

```
MINGW64:/c/Users/hehy/week04
main pushes to main (up to date)
(base)
hehy@LAPTOP-5EC1MMBN MINGW64 ~/week04 (main)
$ ls -l
total 25
-rw-r--r-- 1 hehy 197609 72 3月 31 10:20 environment.yml
-rw-r--r-- 1 hehy 197609 18805 3月 31 10:15 LICENSE
-rw-r--r-- 1 hehy 197609 2239 3月 31 10:15 README.md
(base)
hehy@LAPTOP-5EC1MMBN MINGW64 ~/week04 (main)
$ cat environment.yml
name: week04
channels:
  - conda-forge
dependencies:
  - python=3.12
(base)
hehy@LAPTOP-5EC1MMBN MINGW64 ~/week04 (main)
$ conda env create
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): - |
```

Python处理联系人信息并生成邮件

新建的 `main.py` 文件代码如下:

```
python 复制
# 读取联系人信息并处理
contacts = []
with open('contacts.txt', 'r', encoding='utf-8') as f:
    for line in f:
        line = line.strip()
        if not line:
            continue
        parts = line.split()
        if len(parts) != 3:
            continue # 跳过格式不正确的行
        name, gender, email = parts
        # 分割邮箱为用户名和域名
        try:
            username, domain = email.split('@')
        except ValueError:
            continue # 邮箱格式错误, 跳过
        contacts.append({
            'name': name,
            'gender': gender,
            'email': email,
            'username': username,
            'domain': domain
        })

# 排序: 先按域名, 再按用户名
sorted_contacts = sorted(contacts, key=lambda x: (x['domain'], x['username']))

# 生成邮件内容
lines = []
for contact in sorted_contacts:
    salutation = '先生' if contact['gender'] == '男' else '女士'
    lines.append(f"to: <{contact['email']}>")
    lines.append(f"尊敬的{contact['name']}{salutation}, 您的会员资格即将到期, 请及时续费。")
    lines.append("---")

content = '\n'.join(lines)

# 写入emails.txt
with open('emails.txt', 'w', encoding='utf-8') as f:
    f.write(content)
```

```

(base)
hehy@LAPTOP-5EC1MMBN MINGW64 ~/week04 (main)
$ ls -l
total 25
-rw-r--r-- 1 hehy 197609 72 3月 31 10:20 environment.yml
-rw-r--r-- 1 hehy 197609 18805 3月 31 10:15 LICENSE
done
#
# To activate this environment, use
#
#     $ conda activate week04
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(base)
hehy@LAPTOP-5EC1MMBN MINGW64 ~/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
hehy@LAPTOP-5EC1MMBN MINGW64 ~/week04 (main)
$ conda activate week04
(week04)
hehy@LAPTOP-5EC1MMBN MINGW64 ~/week04 (main)
$ python mian.py
D:\anaconda3\envs\week04\python.exe: can't open file 'C:\\Users\\hehy\\week04\\m
ian.py': [Errno 2] No such file or directory
(week04)
hehy@LAPTOP-5EC1MMBN MINGW64 ~/week04 (main)
$ python main.py
(week04)
hehy@LAPTOP-5EC1MMBN MINGW64 ~/week04 (main)
10         continue # 跳过格式不正确的行
11         name, gender, email = parts
(Pdb) n
> c:\users\hehy\week04\main.py(3)<module>()
-> with open("contacts.txt", "r", encoding="utf-8") as f:
(Pdb) l
1     # 读取联系人信息并处理
2     contacts = []
3     -> with open("contacts.txt", "r", encoding="utf-8") as f:
4         for line in f:
5             line = line.strip()
6             if not line:
7                 continue
8             parts = line.split()
9             if len(parts) != 3:
10                 continue # 跳过格式不正确的行
11             name, gender, email = parts
(Pdb) p contacts
[]
(Pdb) p read_contacts
*** NameError: name 'read_contacts' is not defined
(Pdb) s
--Call--
> <frozen codecs>(309).__init__()
(Pdb) |

```

这个代码主要分为几个步骤：**读取联系人信息 → 处理数据 → 排序 → 生成邮件内容 → 写入文件**。下面我会用最通俗易懂的方式，分步骤为你详细解释：

```
---

#### 一、读取联系人信息（代码第一部分）
```python
contacts = []
with open('contacts.txt', 'r', encoding='utf-8') as f:
 for line in f:
 line = line.strip() # 去掉每行首尾的空格和换行符
 if not line:
 continue # 跳过空行
 parts = line.split() # 按空格分割成列表，例如 ['白展堂', '男',
'baizhantang@163.com']
 if len(parts) != 3:
 continue # 跳过格式不正确的行（比如字段不足 3 个）
 name, gender, email = parts # 解包赋值给变量
 # 分割邮箱为用户名和域名
 try:
 username, domain = email.split('@') # 例如 'baizhantang@163.com' →
['baizhantang', '163.com']
 except ValueError:
 continue # 如果邮箱没有@符号，跳过这一行
 # 把联系人信息存到字典里，方便后续处理
 contacts.append({
 'name': name,
 'gender': gender,
 'email': email,
 'username': username,
 'domain': domain
 })
...
```
```

这段代码做了什么？

1. **打开文件**：`with open(...)` 会自动关闭文件，避免忘记关闭。
2. **逐行读取**：`for line in f` 循环读取每一行。
3. **清洗数据**：用 `strip()` 去掉多余空格和换行符。
4. **分割字段**：用 `split()` 按空格分割姓名、性别、邮箱。
5. **验证格式**：确保每行有 3 个字段，否则跳过。
6. **处理邮箱**：将邮箱拆分为 `username`（用户名）和 `domain`（域名），例如 `baizhantang@163.com` → `username='baizhantang'`，`domain='163.com'`。
7. **存储数据**：把所有联系人信息存到 `contacts` 列表中，每个联系人是一个字典。

二、排序处理（代码第二部分）

```
```python
sorted_contacts = sorted(contacts, key=lambda x: (x['domain'], x['username']))
```
```

这段代码的作用是什么？

- **排序规则**：先按邮箱的 `domain`（域名）排序，再按 `username`（用户名）排序。
- **为什么 `126.com` 会排在 `163.com` 前面？**
 - 因为字符串比较是按字母顺序的，`126.com` 的第一个字符是 `1`，而 `163.com` 的第四个字符是 `6`，所以 `126` 比 `163` “小”，会排在前面。
- **用户名排序**：例如 `guofurong` 和 `lvqinghou` 按字母顺序比较，`g` 在 `l` 前面，所以 `guofurong` 先出现。

三、生成邮件内容（代码第三部分）

```
```python
lines = []
for contact in sorted_contacts:
 salutation = '先生' if contact['gender'] == '男' else '女士' # 根据性别选择称呼
 lines.append(f'to: <{contact['email']}>') # 添加邮箱地址
 lines.append(f'尊敬的{contact['name']}{salutation}，您的会员资格即将到期，请及时续费。') # 添加正文
 lines.append("---") # 添加分隔线

content = '\n'.join(lines) # 将列表中的每一行用换行符连接成一个字符串
```
```

这段代码做了什么？

1. **遍历排序后的联系人**：逐个处理每个联系人。
2. **生成称呼**：根据性别选择“先生”或“女士”。
3. **拼接邮件内容**：按题目要求的格式生成 `to: <邮箱>` 和正文。
4. **添加分隔线**：每封邮件后用 `---` 分隔。
5. **合并所有内容**：用 `'\n'.join(lines)` 把列表中的每一行合并成一个完整的字符串。

四、写入文件（代码第四部分）

```
```python
with open('emails.txt', 'w', encoding='utf-8') as f:
 f.write(content)
```
```

这段代码的作用是什么？

- **创建新文件**：`emails.txt`，如果文件已存在会覆盖它。
- **写入内容**：将之前生成的 `content` 字符串写入文件。
- **指定编码**：`encoding='utf-8'` 确保中文字符正常显示。

五、最终输出效果

生成的 `emails.txt` 内容如下：

'''

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。

... (其他内容)

'''

为什么顺序是这样的？

1. **域名排序**：所有 `126.com` 的邮箱会排在 `163.com` 前面。
2. **用户名排序**：在同一个域名下，按用户名字母顺序排序（例如 `guofurong` 在 `lvqinghou` 前面）。

六、新手常见疑问解答

1. 为什么要用 `try...except` 处理邮箱？

- 如果某行邮箱格式错误（比如没有 `@` 符号），`email.split('@')` 会报错，用 `try...except` 可以跳过这一行，避免程序崩溃。

2. `lambda x: (x['domain'], x['username'])` 是什么意思？

- 这是一个匿名函数，告诉 `sorted()` 函数按什么规则排序。`x` 代表列表中的每个联系人字典，先按 `domain` 排序，`domain` 相同时再按 `username` 排序。

3. 为什么最后多了一个 `---`？

- 代码中每处理完一个联系人都会添加 `---`，包括最后一个联系人。如果想去掉最后的 `---`，可以在生成 `content` 前删掉最后一行（比如 `lines = lines[:-1]`）。

希望这个解释能帮你彻底理解代码！如果有其他问题，欢迎随时提问 😊