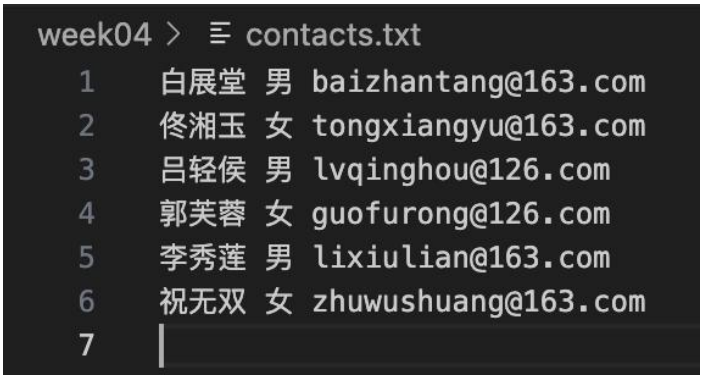
1.新建了一个 environment.yml 文件，指定安装 Python 3.12，然后运行 conda env create 命令创建了 Conda 环境

```
(base) apple@appledeMacBook-Pro week04 % conda env create
Retrieving notices: done
Collecting package metadata (repodata.json): done
Solving environment: done


==> WARNING: A newer version of conda exists. <==
  current version: 24.11.3
  latest version: 25.3.0

Please update conda by running

    $ conda update -n base -c https://repo.anaconda.com/pkgs/main conda

Or to minimize the number of packages updated during conda update use

     conda install conda=25.3.0


Downloading and Extracting Packages:

Preparing transaction: done
Verifying transaction: done
```

2.新建了一个 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



3. 新建了一个 main.py 文件，里面写了 Python 代码
   Python 代码为大模型提供，并通过 ruff 进行了规范和改正代码

通过运行 python main.py 命令，读取了 contacts.txt 文件的内容，进行数据处理后，输出了一个 emails.txt 文件，

```
(week04) apple@appledeMacBook-Pro week04 % python main.py
邮件通知文件已生成。
(week04) apple@appledeMacBook-Pro 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: <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。
---
```

week04 > emails.txt

```
1  to: <guofurong@126.com>
2  尊敬的郭芙蓉女士，您的会员资格即将到期，请及时续费。
3  ---
4  to: <lvqinghou@126.com>
5  尊敬的吕轻侯先生，您的会员资格即将到期，请及时续费。
6  ---
7  to: <baizhantang@163.com>
8  尊敬的白展堂先生，您的会员资格即将到期，请及时续费。
9  ---
10 to: <lixiulian@163.com>
11 尊敬的李秀莲先生，您的会员资格即将到期，请及时续费。
12 ---
13 to: <tongxiangyu@163.com>
14 尊敬的佟湘玉女士，您的会员资格即将到期，请及时续费。
15 ---
16 to: <zhuwushuang@163.com>
17 尊敬的祝无双女士，您的会员资格即将到期，请及时续费。
18 ---
```

4. 运行 python -m pdb main.py 命令，通过调试模式启动 Python 解释器，执行 main.py 里的代码

```
(week04) apple@appledeMacBook-Pro week04 % python -m pdb main.py
> /Users/apple/repo/week04/main.py(1)<module>()
```

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

l：显示当前代码行周围的代码片段，方便查看上下文。多次输入可翻页查看更多代码。

```
(week04) apple@appledeMacBook-Pro week04 % python -m pdb main.py
> /Users/apple/repo/week04/main.py(1)<module>()
-> def read_contacts(file_path):
(Pdb) l
  1  -> def read_contacts(file_path):
  2         try:
  3             contacts = []
  4             with open(file_path, "r", encoding="utf-8") as file:
  5                 for line in file:
  6                     parts = line.strip().split()
  7                     if len(parts) == 3:
  8                         name, gender, email = parts
  9                         contacts.append((name, gender, email))
 10             return contacts
 11         except FileNotFoundError:
```

n：执行当前行代码，然后停在下一行。

```
(Pdb) n
> /Users/apple/repo/week04/main.py(19)<module>()
-> def generate_emails(contacts):
(Pdb) l
 14          except Exception as e:
 15              print(f"错误：发生未知错误：{e}")
 16              return []
 17
 18
 19  -> def generate_emails(contacts):
 20          def get_domain(email):
 21              return email.split("@")[1]
 22
 23          def get_username(email):
 24              return email.split("@")[0]
```

ll：从当前行开始展示更多代码.

```
(Pdb) ll
  1     def read_contacts(file_path):
  2         try:
  3             contacts = []
  4             with open(file_path, "r", encoding="utf-8") as file:
  5                 for line in file:
  6                     parts = line.strip().split()
  7                     if len(parts) == 3:
  8                         name, gender, email = parts
  9                         contacts.append((name, gender, email))
 10             return contacts
 11         except FileNotFoundError:
 12             print("错误：未找到联系人文件！")
 13             return []
 14         except Exception as e:
 15             print(f"错误：发生未知错误：{e}")
 16             return []
 17
 18
 19  -> def generate_emails(contacts):
 20         def get_domain(email):
 21             return email.split("@")[1]
 22
 23         def get_username(email):
```

```
(Pdb) l .
 42             print("邮件通知文件已生成。")
 43         except Exception as e:
 44             print(f"错误：写入邮件通知文件时出错：{e}")
 45
 46
 47  -> if __name__ == "__main__":
 48         contacts = read_contacts("contacts.txt")
 49         if contacts:
 50             email_content = generate_emails(contacts)
 51             write_emails("emails.txt", email_content)
```

l 1,5:显示 1-5 行。

```
(Pdb) l 1,5
  1     def read_contacts(file_path):
  2         try:
  3             contacts = []
  4             with open(file_path, "r", encoding="utf-8") as file:
  5                 for line in file:
```

p:打印表达式的值。

```
(Pdb) p read_contacts
<function read_contacts at 0x10906afc0>
(Pdb)
<function read_contacts at 0x10906afc0>
(Pdb) s
> /Users/apple/repo/week04/main.py(38)<module>()
-> def write_emails(file_path, email_content):
(Pdb) n
> /Users/apple/repo/week04/main.py(47)<module>()
-> if __name__ == "__main__":
```

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

q:终止调试会话，退出 pdb 调试器。

```
(Pdb) q
(week04) apple@appledeMacBook-Pro week04 %
```

c：继续执行程序，直到遇到下一个断点或者程序结束，可快速跳过不需要单步调试的代码段。

```
(Pdb) c
邮件通知文件已生成。
The program finished and will be restarted
> /Users/apple/repo/week04/main.py(1)<module>()
-> def read_contacts(file_path):
```

5. Python 基本概念

Python 语法保留字 (reserved key words)：False、None、True、and、as、assert、async、await、break、class、continue、def、del、elif、else、except、finally、for、from、global、if、import、in、is、lambda、nonlocal、not、or、pass、raise、return、try、while、with、yield。

语句 (statement)：执行一个操作的代码行，类似于 sentence，可以包含表达式。

表达式 (expression)：构成语句的元素，不能包含语句。

缩进 (indent)：Python 通过缩进表示代码的层级。

局部变量 (local variable)：在函数内部定义，仅在函数内可见。

```
(Pdb) wat()
Local variables:
  __builtins__: dict = {…
  __file__: pdb._ScriptTarget = '/Users/apple/repo/week04/main.py'
  __name__: str = '__main__'
  __pdb_convenience_variables: dict = {…
  __spec__: NoneType = None
  generate_emails: function = <function generate_emails at 0x10ce1f420>
  read_contacts: function = <function read_contacts at 0x10ce1eca0>
  wat: wat.inspection.inspection.Wat = <WAT Inspector object>
  write_emails: function = <function write_emails at 0x10ce47b00>
```

全局变量 (global variable)：在模块顶层定义，整个文件可见。

```
(Pdb) wat.globals
Global variables:
  __builtins__: dict = {…
  __file__: pdb._ScriptTarget = '/Users/apple/repo/week04/main.py'
  __name__: str = '__main__'
  __pdb_convenience_variables: dict = {…
  __spec__: NoneType = None
  contacts: list = […
  generate_emails: function = <function generate_emails at 0x10ce1f420>
  read_contacts: function = <function read_contacts at 0x10ce1eca0>
  wat: wat.inspection.inspection.Wat = <WAT Inspector object>
  write_emails: function = <function write_emails at 0x10ce47b00>
```

LEGB 规则：变量查找顺序

Local （局部作用域）→ Enclosing （外层函数的局部变量）→ Global （全局作用域）→ Built-in（内置作用域）

函数 (function) 的定义 (define) 和调用 (call)：使用 def 定义函数，使用括号调用。

字面值 (literal) ：字符串、整数、列表、字典{}和元组

```
(Pdb) p{'a':1}
{'a': 1}
```

字典字面值

运算符 (operator)：==、if、else、+、-、>、<、名称访问运算符.、调用运算符（）。

形参 (parameter)：函数定义时声明的变量（如 def add(a, b) 中的 a 和 b）

实参 (argument)：函数调用时传递的值（如 add(3, 5) 中的 3 和 5）。

返回值 (return value)： return 后的为返回值，有些函数没有返回值。

对象 (object)：Python 所管理的一切，wat 可以深入检查 python 的对象。

类型 (type)：对象的类别，用 type 查看

属性 (attribute)：对象存储的数据。

方法 (method)：对象的行为。

```
value: [
    ('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'),
]
type: list
len: 6

Public attributes:
  def append(object, /) # Append object to the end of the list.
  def clear() # Remove all items from list.
  def copy() # Return a shallow copy of the list.
  def count(value, /) # Return number of occurrences of value.
  def extend(iterable, /) # Extend list by appending elements from the iterable.
  def index(value, start=0, stop=9223372036854775807, /) # Return first index of
  …
  def insert(index, object, /) # Insert object before index.
  def pop(index=-1, /) # Remove and return item at index (default last).…
  def remove(value, /) # Remove first occurrence of value.…
  def reverse() # Reverse *IN PLACE*.
  def sort(*, key=None, reverse=False) # Sort the list in ascending order and re
```