

Week07 作业报告

1、Fork 第 06 周打卡仓库至你的名下，然后将你名下的这个仓库 Clone 到你的本地计算机

```
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 | 393.00 KiB/s, done.

(base) DELL@DESKTOP-ISUNFNG MINGW64 ~/repo
$ ls
myproject/  prj2/  week04/  week05/  week06/

(base) DELL@DESKTOP-ISUNFNG MINGW64 ~/repo
$ ll
total 12
drwxr-xr-x 1 DELL 197121 0 3月 23 00:51 myproject/
drwxr-xr-x 1 DELL 197121 0 3月 22 23:57 prj2/
drwxr-xr-x 1 DELL 197121 0 3月 31 09:34 week04/
drwxr-xr-x 1 DELL 197121 0 4月 12 22:32 week05/
drwxr-xr-x 1 DELL 197121 0 4月 22 16:15 week06/

(base) DELL@DESKTOP-ISUNFNG MINGW64 ~/repo
$ cd week06

(base) DELL@DESKTOP-ISUNFNG MINGW64 ~/repo/week06 (main)
$ ls -l
total 24
-rw-r--r-- 1 DELL 197121 18805 4月 22 16:15 LICENSE
-rw-r--r-- 1 DELL 197121 2239 4月 22 16:15 README.md

(base) DELL@DESKTOP-ISUNFNG MINGW64 ~/repo/week06 (main)
$ cp ../week05/environment.yml ./

(base) DELL@DESKTOP-ISUNFNG MINGW64 ~/repo/week06 (main)
$ ll
total 25
-rw-r--r-- 1 DELL 197121 95 4月 22 16:22 environment.yml
-rw-r--r-- 1 DELL 197121 18805 4月 22 16:15 LICENSE
-rw-r--r-- 1 DELL 197121 2239 4月 22 16:15 README.md

(base) DELL@DESKTOP-ISUNFNG 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

```
Verifying transaction: done
Executing transaction: done
Everything found within the environment (C:\Users\DELL\anaconda3\envs\prj3), including any conda environment configurations and any non-conda files, will be
deleted. Do you wish to continue?
(y/[n])? yy
Invalid choice: yy
Everything found within the environment (C:\Users\DELL\anaconda3\envs\prj3), including any conda environment configurations and any non-conda files, will be
deleted. Do you wish to continue?
(y/[n])? y

(base) DELL@DESKTOP-ISUNFNG MINGW64 ~/repo/week06 (main)
$ conda activate week06

(week06) DELL@DESKTOP-ISUNFNG MINGW64 ~/repo/week06 (main)
$ conda list
# packages in environment at C:\Users\DELL\anaconda3\envs\week06:
#
# Name                        Version           Build    Channel
bz2ip2                        1.0.8             h2466b09_7 conda-forge
ca-certificates              2025.1.31         h4c74964_1 conda-forge
libexpat                     2.7.0             h6c23c2_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           h3f8d4cb_0_cpython conda-forge
setuptools                    79.0.0            pyhff2d567_0 conda-forge
tk                             8.6.13            h5226925_1 conda-forge
tzdata                       2025b             h78e105d_0 conda-forge
ucrt                          10.0.22621.0      h57928b3_1 conda-forge
vc                             14.3              h2b53caa_26 conda-forge
vccll_runtime                 14.42.34438       hfd919c2_26 conda-forge
wat-inspector                 0.4.3             pyhf2d567_0 conda-forge
wheel                         0.45.1            pyhd8ed1ab_1 conda-forge

(week06) DELL@DESKTOP-ISUNFNG 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：

```
(week06)
DELL@DESKTOP-I5UNFNG MINGW64 ~/repo/week06 (main)
$ python -m pdb guessing_game.py
> c:\users\de\l\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\de\l\repo\week06\guessing_game.py(4)<module>()
-> def guessing_game():
(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) l
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
```

```
DELL@DESKTOP-I5UNFNG 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): q
游戏结束，再见。
(week06)
DELL@DESKTOP-I5UNFNG 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): 115
输入无效，输入值应该在 1~100 之间。
(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): 80
猜的数字太大了，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): 120
输入无效，输入值应该在 1~100 之间。
(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):
DELL@DESKTOP-I5UNFNG MINGW64 ~/repo/week06 (main)
$ python guessing_game.py 30
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): 7
猜的数字太小了，再试试。
(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): 60
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): 70
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): 65
猜的数字太小了，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): 67
恭喜你，猜对了！
游戏结束，再见。
(week06)
```

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 循环

豆包

Python for 语句示例

对话

搜索

写作

编程

生成

更多

云盘

历史

Python for 语句示例

手机对话

分析代理成本和内部治理

文章翻译与结论

写句子

python for 语句例子

在 Python 中，for 语句常用于遍历可迭代对象（如列表、元组、字符串等）。下面是一些 for 语句的使用示例：

遍历列表

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

遍历字符串

```
python
message = "Hello"
for char in message:
    print(char)
```

```
EXPLORER
WEEK06
  .gitignore
  ! environment.yml
  ! flow_controls.py
  ! guessing_game.py
  ! flow_controls.py

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

DELL@DESKTOP-ISUNFNG MINGW64 ~/repo/week06 (main)
$ python flow_controls.py
apple
banana
cherry
(main)
```

字符串

```
guessing_game.py
LICENSE
README.md

4
5
6  message = "Hello"
7  for char in message:
8      print(char)
9
10
11  for i in range(5):
12      print(i)
13
```

```
DELL@DESKTOP-ISUNFNG MINGW64 ~/repo/week06 (main)
$ python flow_controls.py
apple
banana
cherry
H
e
l
l
o
0
1
2
3
4
(main)
```

字典

```
13
14
15  person = {"name": "John", "age": 30, "city": "New York"}
16  for key in person.keys():
17      print(key)
18
19
20  person = {"name": "John", "age": 30, "city": "New York"}
21  for value in person.values():
22      print(value)
23
```

```
DELL@DESKTOP-ISUNFNG MINGW64 ~/repo/week06 (main)
$ python flow_controls.py
apple
banana
cherry
H
e
l
l
o
0
1
2
3
4
name
age
city
John
30
New York
(main)
```

While 循环

```
25  count = 0
26  while count < 5:
27      print(count)
28      count = count + 1
29
```

```
DELL@DESKTOP-ISUNFNG MINGW64 ~/repo/week06 (main)
$ python flow_controls.py
apple
banana
cherry
H
e
l
l
o
0
1
2
3
4
name
age
city
John
30
New York
0
1
2
3
4
(week06)
```

for 循环: 会自动对可迭代对象中的每个元素进行迭代，每次迭代时，将当前元素赋值给指定的变量，然后执行循环体代码，直至遍历完所有元素。

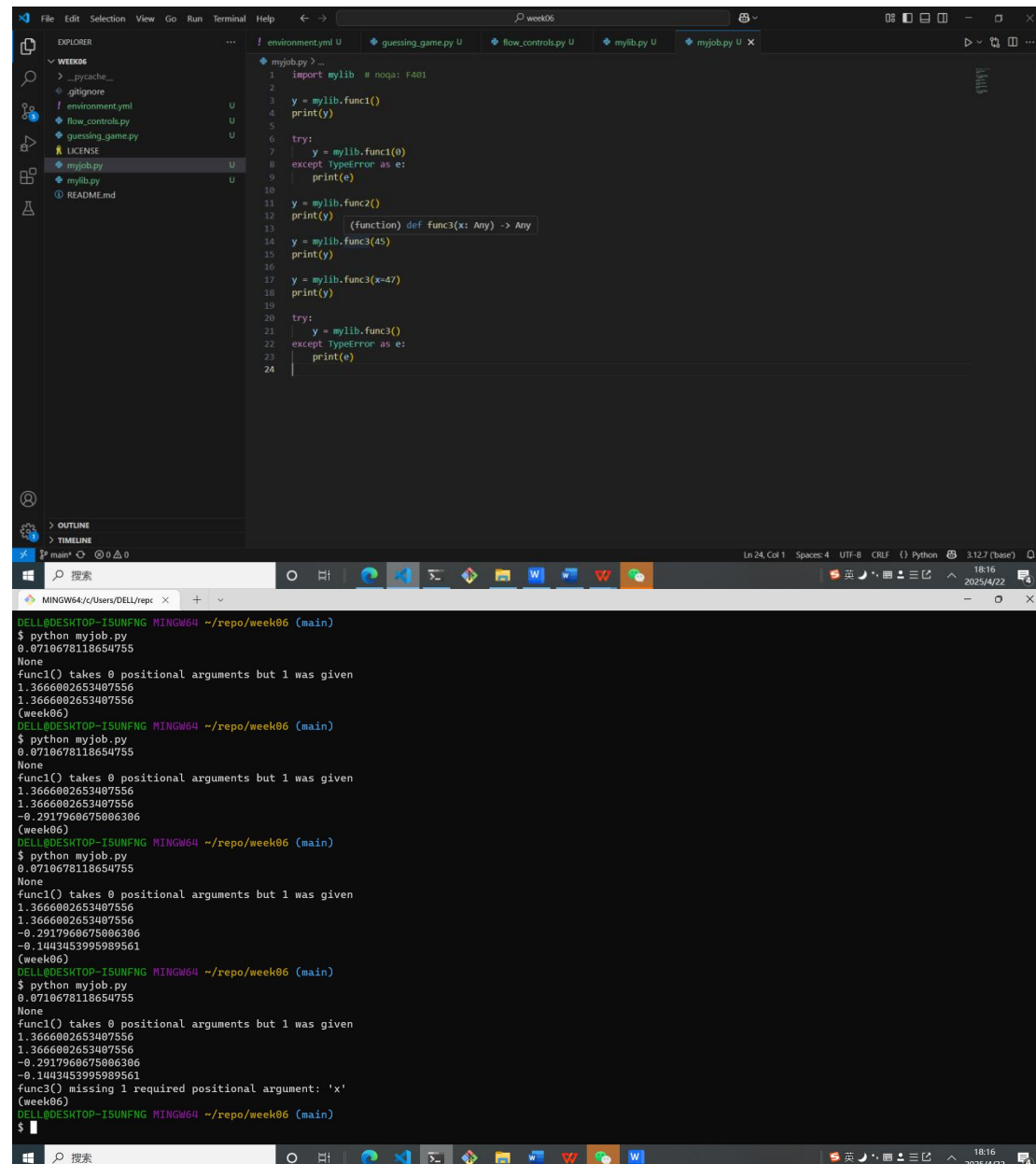
while 循环: 会先判断条件表达式的真假，如果为 `True` 则执行循环体代码，执行完后再次判断条件表达式，若仍为 `True` 则继续执行，直到条件表达式为 `False` 才停止循环。

5、创建一个 `mylib.py` 模块 (module), 在里面定义以下函数, 再创建一个 `myjob.py` 脚本 (script), 从 `mylib.py` 导入函数并尝试调用

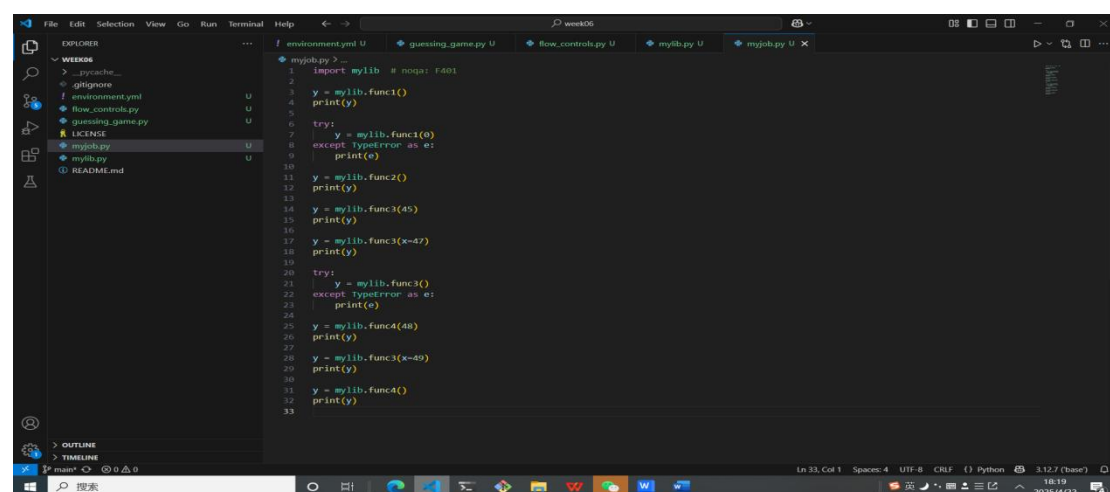
```
1 import mylib # noqa: F401
2
3 y = mylib.func1()
4 print(y)
5
6 try:
7     y = mylib.func1(0)
8 except TypeError as e:
9     print(e)
10
```

```
C:\Users\DELL\repo\week06\python.exe: can't open file 'C:\Users\DELL\repo\week06\myob.py': [Errno 2] No such file or directory
(week06)
DELL@DESKTOP-ISUNFNG MINGW64 ~/repo/week06 (main)
$ python myjob.py
--Return--
> c:\Users\DELL\repo\week06\myjob.py(3)<module>()->None
-> breakpoint()
(Pdb) l
1     import mylib # noqa: F401
2
3 -> breakpoint()
[EOF]
(Pdb) p mylib
<module 'mylib' from 'C:\\Users\\DELL\\repo\\week06\\mylib.py'>
(Pdb) import wat
(Pdb) quit()
Traceback (most recent call last):
  File "C:\\Users\\DELL\\repo\\week06\\myjob.py", line 3, in <module>
    mylib.func1()
    ^^^^^^^^^^^^^
  File "C:\\Users\\DELL\\anaconda3\\envs\\week06\\Lib\\bdb.py", line 104, in trace_dispatch
    return self.dispatch_return(frame, arg)
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\\Users\\DELL\\anaconda3\\envs\\week06\\Lib\\bdb.py", line 166, in dispatch_return
    if self.quitting: raise BdbQuit
    ^^^^^^^^^^^^^^^^^
bdb.BdbQuit
(week06)
DELL@DESKTOP-ISUNFNG MINGW64 ~/repo/week06 (main)
$ python myjob.py
0.0718678118654755
(week06)
DELL@DESKTOP-ISUNFNG MINGW64 ~/repo/week06 (main)
$ python myjob.py
None
func1() takes 0 positional arguments but 1 was given
(week06)
DELL@DESKTOP-ISUNFNG MINGW64 ~/repo/week06 (main)
$
```

位置实参



命名实参



```
MINGW64/c/Users/DELL/rep...
$ python myjob.py
0.0710678118654755
None
func1() takes 0 positional arguments but 1 was given
1.3666082653407556
1.3666082653407556
-0.2917960675006306
-0.1443453995989561
func3() missing 1 required positional argument: 'x'
-0.07179676972449123
(week06)
DELL@DESKTOP-1SUNFNG MINGW64 ~/repo/week06 (main)
$ python myjob.py
0.0710678118654755
None
func1() takes 0 positional arguments but 1 was given
1.3666082653407556
1.3666082653407556
-0.2917960675006306
-0.1443453995989561
func3() missing 1 required positional argument: 'x'
-0.07179676972449123
0.0
(week06)
DELL@DESKTOP-1SUNFNG MINGW64 ~/repo/week06 (main)
$ python myjob.py
0.0710678118654755
None
func1() takes 0 positional arguments but 1 was given
1.3666082653407556
1.3666082653407556
-0.2917960675006306
-0.1443453995989561
func3() missing 1 required positional argument: 'x'
-0.07179676972449123
0.0
0.0710678118654755
(week06)
DELL@DESKTOP-1SUNFNG 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

```
guessing_game.py  U  22
LICENSE           U  23
myjob.py          U  24
mylib.py          U  25
README.md         ①  26
                  27
                  28
                  29
                  30
                  31
def caculate(a, b, operation="add"):
    if operation == "add":
        return a + b
    elif operation == "subtract":
        return a - b
    else:
        return None
```

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

```
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
```

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

```
C:\Users\DELL\repo\week06\README.md 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
```

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)

```
51 def func8(*numbers):
52     total = 0
53     for num in numbers:
54         total = total + num
55     return total
56
```

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)

```
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
```

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`，接受一个位置形参，两个命名形参，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

```
68
69 def func11(arg1, arg2):
70     print(f"arg1 的值是: {arg1}")
71     print(f"arg2 的值是: {arg2}")
72
```

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

```
73
74 def func12(arg1: str, arg2: int, named_arg: str = "default") -> None:
75     """多个参数的调用例子"""
76     print(f"位置实参 arg1: {arg1}")
77     print(f"位置实参 arg2: {arg2}")
78     print(f"命名实参 named_arg: {named_arg}")
79
```

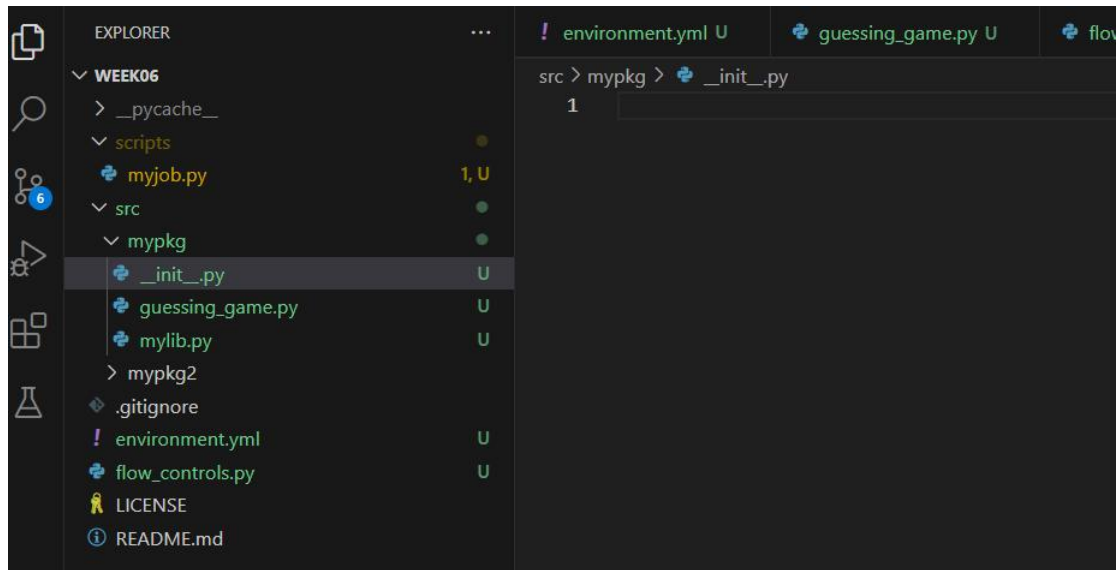
```
MINGW64~/Users/DELL/rep...
0.0718678118654755
None
func1() takes 0 positional arguments but 1 was given
1.3666802653407556
1.3666802653407556
-0.2917968675806306
-0.1443453995989561
func3() missing 1 required positional argument: 'x'
-0.87179676972449123
0.0
0.0718678118654755
15
15
15
-5
-3
func6() got some positional-only arguments passed as keyword arguments: 'a'
5
35
0
name: Alice
age: 25
city: New York
位置实参 arg1: 10
位置实参 arg2: 20
命名实参 named_arg: default
位置实参 arg1: 10
位置实参 arg2: 20
命名实参 named_arg: default
位置实参 arg1: 50
位置实参 arg2: 60
命名实参 named_arg: new value
arg1 的值是: 100
arg2 的值是: 200
位置实参 arg1: 7
位置实参 arg2: 8
命名实参 named_arg: 9
{week06}
DELL@DESKTOP-I5UNFNG MINGW64 ~/repo/week06 (main)
$
```

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

```
{week06}
DELL@DESKTOP-I5UNFNG MINGW64 ~/repo/week06 (main)
$ python scripts/myjob.py
Traceback (most recent call last):
  File "C:\Users\DELL\repo\week06\scripts\myjob.py", line 1, in <module>
    import mylib # noqa: F401
    ^^^^^^^^^^^
ModuleNotFoundError: No module named 'mylib'
{week06}
DELL@DESKTOP-I5UNFNG MINGW64 ~/repo/week06 (main)
$
```

```
EXPLORER
WEEK06
  __pycache__
  mylib.cpython-312.pyc
  scripts
    myjob.py
  src
    mypkg
    __init__.py
    guessing_game.py
    ...

! environment.yml
1 name: week06
2 channels:
3   - conda-forge
4 dependencies:
5   - python=3.12
6   - wat-inspector
7   - pip
8   - pip:
9     - "-e ."
10
```



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

在 pyproject.toml 配置文件里, 按照 文档 填写软件包的 构建 (build) 配置

使用 pip install -e . 以本地可编辑模式把当前软件包安装进当前 Conda 环境

```
DELL@DESKTOP-1SUNFG MINGW64 ~/repo/week06 (main)
$ pip install -e .
Looking in indexes: https://mirrors.tuna.tsinghua.edu.cn/pypi/web/simple
Obtaining file:///C:/Users/DELL/repo/week06
Installing build dependencies ... done
Checking if build backend supports build_editable ... done
Getting requirements to build editable ... done
Installing backend dependencies ... done
Preparing editable metadata (pyproject.toml) ... done
Collecting openpyxl (from mypackage==2025.4.14)
  Downloading https://mirrors.tuna.tsinghua.edu.cn/pypi/web/packages/c0/da/977ded879c29cbd04de313843e76868e6e13480a94ed6b987245dc7c8506/openpyxl-3.1.5-py2.py3-none-any.whl (250 kB)
Collecting et-xmlfile (from openpyxl->mypackage==2025.4.14)
  Downloading https://mirrors.tuna.tsinghua.edu.cn/pypi/web/packages/c1/8b/5fe2cc11fee489817272089c4203e679c63b570a5aaeb18d852ae3cbb6a/et_xmlfile-2.0.0-py3-none-any.whl (18 kB)
Building wheels for collected packages: mypackage
  Building editable for mypackage (pyproject.toml) ... done
  Created wheel for mypackage: filename=mypackage-2025.4.14-py3-none-any.whl size=7276 sha256=2b773119a98eee80a4c137fee60693614bbc6f7fa94e99af2b1a0342b8510aac
  Stored in directory: C:\Users\DELL\AppData\Local\Temp\pip-ephem-wheel-cache-75_csr5\wheels\78\26\3b\3047d49ce5791ac0f98411a5b5d9923295d59e0cde27b493e4
Successfully built mypackage
Installing collected packages: et-xmlfile, openpyxl, mypackage
Successfully installed et-xmlfile-2.0.0 mypackage-2025.4.14 openpyxl-3.1.5
(week06)
```

