

### 1、创建 guessing\_game.py 文件，运用 pdb 试验运行流程（猜数字游戏为例）

The screenshot displays a code editor with a file explorer on the left and a terminal at the bottom. The file explorer shows a project named 'WEEK06' containing files like '.gitignore', 'environment.yml', 'guessing\_game.py', 'LICENSE', and 'README.md'. The 'guessing\_game.py' file is open in the editor, showing a Python script for a number-guessing game. The script generates a random number between 1 and 100 and allows the user to guess it, providing feedback on whether the guess is too low, too high, or correct. The terminal shows the execution of the script, including the game's instructions and the user's input '49', which is correctly identified as the secret number.

```
! environment.yml U  guessing_game.py X
guessing_game.py > ...
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:
12         n += 1
13         # 获取玩家输入
14         guess = input(
15             f"({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:
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:
29             print("输入无效，输入值应该在 1-100 之间。")
30             continue
31
32         if guess == secret_number:
33             print("恭喜你，猜对了！")
34             break
35
```

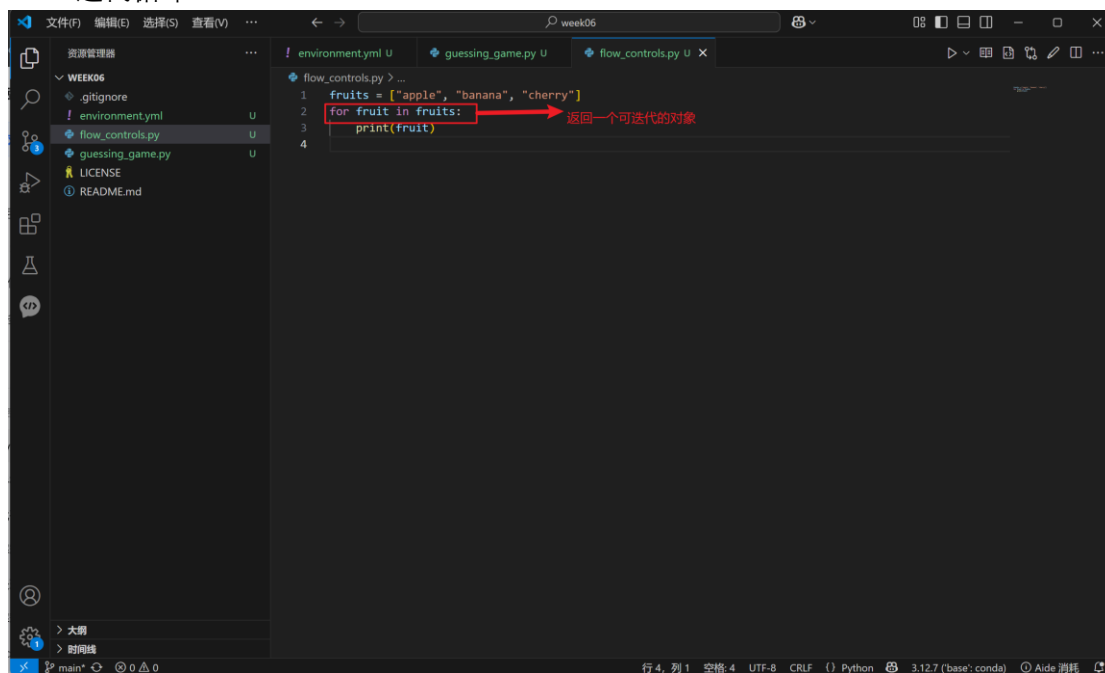
Terminal Output:

```
MINGW64: c:/Users/16386/rep/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): 49
游戏结束，再见！
(week06)
MINGW64: c:/Users/16386/rep/week06 (main)
$ python -m pdb guessing_game.py
> c:\users\16386\rep\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\16386\rep\week06\guessing_game.py(4)<module>()
-> def guessing_game():
(Pdb)
> c:\users\16386\rep\week06\guessing_game.py(49)<module>()
-> if __name__ == "__main__":
(Pdb)
> c:\users\16386\rep\week06\guessing_game.py(50)<module>()
```

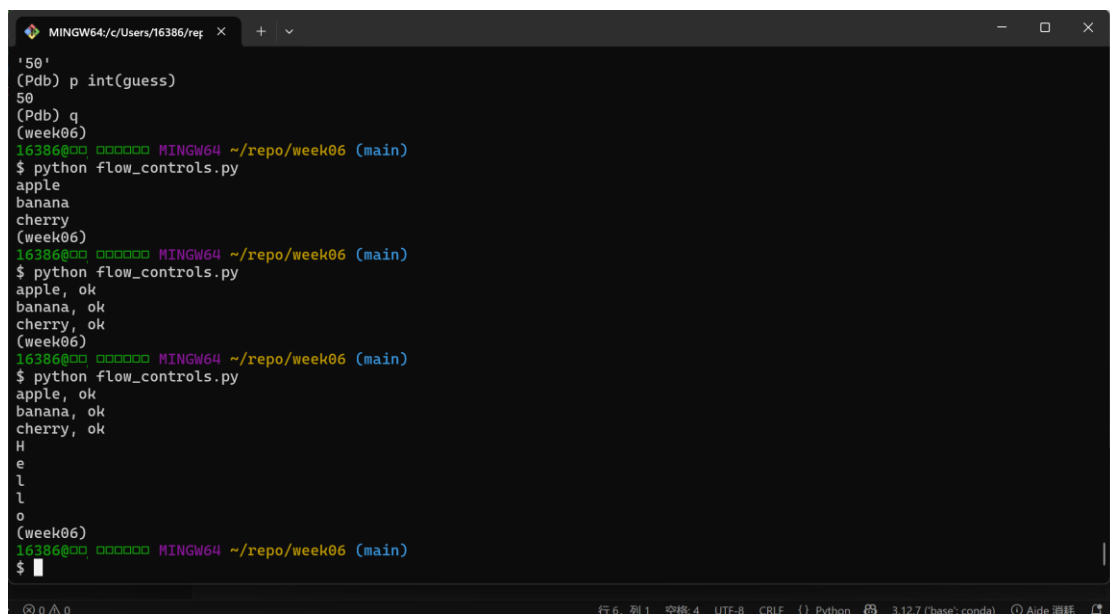
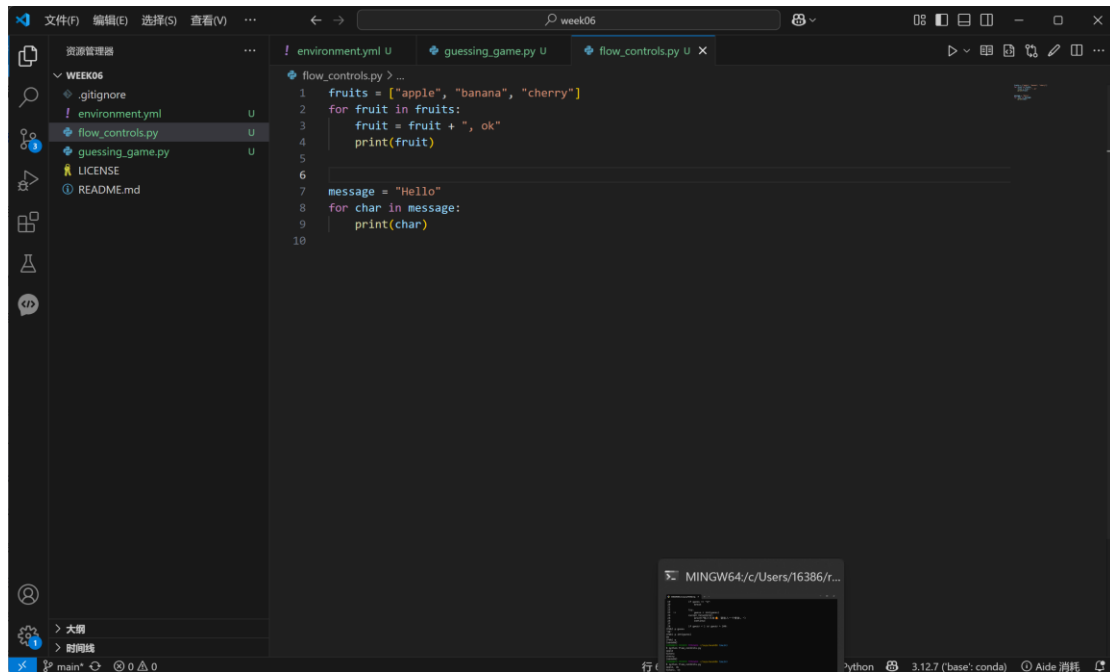
```
MINGW64: c:/Users/16386/rep
18
19     if guess == "q":
20         break
21
22 ->     try:
23         guess = int(guess)
24     except ValueError:
25         print("输入无效 🍌, 请输入一个整数。")
26         continue
27
(Pdb) n
> c:\users\16386\repo\week06\guessing_game.py(23)guessing_game()
-> guess = int(guess)
(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)
```

2、创建 flow\_controls.py 文件，让豆包生成例子，理解以下语句

For: 迭代循环



```
environment.yml U  guessing_game.py U  flow_controls.py X
flow_controls.py > ...
1  fruits = ["apple", "banana", "cherry"]
2  for fruit in fruits:
3      print(fruit)
4
```



The screenshot shows a code editor with three tabs: environment.yml, guessing\_game.py, and flow\_controls.py. The flow\_controls.py tab is active, displaying the following Python code:

```
1 fruits = ["apple", "banana", "cherry"]
2 for fruit in fruits:
3     fruit = fruit + ", ok"
4     print(fruit)
5
6
7 message = "Hello"
8 for char in message:
9     print(char)
10
11 for i in range(5):
12     print(i)
13
```

The left sidebar shows a file explorer with the following files: WEEK06, .gitignore, environment.yml, flow\_controls.py, guessing\_game.py, LICENSE, and README.md. The bottom status bar indicates the current file is 'main', line 13, column 1, using UTF-8 encoding with CRLF line endings, and the Python interpreter is 3.12.7 (base: conda).

The screenshot shows a terminal window with the following output:

```
(week06)
16386@MINGW64 ~/repo/week06 (main)
$ python flow_controls.py
apple, ok
banana, ok
cherry, ok
H
e
l
l
o
(week06)
16386@MINGW64 ~/repo/week06 (main)
$ python flow_controls.py
apple, ok
banana, ok
cherry, ok
H
e
l
l
o
0
1
2
3
4
(week06)
16386@MINGW64 ~/repo/week06 (main)
$
```

The terminal window title is 'MINGW64/c:/Users/16386/rep'. The bottom status bar indicates the current file is 'main', line 13, column 1, using UTF-8 encoding with CRLF line endings, and the Python interpreter is 3.12.7 (base: conda).



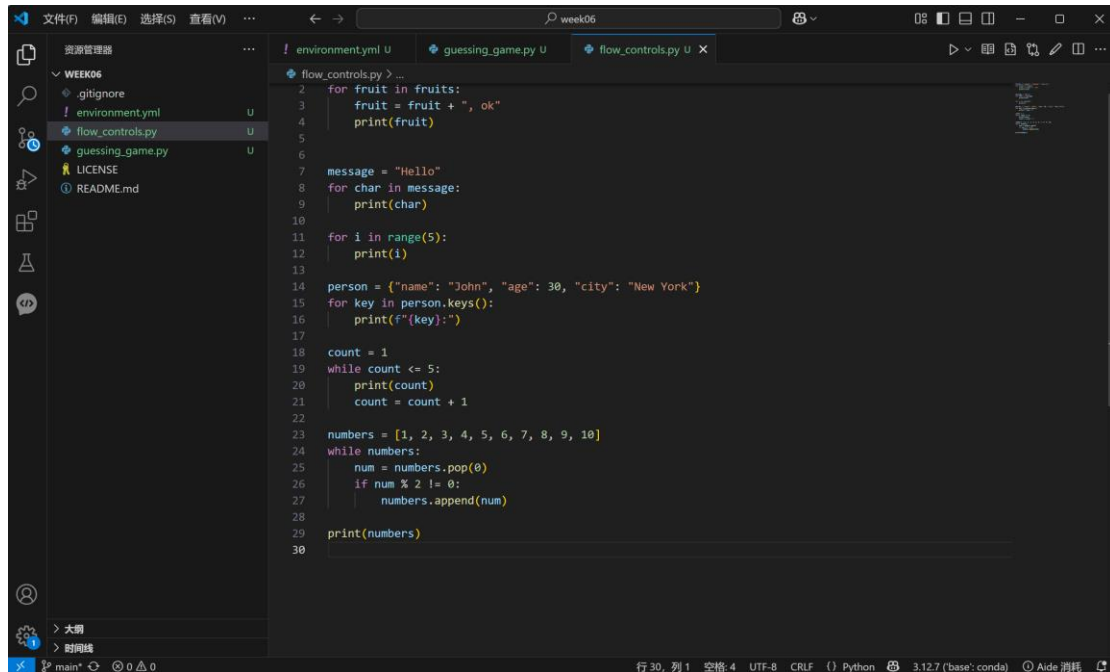
The screenshot shows a code editor with a dark theme. The left sidebar displays a file explorer for a project named 'week06'. The files listed are .gitignore, environment.yml, flow\_controls.py, guessing\_game.py, LICENSE, and README.md. The 'flow\_controls.py' file is selected and open in the main editor. The code in the editor is as follows:

```
1 fruits = ["apple", "banana", "cherry"]
2 for fruit in fruits:
3     fruit = fruit + ", ok"
4     print(fruit)
5
6
7 message = "Hello"
8 for char in message:
9     print(char)
10
11 for i in range(5):
12     print(i)
13
14 person = {"name": "John", "age": 30, "city": "New York"}
15 for key in person.keys():
16     print(f"{key}:")
17
18 count = 1
19 while count <= 5:
20     print(count)
21     count = count + 1
22
```

The status bar at the bottom indicates the current line and column (行 1, 列 39), the file encoding (UTF-8), the line ending (CRLF), the Python version (3.12.7), and the active environment (base: conda).

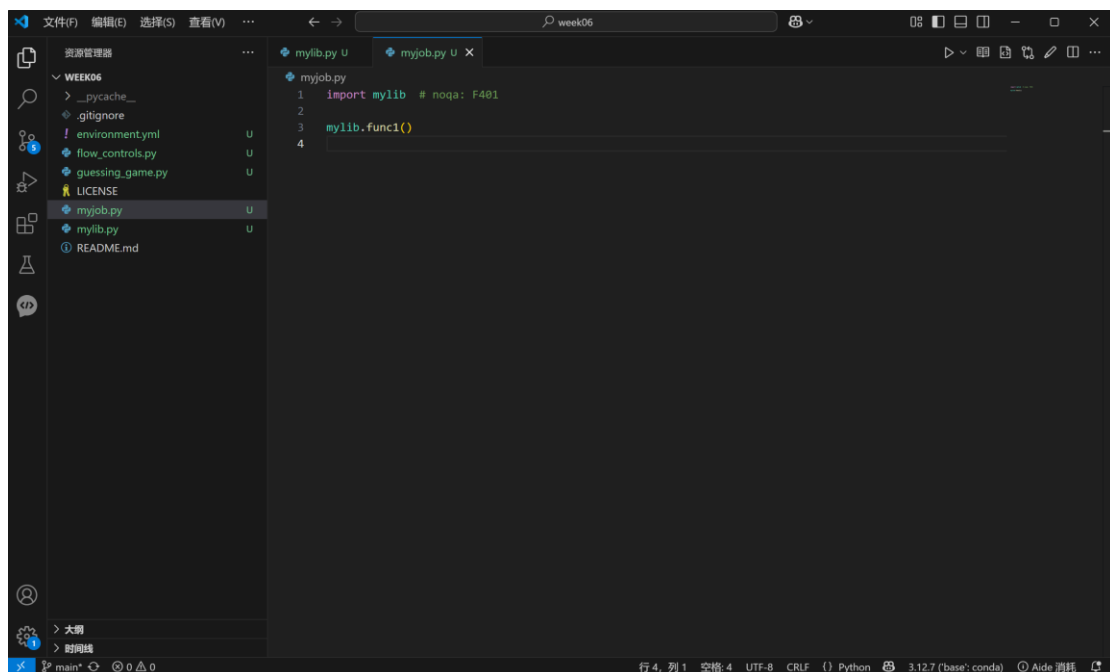
The screenshot shows a terminal window with the following output:

```
name:
age:
city:
(week06)
16386@MINGW64 ~/repo/week06 (main)
$ python flow_controls.py
apple, ok
banana, ok
cherry, ok
Hello
1
2
3
4
name:
age:
city:
1
2
3
4
5
(week06)
16386@MINGW64 ~/repo/week06 (main)
$
```



```
1 #!/usr/bin/env python
2 for fruit in fruits:
3     fruit = fruit + ", ok"
4     print(fruit)
5
6
7 message = "Hello"
8 for char in message:
9     print(char)
10
11 for i in range(5):
12     print(i)
13
14 person = {"name": "John", "age": 30, "city": "New York"}
15 for key in person.keys():
16     print(f"{key}:")
17
18 count = 1
19 while count <= 5:
20     print(count)
21     count = count + 1
22
23 numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
24 while numbers:
25     num = numbers.pop(0)
26     if num % 2 != 0:
27         numbers.append(num)
28
29 print(numbers)
30
```

3、创建 mylib.py 模块,再创建 myjob.py 脚本, 从 mylib.py 导入函数并尝试调用  
定义函数 func1, 没有形参, 没有返回值



```
1 import mylib # noqa: F401
2
3 mylib.func1()
4
```

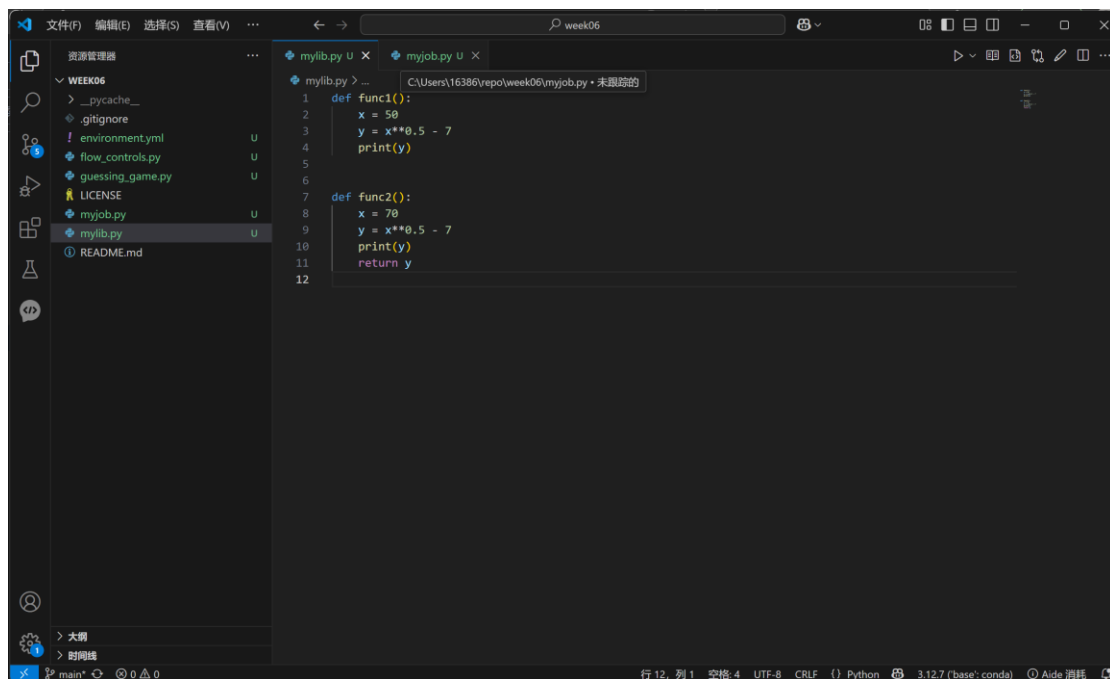
```
MINGW64/c/Users/16386/rep x + v
$ python -m pdb myjob.py
> c:\users\16386\repo\week06\myjob.py(1)<module>()
-> import mylib # noqa: F401
(Pdb) n
> c:\users\16386\repo\week06\myjob.py(3)<module>()
-> breakpoint()
(Pdb) l
  1     import mylib # noqa: F401
  2
  3     -> breakpoint()
[EOF]
(Pdb) p mylib
<module 'mylib' from 'C:\\Users\\16386\\repo\\week06\\mylib.py'>
(Pdb) import wat
(Pdb) wat / mylib

value: <module 'mylib' from 'C:\\Users\\16386\\repo\\week06\\mylib.py'>
type: module

Public attributes:
  def func1()

(Pdb) q
16386@  MINGW64 ~/repo/week06 (main)
$ python myjob.py
0.0710678118654755
16386@  MINGW64 ~/repo/week06 (main)
$
```

定义函数 func2，没有形参，有返回值



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a directory structure with files like `__pycache__`, `.gitignore`, `environment.yml`, `flow_controls.py`, `guessing_game.py`, `LICENSE`, `myjob.py`, `mylib.py`, and `README.md`. The code editor shows the following code:

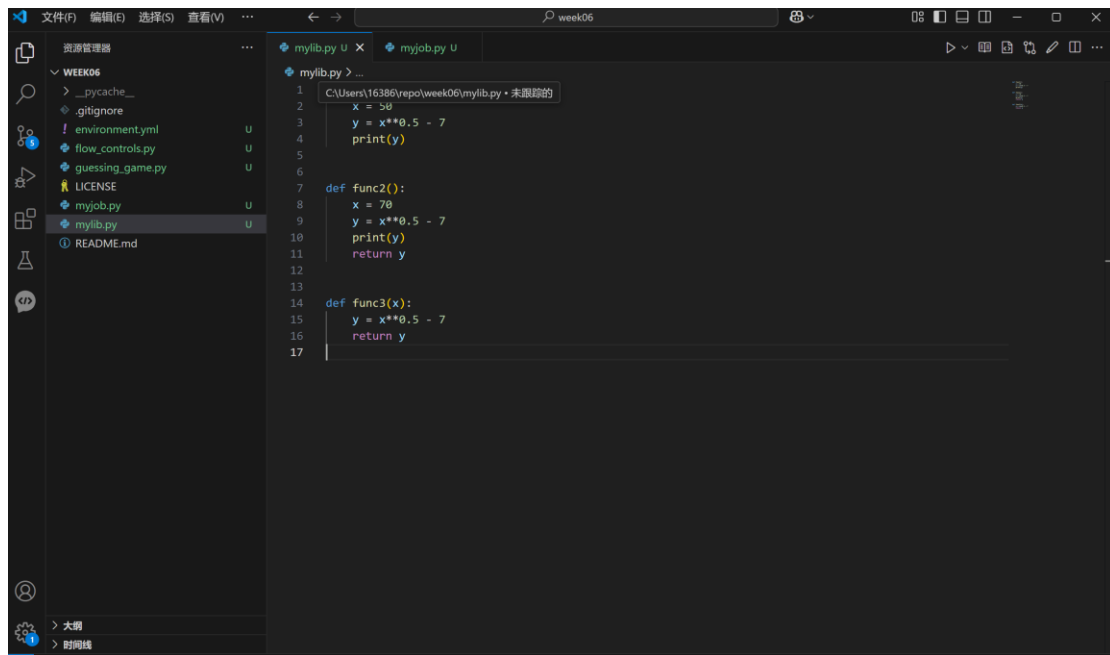
```
1 def func1():
2     x = 50
3     y = x**0.5 - 7
4     print(y)
5
6
7 def func2():
8     x = 70
9     y = x**0.5 - 7
10    print(y)
11    return y
12
```



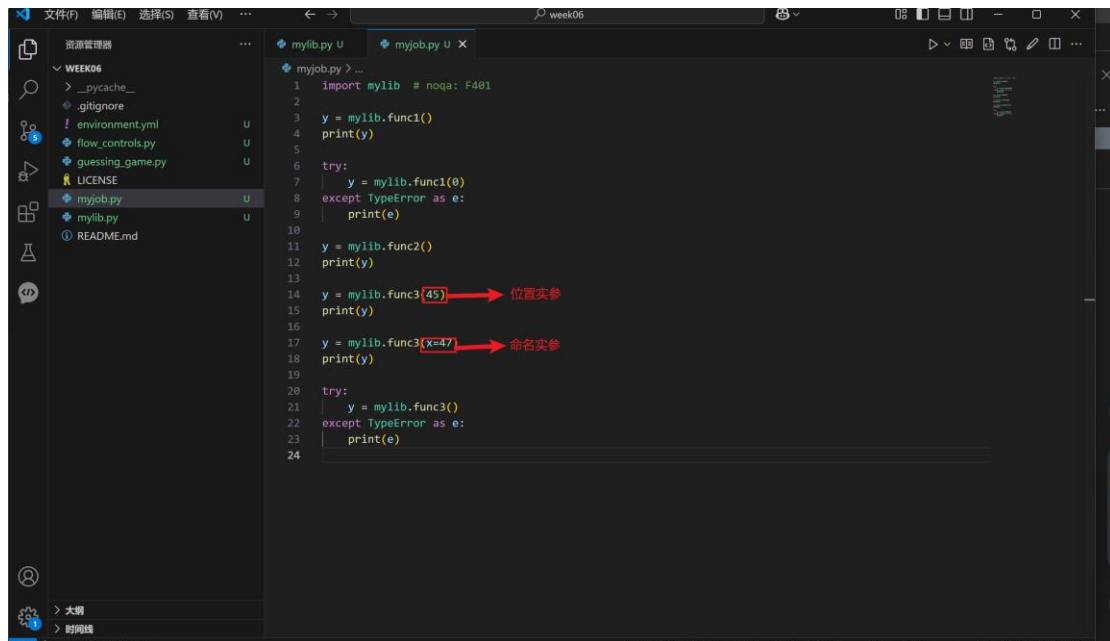
```
1 import mylib # noqa: F401
2
3 y = mylib.func1()
4 print(y)
5
6 try:
7     y = mylib.func1(0)
8 except TypeError as e:
9     print(e)
10
11 y = mylib.func2()
12 print(y)
13
```

```
None
Traceback (most recent call last):
  File "C:\Users\16386\repo\week06\myjob.py", line 6, in <module>
    y = mylib.func1(0)
    ^^^^^^^^^^^^^^^^^
TypeError: func1() takes 0 positional arguments but 1 was given
(week06)
16386@00 000000 MINGW64 ~/repo/week06 (main)
$ python myjob.py
File "C:\Users\16386\repo\week06\myjob.py", line 8
    except TypeError as e:
                      ^
SyntaxError: invalid character ':' (U+FF1A)
(week06)
16386@00 000000 MINGW64 ~/repo/week06 (main)
$ python myjob.py
0.0710678118654755
None
func1() takes 0 positional arguments but 1 was given
(week06)
16386@00 000000 MINGW64 ~/repo/week06 (main)
$ python myjob.py
0.0710678118654755
None
func1() takes 0 positional arguments but 1 was given
1.3666002653407556
1.3666002653407556
(week06)
16386@00 000000 MINGW64 ~/repo/week06 (main)
$
```

定义函数 func3, 只有一个位置形参, 先尝试传入位置实参调用, 再尝试传入命名实参调用, 再尝试不传实参(会报错)

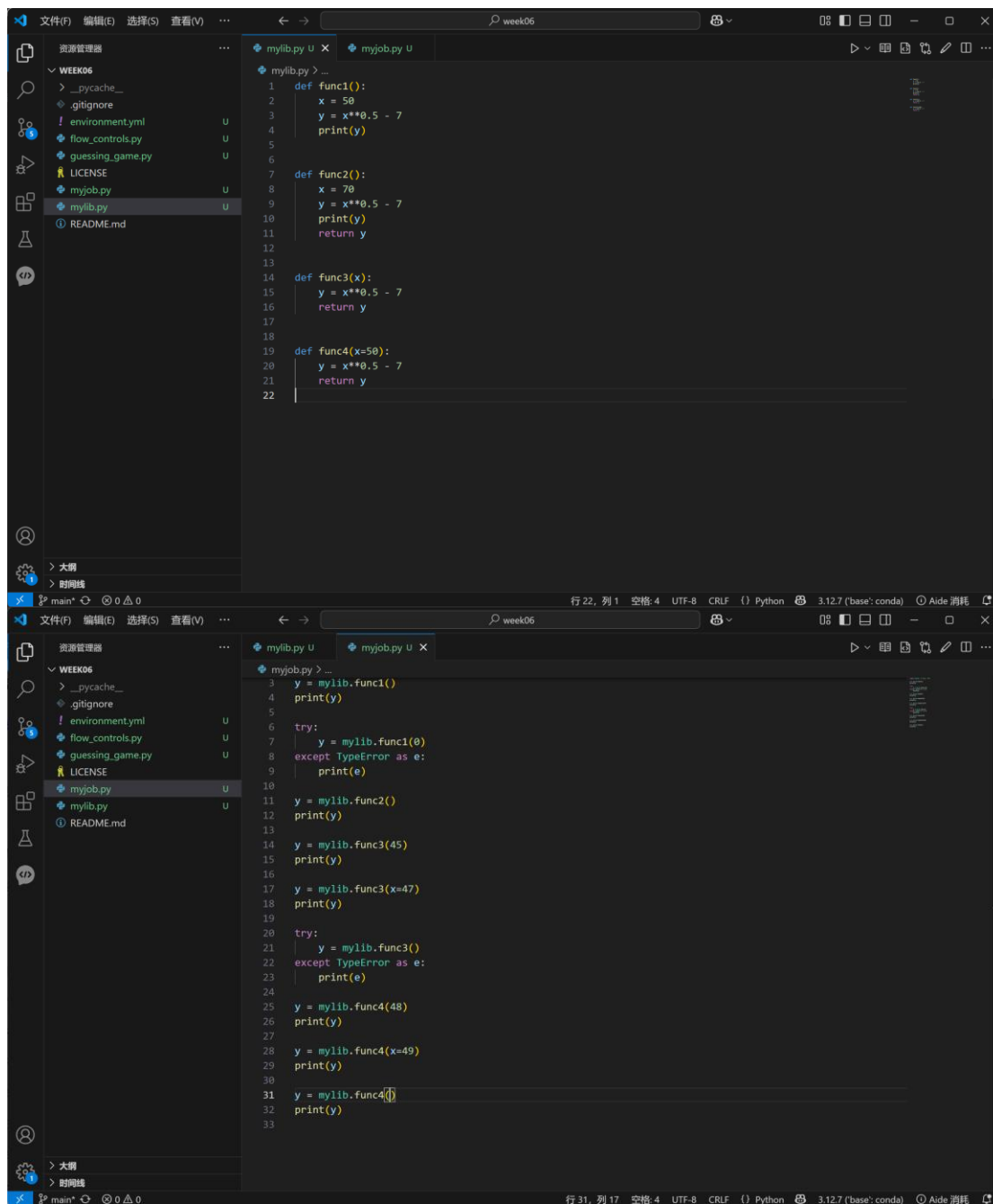


```
1  mylib.py > ...
2  C:\Users\16386\repo\week06\mylib.py · 未跟踪的
3
4  x = 50
5  y = x**0.5 - 7
6  print(y)
7
8  def func2():
9      x = 70
10     y = x**0.5 - 7
11     print(y)
12     return y
13
14  def func3(x):
15     y = x**0.5 - 7
16     return y
17
```



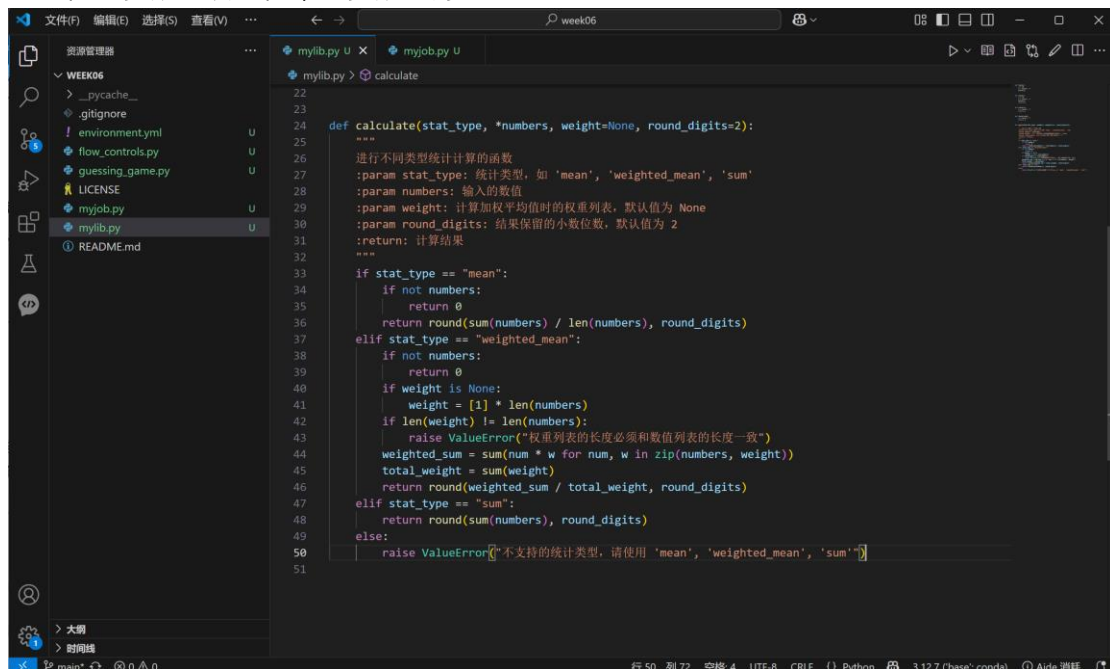
```
1  myjob.py > ...
2  import mylib # noqa: F401
3
4  y = mylib.func1()
5  print(y)
6
7  try:
8      y = mylib.func1(0)
9      print(y)
10
11  y = mylib.func2()
12  print(y)
13
14  y = mylib.func3(45) → 位置实参
15  print(y)
16
17  y = mylib.func3(x=47) → 命名实参
18  print(y)
19
20  try:
21      y = mylib.func3()
22      print(y)
23  except TypeError as e:
24      print(e)
```

定义函数 func4，只有一个命名形参，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）

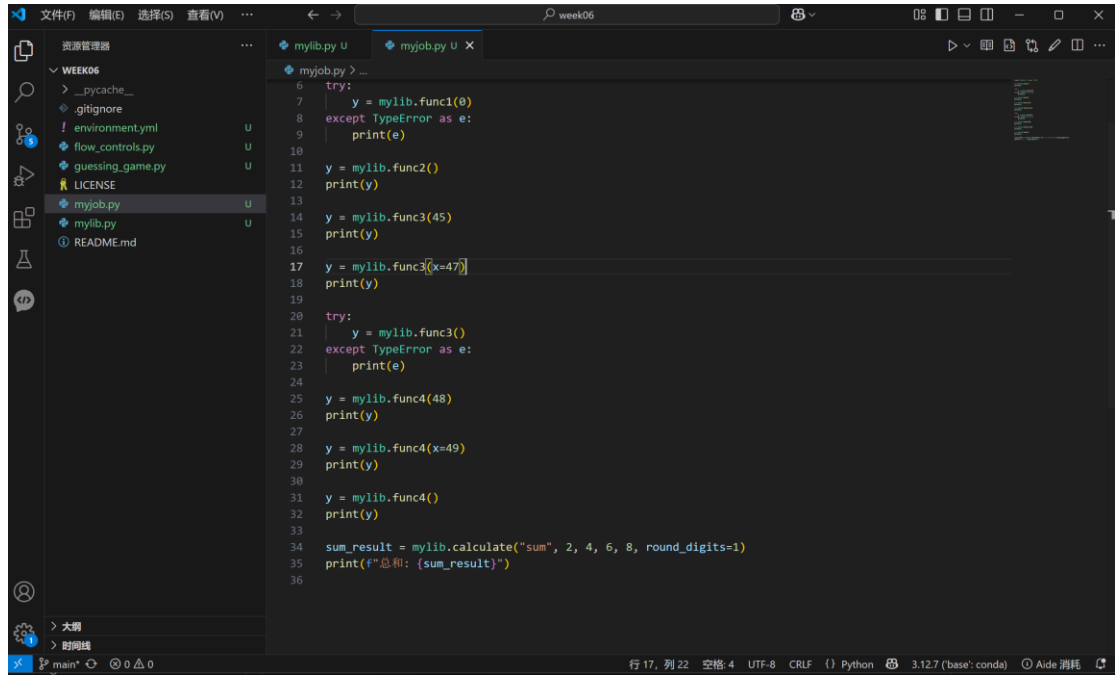


```
MINGW64: c:/Users/16386/rep X + v
(week06)
16386@MINGW64 ~/repo/week06 (main)
$ python myjob.py
0.0710678118654755
None
func1() takes 0 positional arguments but 1 was given
1.3666002653407556
1.3666002653407556
-0.2917960675006306
-0.1443453995989561
func3() missing 1 required positional argument: 'x'
-0.07179676972449123
0.0
(week06)
16386@MINGW64 ~/repo/week06 (main)
$ python myjob.py
0.0710678118654755
None
func1() takes 0 positional arguments but 1 was given
1.3666002653407556
1.3666002653407556
-0.2917960675006306
-0.1443453995989561
func3() missing 1 required positional argument: 'x'
-0.07179676972449123
0.0
0.0710678118654755
(week06)
16386@MINGW64 ~/repo/week06 (main)
$
```

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

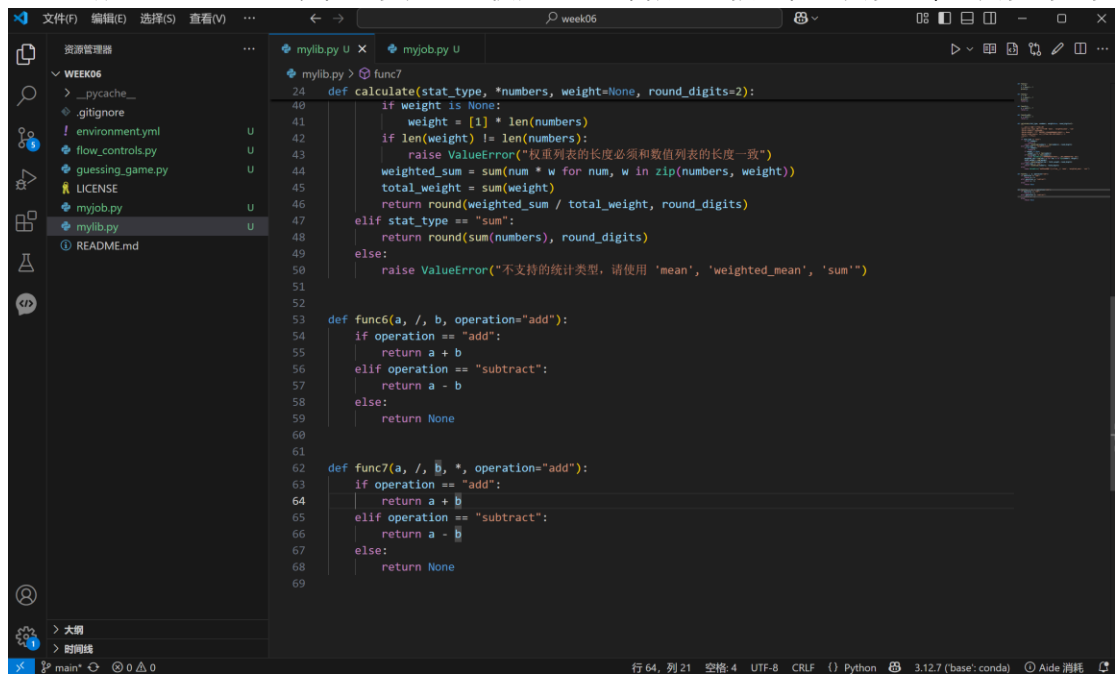


```
22
23
24 def calculate(stat_type, *numbers, weight=None, round_digits=2):
25     """
26     进行不同类型统计计算的函数
27     :param stat_type: 统计类型, 如 'mean', 'weighted_mean', 'sum'
28     :param numbers: 输入的数值
29     :param weight: 计算加权平均值时的权重列表, 默认值为 None
30     :param round_digits: 结果保留的小数位数, 默认值为 2
31     :return: 计算结果
32     """
33     if stat_type == "mean":
34         if not numbers:
35             return 0
36         return round(sum(numbers) / len(numbers), round_digits)
37     elif stat_type == "weighted_mean":
38         if not numbers:
39             return 0
40         if weight is None:
41             weight = [1] * len(numbers)
42         if len(weight) != len(numbers):
43             raise ValueError("权重列表的长度必须和数值列表的长度一致")
44         weighted_sum = sum(num * w for num, w in zip(numbers, weight))
45         total_weight = sum(weight)
46         return round(weighted_sum / total_weight, round_digits)
47     elif stat_type == "sum":
48         return round(sum(numbers), round_digits)
49     else:
50         raise ValueError(f"不支持的统计类型, 请使用 'mean', 'weighted_mean', 'sum'")
51
```



```
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 、func7，在形参列表中使用 / 、\*来限定只接受位置实参、命名实参的形参



```
24 def calculate(stat_type, *numbers, weight=None, round_digits=2):
40     if weight is None:
41         weight = [1] * len(numbers)
42     if len(weight) != len(numbers):
43         raise ValueError("权重列表的长度必须和数值列表的长度一致")
44     weighted_sum = sum(num * w for num, w in zip(numbers, weight))
45     total_weight = sum(weight)
46     return round(weighted_sum / total_weight, round_digits)
47
48 elif stat_type == "sum":
49     return round(sum(numbers), round_digits)
50 else:
51     raise ValueError("不支持的统计类型，请使用 'mean', 'weighted_mean', 'sum'")
52
53 def func6(a, /, b, operation="add"):
54     if operation == "add":
55         return a + b
56     elif operation == "subtract":
57         return a - b
58     else:
59         return None
60
61
62 def func7(a, /, b, *, operation="add"):
63     if operation == "add":
64         return a + b
65     elif operation == "subtract":
66         return a - b
67     else:
68         return None
69
```

The image shows a code editor window with a file explorer on the left and a terminal window at the bottom. The code editor displays a Python file named `myjob.py` with the following content:

```
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 try:
37     print(mylib.func6(a=10, b=5))
38 except TypeError as e:
39     print(e)
40
41 print(mylib.func7(10, 5, operation="subtract"))
42
```

The terminal window shows the output of running `python myjob.py` in a MINGW64 environment. The output is as follows:

```
1.3666002653407556
-0.2917960675006306
-0.1443453995989561
func3() missing 1 required positional argument: 'x'
-0.07179676972449123
0.0
0.0710678118654755
总和: 20
func6() got some positional-only arguments passed as keyword arguments: 'a'
None
(week06)
16386000 MINGW64 ~/repo/week06 (main)
$ python myjob.py
0.0710678118654755
None
func1() takes 0 positional arguments but 1 was given
1.3666002653407556
1.3666002653407556
-0.2917960675006306
-0.1443453995989561
func3() missing 1 required positional argument: 'x'
-0.07179676972449123
0.0
0.0710678118654755
总和: 20
func6() got some positional-only arguments passed as keyword arguments: 'a'
5
(week06)
16386000 MINGW64 ~/repo/week06 (main)
$
```

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

```
24 def calculate(stat_type, *numbers, weight=None, round_digits=2):
25     return round(weighted_sum / total_weight, round_digits)
26
27 elif stat_type == "sum":
28     return round(sum(numbers), round_digits)
29 else:
30     raise ValueError("不支持的统计类型, 请使用 'mean', 'weighted_mean', 'sum'")
31
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
50
51 def func8(*numbers):
52     total = 0
53     for num in numbers:
54         total = total + num
55     return total
```

行 76, 列 1 空格 4 UTF-8 CRLF {} Python 3.12.7 ('base: conda') Aide 消耗

```
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
37 try:
38     print(mylib.func6(a=10, b=5))
39 except TypeError as e:
40     print(e)
41
42 print(mylib.func7(10, 5, operation="subtract"))
43
44 print(mylib.func8(4, 8))
```

行 43, 列 23 空格 4 UTF-8 CRLF {} Python 3.12.7 ('base: conda') Aide 消耗





```
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 try:
37     print(mylib.func6(a=10, b=5))
38 except TypeError as e:
39     print(e)
40
41 print(mylib.func7(10, 5, operation="subtract"))
42
43 print(mylib.func8(4, 8))
44
45 mylib.func9(name="Alice", age=25, city="New York")
46
```

```
0.0
0.0710678118654755
总和: 20
func6() got some positional-only arguments passed as keyword arguments: 'a'
5
12
(week06)
16386@MINGW64 ~/repo/week06 (main)
$ python myjob.py
0.0710678118654755
None
func1() takes 0 positional arguments but 1 was given
1.3666002653407556
1.3666002653407556
-0.2917960675006306
-0.1443453995989561
func3() missing 1 required positional argument: 'x'
-0.07179676972449123
0.0
0.0710678118654755
总和: 20
func6() got some positional-only arguments passed as keyword arguments: 'a'
5
12
name: Alice
age: 25
city: New York
(week06)
16386@MINGW64 ~/repo/week06 (main)
$
```

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 \* 将可迭代对象自动解包，按位置实参传入

```
53 def func6(a, /, b, operation="add"):  
54     return None  
55  
56 def func7(a, /, b, *, operation="add"):  
57     if operation == "add":  
58         return a + b  
59     elif operation == "subtract":  
60         return a - b  
61     else:  
62         return None  
63  
64 def func8(*numbers):  
65     total = 0  
66     for num in numbers:  
67         total = total + num  
68     return total  
69  
70 def func9(**user):  
71     for key, value in user.items():  
72         print(f"{key}: {value}")  
73  
74 def func10(arg1, arg2, named_arg="default"):  
75     print(f"位置实参 arg1: {arg1}")  
76     print(f"位置实参 arg2: {arg2}")  
77     print(f"命名实参 named_arg: {named_arg}")  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87
```

```
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 try:  
37     print(mylib.func6(a=10, b=5))  
38 except TypeError as e:  
39     print(e)  
40  
41 print(mylib.func7(10, 5, operation="subtract"))  
42  
43 print(mylib.func8(4, 8))  
44  
45 mylib.func9(name="Alice", age=25, city="New York")  
46  
47 tuple_args = (3, 5)  
48 mylib.func10(*tuple_args)  
49
```

定义函数 func12，给函数添加内嵌文档，给形参和返回值添加类型注解

```
def func7(a, /, b, *, operation="add"):
    return a + b
    elif operation == "subtract":
        return a - b
    else:
        return None

def func8(*numbers):
    total = 0
    for num in numbers:
        total = total + num
    return total

def func9(**user):
    for key, value in user.items():
        print(f"{key}: {value}")

def func10(arg1, arg2, named_arg="default"):
    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}")

def func12(arg1: str, arg2: int, named_arg: str = "default") -> None:
    """哈哈"""
    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}")
```

```
except TypeError as e:
    print(e)

y = mylib.func4(48)
print(y)

y = mylib.func4(x=49)
print(y)

y = mylib.func4()
print(y)

sum_result = mylib.calculate("sum", 2, 4, 6, 8, round_digits=1)
print(f"总和: {sum_result}")
try:
    print(mylib.func6(a=10, b=5))
except TypeError as e:
    print(e)

print(mylib.func7(10, 5, operation="subtract"))

print(mylib.func8(4, 8))

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

tuple_args = (3, 5)
mylib.func10(*tuple_args)

mylib.func12(7, 8, 9)
```

4、把 mylib 模块转变为软件包安装进当前的 Conda 环境

