

# 第六周学习报告

## 1.新建environment.yml 文件，创建 Conda 环境

```
(base)
hp@LAPTOP-L5E04S06 MINGW64 ~/repo/week06 (main)
$ cp ../week05/environment.yml ./
(base)
```

```
(base)
hp@LAPTOP-L5E04S06 MINGW64 ~/repo/week06 (main)
$ cat environment.yml
name: week05
channels:
  - conda-forge
done
#
# To activate this environment, use
#
#     $ conda activate week06
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(base)
```

## 查看环境下的软件

```
(base)
hp@LAPTOP-L5E04S06 MINGW64 ~/repo/week06 (main)
$ conda activate week06
(week06)
hp@LAPTOP-L5E04S06 MINGW64 ~/repo/week06 (main)
$ conda list
# packages in environment at D:\ANACONDA\envs\week06:
#
# Name                                Version           Build    Channel
bzip2                                1.0.8             h2466b09_7    conda-forge
ca-certificates                      2025.1.31         h56e8100_0    conda-forge
libexpat                             2.7.0             he0c23c2_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           h3f84c4b_0_cpython    conda-forge
setuptools                           78.1.0            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
vc14_runtime                         14.42.34438       hfd919c2_26    conda-forge
wat-inspector                        0.4.3             pyhff2d567_0    conda-forge
wheel                                0.45.1            pyhd8ed1ab_1    conda-forge
```

## 2. 创建 guessing\_game.py 文件，运用 pdb 调试器理解其运行流程

```
MINGW64/c/Users/hp/repo/v x + v
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。
> c:\users\hp\repo\week06\guessing_game.py(11)guessing_game()
-> while True:
(Pdb)
> c:\users\hp\repo\week06\guessing_game.py(12)guessing_game()
-> n += 1
(Pdb) n
> c:\users\hp\repo\week06\guessing_game.py(14)guessing_game()
-> 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): ")
(Pdb) p n
1
(Pdb) p secret_number
58
(Pdb) n
(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): 79
> c:\users\hp\repo\week06\guessing_game.py(15)guessing_game()
-> guess = guess.strip() # 去除多余空白字符
(Pdb) n
> c:\users\hp\repo\week06\guessing_game.py(17)guessing_game()
-> if guess == "q":
(Pdb) l
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         guess = guess.strip() # 去除多余空白字符
16
17 ->         if guess == "q":
18             break
19
20         try:
```

```
MINGW64/c/Users/hp/repo/v x + v
35         print("猜的数字太小了, 再试试｡｡。")
(Pdb) n
> c:\users\hp\repo\week06\guessing_game.py(34)guessing_game()
-> if guess < secret_number:
(Pdb) p guess
79
(Pdb) n
> c:\users\hp\repo\week06\guessing_game.py(38)guessing_game()
-> if guess > secret_number:
(Pdb) l
33
34         if guess < secret_number:
35             print("猜的数字太小了, 再试试｡｡。")
36             continue
37
38 ->         if guess > secret_number:
39             print("猜的数字太大了, 再试试｡｡。")
40             continue
41
42         raise NotImplementedError
43
(Pdb) n
> c:\users\hp\repo\week06\guessing_game.py(39)guessing_game()
-> print("猜的数字太大了, 再试试｡｡。")
(Pdb) n
猜的数字太大了, 再试试｡｡。
> c:\users\hp\repo\week06\guessing_game.py(40)guessing_game()
-> continue
(Pdb) n
> c:\users\hp\repo\week06\guessing_game.py(11)guessing_game()
```

```

MINGW64: c:\Users\hp\repo\w  +  v
-> while True:
(Pdb) n
> c:\users\hp\repo\week06\guessing_game.py(12)guessing_game()
-> n += 1
(Pdb)
> c:\users\hp\repo\week06\guessing_game.py(14)guessing_game()
-> 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): ")
(Pdb)
(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): 58
> c:\users\hp\repo\week06\guessing_game.py(15)guessing_game()
-> guess = guess.strip() # 去除多余空白字符
(Pdb)
> c:\users\hp\repo\week06\guessing_game.py(17)guessing_game()
-> if guess == "q":
(Pdb)
> c:\users\hp\repo\week06\guessing_game.py(20)guessing_game()
-> try:
(Pdb)
> c:\users\hp\repo\week06\guessing_game.py(21)guessing_game()
-> guess = int(guess)
(Pdb)
> c:\users\hp\repo\week06\guessing_game.py(26)guessing_game()
-> if guess < 1 or guess > 100:
(Pdb)
> c:\users\hp\repo\week06\guessing_game.py(30)guessing_game()
-> if guess == secret_number:
(Pdb)
> c:\users\hp\repo\week06\guessing_game.py(31)guessing_game()
-> print("恭喜你 🎉, 猜对了!")
(Pdb) █

```

```

(week06)
hp@LAPTOP-L5E04S06 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): 68
猜的数字太大了, 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): 47
恭喜你 🎉, 猜对了!
游戏结束, 再见 🙌。

```

```

hp@LAPTOP-L5E04S06 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): 67
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): 89
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): 75
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): 82
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): 85
猜的数字太小了, 再试试 🔄。
(第 6 次尝试) 请输入你猜的数字 (输入整数, 或者输入 q 回车退出): 88
猜的数字太大了, 再试试 🔄。
(第 7 次尝试) 请输入你猜的数字 (输入整数, 或者输入 q 回车退出): 86
猜的数字太小了, 再试试 🔄。
(第 8 次尝试) 请输入你猜的数字 (输入整数, 或者输入 q 回车退出): 87
恭喜你 🎉, 猜对了!
游戏结束, 再见 🙌。

```

### 3. 创建 flow\_controls.py 文件，理解 Python 流程控制语句

#### ❖ for 迭代循环 (iteration loop)

for 语句在 Python 里用于遍历可迭代对象，像列表、元组、字符串、字典等。

##### ➤ 遍历列表

```
flow_controls.py > ...
1  fruits = ["apple", "banana", "cherry"]
2  for fruit in fruits:
3      fruit += ".ok"
4      print(fruit)
5
```

```
apple.ok
banana.ok
cherry.ok
```

##### ➤ 遍历字符串

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

```
H
e
l
l
o
```

##### ➤ 遍历元组

```
numbers = (1, 2, 3, 4, 5)
for num in numbers:
    print(num * 2)
```

```
2
4
6
8
10
```

##### ➤ 遍历字典

```
student = {"name": "Alice", "age": 20, "major": "Computer Science"}

# 遍历字典的键
for key in student:
    print(key)

# 遍历字典的值
for value in student.values():
    print(value)

# 同时遍历字典的键和值
for key, value in student.items():
    print(f"{key}: {value}")
```

```
name
age
major
Alice
20
Computer Science
name: Alice
age: 20
major: Computer Science
```

➤ 使用 range() 函数

range(5) 会生成一个从 0 到 4 的整数序列，for 循环会依次遍历这个序列并打印每个整数。

```
for i in range(5):  
    print(i)
```

```
0  
1  
2  
3  
4
```

❖ while 条件循环 (conditional loop)

while 语句在 Python 中用于创建一个循环，只要指定的条件为真，就会重复执行一段代码块。

➤ 简单计数循环

```
count = 0  
while count < 5:  
    print(count)  
    count = count + 1
```

```
0  
1  
2  
3  
4
```

➤ 用户输入验证

```
valid_input = False  
while not valid_input:  
    user_input = input("请输入一个大于 10 的整数: ")  
    try:  
        num = int(user_input)  
        if num > 10:  
            valid_input = True  
            print("输入有效, 你输入的数字是:", num)  
        else:  
            print("输入无效, 请输入一个大于 10 的整数。")  
    except ValueError:  
        print("输入无效, 请输入一个有效的整数。")
```

```
请输入一个大于 10 的整数: 100  
输入有效, 你输入的数字是: 100
```

❖ break 打断跳出循环

在 Python 中，break 语句用于跳出当前所在的循环（for 循环或 while 循环），一旦执行了 break 语句，程序就会立即终止循环，并继续执行循环后面的代码。

➤ 在 while 循环结合条件判断使用 break

```

while True:
    user_input = input("请输入一个数字 (输入 q 退出) : ")
    if user_input == "q":
        break
    try:
        num = int(user_input)
        print(f"你输入的数字是 {num}")
    except ValueError:
        print("输入无效, 请输入一个数字或 q。")

```

```

请输入一个数字 (输入 q 退出) : 10
你输入的数字是 10
请输入一个数字 (输入 q 退出) : 18
你输入的数字是 18
请输入一个数字 (输入 q 退出) : q

```

#### ❖ continue 跳至下一轮循环

在 Python 中, continue 语句用于跳过当前循环中剩余的代码, 直接进入下一次循环的条件判断。

#### ❖ for...else 循环未被打断的处理

在 Python 中, for...else 结构里的 else 子句会在 for 循环正常结束 (也就是没有被 break 语句中断) 时执行。

#### ❖ if...elif[...elif] 多重条件分支

在 Python 里, if...elif...[elif]...else 结构能够实现多重条件分支, 让程序依据不同的条件执行不同的代码块。

#### ❖ if...else 未满足条件的处理

在 Python 中, if...else 语句用于根据条件的真假来执行不同的代码块。当 if 后面的条件不满足 (即条件表达式的结果为 False) 时, 程序会执行 else 子句中的代码。

```

# 获取玩家的得分
score = int(input("请输入你的游戏得分: "))

# 综合运用条件判断
if score < 0:
    print("得分无效, 得分不能为负数。")
elif score < 30:
    print("你的得分较低, 继续加油哦! 没有奖励。")
elif score < 60:
    print("你的表现一般, 奖励你一个小徽章。")
elif score < 80:
    print("你表现得不错, 奖励你一个精美钥匙链。")
elif score < 100:
    print("你太厉害了, 奖励你一张游戏点卡。")
elif score == 100:
    print("满分! 超级厉害, 奖励你一台游戏机。")
else:
    print("得分无效, 满分是 100 分。")

```

```

请输入你的游戏得分: 40
你的表现一般, 奖励你一个小徽章。

```

❖ try...except[...except...else...finally] 捕捉异常的处理

```
try:
    num1 = int(input("请输入被除数: "))
    num2 = int(input("请输入除数: "))
    result = num1 / num2
    print(f"结果是: {result}")
except ValueError:
    print("输入无效, 请输入有效的整数。")
except ZeroDivisionError:
    print("除数不能为零。")
else:
    print("运算成功完成。")
finally:
    print("程序结束。")
```

```
请输入被除数: 46
请输入除数: 5.6
输入无效, 请输入有效的整数。
程序结束。
```

❖ raise 主动抛出异常

```
class NegativeNumberError(Exception):
    pass

def calculate_square_root(num):
    if num < 0:
        raise NegativeNumberError("不能计算负数的平方根。")
    return num**0.5

try:
    result = calculate_square_root(-4)
    print(result)
except NegativeNumberError as e:
    print(f"捕获到异常: {e}")
```

#### 4. 创建 mylib.py 模块 (module) 和 myjob.py 脚本 (script)，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

```
mylib.py > ...  
1  def func1():  
2      x = 100  
3      y = x*0.5 + 3  
4      print(y)
```

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

```
hp@LAPTOP-L5E04S06 MINGW64 ~/repo/week06 (main)  
$ python myjob.py  
13.0
```

- ❖ 定义函数 func2，没有形参，有返回值

```
def func2():  
    x = 200  
    y = x*0.5 + 3  
    print(y)  
    return y
```

```
y = mylib.func2()  
print(y)
```

```
17.14213562373095  
17.14213562373095
```

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

```
def func3(x):  
    y = x*0.5 + 3  
    return y
```

```
y = mylib.func3(60)  
print(y)  
  
y = mylib.func3(x=6)  
print(y)
```

```
10.745966692414834  
5.449489742783178
```



- ❖ 定义函数 func4，只有一个命名形参 (named parameter)，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)

```
def func4(x=8):  
    y = x**0.5 + 3  
    return y
```

```
y = mylib.func4(44)  
print(y)  
  
y = mylib.func4(x=44)  
print(y)  
|  
y = mylib.func4()  
print(y)
```

```
9.6332495807108  
9.6332495807108  
5.82842712474619
```

- ❖ 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**

```
# 定义一个函数，包含位置形参和命名形参  
def describe_person(name, age, job="未指定", **other_info):  
    """  
    打印人物的信息  
    :param name: 位置形参，人物的姓名  
    :param age: 位置形参，人物的年龄  
    :param job: 命名形参，人物的工作，默认值为 "未指定"  
    :param other_info: 命名形参，收集其他任意的关键字参数  
    """  
    print(f"姓名: {name}")  
    print(f"年龄: {age}")  
    print(f"工作: {job}")  
    for key, value in other_info.items():  
        print(f"{key}: {value}")
```

```
# 使用位置实参调用函数  
print("使用位置实参调用函数:")  
print(mylib.describe_person("张三", 25, "教师"))  
  
# 使用命名实参调用函数  
print("\n使用命名实参调用函数:")  
print(mylib.describe_person(name="李四", age=30, job="工程师", 爱好="阅读"))  
  
# 混合使用位置实参和命名实参  
print("\n混合使用位置实参和命名实参:")  
print(mylib.describe_person("王五", 22, 爱好="篮球", 城市="北京"))
```

使用位置实参调用函数：

姓名：张三

年龄：25

工作：教师

None

使用命名实参调用函数：

姓名：李四

年龄：30

工作：工程师

爱好：阅读

None

混合使用位置实参和命名实参：

姓名：王五

年龄：22

工作：未指定

爱好：篮球

城市：北京

None

(Continued)

❖ 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

```
def func6(a, b, /, c=10, d=20):  
    """  
    该函数中 a 和 b 只能作为位置实参传入，c 和 d 可作为位置实参或命名实参传入  
    :param a: 仅接受位置实参  
    :param b: 仅接受位置实参  
    :param c: 可接受位置实参或命名实参，默认值为 10  
    :param d: 可接受位置实参或命名实参，默认值为 20  
    :return: a、b、c、d 四个参数的和  
    """  
    return a + b + c + d
```

❖ 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

```
def func7(a, b, *, c, d):  
    """  
    此函数中 a 和 b 可以通过位置或命名传递，c 和 d 只能通过命名传递  
    :param a: 可以位置或命名传递的参数  
    :param b: 可以位置或命名传递的参数  
    :param c: 只能命名传递的参数  
    :param d: 只能命名传递的参数  
    :return: a、b、c、d 四个参数的乘积  
    """  
    return a * b * c * d
```

```
# 正确调用，a、b 用位置传递，c、d 用命名传递  
result1 = mylib.func7(1, 2, c=3, d=4)  
print(f"a、b 位置传递，c、d 命名传递的结果: {result1}")
```

```
# 正确调用，a、b、c、d 都用命名传递  
result2 = mylib.func7(a=1, b=2, c=3, d=4)  
print(f"a、b、c、d 都用命名传递的结果: {result2}")
```

```
# 错误调用示例，会引发 TypeError
```

```
try:  
    mylib.func7(1, 2, 3, 4)  
except TypeError as e:  
    print(f"错误调用，引发异常: {e}")
```

```
a、b 位置传递，c、d 命名传递的结果：24
a、b、c、d 都用命名传递的结果：24
错误调用，引发异常：func7() takes 2 positional arguments but 4 were given
```

- ❖ 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)

```
def func8(a, b, *args):
    """
    函数接收两个固定位置形参 a 和 b，以及任意数量的额外位置实参
    :param a: 第一个位置形参
    :param b: 第二个位置形参
    :param args: 接收任意数量的额外位置实参，打包为元组
    :return: 所有参数的总和
    """
    total = a + b
    for num in args:
        total += num
    return total
```

```
# 调用函数，只传入两个位置实参
result1 = mylib.func8(1, 2)
print(f"只传入两个位置实参的结果：{result1}")

# 调用函数，传入两个位置实参和额外的位置实参
result2 = mylib.func8(1, 2, 3, 4, 5)
print(f"传入两个位置实参和额外位置实参的结果：{result2}")
```

```
只传入两个位置实参的结果：3
传入两个位置实参和额外位置实参的结果：15
```

- ❖ 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)

```
def func9(name, age, **kwargs):
    """
    该函数接收两个固定的命名形参 name 和 age，以及任意数量的额外命名实参
    :param name: 表示姓名的命名形参
    :param age: 表示年龄的命名形参
    :param kwargs: 接收任意数量的额外命名实参，打包为字典
    :return: 包含所有信息的字符串
    """
    info = f"姓名：{name}，年龄：{age}"
    for key, value in kwargs.items():
        info += f", {key}: {value}"
    return info
```

```
# 调用函数，只传入两个必需的命名实参
result1 = mylib.func9(name="张三", age=25)
print(result1)

# 调用函数，传入两个必需的命名实参和额外的命名实参
result2 = mylib.func9(name="李四", age=30, 职业="工程师", 爱好="阅读")
print(result2)
```

```
传入两个位置实参和默认位置实参的结果：13  
姓名：张三，年龄：25  
姓名：李四，年龄：30，职业：工程师，爱好：阅读
```

- ❖ 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

```
def func10(a, b, c=10):  
    return a + b + c
```

```
# 定义一个元组  
data_tuple = (1, 2)  
# 使用 * 解包元组作为位置实参传入函数  
result1 = mylib.func10(*data_tuple)  
print(f"使用元组解包传入位置实参的结果: {result1}")  
  
# 定义一个列表  
data_list = [3, 4]  
# 使用 * 解包列表作为位置实参传入函数，并指定命名形参 c 的值  
result2 = mylib.func10(*data_list, c=5)  
print(f"使用列表解包传入位置实参并指定命名形参的结果: {result2}")
```

```
使用元组解包传入位置实参的结果：13  
使用列表解包传入位置实参并指定命名形参的结果：12
```

- ❖ 定义函数 func11，接受一个命名形参，两个命名形参，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

```
def func11(name, age):  
    return f"姓名: {name}, 年龄: {age}"
```

```
# 定义一个字典  
person_info = {"name": "王五", "age": 28}  
  
# 使用 ** 解包字典作为命名实参传入函数  
result = mylib.func11(**person_info)  
print(result)
```

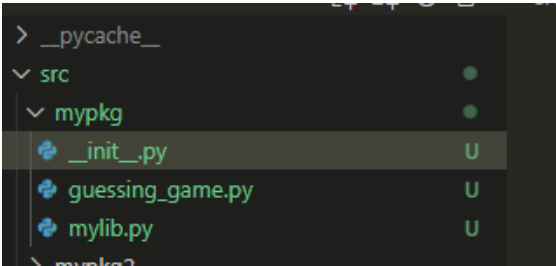
```
姓名：王五，年龄：28
```

- ❖ 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

```
def func12(a: int, b: int) -> int:  
    """该函数用于计算两个整数的和"""  
    return a + b
```

5.把 mylib 模块转变为 软件包 (package) 安装进当前的 Conda 环境来使用

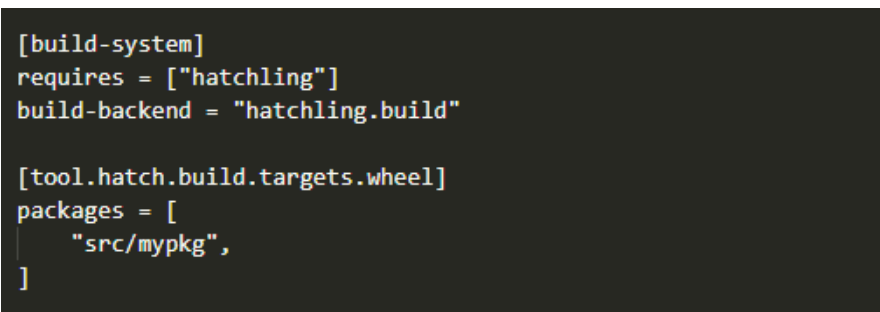
❖ 将 mylib.py 模块移动至 src/mypkg/mylib.py，创建 src/mypkg/\_\_init\_\_.py 文件



❖ 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

```
(week06)
hp@LAPTOP-L5E9J586 MINGW64 ~/repo/week06 (main)
$ pip install -e .
Looking in indexes: https://mirrors.tuna.tsinghua.edu.cn/pypi/web/simple
Obtaining file:///C:/Users/hp/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
Requirement already satisfied: django>2.0 in d:\anaconda\envs\week06\lib\site-packages (from mypackage==2025.4.14) (5.2)
Requirement already satisfied: gidgethub>4.0.0 in d:\anaconda\envs\week06\lib\site-packages (from gidgethub[httpx]>4.0.0->mypackage==2025.4.14) (5.3.0)
Requirement already satisfied: httpx in d:\anaconda\envs\week06\lib\site-packages (from mypackage==2025.4.14) (0.28.1)
Requirement already satisfied: asgiref>3.8.1 in d:\anaconda\envs\week06\lib\site-packages (from django>2.0->mypackage==2025.4.14) (3.8.1)
Requirement already satisfied: sqlparse>0.3.1 in d:\anaconda\envs\week06\lib\site-packages (from django>2.0->mypackage==2025.4.14) (0.6.3)
Requirement already satisfied: tzdata in d:\anaconda\envs\week06\lib\site-packages (from django>2.0->mypackage==2025.4.14) (2025.2)
Requirement already satisfied: uritemplate>3.0.1 in d:\anaconda\envs\week06\lib\site-packages (from gidgethub>4.0.0->gidgethub[httpx]>4.0.0->mypackage==2025.4.14) (4.1.1)
Requirement already satisfied: PyJWT[crypto]>2.4.0->gidgethub[httpx]>4.0.0->mypackage==2025.4.14) (2.10.1)
Requirement already satisfied: anyio in d:\anaconda\envs\week06\lib\site-packages (from httpx->mypackage==2025.4.14) (4.9.0)
Requirement already satisfied: certifi in d:\anaconda\envs\week06\lib\site-packages (from httpx->mypackage==2025.4.14) (2025.1.31)
Requirement already satisfied: httpcore>1.* in d:\anaconda\envs\week06\lib\site-packages (from httpx->mypackage==2025.4.14) (1.0.8)
Requirement already satisfied: idna in d:\anaconda\envs\week06\lib\site-packages (from httpx->mypackage==2025.4.14) (3.10)
Requirement already satisfied: h11>0.15, <=0.13 in d:\anaconda\envs\week06\lib\site-packages (from httpcore>1.*->httpx->mypackage==2025.4.14) (0.14.0)
Requirement already satisfied: cryptography>3.4.0 in d:\anaconda\envs\week06\lib\site-packages (from PyJWT[crypto]>2.4.0->gidgethub[httpx]>4.0.0->mypackage==2025.4.14) (44.0.2)
Requirement already satisfied: sniffio>1.1 in d:\anaconda\envs\week06\lib\site-packages (from anyio->httpx->mypackage==2025.4.14) (1.3.1)
Requirement already satisfied: typing_extensions>=4.5 in d:\anaconda\envs\week06\lib\site-packages (from anyio->httpx->mypackage==2025.4.14) (4.13.2)
Requirement already satisfied: cffi>1.12 in d:\anaconda\envs\week06\lib\site-packages (from cryptography>3.4.0->PyJWT[crypto]>2.4.0->gidgethub[httpx]>4.0.0->mypackage==2025.4.14) (1.17.1)
Requirement already satisfied: pycparser in d:\anaconda\envs\week06\lib\site-packages (from cffi>1.12->cryptography>3.4.0->PyJWT[crypto]>2.4.0->gidgethub[httpx]>4.0.0->mypackage==2025.4.14) (2.22)
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=7372 sha256=f7eee293b2dd19e1bb97ef1ed4fbfd78ca7b16f577e9778c28b522afc1fb51b
Stored in directory: C:\Users\hp\AppData\Local\Temp\pip-ephem-wheel-cache-u_hrpqul\wheels\26\95\ed\184341b0490943b9cabcc4dc59fc67f4951c87e956cf34814
Successfully built mypackage
Installing collected packages: mypackage
Attempting uninstall: mypackage
Found existing installation: mypackage 2025.4.14
Uninstalling mypackage-2025.4.14:
Successfully uninstalled mypackage-2025.4.14
Successfully installed mypackage-2025.4.14
```

## ❖ 修改 `environment.yml` 文件，使得 `conda env create` 自动安装本地可编辑软件包

```
! 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
```

```
MINGW64/C:/Users/hp/repo/v x + -
Collecting sniffio>=1.1 (from anyio->httpx->mypackage==2025.4.14->r C:\Users\hp\repo\week06\condaenv.11c9wrym.requirem)
Downloading https://mirrors.tuna.tsinghua.edu.cn/pypi/web/packages/e9/44/75a9c9421471a6c4805dbf7c2356f7c181a29c1879239a)
Collecting typing_extensions>=4.5 (from anyio->httpx->mypackage==2025.4.14->r C:\Users\hp\repo\week06\condaenv.11c9wry)
Downloading https://mirrors.tuna.tsinghua.edu.cn/pypi/web/packages/8b/54/b1ae86c0973cc6f0210b53d508ca3641fb6d0c56823f)
Collecting cffi>=1.12 (from cryptography>=3.4.0->PyJWT[crypto]>=2.4.0->gidgethub>4.0.0->gidgethub[httpx]>4.0.0->mypacka)
Downloading https://mirrors.tuna.tsinghua.edu.cn/pypi/web/packages/50/b9/db34c4755a7bd1cb2d1603ac3863f22bcecbd1ba29e5)
Collecting pycparser (from cffi>=1.12->cryptography>=3.4.0->PyJWT[crypto]>=2.4.0->gidgethub>4.0.0->gidgethub[httpx]>4.0.0)
Downloading https://mirrors.tuna.tsinghua.edu.cn/pypi/web/packages/13/a3/a812df4e2dd5696d1f351d58b8fe16a405b234ad2886)
Building wheels for collected packages: mypackage
Building editable for mypackage (pyproject.toml): started
Building editable for mypackage (pyproject.toml): finished with status 'done'
Created wheel for mypackage: filename=mypackage-2025.4.14-py3-none-any.whl size=7372 sha256=f7eee293b2dd19e1bb97ef1edb
Stored in directory: C:\Users\hp\AppData\Local\Temp\pip-ephem-wheel-cache-hyv7dqai\wheels\26\95\ed\184341b0490943b9ca4
Successfully built mypackage
Installing collected packages: uritemplate, tzdata, typing_extensions, sqlparse, sniffio, PyJWT, pycparser, idna, h11, e
Successfully installed PyJWT-2.10.1 anyio-4.9.0 asgiref-3.8.1 certifi-2025.1.31 cffi-1.17.1 cryptography-44.0.2 django-1
done
#
# To activate this environment, use
#
# $ conda activate week06
#
# To deactivate an active environment, use
#
# $ conda deactivate
(base)
hp@LAPTOP-L5E9J586 MINGW64 ~/repo/week06 (main)
$
```