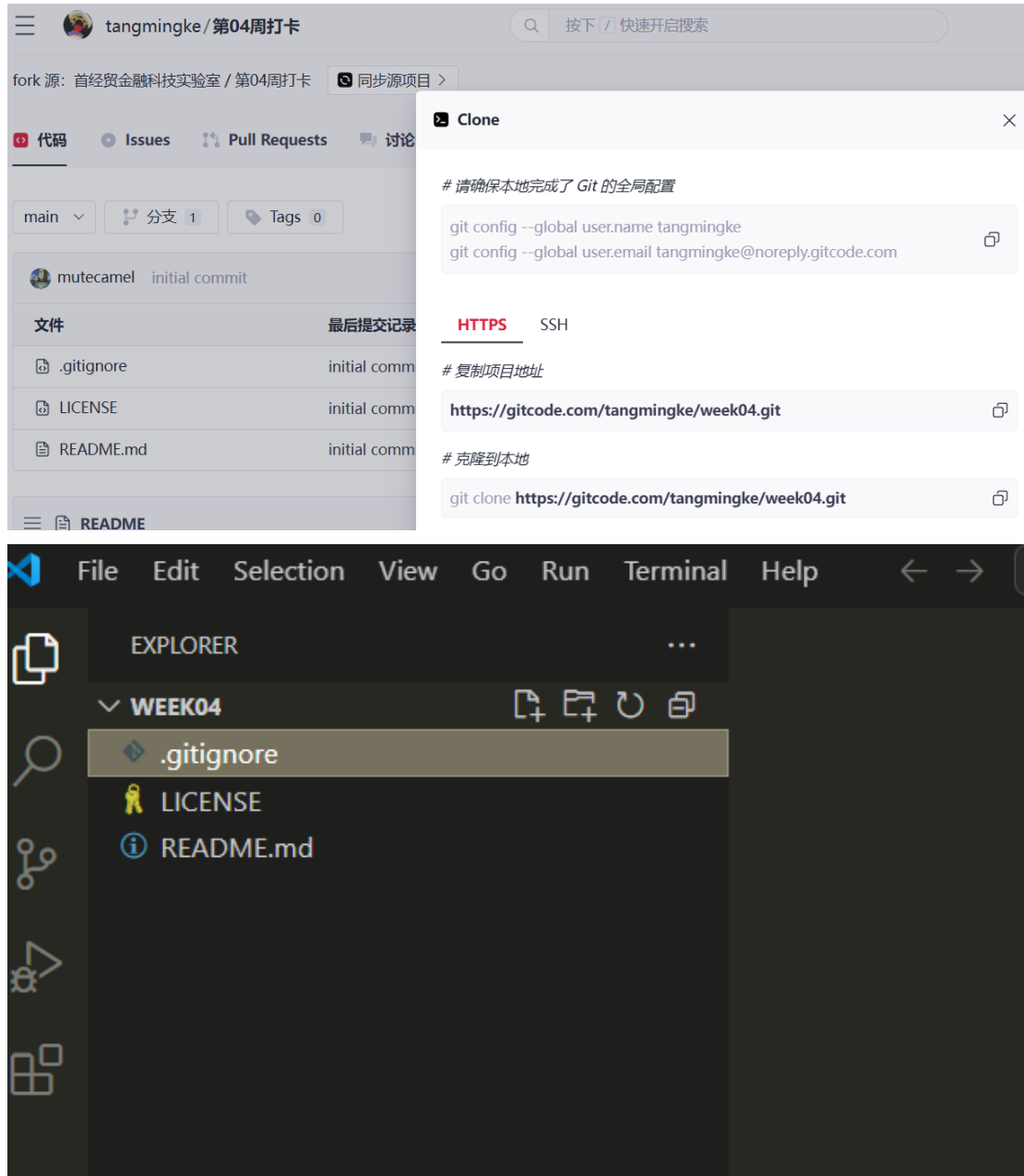
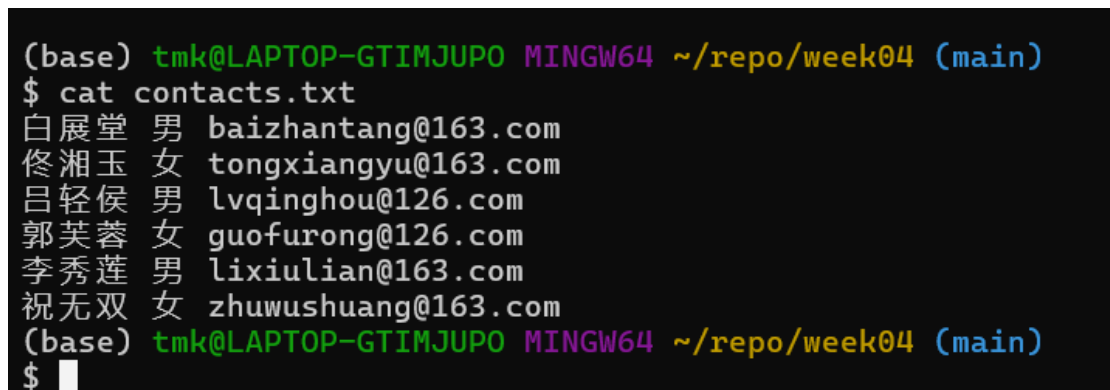


第四周学习报告

● 克隆



● 读取内容



- 拼接

```
(base) tmk@LAPTOP-GTIMJUPO MINGW64 ~/repo/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
(base) tmk@LAPTOP-GTIMJUPO MINGW64 ~/repo/week04 (main)
$
```

- 运行代码

```
17 emails = generate_emails(contacts)
18 write_emails(emails, 'emails.txt')
19 print("邮件已成功生成并保存到 emails.txt 文件中。")

(base) tmk@LAPTOP-GTIMJUPO MINGW64 ~/repo/week04 (main)
$ python main.py
邮件已成功生成并保存到 emails.txt 文件中。
```

```
(base) tmk@LAPTOP-GTIMJUPO 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.com>
尊敬的祝无双女士，您的会员资格即将到期，请及时续费。
---
```

- 调试 PBD

```

(Pdb)
> c:\users\tmk\repo\week04\main.py(33)<module>()
-> contacts = read_contacts('contacts.txt')
(Pdb) l
28         except Exception as e:
29             print(f"错误：写入文件时出现问题：{e}")
30
31
32     if __name__ == "__main__":
33 ->         contacts = read_contacts('contacts.txt')
34         if contacts:
35             emails = generate_emails(contacts)
36             write_emails(emails, 'emails.txt')
37             print("邮件已成功生成并保存到 emails.txt 文件中")
38
(Pdb) n
> c:\users\tmk\repo\week04\main.py(34)<module>()
-> if contacts:
(Pdb) s
> c:\users\tmk\repo\week04\main.py(35)<module>()
-> emails = generate_emails(contacts)
(Pdb) s
--Call--
> c:\users\tmk\repo\week04\main.py(14)generate_emails()
-> def generate_emails(contacts):
(Pdb) n
> c:\users\tmk\repo\week04\main.py(15)generate_emails()
-> emails = []
(Pdb) n
-> 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)
> c:\users\tmk\repo\week04\main.py(19)generate_emails()
-> emails.append(email_content)
(Pdb)
> c:\users\tmk\repo\week04\main.py(16)generate_emails()
-> for name, gender, email in sorted(contacts, key=lambda x: (x[2].split('@')[1], x[2].split('@')[0]]):
(Pdb)
> c:\users\tmk\repo\week04\main.py(17)generate_emails()
-> title = "先生" if gender == "男" else "女士"
(Pdb)
> c:\users\tmk\repo\week04\main.py(18)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)
> c:\users\tmk\repo\week04\main.py(19)generate_emails()
-> emails.append(email_content)
(Pdb)
> c:\users\tmk\repo\week04\main.py(16)generate_emails()
-> for name, gender, email in sorted(contacts, key=lambda x: (x[2].split('@')[1], x[2].split('@')[0]]):
(Pdb)
> c:\users\tmk\repo\week04\main.py(17)generate_emails()
-> title = "先生" if gender == "男" else "女士"
(Pdb)
> c:\users\tmk\repo\week04\main.py(18)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)
> c:\users\tmk\repo\week04\main.py(19)generate_emails()
-> emails.append(email_content)
(Pdb)

```

```

(Pdb)
--Return--
> c:\users\tmk\repo\week04\main.py(20)generate_emails()->['to: <guofuro...期, 请及时续费。\\n---', 'to: <lvqingh...期, 请
及时续费。\\n---', 'to: <baizhan...期, 请及时续费。\\n---', 'to: <lixili...期, 请及时续费。\\n---', 'to: <tongxia...期, 请
及时续费。\\n---', 'to: <zhuwush...期, 请及时续费。\\n---']
-> return emails
(Pdb)
> c:\users\tmk\repo\week04\main.py(36)<module>()
-> write_emails(emails, 'emails.txt')
(Pdb) r
邮件已成功生成并保存到 emails.txt 文件中。
--Return--
> c:\users\tmk\repo\week04\main.py(37)<module>()->None
-> print("邮件已成功生成并保存到 emails.txt 文件中。")
(Pdb) n
--Return--
> <string>(1)<module>()->None
(Pdb) r
> d:\ana2\lib\bdb.py(610)run()
-> self.quitting = True
(Pdb) r
The program finished and will be restarted
> c:\users\tmk\repo\week04\main.py(1)<module>()
-> def read_contacts(file_path):
(Pdb) r
邮件已成功生成并保存到 emails.txt 文件中。
--Return--
> c:\users\tmk\repo\week04\main.py(37)<module>()->None
an@163.com'), ('祝无双', '女', 'zhuwushuang@163.com')]
-> with open(file_path, 'r', encoding='utf-8') as file:
(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', 'lixili@163.com'),
 ('祝无双', '女', 'zhuwushuang@163.com')]
(Pdb) n
> c:\users\tmk\repo\week04\main.py(34)<module>()
-> if contacts:
(Pdb) n
> c:\users\tmk\repo\week04\main.py(35)<module>()
-> emails = generate_emails(contacts)
(Pdb) n
> c:\users\tmk\repo\week04\main.py(36)<module>()
-> write_emails(emails, 'emails.txt')
(Pdb) n
> c:\users\tmk\repo\week04\main.py(37)<module>()
-> print("邮件已成功生成并保存到 emails.txt 文件中。")
(Pdb) n
邮件已成功生成并保存到 emails.txt 文件中。
--Return--
> c:\users\tmk\repo\week04\main.py(37)<module>()->None
-> print("邮件已成功生成并保存到 emails.txt 文件中。")
(Pdb)
--Return--
> <string>(1)<module>()->None
(Pdb)

```

- 下载 wat-inspector

```
done
#
# To activate this environment, use
#
#     $ conda activate week04
#
# To deactivate an active environment, use
#
#     $ conda deactivate
```

● Python 基本概念笔记

1. Python 语法保留字 (reserved key words)

Python 保留字是指在 Python 语言中有特定含义的单词，不能将它们用作变量名、函数名或其他标识符。你可以使用 `keyword` 模块来查看所有的保留字：

```
python ^
import keyword
print(keyword.kwlist)
```

常见的保留字有 `if`、`else`、`for`、`while`、`def`、`class` 等。

2. 语句 (statement) 和表达式 (expression)

- **语句**：是 Python 程序中执行操作的基本单位，通常以换行符结束。语句可以是赋值语句、条件语句、循环语句等。例如：

```
python ^
x = 5 # 赋值语句
if x > 3:
    print("x 大于 3") # 条件语句
```

- **表达式**：是由值、变量、运算符和函数调用组成的组合，它会计算出一个值。例如：

```
python ^
y = 2 + 3 # 2 + 3 是一个表达式，计算结果为 5
```

3. 缩进 (indent)

Python 使用缩进来表示代码块，而不是像其他语言那样使用大括号 `{}`。通常使用 4 个空格或一个制表符作为缩进。例如：

```
python ^
if True:
    print("这是一个代码块")
    print("缩进相同的语句属于同一个代码块")
```

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

- **局部变量**：在函数内部定义的变量，只能在函数内访问。例如：

```
python ^
def my_function():
    local_var = 10 # 局部变量
    print(local_var)

my_function()
```

- **全局变量**：在函数外部定义的变量，可以在整个程序中访问。例如：

```
python ^

global_var = 20 # 全局变量

def my_function():
    print(global_var)

my_function()
```

- **LEGB 规则**：是 Python 查找变量的顺序，即 Local（局部作用域）-> Enclosing（闭包作用域）-> Global（全局作用域）-> Built-in（内置作用域）。Python 按照这个顺序依次查找变量，如果在某个作用域中找到了变量，就会使用该变量的值。

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

- **定义函数**：使用 def 关键字来定义函数，后面跟着函数名和参数列表。例如：

```
python ^

def add_numbers(a, b):
    return a + b
```

- **调用函数**：通过函数名和参数列表来调用函数。例如：

```
python ^

result = add_numbers(3, 5)
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。例如：'hello' 或 "world"。
- **整数**：表示整数的数值。例如：10、-20。
- **列表**：用方括号 [] 括起来的有序元素集合。例如：[1, 2, 3]。

- **列表**: 用方括号 `[]` 括起来的有序元素集合。例如: `[1, 2, 3]`。
- **字典**: 用花括号 `{}` 括起来的键值对集合。例如: `{'name': 'John', 'age': 30}`。
- **元组**: 用圆括号 `()` 括起来的有序元素集合, 元组是不可变的。例如: `(1, 2, 3)`。

7. 运算符 (operator)

Python 提供了多种运算符, 包括算术运算符 (如 `+`、`-`、`*`、`/`)、比较运算符 (如 `==`、`!=`、`>`、`<`)、逻辑运算符 (如 `and`、`or`、`not`) 等。例如:

```
python ^  
  
x = 5  
y = 3  
print(x + y) # 算术运算符  
print(x > y) # 比较运算符  
print(x > 2 and y < 4) # 逻辑运算符
```

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

- **形参**: 在函数定义时指定的参数, 用于接收传递给函数的值。例如:

```
python ^  
  
def my_function(param1, param2):  
    return param1 + param2
```

这里的 `param1` 和 `param2` 就是形参。

- **实参**: 在函数调用时传递给函数的实际值。例如:

```
python ^  
  
result = my_function(3, 5)
```

这里的 `3` 和 `5` 就是实参。

- **返回值**: 函数执行完毕后返回给调用者的值, 使用 `return` 语句返回。例如, 上面的 `my_function` 函数返回两个参数的和。
- **方法**: 对象可以执行的操作, 是与对象关联的函数。例如, 列表对象有 `append()` 方法用于在列表末尾添加元素。

```
python ^  
  
my_list = [1, 2, 3]  
my_list.append(4) # 调用 append() 方法  
print(my_list)
```

- **方法**：对象可以执行的操作，是与对象关联的函数。例如，列表对象有 `append()` 方法用于在列表末尾添加元素。

```
python ^  
  
my_list = [1, 2, 3]  
my_list.append(4) # 调用 append() 方法  
print(my_list)
```

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

- **对象**：Python 中的一切都是对象，每个对象都有自己的类型和属性。例如：

```
python ^  
  
x = 5 # x 是一个整数对象
```

- **类型**：对象的类别，决定了对象可以执行的操作。可以使用 `type()` 函数查看对象的类型。例如：

```
python ^  
  
print(type(x)) # 输出 <class 'int'>
```

- **属性**：对象的特性或数据。例如，列表对象有 `length` 属性表示列表的长度。
- **方法**：对象可以执行的操作，是与对象关联的函数。例如，列表对象有 `append()` 方法用于在列表末尾添加元素。

