

1. Fork 第 04 周打卡仓库至我的名下,然后将我名下的这个仓库 Clone 到本地计算机里的 repo 文件夹。

```
MINGW64/c/Users/PC/repo/ x + v
(base) PC@DESKTOP-326MLMC MINGW64 ~
$ cd repo

(base) PC@DESKTOP-326MLMC MINGW64 ~/repo
$ ls -l
total 16
drwxr-xr-x 1 PC 197121 0 3月 23 13:43 mywork/
drwxr-xr-x 1 PC 197121 0 3月 23 11:56 proj1/
-rw-r--r-- 1 PC 197121 0 3月 15 17:40 script1.py
drwxr-xr-x 1 PC 197121 0 3月 15 19:46 week01/
drwxr-xr-x 1 PC 197121 0 3月 22 21:05 week02/
drwxr-xr-x 1 PC 197121 0 3月 23 14:47 week03/

(base) PC@DESKTOP-326MLMC MINGW64 ~/repo
$ git clone git@gitcode.com:GJ2025/week04.git
Cloning into 'week04'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 5 (delta 0), reused 5 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (5/5), 8.45 KiB | 279.00 KiB/s, done.

(base) PC@DESKTOP-326MLMC MINGW64 ~/repo
$ cd week04/

(base) PC@DESKTOP-326MLMC MINGW64 ~/repo/week04 (main)
$ pwd
/c/Users/PC/repo/week04
```

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。

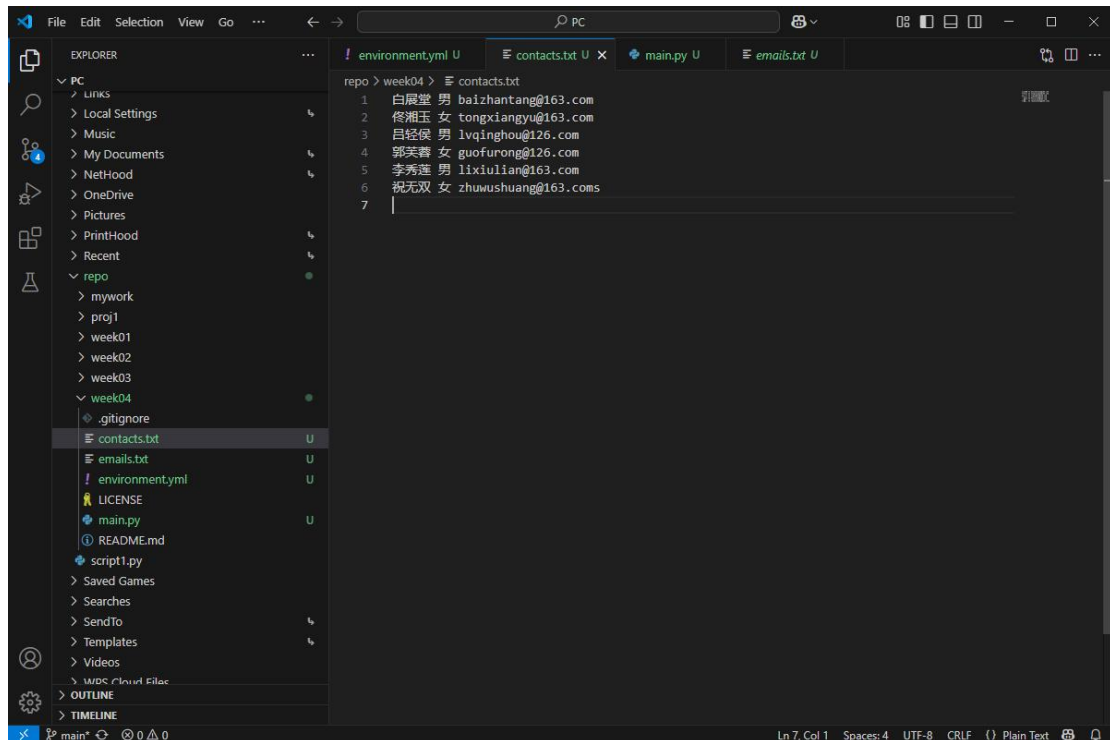
```
MINGW64/c/Users/PC/repo/ x + v
main merges with remote main
Local ref configured for 'git push':
main pushes to main (up to date)

(base) PC@DESKTOP-326MLMC MINGW64 ~/repo/week04 (main)
$ cp ../mywork/environment.yml ./

(base) PC@DESKTOP-326MLMC MINGW64 ~/repo/week04 (main)
$ ls -l
total 25
-rw-r--r-- 1 PC 197121 84 3月 29 20:51 environment.yml
-rw-r--r-- 1 PC 197121 18805 3月 29 20:47 LICENSE
-rw-r--r-- 1 PC 197121 2239 3月 29 20:47 README.md

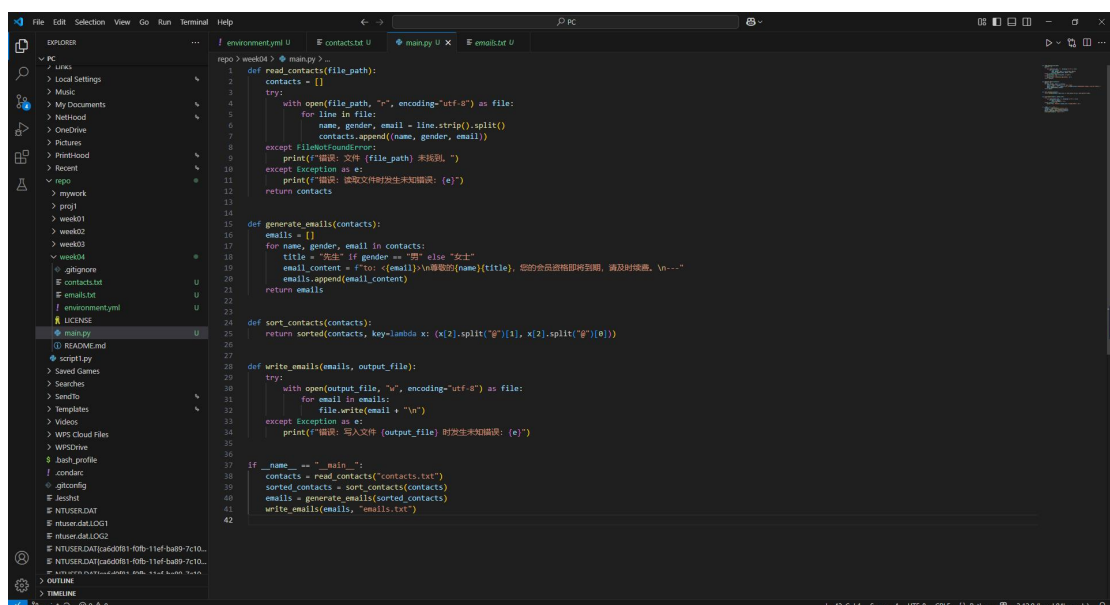
(base) PC@DESKTOP-326MLMC MINGW64 ~/repo/week04 (main)
$ cat environment.yml
name: mywork
channels:
  - conda-forge
dependencies:
done
#
# To activate this environment, use
#
# $ conda activate week04
#
# To deactivate an active environment, use
#
# $ conda deactivate
```

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。



```
1 白晨堂 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
7
```

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。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）。



```
1 def read_contacts(file_path):
2     contacts = []
3     try:
4         with open(file_path, "r", encoding="utf-8") as file:
5             for line in file:
6                 name, gender, email = line.strip().split()
7                 contacts.append((name, gender, email))
8     except FileNotFoundError:
9         print(f"错误: 文件 {file_path} 未找到。")
10    except Exception as e:
11        print(f"错误: 读取文件时发生未知错误: {e}")
12    return contacts
13
14 def generate_emails(contacts):
15    emails = []
16    for name, gender, email in contacts:
17        title = "先生" if gender == "男" else "女士"
18        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---"
19        emails.append(email_content)
20    return emails
21
22 def sort_contacts(contacts):
23    return sorted(contacts, key=lambda x: (x[2].split("@")[1], x[2].split("@")[0]))
24
25 def write_emails(emails, output_file):
26    try:
27        with open(output_file, "w", encoding="utf-8") as file:
28            for email in emails:
29                file.write(email + "\n")
30    except Exception as e:
31        print(f"错误: 写入文件 {output_file} 时发生未知错误: {e}")
32
33 if __name__ == "__main__":
34    contacts = read_contacts("contacts.txt")
35    sorted_contacts = sort_contacts(contacts)
36    emails = generate_emails(sorted_contacts)
37    write_emails(emails, "emails.txt")
```


5. 运行 `python main.py` 命令，启动 Python3.12.9 解释器，执行 `main.py` 里的代码，检查运行结果是否符合预期。

```
MINGW64~/c/Users/PC/repo/... x + -
(base) PC@DESKTOP-326MLMC MINGW64 ~/repo/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.coms
(base) PC@DESKTOP-326MLMC MINGW64 ~/repo/week04 (main)
$ conda activate week04
(week04)
PC@DESKTOP-326MLMC MINGW64 ~/repo/week04 (main)
$ python main.py
(week04)
PC@DESKTOP-326MLMC MINGW64 ~/repo/week04 (main)
$ ls -l
total 31
-rw-r--r-- 1 PC 197121 205 3月 29 20:59 contacts.txt
-rw-r--r-- 1 PC 197121 667 3月 29 21:20 emails.txt
-rw-r--r-- 1 PC 197121 72 3月 29 20:53 environment.yml
-rw-r--r-- 1 PC 197121 18805 3月 29 20:47 LICENSE
-rw-r--r-- 1 PC 197121 1444 3月 29 21:06 main.py
-rw-r--r-- 1 PC 197121 2239 3月 29 20:47 README.md
(week04)
PC@DESKTOP-326MLMC 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。
---
```

```
MINGW64~/c/Users/PC/repo/... x + -
-rw-r--r-- 1 PC 197121 205 3月 29 20:59 contacts.txt
-rw-r--r-- 1 PC 197121 667 3月 29 21:20 emails.txt
-rw-r--r-- 1 PC 197121 72 3月 29 20:53 environment.yml
-rw-r--r-- 1 PC 197121 18805 3月 29 20:47 LICENSE
-rw-r--r-- 1 PC 197121 1444 3月 29 21:06 main.py
-rw-r--r-- 1 PC 197121 2239 3月 29 20:47 README.md
(week04)
PC@DESKTOP-326MLMC 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: <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.coms>
尊敬的祝无双女士，您的会员资格即将到期，请及时续费。
---
(week04)
PC@DESKTOP-326MLMC MINGW64 ~/repo/week04 (main)
$
```

6. 运行 `python -m pdb main.py` 命令，以调试模式（debug mode）启动 Python 3.12.9 解释器，准备执行 `main.py` 里的代码。

（1）在（pdb）提示符下使用 `l`（显示代码）、`n`（执行当前行）。

```
MINGW64/c/Users/PC/repo/n  +  -
PC@DESKTOP-326MLMC MINGW64 ~/repo/week04 (main)
$ python -m pdb main.py
> c:\users\pc\repo\week04\main.py(1)<module>()
-> def read_contacts(file_path):
(Pdb) l
1  -> def read_contacts(file_path):
2      contacts = []
3      try:
4          with open(file_path, "r", encoding="utf-8") as file:
5              for line in file:
6                  name, gender, email = line.strip().split()
7                  contacts.append((name, gender, email))
8      except FileNotFoundError:
9          print(f"错误: 文件 {file_path} 未找到。")
10     except Exception as e:
11         print(f"错误: 读取文件时发生未知错误: {e}")
(Pdb) n
> c:\users\pc\repo\week04\main.py(15)<module>()
-> def generate_emails(contacts):
(Pdb) l
10     except Exception as e:
11         print(f"错误: 读取文件时发生未知错误: {e}")
12     return contacts
13
14
15 -> def generate_emails(contacts):
16     emails = []
17     for name, gender, email in contacts:
18         title = "先生" if gender == "男" else "女士"
19         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---"
```

```
MINGW64/c/Users/PC/repo/n  +  -
(Pdb) ll
1  def read_contacts(file_path):
2      contacts = []
3      try:
4          with open(file_path, "r", encoding="utf-8") as file:
5              for line in file:
6                  name, gender, email = line.strip().split()
7                  contacts.append((name, gender, email))
8      except FileNotFoundError:
9          print(f"错误: 文件 {file_path} 未找到。")
10     except Exception as e:
11         print(f"错误: 读取文件时发生未知错误: {e}")
12     return contacts
13
14
15 -> def generate_emails(contacts):
16     emails = []
17     for name, gender, email in contacts:
18         title = "先生" if gender == "男" else "女士"
19         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---"
20         emails.append(email_content)
21     return emails
22
23
24     def sort_contacts(contacts):
25         return sorted(contacts, key=lambda x: (x[2].split("@")[1], x[2].split("@")[0]))
26
27
28     def write_emails(emails, output_file):
29         try:
```

(2) 在 (pdb) 提示符下使用 p (打印表达式)、s (步入调用)。

```
MINGW64/c/Users/PC/repo/  x  +  v
(Pdb) p contacts
*** NameError: name 'contacts' is not defined
(Pdb) p read_contacts
<function read_contacts at 0x000001ECC5D83A60>
(Pdb) s
> c:\users\pc\repo\week04\main.py(24)<module>()
-> def sort_contacts(contacts):
(Pdb) n
> c:\users\pc\repo\week04\main.py(28)<module>()
-> def write_emails(emails, output_file):
(Pdb) p sort_contacts
<function sort_contacts at 0x000001ECC5D83740>
(Pdb) p generate_emails
<function generate_emails at 0x000001ECC5D837E0>
(Pdb) p file_path
*** NameError: name 'file_path' is not defined
(Pdb) p contacts
*** NameError: name 'contacts' is not defined
(Pdb) p write_emails
*** NameError: name 'write_emails' is not defined
(Pdb) s
> c:\users\pc\repo\week04\main.py(37)<module>()
-> if __name__ == "__main__":
(Pdb) p write_emails
<function write_emails at 0x000001ECC5D83880>
(Pdb) l
32         file.write(email + "\n")
33     except Exception as e:
34         print(f"错误: 写入文件 {output_file} 时发生未知错误: {e}")
35
```

```
MINGW64/c/Users/PC/repo/  x  +  v
(Pdb) l .
32         file.write(email + "\n")
33     except Exception as e:
34         print(f"错误: 写入文件 {output_file} 时发生未知错误: {e}")
35
36
37 -> if __name__ == "__main__":
38     contacts = read_contacts("contacts.txt")
39     sorted_contacts = sort_contacts(contacts)
40     emails = generate_emails(sorted_contacts)
41     write_emails(emails, "emails.txt")
[EOF]
(Pdb) l 1,5
1     def read_contacts(file_path):
2         contacts = []
3         try:
4             with open(file_path, "r", encoding="utf-8") as file:
5                 for line in file:
(Pdb) l 30,39
30         with open(output_file, "w", encoding="utf-8") as file:
31             for email in emails:
32                 file.write(email + "\n")
33         except Exception as e:
34             print(f"错误: 写入文件 {output_file} 时发生未知错误: {e}")
35
36
37 -> if __name__ == "__main__":
38     contacts = read_contacts("contacts.txt")
39     sorted_contacts = sort_contacts(contacts)
(Pdb) l .
```

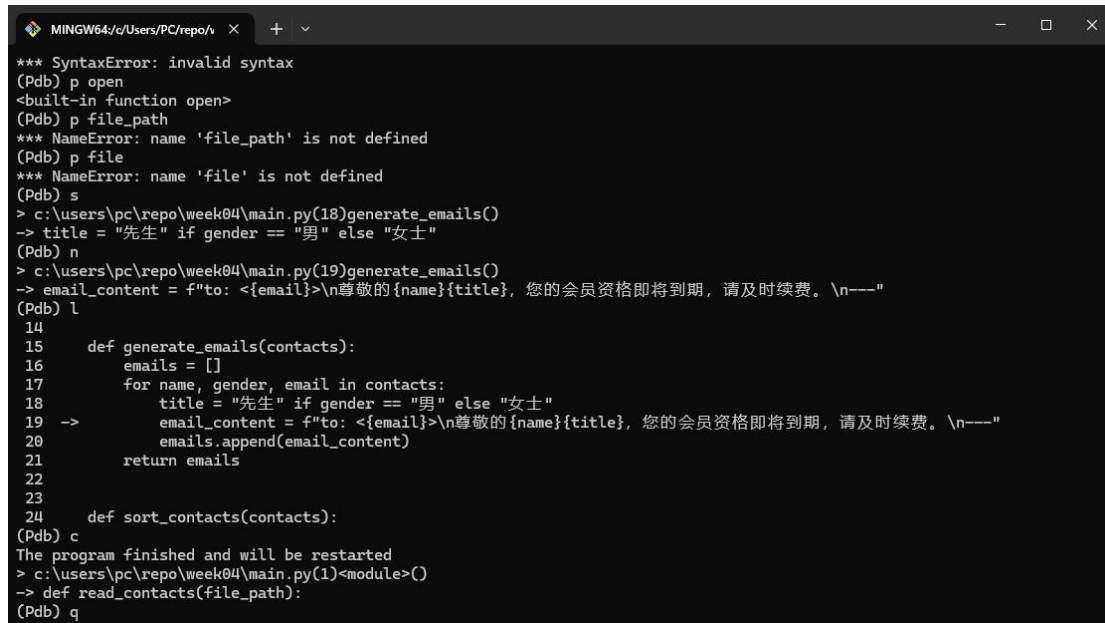


```
MINGW64/c/Users/PC/repo/a x + v
(Pdb) p __name__
'__main__'
(Pdb) p __name__ == "__main__"
True
(Pdb) n
> c:\users\pc\repo\week04\main.py(38)<module>()
-> contacts = read_contacts("contacts.txt")
(Pdb) l
33     except Exception as e:
34         print(f"错误: 写入文件 {output_file} 时发生未知错误: {e}")
35
36
37     if __name__ == "__main__":
38 ->     contacts = read_contacts("contacts.txt")
39         sorted_contacts = sort_contacts(contacts)
40         emails = generate_emails(sorted_contacts)
41         write_emails(emails, "emails.txt")
[EOF]
(Pdb) n
> c:\users\pc\repo\week04\main.py(39)<module>()
-> sorted_contacts = sort_contacts(contacts)
(Pdb) l
34         print(f"错误: 写入文件 {output_file} 时发生未知错误: {e}")
35
36
37     if __name__ == "__main__":
38 ->     contacts = read_contacts("contacts.txt")
39         sorted_contacts = sort_contacts(contacts)
40         emails = generate_emails(sorted_contacts)
41         write_emails(emails, "emails.txt")
```

(3) 在 (pdb) 提示符下使用 pp (美观打印) 。

```
MINGW64/c/Users/PC/repo/a x + v
(Pdb) p contacts.txt
*** AttributeError: 'list' object has no attribute 'txt'
(Pdb) p contacts
[('白宸堂', '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', 'lixiliu@163.com'), ('祝无双', '女', 'zhuwushuang@163.coms')]
(Pdb) pp contacts
[('白宸堂', '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', 'lixiliu@163.com'), ('祝无双', '女', 'zhuwushuang@163.coms')]
(Pdb) p type(contacts)
<class 'list'>
(Pdb) p len(contacts)
6
(Pdb) q
(week04)
PC@DESKTOP-326MLMC MINGW64 ~/repo/week04 (main)
$ python -m pdb main.py
> c:\users\pc\repo\week04\main.py(1)<module>()
-> def read_contacts(file_path):
(Pdb) l
1 -> def read_contacts(file_path):
2     contacts = []
3     try:
4         with open(file_path, "r", encoding="utf-8") as file:
5             for line in file:
6                 name, gender, email = line.strip().split()
```

(4) 在 (pdb) 提示符下使用 c (继续执行)。



```
MINGW64/c/Users/PC/repo/n x + v
*** SyntaxError: invalid syntax
(Pdb) p open
<built-in function open>
(Pdb) p file_path
*** NameError: name 'file_path' is not defined
(Pdb) p file
*** NameError: name 'file' is not defined
(Pdb) s
> c:\users\pc\repo\week04\main.py(18)generate_emails()
-> title = "先生" if gender == "男" else "女士"
(Pdb) n
> c:\users\pc\repo\week04\main.py(19)generate_emails()
-> 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---"
(Pdb) l
14
15     def generate_emails(contacts):
16         emails = []
17         for name, gender, email in contacts:
18             title = "先生" if gender == "男" else "女士"
19     ->         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---"
20             emails.append(email_content)
21         return emails
22
23     def sort_contacts(contacts):
(Pdb) c
The program finished and will be restarted
> c:\users\pc\repo\week04\main.py(1)<module>()
-> def read_contacts(file_path):
(Pdb) q
```

7. Python 基本概念

(1) Python 语法保留字 (reserved key words)

保留字即 Python 语言里被赋予特定含义的单词, 你不能把它们用作变量名、函数名等标识符。比如 if、else、for、while、def 等。

(2) 语句 (statement) 和表达式 (expression)

语句: 是 Python 程序里可执行的指令, 像赋值语句、if 语句、for 语句等。语句一般用于完成特定操作, 可进行语句嵌套。

表达式: 是能计算出值的代码片段, 像算术表达式、函数调用等, 可进行表达式嵌套。

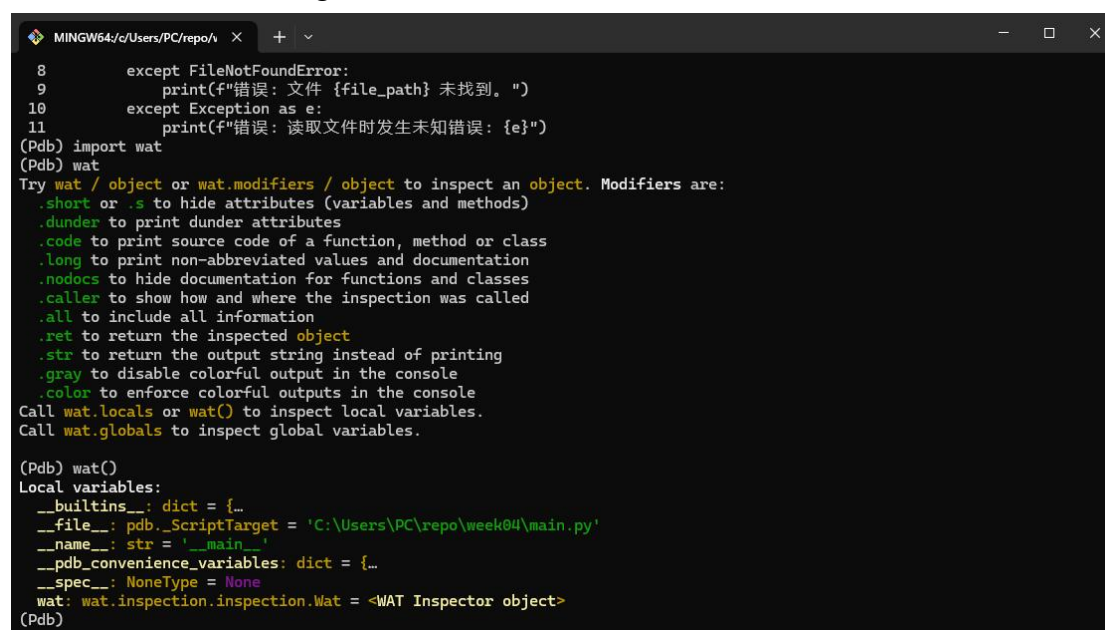
(3) 缩进 (indent)

Python 运用缩进来表示代码块, 而非使用大括号 { }。通常采用 4 个空格或者一个制表符进行缩进。缩进相同的代码属于同一个代码块。

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

局部变量：在函数或者代码块内部定义的变量，其作用域仅限于该函数或代码块。

全局变量：在函数外部定义的变量，其作用域是整个程序。若要在函数内部修改全局变量，需要使用 `global` 关键字。



```
8         except FileNotFoundError:
9             print(f"错误：文件 {file_path} 未找到。")
10        except Exception as e:
11            print(f"错误：读取文件时发生未知错误：{e}")
(Pdb) import wat
(Pdb) wat
Try wat / object or wat.modifiers / object to inspect an object. Modifiers are:
.short or .s to hide attributes (variables and methods)
.dunder to print dunder attributes
.code to print source code of a function, method or class
.long to print non-abbreviated values and documentation
.nodocs to hide documentation for functions and classes
.caller to show how and where the inspection was called
.all to include all information
.ret to return the inspected object
.str to return the output string instead of printing
.gray to disable colorful output in the console
.color to enforce colorful outputs in the console
Call wat.locals or wat() to inspect local variables.
Call wat.globals to inspect global variables.
(Pdb) wat()
Local variables:
__builtins__: dict = {...}
__file__: pdb._ScriptTarget = 'C:\Users\PC\repo\week04\main.py'
__name__: str = '__main__'
__pdb_convenience_variables__: dict = {...}
__spec__: NoneType = None
wat: wat.inspection.inspection.Wat = <WAT Inspector object>
(Pdb)
```

LEGB 规则：这是 Python 查找变量的顺序，依次为局部作用域 (Local)、闭包函数作用域 (Enclosing)、全局作用域 (Global)、内置作用域 (Built-in)。

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

函数定义：使用 `def` 关键字来定义函数，格式为 `def 函数名(参数列表)`。

函数调用：通过函数名和参数来调用函数。

(6) 字面值 (literal)

字符串 (str)：用单引号或者双引号括起来的字符序列。示例：“hello”、“world”。

整数 (int)：表示整数数值。示例：10、-20。

列表 (list)：用中括号括起来的有序元素集合，元素类型可以不同。示例：`[1, 2, "a"]`。

字典 (dict)：用大括号括起来的键值对集合。示例：`{"name": "Alice", "age": 20}`。

元组 (tuple)：用小括号括起来的有序元素集合，元素类型可以不同，且不可变。

示例：(1, 2, "a")。

(7) 运算符 (operator)

运算符用于对操作数进行运算，常见的运算符有算术运算符（+、-、*、/ 等）、比较运算符（>、<、== 等）、逻辑运算符（and、or、not）等。

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

形参：在函数定义时的参数，用于接收调用函数时传递的值。示例：

```
def add(a, b): # a 和 b 是形参
    return a + b
```

实参：在函数调用时传递给函数的实际值。示例：

```
result = add(2, 3) # 2 和 3 是实参
```

返回值：函数执行完毕后返回的结果，使用 `return` 语句返回。示例：

```
def add(a, b):
    return a + b # 返回值
```

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

对象：Python 里一切皆对象，对象是类的实例。示例：`x = 10`，`x` 就是一个整数对象。

类型：对象所属的类别，例如整数类型 `int`、字符串类型 `str` 等。可以使用 `type()` 函数查看对象的类型。

属性：对象所拥有的数据，通过 `.` 来访问。

方法：对象所拥有的函数，通过 `.` 来调用。