

将上一级目录里的文件复制到当前目录里的命令

```
(base) xzy01@xzy MINGW64 ~/repo/week04 (main)
$ cp ../myproject/environment.yml ./

(base) xzy01@xzy MINGW64 ~/repo/week04 (main)
$ ls -l
total 25
-rw-r--r-- 1 xzy01 197609 18805 Mar 29 17:32 LICENSE
-rw-r--r-- 1 xzy01 197609 2239 Mar 29 17:32 README.md
-rw-r--r-- 1 xzy01 197609 87 Mar 29 17:38 environment.yml
```

*

```
xzy01@xzy MINGW64 ~/repo/week04 (main)
$ python main.py
邮件已成功写入 emails.txt。
(week04)
xzy01@xzy MINGW64 ~/repo/week04 (main)
$ ls -l
total 31
-rw-r--r-- 1 xzy01 197609 18805 Mar 29 17:32 LICENSE
-rw-r--r-- 1 xzy01 197609 2239 Mar 29 17:32 README.md
-rw-r--r-- 1 xzy01 197609 202 Mar 29 17:43 contacts.txt
-rw-r--r-- 1 xzy01 197609 666 Mar 29 19:10 emails.txt
-rw-r--r-- 1 xzy01 197609 72 Mar 29 17:41 environment.yml
-rw-r--r-- 1 xzy01 197609 1495 Mar 29 19:05 main.py
(week04)
xzy01@xzy 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: <lixuilian@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)

```
xzy01@xzy MINGW64 ~/repo/week04 (main)
$ python -m pdb main.py
> c:\users\xzy01\repo\week04\main.py(1)<module>()
-> def read_contacts(file_path):
(Pdb) |
```

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

→ wat / ***

```
NAMEError: name 'wat' is not defined
(Pdb) import wat
(Pdb) wat / write_emails

value: <function write_emails at 0x0000011E92C07600>
type: function
signature: def write_emails(emails, output_file)
```

main.py 代码解释

1. read_contacts 函数

```
python ^
def read_contacts(file_path):
    try:
        with open(file_path, "r", encoding="utf-8") as file:
            contacts = []
            for line in file:
                name, gender, email = line.strip().split()
                contacts.append((name, gender, email))
            return contacts
    except FileNotFoundError:
        print(f"错误: 未找到 {file_path} 文件。")
        return []
    except Exception as e:
        print(f"错误: 读取文件时发生未知错误 - {e}")
        return []
```

- **功能:** 该函数用于读取指定路径下的联系人信息文件。
- **参数:** `file_path` 是要读取的文件的途径。
- **代码逻辑:**
 - 使用 `with` 语句打开文件，确保文件在使用完后能正确关闭。
 - 遍历文件的每一行，使用 `strip()` 方法去除行首尾的空白字符，再使用 `split()` 方法按空格分割每行内容，得到姓名、性别和邮箱。
 - 将每个联系人的信息以元组 `(name, gender, email)` 的形式添加到 `contacts` 列表中。

- 若文件不存在，捕获 `FileNotFoundError` 异常并输出错误信息，返回空列表。
- 若出现其他未知错误，捕获 `Exception` 异常并输出错误信息，返回空列表。

2. generate_emails 函数

```
python ^  
  
def generate_emails(contacts):  
    emails = []  
    for name, gender, email in sorted(  
        contacts, key=lambda x: (x[2].split("@")[1], x[2].split("@")[0])  
    ):  
        title = "先生" if gender == "男" else "女士"  
        email_content = f"to: <{email}>\n尊敬的{name}{title}, 您的会员资格即将到期, 请及时续费。 \n---"  
        emails.append(email_content)  
    return emails
```

- **功能：**该函数用于根据联系人信息生成会员到期提醒邮件内容。
- **参数：**`contacts` 是一个包含联系人信息的列表，每个元素是一个元组 (`name`, `gender`, `email`)。
- **代码逻辑：**
 - 使用 `sorted()` 函数对 `contacts` 列表进行排序，排序规则是先按邮箱域名 (`x[2].split("@")[1]`) 排序，再按邮箱用户名 (`x[2].split("@")[0]`) 排序。
 - 遍历排序后的联系人列表，根据性别确定称呼（“先生”或“女士”）。
 - 使用 f-string 生成邮件内容，包含收件人邮箱、✓ 和会员到期提醒信息。
 - 将生成的邮件内容添加到 `emails` 列表中。
 - 最后返回 `emails` 列表。

3. write_emails 函数

```
python ^  
  
def write_emails(emails, output_file):  
    try:  
        with open(output_file, "w", encoding="utf-8") as file:  
            for email in emails:  
                file.write(email + "\n")  
        print(f"邮件已成功写入 {output_file}。")  
    except Exception as e:  
        print(f"错误: 写入文件时发生未知错误 - {e}")
```

- **功能：**该函数用于将生成的邮件内容写入指定的文件。
- **参数：**`emails` 是包含邮件内容的列表，`output_file` 是要写入的文件路径。
- **代码逻辑：**
 - 使用 `with` 语句以写入模式打开文件。
 - 遍历 `emails` 列表，将每个邮件内容写入文件，并在末尾添加换行符。
 - 若写入过程中出现错误，捕获 `Exception` 异常并输出错误信息。
 - 若写入成功，输出成功信息。

4. 主程序

```
python ^  
  
if __name__ == "__main__":  
    contacts = read_contacts("contacts.txt")  
    if contacts:  
        email_contents = generate_emails(contacts)  
        write_emails(email_contents, "emails.txt")
```

- **功能：**程序的入口点，调用上述三个函数完成联系人信息读取、邮件内容生成和写入文件的操作。
- **代码逻辑：**
 - 调用 `read_contacts` 函数读取 `contacts.txt` 文件中的联系人信息。
 - 若成功读取到联系人信息（即 `contacts` 列表不为空），调用 `generate_emails` 函数生成邮件内容。
 - 最后调用 `write_emails` 函数将邮件内容写入 `emails.txt` 文件。