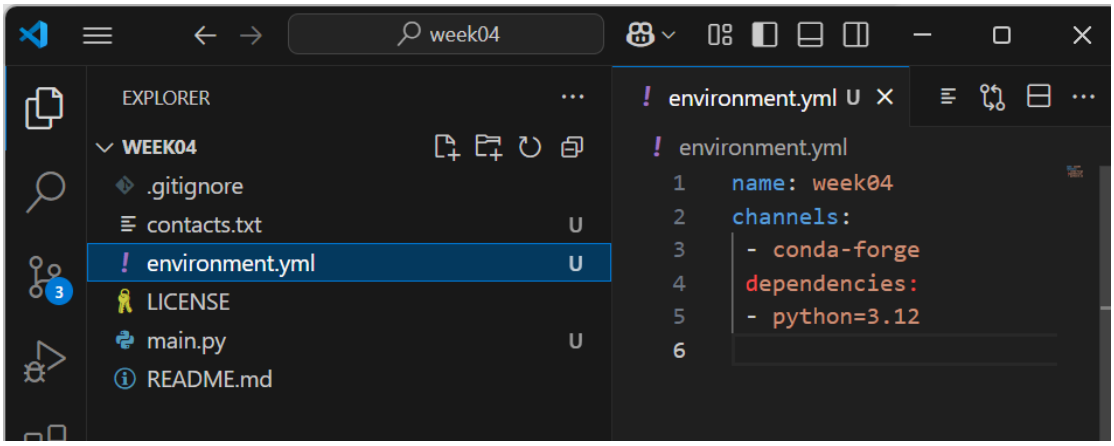
# 第三周 学习笔记——Python 数据类型 (初级)

一、 在 Fork 第 04 周打卡仓库至个人名下，然后将个人名下的仓库 Clone 到本地计算机

```
Lenovo@LAPTOP-8MNNHDHQ MINGW64 ~
$ cd E:/研究生上课资料/研一下\ 上课资料/金融编程与计算
Lenovo@LAPTOP-8MNNHDHQ MINGW64 /e/研究生上课资料/研一下 上课资料/金融编程与计算
$ git clone https://gitcode.com/twinkledehaha/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)
Unpacking objects: 100% (5/5), 8.43 KiB | 454.00 KiB/s, done.
Lenovo@LAPTOP-8MNNHDHQ MINGW64 /e/研究生上课资料/研一下 上课资料/金融编程与计算
```

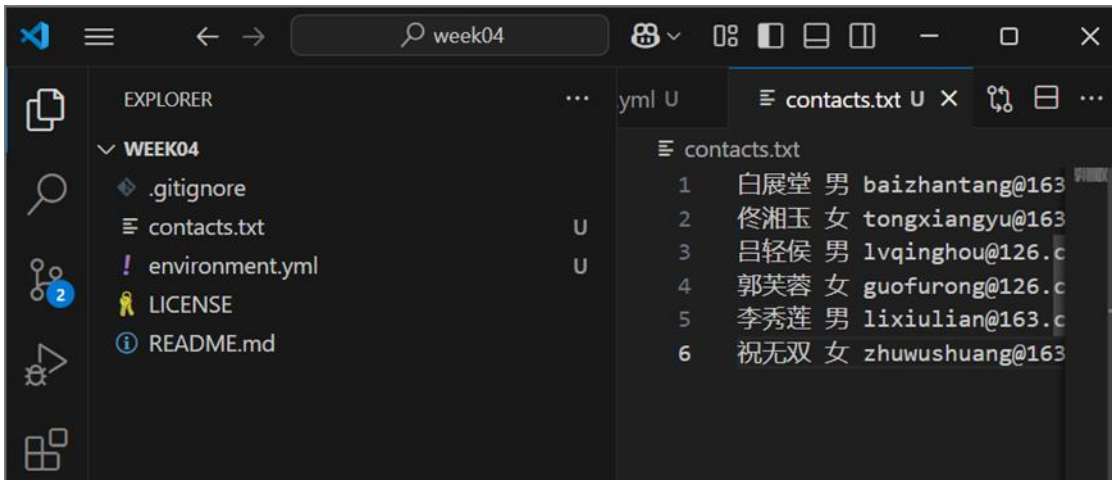
二、用 VS Code 打开项目目录，新建一个 environment.yml 文件，指定安装 Python 3.12，然后运行 conda env create 命令创建 Conda 环境




```
Lenovo@DESKTOP-C7OG2VK MINGW64 /e/研究生上课资料/研一下 上课资料
/金融编程与计算/week04
$ conda env list
# conda environments:
#
base                  *  D:\Anaconda\Anaconda3
prj1                     D:\Anaconda\Anaconda3\envs\prj1
prj2                     D:\Anaconda\Anaconda3\envs\prj2
week04                   D:\Anaconda\Anaconda3\envs\week04

(base)
```

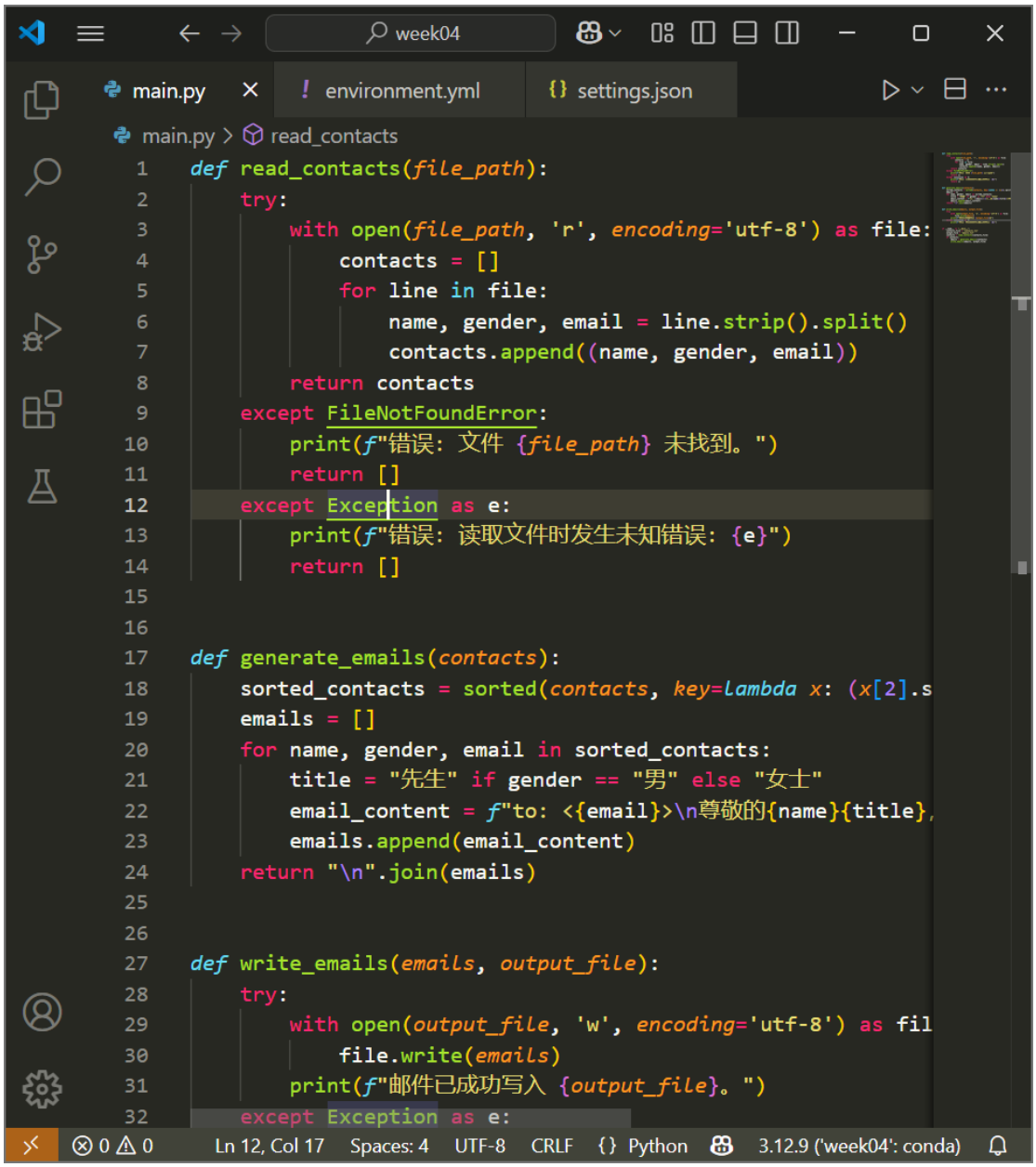
三、新建一个 contacts.txt 文件，每行写一个联系人，每个联系人都包含姓名、性别、邮箱三个字段，用空格分隔，例如：



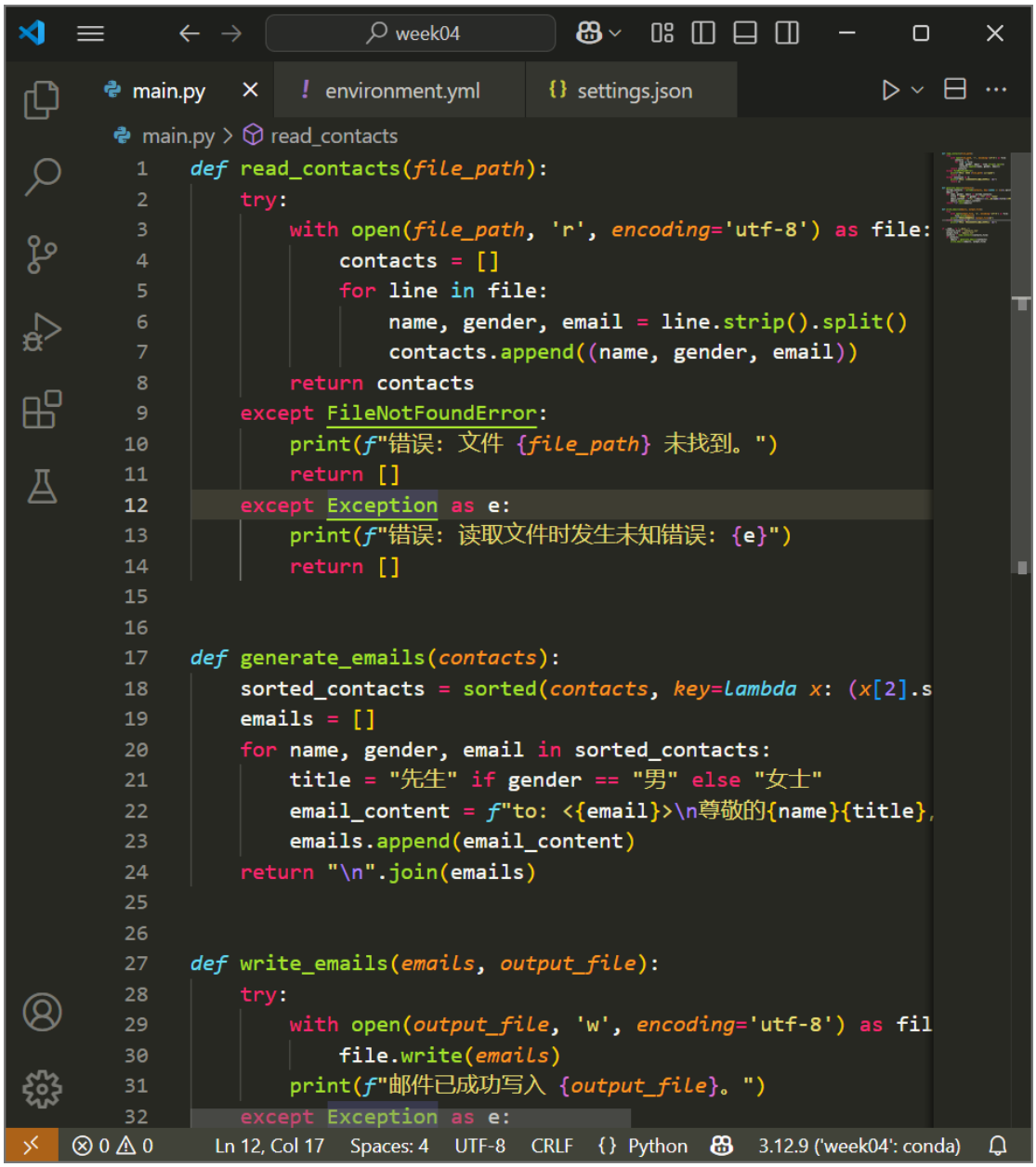

```
$ cat contacts.txt
白展堂 男 baizhantang@163.com
佟湘玉 女 tongxiangyu@163.com
吕轻侯 男 lvqinghou@126.com
郭芙蓉 女 guofurong@126.com
李秀莲 男 lixiulian@163.com
祝无双 女 zhuwushuang@163.comLenovo@LAPTOP-8MNNHDHQ MINGW64 /e/研究生上课资料/研
一下 上课资料/金融编程与计算/week04 (main)
$ cat contacts.txt environment.yml
白展堂 男 baizhantang@163.com
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.comname: week04
channels:
 - conda-forge
 dependencies:
 - python=3.12
Lenovo@LAPTOP-8MNNHDHQ MINGW64 /e/研究生上课资料/研一下 上课资料/金融编程与计算/
week04 (main)
```

四、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

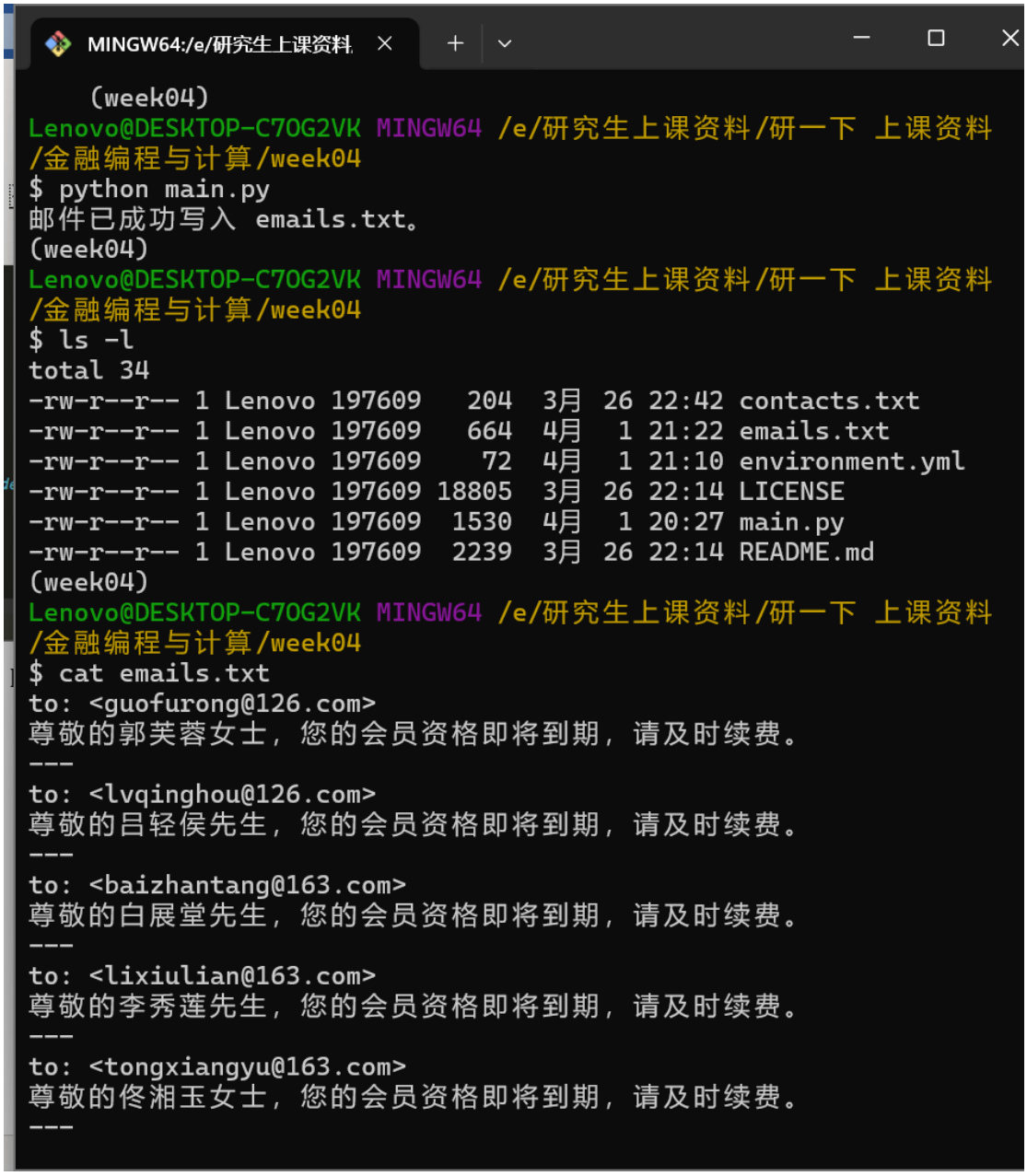
4.1 将大模型提供的代码复制粘贴进 main.py 文件




**4.2 在 VS Code 扩展商店里安装 Python 扩展，使得在编写 main.py 文件时能够显示和选择 Python 解释器**




**4.3 运行 python main.py 命令**

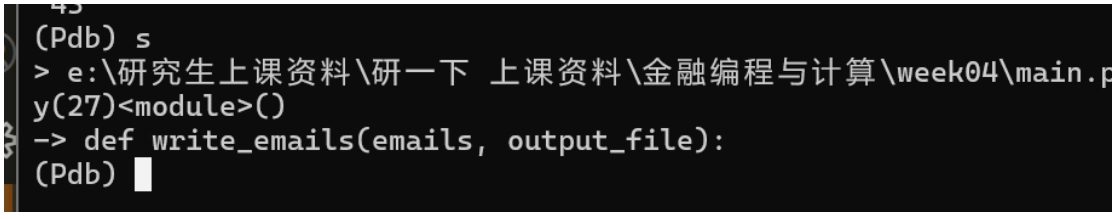



### 4.4 运行 python -m pdb main.py 命令

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。

```
$ python -m pdb main.py
> e:\研究生上课资料\研一下 上课资料\金融编程与计算\week04\main.p
y(1)<module>()
-> def read_contacts(file_path):
(Pdb) l
  1  -> def read_contacts(file_path):
  2         try:
  3             with open(file_path, 'r', encoding='utf-8') as f
ile:
  4                 contacts = []
  5                 for line in file:
  6                     name, gender, email = line.strip().split
()
  7                     contacts.append((name, gender, email))
  8             return contacts
  9         except FileNotFoundError:
 10             print(f"错误：文件 {file_path} 未找到。")
 11             return []
(Pdb) n
> e:\研究生上课资料\研一下 上课资料\金融编程与计算\week04\main.p
y(17)<module>()
-> def generate_emails(contacts):
(Pdb)
```


MINGW64:/e/研究生上课资料
(Pdb) p read_contacts
<function read_contacts at 0x00000258D6563880>
(Pdb) ll
1 def read_contacts(file_path):
2 try:
3 with open(file_path, 'r', encoding='utf-8') as f
ile:
4 contacts = []
5 for line in file:
6 name, gender, email = line.strip().split
()
7 contacts.append((name, gender, email))
8 return contacts
9 except FileNotFoundError:
10 print(f"错误：文件 {file_path} 未找到。")
11 return []
12 except Exception as e:
13 print(f"错误：读取文件时发生未知错误：{e}")
14 return []
15
16
17 -> def generate_emails(contacts):
18 sorted_contacts = sorted(contacts, key=lambda x: (x[
2].split('@')[1], x[2].split('@')[0]))
19 emails = []
20 for name, gender, email in sorted_contacts:
21 title = "先生" if gender == "男" else "女士"
22 email_content = f"to: <{email}>\n尊敬的{name}{ti
tle}，您的会员资格即将到期，请及时续费。\n---"
23 emails.append(email_content)
24 return "\n".join(emails)
25
26
27 def write_emails(emails, output_file):
28 try:

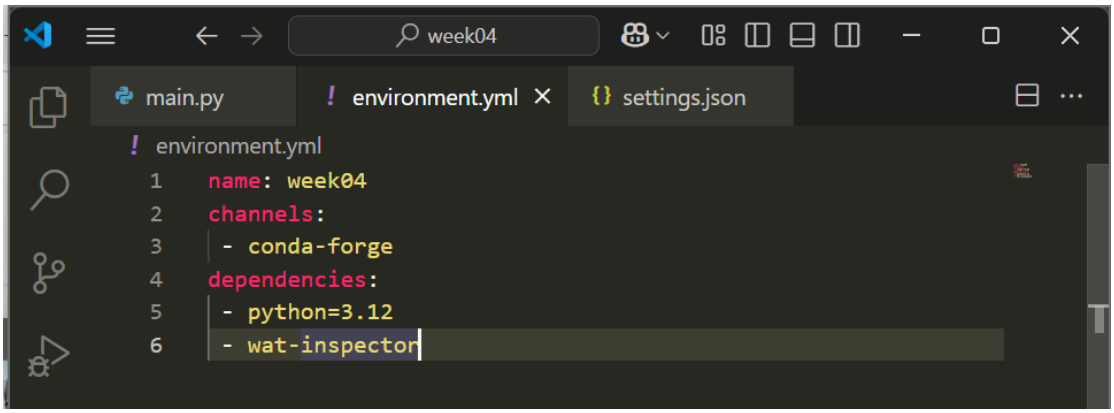

(Pdb) s
> e:\研究生上课资料\研一下 上课资料\金融编程与计算\week04\main.p
y(27)<module>()
-> def write_emails(emails, output_file):
(Pdb)

```
(Pdb) pp contacts
[('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')]
(Pdb)
```

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

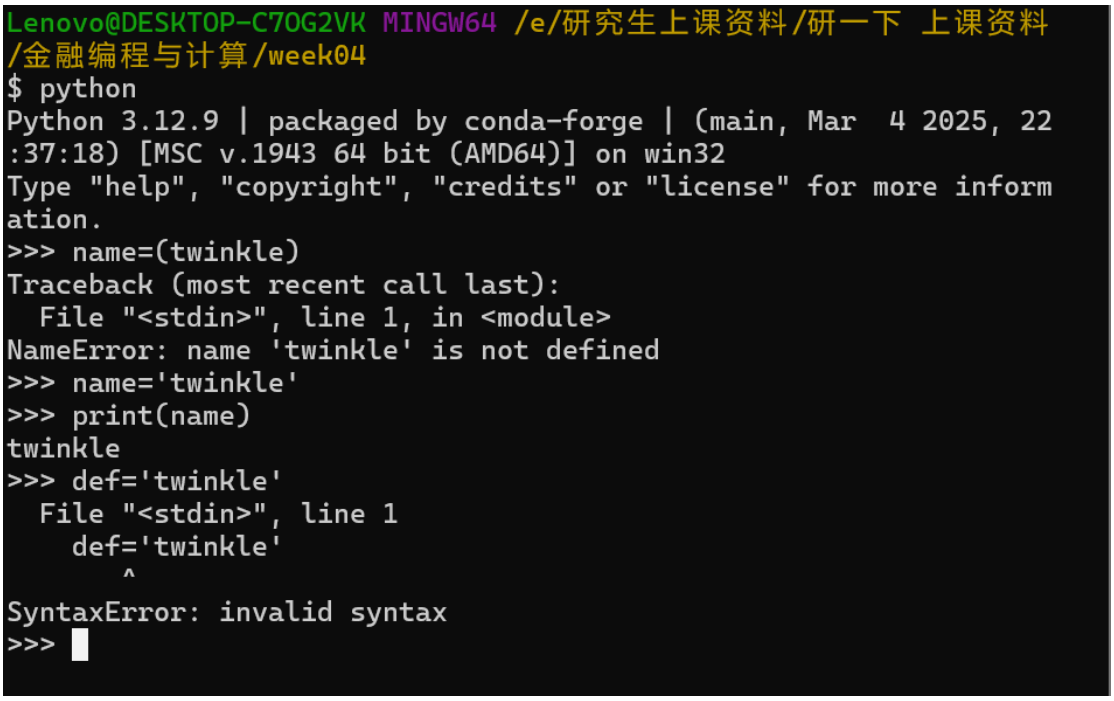
```
(Pdb) c
The program finished and will be restarted
> e:\研究生上课资料\研一下 上课资料\金融编程与计算\week04\main.p
y(1)<module>()
-> def read_contacts(file_path):
(Pdb)
```

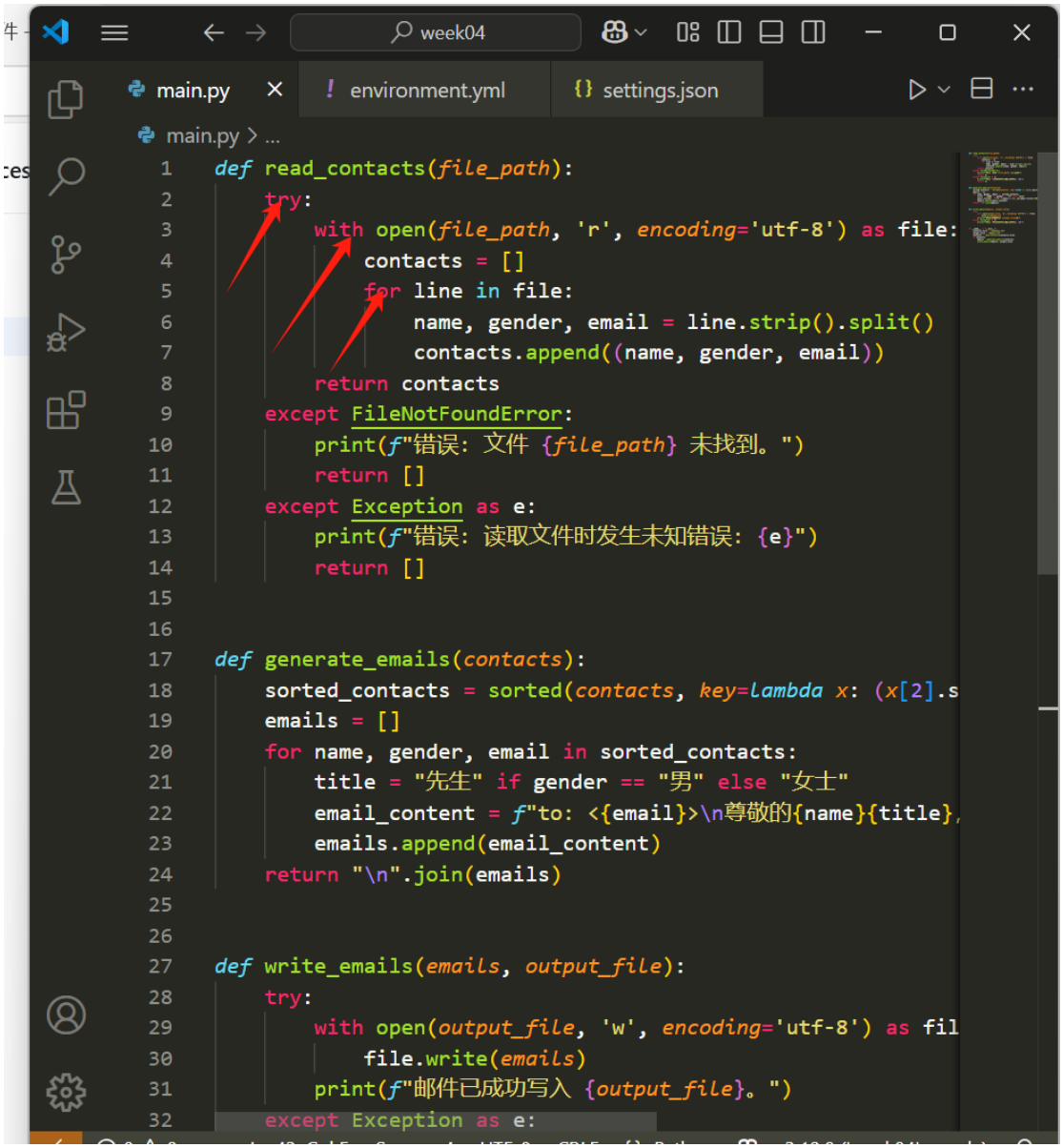
**4.5 利用 wat-inspector (第三方软件包，需要安装) 检查 (inspect)各种对象**




## 五、Python 基本概念

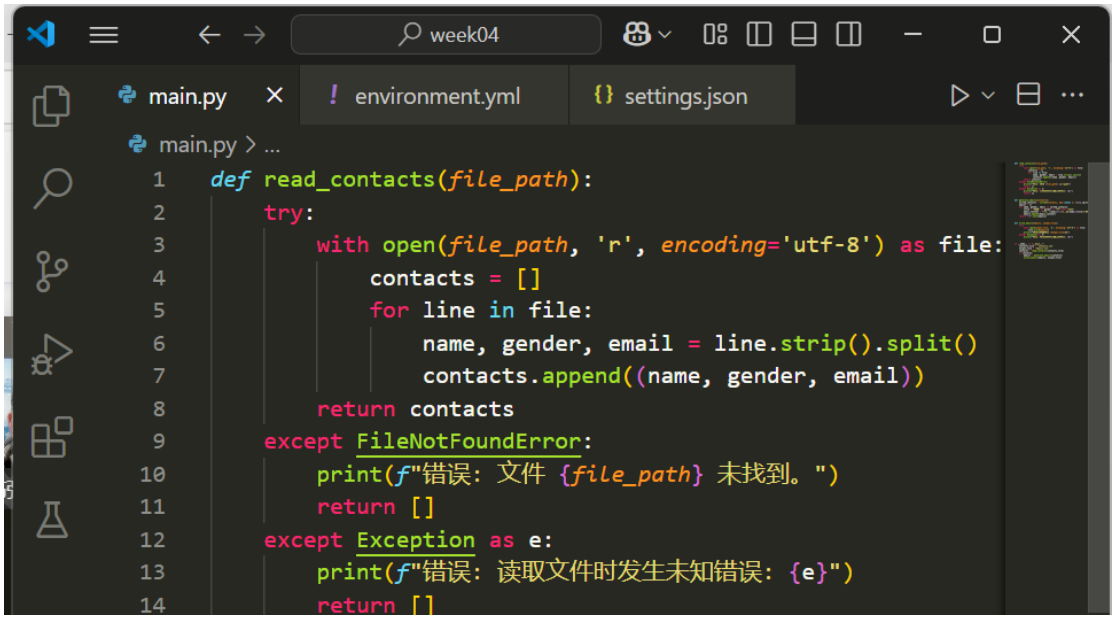
### 5.1 Python 语法保留字 (reserved key words)


Lenovo@DESKTOP-C7OG2VK MINGW64 /e/研究生上课资料/研一下 上课资料/金融编程与计算/week04
$ python
Python 3.12.9 | packaged by conda-forge | (main, Mar 4 2025, 22:37:18) [MSC v.1943 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> name=(twinkle)
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
NameError: name 'twinkle' is not defined
>>> name='twinkle'
>>> print(name)
twinkle
>>> def='twinkle'
File "<stdin>", line 1
def='twinkle'
^
SyntaxError: invalid syntax
>>>

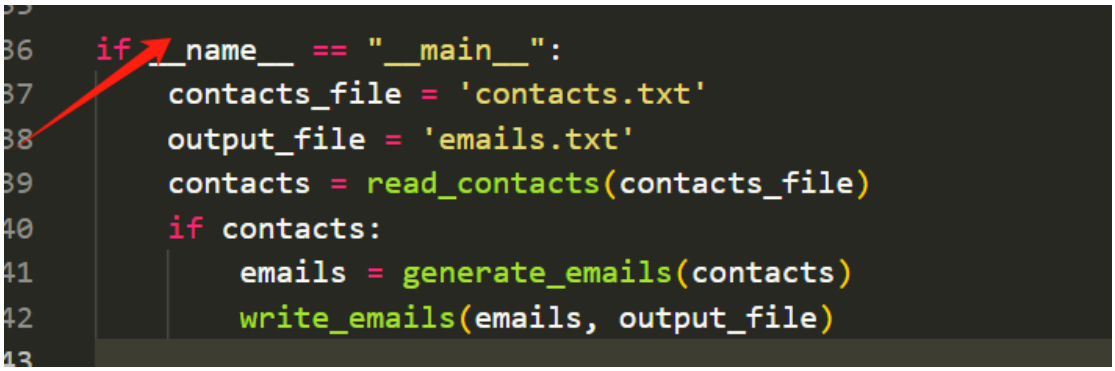



上图红色字体即为保留字。

### 5.2 语句（statement）和表达式（expression）

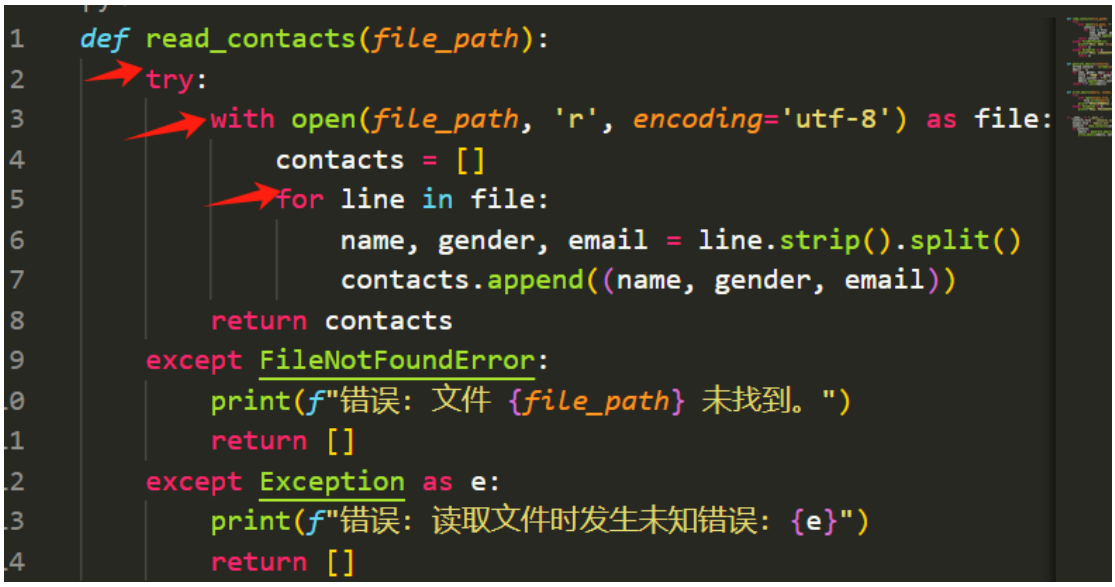



上图 1-14 行为一个语句，语句包含表达式，语句包含子语句。




上图 if 后面就是一个表达式。

### 5.3 缩进 (indent)




通过缩进明确子语句在哪里。

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

```
Local variables:
  __builtins__: dict = {…
  __file__: pdb._ScriptTarget = 'E:\研究生上课资料\研一下 上课资
料\金融编程与计算\week04\main.py'
  __name__: str = '__main__'
  __pdb_convenience_variables: dict = {…
  __spec__: NoneType = None
  contacts_file: str = 'contacts.txt'
  generate_emails: function = <function generate_emails at 0x000
001EDE22F36A0>
  read_contacts: function = <function read_contacts at 0x000001E
DE22F3600>
  wat: wat.inspection.inspection.Wat = <WAT Inspector object>
  write_emails: function = <function write_emails at 0x000001EDE
230E160>
(Pdb)
```

```
(Pdb) wat.globals
Global variables:
  __builtins__: dict = {…
  __file__: pdb._ScriptTarget = 'E:\研究生上课资料\研一下 上课资
料\金融编程与计算\week04\main.py'
  __name__: str = '__main__'
  __pdb_convenience_variables: dict = {…
  __spec__: NoneType = None
  contacts: list = […
  contacts_file: str = 'contacts.txt'
  generate_emails: function = <function generate_emails at 0x000
001EDE22F36A0>
  output_file: str = 'emails.txt'
  read_contacts: function = <function read_contacts at 0x000001E
DE22F3600>
  wat: wat.inspection.inspection.Wat = <WAT Inspector object>
  write_emails: function = <function write_emails at 0x000001EDE
230E160>
(Pdb)
```

在 Python 里，变量的查找遵循 LEGB 规则。该规则规定了 Python 解释器在查找变量时所采用的顺序。“LEGB” 分别代表局部作用域（Local）、闭包作用域（Enclosing）、全局作用域（Global）和内置作用域（Built-in）。

### 5.5 函数 (function) 的定义 (define) 和调用 (call)

```
def generate_emails(contacts):
    sorted_contacts = sorted(contacts, key=lambda x: (x[2].s
```

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

```
(Pdb) p {'a': 1, 'b': 2}
{'a': 1, 'b': 2}
(Pdb)
```

### 5.7 运算符（operator)

```
    for name, gender, email in sorted_contacts:
        title = "先生" if gender == "男" else "女士"
        email_content = f"to: <{email}>\n尊敬的{name}{title},
        emails.append(email_content)
    return "\n".join(emails)
```

### 5.8 形参（parameter)、实参（argument)、返回值（return value)

```
def generate_emails(contacts):
    sorted_contacts = sorted(contacts, key=lambda x: (x[2].s
    emails = []
    for name, gender, email in sorted_contacts:
        title = "先生" if gender == "男" else "女士"
        email_content = f"to: <{email}>\n尊敬的{name}{title},
        emails.append(email_content)
    return "\n".join(emails)


def write_emails(emails, output_file):
    try:
        with open(output_file, 'w', encoding='utf-8') as fil
            file.write(emails)
        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}")


if __name__ == "__main__":
    contacts_file = 'contacts.txt'
    output_file = 'emails.txt'
    contacts = read_contacts(contacts_file)
```

形参

实参

```python
def generate_emails(contacts):
    sorted_contacts = sorted(contacts, key=lambda x: (x[2].s
    emails = []
    for name, gender, email in sorted_contacts:
        title = "先生" if gender == "男" else "女士"
        email_content = f"to: <{email}>\n尊敬的{name}{title},
        emails.append(email_content)
    return "\n".join(emails)
```

**5.9 对象（object）、类型（type）、属性（attribute）、方法（method）**

```
(Pdb) wat / contacts
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 fr
om the iterable.
  def index(value, start=0, stop=9223372036854775807, /) # Retur
n first index of value.…
  def insert(index, object, /) # Insert object before index.
  def pop(index=-1, /) # Remove and return item at index (defaul
t last).…
  def remove(value, /) # Remove first occurrence of value.…
  def reverse() # Reverse *IN PLACE*.
  def sort(*, key=None, reverse=False) # Sort the list in ascend
ing order and return None.…

(Pdb)
```