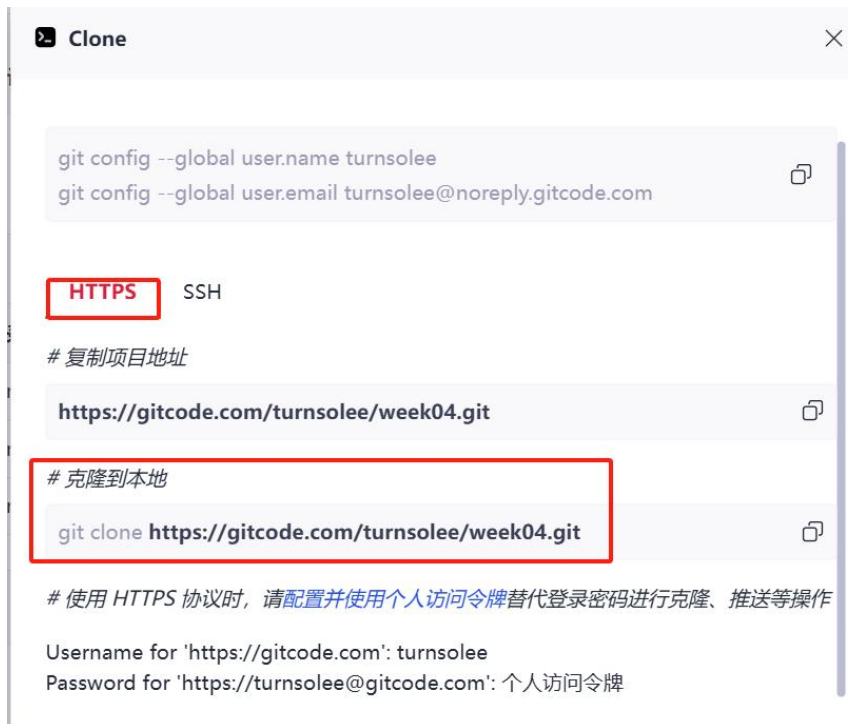


第四周学习报告

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



```
(base)
李意如@LAPTOP-9J8HOMDD MINGW64 /c/Users/李意如/repo
$ git clone https://gitcode.com/turnsolee/week04.git
```

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)
李意如@LAPTOP-9J8HOMDD MINGW64 /c/Users/李意如/repo/week04 (main)
$ cat ../myproject/environment.yml
name: myproject
channels:
  - conda-forge
dependencies:
  - python=3.12
  - pandas(base)
```

cat 命令可以拼接不同的文件

小知识: `ctrl+a` 光标跳到最前，`ctrl+e` 光标跳到最后

```
李意如@LAPTOP-9J8HOMDD MINGW64 /c/Users/李意如/repo/week04 (main)
$ cp ../myproject/environment.yml ./
```

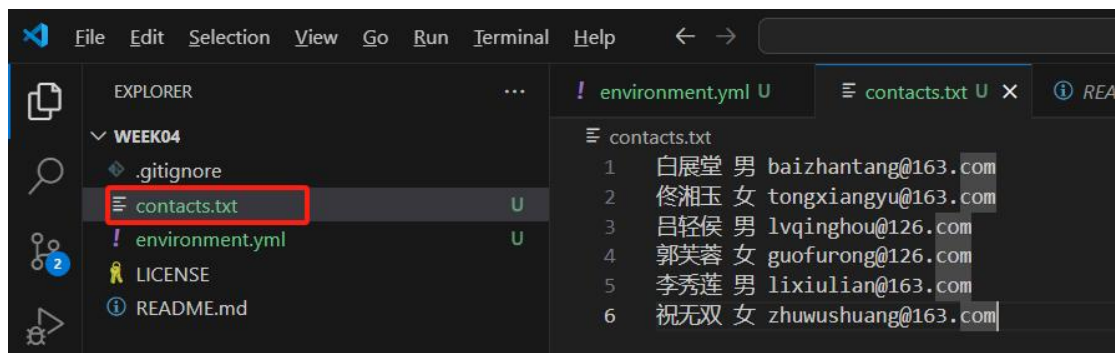
代码的含义是把上一级文件夹中的 `environment.yml` 复制到当前文件夹中

```
李意如@LAPTOP-9J8HOMDD MINGW64 /c/Users/李意如/repo/week04 (main)
$ conda env create
```

创建 conda 环境

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

```
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
```

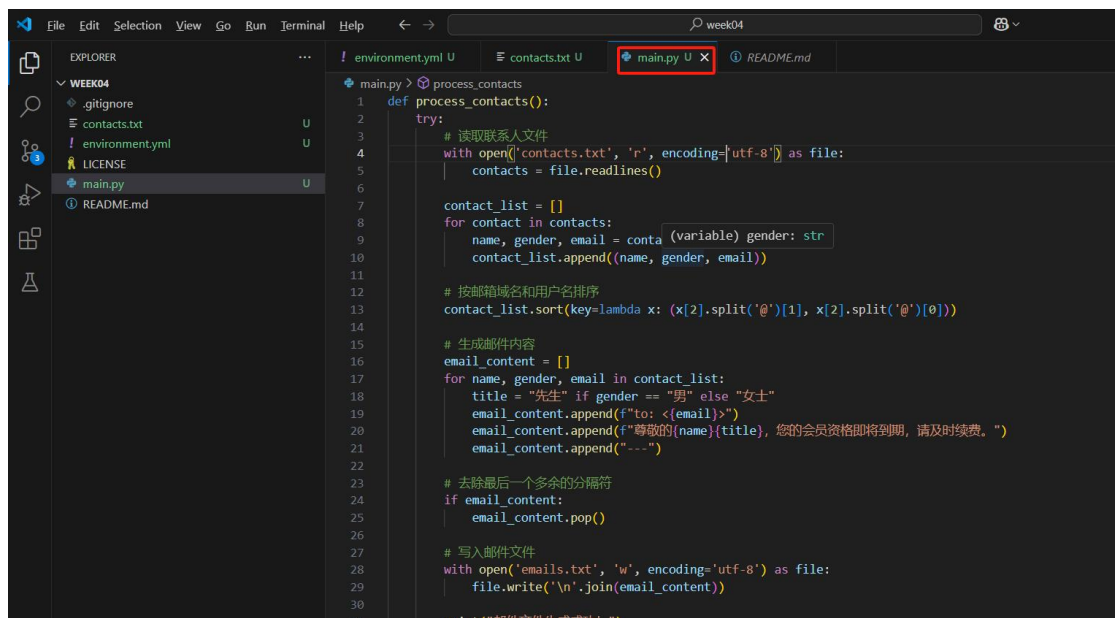
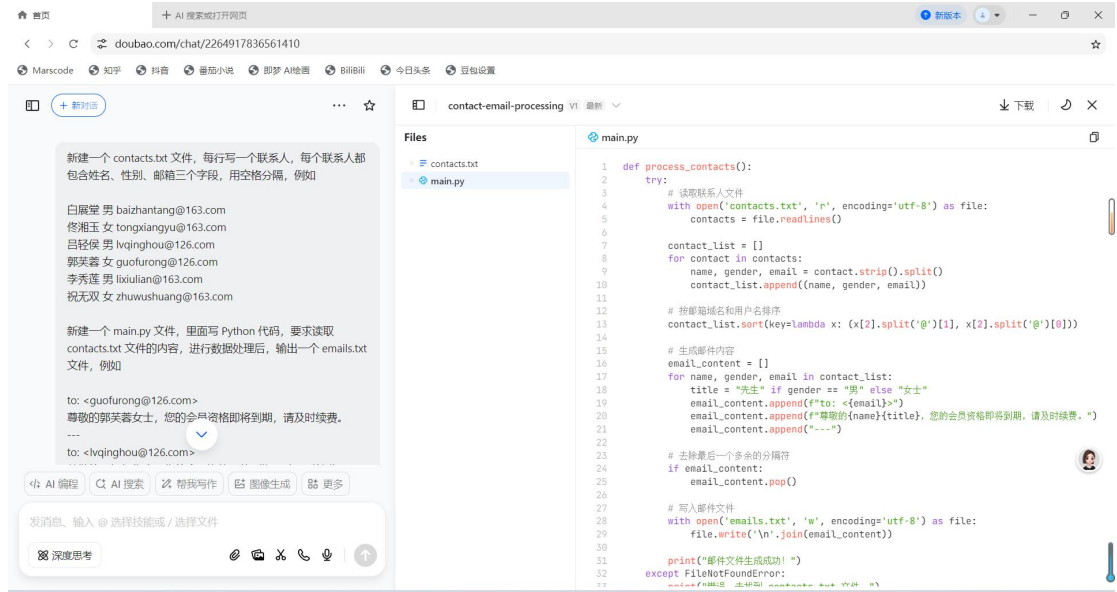


4. 新建一个 `main.py` 文件，里面写 Python 代码，要求读取 `contacts.txt` 文件的内容，进行数据处理后，输出一个 `emails.txt` 文件，例如

```
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: <lixiulian@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。
```

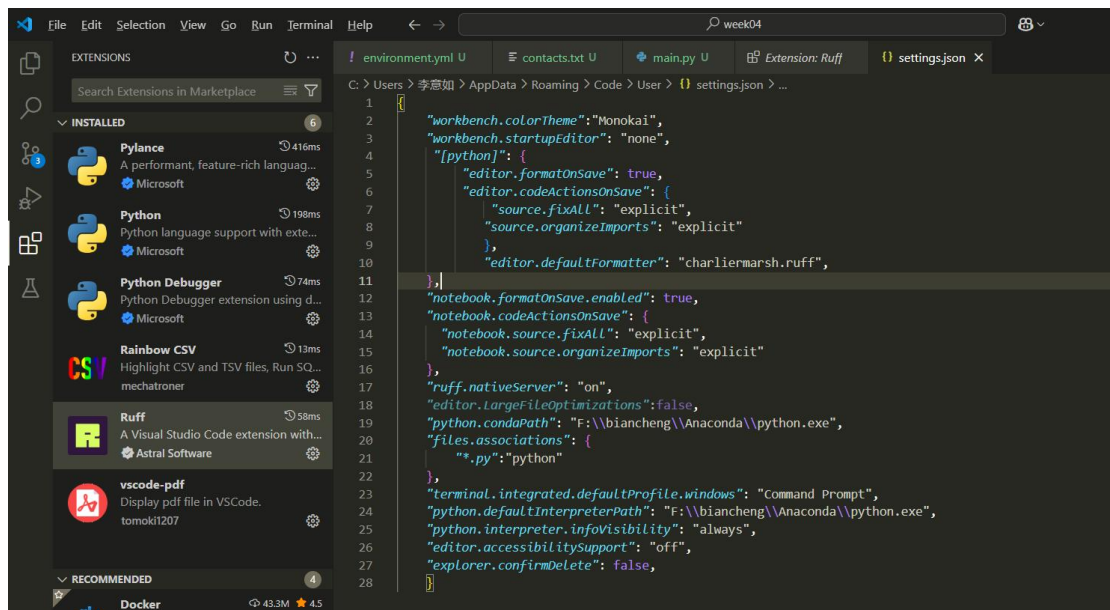
要求输出是先按邮箱域名排序 (126.com 排在 163.com 之前)，然后再按邮箱用户名排序 (guofurong 排在 lvqinghou 之前)

5. 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

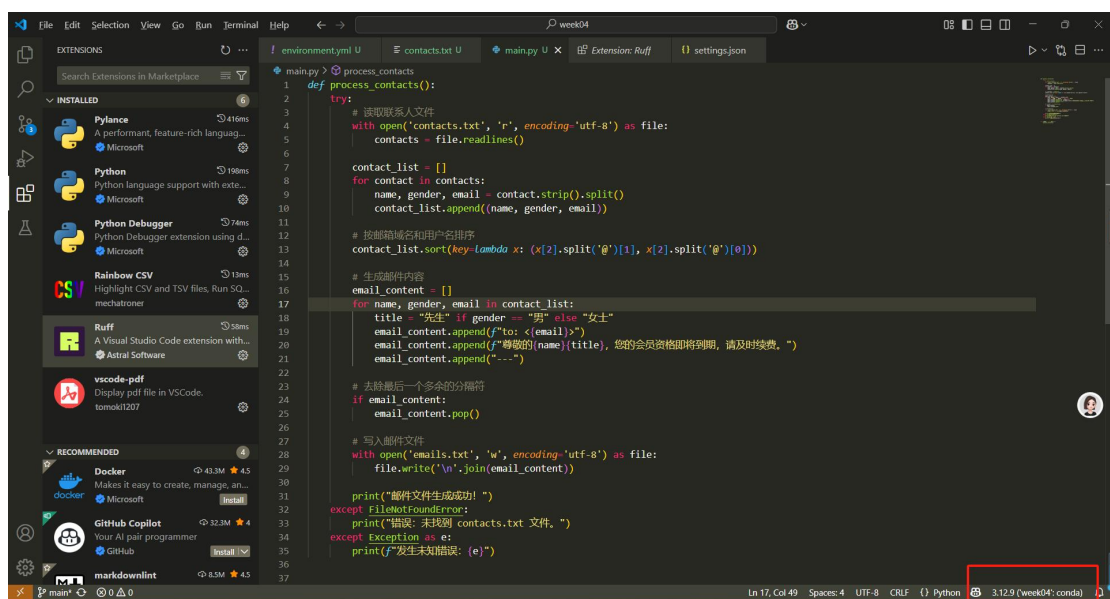
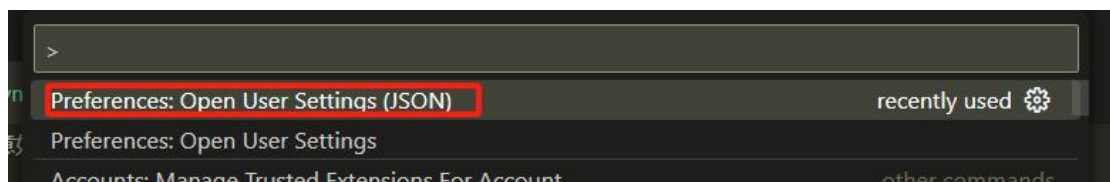


6. AI 回复的只是静态代码，而且可能含有错误，所以我们必须在 Conda 环境里运行代码，逐行调试，检查每一行代码的运行都符合我们的期望 (越是初学者越应该慢慢调试、检查、试验，借此学习)

- 将大模型提供的代码复制粘贴进 main.py 文件，记得保存
- 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)
- 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



Ruff 的设置，ctrl+shift+p, ruff 可以规范格式



选择 python 版本

- 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


```

(week04)
李意如@LAPTOP-9J8HOMDD MINGW64 /c/Users/李意如/repo/week04 (main)
$ python main.py
邮件文件生成成功!
(week04)
李意如@LAPTOP-9J8HOMDD MINGW64 /c/Users/李意如/repo/week04 (main)
$ ls -l
total 31
-rw-r--r-- 1 李意如 197121 204 3月 27 20:43 contacts.txt
-rw-r--r-- 1 李意如 197121 659 3月 27 21:40 emails.txt
-rw-r--r-- 1 李意如 197121 72 3月 27 20:34 environment.yml
-rw-r--r-- 1 李意如 197121 18805 3月 27 20:20 LICENSE
-rw-r--r-- 1 李意如 197121 1421 3月 27 21:38 main.py
-rw-r--r-- 1 李意如 197121 2239 3月 27 20:20 README.md

```

符合预期

```

(week04)
李意如@LAPTOP-9J8HOMDD MINGW64 /c/Users/李意如/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)

```

◦ 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)

```

(week04)
李意如@LAPTOP-9J8HOMDD MINGW64 /c/Users/李意如/repo/week04 (main)
$ python -m pdb main.py
> c:\users\李意如\repo\week04\main.py(1)<module>()
-> def process_contacts():
(Pdb) l
1  -> def process_contacts():
2      try:
3          # 读取联系人文件
4          with open("contacts.txt", "r", encoding="utf-8") as file:
5              contacts = file.readlines()
6
7          contact_list = []
8          for contact in contacts:
9              name, gender, email = contact.strip().split()
10             contact_list.append((name, gender, email))
11
(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)

l xx,xx 显示代码 l.显示当前行上下五行的代码 h 帮助

```
(week04)
李意如@LAPTOP-9J8HOMDD MINGW64 /c/Users/李意如/repo/week04 (main)
$ python -m pdb main.py
> c:\users\李意如\repo\week04\main.py(1)<module>()
-> def process_contacts():
(Pdb) l
1  -> def process_contacts():
2      try:
3          # 读取联系人文件
4          with open("contacts.txt", "r", encoding="utf-8") as file:
5              contacts = file.readlines()
6
7          contact_list = []
8          for contact in contacts:
9              name, gender, email = contact.strip().split()
10             contact_list.append((name, gender, email))
11
(Pdb)
```

显示的即将运行但没有运行的代码

“n” next 显示所有代码

```
(Pdb) n
> c:\users\李意如\repo\week04\main.py(40)<module>()
-> if __name__ == "__main__":
(Pdb) l
35     print("错误: 未找到 contacts.txt 文件。")
36     except Exception as e:
37         print(f"发生未知错误: {e}")
38
39
40 -> if __name__ == "__main__":
41     process_contacts()
[EOF]
(Pdb) ll
1  def process_contacts():
2      try:
3          # 读取联系人文件
4          with open("contacts.txt", "r", encoding="utf-8") as file:
5              contacts = file.readlines()
6
7          contact_list = []
8          for contact in contacts:
9              name, gender, email = contact.strip().split()
10             contact_list.append((name, gender, email))
11
12             # 按邮箱域名和用户名排序
```

“p” 打印运行过的表达式，没运行过的找不到 “pp” 美观打印

```
(Pdb) p process_contacts
<function process_contacts at 0x000002A02CD33600>
```

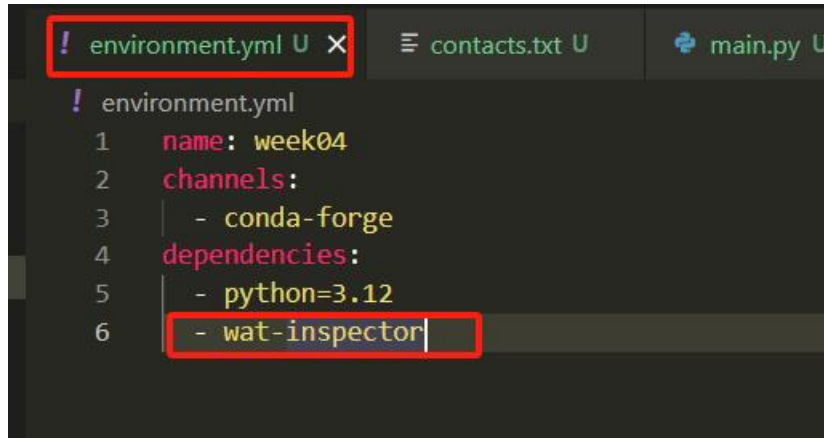
“s” 步入调用 “q” 退掉

学会预判

c(继续执行)

```
(Pdb) c
邮件文件生成成功!
The program finished and will be restarted
> c:\users\李意如\repo\week04\main.py(1)<module>()
-> def process_contacts():
```

- 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)



The screenshot shows a code editor with three tabs: `environment.yml`, `contacts.txt`, and `main.py`. The `environment.yml` tab is active and contains the following content:

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

```
李意如@LAPTOP-9J8HOMDD MINGW64 /c/Users/李意如/repo/week04 (main)
$ cat environment.yml
name: week04
channels:
  - conda-forge
dependencies:
  - python=3.12
  - wat-inspector(week04)
李意如@LAPTOP-9J8HOMDD MINGW64 /c/Users/李意如/repo/week04 (main)
$ conda env update
D:\anaconda3\Lib\argparse.py:2006: FutureWarning: `remote_definition
nda env create --file=URL` instead.
  action(self, namespace, argument_values, option_string)
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
#
# To activate this environment, use
```

。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)

1. 保留字不是变量，不可以赋值，有特殊含义



```
main.py > process_contacts
1 def process_contacts():
2     try:
3         # 读取联系人文件
4         with open("contacts.txt", "r", encoding="utf-8") as file:
5             contacts = file.readlines()
6
7         contact_list = []
8         for contact in contacts:
9             name, gender, email = contact.strip().split()
10            contact_list.append((name, gender, email))
11
12        # 按邮箱域名和用户名排序
13        contact_list.sort(key=lambda x: (x[2].split("@")[1], x[2].split("@")[0]))
14
15        # 生成邮件内容
16        email_content = []
17        for name, gender, email in contact_list:
18            title = "先生" if gender == "男" else "女士"
19            email_content.append(f"to: <{email}>")
20            email_content.append(
21                f"尊敬的{name}{title}, 您的会员资格即将到期, 请及时续费。"
22            )
23            email_content.append("---")
24
```

2. 语句 (statement) 和表达式 (expression)

表达式是构成语句的元素

3. 缩进 (indent): 严格对齐，代表层级

4. 局部变量 (local variable)、全局变量 (global variable)、LEGB 规则

局部变量 (local variable): 在函数、代码块（如循环体、条件语句块等）内部定义的变量，仅限于定义它的函数或代码块内部。出了这个范围就无法访问；

全局变量 (global variable): 在函数外部定义的变量，通常在程序开头或全局作用域中定义，整个程序，即程序中任何地方（只要在定义之后）都能访问；

LEGB 规则：

5. 函数 (function) 的定义 (define) 和调用 (call)

6. 字面值 (literal) (字符串 (str)、整数 (int)、列表 (list)、字典 (dict)、元组 (tuple))

字典 {}

7. 运算符 (operator): == (一个等号是赋值语句); if else; . (名称访问运算符)

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

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