

1. 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

```
admin@LAPTOP-05T5IN81 MINGW64 /D/aaa/A
$ git clone git@gitcode.com:Sharrair/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)
Receiving objects: 100% (5/5), 8.45 KiB | 1.41 MiB/s, done.

admin@LAPTOP-05T5IN81 MINGW64 /D/aaa/A
$ code week04
```

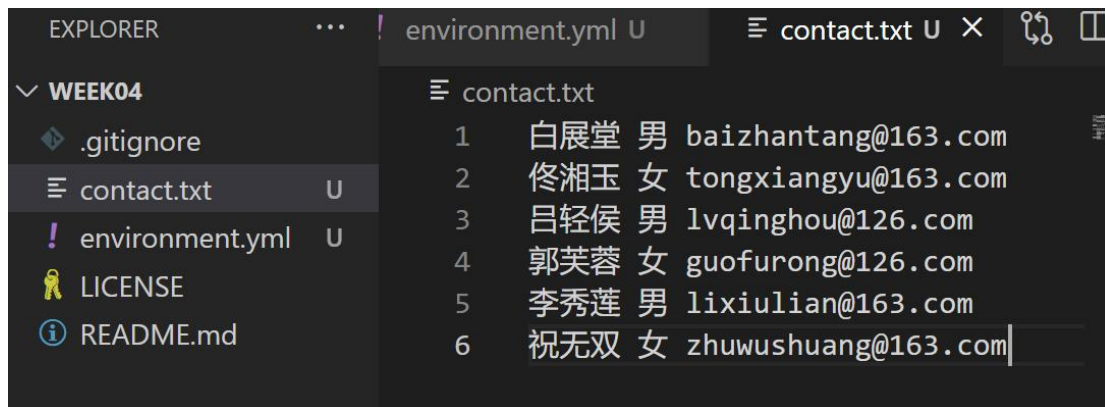
2. 用 VS Code 打开项目目录，新建一个 environment.yml 文件，指定安装 Python 3.12，然后运行 conda env create 命令创建 Conda 环境

```
name: weeko4
channels:
  - conda-forge
dependencies:
  - python=3.12
```

```
(base) sherry@LAPTOP-05T5IN81:/mnt/d/aaa/A/week04$ conda env create
Collecting package metadata (repodata.json): done
Solving environment: done
```

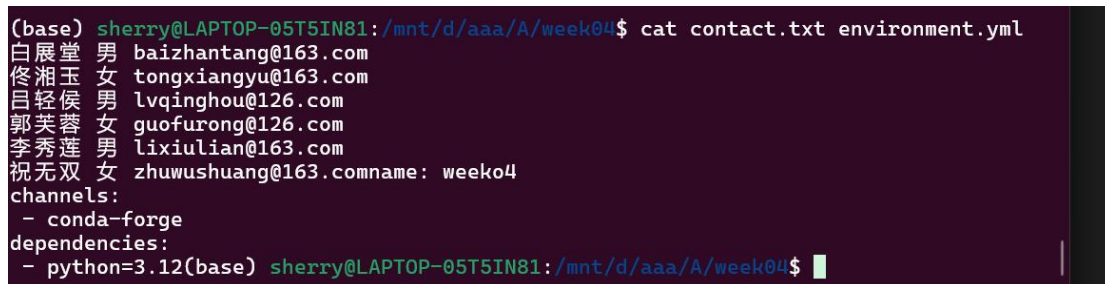
```
(base) sherry@LAPTOP-05T5IN81:/mnt/d/aaa/A/week04$ conda env list
# conda environments:
#
base                *  /home/sherry/miniconda3
aaa                  /home/sherry/miniconda3/envs/aaa
bbb                  /home/sherry/miniconda3/envs/bbb
e0101-terminal       /home/sherry/miniconda3/envs/e0101-terminal
myenv                /home/sherry/miniconda3/envs/myenv
myproject            /home/sherry/miniconda3/envs/myproject
prj1                 /home/sherry/miniconda3/envs/prj1
pystudy              /home/sherry/miniconda3/envs/pystudy
weeko4               /home/sherry/miniconda3/envs/weeko4
```

3. 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



The image shows the VS Code Explorer on the left with the file tree for 'WEEK04' containing .gitignore, contact.txt, environment.yml, LICENSE, and README.md. The main editor on the right displays the content of 'contact.txt' with six lines of contact information.

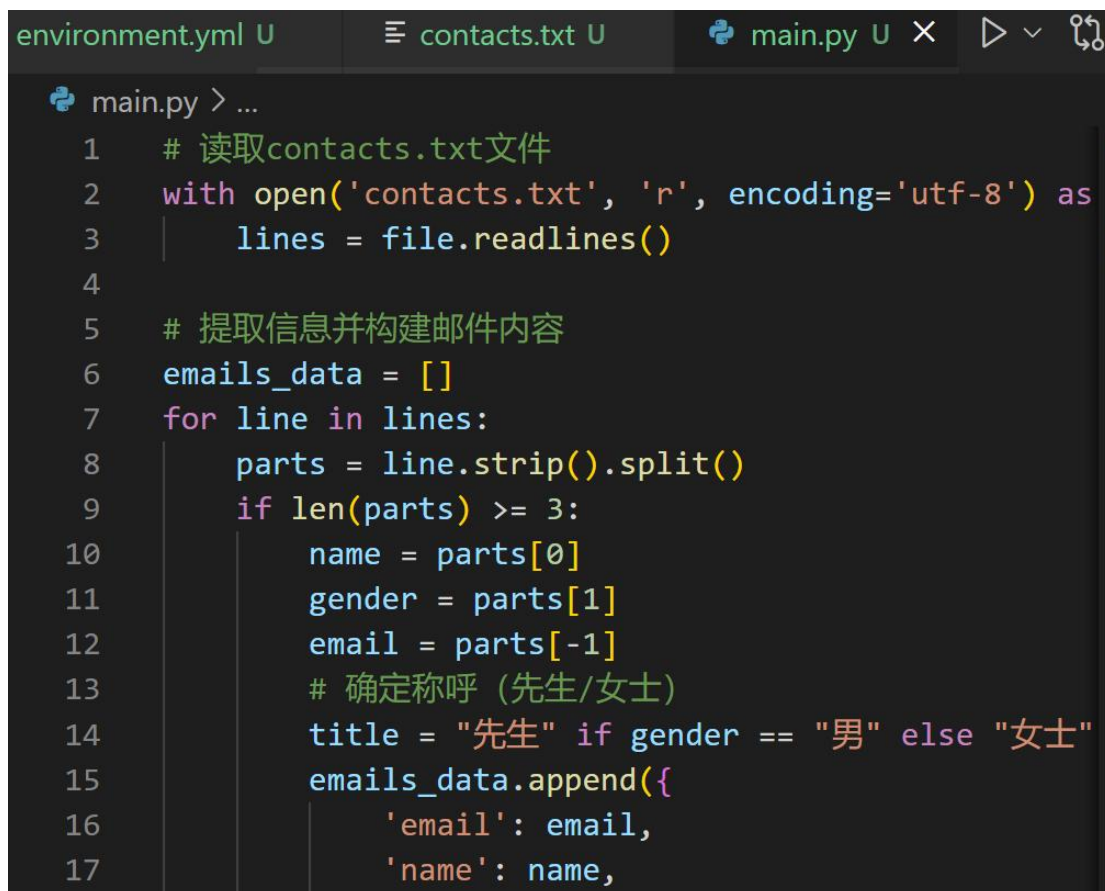
```
1 白展堂 男 baizhantang@163.com
2 佟湘玉 女 tongxiangyu@163.com
3 吕轻侯 男 lvqinghou@126.com
4 郭芙蓉 女 guofurong@126.com
5 李秀莲 男 lixiulian@163.com
6 祝无双 女 zhuwushuang@163.com
```



The terminal window shows the output of the command 'cat contact.txt environment.yml'. It displays the contact information from the previous block followed by the content of environment.yml.

```
(base) sherry@LAPTOP-05T5IN81: /mnt/d/aaa/A/week04$ cat contact.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
(base) sherry@LAPTOP-05T5IN81: /mnt/d/aaa/A/week04$
```

4. 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)



The image shows the VS Code editor with the 'main.py' file open. The code reads 'contacts.txt', processes the data to extract names, genders, and emails, and then appends this information to a list. The code is as follows:

```
main.py > ...
1 # 读取contacts.txt文件
2 with open('contacts.txt', 'r', encoding='utf-8') as
3     lines = file.readlines()
4
5 # 提取信息并构建邮件内容
6 emails_data = []
7 for line in lines:
8     parts = line.strip().split()
9     if len(parts) >= 3:
10         name = parts[0]
11         gender = parts[1]
12         email = parts[-1]
13         # 确定称呼 (先生/女士)
14         title = "先生" if gender == "男" else "女士"
15         emails_data.append({
16             'email': email,
17             'name': name,
```

```
(base) sherry@LAPTOP-05T5IN81:/mnt/d/aaa/A/week04$ code week04
(base) sherry@LAPTOP-05T5IN81:/mnt/d/aaa/A/week04$ python main.py
错误: 文件 contacts.txt 未找到。
(base) sherry@LAPTOP-05T5IN81:/mnt/d/aaa/A/week04$ python main.py
邮件已成功写入 emails.txt。
(base) sherry@LAPTOP-05T5IN81:/mnt/d/aaa/A/week04$ 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。
---(base) sherry@LAPTOP-05T5IN81:/mnt/d/aaa/A/week04$ █
```

```
-> lines = file.readlines()
(Pdb) p lines
*** NameError: name 'lines' is not defined
(Pdb) n
> /mnt/d/aaa/A/week04/main.py(6)<module>()
-> emails_data = []
(Pdb) p lines
['白展堂 男 baizhantang@163.com\n', '佟湘玉 女 tongxiangyu@163.com\n',
'吕轻侯 男 lvqinghou@126.com\n', '郭芙蓉 女 guofurong@126.com\n', '李秀
莲 男 lixiluan@163.com\n', '祝无双 女 zhuwushuang@163.com']
(Pdb) pp lines
['白展堂 男 baizhantang@163.com\n',
'佟湘玉 女 tongxiangyu@163.com\n',
'吕轻侯 男 lvqinghou@126.com\n',
'郭芙蓉 女 guofurong@126.com\n',
'李秀莲 男 lixiluan@163.com\n',
'祝无双 女 zhuwushuang@163.com']
(Pdb) ll
1      # 读取contacts.txt文件
2      with open('contacts.txt', 'r', encoding='utf-8') as file:
3          lines = file.readlines()
4
5      # 提取信息并构建邮件内容
6      -> emails_data = []
7      for line in lines:
8          parts = line.strip().split()
9          if len(parts) >= 3:
10             name = parts[0]
11             gender = parts[1]
12             email = parts[-1]
```



```

(Pdb) s
> /mnt/d/aaa/A/week04/main.py(7)<module>()
-> for line in lines:
(Pdb) s
> /mnt/d/aaa/A/week04/main.py(8)<module>()
-> parts = line.strip().split()
(Pdb) n
> /mnt/d/aaa/A/week04/main.py(9)<module>()
-> if len(parts) >= 3:
(Pdb) c
处理完成, 结果已保存到emails.txt
The program finished and will be restarted
> /mnt/d/aaa/A/week04/main.py(2)<module>()
-> with open('contacts.txt', 'r', encoding='utf-8') as file:
(Pdb) quit
(base) sherry@LAPTOP-05T5IN81:/mnt/d/aaa/A/week04$

```

利用 wat-inspector (第三方软件包, 需要安装) 检查 (inspect)各种对象

```

(base) sherry@LAPTOP-05T5IN81:/mnt/d/aaa/A/week04$ pip install wat-insp
ector
Collecting wat-inspector
  Downloading wat_inspector-0.4.3-py3-none-any.whl (15 kB)
Installing collected packages: wat-inspector
Successfully installed wat-inspector-0.4.3

```

5. 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 (建议观看下面的录播讲解)

- 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)、全局变量 (global variable)、LEGB 规则
- 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)

我新建了文件 **main_study.py**, 内容为在 **main.py** 中各行命令后面直接加了注释, 注释内容即该句命令中涉及了什么以上的概念, 只在在该概念第一次出现的命令行注释, 未出现的概念单独注释于代码底部并写了例子