

# 金融编程与计算-学习报告-week06

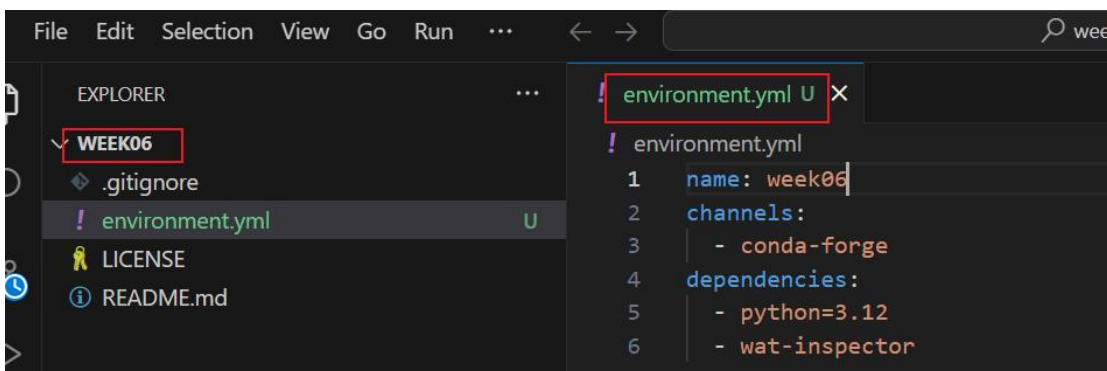
1. Fork 第 06 周打卡仓库至你的名下，然后将你名下的这个仓库 Clone 到你的本地计算机

```
1@DESKTOP-IUD6F9I MINGW64 ~/repo
$ git clone git@github.com:cherishdokyem/week06.git
Cloning into 'week06'...
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 | 279.00 KiB/s, done.
(base)
1@DESKTOP-IUD6F9I MINGW64 ~/repo
$ cd week06
(base)
1@DESKTOP-IUD6F9I MINGW64 ~/repo/week06 (main)
$ ls -l
total 24
-rw-r--r-- 1 1 197609 18805  4月 14 16:45 LICENSE
-rw-r--r-- 1 1 197609  2239  4月 14 16:45 README.md
(base)
```

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

将 week05 下的 environment.yml 复制到 week06，再进行相应修改。

```
$ cp ../week05/environment.yml ./
(base)
1@DESKTOP-IUD6F9I MINGW64 ~/repo/week06 (main)
$ ll
total 25
-rw-r--r-- 1 1 197609    91  4月 14 16:50 environment.yml
-rw-r--r-- 1 1 197609 18805  4月 14 16:45 LICENSE
-rw-r--r-- 1 1 197609  2239  4月 14 16:45 README.md
(base)
1@DESKTOP-IUD6F9I MINGW64 ~/repo/week06 (main)
```



```
File Edit Selection View Go Run ... < ->
EXPLORER
WEEK06
  .gitignore
  ! environment.yml U
  LICENSE
  README.md
environment.yml U
! environment.yml
1 name: week06
2 channels:
3   - conda-forge
4 dependencies:
5   - python=3.12
6   - wat-inspector
```

```
$ cat environment.yml
name: week06
channels:
  - conda-forge
dependencies:
  - python=3.12
  - wat-inspector(base)
1@DESKTOP-IUD6F9I MINGW64 ~/repo/week06 (main)
$ conda env create
Retrieving notices: done
Channels:
  - conda-forge
  - defaults
  - https://repo.anaconda.com/pkgs/main
  - https://repo.anaconda.com/pkgs/r
  - https://repo.anaconda.com/pkgs/msys2
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

Downloading and Extracting Packages:

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

```
(base)
1@DESKTOP-IUD6F9I MINGW64 ~/repo/week06 (main)
$ conda env list

# conda environments:
#
base                  * F:\biancheng\Anaconda
myproject             F:\biancheng\Anaconda\envs\myproject
prj11                 F:\biancheng\Anaconda\envs\prj11
prj2                  F:\biancheng\Anaconda\envs\prj2
week04                F:\biancheng\Anaconda\envs\week04
week05                F:\biancheng\Anaconda\envs\week05
week06                F:\biancheng\Anaconda\envs\week06

(base)
```

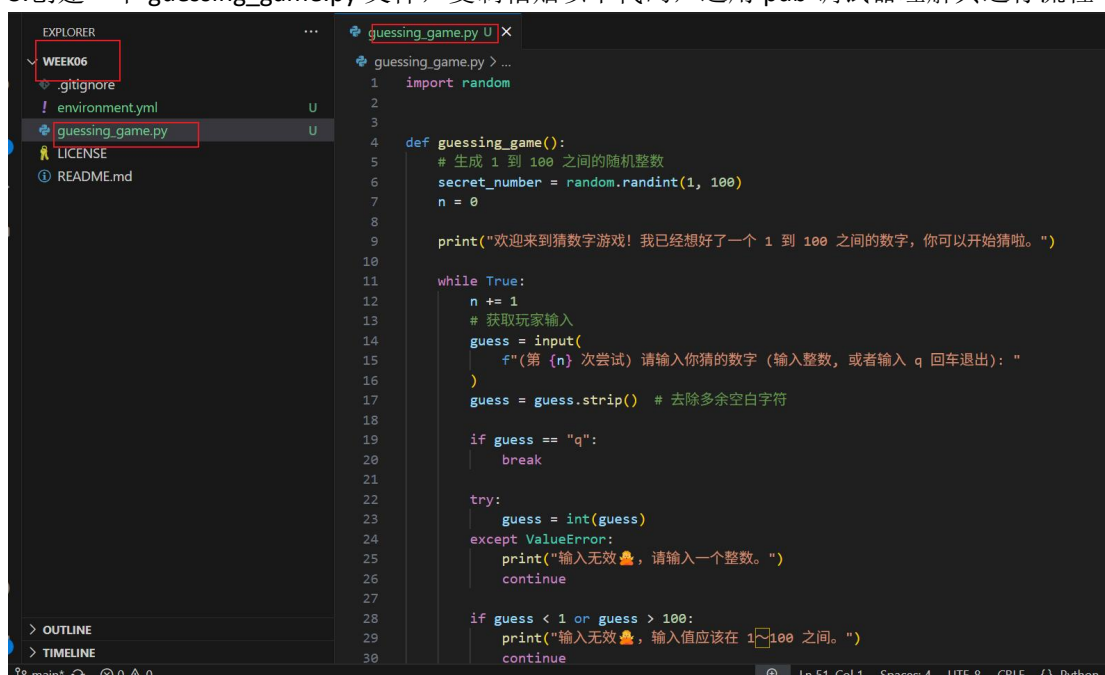
删除环境: `conda env remove -n prj11` (环境名字)

```

$ conda activate week06
(week06)
1@DESKTOP-IUD6F9I MINGW64 ~/repo/week06 (main)
$ conda list
# packages in environment at F:\biancheng\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
(week06)
1@DESKTOP-IUD6F9I 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



```

guessing_game.py
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:
12         n += 1
13         # 获取玩家输入
14         guess = input(
15             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): "
16         )
17         guess = guess.strip() # 去除多余空白字符
18
19         if guess == "q":
20             break
21
22         try:
23             guess = int(guess)
24         except ValueError:
25             print("输入无效，请输入一个整数。")
26             continue
27
28         if guess < 1 or guess > 100:
29             print("输入无效，输入值应该在 1~100 之间。")
30             continue

```

```

1@DESKTOP-IUD6F9I 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): 9
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): 50
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): 99
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): 8
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): 80
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): 90
猜的数字太小了，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): 95
猜的数字太大了，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): 93
猜的数字太大了，再试试。
(第 9 次尝试) 请输入你猜的数字 (输入整数，或者输入 q 回车退出): 92
恭喜你 🎉，猜对了！
游戏结束，再见 🙋。
(week06)
1@DESKTOP-IUD6F9I MINGW64 ~/repo/week06 (main)

```

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 Python 流程控制语句

```

1  # for
2
3  # 遍历列表
4  fruits = ["apple", "banana", "cherry"]
5  for fruit in fruits:
6      fruit += ",OK"
7      print(fruit)
8
9  # 遍历字符串
10 message = "Hello"
11 for char in message:
12     print(char)
13
14 # 结合range()函数
15 for i in range(5):
16     print(i)
17
18 # 遍历字典
19 person = {"name": "John", "age": 30, "city": "New York"}
20 for key in person: # person.keys() 键 person.values() 值
21     print(key, ":", person[key])
22
23 # while
24
25 # 简单计数循环
26 i = 0
27 while i < 5:
28     print(i)
29     i = i + 1

```

```
32 numbers = [1, 2, 3, 4, 5]
33 while numbers:
34     print(numbers.pop())
35
36 # 验证用户输入
37 valid_input = False
38 while not valid_input:
39     try:
40         num = int(input("请输入一个整数: "))
41         valid_input = True
42     except ValueError:
43         print("输入无效, 请输入一个整数。")
44
45 print("你输入的整数是:", num)
46
47 # if...elif...elif 多重条件分支
48 # 门票价格计算
49 age = 30
50 is_member = True
51
52 if age < 12:
53     price = 0 # 儿童免费
54 elif 12 <= age <= 65:
55     price = 20 if not is_member else 10 # 会员半价
56 else:
57     price = 15 # 老年人优惠
58
59 print(f"Ticket price: ${price}") # 输出: $10 (会员享受折扣)
60
```

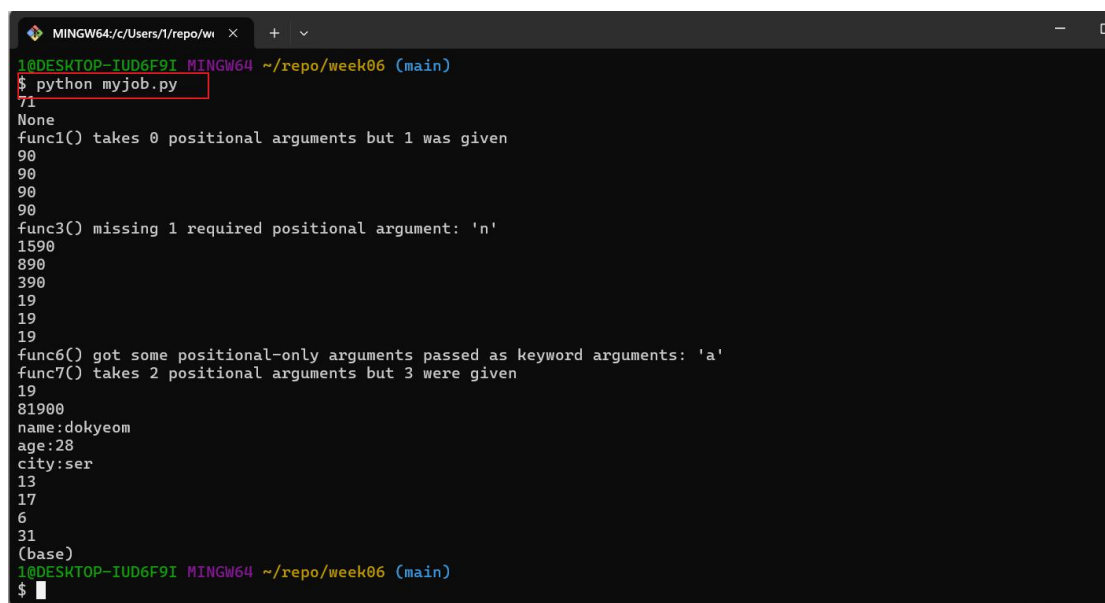
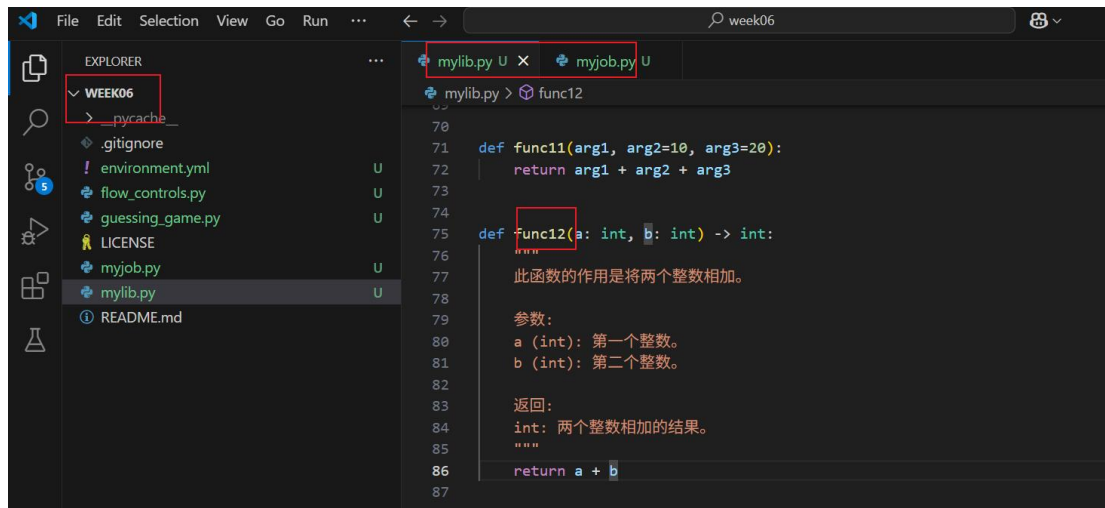
```
MINGW64:/c/Users/1/repo/wi X + v
cherry,OK
H
e
l
l
o
0
1
2
3
4
name : John
age : 30
city : New York
0
1
2
3
4
5
4
3
2
1
请输入一个整数: 1
你输入的整数是: 1
Ticket price: $10
(week06)
1@DESKTOP-IUD6F9I MINGW64 ~/repo/week06 (main)
$
```

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

```
EXPLORER
WEEK06
  __pycache__
  .gitignore
  environment.yml
  flow_controls.py
  guessing_game.py
  LICENSE
  myjob.py
  mylib.py
  README.md

mylib.py
1 def func1():
2     m = 9
3     n = m**2 - 10
4     print(n)
5     # 相当于return None
6
7
8 def func2():
9     m = 10
10    n = m**2 - 10
11    print(n)
12    return n # return直接跳出
13
14
15 def func3(n):
16     m = n**2 - 10
17     return m
18
19
20 def func4(n=20):
21     m = n**2 - 10
22     return m
23
24
25 def cherish(a, b, dokyem="lovely"): # 位置参数必须排在命名参数之前
26     if dokyem == "lovely":
27         return a + b
28     elif dokyem == "cute":
29         return a - b
30
```





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

```
MINGW64:/c/Users/1/repo/wn
$ cat pyproject.toml
[project]
name = "mypackage"
version = "2025.4.14"
dependencies = [
    "openpyxl",
]
authors = [
    {name = "DK", email = "DK@163.com"},
]
description = "测试用的软件包"

[project.optional-dependencies]
dev = [
    "pytest",
]

[build-system]
requires = ["hatchling"]
build-backend = "hatchling.build"

[tool.hatch.build.targets.wheel]
packages = ["src/foo"](base)
1@DESKTOP-IUD6F9I MINGW64 ~/repo/week06 (main)
$ pip install -e .
Looking in indexes: https://mirrors.tuna.tsinghua.edu.cn/pypi/web/simple
Obtaining file:///C:/Users/1/repo/week06
Installing build dependencies ... done
Checking if build backend supports build_editable ... done
Getting requirements to build editable ... done
```

