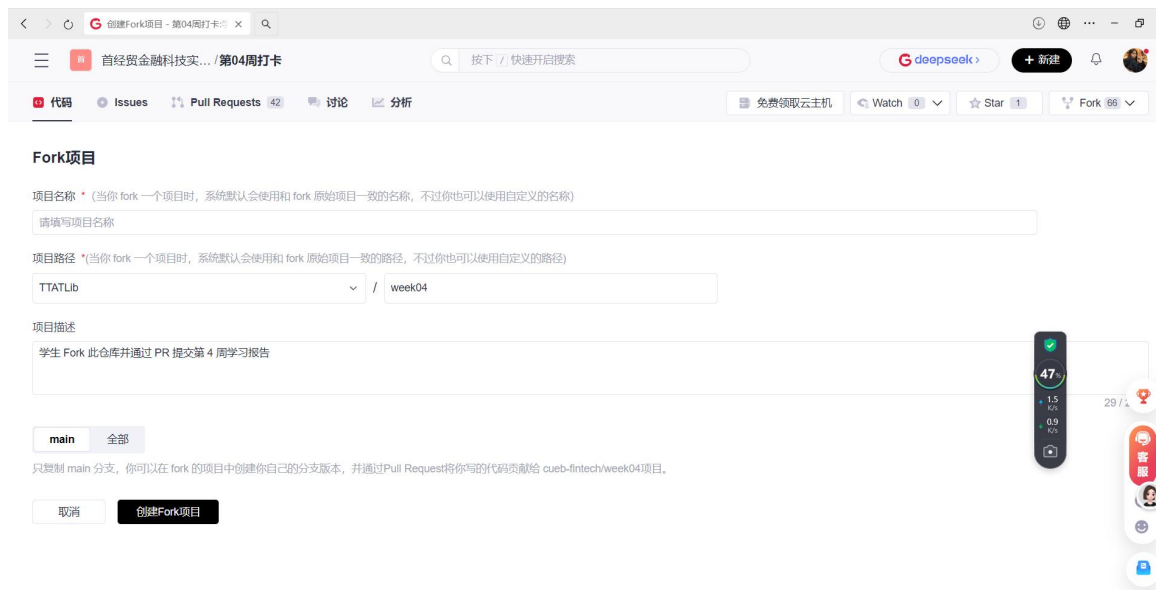


金融编程与计算

Week4——Python 数据类型（初级）

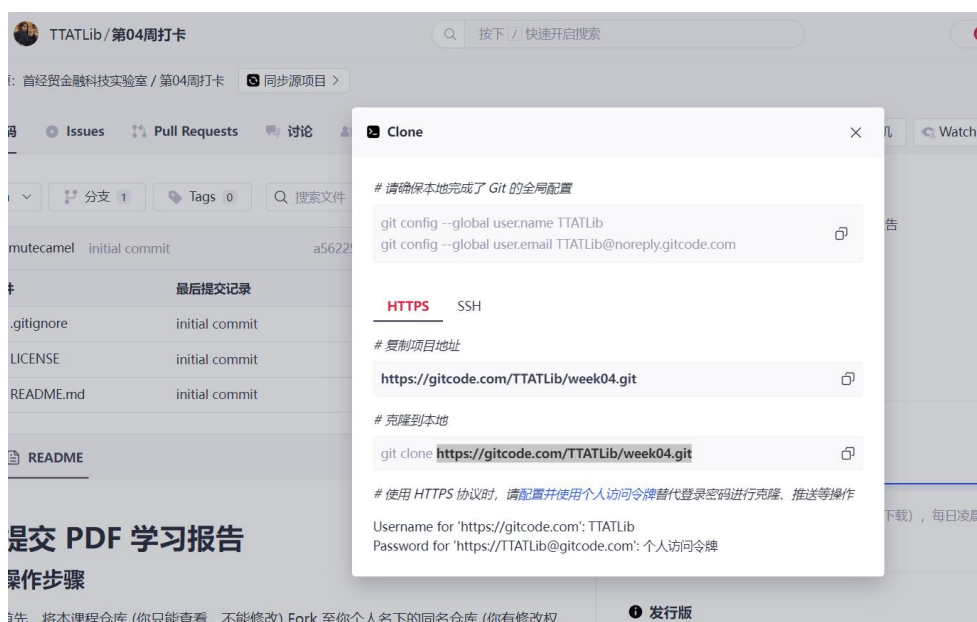
一、Fork 第 04 周打卡 仓库至你的名下，然后将你名下的这个仓库 Clone 到你的本地计算机

（一）fork



直接点创建

（二）Clone



```

(base) Administrator@PC-20220115W0JX MINGW64 ~
$ cd repo

(base) Administrator@PC-20220115W0JX MINGW64 ~/repo
$ ls -l
total 20
drwxr-xr-x 1 Administrator 197121 0 3月 24 01:38 myproject/
drwxr-xr-x 1 Administrator 197121 0 3月 17 01:10 mywork/
drwxr-xr-x 1 Administrator 197121 0 3月 23 00:57 prj1/
-rw-r--r-- 1 Administrator 197121 0 3月 10 00:44 script1.py
drwxr-xr-x 1 Administrator 197121 0 3月 10 02:18 week01/
drwxr-xr-x 1 Administrator 197121 0 3月 17 01:17 week02/
drwxr-xr-x 1 Administrator 197121 0 3月 24 02:34 week03/
drwxr-xr-x 1 Administrator 197121 0 3月 19 21:03 work/

(base) Administrator@PC-20220115W0JX MINGW64 ~/repo
$ git clone https://gitcode.com/TTATLib/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 | 278.00 KiB/s, done.

```

打开文件夹：week04 已 clone 过来

› 此电脑 › Win10 (C:) › Users › Administrator › repo

| 名称 | 修改日期 | 类型 | 大小 |
|-----------|-----------------|------------|------|
| myproject | 2025/3/24 1:38 | 文件夹 | |
| mywork | 2025/3/17 1:10 | 文件夹 | |
| prj1 | 2025/3/23 0:57 | 文件夹 | |
| week01 | 2025/3/10 2:18 | 文件夹 | |
| week02 | 2025/3/17 1:17 | 文件夹 | |
| week03 | 2025/3/24 2:34 | 文件夹 | |
| week04 | 2025/3/30 16:06 | 文件夹 | |
| work | 2025/3/19 21:03 | 文件夹 | |
| script1 | 2025/3/10 0:44 | Python 源文件 | 0 KB |

git remote show origin: 查看远程的地址

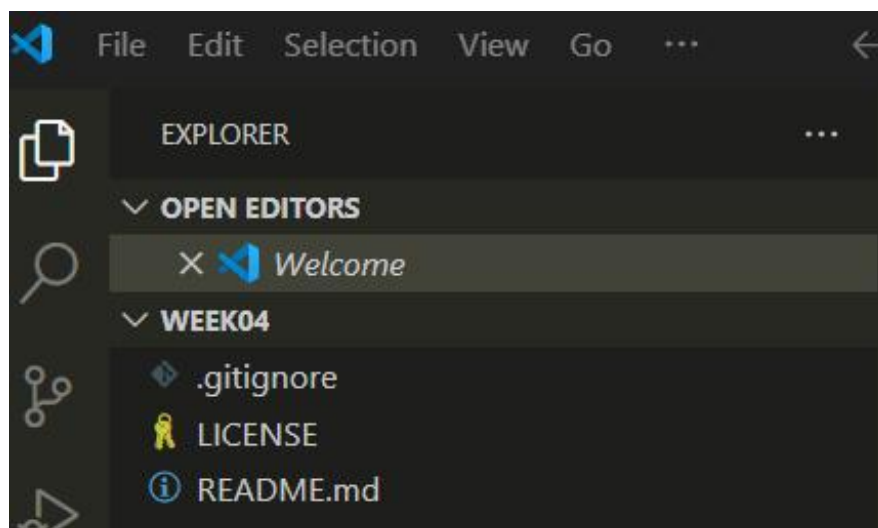
```
(base) Administrator@PC-20220115W0JX MINGW64 ~/repo
$ cd week04/

(base) Administrator@PC-20220115W0JX MINGW64 ~/repo/week04 (main)
$ pwd
/c/Users/Administrator/repo/week04

(base) Administrator@PC-20220115W0JX MINGW64 ~/repo/week04 (main)
$ git remote show origin
* remote origin
  Fetch URL: https://gitcode.com/TTATLib/week04.git
  Push URL: https://gitcode.com/TTATLib/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)
```

二、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

Vscode 里手动打开 week04 文件夹



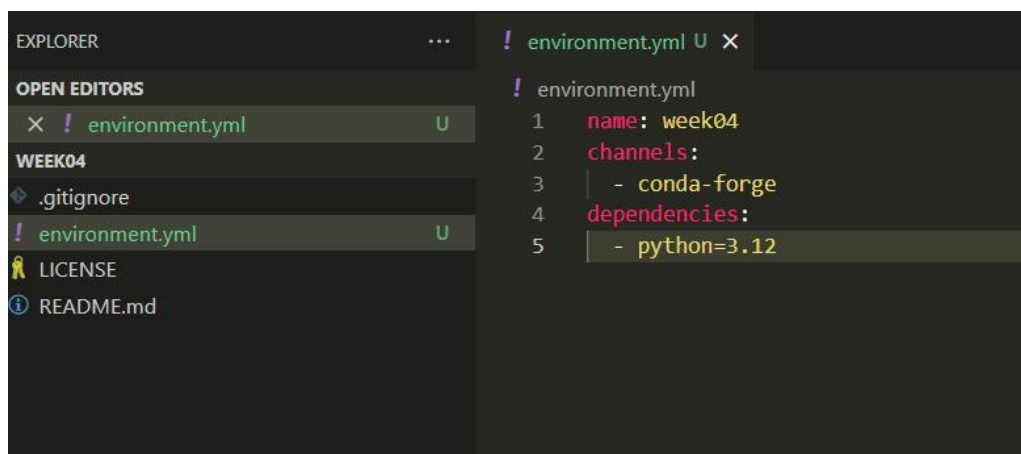
(一) 之前创建过 **environment.yml**，无需新建，复制到当前文件夹即可：

```
(base) Administrator@PC-20220115W0JX MINGW64 ~/repo/week04 (main)
$ ls -l ../myproject
total 52114
-rw-r--r-- 1 Administrator 197121      87  3月 24 01:28 environment.yml
-rw-r--r-- 1 Administrator 197121 53362688  3月 24 01:53 EPA_SmartLocationDatabase_V3_Jan_2021_Final.csv
-rw-r--r-- 1 Administrator 197121      713  3月 24 01:52 main.py

(base) Administrator@PC-20220115W0JX MINGW64 ~/repo/week04 (main)
$ cat ../myproject/environment.yml
name: myproject
channels:
  - conda-forge
dependencies:
  - python=3.12
  - pandas

(base) Administrator@PC-20220115W0JX MINGW64 ~/repo/week04 (main)
$ cp ../myproject/environment.yml ./
```

改名字，name 改为 week04



(二) 运行 **conda env create** 命令创建 Conda 环境

```
(base) Administrator@PC-20220115W0JX MINGW64 ~/repo/week04 (main)
$ conda env create
Retrieving notices: done
Channels:
  - conda-forge
  - https://repo.anaconda.com/pkgs/main
  - https://repo.anaconda.com/pkgs/r
  - https://repo.anaconda.com/pkgs/msys2
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

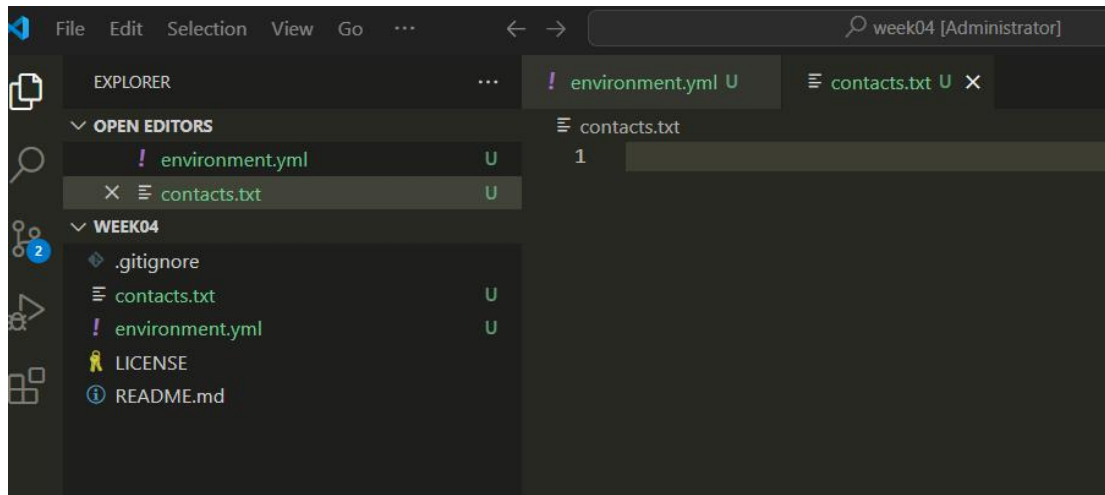
Downloading and Extracting Packages:

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate week04
#
# To deactivate an active environment, use
#
#     $ conda deactivate
```



三、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

(一) 在 VS code 里新建 `contacts.txt`



在终端里查看

```
(base) Administrator@PC-20220115W0JX MINGW64 ~/repo/week04 (49%)
$ ls -l
total 26
-rw-r--r-- 1 Administrator 197121 202 3月 30 16:39 contacts.txt
-rw-r--r-- 1 Administrator 197121 72 3月 30 16:22 environment.yml
-rw-r--r-- 1 Administrator 197121 18805 3月 30 16:06 LICENSE
-rw-r--r-- 1 Administrator 197121 2239 3月 30 16:06 README.md

(base) Administrator@PC-20220115W0JX 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) Administrator@PC-20220115W0JX MINGW64 ~/repo/week04 (main)
$
```

注：cat 是 contact 的缩写，拼接的意思

Unix 规范：文本最后要多加一个换行

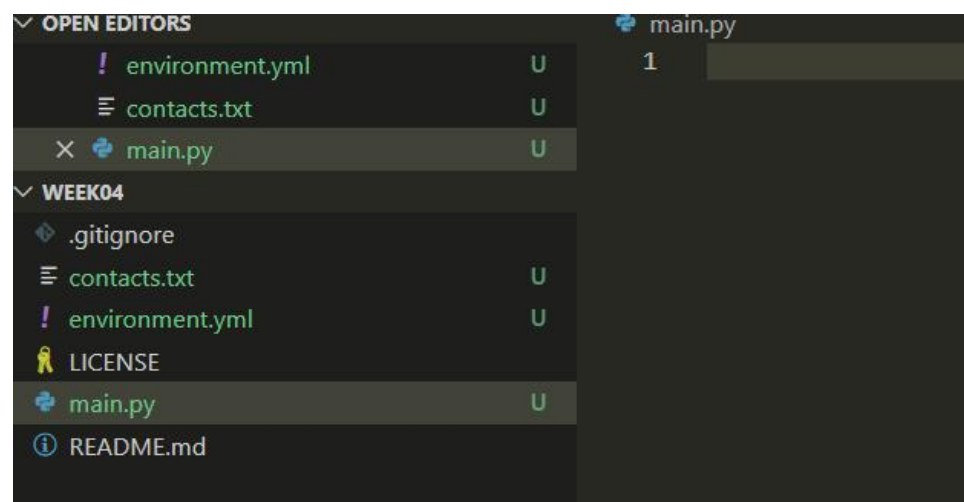
cat contacts.txt environment .yaml

把这两个文件拼接到一起

```
(base) Administrator@PC-20220115W0JX MINGW64 ~/repo/week04 (main)
$ cat contacts.txt environment.yml
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
name: week04
channels:
  - conda-forge
dependencies:
  - python=3.12
```

四、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

(一) 新建文件

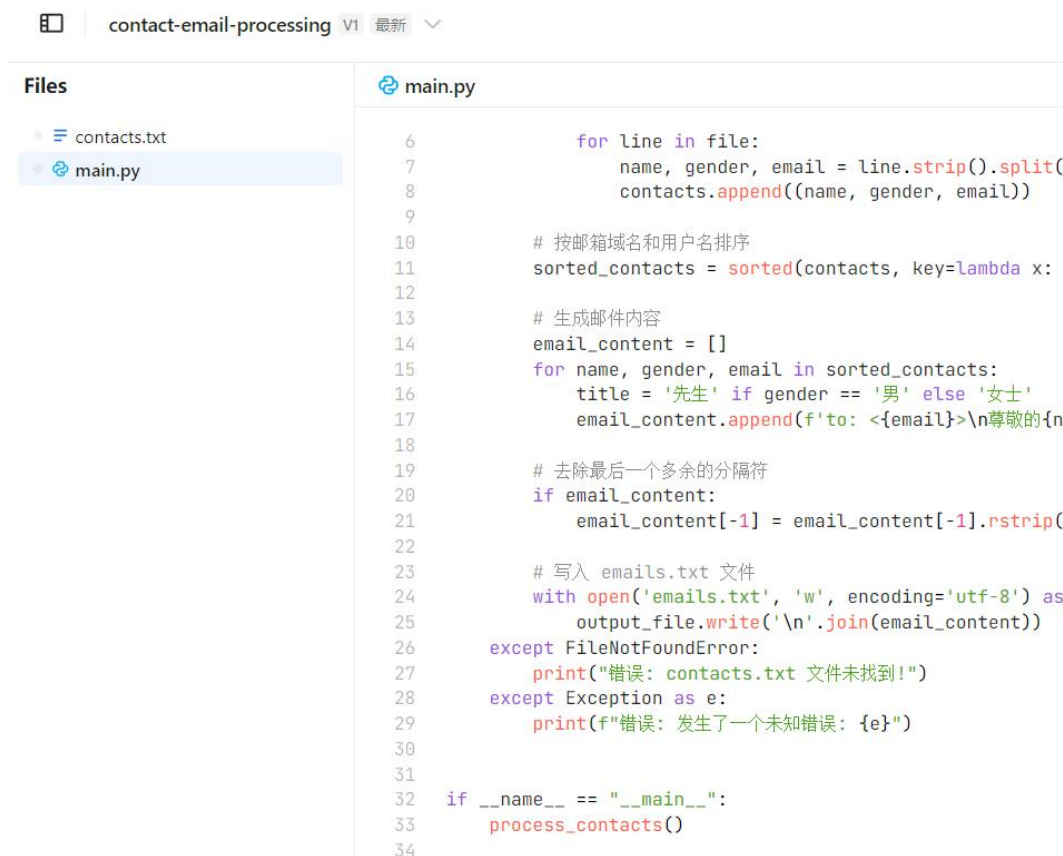


(二) 里面写 python 代码

```
! environment.yml U  contacts.txt U  main.py U  ●
main.py > ...
1  def process_contacts():
2      try:
3          # 读取 contacts.txt 文件
4          with open('contacts.txt', 'r', encoding='utf-8') as file:
5              contacts = []
6              for line in file:
7                  name, gender, email = line.strip().split()
8                  contacts.append((name, gender, email))
9
10         # 按邮箱域名和用户名排序
11         sorted_contacts = sorted(contacts, key=lambda x: (x[2].split('@')[1], x[2].split('@')[0]))
12
13         # 生成邮件内容
14         email_content = []
15         for name, gender, email in sorted_contacts:
16             title = '先生' if gender == '男' else '女士'
17             email_content.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')
18
19         # 去除最后一个多余的分隔符
20         if email_content:
21             email_content[-1] = email_content[-1].rstrip('---')
22
23         # 写入 emails.txt 文件
24         with open('emails.txt', 'w', encoding='utf-8') as output_file:
25             output_file.write('\n'.join(email_content))
26     except FileNotFoundError:
27         print("错误: contacts.txt 文件未找到!")
28     except Exception as e:
29         print(f"错误: 发生了一个未知错误: {e}")
30
31
32 if __name__ == "__main__":
33     process_contacts()
34
```

五、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

问豆包编写程序



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows two files: 'contacts.txt' and 'main.py'. The code editor shows the content of 'main.py', which is a Python script for processing contact emails. The script reads lines from a file, splits them into name, gender, and email, sorts them by email domain and name, generates email content, and writes it to 'emails.txt'. It includes error handling for file not found and other exceptions.

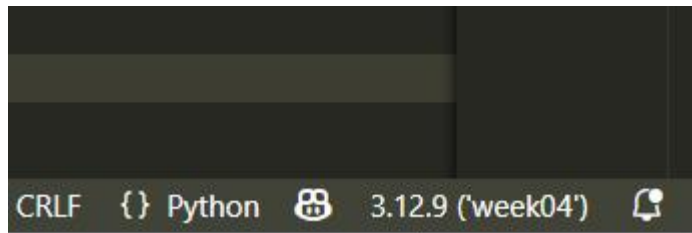
```
6         for line in file:
7             name, gender, email = line.strip().split(
8                 contacts.append((name, gender, email))
9
10            # 按邮箱域名和用户名排序
11            sorted_contacts = sorted(contacts, key=lambda x:
12
13            # 生成邮件内容
14            email_content = []
15            for name, gender, email in sorted_contacts:
16                title = '先生' if gender == '男' else '女士'
17                email_content.append(f'to: <{email}>\n尊敬的{n
18
19            # 去除最后一个多余的分隔符
20            if email_content:
21                email_content[-1] = email_content[-1].rstrip(
22
23            # 写入 emails.txt 文件
24            with open('emails.txt', 'w', encoding='utf-8') as
25                output_file.write('\n'.join(email_content))
26            except FileNotFoundError:
27                print("错误: contacts.txt 文件未找到!")
28            except Exception as e:
29                print(f"错误: 发生了一个未知错误: {e}")
30
31
32            if __name__ == "__main__":
33                process_contacts()
34
```

六、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 (越是初学者越应该慢慢调试、检查、试验，借此学习)

(一) 将大模型提供的代码复制粘贴进 main.py 文件，记得保存

(二) 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)

右下角切换成 week04



输入 `python main.py`

```
Administrator@PC-20220115WOJX MINGW64
$ python main.py
(week04)
```

```
Administrator@PC-20220115WOJX MINGW64 ~/repo/week04 (main)
$ ls -l
total 31
-rw-r--r-- 1 Administrator 197121 204 3月 30 16:41 contacts.txt
-rw-r--r-- 1 Administrator 197121 661 3月 30 20:27 emails.txt
-rw-r--r-- 1 Administrator 197121 72 3月 30 16:22 environment.yml
-rw-r--r-- 1 Administrator 197121 18805 3月 30 16:06 LICENSE
-rw-r--r-- 1 Administrator 197121 1316 3月 30 19:32 main.py
-rw-r--r-- 1 Administrator 197121 2239 3月 30 16:06 README.md
(week04)
Administrator@PC-20220115WOJX 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: <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)
```

(三) 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)

删掉 `emil.py`

输入 `python -m pdb main.py`

```
Administrator@PC-20220115W0JX MINGW64 ~/repo/week04 (main)
$ python -m pdb main.py
> c:\users\administrator\repo\week04\main.py(1)<module>()
-> def process_contacts():
(Pdb)
```

注：（1）表示第一行

（四）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)

l: 显示代码

```
Administrator@PC-20220115W0JX MINGW64 ~/repo/week04 (main)
$ python -m pdb main.py
> c:\users\administrator\repo\week04\main.py(1)<module>()
-> def process_contacts():
(Pdb) l
1  -> def process_contacts():
2      try:
3          # 读取 contacts.txt 文件
4          with open('contacts.txt', 'r', encoding='utf-8') as file:
5              contacts = []
6              for line in file:
7                  name, gender, email = line.strip().split()
8                  contacts.append((name, gender, email))
9
10             # 按邮箱域名和用户名排序
11             sorted_contacts = sorted(contacts, key=lambda x: (x[2].split
('@')[1], x[2].split('@')[0]))
(Pdb)
```

箭头表示即将运行但还没运行的代码

n: 执行当前行

```
(Pdb) n
> c:\users\administrator\repo\week04\main.py(32)<module>()
-> if __name__ == "__main__":
```

p: 显示表达式 (print)

s: 步入 (step in)

pp: 美观打印, 对齐

```
(Pdb) p contacts
[('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', 'lixliulian@163.com')]
(Pdb) pp contacts
[('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', 'lixliulian@163.com')]
(Pdb)
```

```
(Pdb) n
The program finished and will be restarted
> c:\users\administrator\repo\week04\main.py(1)<module>()
-> def process_contacts():
(Pdb) l
1  -> def process_contacts():
2      try:
3          # 读取 contacts.txt 文件
4          with open('contacts.txt', 'r', encoding='utf-8') as file:
5              contacts = []
6              for line in file:
7                  name, gender, email = line.strip().split()
8                  contacts.append((name, gender, email))
9
10             # 按邮箱域名和用户名排序
11             sorted_contacts = sorted(contacts, key=lambda x: (x[2].split('@')[1], x[2].split('@')[0]))
(Pdb) n
> c:\users\administrator\repo\week04\main.py(32)<module>()
-> if __name__ == "__main__":
(Pdb)
```

q: 退出调试环境

```
(Pdb) q
(week04)
Administrator@PC-20220115W0JX MINGW64 ~/repo/week04 (main)
```

变量还没定义时，p 会显示不出来

内置函数不需要定义

```
(Pdb) p __name__
'__main__'
(Pdb)
```

```
(Pdb) p len
<built-in function len>
(Pdb)
```

```

(Pdb) n
> c:\users\administrator\repo\week04\main.py(32)<module>()
-> if __name__ == "__main__":
(Pdb)
> c:\users\administrator\repo\week04\main.py(33)<module>()
-> process_contacts()
(Pdb)
--Return--
> c:\users\administrator\repo\week04\main.py(33)<module>()->None
-> process_contacts()
(Pdb)
--Return--
> <string>(1)<module>()->None
(Pdb)
The program finished and will be restarted
> c:\users\administrator\repo\week04\main.py(1)<module>()
-> def process_contacts():
(Pdb)
> c:\users\administrator\repo\week04\main.py(32)<module>()
-> if __name__ == "__main__":
(Pdb)
> c:\users\administrator\repo\week04\main.py(33)<module>()
-> process_contacts()
(Pdb) s
--Call--
> c:\users\administrator\repo\week04\main.py(1)process_contacts()
-> def process_contacts():
(Pdb)

```

n 循环 return 循环到返回

```

> c:\users\administrator\repo\week04\main.py(7)process_contacts()
-> name, gender, email = line.strip().split()
(Pdb)
> c:\users\administrator\repo\week04\main.py(8)process_contacts()
-> contacts.append((name, gender, email))
(Pdb)
> c:\users\administrator\repo\week04\main.py(6)process_contacts()
-> for line in file:
(Pdb)
> c:\users\administrator\repo\week04\main.py(7)process_contacts()
-> name, gender, email = line.strip().split()
(Pdb)
> c:\users\administrator\repo\week04\main.py(8)process_contacts()
-> contacts.append((name, gender, email))
(Pdb)
> c:\users\administrator\repo\week04\main.py(6)process_contacts()
-> for line in file:
(Pdb)
> c:\users\administrator\repo\week04\main.py(7)process_contacts()
-> name, gender, email = line.strip().split()
(Pdb)
> c:\users\administrator\repo\week04\main.py(8)process_contacts()
-> contacts.append((name, gender, email))
(Pdb)
> c:\users\administrator\repo\week04\main.py(6)process_contacts()
-> for line in file:
(Pdb) p line
'李秀莲 男 lixiulian@163.com\n'
(Pdb)

```

```

(Pdb)
> c:\users\administrator\repo\week04\main.py(16)process_contacts()
-> title = '先生' if gender == '男' else '女士'
(Pdb)
> c:\users\administrator\repo\week04\main.py(17)process_contacts()
-> email_content.append(f'to: <{email}>\n尊敬的{name}{title}, 您的会
将到期, 请及时续费.\n---')
(Pdb)
> c:\users\administrator\repo\week04\main.py(15)process_contacts()
-> for name, gender, email in sorted_contacts:
(Pdb)
> c:\users\administrator\repo\week04\main.py(20)process_contacts()
-> if email_content:
(Pdb)
> c:\users\administrator\repo\week04\main.py(21)process_contacts()
-> email_content[-1] = email_content[-1].rstrip('---')
(Pdb)
> c:\users\administrator\repo\week04\main.py(24)process_contacts()
-> with open('emails.txt', 'w', encoding='utf-8') as output_file:
(Pdb)
> c:\users\administrator\repo\week04\main.py(25)process_contacts()
-> output_file.write('\n'.join(email_content))
(Pdb)
> c:\users\administrator\repo\week04\main.py(24)process_contacts()
-> with open('emails.txt', 'w', encoding='utf-8') as output_file:
(Pdb)
--Return--

```

p line : 读取运行到的哪一行

```

(Pdb) p line
'李秀莲 男 lixiulian@163.com\n'
(Pdb) p name
'李秀莲'
(Pdb) p email
'lixiulian@163.com'
(Pdb)

```