

## Week04

1. 用 VS Code 打开项目目录，新建一个 environment.yml 文件，指定安装 Python3.12，然后运行 conda env create 命令创建 Conda 环境

```
(base)
Zhaogs@LAPTOP-MGC7R7G2 MINGW64 ~/Desktop/金融编程/week04 (main)
$ cd week04
bash: cd: week04: No such file or directory
(base)
Zhaogs@LAPTOP-MGC7R7G2 MINGW64 ~/Desktop/金融编程/week04 (main)
$ code .
(base)
Zhaogs@LAPTOP-MGC7R7G2 MINGW64 ~/Desktop/金融编程/week04 (main)
$ cp C:/Users/Zhaogs/repo/myproject/environment.yml ./
(base)
Zhaogs@LAPTOP-MGC7R7G2 MINGW64 ~/Desktop/金融编程/week04 (main)
$ ls -l
total 25
-rw-r--r-- 1 Zhaogs 197121 87 3月 27 10:25 environment.yml
-rw-r--r-- 1 Zhaogs 197121 18805 3月 27 09:59 LICENSE
-rw-r--r-- 1 Zhaogs 197121 2239 3月 27 09:59 README.md
(base)
Zhaogs@LAPTOP-MGC7R7G2 MINGW64 ~/Desktop/金融编程/week04 (main)
$ cat environment.yml
name: myproject
channels:
  - conda-forge
dependencies:
  - python=3.12
  - pandas(base)
Zhaogs@LAPTOP-MGC7R7G2 MINGW64 ~/Desktop/金融编程/week04 (main)
$ conda env create

CondaValueError: prefix already exists: E:\Anaconda3\envs\myproject

(base)
Zhaogs@LAPTOP-MGC7R7G2 MINGW64 ~/Desktop/金融编程/week04 (main)
$ conda poj create

CommandNotFoundError: No command 'conda poj'.
```

2. 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。

```
contacts.txt
1 白展堂 男 baizhantang@163.com
2 佟湘玉 女 tongxiangyu@163.com
3 吕轻侯 男 lvqinghou@126.com
4 郭芙蓉 女 guofurong@126.com
5 李秀莲 男 lixiulian@163.com
6 祝无双 女 zhuwushuang@163.com
```

3. 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

```
main.py > ...
1  def read_contacts(filename):
2      return contacts
3
4  def sort_contacts(contacts):
5      # Sort by domain and then by username
6      contacts.sort(key=lambda x: (x[2].split('@')[1], x[2].split('@')[0]))
7      return contacts
8
9  def generate_emails(contacts):
10     emails_content = []
11     for name, gender, email in contacts:
12         if gender == '男':
13             gender_title = '先生'
14         else:
15             gender_title = '女士'
16         message = f"to: <{email}>\n尊敬的{name}{gender_title}, 您的会员资格即将到期, 请及时续费。"
17         emails_content.append(message)
18     return emails_content
19
20 def write_emails(emails_content, filename):
21     with open(filename, 'w', encoding='utf-8') as file:
22         file.writelines(emails_content)
23
24 if __name__ == "__main__":
25     # Step 1: Read contacts from file
26     contacts = read_contacts('contacts.txt')
27
28     # Step 2: Sort contacts
29     sorted_contacts = sort_contacts(contacts)
30
31     # Step 3: Generate email content
32     emails_content = generate_emails(sorted_contacts)
33
34     # Step 4: Write emails to file
35     write_emails(emails_content, 'emails.txt')
```

4. 在 VS Code 扩展商店里安装 Ruff 扩展，并配置

```
C: > Users > Zhaogs > AppData > Roaming > Code > User > settings.json > ...
1  {
2      "workbench.colorTheme": "Visual Studio Dark",
3      "workbench.iconTheme": "vs-minimal",
4      "python.defaultInterpreterPath": "E:\\Anaconda3\\python.exe",
5      "[python]": {
6          "editor.formatOnSave": true,
7          "editor.codeActionsOnSave": {
8              "source.fixAll": "explicit",
9              "source.organizeImports": "explicit"
10         },
11         "editor.defaultFormatter": "charliermarsh.ruff"
12     },
13     "notebook.formatOnSave.enabled": true,
14     "notebook.codeActionsOnSave": {
15         "notebook.source.fixAll": "explicit",
16         "notebook.source.organizeImports": "explicit"
17     },
18 }
```

5. 运行 python main.py 命令：顺利，且结果符合预期

```
发行说明: 1.98.2  environment.yml U  contacts.txt U  ma
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: <lixiluan@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 ---
19
```

## 6. 运行 python -m pdb main.py 命令

```
$ python -m pdb main.py
1 -> def read_contacts(filename):
2     contacts = []
3     with open(filename, 'r', encoding='utf-8') as file:
4         for line in file:
5             name, gender, email = line.strip().split()
25 def write_emails(emails_content, filename):
26     with open(filename, 'w', encoding='utf-8') as file:
27         file.writelines(emails_content)
28
29 if __name__ == "__main__":
30     # Step 1: Read contacts from file
31     contacts = read_contacts('contacts.txt')
32
33     # Step 2: Sort contacts
34     sorted_contacts = sort_contacts(contacts)
35
36     # Step 3: Generate email content
37     emails_content = generate_emails(sorted_contacts)
38
39     # Step 4: Write emails to file
40     write_emails(emails_content, 'emails.txt')
41
42     print("Emails have been generated successfully.")
(Pdb) p contacts
*** NameError: name 'contacts' is not defined
(Pdb) p sort_contacts
*** NameError: name 'sort_contacts' is not defined
(Pdb) l 30, 39
30     # Step 1: Read contacts from file
31     contacts = read_contacts('contacts.txt')
32
33     # Step 2: Sort contacts
34     sorted_contacts = sort_contacts(contacts)
35
36     # Step 3: Generate email content
37     emails_content = generate_emails(sorted_contacts)
38
39     # Step 4: Write emails to file
(Pdb) p _name__
*** NameError: name '_name__' is not defined
(Pdb) l .
4         for line in file:
5             name, gender, email = line.strip().split()
6             contacts.append((name, gender, email))
7         return contacts
8
9 -> def sort_contacts(contacts):
10     # Sort by domain and then by username
11     contacts.sort(key=lambda x: (x[2].split('@')[1], x[2].split('@')[0]))
12     return contacts
13
14 def generate_emails(contacts):
(Pdb) n
> c:\users\zhaogs\desktop\金融编程\week04\main.py(14)<module>()
-> def generate_emails(contacts):
(Pdb) p contacts
```



7. 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

```
zhaogs@LAPTOP-MGC7R7G2 MINGW64 ~/Desktop/金融编程/week04 (main)
$ ls -l
total 31
-rw-r--r-- 1 Zhaogs 197121 202 3月 27 10:34 contacts.txt
-rw-r--r-- 1 Zhaogs 197121 666 3月 27 11:07 emails.txt
-rw-r--r-- 1 Zhaogs 197121 91 3月 27 23:08 environment.yml
-rw-r--r-- 1 Zhaogs 197121 18805 3月 27 09:59 LICENSE
-rw-r--r-- 1 Zhaogs 197121 1433 3月 27 11:09 main.py
-rw-r--r-- 1 Zhaogs 197121 2239 3月 27 09:59 README.md
(base)
zhaogs@LAPTOP-MGC7R7G2 MINGW64 ~/Desktop/金融编程/week04 (main)
$ cat environment.yml
name: week04
channels:
  - conda-forge
dependencies:
  - python=3.12
  - wat-inspector(base)
zhaogs@LAPTOP-MGC7R7G2 MINGW64 ~/Desktop/金融编程/week04 (main)
```

## 8. 理解概念

### ① Python 语法保留字

Python 中有一些保留字, 它们具有特定的含义和用途, 不能用作变量名或其他标识符。常见的保留字包括:

- if, else, elif
- for, while
- def, class
- return, yield
- import, from
- pass, break, continue

### ②语句 (statement) 和表达式 (expression)

语句是执行某些操作或控制流程的指令, 比如

```
print("Hello, World!")
```

```
x = 5
```

表达式用来计算值的组合, 比如

```
x + y
```

### ③缩进 (indent)

Python 使用缩进来表示代码块。缩进必须一致, 通常使用 4 个空格或一个制表符 (tab), def 之后缩进的语句可以被收起来。

```
if x > 0:
```

```
    print("x is positive")
```

### ④局部变量 (local variable) vs. 全局变量 (global variable)

局部变量: 在函数内部定义的变量, 只能在该函数内部访问。

```
def my_function():
```

```
    local_var = 10
```

全局变量: 在函数外部定义的变量, 可以在整个程序中访问。

```
global_var = 20
```

```
def my_function():
```

```
print(global_var)
```

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

定义函数：使用 `def` 关键字定义函数。

```
def greet(name):
```

```
    return f"Hello, {name}!"
```

调用函数：通过函数名和括号来调用函数。

```
result = greet("Alice")
```

```
print(result)
```

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

字符串：用单引号或双引号括起来的文本。

```
message = "Hello, World!"
```

整数：没有小数点的数字。

```
age = 25
```

列表：有序的元素集合，用方括号表示。

```
numbers = [1, 2, 3, 4, 5]
```

字典：无序的键值对集合，用花括号表示。

```
person = {"name": "Alice", "age": 30}
```

元组：有序的不可变元素集合，用圆括号表示。

```
coordinates = (10, 20)
```

⑦运算符 (operator)

Python 支持多种运算符，包括算术运算符、比较运算符、逻辑运算符等。

算术运算符：

```
a = 10 b = 5
```

```
print(a + b)
```

```
print(a - b)
```

```
print(a * b)
```

```
print(a / b)
```

比较运算符：

```
print(a > b)
```

```
print(a < b)
```

```
print(a == b)
```

逻辑运算符：

```
a = True b = False
```

```
print(a and b) 与
```

```
print(a or b) 或
```

```
print(not a) 非
```

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

形参：函数定义时指定的参数。

```
def add(x, y):
```

```
    return x + y
```

实参：函数调用时传递的实际值。

```
result = add(3, 5)
```

返回值：函数执行完毕后返回的结果。

```
result = add(3, 5)
```

```
    print(result)    输出 8
```

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

对象: Python 中的一切都是对象。

类型: 对象所属的类别, 如 int, str, list 等。

```
num = 10
```

```
print(type(num))    输出 <class 'int'>
```

属性: 对象的特性或状态。

```
person = {"name": "Alice", "age": 30}
```

```
print(person["name"])    输出 Alice
```

方法: 对象的行为或功能。

```
numbers = [1, 2, 3, 4, 5]
```

```
print(numbers.append(6))    添加元素
```

```
print(numbers)    输出 [1, 2, 3, 4, 5, 6]
```