

The image shows a Visual Studio Code editor window with a Python file named `guessing_game.py` open. The file contains a function `guessing_game()` that implements a number guessing game. The game generates a random integer between 1 and 100, and the player has 10 attempts to guess it. The script includes various prompts in Chinese and feedback messages. The status bar at the bottom indicates the file is at line 48, column 20, with 4 spaces, UTF-8 encoding, and CRLF line endings. The Python version is 3.9.21.

```
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")
10
11
12     while True:
13         n += 1
14         # 获取玩家输入
15         guess = input(f"第 {n} 次尝试) 请输入你猜的数字 (输入整数,")
16         guess = guess.strip() # 去除多余空白字符
17
18         if guess == "q":
19             break
20
21         try:
22             guess = int(guess)
23         except ValueError:
24             print("输入无效🙅，请输入一个整数。")
25             continue
26
27         if guess < 1 or guess > 100:
28             print("输入无效🙅，输入值应该在 1~100 之间。")
29             continue
30
31         if guess == secret_number:
32             print("恭喜你🎉，猜对了！")
33             break
34
35         if guess < secret_number:
36             print("猜的数字太小了，再试试🔍。")
37             continue
38
39         if guess > secret_number:
40             print("猜的数字太大了，再试试🔍。")
41             continue
42
43         raise NotImplementedError
44
45     print("游戏结束，再见👋。")
46
47 if __name__ == "__main__":
48     guessing_game()
```

The image shows a Visual Studio Code editor window with a file explorer on the left and a code editor in the center. The file explorer shows a project named 'WEEK06' with files: '.gitignore', 'environment.yml', 'guessing\_game.py', 'LICENSE', and 'README.md'. The code editor displays the 'guessing\_game.py' file, which contains a Python script for a guessing game. The script generates a random number between 1 and 100 and allows the user to guess it. It includes a while loop for multiple attempts and a try-except block for handling non-integer inputs. A pdb breakpoint is set at line 16.

```
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(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): ")
15         import pdb; pdb.set_trace()
16         guess = guess.strip() # 去除多余空白字符
17
18         if guess == "q":
19             break
20
21         try:
22             guess = int(guess)
23         except ValueError:
24             print("输入无效, 请输入一个整数。")
25             continue
26
27         if guess < 1 or guess > 100:
28             print("输入无效, 输入值应该在 1-100 之间。")
```

A terminal window is open in the foreground, showing the following commands and output:

```
MINGW64/c/Users/liu/week06
$ conda env remove -n pybasic
conda env create -f environment.yml

CondaEnvironmentError: Cannot remove current environment. Deactivate and run conda remove again

CondaValueError: prefix already exists: C:\Users\liu\anaconda3\envs\pybasic

(pybasic)
liu@liujin MINGW64 ~/week06 (main)
$ conda activate pybasic
(pybasic)
liu@liujin MINGW64 ~/week06 (main)
$ python --version
Python 3.9.21
(pybasic)
liu@liujin MINGW64 ~/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): 50
> c:\users\liu\week06\guessing_game.py(16)guessing_game()
-> guess = guess.strip() # 去除多余空白字符
(Pdb) |
```

The status bar at the bottom indicates the current file is 'main\*', the cursor is at line 15, column 36, and the environment is '3.9.21 (pybasic: conda)'.

The image shows a Visual Studio Code editor window with a file explorer on the left and a code editor in the center. The file explorer shows a project named 'WEEK06' with files: '.gitignore', 'environment.yml', 'guessing\_game.py', 'LICENSE', and 'README.md'. The code editor displays the 'guessing\_game.py' file, which contains a Python script for a guessing game. The script generates a random number between 1 and 100 and allows the user to guess it. It includes a while loop for multiple attempts and a try-except block for handling non-integer inputs. A terminal window is open at the bottom, showing the execution of the script. The user has entered '25', '75', and '100', which were all incorrect. The terminal output shows the program's response to each guess and the user's input. The status bar at the bottom indicates the file is 'main\*' at line 15, column 36, with 4 spaces, UTF-8 encoding, CRLF line endings, and Python 3.9.21 (pybasic: conda) interpreter.

```
def guessing_game():
    # 生成 1 到 100 之间的随机整数
    secret_number = random.randint(1, 100)
    n = 0

    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")

    while True:
        n += 1
        # 获取玩家输入
        guess = input(f"(第 {n} 次尝试) 请输入你猜的数字（输入整数，")
        import pdb; pdb.set_trace()
        guess = guess.strip() # 去除多余空白字符

        if guess == "q":
            break

        try:
            guess = int(guess)
        except ValueError:
            print("输入无效👉，请输入一个整数。")
            continue

        if guess < 1 or guess > 100:
            print("输入无效👉，输入值应该在 1-100 之间。")

if __name__ == "__main__":
    guessing_game()
```

MINGW64: c:/Users/liu/week06

```
'25'
(Pdb) c
猜的数字太小了，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): 75
> c:\users\liu\week06\guessing_game.py(15)guessing_game()
-> import pdb; pdb.set_trace()
(Pdb) n
> c:\users\liu\week06\guessing_game.py(16)guessing_game()
-> guess = guess.strip() # 去除多余空白字符
(Pdb) p guess
'75'
(Pdb) c
猜的数字太小了，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): 100
> c:\users\liu\week06\guessing_game.py(15)guessing_game()
-> import pdb; pdb.set_trace()
(Pdb) n
> c:\users\liu\week06\guessing_game.py(16)guessing_game()
-> guess = guess.strip() # 去除多余空白字符
(Pdb) p guess
'100'
(Pdb) c
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):
```

Ln 15, Col 36 Spaces: 4 UTF-8 CRLF {} Python 3.9.21 (pybasic: conda)

The image shows a Visual Studio Code editor window with a project named 'week06'. The Explorer sidebar on the left lists files: `.gitignore`, `environment.yml`, `flow_controls.py`, `guessing_game.py`, `LICENSE`, and `README.md`. The main editor area displays the `flow_controls.py` file, which contains a Python for loop example:

```
1 # for 循环示例: 打印 1 到 5
2 for i in range(1, 6):
3     print(f"第 {i} 次循环")
4
```

Below the editor, a terminal window is open, showing the execution of the script. The terminal output includes the following lines:

```
File "C:\Users\liu\week06\guessing_game.py", line 49, in <module>
    guessing_game()
File "C:\Users\liu\week06\guessing_game.py", line 15, in guessing_game
    import pdb; pdb.set_trace()
File "C:\Users\liu\week06\guessing_game.py", line 15, in guessing_game
    import pdb; pdb.set_trace()
File "C:\Users\liu\anaconda3\envs\pybasic\lib\bdb.py", line 88, in trace_dispatch
    return self.dispatch_line(frame)
File "C:\Users\liu\anaconda3\envs\pybasic\lib\bdb.py", line 113, in dispatch_line
    if self.quitting: raise BdbQuit
bdb.BdbQuit
(pybasic)
liu@liujin MINGW64 ~/week06 (main)
$ python flow_controls.py
第 1 次循环
第 2 次循环
第 3 次循环
第 4 次循环
第 5 次循环
(pybasic)
liu@liujin MINGW64 ~/week06 (main)
$
```

The status bar at the bottom of the editor indicates the current file is `main*`, the cursor is at line 4, column 1, and the file is encoded in UTF-8 with CRLF line endings. The Python version is 3.9.21 from the 'pybasic' conda environment.

The image shows a Visual Studio Code editor window with a project named 'week06'. The Explorer sidebar on the left lists files: `.gitignore`, `environment.yml`, `flow_controls.py`, `guessing_game.py`, `LICENSE`, and `README.md`. The main editor area displays the `flow_controls.py` file with the following Python code:

```
1 # for 循环示例：打印 1 到 5
2 for i in range(1, 6):
3     print(f"第 {i} 次循环")
4 # while 循环示例：打印 1 到 5
5 n = 1
6 while n <= 5:
7     print(f"[while] 第 {n} 次循环")
8     n += 1
9
```

Below the editor, a terminal window titled 'MINGW64/c/Users/liu/week06' shows the execution of the script. The output is as follows:

```
(pybasic)
liu@liujin MINGW64 ~/week06 (main)
$ python flow_controls.py
第 1 次循环
第 2 次循环
第 3 次循环
第 4 次循环
第 5 次循环
(pybasic)
liu@liujin MINGW64 ~/week06 (main)
$ python flow_controls.py
第 1 次循环
第 2 次循环
第 3 次循环
第 4 次循环
第 5 次循环
[while] 第 1 次循环
[while] 第 2 次循环
[while] 第 3 次循环
[while] 第 4 次循环
[while] 第 5 次循环
(pybasic)
liu@liujin MINGW64 ~/week06 (main)
$
```

The status bar at the bottom indicates the current position is Line 9, Column 1, with 4 spaces, UTF-8 encoding, CRLF line endings, Python 3.9.21 interpreter, and a conda environment.

The image shows a Visual Studio Code editor window with a file explorer on the left and a code editor in the center. The file explorer shows a project named 'WEEK06' with files: '.gitignore', 'environment.yml', 'flow\_controls.py', 'guessing\_game.py', 'LICENSE', and 'README.md'. The code editor is open to 'flow\_controls.py' and contains the following Python code:

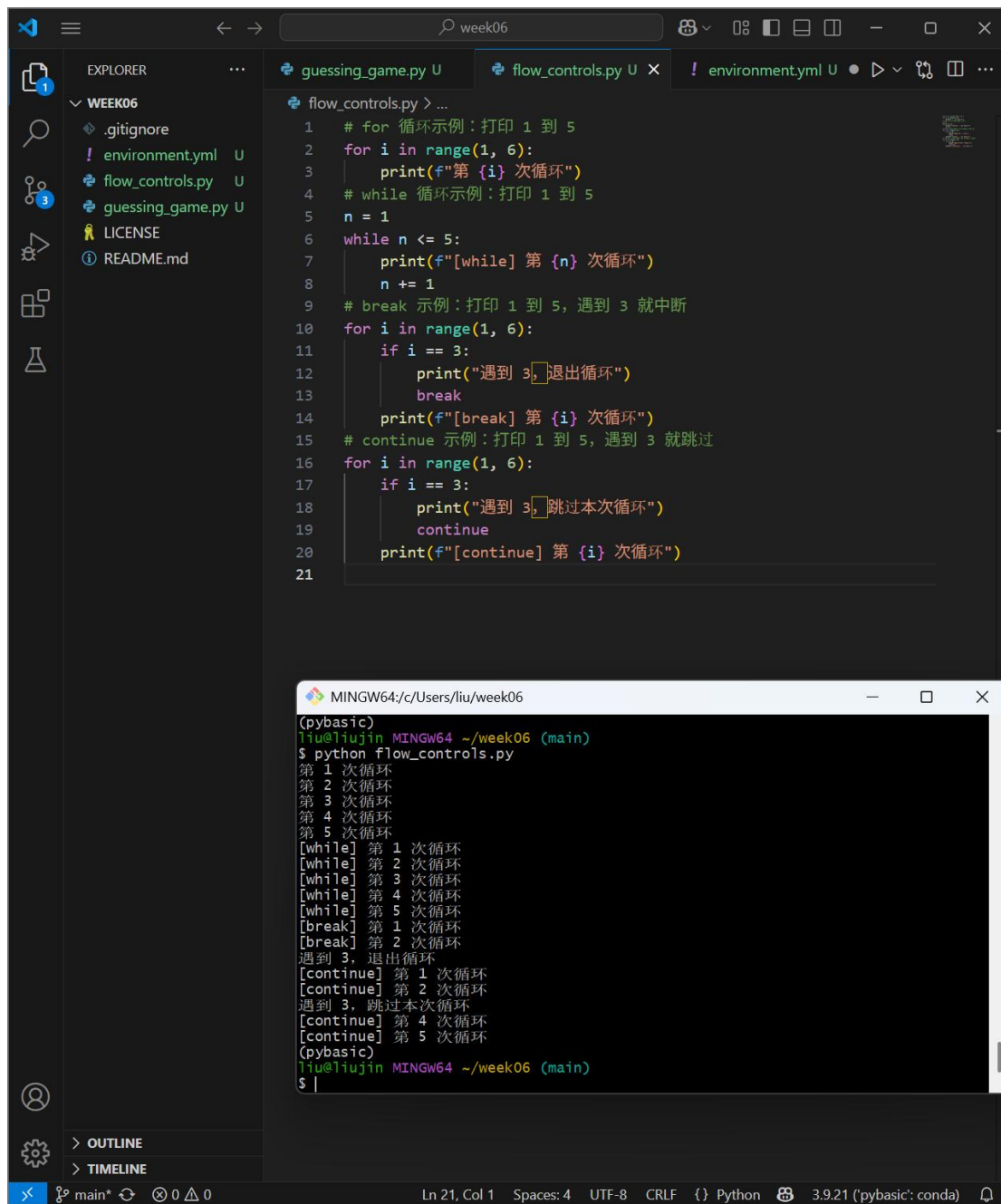
```
1 # for 循环示例: 打印 1 到 5
2 for i in range(1, 6):
3     print(f"第 {i} 次循环")
4 # while 循环示例: 打印 1 到 5
5 n = 1
6 while n <= 5:
7     print(f"[while] 第 {n} 次循环")
8     n += 1
9 # break 示例: 打印 1 到 5, 遇到 3 就中断
10 for i in range(1, 6):
11     if i == 3:
12         print("遇到 3, 退出循环")
13         break
14     print(f"[break] 第 {i} 次循环")
15
```

Below the code editor, a terminal window is open, showing the execution of the code. The terminal output is as follows:

```
MINGW64/c/Users/liu/week06
[while] 第 5 次循环
(pybasic)
liu@liujin MINGW64 ~/week06 (main)
$ ^[[200~python flow_controls.py
bash: $'\E[200~python': command not found
(pybasic)
liu@liujin MINGW64 ~/week06 (main)
$ python flow_controls.py
第 1 次循环
第 2 次循环
第 3 次循环
第 4 次循环
第 5 次循环
[while] 第 1 次循环
[while] 第 2 次循环
[while] 第 3 次循环
[while] 第 4 次循环
[while] 第 5 次循环
[break] 第 1 次循环
[break] 第 2 次循环
遇到 3, 退出循环
(pybasic)
liu@liujin MINGW64 ~/week06 (main)
$
```

The status bar at the bottom of the editor shows 'Ln 15, Col 1', 'Spaces: 4', 'UTF-8', 'CRLF', 'Python', and '3.9.21 (pybasic: conda)'.





The image shows a Visual Studio Code editor window with a file explorer on the left and a terminal window at the bottom. The file explorer shows a project named 'WEEK06' with files: .gitignore, environment.yml, flow\_controls.py, guessing\_game.py, LICENSE, and README.md. The editor is open to 'flow\_controls.py', which contains Python code for flow controls. The code includes comments in Chinese and uses 'break', 'continue', and 'for...else' loops. The terminal window shows the output of running 'python flow\_controls.py', displaying the execution of these loops and their results.

```
14     print(f"[break] 第 {i} 次循环")
15     # continue 示例：打印 1 到 5，遇到 3 就跳过
16     for i in range(1, 6):
17         if i == 3:
18             print("遇到 3, 跳过本次循环")
19             continue
20         print(f"[continue] 第 {i} 次循环")
21     # for...else 示例：遍历列表，没遇到 break 就执行 else
22     for i in range(1, 6):
23         print(f"[for...else] 正在检查数字：{i}")
24         if i == 10:
25             print("找到了 10!")
26             break
27     else:
28         print(f"[for...else] 没有找到 10, 执行了 else 块")
29     # if...else 示例：判断奇偶
30     for i in range(1, 6):
31         if i % 2 == 0:
32             print(f"[if...else] {i} 是偶数")
33         else:
34             print(f"[if...else] {i} 是奇数")
35
```

```
(pybasic)
liu@liujin MINGW64 ~/week06 (main)
$ python flow_controls.py
第 1 次循环
第 2 次循环
第 3 次循环
第 4 次循环
第 5 次循环
[while] 第 1 次循环
[while] 第 2 次循环
[while] 第 3 次循环
[while] 第 4 次循环
[while] 第 5 次循环
[break] 第 1 次循环
[break] 第 2 次循环
遇到 3, 退出循环
[continue] 第 1 次循环
[continue] 第 2 次循环
遇到 3, 跳过本次循环
[continue] 第 4 次循环
[continue] 第 5 次循环
[for...else] 正在检查数字：1
[for...else] 正在检查数字：2
[for...else] 正在检查数字：3
[for...else] 正在检查数字：4
[for...else] 正在检查数字：5
[for...else] 没有找到 10, 执行了 else 块
[if...else] 1 是奇数
[if...else] 2 是偶数
[if...else] 3 是奇数
[if...else] 4 是偶数
[if...else] 5 是奇数
(pybasic)
liu@liujin MINGW64 ~/week06 (main)
$
```



The image shows a Visual Studio Code editor window with a file explorer on the left and a code editor in the center. The file explorer shows a project named 'WEEK06' with files: .gitignore, environment.yml, flow\_controls.py, guessing\_game.py, LICENSE, and README.md. The code editor displays the contents of 'flow\_controls.py', which includes a for loop, an if-else statement, a try-except block, and a raise statement. A terminal window is open in the foreground, showing the execution of the script. The output includes comments from the code, such as '找到了 10!', '没有找到 10, 执行了 else 块', and '1 是奇数', '2 是偶数', etc. The terminal also shows the user's input '4' in response to the prompt '请输入一个整数: '.

```
25         print("找到了 10!")
26         break
27     else:
28         print("[for...else] 没有找到 10, 执行了 else 块")
29     # if...else 示例: 判断奇偶
30     for i in range(1, 6):
31         if i % 2 == 0:
32             print(f"[if...else] {i} 是偶数")
33         else:
34             print(f"[if...else] {i} 是奇数")
35     # try...except 示例: 捕获非数字输入
36     for i in range(3):
37         try:
38             x = int(input("[try...except] 请输入一个整数: "))
39             print(f"[try...except] 你输入的是: {x}")
40         except ValueError:
41             print("[try...except] 输入不是有效的整数!")
42     # raise 示例: 遇到 3 主动抛出异常
43     for i in range(1, 6):
44         print(f"[raise] 当前数字: {i}")
45         if i == 3:
46             raise ValueError("[raise] 我不喜欢 3 主动抛出异常")
```

```
[if...else] 4 是偶数
[if...else] 5 是奇数
(pybasic)
liu@liujin MINGW64 ~/week06 (main)
$ python flow_controls.py
第 1 次循环
第 2 次循环
第 3 次循环
第 4 次循环
第 5 次循环
[while] 第 1 次循环
[while] 第 2 次循环
[while] 第 3 次循环
[while] 第 4 次循环
[while] 第 5 次循环
[break] 第 1 次循环
[break] 第 2 次循环
遇到 3, 退出循环
[continue] 第 1 次循环
[continue] 第 2 次循环
遇到 3, 跳过本次循环
[continue] 第 4 次循环
[continue] 第 5 次循环
[for...else] 正在检查数字: 1
[for...else] 正在检查数字: 2
[for...else] 正在检查数字: 3
[for...else] 正在检查数字: 4
[for...else] 正在检查数字: 5
[for...else] 没有找到 10, 执行了 else 块
[if...else] 1 是奇数
[if...else] 2 是偶数
[if...else] 3 是奇数
[if...else] 4 是偶数
[if...else] 5 是奇数
[try...except] 请输入一个整数: 4
```

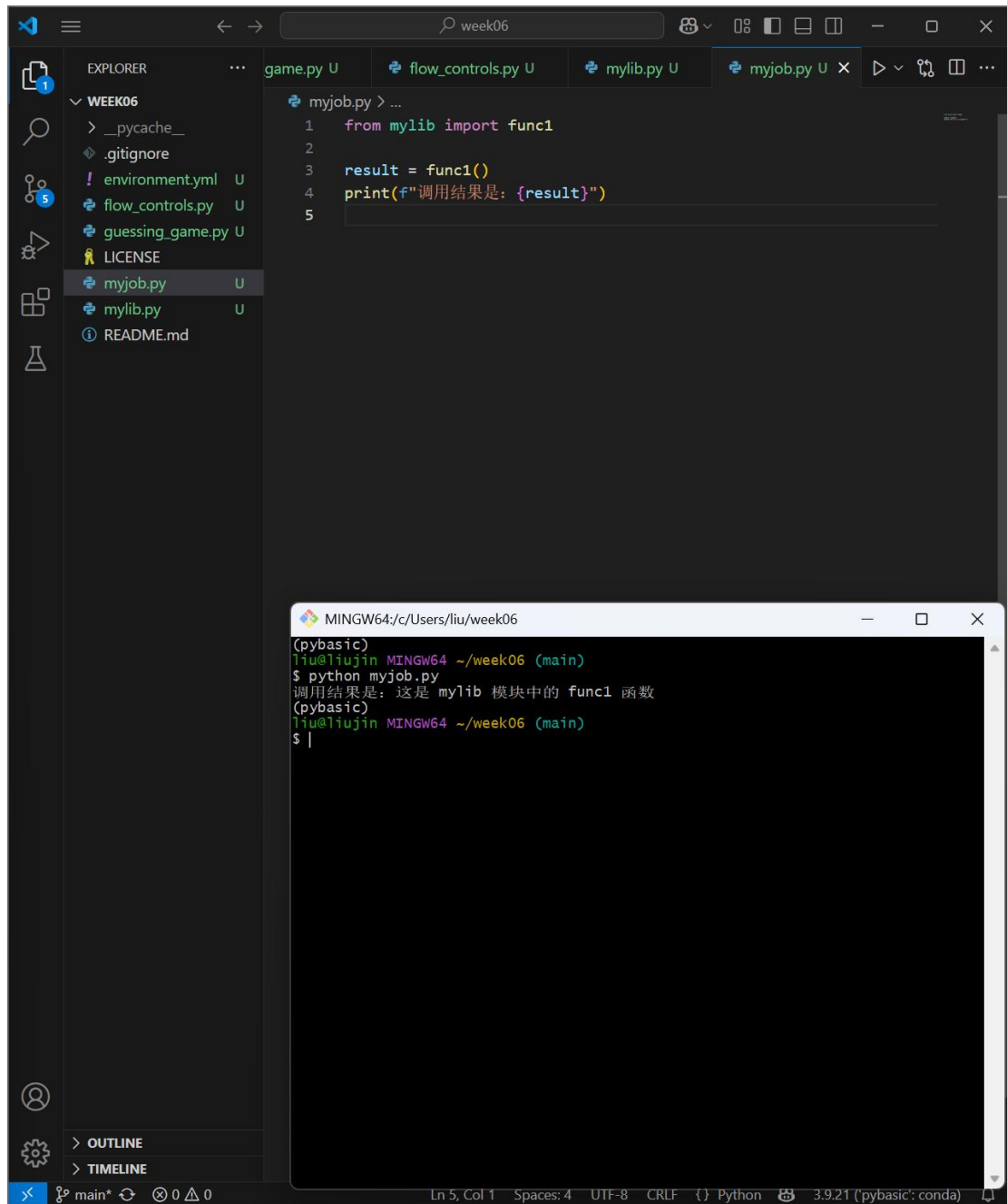
The image shows a VS Code editor window with a file explorer on the left and a code editor in the center. The file explorer shows a project named 'WEEK06' with files: '.gitignore', 'environment.yml', 'flow\_controls.py', 'guessing\_game.py', 'LICENSE', and 'README.md'. The code editor shows the 'flow\_controls.py' file with the following code:

```
25     print("找到了 10!")
26     break
27 else:
28     print("[for...else] 没有找到 10, 执行了 else 块")
29 # if...else 示例: 判断奇偶
30 for i in range(1, 6):
31     if i % 2 == 0:
32         print(f"[if...else] {i} 是偶数")
33     else:
34         print(f"[if...else] {i} 是奇数")
35 # try...except 示例: 捕获非数字输入
36 for i in range(3):
37     try:
38         x = int(input("[try...except] 请输入一个整数: "))
39         print(f"[try...except] 你输入的是: {x}")
40     except ValueError:
41         print("[try...except] 输入不是有效的整数!")
42 # raise 示例: 遇到 3 主动抛出异常
43 for i in range(1, 6):
44     print(f"[raise] 当前数字: {i}")
45     if i == 3:
46         raise ValueError("[raise] 我不喜欢 3, 主动抛出异常!")
```

The terminal window shows the execution output:

```
[break] 第 1 次循环
[break] 第 2 次循环
遇到 3, 退出循环
[continue] 第 1 次循环
[continue] 第 2 次循环
遇到 3, 跳过本次循环
[continue] 第 4 次循环
[continue] 第 5 次循环
[for...else] 正在检查数字: 1
[for...else] 正在检查数字: 2
[for...else] 正在检查数字: 3
[for...else] 正在检查数字: 4
[for...else] 正在检查数字: 5
[for...else] 没有找到 10, 执行了 else 块
[if...else] 1 是奇数
[if...else] 2 是偶数
[if...else] 3 是奇数
[if...else] 4 是偶数
[if...else] 5 是奇数
[try...except] 请输入一个整数: 3
[try...except] 你输入的是: 3
[try...except] 请输入一个整数: 5
[try...except] 你输入的是: 5
[try...except] 请输入一个整数: 3
[try...except] 你输入的是: 3
[raise] 当前数字: 1
[raise] 当前数字: 2
[raise] 当前数字: 3
Traceback (most recent call last):
  File "C:\Users\liu\week06\flow_controls.py", line 46, in <module>
    raise ValueError("[raise] 我不喜欢 3, 主动抛出异常!")
ValueError: [raise] 我不喜欢 3, 主动抛出异常!
liu@liujin MINGW64 ~/week06 (main)
$
```

开始有疑问, 为什么开始输入 3 就没事, 第三次输入 3 就不喜欢 3, 后来发现这段代码的意思是第三次固定报错。



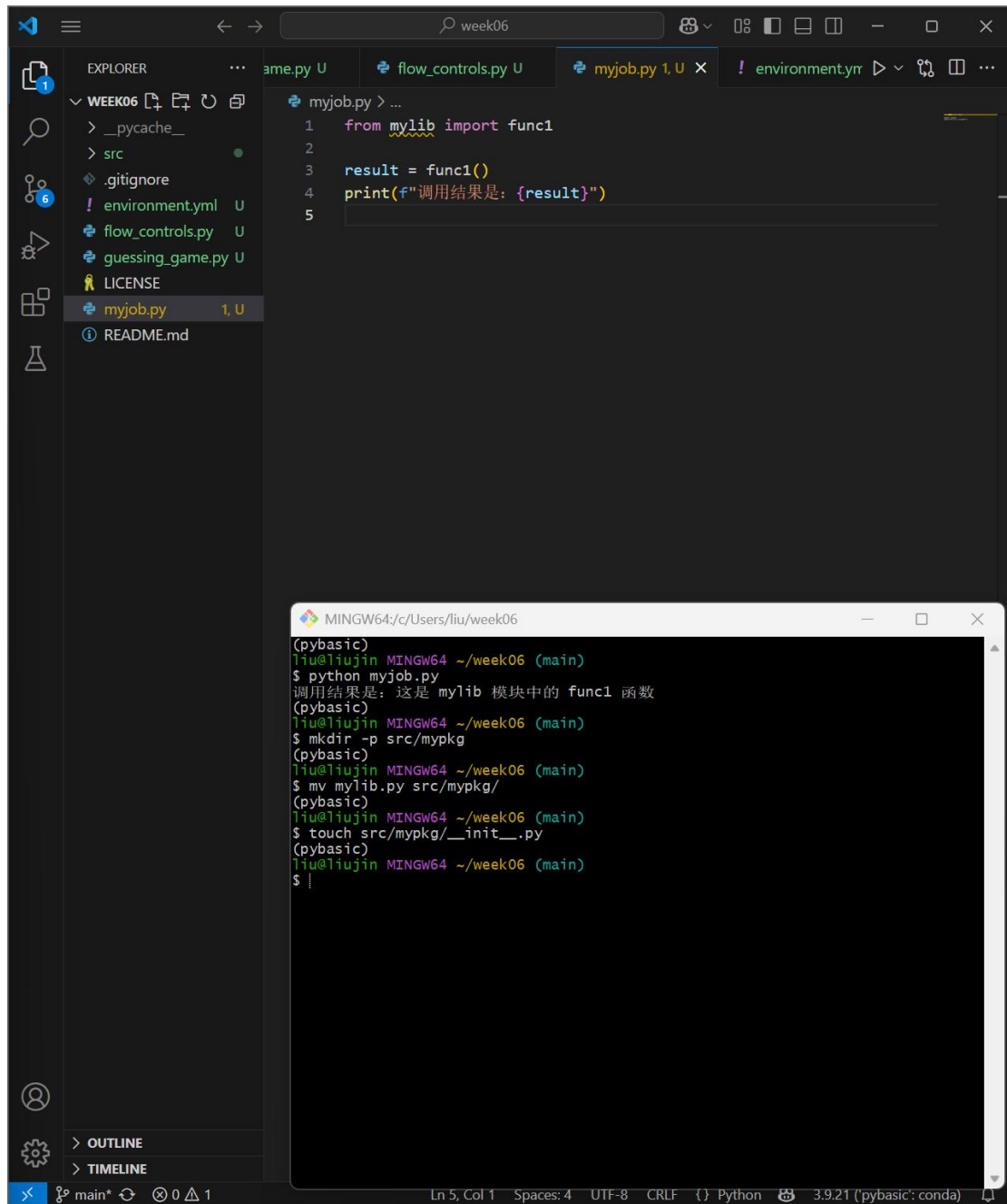
The image shows a Visual Studio Code editor window with a dark theme. The Explorer sidebar on the left shows a project named 'WEEK06' with files: `__pycache__`, `src`, `.gitignore`, `environment.yml`, `flow_controls.py`, `guessing_game.py`, `LICENSE`, `myjob.py` (selected), and `README.md`. The main editor area shows the content of `myjob.py`:

```
1 from mylib import func1
2
3 result = func1()
4 print(f"调用结果是: {result}")
5
```

Below the editor, a terminal window is open with the title 'MINGW64/c/Users/liu/week06'. It shows the following commands and output:

```
(pybasic)
liu@liujin MINGW64 ~/week06 (main)
$ python myjob.py
调用结果是: 这是 mylib 模块中的 func1 函数
(pybasic)
liu@liujin MINGW64 ~/week06 (main)
$ mkdir -p src/mypkg
(pybasic)
liu@liujin MINGW64 ~/week06 (main)
$ mv mylib.py src/mypkg/
(pybasic)
liu@liujin MINGW64 ~/week06 (main)
$
```

The status bar at the bottom indicates the current file is `main*`, with 0 errors and 1 warning. It also shows the cursor is at line 5, column 1, with 4 spaces, UTF-8 encoding, CRLF line endings, and the file is a Python script using the 3.9.21 interpreter from conda.



VS Code Explorer sidebar showing the file structure of 'WEEK06':

- \_\_pycache\_\_
- src
- .gitignore
- environment.yml U
- flow\_controls.py U
- guessing\_game.py U
- LICENSE
- myjob.py 1, U
- pyproject.toml U
- README.md

VS Code Editor showing 'myjob.py':

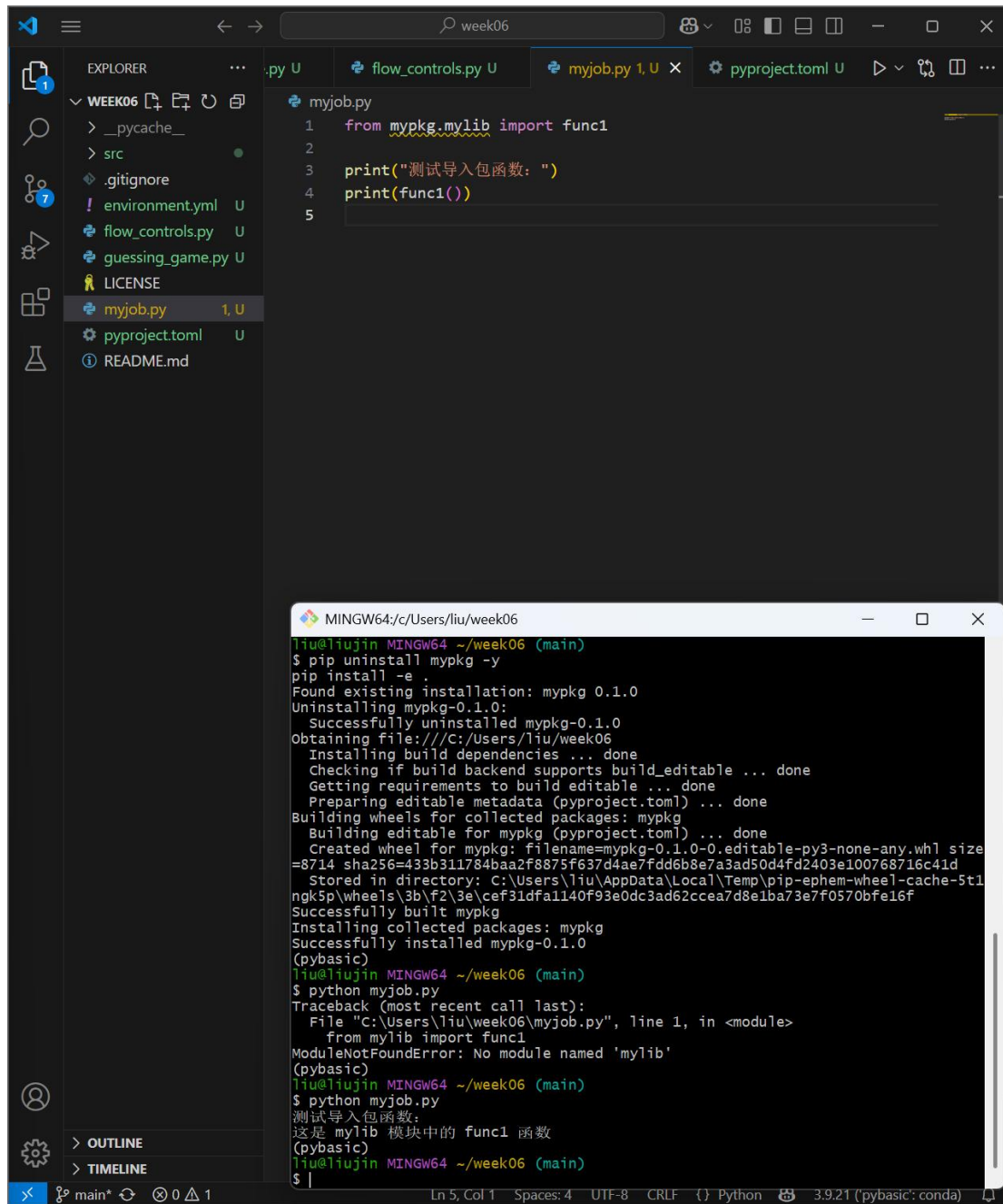
```
1 from mypkg.mylib import func1
2
3 print('测试导入包函数: ')
4 print(func1())
5
```

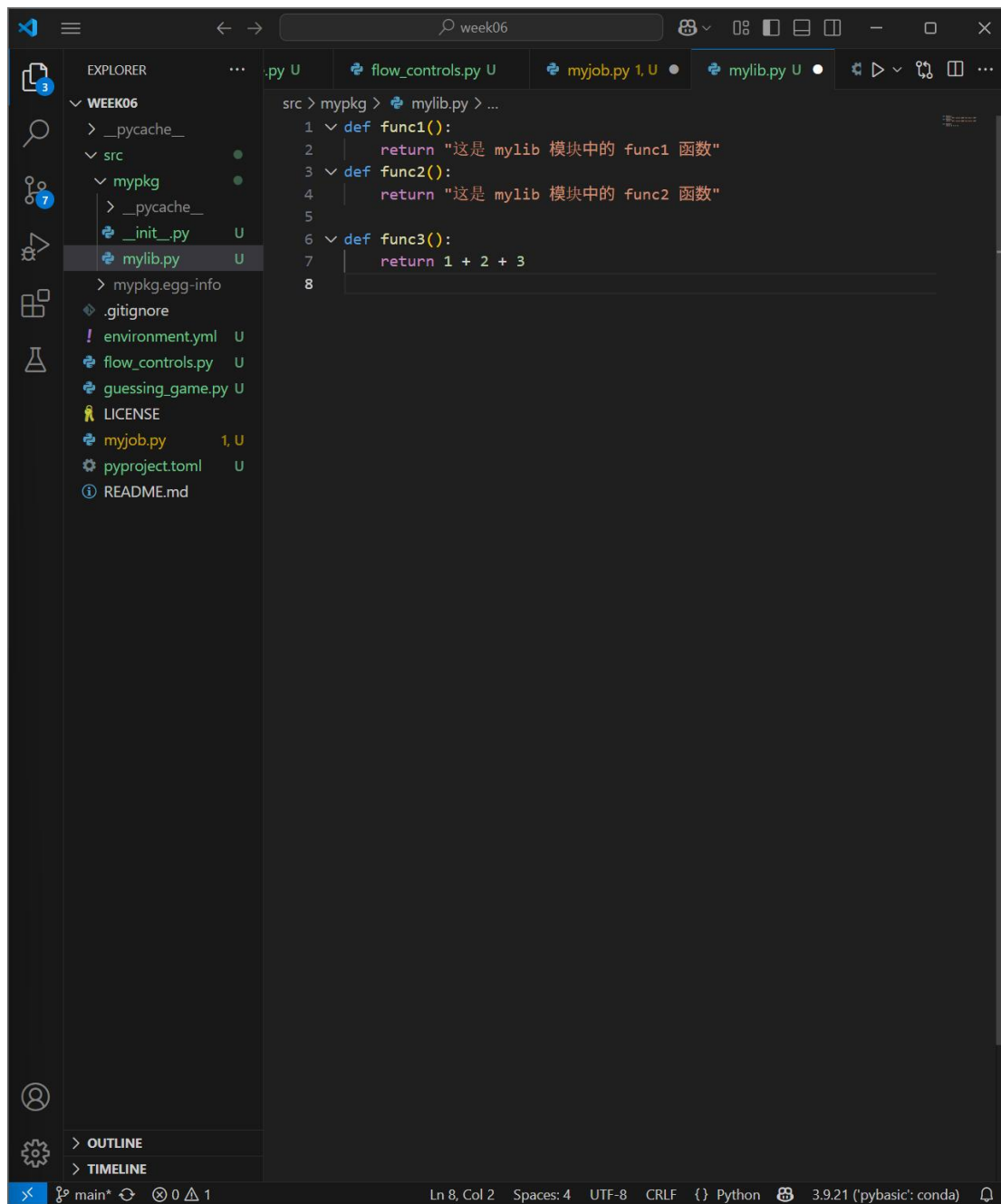
Terminal window output:

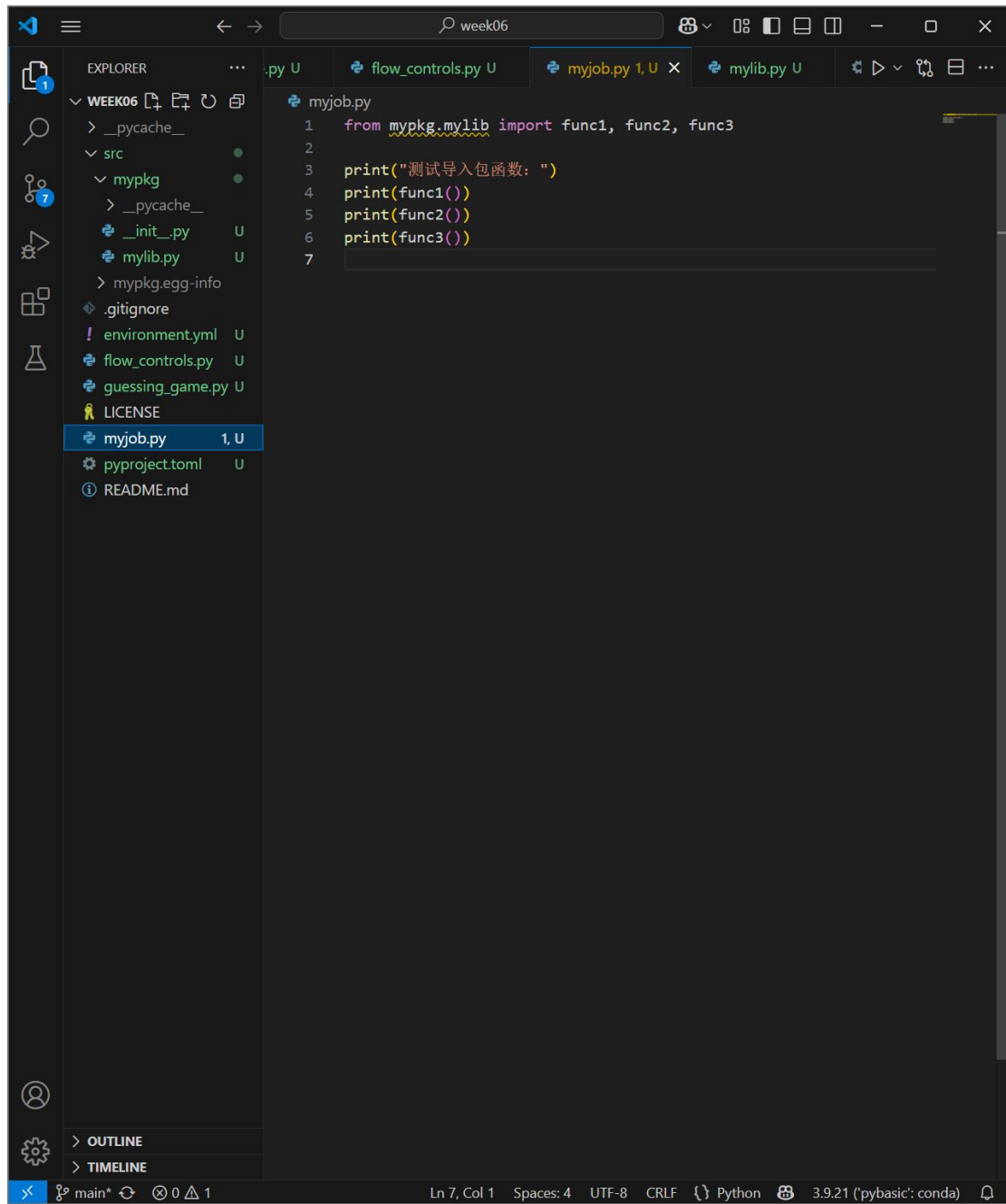
```
MINGW64: c:/Users/liu/week06
liu@liujin MINGW64 ~/week06 (main)
$ mv mylib.py src/mypkg/
(pybasic)
liu@liujin MINGW64 ~/week06 (main)
$ touch src/mypkg/__init__.py
(pybasic)
liu@liujin MINGW64 ~/week06 (main)
$ conda activate pybasic
(pybasic)
liu@liujin MINGW64 ~/week06 (main)
$ pip install -e .
Obtaining file:///C:/Users/liu/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
Building wheels for collected packages: mypkg
Building editable for mypkg (pyproject.toml) ... done
Created wheel for mypkg: filename=mypkg-0.1.0-0.editable-py3-none-any.whl size
=8714 sha256=546365e77e5fe87e2c8d3dc124881dc40405e63713ae6d43293e96df5fcd2e39
Stored in directory: C:\Users\liu\AppData\Local\Temp\pip-ephem-wheel-cache-i7_
mgabx\wheels\3b\f2\3e\cef31dfa1140f93e0dc3ad62ccea7d8e1ba73e7f0570bfe16f
Successfully built mypkg
Installing collected packages: mypkg
Successfully installed mypkg-0.1.0
(pybasic)
liu@liujin MINGW64 ~/week06 (main)
$ python myjob.py
Traceback (most recent call last):
  File "C:\Users\liu\week06\myjob.py", line 1, in <module>
    from mylib import func1
ModuleNotFoundError: No module named 'mylib'
(pybasic)
liu@liujin MINGW64 ~/week06 (main)
$
```

Status bar: main\* Ln 5, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.9.21 (pybasic: conda)









VS Code Explorer sidebar showing the file structure of the 'WEEK06' project:

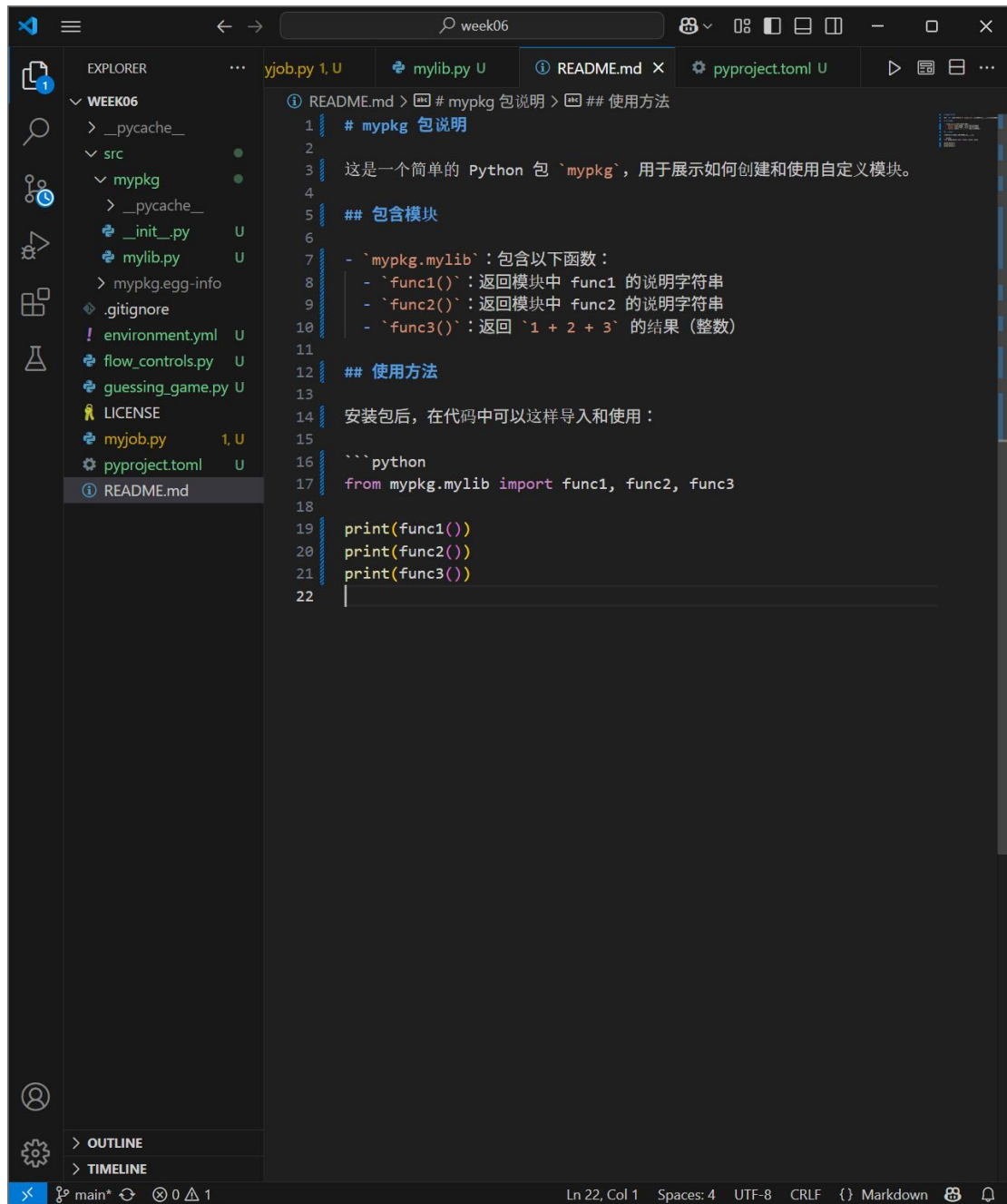
- WEEK06
  - > \_\_pycache\_\_
  - src
    - > mypkg
      - > \_\_pycache\_\_
      - \_\_init\_\_.py U
      - mylib.py U
    - mypkg.egg-info
    - .gitignore
    - ! environment.yml U
    - flow\_controls.py U
    - guessing\_game.py U
    - LICENSE
    - myjob.py 1, U
    - pyproject.toml U
    - README.md

The myjob.py file is open in the editor, showing the following code:

```
1 from mypkg.mylib import func1, func2, func3
2
3 print("测试导入包函数: ")
4 print(func1())
5 print(func2())
6 print(func3())
7
```

The terminal window shows the output of the pip install and python myjob.py commands:

```
Installing build dependencies ... done
Checking if build backend supports build_editable ... done
Getting requirements to build editable ... done
Preparing editable metadata (pyproject.toml) ... done
Building wheels for collected packages: mypkg
Building editable for mypkg (pyproject.toml) ... done
Created wheel for mypkg: filename=mypkg-0.1.0-0.editable-py3-none-any.whl size
=8714 sha256=433b311784baa2f8875f637d4ae7fdd6b8e7a3ad50d4fd2403e100768716c41d
Stored in directory: C:\Users\liu\AppData\Local\Temp\pip-ephem-wheel-cache-5t1
ngk5p\wheels\3b\f2\3e\cef31dfa1140f93e0dc3ad62ccea7d8e1ba73e7f0570bfe16f
Successfully built mypkg
Installing collected packages: mypkg
Successfully installed mypkg-0.1.0
(pybasic)
liu@liujin MINGW64 ~/week06 (main)
$ python myjob.py
Traceback (most recent call last):
  File "C:\Users\liu\week06\myjob.py", line 1, in <module>
    from mylib import func1
ModuleNotFoundError: No module named 'mylib'
(pybasic)
liu@liujin MINGW64 ~/week06 (main)
$ python myjob.py
测试导入包函数:
这是 mylib 模块中的 func1 函数
(pybasic)
liu@liujin MINGW64 ~/week06 (main)
$ python myjob.py
测试导入包函数:
这是 mylib 模块中的 func1 函数
这是 mylib 模块中的 func2 函数
6
(pybasic)
liu@liujin MINGW64 ~/week06 (main)
$
```



The image shows a Visual Studio Code editor window with a dark theme. The Explorer sidebar on the left shows a project named 'WEEK06' with a 'src' directory containing a 'mypkg' subdirectory. The 'mypkg' directory contains files like '\_\_init\_\_.py', 'mylib.py', and 'mypkg.egg-info'. The 'README.md' file is selected and open in the main editor. The README content is in Chinese and describes a Python package named 'mypkg'. It includes sections for '包说明' (Package Description) and '包含模块' (Included Modules), listing functions like 'func1', 'func2', and 'func3'. It also has a '使用方法' (Usage) section showing how to install and use the package. Below the editor, a terminal window is open, showing the execution of a Python script 'myjob.py' and a series of Git commands to add, commit, and push the changes. The terminal output shows the successful execution of the script and the completion of the Git operations. The status bar at the bottom indicates the current file is 'main\*' and the cursor is at line 22, column 1.

WEEK06

- > \_\_pycache\_\_
- > src
  - > mypkg
    - > \_\_pycache\_\_
    - \_\_init\_\_.py U
    - mylib.py U
  - > mypkg.egg-info
- .gitignore
- ! environment.yml U
- flow\_controls.py U
- guessing\_game.py U
- LICENSE
- myjob.py 1, U
- pyproject.toml U
- ① README.md M

① README.md > # mypkg 包说明 > ## 使用方法

```
1 # mypkg 包说明
2
3 这是一个简单的 Python 包 `mypkg`, 用于展示如何创建和使用自定义模块。
4
5 ## 包含模块
6
7 - `mypkg.mylib`: 包含以下函数:
8   - `func1`: 返回模块中 func1 的说明字符串
9   - `func2`: 返回模块中 func2 的说明字符串
10  - `func3`: 返回 `1 + 2 + 3` 的结果 (整数)
11
12 ## 使用方法
13
14 安装包后, 在代码中可以这样导入和使用:
15
16 ```python
17 from mypkg.mylib import func1, func2, func3
18
19 print(func1())
20 print(func2())
21 print(func3())
22
```

MINGW64/c:/Users/liu/week06

测试导入包函数:  
这是 mylib 模块中的 func1 函数  
(pybasic)  
liu@liujin MINGW64 ~/week06 (main)  
\$ python myjob.py  
测试导入包函数:  
这是 mylib 模块中的 func1 函数  
这是 mylib 模块中的 func2 函数  
6  
(pybasic)  
liu@liujin MINGW64 ~/week06 (main)  
\$ git add .  
git commit -m "完成 week06 作业, 添加包模块及说明文档"  
git push  
[main 86a7e83] 完成 week06 作业, 添加包模块及说明文档  
8 files changed, 147 insertions(+), 16 deletions(-)  
create mode 100644 environment.yml  
create mode 100644 flow\_controls.py  
create mode 100644 guessing\_game.py  
create mode 100644 myjob.py  
create mode 100644 pyproject.toml  
create mode 100644 src/mypkg/\_\_init\_\_.py  
create mode 100644 src/mypkg/mylib.py  
Enumerating objects: 14, done.  
Counting objects: 100% (14/14), done.  
Delta compression using up to 22 threads  
Compressing objects: 100% (10/10), done.  
Writing objects: 100% (12/12), 2.83 KiB | 263.00 KiB/s, done.  
Total 12 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)  
remote: Start Git Hooks Checking  
To gitcode.com:liujin196/week06.git  
3c57a68..86a7e83 main -> main  
(pybasic)  
liu@liujin MINGW64 ~/week06 (main)  
\$

Ln 22, Col 1 Spaces: 4 UTF-8 CRLF {} Markdown