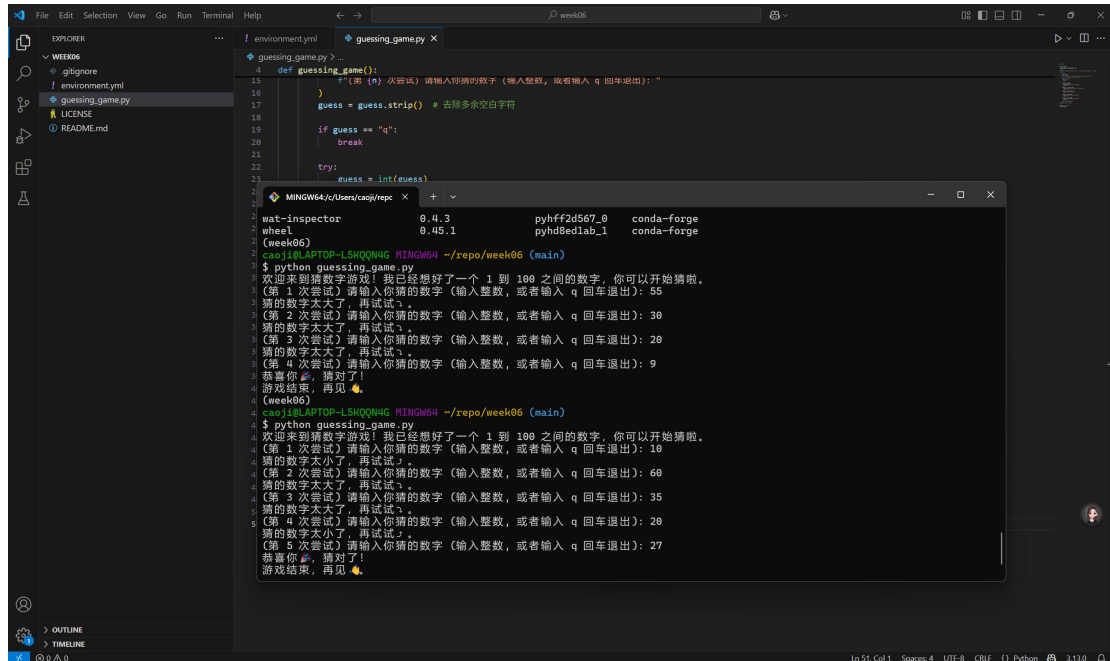


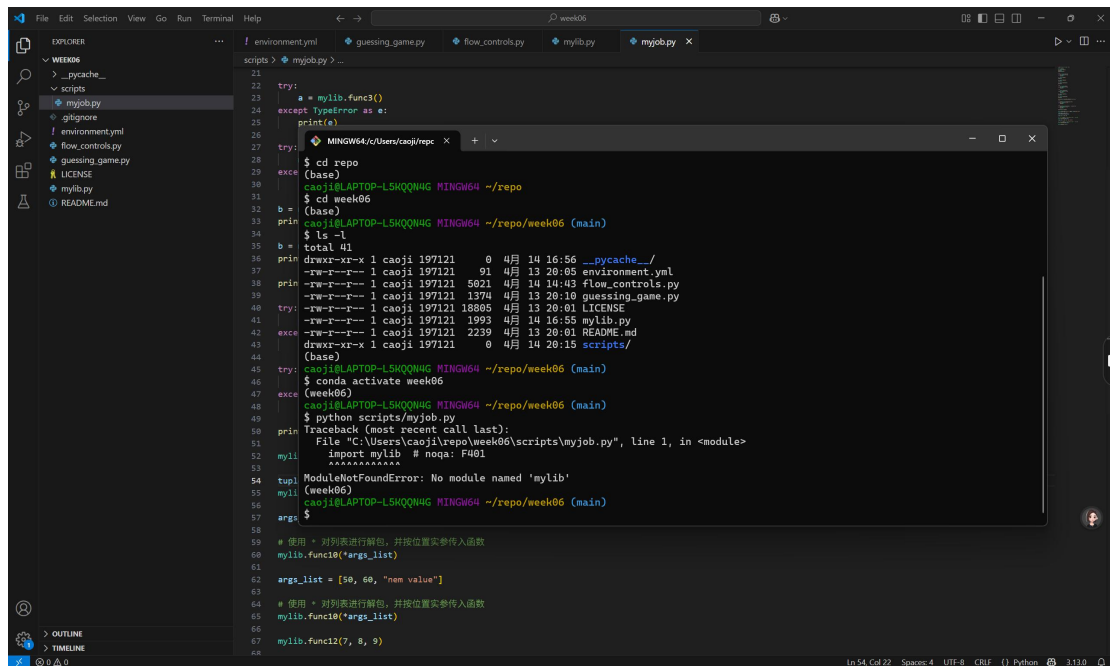
第六周学习笔记



The screenshot shows a VS Code editor with a file explorer on the left containing files like `environment.yml`, `guessing_game.py`, `gignore`, `LICENSE`, and `README.md`. The main editor displays the `guessing_game.py` script, which defines a `guessing_game()` function. The function prompts the user to guess a number between 1 and 100, checks if the guess is correct, too high, or too low, and allows the user to quit with 'q'. A terminal window below the editor shows the command `python guessing_game.py` being executed, with the game's output displayed in real-time.

```
def guessing_game():
    """(in {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): """
    guess = guess.strip() # 去除多余空白字符
    if guess == "q":
        break
    try:
        guess = int(guess)
        # ... (rest of the function logic) ...
```

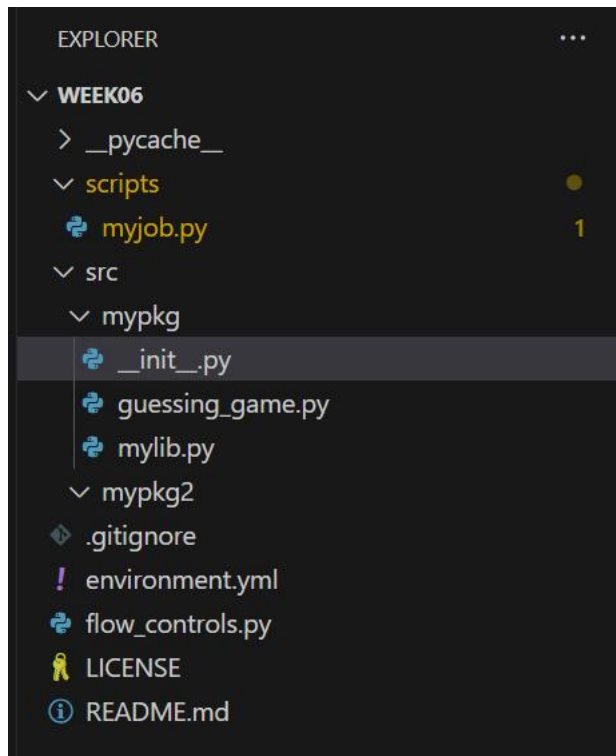
复制老师的代码进行运行，这个游戏还蛮有意思。



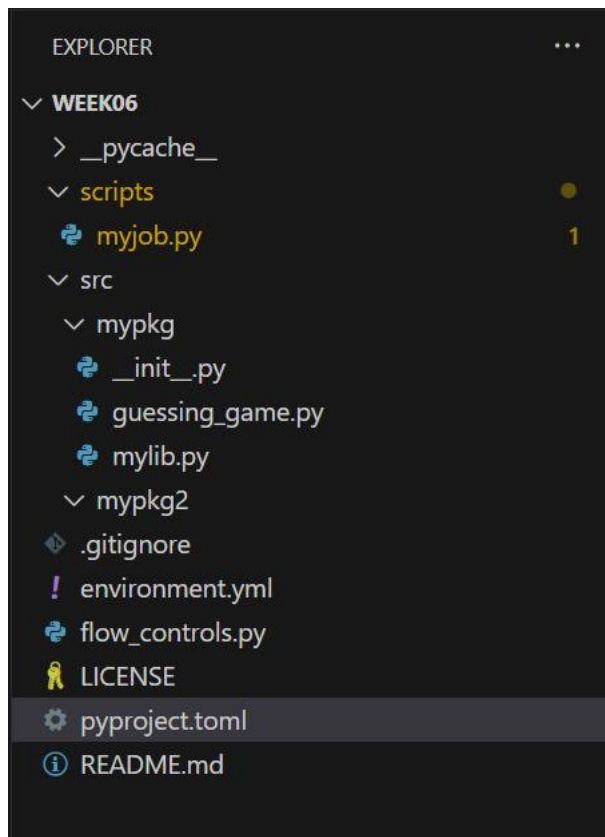
The screenshot shows a VS Code editor with a file explorer on the left containing files like `environment.yml`, `flow_controls.py`, `mylib.py`, and `myjob.py`. The main editor displays the `myjob.py` script, which contains a `main` function that performs various system operations, including directory navigation, file listing, and package management. A terminal window below the editor shows the command `python scripts/myjob.py` being executed, with the script's output displayed in real-time.

```
def main():
    """Main function"""
    # ... (rest of the main function logic) ...
    # Example output:
    # $ cd repo
    # (base)
    # $ cd week06
    # (base)
    # $ ls -l
    # total 41
    # drwxr-xr-x 1 caoji 197121 0 4月 14 16:56 __pycache__/_
    # -rw-r--r-- 1 caoji 197121 91 4月 13 20:05 environment.yml
    # -rw-r--r-- 1 caoji 197121 5021 4月 14 19:43 flow_controls.py
    # -rw-r--r-- 1 caoji 197121 1374 4月 13 20:10 guessing_game.py
    # -rw-r--r-- 1 caoji 197121 18805 4月 13 20:01 LICENSE
    # -rw-r--r-- 1 caoji 197121 1993 4月 14 16:55 mylib.py
    # -rw-r--r-- 1 caoji 197121 2239 4月 13 20:01 README.md
    # drwxr-xr-x 1 caoji 197121 0 4月 14 20:15 scripts/
    # (base)
    # $ conda activate week06
    # (week06)
    # $ python scripts/myjob.py
    # Feedback (most recent call last):
    # File "C:\Users\caoji\repo\week06\scripts\myjob.py", line 1, in <module>
    # import mylib # noqa: F401
    # ModuleNotFoundError: No module named 'mylib'
    # (week06)
    # $ python scripts/myjob.py
    # mylib.func1(7, 8, 9)
```

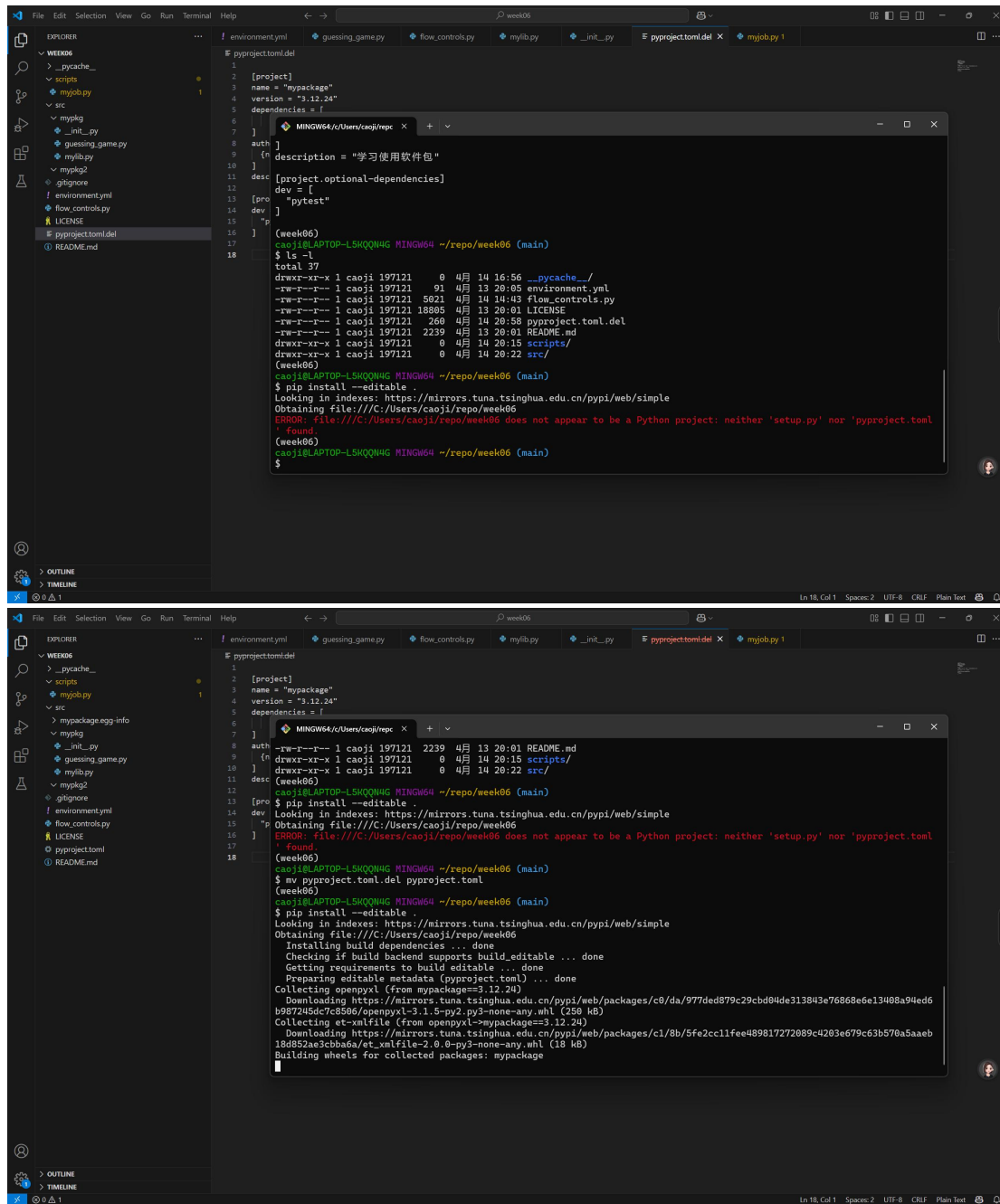
因为我们自己写的软件包并没有上传到平台，也算是第三方软件包，也需要安装。



有了`__init__.py`之后，这个文件可以无内容，但只要有了这个文件，python 就会自动识别其为软件包。



需要一个配置文件，且命名一定要是 `pyproject.toml`，且需要在主目录下创建。



更改完文件名即可

The image shows a VS Code editor with a file explorer on the left and a terminal window on the right. The file explorer shows a project named 'week06' with a folder 'scripts' containing 'myjob.py'. The terminal window shows the output of running 'myjob.py'. The script defines a class 'mylib' with methods 'func1', 'func2', 'func3', 'func4', 'func5', 'func6', 'func7', and 'calculate_total'. It then creates an instance 'Alice' and calls these methods with various arguments, including positional, keyword, and default values. The output shows the results of these calls, including the calculation of a total value.

```
1 from mypkg import mylib # noqa: F401
2
3 mylib.func1()
4 y = mylib.func1()
5 print(y)
6 print
7
8 try:
9     15.0
10    15.0
11 except:
12     32
13
14 func3() missing 1 required positional argument: 'a'
15 y = func3() got an unexpected keyword argument 'b'
16 print 7.714285714285714
17 8.714285714285715
18 y = 45.0
19 print func6() got some positional-only arguments passed as keyword arguments: 'a'
20 func7() takes 1 positional argument but 3 were given
21 print name: Alice
22 age: 25
23 try:
24     city: New York
25     位置形参 arg1 的值为: 10
26     位置形参 arg2 的值为: 20
27     命名形参 named_arg 的值为: default_value
28     位置形参 arg1 的值为: 30
29     位置形参 arg2 的值为: 40
30     命名形参 named_arg 的值为: default_value
31     位置形参 arg1 的值为: 50
32     位置形参 arg2 的值为: 60
33     命名形参 named_arg 的值为: nem value
34     b = 位置形参 arg1 的值为: 7
35     位置形参 arg2 的值为: 8
36     命名形参 named_arg 的值为: 9
37 b = (week06)
38 print caojieLAPTOP-L5WQNW4G MINGW64 ~/repo/week06 (main)
39 $
40 print(mylib.calculate_total(10, 5, 0.1))
41
42 try:
43     print(mylib.func6(a=10, b=5))
44 except TypeError as e:
45     print(e)
46
47 try:
48     print(mylib.func7(10, 5, "subtract"))
49 except TypeError as e:
50     print(e)
```