

第四周

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

```
(base) 慧慧子@LAPTOP-A0MIIJ99 MINGW64 ~  
$ cd repo  
  
(base) 慧慧子@LAPTOP-A0MIIJ99 MINGW64 ~/repo  
$ ls -l  
total 4  
drwxr-xr-x 1 慧慧子 197121 0 3月 20 22:03 myproject/  
drwxr-xr-x 1 慧慧子 197121 0 3月 20 20:44 wang/  
  
(base) 慧慧子@LAPTOP-A0MIIJ99 MINGW64 ~/repo  
$ git clone https://gitcode.com/wangxuminghao/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.
```

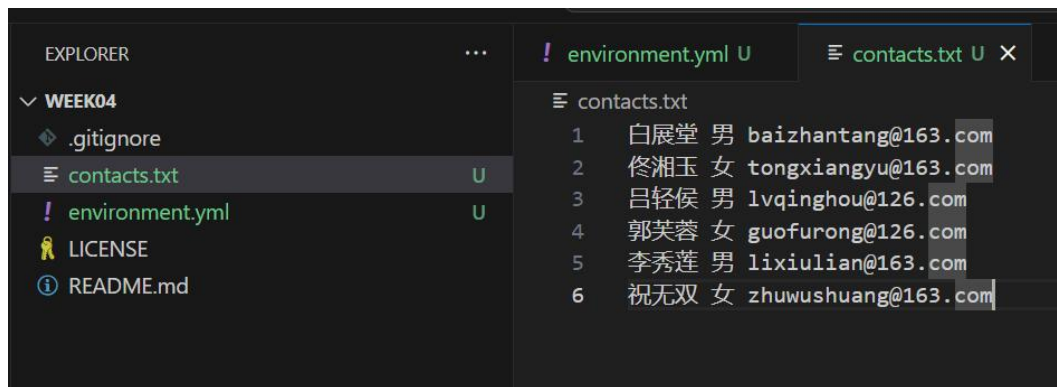
```
$ cd week04/  
  
(base) 慧慧子@LAPTOP-A0MIIJ99 MINGW64 ~/repo/week04 (main)  
$ pwd  
/c/Users/慧慧子/repo/week04  
  
(base) 慧慧子@LAPTOP-A0MIIJ99 MINGW64 ~/repo/week04 (main)  
$ git remote show origin  
* remote origin  
Fetch URL: https://gitcode.com/wangxuminghao/week04.git  
Push URL: https://gitcode.com/wangxuminghao/week04.git  
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)
```

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

```
(base) 慧慧子@LAPTOP-A0MIIJ99 MINGW64 ~/repo/week04 (main)  
$ ls -l ../myproject  
total 196849  
-rw-r--r-- 1 慧慧子 197121 83 3月 20 22:25 environment.yml  
-rw-r--r-- 1 慧慧子 197121 201568176 3月 20 22:13 EPA_SmartLocationDatabase_V3_Jan_2021_Final.csv  
-rw-r--r-- 1 慧慧子 197121 713 3月 20 22:24 main.py  
  
(base) 慧慧子@LAPTOP-A0MIIJ99 MINGW64 ~/repo/week04 (main)  
$ cat ../myproject/environment.yml  
name: myproject  
channels:  
- conda-forge  
dependencies:  
- python  
- pandas  
  
(base) 慧慧子@LAPTOP-A0MIIJ99 MINGW64 ~/repo/week04 (main)  
$ cp ../myproject/environment.yml ./  
  
(base) 慧慧子@LAPTOP-A0MIIJ99 MINGW64 ~/repo/week04 (main)  
$ ls -l  
total 25  
-rw-r--r-- 1 慧慧子 197121 83 3月 27 19:58 environment.yml  
-rw-r--r-- 1 慧慧子 197121 18805 3月 27 19:54 LICENSE  
-rw-r--r-- 1 慧慧子 197121 2239 3月 27 19:54 README.md
```

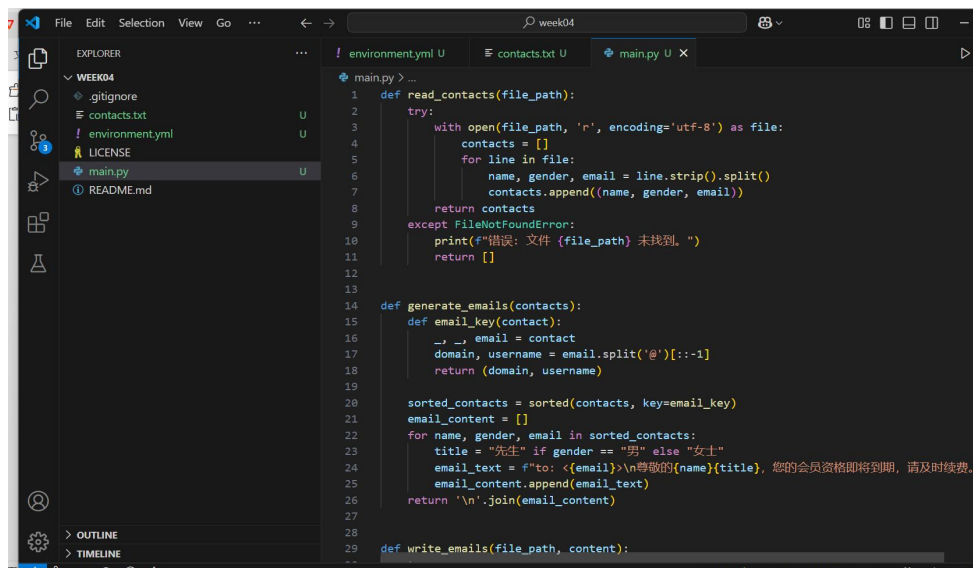
```
(base) 慧慧子@LAPTOP-A0MIIJ99 MINGW64 ~/repo/week04 (main)
$ cat environment.yml
name: myproject
channels:
  - conda-forge
dependencies:
  - python
  - pandas
(base) 慧慧子@LAPTOP-A0MIIJ99 MINGW64 ~/repo/week04 (main)
done
#
# To activate this environment, use
#
#     $ conda activate week04
#
# To deactivate an active environment, use
#
#     $ conda deactivate
```

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



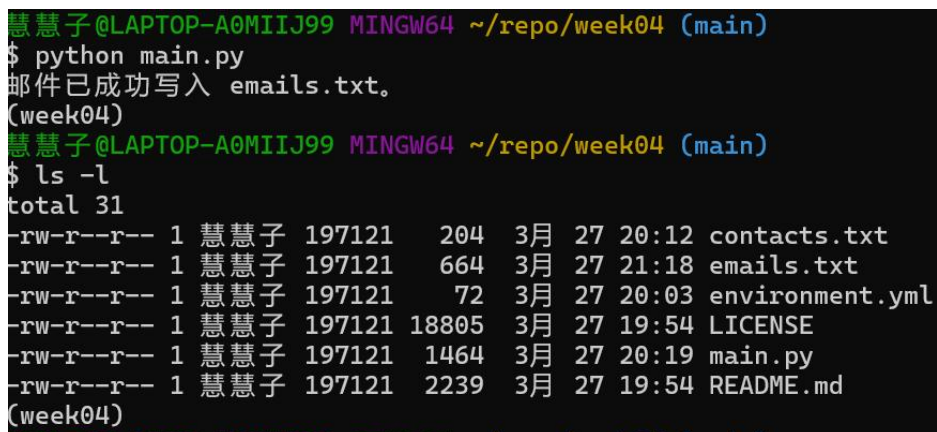
```
(base) 慧慧子@LAPTOP-A0MIIJ99 MINGW64 ~/repo/week04 (main)
$ ls -l
total 26
-rw-r--r-- 1 慧慧子 197121 202 3月 27 20:07 contacts.txt
-rw-r--r-- 1 慧慧子 197121 72 3月 27 20:03 environment.yml
-rw-r--r-- 1 慧慧子 197121 18805 3月 27 19:54 LICENSE
-rw-r--r-- 1 慧慧子 197121 2239 3月 27 19:54 README.md
(base) 慧慧子@LAPTOP-A0MIIJ99 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) 慧慧子@LAPTOP-A0MIIJ99 MINGW64 ~/repo/week04 (main)
```

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



```
1 def read_contacts(file_path):
2     try:
3         with open(file_path, '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     except FileNotFoundError:
9         print(f"错误: 文件 {file_path} 未找到。")
10    return []
11
12
13
14 def generate_emails(contacts):
15     def email_key(contact):
16         _, _, email = contact
17         domain, username = email.split('@')[::-1]
18         return (domain, username)
19
20     sorted_contacts = sorted(contacts, key=email_key)
21     email_content = []
22     for name, gender, email in sorted_contacts:
23         title = "先生" if gender == "男" else "女士"
24         email_text = f"to: <{email}>\n尊敬的{name}{title}, 您的会员资格即将到期, 请及时续费。"
25         email_content.append(email_text)
26     return '\n'.join(email_content)
27
28
29 def write_emails(file_path, content):
```

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（越是初学者越应该慢慢调试、检查、试验，借此学习）



```
慧慧子@LAPTOP-A0MIIJ99 MINGW64 ~/repo/week04 (main)
$ python main.py
邮件已成功写入 emails.txt。
(week04)
慧慧子@LAPTOP-A0MIIJ99 MINGW64 ~/repo/week04 (main)
$ ls -l
total 31
-rw-r--r-- 1 慧慧子 197121 204 3月 27 20:12 contacts.txt
-rw-r--r-- 1 慧慧子 197121 664 3月 27 21:18 emails.txt
-rw-r--r-- 1 慧慧子 197121 72 3月 27 20:03 environment.yml
-rw-r--r-- 1 慧慧子 197121 18805 3月 27 19:54 LICENSE
-rw-r--r-- 1 慧慧子 197121 1464 3月 27 20:19 main.py
-rw-r--r-- 1 慧慧子 197121 2239 3月 27 19:54 README.md
(week04)
```



```
慧慧子@LAPTOP-A0MIIJ99 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: <lixliulian@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)
```

```

慧慧子@LAPTOP-A0MIIJ99 MINGW64 ~/repo/week04 (main)
$ ls -l
total 31
-rw-r--r-- 1 慧慧子 197121 204 3月 27 20:12 contacts.txt
-rw-r--r-- 1 慧慧子 197121 664 3月 27 21:18 emails.txt
-rw-r--r-- 1 慧慧子 197121 72 3月 27 20:03 environment.yml
-rw-r--r-- 1 慧慧子 197121 18805 3月 27 19:54 LICENSE
-rw-r--r-- 1 慧慧子 197121 1464 3月 27 20:19 main.py
-rw-r--r-- 1 慧慧子 197121 2239 3月 27 19:54 README.md
(week04)
慧慧子@LAPTOP-A0MIIJ99 MINGW64 ~/repo/week04 (main)
$ rm emails.txt
(week04)
慧慧子@LAPTOP-A0MIIJ99 MINGW64 ~/repo/week04 (main)
$ ls -l
total 30
-rw-r--r-- 1 慧慧子 197121 204 3月 27 20:12 contacts.txt
-rw-r--r-- 1 慧慧子 197121 72 3月 27 20:03 environment.yml
-rw-r--r-- 1 慧慧子 197121 18805 3月 27 19:54 LICENSE
-rw-r--r-- 1 慧慧子 197121 1464 3月 27 20:19 main.py
-rw-r--r-- 1 慧慧子 197121 2239 3月 27 19:54 README.md
(week04)

```

```

慧慧子@LAPTOP-A0MIIJ99 MINGW64 ~/repo/week04 (main)
$ ls -l
total 30
-rw-r--r-- 1 慧慧子 197121 204 3月 27 20:12 contacts.txt
-rw-r--r-- 1 慧慧子 197121 72 3月 27 20:03 environment.yml
-rw-r--r-- 1 慧慧子 197121 18805 3月 27 19:54 LICENSE
-rw-r--r-- 1 慧慧子 197121 1464 3月 27 20:19 main.py
-rw-r--r-- 1 慧慧子 197121 2239 3月 27 19:54 README.md
(week04)
慧慧子@LAPTOP-A0MIIJ99 MINGW64 ~/repo/week04 (main)
$ python -m pdb main.py
> c:\users\慧慧子\repo\week04\main.py(1)<module>()
-> def read_contacts(file_path):
(Pdb) l
1  -> def read_contacts(file_path):
2      try:
3          with open(file_path, '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          return contacts
9      except FileNotFoundError:
10         print(f"错误: 文件 {file_path} 未找到。")

```

```

12         write_emails('emails.txt', email_content)
42
(Pdb) p read_contacts
<function read_contacts at 0x0000020ECD539800>
(Pdb) s
> c:\users\慧慧子\repo\week04\main.py(29)<module>()
-> def write_emails(file_path, content):
(Pdb)

```

```

9         for line in file:
(Pdb) l
6             name, gender, email = line.strip().split()
7             contacts.append((name, gender, email))
8         return contacts
9     except FileNotFoundError:
10         print(f"错误: 文件 {file_path} 未找到。")
11         return []
12
13     def generate_emails(contacts):
14         def email_key(contact):
15             _, _, email = contact
16
(Pdb) l
33         print(f"邮件已成功写入 {file_path}。")
34     except Exception as e:
35         print(f"错误: 写入文件 {file_path} 时出现问题: {e}")
36
37
38 -> if __name__ == "__main__":
39     contacts = read_contacts('contacts.txt')
40     email_content = generate_emails(contacts)
41     write_emails('emails.txt', email_content)
42
[EOF]
(Pdb)

```



```

42
[EOF]
(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', 'lixiliu@163.com'), ('祝无双', '女', 'zhuwushuang@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', 'lixiliu@163.com'), ('祝无双', '女', 'zhuwushuang@163.com')]
(Pdb)

```

Python 语法保留字 (reserved key words)

语句 (statement) 和表达式 (expression)

缩进 (indent)——代表层级，界定子语句的边界

局部变量 (local variable) vs. 全局变量 (global variable)

```

慧慧子@LAPTOP-A6M11J99 MINGW64 ~/repo/week04 (main)
$ python -m pdb main.py
> c:\users\慧慧子\repo\week04\main.py(1)<module>()
-> def read_contacts(file_path):
(Pdb) l
1  -> def read_contacts(file_path):
2      try:
3          with open(file_path, '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              return contacts
9      except FileNotFoundError:
10         print(f"错误: 文件 {file_path} 未找到。")
11         return []
(Pdb) wat
*** NameError: name 'wat' is not defined
(Pdb) import wat
(Pdb) wat
Try wat / object or wat.modifiers / object to inspect an object. Modifiers are:
.short or .s to hide attributes (variables and methods)
.dunder to print dunder attributes
.code to print source code of a function, method or class
.long to print non-abbreviated values and documentation
.nodocs to hide documentation for functions and classes

```

```

Try wat / object or wat.modifiers / object to inspect an object. Modifiers are:
.short or .s to hide attributes (variables and methods)
.dunder to print dunder attributes
.code to print source code of a function, method or class
.long to print non-abbreviated values and documentation
.nodocs to hide documentation for functions and classes
.caller to show how and where the inspection was called
.all to include all information
.ret to return the inspected object
.str to return the output string instead of printing
.gray to disable colorful output in the console
.color to enforce colorful outputs in the console
Call wat.locals or wat() to inspect local variables.
Call wat.globals to inspect global variables.

(Pdb) wat()
Local variables:
__builtins__: dict = {...
__file__: pdb._ScriptTarget = 'C:\Users\慧慧子\repo\week04\main.py'
__name__: str = '__main__'
__pdb_convenience_variables__: dict = {...
__spec__: NoneType = None
wat: wat.inspection.inspection.Wat = <WAT Inspector object>

```

```
(Pdb) wat.globals
Global variables:
  __builtins__: dict = {...
  __file__: pdb._ScriptTarget = 'C:\Users\慧慧子\repo\week04\main.py'
  __name__: str = '__main__'
  __pdb_convenience_variables__: dict = {...
  __spec__: NoneType = None
  wat: wat.inspection.inspection.Wat = <WAT Inspector object>
(Pdb)
```

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)——两个等号是运算符，.是名称访问运算符，()是运输符，定义后的不属于

形参 (parameter)、实参 (argument)、返回值 (return value)

对象 (object)、类型 (type)、属性 (attribute)、方法 (method)