

## 第六周 学习笔记——Python 代码组织（初级）

一、安装 Fork 第 06 周打卡仓库至自己的名下，然后将名下的这个仓库 Clone 到本地计算机

```
(base)
Lenovo@DESKTOP-C70G2VK 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 git@gitcode.com:twinkledahaha/week06.git
Cloning into 'week06'...
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 | 8.45 MiB/s, done.
(base)
Lenovo@DESKTOP-C70G2VK MINGW64 /e/研究生上课资料/研一下 上课资料/金融编程与计算
$
```

二、安装用 VS Code 打开项目目录，新建 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

```
$ cd week06
(base)
Lenovo@DESKTOP-C70G2VK MINGW64 /e/研究生上课资料/研一下 上课资料/金融编程与计算/week06 (main)
$ ls
environment.yml  LICENSE  README.md
(base)
Lenovo@DESKTOP-C70G2VK MINGW64 /e/研究生上课资料/研一下 上课资料/金融编程与计算/week06 (main)
$ cat environment.yml
(base)
Lenovo@DESKTOP-C70G2VK MINGW64 /e/研究生上课资料/研一下 上课资料/金融编程与计算/week06 (main)
$ cat environment.yml
name: week06
channels:
  - conda-forge
done
#
# To activate this environment, use
#
#     $ conda activate week06
#
# To deactivate an active environment, use
#
#     $ conda deactivate
#
(base)
Lenovo@DESKTOP-C70G2VK MINGW64 /e/研究生上课资料/研一下 上课资料/金融编程与计算/week06 (main)
$
```

```

Lenovo@DESKTOP-C70G2VK MINGW64 /e/研究生上课资料/研一下 上课资料
/金融编程与计算/week06 (main)
$ 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
week05 D:\Anaconda\Anaconda3\envs\week05
week06 D:\Anaconda\Anaconda3\envs\week06

(base)

```

三、创建一个 `guessing_game.py` 文件，复制粘贴以下代码，运用 `pdb` 调试器理解其运行流程

```

/金融编程与计算/week06 (main)
$ python -m pdb guessing_game.py
> e:\研究生上课资料\研一下 上课资料\金融编程与计算\week06\guessi
ng_game.py(1)<module>()
-> import random
(Pdb) l
1  -> import random
2
3
4     def guessing_game():
5         # 生成 1 到 100 之间的随机整数
6         secret_number = random.randint(1, 100)
7         n = 0
8
9         print("欢迎来到猜数字游戏！我已经想好了一个 1 到 100
之间的数字，你可以开始猜啦。")
10
11        while True:
(Pdb) n
> e:\研究生上课资料\研一下 上课资料\金融编程与计算\week06\guessi
ng_game.py(4)<module>()
-> def guessing_game():
(Pdb) l
1     import random
2
3
4  -> def guessing_game():
5         # 生成 1 到 100 之间的随机整数
6         secret_number = random.randint(1, 100)
7         n = 0
8
9         print("欢迎来到猜数字游戏！我已经想好了一个 1 到 100
之间的数字，你可以开始猜啦。")
10
11        while True:

```

```

(Pdb) n
> e:\研究生上课资料\研一下 上课资料\金融编程与计算\week06\guessi
ng_game.py(49)<module>()
-> if __name__ == "__main__":
(Pdb) l
44         raise NotImplementedError
45
46         print("游戏结束, 再见 🙋。")
47
48
49 -> if __name__ == "__main__":
50     guessing_game()
[EOF]
(Pdb) n
> e:\研究生上课资料\研一下 上课资料\金融编程与计算\week06\guessi
ng_game.py(50)<module>()
-> guessing_game()
(Pdb) l
45
46     print("游戏结束, 再见 🙋。")
47
48
49     if __name__ == "__main__":
50 ->     guessing_game()
[EOF]
(Pdb) p __name__
'__main__'
(Pdb) s
--Call--
> e:\研究生上课资料\研一下 上课资料\金融编程与计算\week06\guessi
ng_game.py(4)guessing_game()
-> def guessing_game():
(Pdb) l

```

```

guessing_game()
(Pdb) l
45
46     print("游戏结束，再见 🙋。")
47
48
49     if __name__ == "__main__":
50 ->     guessing_game()
[EOF]
(Pdb) p __name__
'__main__'
(Pdb) s
--Call--
> e:\研究生上课资料\研一下 上课资料\金融编程与计算\week06\guessi
ng_game.py(4)guessing_game()
-> def guessing_game():
(Pdb) l
1     import random
2
3
4 -> def guessing_game():
5     # 生成 1 到 100 之间的随机整数
6     secret_number = random.randint(1, 100)
7     n = 0
8
9     print("欢迎来到猜数字游戏！我已经想好了一个 1 到 100
之间的数字，你可以开始猜啦。")
10
11     while True:
(Pdb) n
> e:\研究生上课资料\研一下 上课资料\金融编程与计算\week06\guessi
ng_game.py(6)guessing_game()
-> secret_number = random.randint(1, 100)
(Pdb) l
1     import random

```

### 3.1 调用 `p random.randint(1, 100)` 函数

```

(Pdb) l
1      import random
2
3
4      def guessing_game():
5          # 生成 1 到 100 之间的随机整数
6      ->      secret_number = random.randint(1, 100)
7              n = 0
8
9              print("欢迎来到猜数字游戏！我已经想好了一个 1 到 100
之间的数字，你可以开始猜啦。")
10
11          while True:
(Pdb) p random.randint(1, 100)
57
(Pdb)
91
(Pdb)
13
(Pdb)
24
(Pdb)

```

### 3.2 调用 p secret\_number 函数

```

(Pdb) n
> e:\研究生上课资料\研一下 上课资料\金融编程与计算\week06\guessi
ng_game.py(7)guessing_game()
-> n = 0
(Pdb) l
2
3
4      def guessing_game():
5          # 生成 1 到 100 之间的随机整数
6          secret_number = random.randint(1, 100)
7      ->      n = 0
8
9          print("欢迎来到猜数字游戏！我已经想好了一个 1 到 100
之间的数字，你可以开始猜啦。")
10
11          while True:
12              n += 1
(Pdb) p secret_number
88
(Pdb)

```

### 3.3 调用 p guess 函数

```
(Pdb) l
12         n += 1
13         # 获取玩家输入
14         guess = input(
15             f"(第 {n} 次尝试) 请输入你猜的数字 (输入整数
, 或者输入 q 回车退出): "
16         )
17 ->         guess = guess.strip() # 去除多余空白字符
18
19         if guess == "q":
20             break
21
22         try:
(Pdb) p guess
'35'
(Pdb) p type(guess)
<class 'str'>
(Pdb) p 'abd '.strip()
'abd'
(Pdb) p 'a  bc'.strip()
'a  bc'
(Pdb) p '   abc'.strip()
'abc'
(Pdb)
```

```
(Pdb) import wat
(Pdb) wat / str.strip

value: <method 'strip' of 'str' objects>
type: method_descriptor
signature: def strip(self, chars=None, /)
"""
Return a copy of the string with leading and trailing whitespace
removed.

If chars is given and not None, remove characters in chars inste
ad.
"""
```

```

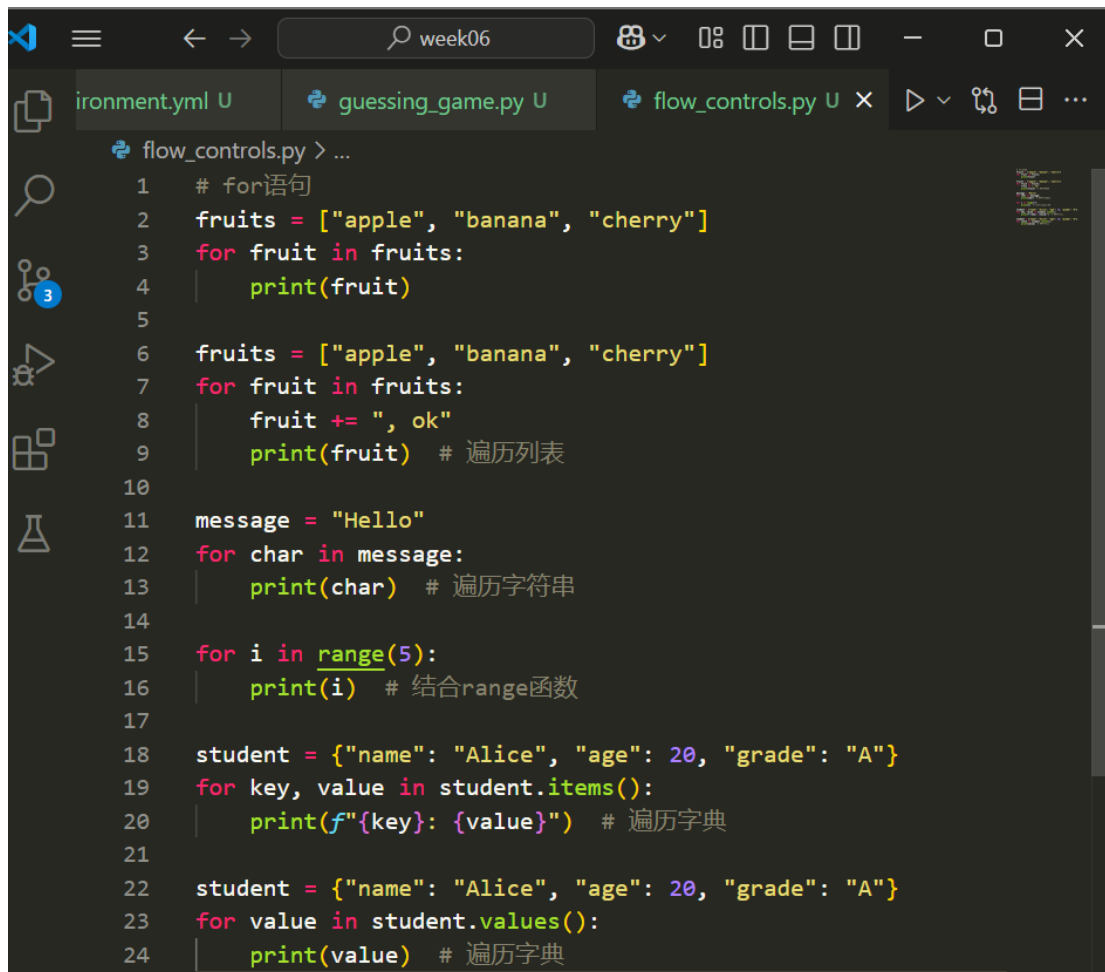
Lenovo@DESKTOP-C70G2VK MINGW64 /e/研究生上课资料/研一下 上课资料
/金融编程与计算/week06 (main)
$ python guessing_game.py
欢迎来到猜数字游戏！我已经想好了一个 1 到 100 之间的数字，你可以
开始猜啦。
(第 1 次尝试) 请输入你猜的数字 (输入整数，或者输入 q 回车退出):4
5
猜的数字太大了，再试试~。
(第 2 次尝试) 请输入你猜的数字 (输入整数，或者输入 q 回车退出):3
0
猜的数字太大了，再试试~。
(第 3 次尝试) 请输入你猜的数字 (输入整数，或者输入 q 回车退出):2
0
猜的数字太大了，再试试~。
(第 4 次尝试) 请输入你猜的数字 (输入整数，或者输入 q 回车退出):1
5
猜的数字太大了，再试试~。
(第 5 次尝试) 请输入你猜的数字 (输入整数，或者输入 q 回车退出):1
0
猜的数字太大了，再试试~。
(第 6 次尝试) 请输入你猜的数字 (输入整数，或者输入 q 回车退出):6
猜的数字太大了，再试试~。
(第 7 次尝试) 请输入你猜的数字 (输入整数，或者输入 q 回车退出):3
猜的数字太大了，再试试~。
(第 8 次尝试) 请输入你猜的数字 (输入整数，或者输入 q 回车退出):2
恭喜你 🎉，猜对了！
游戏结束，再见 🙋。
(week06)

```

写程序循环时，我们要先处理一些特殊的情况！

四、创建一个 `flow_controls.py` 文件，运行以下 Python 流程  
控制语句

#### 4.1 for 迭代循环 (iteration loop)



```
1  # for语句
2  fruits = ["apple", "banana", "cherry"]
3  for fruit in fruits:
4      print(fruit)
5
6  fruits = ["apple", "banana", "cherry"]
7  for fruit in fruits:
8      fruit += ", ok"
9      print(fruit) # 遍历列表
10
11 message = "Hello"
12 for char in message:
13     print(char) # 遍历字符串
14
15 for i in range(5):
16     print(i) # 结合range函数
17
18 student = {"name": "Alice", "age": 20, "grade": "A"}
19 for key, value in student.items():
20     print(f"{key}: {value}") # 遍历字典
21
22 student = {"name": "Alice", "age": 20, "grade": "A"}
23 for value in student.values():
24     print(value) # 遍历字典
```



```

Lenovo@DESKTOP-C70G2VK MINGW64 /e/研究生上课资料/研一下 上课资料
/金融编程与计算/week06 (main)
$ python flow_controls.py
apple
banana
cherry
apple, ok
banana, ok
cherry, ok
H
e
l
l
o
0
1
2
3
4
name: Alice
age: 20
grade: A
Alice
20
A
(week06)

```

## 4.2 while 条件循环 (conditional loop)

```

26 # while语句
27 count = 0
28 while count < 5:
29     print(count)
30     count = count + 1 # 简单计数
31
32 numbers = [1, 2, 3, 4, 5]
33 while numbers:
34     print(numbers.pop()) # 从列表中移除元素
35
36 valid_input = False
37 while not valid_input:
38     user_input = input("请输入一个大于 10 的数字: ")
39     try:
40         num = int(user_input)
41         if num > 10:
42             print("输入有效。")
43             valid_input = True
44         else:
45             print("输入的数字必须大于 10, 请重新输入。")
46     except ValueError:
47         print("输入无效, 请输入一个有效的数字。") # 用户输入验证
48

```

```

0
1
2
3
4
5
4
3
2
1
请输入一个大于 10 的数字: 11
输入有效。
(week06)
Lenovo@DESKTOP-C70G2VK MINGW64 /e/研究生上课资料/研一下 上课资料
/金融编程与计算/week06 (main)

```

### 4.3 break 打断跳出循环

```

49 # break语句
50 fruits = ["apple", "banana", "cherry", "date"]
51 for fruit in fruits:
52     if fruit == "cherry":
53         break
54     print(fruit) # 在 for 循环中使用 break
55
56 count = 0
57 while count < 10:
58     if count == 5:
59         break
60     print(count)
61     count = count + 1 # 在 while 循环中使用 break
62
63 for i in range(3):
64     for j in range(3):
65         if i + j == 2:
66             break
67         print(f"({i}, {j})") # 嵌套循环中使用 break

```

```

apple
banana
0
1
2
3
4
(0, 0)
(0, 1)
(1, 0)
(week06)
Lenovo@DESKTOP-C70G2VK MINGW64 /e/研究生上课资料/研一下 上课资料
/金融编程与计算/week06 (main)

```

#### 4.4 continue 跳至下一轮循环

```
69 # continue语句
70 numbers = [1, 2, 3, 4, 5]
71 for num in numbers:
72     if num == 3:
73         continue
74     print(num) # 在 for 循环中使用 continue
75
76 count = 0
77 while count < 5:
78     count = count + 1
79     if count == 2:
80         continue
81     print(count) # 在 while 循环中使用 continue
82
83 count = 0
84 while count < 5:
85     count = count + 1
86     if count == 2:
87         continue
88     print(count) # 嵌套循环中使用 continue
89
```

```
1  
2  
4  
5  
1  
3  
4  
5  
1  
3  
4  
5  
(week06)  
Lenovo@DESKTOP-C70G2VK MINGW64 /e/研究生上课资料/研一下 上课资料
```

#### 4.5 for...else 循环未被打断的处理

```

90 # for...else语句
91 numbers = [1, 3, 5, 7, 9]
92 for num in numbers:
93     if num % 2 == 0:
94         print(f"找到了偶数: {num}")
95         break
96 else:
97     print("列表中没有偶数。") # 查找列表中的偶数
98
99 text = "Hello, World!"
100 target = "z"
101 for char in text:
102     if char == target:
103         print(f"找到了字符 '{target}'. ")
104         break
105 else:
106     print(f"字符串中没有字符 '{target}'. ") # 检查字符串中是否
107
108 student_scores = {"Alice": 85, "Bob": 90, "Charlie": 78}
109 search_name = "David"
110 for name in student_scores.keys():
111     if name == search_name:
112         print(f"找到了 {search_name}, 分数是 {student_scores[...]}")
113         break
114 else:
115     print(f"没有找到 {search_name} 的记录。") # 遍历字典的键
116

```

```

列表中没有偶数。
字符串中没有字符 'z'。
没有找到 David 的记录。
(week06)

```

```

Lenovo@DESKTOP-C70G2VK MINGW64 /e/研究生上课资料/研一下 上课资料
/金融编程与计算/week06 (main)
#

```

#### 4.6 if 条件分支

```

# if语句
age = 18
if age >= 18:
    print("你已经成年了。") # 简单的 if 语句

num = 10
if num % 2 == 0:
    print(f"{num} 是偶数。")
else:
    print(f"{num} 是奇数。") # if-else 语句

score = 85
if score >= 90:
    print("成绩为 A。")
elif score >= 80:
    print("成绩为 B。")
elif score >= 70:
    print("成绩为 C。")
elif score >= 60:
    print("成绩为 D。")
else:
    print("成绩为 F。") # if-elif-else 语句

is_member = True
money = 200
product_price = 150
if is_member:
    if money >= product_price:
        print("你是会员，且余额足够，可以购买该商品。")
    else:
        print("你是会员，但余额不足，无法购买该商品。")

```

你已经成年了。

10 是偶数。

成绩为 B。

你是会员，且余额足够，可以购买该商品。

(week06)

Lenovo@DESKTOP-C70G2VK MINGW64 /e/研究生上课资料/研一下 上课资料  
/金融编程与计算/week06 (main)

#### 4.7 if...elif[...elif] 多重条件分支

```

154     # if...elif语句
155     month = 7
156     if 3 <= month <= 5:
157         print("当前季节是春季")
158     elif 6 <= month <= 8:
159         print("当前季节是夏季")
160     elif 9 <= month <= 11:
161         print("当前季节是秋季")
162     elif month == 12 or 1 <= month <= 2:
163         print("当前季节是冬季")
164     else:
165         print("输入的月份无效")
166

```

```

当前季节是夏季
(week06)

```

```

Lenovo@DESKTOP-C70G2VK MINGW64 /e/研究生上课资料/研一下 上课资料

```

#### 4.8 if...else 未满足条件的处理

```

167     # if...else语句
168     number = 10
169     if number % 2 == 0:
170         print(f"{number} 是偶数。")
171     else:
172         print(f"{number} 是奇数。")
173

```

```

10 是偶数。
(week06)

```

```

Lenovo@DESKTOP-C70G2VK MINGW64 /e/研究生上课资料/研一下 上课资料
/金融编程与计算/week06 (main)

```

#### 4.9 try...except[...except...else...finally] 捕捉异常的处理

```

174 # try...except语句
175 try:
176     result = 10 / 0
177     print(result)
178 except ZeroDivisionError:
179     print("错误：除数不能为零。")
180
181 # try...except...else...finally语句
182 try:
183     num1 = int(input("请输入第一个整数："))
184     num2 = int(input("请输入第二个整数："))
185     result = num1 / num2
186 except ValueError:
187     print("输入错误：请输入有效的整数。")
188 except ZeroDivisionError:
189     print("错误：除数不能为零。")
190 else:
191     print(f"除法运算结果是：{result}")
192 finally:
193     print("无论是否发生异常，此代码块都会执行。")

```

```

错误：除数不能为零。
请输入第一个整数：3
请输入第二个整数：4
除法运算结果是：0.75
无论是否发生异常，此代码块都会执行。
(week06)

```

#### 4.10 raise 主动抛出异常

```

196 # raise语句
197 def divide_numbers(a, b):
198     if b == 0:
199         raise ZeroDivisionError("除数不能为零。")
200     return a / b
201
202
203 try:
204     result = divide_numbers(10, 0)
205     print(result)
206 except ZeroDivisionError as e:
207     print(f"捕获到异常: {e}")
208

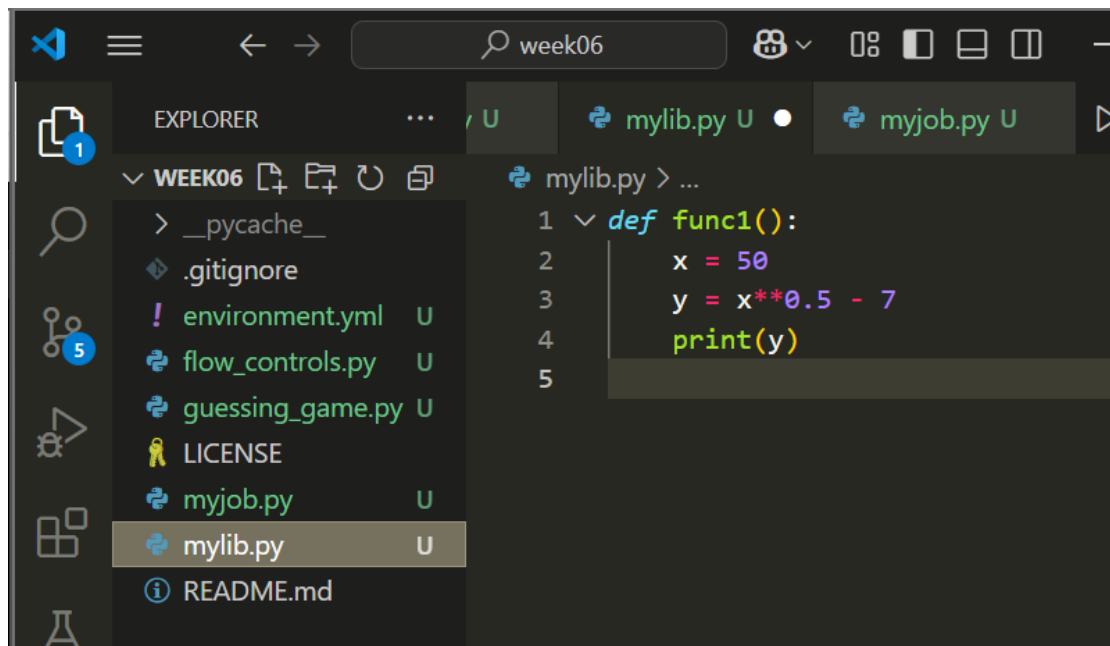
```

捕获到异常: 除数不能为零。

(week06)

Lenovo@DESKTOP-C70G2VK MINGW64 /e/研究生上课资料/研一下 上课资料/金融编程与计算/week06 (main)

## 五、创建 mylib.py 模块 (module) 和 myjob.py 脚本 (script)



### 5.1. 定义函数 func1, 没有形参, 没有返回值



```

1  import mylib # noqa: F401
2
3  y = mylib.func1()
4  print(y)
5

```

```

Lenovo@DESKTOP-C70G2VK MINGW64 /e/研究生上课资料/研一下 上课资料
/金融编程与计算/week06 (main)
$ python myjob.py
0.0710678118654755
None
(main)

```

5.2. 定义函数 func2，没有形参，有返回值

```

7  def func2():
8      x = 70
9      y = x**0.5 - 7
10     print(y)
11     return y
12
13

```

```

1.3666002653407556
1.3666002653407556
(main)
Lenovo@DESKTOP-C70G2VK MINGW64 /e/研究生上课资料/研一下 上课资料
/金融编程与计算/week06 (main)

```

5.3 定义函数 func3，只有一个 位置形参 (positional parameter)，先尝试传入 位置实参 (positional argument) 调用，再尝试传入 命名实参 (named argument) 调用，再尝试不传实参 (会报错)

```

14  def func3(x):
15      y = x**0.5 - 7
16      return y
17

```

```

13
14     y = mylib.func3(45)
15     print(y) # 位置实参
16
17     y = mylib.func3(x=47)
18     print(y) # 命名实参
19
20     try:
21         y = mylib.func3()
22     except TypeError as e:
23         print(e) # 不传实参
24
25     try:
26         mylib.func3(x=47)
27     except TypeError as e:
28         print(e)

```

```

-0.2917960675006306
-0.1443453995989561
func3() missing 1 required positional argument: 'x'
(week06)

```

5.4 定义函数 func4，只有一个 命名形参 (named parameter)，先传入 位置实参 调用，再传入 命名实参 调用，再尝试不传实参 (取默认值)

```

19     def func4(x=50):
20         y = x**0.5 - 7
21         return y
22

```

```

29     y = mylib.func4(48)
30     print(y) # 位置实参
31
32     y = mylib.func4(x=49)
33     print(y) # 命名实参
34
35     y = mylib.func4()
36     print(y) # 不传实参
37

```

```

-0.07179676972449123
0.0
0.0710678118654755
(week06)
Lenovo@DESKTOP-C70G2VK MINGW64 /e/研究生上课资料/研一下 上课资料
/金融编程与计算/week06 (main)

```

5.5 定义函数 func5，接受多个位置形参和命名形参，尝试以位置/命名各种不同方式传入实参，注意位置参数必须排在命名参数之前

```

24     def caculate(a, b, operation='add'):
25         if operation == 'add':
26             return a + b
27         elif operation == 'subtract':
28             return a - b
29         else:
30             return None
31
32

```

```

38     print(mylib.caculate(5, 10, "add"))
39     print(mylib.caculate(operation="add", b=5, a=10))
40     print(mylib.caculate(b=5, a=10))
41     print(mylib.caculate(5, 10, (parameter) operation: str))
42     print(mylib.caculate(5, 8, operation="subtract"))
43

```

```

15
15
15
-5
-3
(week06)
Lenovo@DESKTOP-C70G2VK MINGW64 /e/研究生上课资料/研一下 上课资料
/金融编程与计算/week06 (main)

```

5.6 定义函数 func6，在形参列表中使用 / 来限定只接受位置实参的形参

5.7 定义函数 func7，在形参列表中使用 \* 来限定只接受命名实参的形参

```

32
33 def func6(a, /, b, operation="add"):
34     if operation == "add":
35         return a + b
36     elif operation == "subtract":
37         return a - b
38     else:
39         return None
40
41
42 def func7(a, /, b, *, operation="add"):
43     if operation == "add":
44         return a + b
45     elif operation == "subtract":
46         return a - b
47     else:
48         return None
49

```

```

44 try:
45     print(mylib.func6(a=10, b=5))
46 except TypeError as e:
47     print(e)
48
49 try:
50     print(mylib.func7(10, 5, operation="subtract"))
51 except TypeError as e:
52     print(e)
53

```

```
func6() got some positional-only arguments passed as keyword arguments: 'a'
5
(week06)
Lenovo@DESKTOP-C70G2VK MINGW64 /e/研究生上课资料/研一下 上课资料
/金融编程与计算/week06 (main)
```

5.8 定义函数 func8，在位置形参的最后，在形参名称前使用 \* 允许传入任意数量的位置实参 (被打包为元组)

```
51 def func8(*numbers):
52     total = 0
53     for num in numbers:
54         total = total + num
55     return total
56
```

```
54 print(mylib.func8(4, 8, 18, 5))
55 print(mylib.func8())
56
```

```
35
0
(week06)
Lenovo@DESKTOP-C70G2VK MINGW64 /e/研究生上课资料/研一下 上课资料
/金融编程与计算/week06 (main)
```

5.9 定义函数 func9，在命名形参的最后，在形参名称前使用 \*\* 允许传入任意数量的命名实参 (被打包为字典)

```
58 def func9(**user):
59     for key, value in user.items():
60         print(f"{key}: {value}")
61
```

```
57 mylib.func9(name="Alice", age=25, city="New York")
```

```
name: Alice
age: 25
city: New York
(week06)
```

5.10 定义函数 `func10`，接受两个位置形参，一个命名形参，尝试在调用时使用 `*` 将可迭代对象（如元组或列表）自动解包，按位置实参传入

5.11 定义函数 `func11`，接受一个命名形参，两个命名形参，尝试在调用时使用 `**` 将映射对象（如字典）自动解包，按命名实参传入

```
63 def func10(arg1, arg2, named_arg="default"):
64     print(f"位置实参 arg1: {arg1}")
65     print(f"位置实参 arg2: {arg2}")
66     print(f"命名实参 named_arg: {named_arg}")
67
68
69 def func11(arg1, arg2):
70     print(f"arg1 的值是: {arg1}")
71     print(f"arg2 的值是: {arg2}")
72
```

```
59 tuple_args = (10, 20)
60 mylib.func10(*tuple_args)
61
62 list_args = [10, 20]
63 mylib.func10(*list_args)
64
65 list_args = [50, 60, "new value"]
66 mylib.func10(*list_args)
67
68 params = {"arg1": 100, "arg2": 200}
69 mylib.func11(**params)
70
```

```

位置实参  arg1): 10
位置实参  arg2): 20
命名实参  named_arg): default
位置实参  arg1): 10
位置实参  arg2): 20
命名实参  named_arg): default
位置实参  arg1): 50
位置实参  arg2): 60
命名实参  named_arg): new value
arg1 的值是: 100
arg2 的值是: 200
(week06)
Lenovo@DESKTOP-C70G2VK MINGW64 /e/研究生上课资料/研一下 上课资料
/金融编程与计算/week06 (main)

```

5.12 定义函数 func12，给函数添加 内嵌文档 (docstring)，给形参和返回值添加 类型注解 (type annotation)，提高函数签名的可读性

```

def func12(arg1: str, arg2: int, named_arg: str = "default")
    "多个参数的调用例子"
    print(f"位置实参  arg1): {arg1}")
    print(f"位置实参  arg2): {arg2}")
    print(f"命名实参  named_arg): {named_arg}")

```

```

71  mylib.func12(7, 8, 9)
72

```

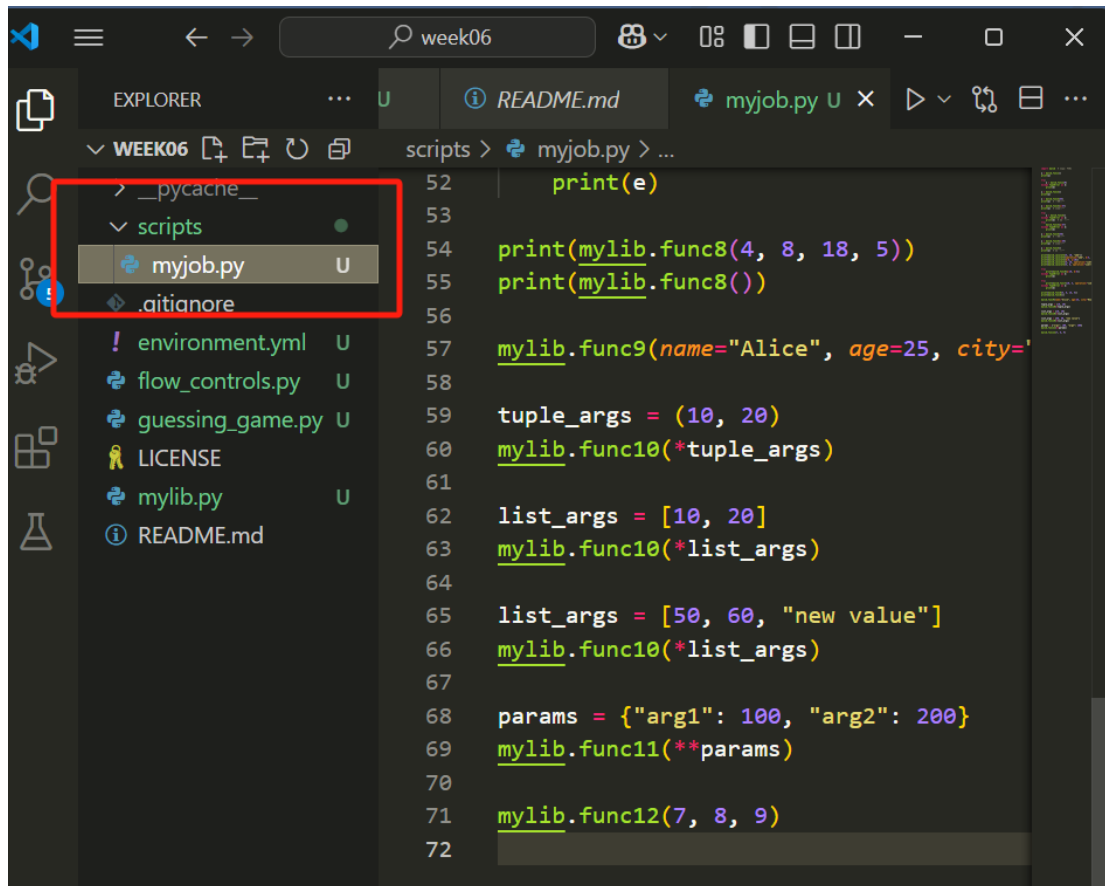
```

位置实参  arg1): 7
位置实参  arg2): 8
命名实参  named_arg): 9
(week06)
Lenovo@DESKTOP-C70G2VK MINGW64 /e/研究生上课资料/研一下 上课资料
/金融编程与计算/week06 (main)
$

```

六、 mylib 模块转变为软件包 (package) 安装进当前的 Conda 环境来使用

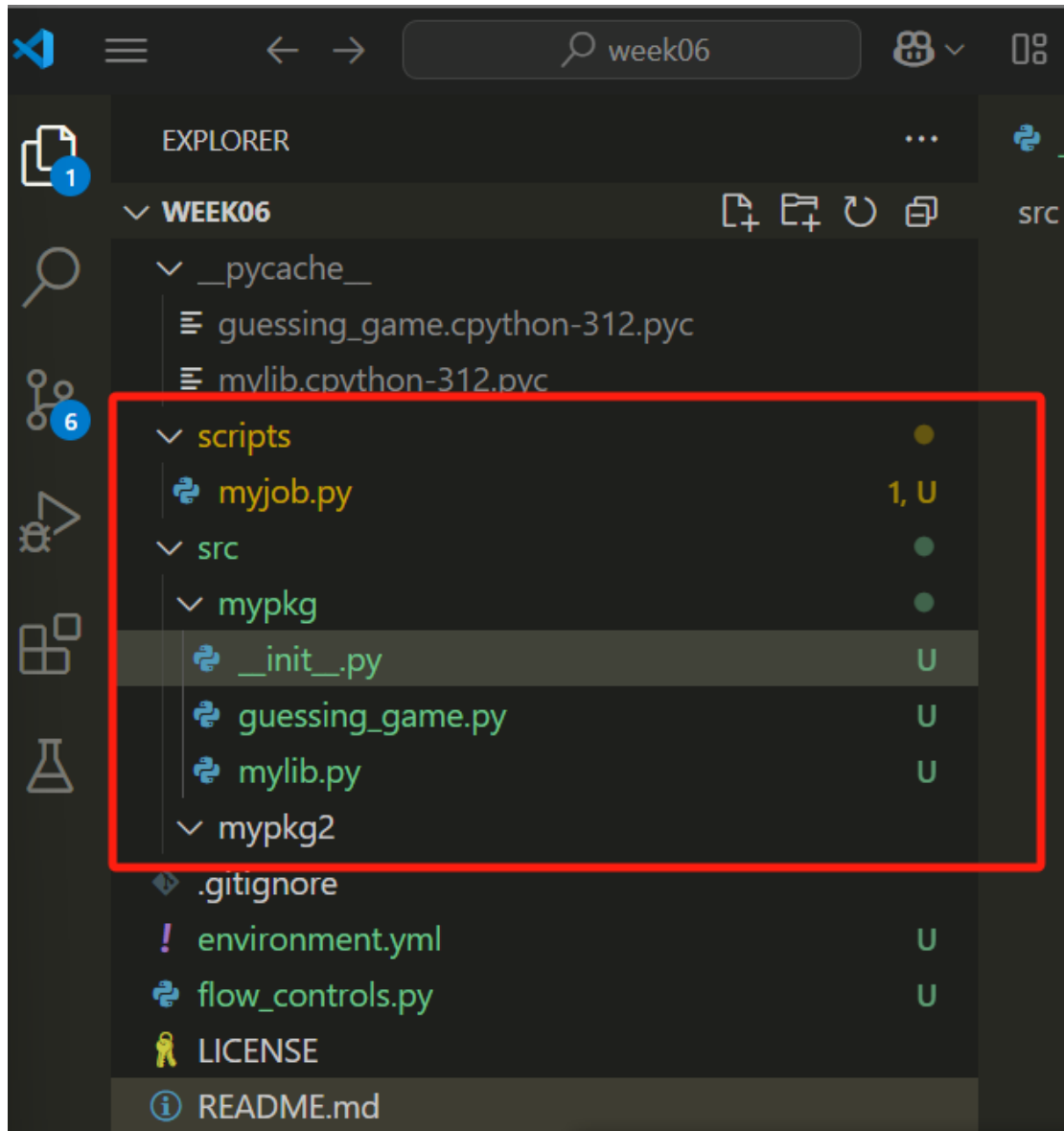
6.1 把 myjob.py 脚本移动至 scripts/myjob.py，再次尝试运行，会发现 import mylib 失败，这是由于 mylib 并没有打包成软件包 (package) 安装



```
Lenovo@DESKTOP-C70G2VK MINGW64 /e/研究生上课资料/研一下 上课资料
/金融编程与计算/week06 (main)
$ python scripts/myjob.py
Traceback (most recent call last):
  File "E:\研究生上课资料\研一下 上课资料\金融编程与计算\week06\
scripts\myjob.py", line 1, in <module>
    import mylib # noqa: F401
    ^^^^^^^^^^^^^
ModuleNotFoundError: No module named 'mylib'
(week06)
Lenovo@DESKTOP-C70G2VK MINGW64 /e/研究生上课资料/研一下 上课资料
/金融编程与计算/week06 (main)
$ pwd
/e/研究生上课资料/研一下 上课资料/金融编程与计算/week06
(week06)
```

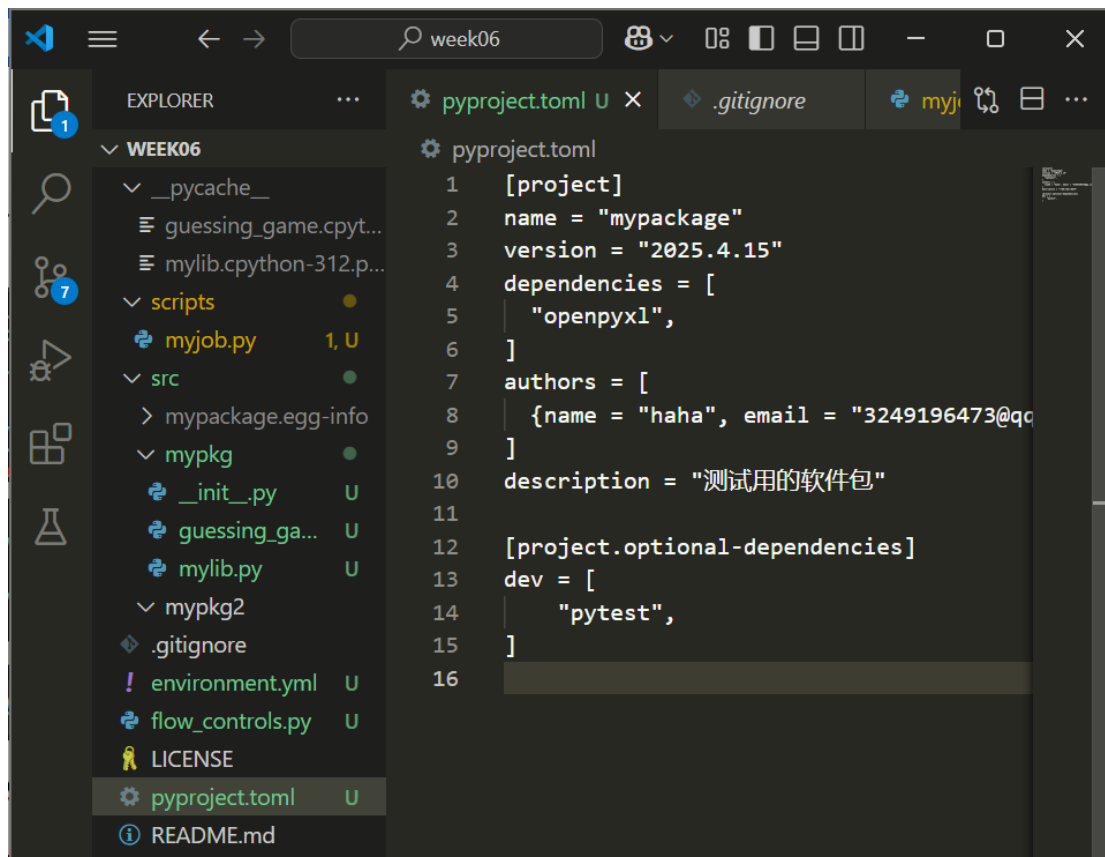
6.2 将 mylib.py 模块移动至 src/mypkg/mylib.py，创建 src/mypkg/\_\_init\_\_.py 文件，准备好软件包的源代码





6.3 创建 `pyproject.toml` 配置文件, 按照文档填写基本的软件包信息

6.4 在 `pyproject.toml` 配置文件里, 按照文档填写软件包的构建(build)配置

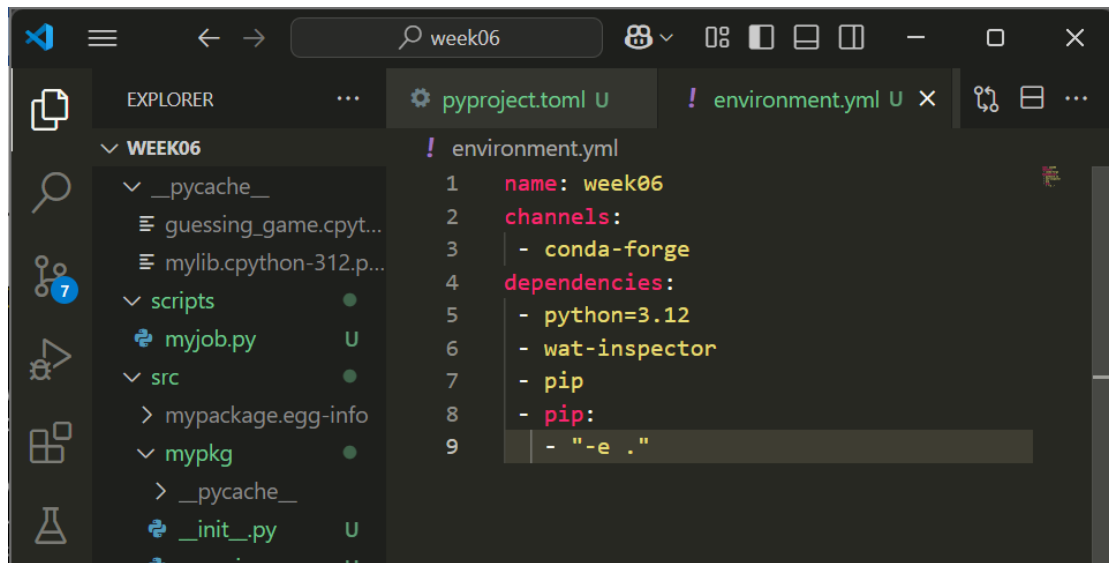


```
Lenovo@DESKTOP-C70G2VK MINGW64 /e/研究生上课资料/研一下 上课资料/金融编程与计算/week06 (main)
$ python
Python 3.12.10 | packaged by conda-forge | (main, Apr 10 2025, 2
2:08:16) [MSC v.1943 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more inform
ation.
>>> import mypkg
>>> mypkg.__file__
'E:\\研究生上课资料\\研一下 上课资料\\金融编程与计算\\week06\\sr
c\\mypkg\\__init__.py'
>>> import mypkg.mylib
>>> mypkg.mylib func1()
File "<stdin>", line 1
mypkg.mylib func1()
^^^^^
SyntaxError: invalid syntax
>>> mypkg.mylib.func1()
0.0710678118654755
>>> import mypkg.guessing_game
>>> mypkg.guessing_game
<module 'mypkg.guessing_game' from 'E:\\研究生上课资料\\研一下
上课资料\\金融编程与计算\\week06\\src\\mypkg\\guessing_game.py'>
>>> mypkg.guessing_game.guessing_game
<function guessing_game at 0x00000220D723D580>
>>> mypkg.guessing_game.guessing_game()
欢迎来到猜数字游戏！我已经想好了一个 1 到 100 之间的数字，你可以
开始猜啦。
(第 1 次尝试) 请输入你猜的数字（输入整数，或者输入 q 回车退出):a
```

6.5 使用 `pip install -e .` 以本地可编辑模式把当前软件包安装进当前 Conda 环境

```
Lenovo@DESKTOP-C70G2VK MINGW64 /e/研究生上课资料/研一下 上课资料
/金融编程与计算/week06 (main)
$ pip install -e .
Obtaining file:///E:/%E7%A0%94%E7%A9%B6%E7%94%9F%E4%B8%8A%E8%AF%
BE%E8%B5%84%E6%96%99/%E7%A0%94%E4%B8%80%E4%B8%8B%20%E4%B8%8A%E8%
AF%BE%E8%B5%84%E6%96%99/%E9%87%91%E8%9E%8D%E7%BC%96%E7%A8%8B%E4%
B8%8E%E8%AE%A1%E7%AE%97/week06
  Installing build dependencies ... done
  Checking if build backend supports build_editable ... done
  Getting requirements to build editable ... done
  Preparing editable metadata (pyproject.toml) ... done
Collecting openpyxl (from mypackage==2025.4.15)
  Downloading openpyxl-3.1.5-py2.py3-none-any.whl.metadata (2.5
kB)
Collecting et_xmlfile (from openpyxl->mypackage==2025.4.15)
  Downloading et_xmlfile-2.0.0-py3-none-any.whl.metadata (2.7 kB
)
  Downloading openpyxl-3.1.5-py2.py3-none-any.whl (250 kB)
  Downloading et_xmlfile-2.0.0-py3-none-any.whl (18 kB)
Building wheels for collected packages: mypackage
  Building editable for mypackage (pyproject.toml) ... done
  Created wheel for mypackage: filename=mypackage-2025.4.15-0.ed
itable-py3-none-any.whl size=7619 sha256=f28997d728a3aaaa2cfec8a
1b7a024d42b89b56c5eec543a65d4c24cb052c721
  Stored in directory: C:\Users\Lenovo\AppData\Local\Temp\pip-ep
hem-wheel-cache-2lcucrw3\wheels\e1\bb\45\af37526f4d5ab073db33ab8
cacfc75ef404c32f29dbf6caebc
Successfully built mypackage
Installing collected packages: et_xmlfile, openpyxl, mypackage
Successfully installed et_xmlfile-2.0.0 mypackage-2025.4.15 open
pyxl-3.1.5
(week06)
```

6.6 修改 `environment.yml` 文件, 使得 `conda env create` 自动安装本地可编辑软件包



```
Successfully installed et-xmlfile-2.0.0 mypackage-2025.4.15 ope5
done
#
# To activate this environment, use
#
#     $ conda activate week06
#
# To deactivate an active environment, use
#
#     $ conda deactivate
#
(base)
```