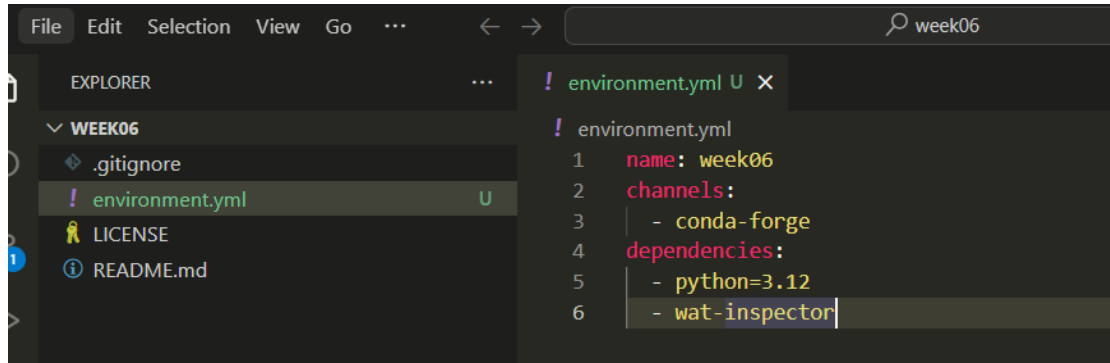
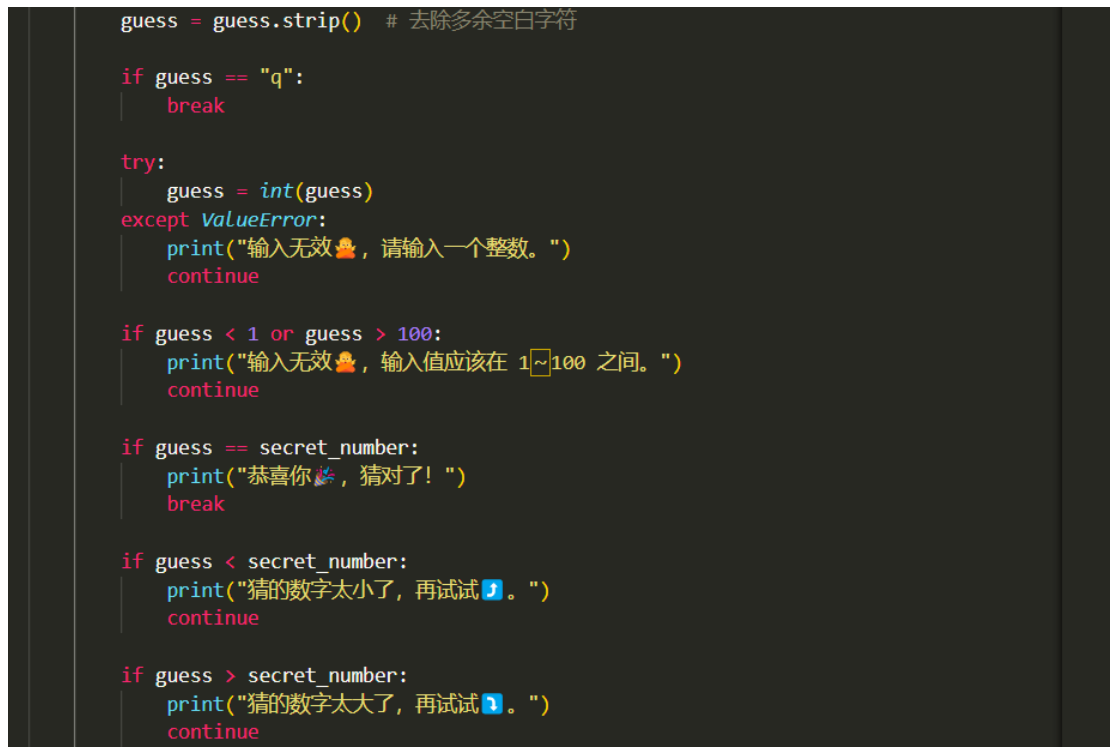


第六周学习报告

● 复制到 week06



● 运行程序



● 例子

python for 语句 例子

在 Python 里, `for` 语句可用于对序列 (像列表、元组、字符串) 或者其他可迭代对象开展迭代操作。下面是几种不同的 `for` 语句示例:

1. 遍历列表

```
python ^

fruits = ["apple", "banana", "cherry"]
for fruit in fruits:
    print(fruit)
```

2. 遍历字符串

```
(base) tmk@LAPTOP-GTIMJUPO MINGW64 ~/repo/week06 (main)
$ python flow_controls.py
apple, ok
banana, ok
cherry, ok
h
e
l
l
o
0
1
2
3
4
name: John
age: 30
city: New York
(0, 0)
(0, 1)
(1, 0)
(1, 1)
(2, 0)
(2, 1)
```

1. 简单计数循环

此例中，`while` 循环从 0 开始计数，直到 `i` 不再小于 5 时停止。

```
python ^
i = 0
while i < 5:
    print(i)
    i = i + 1
```

2. 条件控制循环

该例子通过 `while` 循环不断要求用户输入，直到用户输入 "quit" 为止。

```
python ^
user_input = ""
while user_input != "quit":
```

```
1
2
3
4
5
1 到 10 的和是：55
```

- 函数，在模板里导入写函数

- **mylib.py 模块**
- 此模块会定义几个函数，之后在 **myjob.py** 脚本里会导入并调用这些函数。
- **myjob.py 脚本**
- 该脚本会从 **mylib.py** 模块导入函数，并且调用这些函数。

位置形参、命名形参

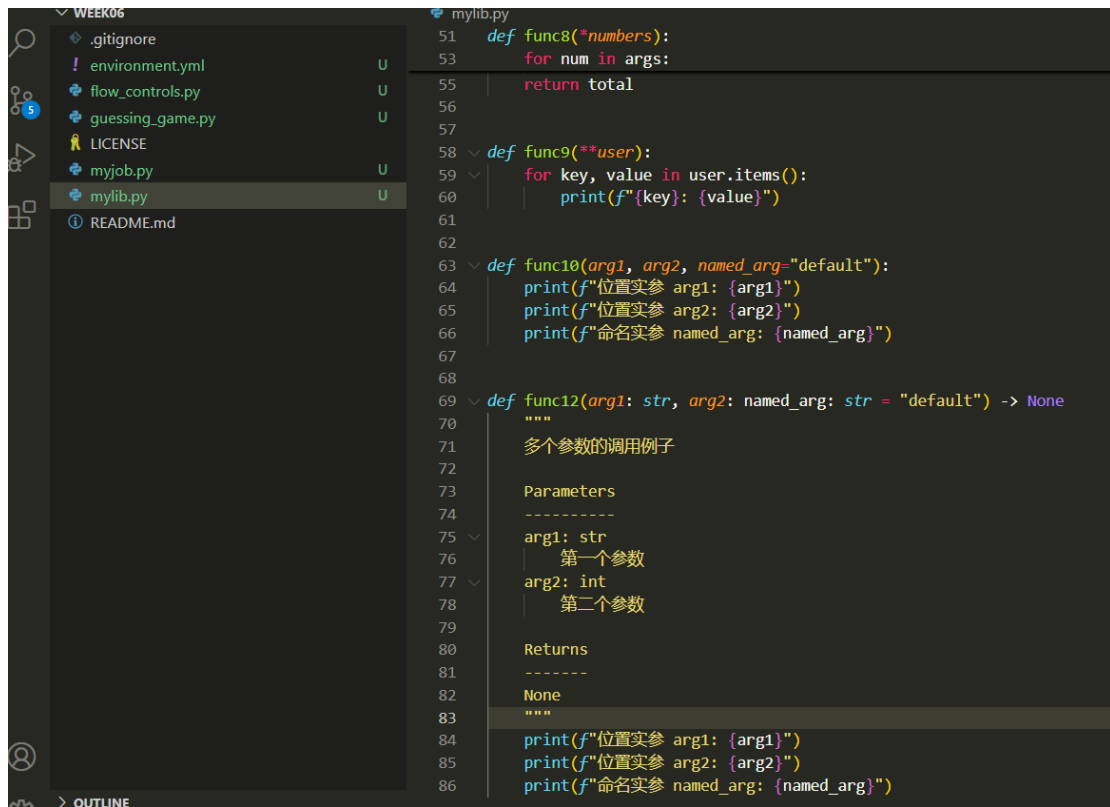
```
myjob.py
3  y = mylib.func1()
4  print(y)
5
6  try:
7      y = mylib.func1(0)
8  except TypeError as e:
9      print(e)
10
11 y = mylib.func2()
12 print(y)
13
14 y = mylib.func3(45)
15 print(y)
16
17 y = mylib.func3(x=47)
18 print(y)
19
20 try:
21     y = mylib.func3()
22 except TypeError as e:
23     print(e)
24 try:
25     mylib.func3(y=47)
26 except TypeError as e:
27     print(e)
28
29 y = mylib.func4(48)
30 print(y)
31
32 y = mylib.func4(49)
33 print(y)]
```

在 Python 里，函数调用时的参数可分为位置形参、命名形参、位置实参和命名实参。下面结合之前的 `mylib.py` 模块，通过示例来详细解释这些概念。

位置形参和位置实参

位置形参指的是在函数定义时按照位置顺序列出的参数，而位置实参则是在调用函数时按照位置顺序传递给函数的参数。

```
python ^  
  
# mylib.py 模块  
def add_numbers(a, b): # a 和 b 是位置形参  
    return a + b  
  
# myjob.py 脚本  
from mylib import add_numbers
```



```
mylib.py  
51 def func8(*numbers):  
53     for num in args:  
55         return total  
56  
57  
58 def func9(**user):  
59     for key, value in user.items():  
60         print(f"{key}: {value}")  
61  
62  
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 func12(arg1: str, arg2: named_arg: str = "default") -> None  
70     """  
71     多个参数的调用例子  
72  
73     Parameters  
74     -----  
75     arg1: str  
76         第一个参数  
77     arg2: int  
78         第二个参数  
79  
80     Returns  
81     -----  
82     None  
83     """  
84     print(f"位置实参 arg1: {arg1}")  
85     print(f"位置实参 arg2: {arg2}")  
86     print(f"命名实参 named_arg: {named_arg}")
```

