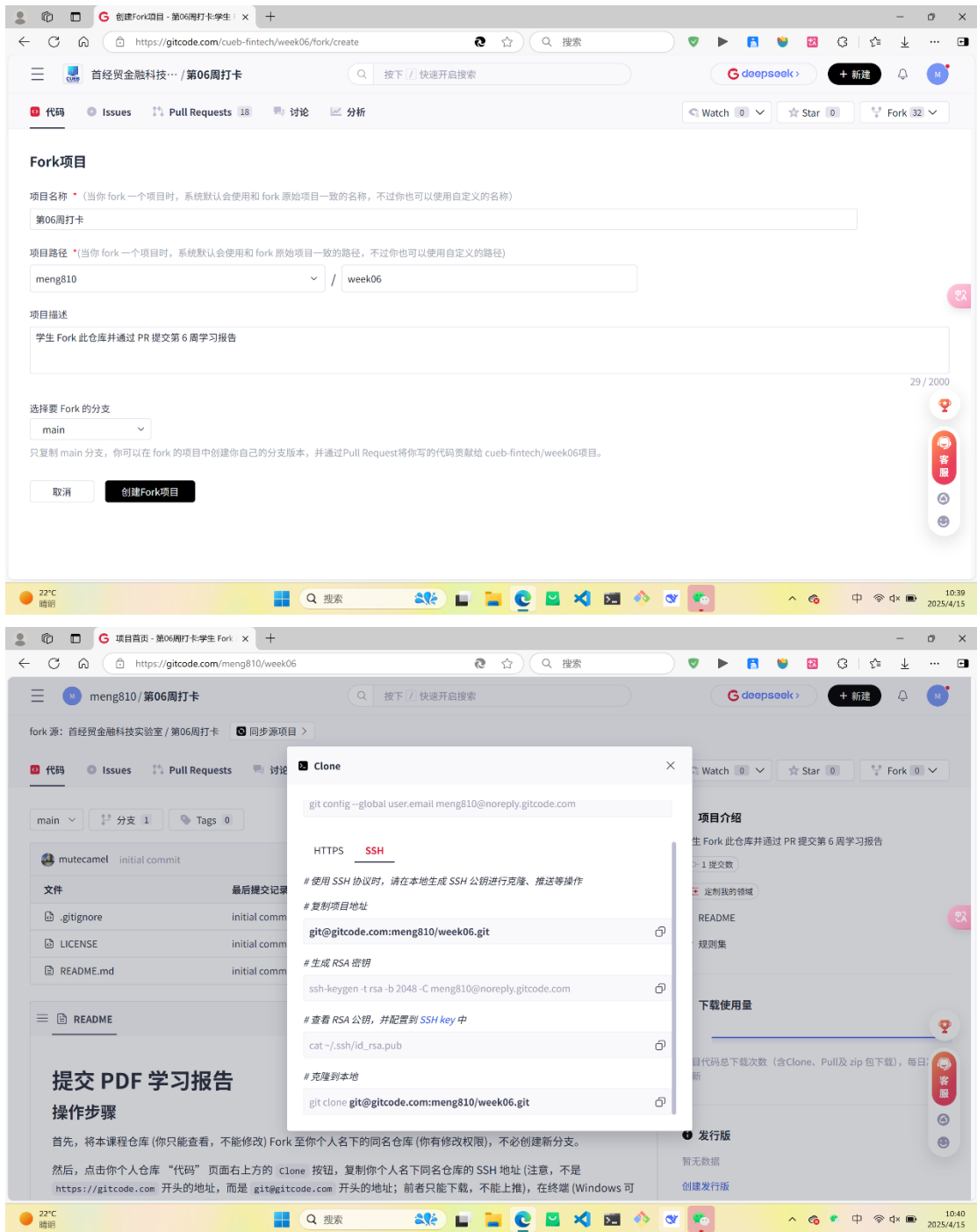


1. Fork 第 06 周打卡仓库至名下，然后将仓库 Clone 到你的本地计算机



```
MINGW64/c/Users/aa/repo/ 0 3月 30 09:56 week04/
(base) aa@LAPTOP-GU1RAG6H MINGW64 ~
$ cd repo

(base) aa@LAPTOP-GU1RAG6H MINGW64 ~/repo
$ git clone git@github.com:meng810/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 | 2.82 MiB/s, done.

(base) aa@LAPTOP-GU1RAG6H MINGW64 ~/repo
$ ll
total 25
drwxr-xr-x 1 aa 197121 0 3月 22 19:53 myproject/
drwxr-xr-x 1 aa 197121 0 3月 21 19:23 prj1/
drwxr-xr-x 1 aa 197121 0 3月 15 18:23 repo/
-rw-r--r-- 1 aa 197121 514 3月 10 23:03 script1.py
drwxr-xr-x 1 aa 197121 0 3月 13 09:05 week01/
drwxr-xr-x 1 aa 197121 0 3月 15 21:00 week02/
drwxr-xr-x 1 aa 197121 0 3月 23 10:08 week03/
drwxr-xr-x 1 aa 197121 0 3月 31 19:15 week04/
drwxr-xr-x 1 aa 197121 0 4月 13 17:04 week05/
drwxr-xr-x 1 aa 197121 0 4月 15 10:46 week06/

(base) aa@LAPTOP-GU1RAG6H MINGW64 ~/repo
$ cd week06

(base) aa@LAPTOP-GU1RAG6H MINGW64 ~/repo/week06 (main)
$ ls -l
total 24
-rw-r--r-- 1 aa 197121 18805 4月 15 10:46 LICENSE
-rw-r--r-- 1 aa 197121 2239 4月 15 10:46 README.md

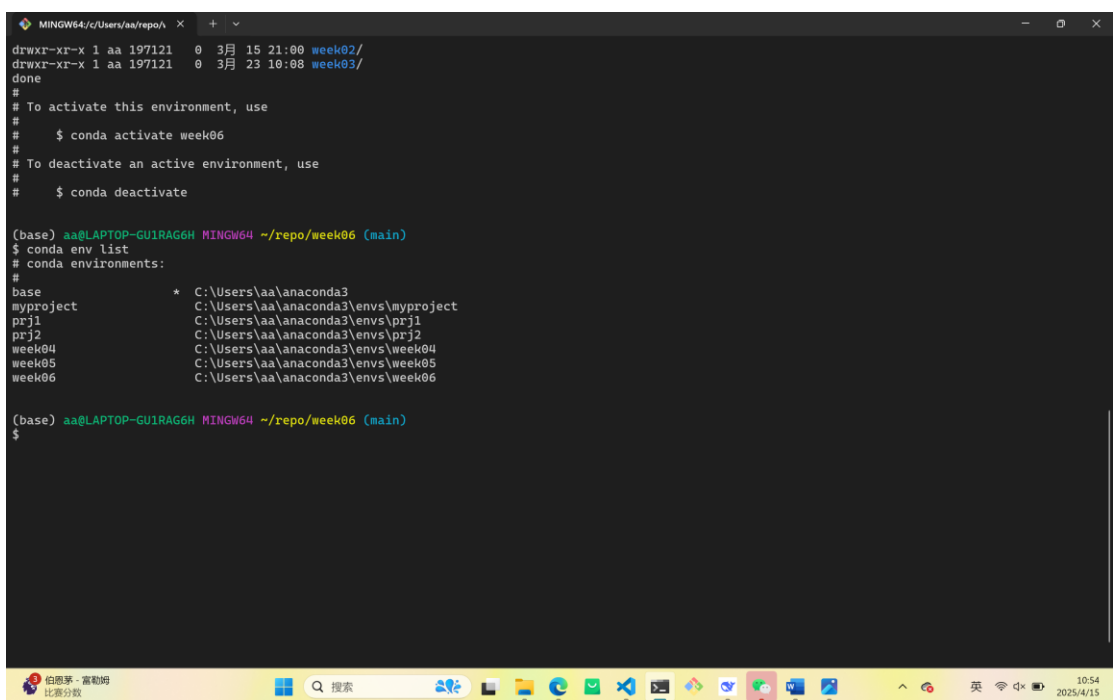
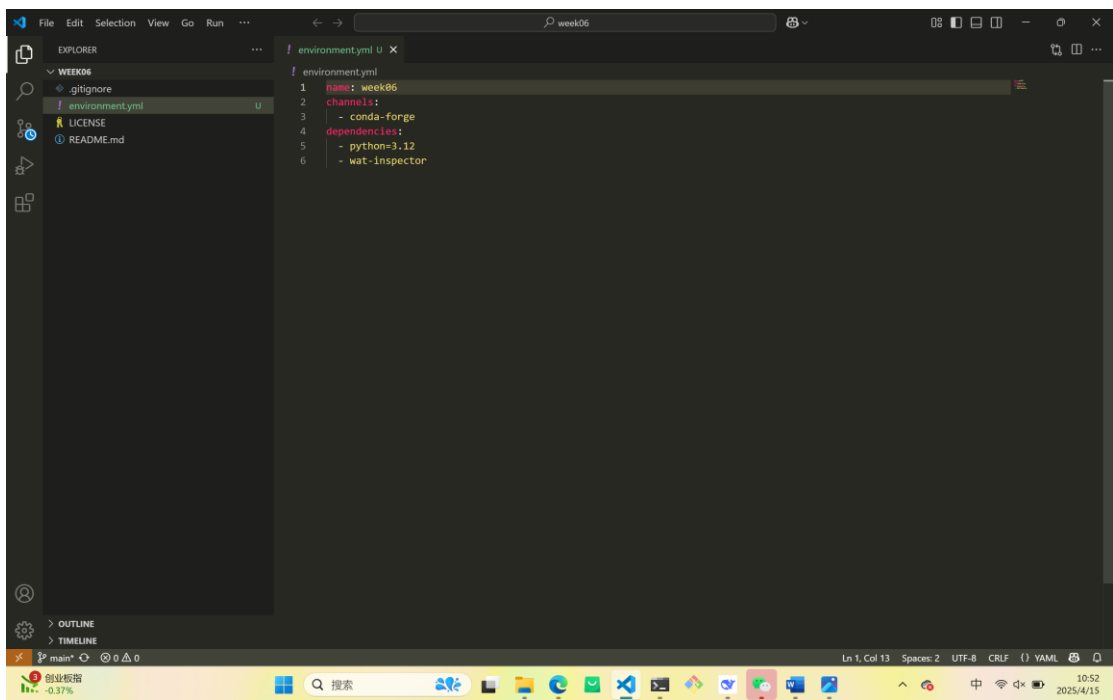
(base) aa@LAPTOP-GU1RAG6H MINGW64 ~/repo/week06 (main)
$
```

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

```
(base) aa@LAPTOP-GU1RAG6H MINGW64 ~/repo/week06 (main)
$ cp ../week05/environment.yml ./

(base) aa@LAPTOP-GU1RAG6H MINGW64 ~/repo/week06 (main)
$ ll
total 25
-rw-r--r-- 1 aa 197121 91 4月 15 10:50 environment.yml
-rw-r--r-- 1 aa 197121 18805 4月 15 10:46 LICENSE
-rw-r--r-- 1 aa 197121 2239 4月 15 10:46 README.md

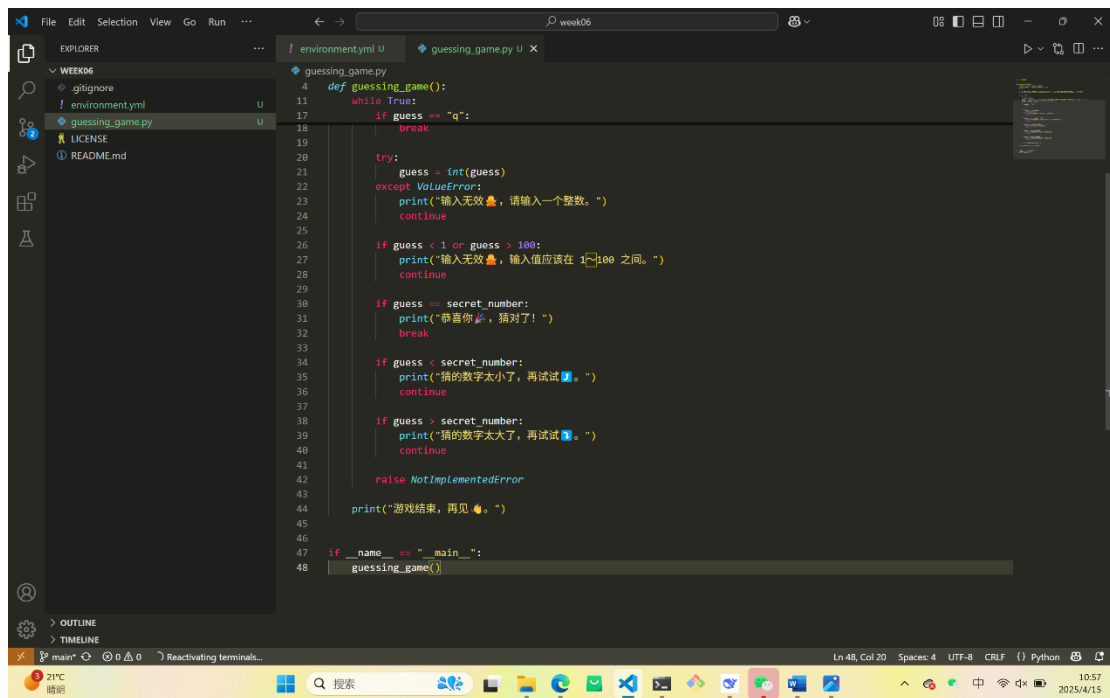
(base) aa@LAPTOP-GU1RAG6H MINGW64 ~/repo/week06 (main)
$
```



```
MINGW64/c:/Users/aa/repo/aa x + v
(base) aa@LAPTOP-GUIRAG6H MINGW64 ~/repo/week06 (main)
$ conda env list
# conda environments:
#
base                  * C:\Users\aa\anaconda3
myproject             C:\Users\aa\anaconda3\envs\myproject
prj1                  C:\Users\aa\anaconda3\envs\prj1
prj2                  C:\Users\aa\anaconda3\envs\prj2
week04                C:\Users\aa\anaconda3\envs\week04
week05                C:\Users\aa\anaconda3\envs\week05
week06                C:\Users\aa\anaconda3\envs\week06

(base) aa@LAPTOP-GUIRAG6H MINGW64 ~/repo/week06 (main)
$ conda activate week06
(week06)
aa@LAPTOP-GUIRAG6H MINGW64 ~/repo/week06 (main)
$ conda list
# packages in environment at C:\Users\aa\anaconda3\envs\week06:
#
# Name                        Version                        Build      Channel
bz2ip2                        1.0.8                        h2466b09_7  conda-forge
ca-certificates               2025.1.31                    h5668100_0  conda-forge
libexpat                      2.7.0                        he0923c2_0  conda-forge
libffi                         3.4.6                        h537db12_1  conda-forge
liblzma                       5.8.1                        h2466b09_0  conda-forge
libsqlite                     3.49.1                       h67fdade_2  conda-forge
libzlib                       1.3.1                        h2466b09_2  conda-forge
openssl                       3.5.0                        ha4e3fda_0  conda-forge
pip                           25.0.1                       pyh8b19718_0 conda-forge
python                        3.12.10                      h3f84c4b_0_cpython conda-forge
setuptools                    78.1.0                       pyhff2d567_0 conda-forge
tk                             8.6.13                       h5226925_1  conda-forge
tzdata                        2025b                         h78e185d_0  conda-forge
ucrt                          10.0.22621.0                 h57928b3_1  conda-forge
vc                             14.3                         h2b53caa_26  conda-forge
vc14_runtime                  14.42.34438                  hfd919c2_26  conda-forge
wat-Inspector                  0.4.3                        pyhff2d567_0 conda-forge
wheel                         0.45.1                       pyhd8ed1ab_1 conda-forge
(week06)
aa@LAPTOP-GUIRAG6H MINGW64 ~/repo/week06 (main)
$
```

3. 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：



```
environment.yml U  guessing_game.py X
guessing_game.py
4 def guessing_game():
11     while True:
17         if guess == "q":
18             break
19
20         try:
21             guess = int(guess)
22         except ValueError:
23             print("输入无效，请输入一个整数。")
24             continue
25
26         if guess < 1 or guess > 100:
27             print("输入无效，输入值应该在 1~100 之间。")
28             continue
29
30         if guess == secret_number:
31             print("恭喜你，猜对了！")
32             break
33
34         if guess < secret_number:
35             print("猜的数字太小了，再试试。")
36             continue
37
38         if guess > secret_number:
39             print("猜的数字太大了，再试试。")
40             continue
41
42         raise NotImplementedError
43
44     print("游戏结束，再见。")
45
46
47 if __name__ == "__main__":
48     guessing_game()
```

```

aa@LAPTOP-GUIRAG6H MINGW64 ~/repo/week06 (main)
$ python -m pdb guessing_game.py
> c:\users\aa\repo\week06\guessing_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
> c:\users\aa\repo\week06\guessing_game.py(4)<module>()
-> def guessing_game():
(Pdb)
> c:\users\aa\repo\week06\guessing_game.py(47)<module>()
-> if __name__ == "__main__":
(Pdb)
> c:\users\aa\repo\week06\guessing_game.py(48)<module>()
-> guessing_game()

```

```

18
19         if guess == "q":
20             break
21
22     ->     try:
23         guess = int(guess)
24     except ValueError:
25         print("输入无效 🙅, 请输入一个整数。")
26         continue
27
(Pdb) n
> c:\users\16386\repo\week06\guessing_game.py(23)guessing_game()
-> guess = int(guess)
(Pdb) l
18
19         if guess == "q":
20             break
21
22     ->     try:
23         guess = int(guess)
24     except ValueError:
25         print("输入无效 🙅, 请输入一个整数。")
26         continue
27
28         if guess < 1 or guess > 100:
(Pdb) p guess
'50'
(Pdb) p int(guess)
50
(Pdb)

```

```
MINGW64-~/Users/aa/repo/h + ~
16
17     if guess == "q":
18         break
19
20     try:
21         guess = int(guess)
22     except ValueError:
(Pdb) n
> c:\users\aa\repo\week06\guessing_game.py(4)<module><C>
-> def guessing_game():
(Pdb) p n
*** NameError: name 'n' is not defined
(Pdb) quit()
(week06)
aa@LAPTOP-GUIRAG6H MINGW64 ~/repo/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): a
输入无效 🚫，请输入一个整数。
(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): bb
输入无效 🚫，请输入一个整数。
(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): 30
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): 88
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): 2
猜的数字太小了，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): 40
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): 12
猜的数字太小了，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): 20
猜的数字太大了，再试试 🔄。
(第 9 次尝试) 请输入你猜的数字 (输入整数，或者输入 q 回车退出): 15
猜的数字太小了，再试试 🔄。
(第 10 次尝试) 请输入你猜的数字 (输入整数，或者输入 q 回车退出): 17
猜的数字太小了，再试试 🔄。
(第 11 次尝试) 请输入你猜的数字 (输入整数，或者输入 q 回车退出): 18
恭喜你 🎉，猜对了！
游戏结束，再见 👋。
(week06)
aa@LAPTOP-GUIRAG6H MINGW64 ~/repo/week06 (main)
$
```

4. 创建一个 `flow_controls.py` 文件，让豆包（或 DeepSeek 等任何大模型）生成例子，尝试运行，体会理解以下 Python 流程控制语句：
- for 迭代循环 (iteration loop)

The screenshot displays the Visual Studio Code (VS Code) interface on a Windows 10 desktop. The Explorer sidebar on the left shows the project structure, including files like `__pycache__`, `.gitignore`, `environment.yml`, `flow_controls.py`, `guessing_game.py`, `LICENSE`, and `README.md`. The `flow_controls.py` file is open in the editor, showing the following Python code:

```

1  # 遍历列表中的元素
2  fruits = ["apple", "banana", "cherry"]
3  for fruit in fruits:
4      print(fruit)
5
6  # 打印 0 到 4 (不包含 5)
7  for i in range(5):
8      print(i)
9
10 # 指定起始和步长 (输出 2,4,6,8)
11 for i in range(2, 10, 2):
12     print(i)
13
14
15 # 遍历字典的键和值
16 person = {"name": "Alice", "age": 25, "city": "New York"}
17 for key, value in person.items():
18     print(f"{key}: {value}")

```

The status bar at the bottom indicates the current file is `main.py` in the `main` directory, with 18 lines of code, 29 columns, 4 spaces, UTF-8 encoding, CRLF line endings, and Python syntax highlighting. The system tray at the bottom right shows the date and time as 11:25 on 2025/4/15.

```
MINGW64/c/Users/aa/repo/  ×  +  -
恭喜你 🎉, 猜对了!
游戏结束, 再见 🙋.
(repo06)
aa@LAPTOP-GU1RAG6H MINGW64 ~/repo/week06 (main)
$ python flow_controls.py
apple
banana
cherry
(repo06)
aa@LAPTOP-GU1RAG6H MINGW64 ~/repo/week06 (main)
$ python flow_controls.py
apple
banana
cherry
0
1
2
3
4
2
4
6
8
(repo06)
aa@LAPTOP-GU1RAG6H MINGW64 ~/repo/week06 (main)
$ python flow_controls.py
apple
banana
cherry
0
1
2
3
4
2
4
6
8
name: Alice
age: 25
city: New York
(repo06)
aa@LAPTOP-GU1RAG6H MINGW64 ~/repo/week06 (main)
```

while 条件循环 (conditional loop)

```
20 # 从 1 数到 5
21 count = 1
22 while count <= 5:
23     print(count)
24     count += 1 # 等价于 count = count + 1
25
26
27 # 要求用户输入 "yes" 才继续
28 user_input = ""
29 while user_input.lower() != "yes":
30     user_input = input('请输入 "yes" 继续: ')
31 print("程序继续执行! ")
32
33
34 # 输入 "quit" 退出循环
35 while True:
36     command = input("输入命令 (输入 quit 退出): ")
37     if command == "quit":
38         break
39     print(f"执行命令: {command}")
40 print("循环结束")
```

```
MINGW64/c/Users/aa/repo/  ×  +  -
4
6
8
name: Alice
age: 25
city: New York
1
2
3
4
5
请输入 "yes" 继续: yes
程序继续执行!
(loop06)
aa@LAPTOP-GU1RAG6H MINGW64 ~/repo/week06 (main)
$ python flow_controls.py
apple
banana
cherry
0
1
2
3
4
5
6
7
8
name: Alice
age: 25
city: New York
1
2
3
4
5
请输入 "yes" 继续: yes
程序继续执行!
输入命令 (输入 quit 退出): quit
循环结束
(loop06)
aa@LAPTOP-GU1RAG6H MINGW64 ~/repo/week06 (main)
$
```

try...except[...except...else...finally] 捕捉异常的处理、raise 主动抛出异常

```
def check_age(age):
    try:
        if age < 0:
            raise ValueError("年龄不能为负数!") # 主动抛出异常
        elif age < 18:
            print("未成年人")
        else:
            print("成年人")
    except ValueError as e:
        print(f"错误: {e}")

check_age(25) # 输出: 成年人
check_age(-5) # 输出: 错误: 年龄不能为负数!
```

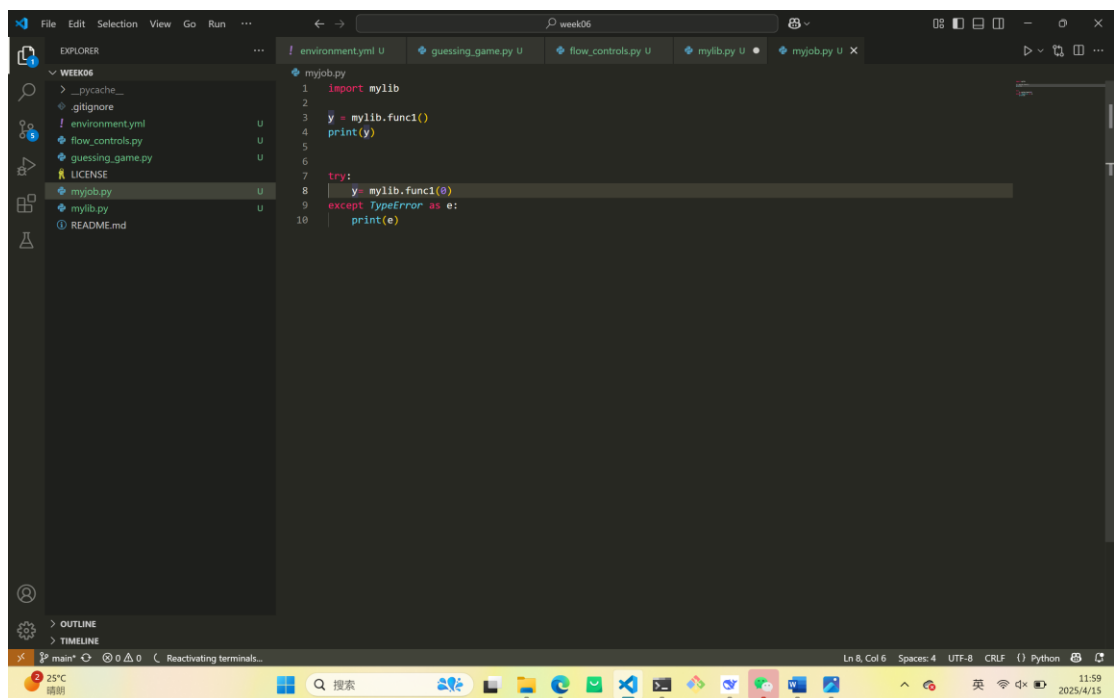


```

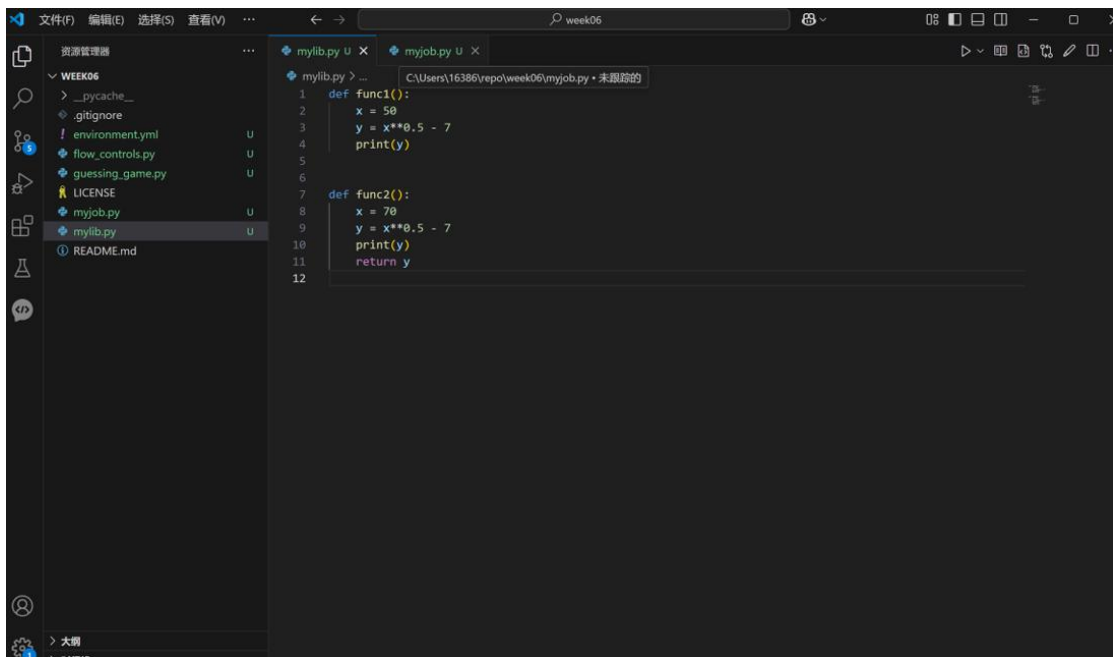
aa@LAPTOP-GU1RAG6H MINGW64 ~/repo/week06 (main)
$ python myjob.py
--Return--
> c:\users\aa\repo\week06\myjob.py(4)<module>()->None
-> breakpoint()
(Pdb) p mylib
<module 'mylib' from 'C:\\Users\\aa\\repo\\week06\\mylib.py'>
(Pdb) l
   1      import mylib
   2
   3
   4  -> breakpoint()
[EOF]
(Pdb) p mylib
<module 'mylib' from 'C:\\Users\\aa\\repo\\week06\\mylib.py'>
(Pdb) wat / mylib

```

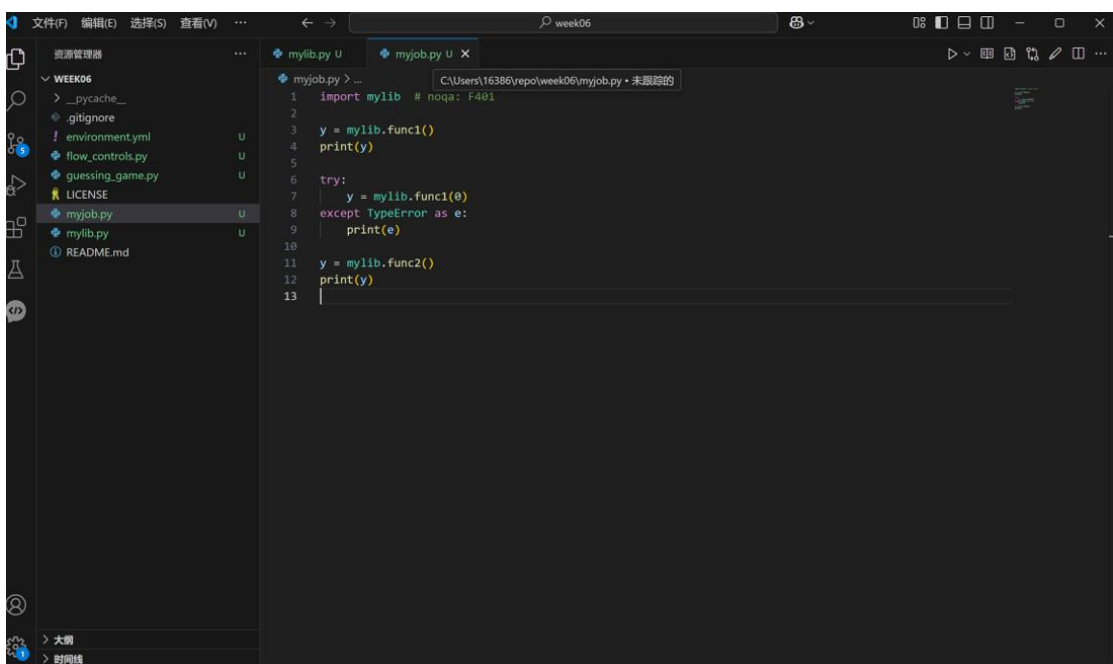
定义函数 func1，没有形参，没有返回值



定义函数 func2，没有形参，有返回值

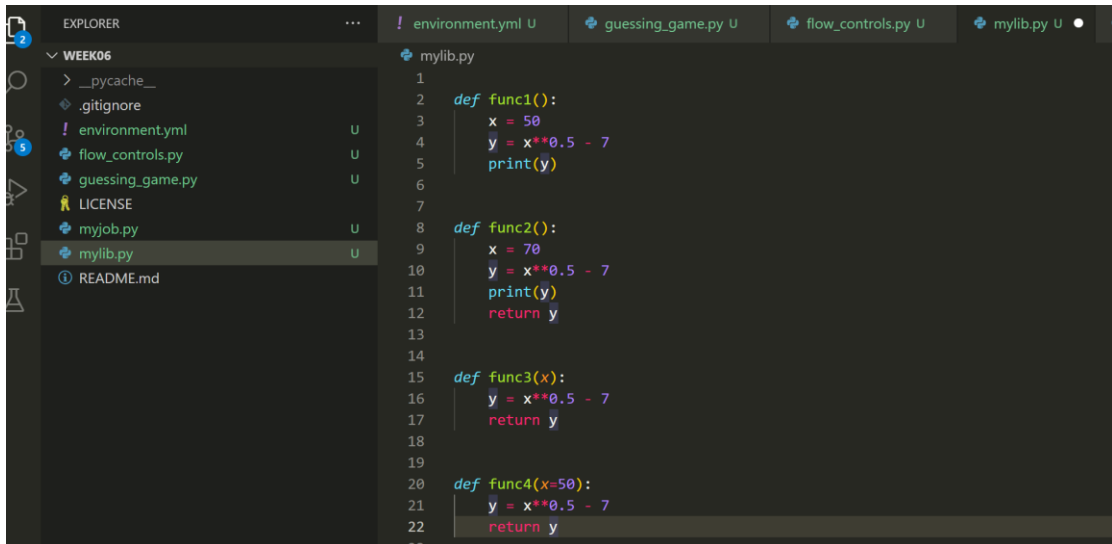


```
1 def func1():
2     x = 50
3     y = x**0.5 - 7
4     print(y)
5
6
7 def func2():
8     x = 70
9     y = x**0.5 - 7
10    print(y)
11    return y
12
```

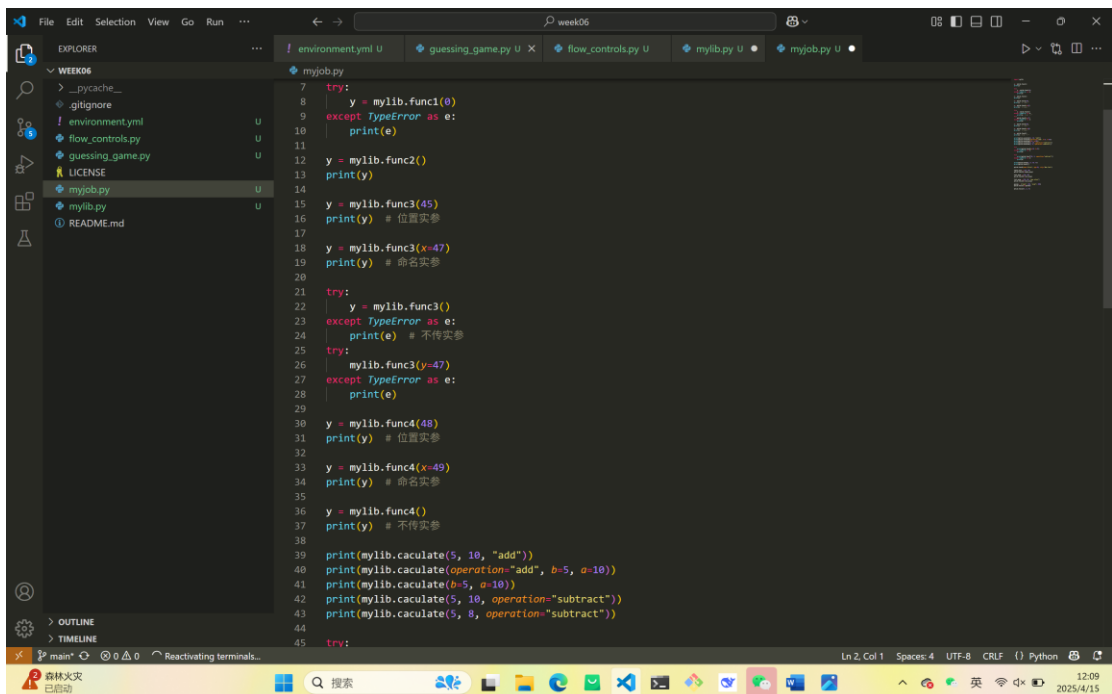


```
1 import mylib # noqa: F401
2
3 y = mylib.func1()
4 print(y)
5
6 try:
7     y = mylib.func1(0)
8 except TypeError as e:
9     print(e)
10
11 y = mylib.func2()
12 print(y)
13
```

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)

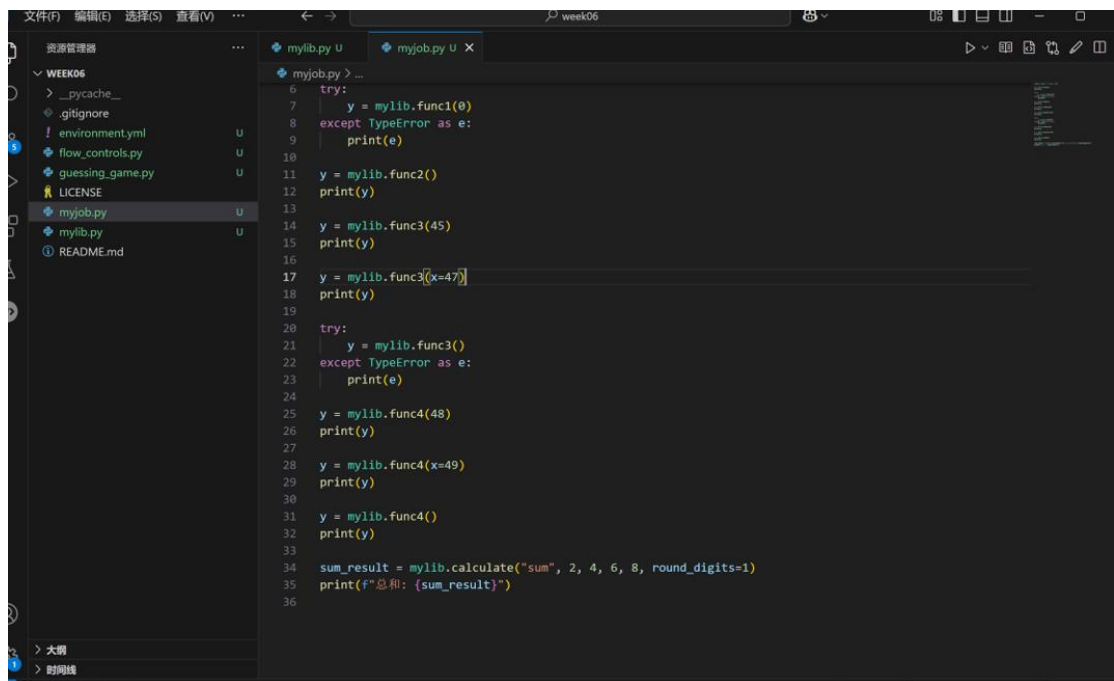


```
1
2 def func1():
3     x = 50
4     y = x*0.5 - 7
5     print(y)
6
7
8 def func2():
9     x = 70
10    y = x*0.5 - 7
11    print(y)
12    return y
13
14
15 def func3(x):
16    y = x*0.5 - 7
17    return y
18
19
20 def func4(x=50):
21    y = x*0.5 - 7
22    return y
23
```



```
7 try:
8     y = mylib.func1(0)
9 except TypeError as e:
10    print(e)
11
12 y = mylib.func2()
13 print(y)
14
15 y = mylib.func3(45)
16 print(y) # 位置实参
17
18 y = mylib.func3(x=47)
19 print(y) # 命名实参
20
21 try:
22     y = mylib.func3()
23 except TypeError as e:
24     print(e) # 不得实参
25
26 try:
27     mylib.func3(y=47)
28 except TypeError as e:
29     print(e)
30
31 y = mylib.func4(48)
32 print(y) # 位置实参
33
34 y = mylib.func4(x=49)
35 print(y) # 命名实参
36
37 y = mylib.func4()
38 print(y) # 不得实参
39
40 print(mylib.caculate(5, 10, "add"))
41 print(mylib.caculate(operation="add", b=5, a=10))
42 print(mylib.caculate(b=5, a=10))
43 print(mylib.caculate(5, 10, operation="subtract"))
44 print(mylib.caculate(5, 8, operation="subtract"))
45
46 try:
```

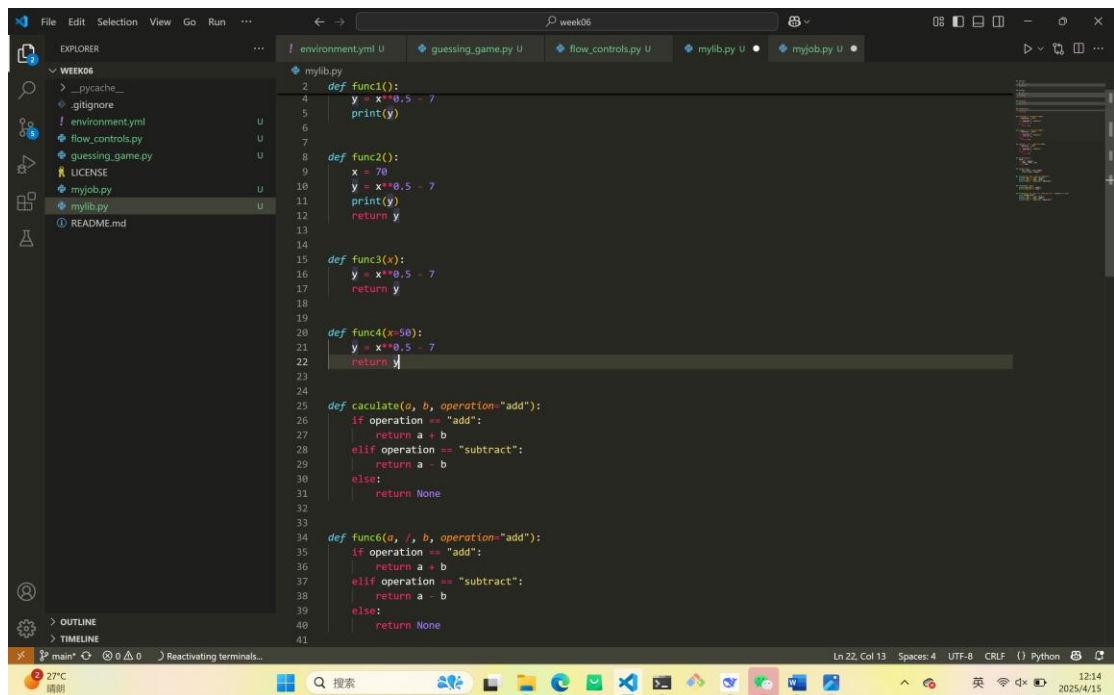
定义函数 func5，接受多个位置形参和命名形参，尝试以位置/命名各种不同方式传入实参，注意位置参数必须排在命名参数之前



The screenshot shows a code editor with two files open: `mylib.py` and `myjob.py`. The `myjob.py` file contains the following Python code:

```
6 try:
7     y = mylib.func1(0)
8 except TypeError as e:
9     print(e)
10
11 y = mylib.func2()
12 print(y)
13
14 y = mylib.func3(45)
15 print(y)
16
17 y = mylib.func3(x=47)
18 print(y)
19
20 try:
21     y = mylib.func3()
22 except TypeError as e:
23     print(e)
24
25 y = mylib.func4(48)
26 print(y)
27
28 y = mylib.func4(x=49)
29 print(y)
30
31 y = mylib.func4()
32 print(y)
33
34 sum_result = mylib.calculate("sum", 2, 4, 6, 8, round_digits=1)
35 print(f"总和: {sum_result}")
36
```

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



The screenshot shows a code editor with the `mylib.py` file open. The file contains the following Python code:

```
2 def func1():
3     y = x**0.5 - 7
4     print(y)
5
6 def func2():
7     x = 70
8     y = x**0.5 - 7
9     print(y)
10    return y
11
12 def func3(x):
13     y = x**0.5 - 7
14     return y
15
16 def func4(x=50):
17     y = x**0.5 - 7
18     return y
19
20 def caculate(a, b, operation="add"):
21     if operation == "add":
22         return a + b
23     elif operation == "subtract":
24         return a - b
25     else:
26         return None
27
28 def func6(a, /, b, operation="add"):
29     if operation == "add":
30         return a + b
31     elif operation == "subtract":
32         return a - b
33     else:
34         return None
35
```

```
12 y = mylib.func2()
13 print(y)
14
15 y = mylib.func3(45)
16 print(y) # 位置实参
17
18 y = mylib.func3(x=47)
19 print(y) # 命名实参
20
21 try:
22     y = mylib.func3()
23 except TypeError as e:
24     print(e) # 不传实参
25
26 mylib.func3(y=47)
27 except TypeError as e:
28     print(e)
29
30 y = mylib.func4(48)
31 print(y) # 位置实参
32
33 y = mylib.func4(x=49)
34 print(y) # 命名实参
35
36 y = mylib.func4()
37 print(y) # 不传实参
38
39 print(mylib.caculate(5, 10, "add"))
40 print(mylib.caculate(operation="add", b=5, a=10))
41 print(mylib.caculate(b=5, a=10))
42 print(mylib.caculate(5, 10, operation="subtract"))
43 print(mylib.caculate(5, 8, operation="subtract"))
44
45
46 try:
47     print(mylib.func6(a=10, b=5))
48 except TypeError as e:
49     print(e)
```

- 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
- 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)
- 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)
- 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
- 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
- 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。

```
42
43 def func7(a, b, *, operation="add"):
44     if operation == "add":
45         return a + b
46     elif operation == "subtract":
47         return a - b
48     else:
49         return None
50
51
52 def func8(*numbers):
53     total = 0
54     for num in numbers:
55         total = total + num
56     return total
57
58
59 def func9(**user):
60     for key, value in user.items():
61         print(f"{key}: {value}")
62
63
64 def func10(arg1, arg2, named_arg="default"):
65     print(f"位置实参 arg1: {arg1}")
66     print(f"位置实参 arg2: {arg2}")
67     print(f"命名实参 named_arg: {named_arg}")
68
69
70 def func11(arg1, arg2):
71     print(f"arg1 的值是: {arg1}")
72     print(f"arg2 的值是: {arg2}")
73
74
75 def func12(arg1: str, arg2: int, named_arg: str = "default") -> None:
76     "多个参数的调用例子"
77     print(f"位置实参 arg1: {arg1}")
78     print(f"位置实参 arg2: {arg2}")
79     print(f"命名实参 named_arg: {named_arg}")
80
```

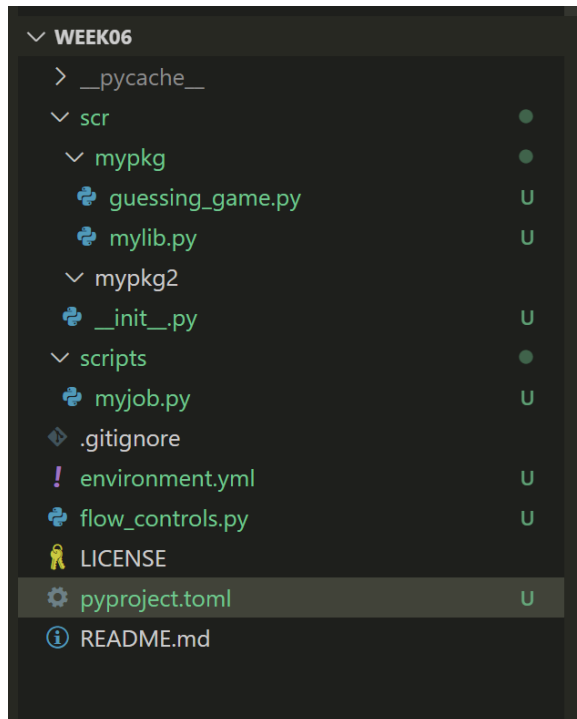
```
35
36 y = mylib.func4()
37 print(y) # 不得实参
38
39 print(mylib.caculate(5, 10, "add"))
40 print(mylib.caculate(operation="add", b=5, a=10))
41 print(mylib.caculate(b=5, a=10))
42 print(mylib.caculate(5, 10, operation="subtract"))
43 print(mylib.caculate(5, 8, operation="subtract"))
44
45 try:
46     print(mylib.func6(a=10, b=5))
47 except TypeError as e:
48     print(e)
49
50 try:
51     print(mylib.func7(10, 5, operation="subtract"))
52 except TypeError as e:
53     print(e)
54
55 print(mylib.func8(4, 8, 10, 5))
56 print(mylib.func8())
57
58 mylib.func9(name="Alice", age=25, city="New York")
59
60 tuple_args = (10, 20)
61 mylib.func10(*tuple_args)
62
63 list_args = [10, 20]
64 mylib.func10(*list_args)
65
66 list_args = [50, 60, "new value"]
67 mylib.func10(*list_args)
68
69 params = {"arg1": 100, "arg2": 200}
70 mylib.func11(**params)
71
72 mylib.func12(7, 8, 9)
73
```

6. 把 mylib 模块转变为 **软件包** (package) 安装进当前的 Conda 环境来使用

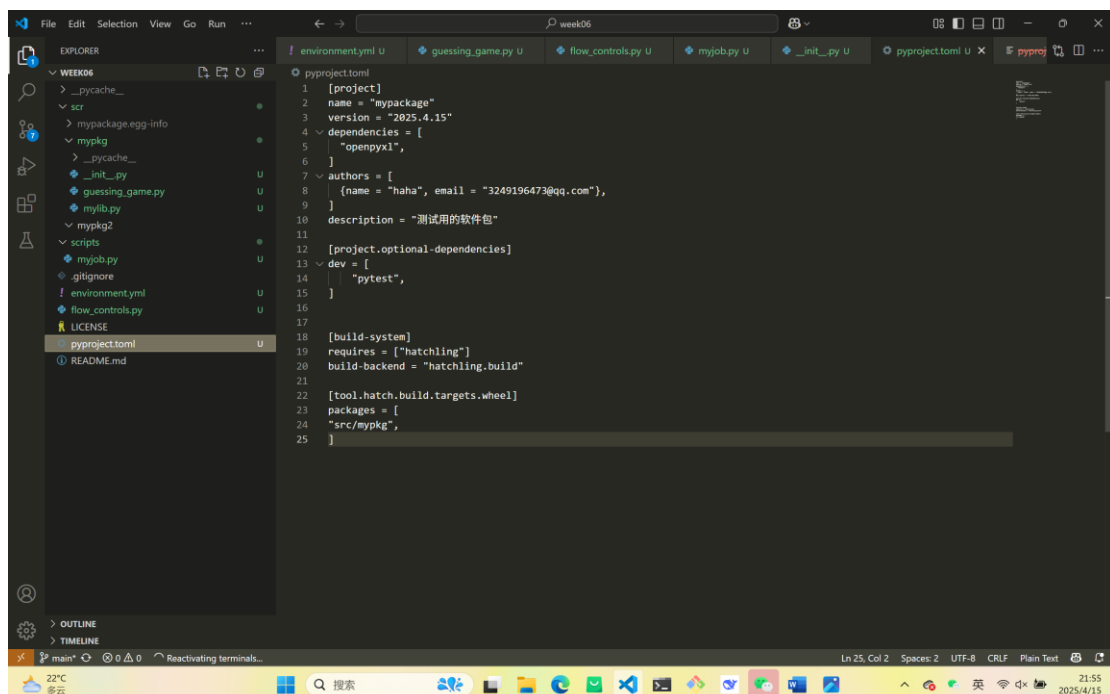
把 myjob.py 脚本移动至 scripts/myjob.py, 再次尝试运行, 会发现 import mylib 失败, 这是由于 mylib 并没有打包成 软件包 (package) 安装



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



创建 pyproject.toml 配置文件，按照文档填写基本的软件包信息



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

