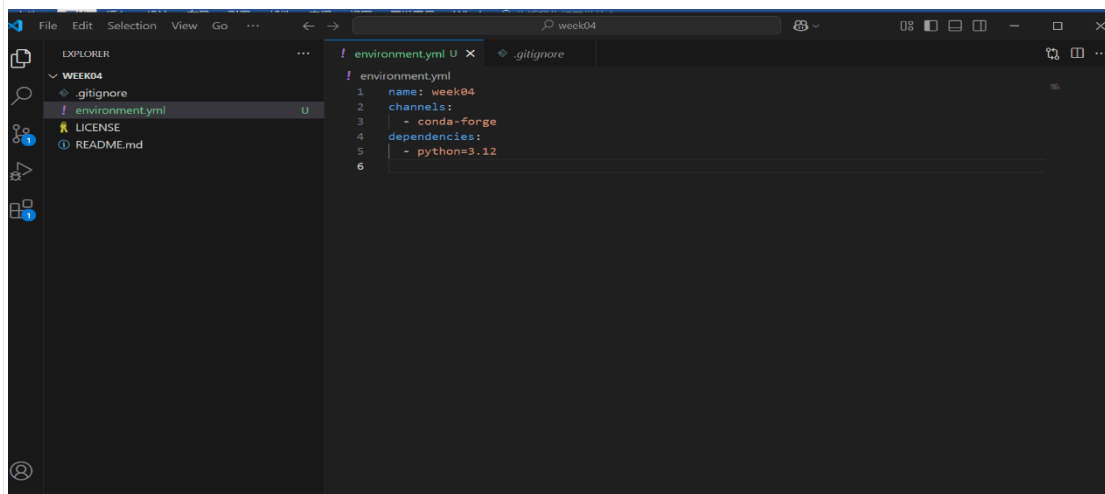
(base) Lenovo@DESKTOP-NT0RQTH MINGW64 ~/repo
$ cd week04/

(base) Lenovo@DESKTOP-NT0RQTH MINGW64 ~/repo/week04 (main)
$ pwd
/c/Users/Lenovo/repo/week04

(base) Lenovo@DESKTOP-NT0RQTH MINGW64 ~/repo/week04 (main)
$ git remote show origin
* remote origin
  Fetch URL: https://gitcode.com/xiaomayunyunooooooo/week04.git
  Push URL: https://gitcode.com/xiaomayunyunooooooo/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) Lenovo@DESKTOP-NT0RQTH MINGW64 ~/repo/week04 (main)
$
```

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：

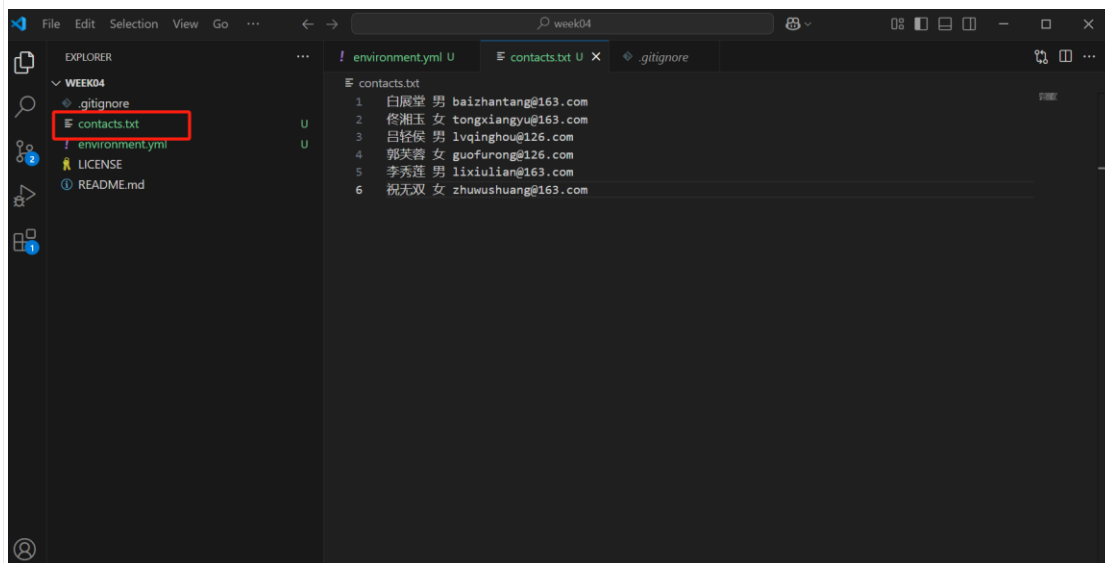


The screenshot shows the VS Code editor with the 'environment.yml' file open. The file content is as follows:

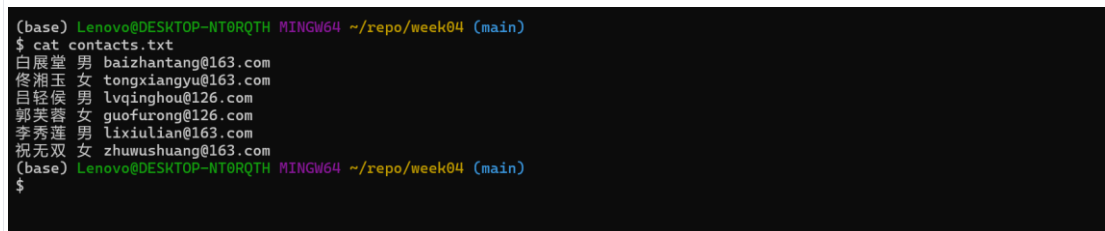
```
environment.yml
1 name: week04
2 channels:
3   - conda-forge
4 dependencies:
5   - python=3.12
6
```

```
(base) Lenovo@DESKTOP-NT0RQTH MINGW64 ~/repo/week04 (main)
$ cat environment.yml
name: prj1
channels:
- conda-forge
- https://repo.anaconda.com/pkgs/main
- https://repo.anaconda.com/pkgs/r
- https://repo.anaconda.com/pkgs/msys2
dependencies:
- brotli-python=1.1.0=py312h275cf98_2
- bzip2=1.0.8=h2466b09_7
- ca-certificates=2025.1.31=h56e8100_0
- certifi=2025.1.31=pyhd8ed1ab_0
- cffi=1.17.1=py312h4389bb4_0
- charset-normalizer=3.4.1=pyhd8ed1ab_0
- h2=4.2.0=pyhd8ed1ab_0
- hpack=4.1.0=pyhd8ed1ab_0
- hyperframe=6.1.0=pyhd8ed1ab_0
- idna=3.10=pyhd8ed1ab_1
- intel-openmp=2024.2.1=h57928b3_1083
- libblas=3.9.0=31_h641d27c_mkl
- libcbblas=3.9.0=31_h5e41251_mkl
- libexpat=2.7.0=he0c23c2_0
- libffi=3.4.6=h537db12_1
- libhwloc=2.11.2=default_ha69328c_1001
- libiconv=1.18=h135ad9c_1
- liblapack=3.9.0=31_h1aa476e_mkl
- liblzma=5.6.4=h2466b09_0
- libsqlite=3.49.1=h67fdade_2
```

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：

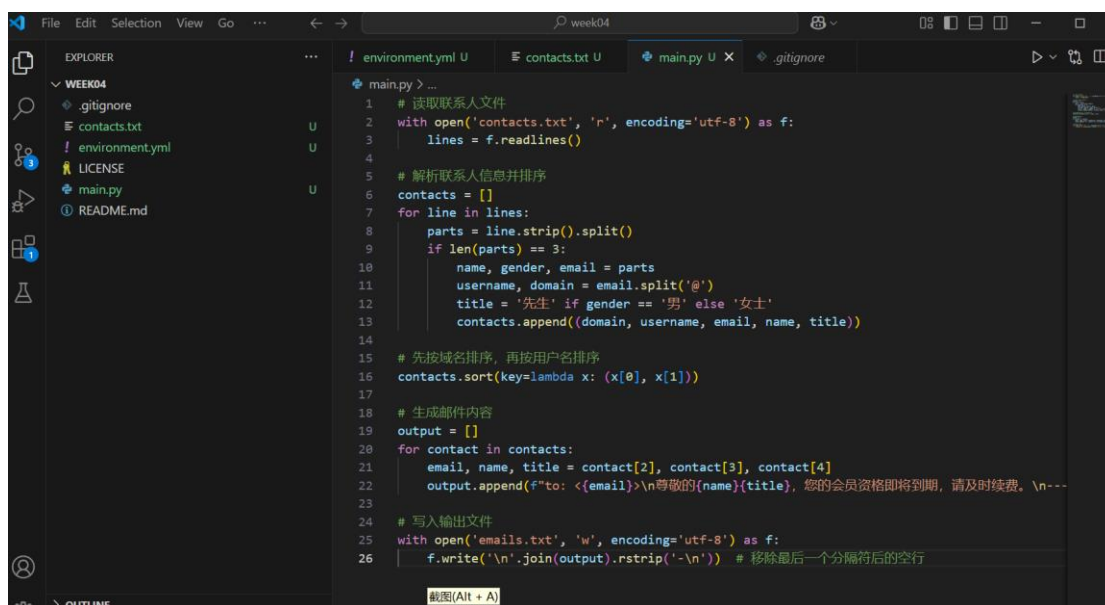


```
contacts.txt
1 白展堂 男 baizhantang@163.com
2 佟湘玉 女 tongxiangyu@163.com
3 吕轻侯 男 lvqinghou@126.com
4 郭芙蓉 女 guofurong@126.com
5 李秀莲 男 lixiulian@163.com
6 祝无双 女 zhuwushuang@163.com
```



```
(base) Lenovo@DESKTOP-NT0RQTH 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
祝无双 女 zhuwushuang@163.com
(base) Lenovo@DESKTOP-NT0RQTH MINGW64 ~/repo/week04 (main)
$
```

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：



```
main.py
1 # 读取联系人文件
2 with open('contacts.txt', 'r', encoding='utf-8') as f:
3     lines = f.readlines()
4
5 # 解析联系人信息并排序
6 contacts = []
7 for line in lines:
8     parts = line.strip().split()
9     if len(parts) == 3:
10         name, gender, email = parts
11         username, domain = email.split('@')
12         title = '先生' if gender == '男' else '女士'
13         contacts.append((domain, username, email, name, title))
14
15 # 先按域名排序，再按用户名排序
16 contacts.sort(key=lambda x: (x[0], x[1]))
17
18 # 生成邮件内容
19 output = []
20 for contact in contacts:
21     email, name, title = contact[2], contact[3], contact[4]
22     output.append(f"to: <{email}>\n尊敬的{name}{title}，您的会员资格即将到期，请及时续费。")
23
24 # 写入输出文件
25 with open('emails.txt', 'w', encoding='utf-8') as f:
26     f.write('\n'.join(output).rstrip('\n')) # 移除最后一个分隔符后的空行
```

```
(base) Lenovo@DESKTOP-NT0RQTH MINGW64 ~/repo/week04 (main)
$ conda activate week04
(week04)
Lenovo@DESKTOP-NT0RQTH MINGW64 ~/repo/week04 (main)
$ ls-l
bash: ls-l: command not found
(week04)
Lenovo@DESKTOP-NT0RQTH 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
祝无双 女 zhuwushuang@163.com
(week04)
Lenovo@DESKTOP-NT0RQTH MINGW64 ~/repo/week04 (main)
$ cat main.py
# 读取联系人文件
with open('contacts.txt', 'r', encoding='utf-8') as f:
    lines = f.readlines()

# 解析联系人信息并排序
contacts = []
for line in lines:
    parts = line.strip().split()
    if len(parts) == 3:
        name, gender, email = parts
        username, domain = email.split('@')
        title = '先生' if gender == '男' else '女士'
        contacts.append((domain, username, email, name, title))
```

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:

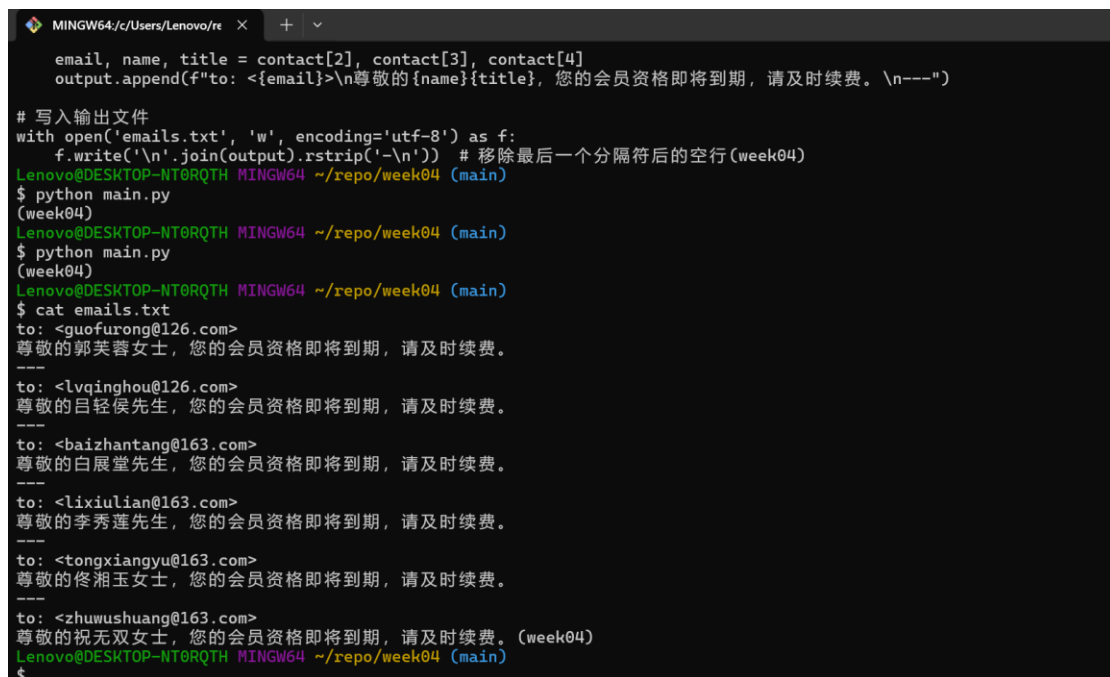
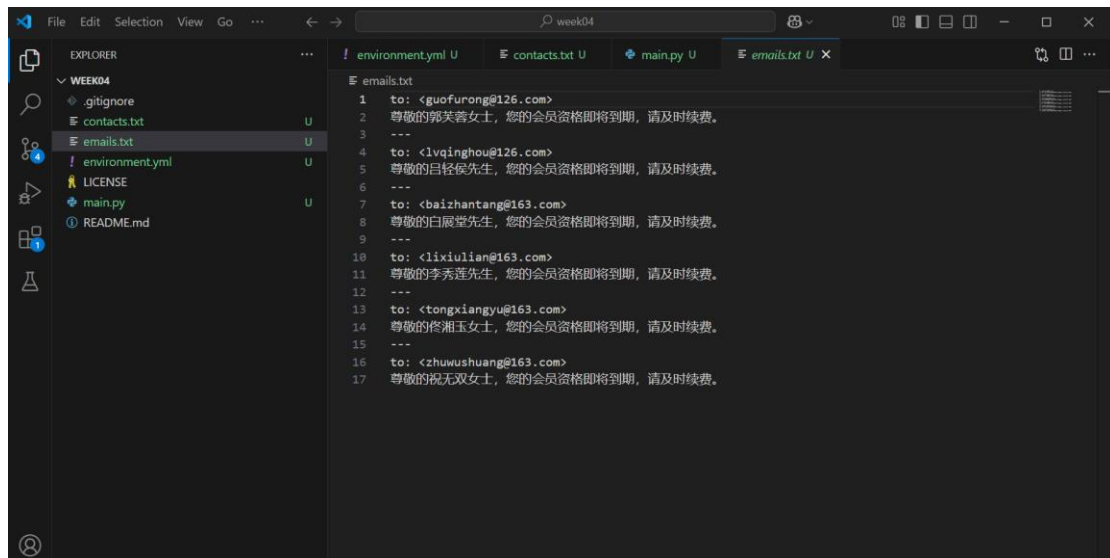
```
MINGW64:/c/Users/Lenovo/re  ×  +  ▾
with open('contacts.txt', 'r', encoding='utf-8') as f:
    lines = f.readlines()

# 解析联系人信息并排序
contacts = []
for line in lines:
    parts = line.strip().split()
    if len(parts) == 3:
        name, gender, email = parts
        username, domain = email.split('@')
        title = '先生' if gender == '男' else '女士'
        contacts.append((domain, username, email, name, title))

# 先按域名排序, 再按用户名排序
contacts.sort(key=lambda x: (x[0], x[1]))

# 生成邮件内容
output = []
for contact in contacts:
    email, name, title = contact[2], contact[3], contact[4]
    output.append(f"to: <{email}>\n尊敬的{name}{title}, 您的会员资格即将到期, 请及时续费.\n---")

# 写入输出文件
with open('emails.txt', 'w', encoding='utf-8') as f:
    f.write('\n'.join(output).rstrip('\n')) # 移除最后一个分隔符后的空行(week04)
Lenovo@DESKTOP-NT0RQTH MINGW64 ~/repo/week04 (main)
$ python main.py
(week04)
Lenovo@DESKTOP-NT0RQTH MINGW64 ~/repo/week04 (main)
$ python main.py
(week04)
Lenovo@DESKTOP-NT0RQTH MINGW64 ~/repo/week04 (main)
$
```



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/Lenovo/re x + v
to: <lixiluan@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。(week04)
Lenovo@DESKTOP-NT0RQTH MINGW64 ~/repo/week04 (main)
$ ls-l
bash: ls-l: command not found
(week04)
Lenovo@DESKTOP-NT0RQTH MINGW64 ~/repo/week04 (main)
$ rm emails.txt
(week04)
Lenovo@DESKTOP-NT0RQTH MINGW64 ~/repo/week04 (main)
$ python -m pdb main.py
> c:\users\lenovo\repo\week04\main.py(2)<module>()
-> with open('contacts.txt', 'r', encoding='utf-8') as f:
(Pdb)
(Pdb) l
1      # 读取联系人文件
2      -> with open('contacts.txt', 'r', encoding='utf-8') as f:
3          lines = f.readlines()
4
5      # 解析联系人信息并排序
6      contacts = []
7      for line in lines:
8          parts = line.strip().split()
9          if len(parts) == 3:
10             name, gender, email = parts
11             username, domain = email.split('@')
(Pdb)
```

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/Lenovo/re x + v
to: <lixiluan@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。(week04)
Lenovo@DESKTOP-NT0RQTH MINGW64 ~/repo/week04 (main)
$ ls-l
bash: ls-l: command not found
(week04)
Lenovo@DESKTOP-NT0RQTH MINGW64 ~/repo/week04 (main)
$ rm emails.txt
(week04)
Lenovo@DESKTOP-NT0RQTH MINGW64 ~/repo/week04 (main)
$ python -m pdb main.py
> c:\users\lenovo\repo\week04\main.py(2)<module>()
-> with open('contacts.txt', 'r', encoding='utf-8') as f:
(Pdb)
(Pdb) l
1      # 读取联系人文件
2      -> with open('contacts.txt', 'r', encoding='utf-8') as f:
3          lines = f.readlines()
4
5      # 解析联系人信息并排序
6      contacts = []
7      for line in lines:
8          parts = line.strip().split()
9          if len(parts) == 3:
10             name, gender, email = parts
11             username, domain = email.split('@')
(Pdb)
```

```
MINGW64/c/Users/Lenovo/re x + v
lenovo@DESKTOP-NT0RQTH MINGW64 ~/repo/week04 (main)
$ rm emails.txt
(lenovo@DESKTOP-NT0RQTH MINGW64 ~/repo/week04 (main))
$ python -m pdb main.py
> c:\users\lenovo\repo\week04\main.py(2)<module>()
-> with open('contacts.txt', 'r', encoding='utf-8') as f:
(Pdb)
(Pdb) l
1 # 读取联系人文件
2 -> with open('contacts.txt', 'r', encoding='utf-8') as f:
3     lines = f.readlines()
4
5 # 解析联系人信息并排序
6 contacts = []
7 for line in lines:
8     parts = line.strip().split()
9     if len(parts) == 3:
10         name, gender, email = parts
11         username, domain = email.split('@')
(Pdb) l
12         title = '先生' if gender == '男' else '女士'
13         contacts.append((domain, username, email, name, title))
14
15 # 先按域名排序, 再按用户名排序
16 contacts.sort(key=lambda x: (x[0], x[1]))
17
18 # 生成邮件内容
19 output = []
20 for contact in contacts:
21     email, name, title = contact[2], contact[3], contact[4]
22     output.append(f"to: <{email}>\n尊敬的{name}{title}, 您的会员资格即将到期, 请及时续费。\\n---")
(Pdb)
```

```
MINGW64/c/Users/Lenovo/re x + v
18 # 生成邮件内容
19 output = []
20 for contact in contacts:
21     email, name, title = contact[2], contact[3], contact[4]
22     output.append(f"to: <{email}>\n尊敬的{name}{title}, 您的会员资格即将到期, 请及时续费。\\n---")
(Pdb) ll
1 # 读取联系人文件
2 -> with open('contacts.txt', 'r', encoding='utf-8') as f:
3     lines = f.readlines()
4
5 # 解析联系人信息并排序
6 contacts = []
7 for line in lines:
8     parts = line.strip().split()
9     if len(parts) == 3:
10         name, gender, email = parts
11         username, domain = email.split('@')
12         title = '先生' if gender == '男' else '女士'
13         contacts.append((domain, username, email, name, title))
14
15 # 先按域名排序, 再按用户名排序
16 contacts.sort(key=lambda x: (x[0], x[1]))
17
18 # 生成邮件内容
19 output = []
20 for contact in contacts:
21     email, name, title = contact[2], contact[3], contact[4]
22     output.append(f"to: <{email}>\n尊敬的{name}{title}, 您的会员资格即将到期, 请及时续费。\\n---")
23
24 # 写入输出文件
25 with open('emails.txt', 'w', encoding='utf-8') as f:
26     f.write('\\n'.join(output).rstrip('\\n')) # 移除最后一个分隔符后的空行
```

```
MINGW64/c/Users/Lenovo/re  X + v

List source code for the current file. Without arguments,
list 11 lines around the current line or continue the previous
listing. With . as argument, list 11 lines around the current
line. With one argument, list 11 lines starting at that line.
With two arguments, list the given range; if the second
argument is less than the first, it is a count.

The current line in the current frame is indicated by "->".
If an exception is being debugged, the line where the
exception was originally raised or propagated is indicated by
">>", if it differs from the current line.
(Pdb) l .
304     """
305     This subclass of IncrementalDecoder can be used as the baseclass for an
306     incremental decoder if the decoder must be able to handle incomplete
307     byte sequences.
308     """
309     -> def __init__(self, errors='strict'):
310         IncrementalDecoder.__init__(self, errors)
311         # undecoded input that is kept between calls to decode()
312         self.buffer = b""
313
314     def _buffer_decode(self, input, errors, final):
(Pdb) l 30,39
30         "strict_errors", "ignore_errors", "replace_errors",
31         "xmlcharrefreplace_errors",
32         "backslashreplace_errors", "namereplace_errors",
33         "register_error", "lookup_error"]
34
35     ### Constants
36
37     #
38     # Byte Order Mark (BOM = ZERO WIDTH NO-BREAK SPACE = U+FEFF)
```

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:

```
MINGW64/c/Users/Lenovo/re  X + v

$ rm emails.txt
(week04)
Lenovo@DESKTOP-NT0RQTH MINGW64 ~/repo/week04 (main)
$ python -m pdb main.py
> c:\users\lenovo\repo\week04\main.py(2)<module>()
-> with open('contacts.txt', 'r', encoding='utf-8') as f:
(Pdb)
(Pdb) l
1     # 读取联系人文件
2     -> with open('contacts.txt', 'r', encoding='utf-8') as f:
3         lines = f.readlines()
4
5     # 解析联系人信息并排序
6     contacts = []
7     for line in lines:
8         parts = line.strip().split()
9         if len(parts) == 3:
10            name, gender, email = parts
11            username, domain = email.split('@')
(Pdb) l
12            title = '先生' if gender == '男' else '女士'
13            contacts.append((domain, username, email, name, title))
14
15    # 先按域名排序, 再按用户名排序
16    contacts.sort(key=lambda x: (x[0], x[1]))
17
18    # 生成邮件内容
19    output = []
20    for contact in contacts:
21        email, name, title = contact[2], contact[3], contact[4]
22        output.append(f"to: <{email}>\n尊敬的{name}{title}, 您的会员资格即将到期, 请及时续费。 \n---")
(Pdb) ll
1     # 读取联系人文件
```