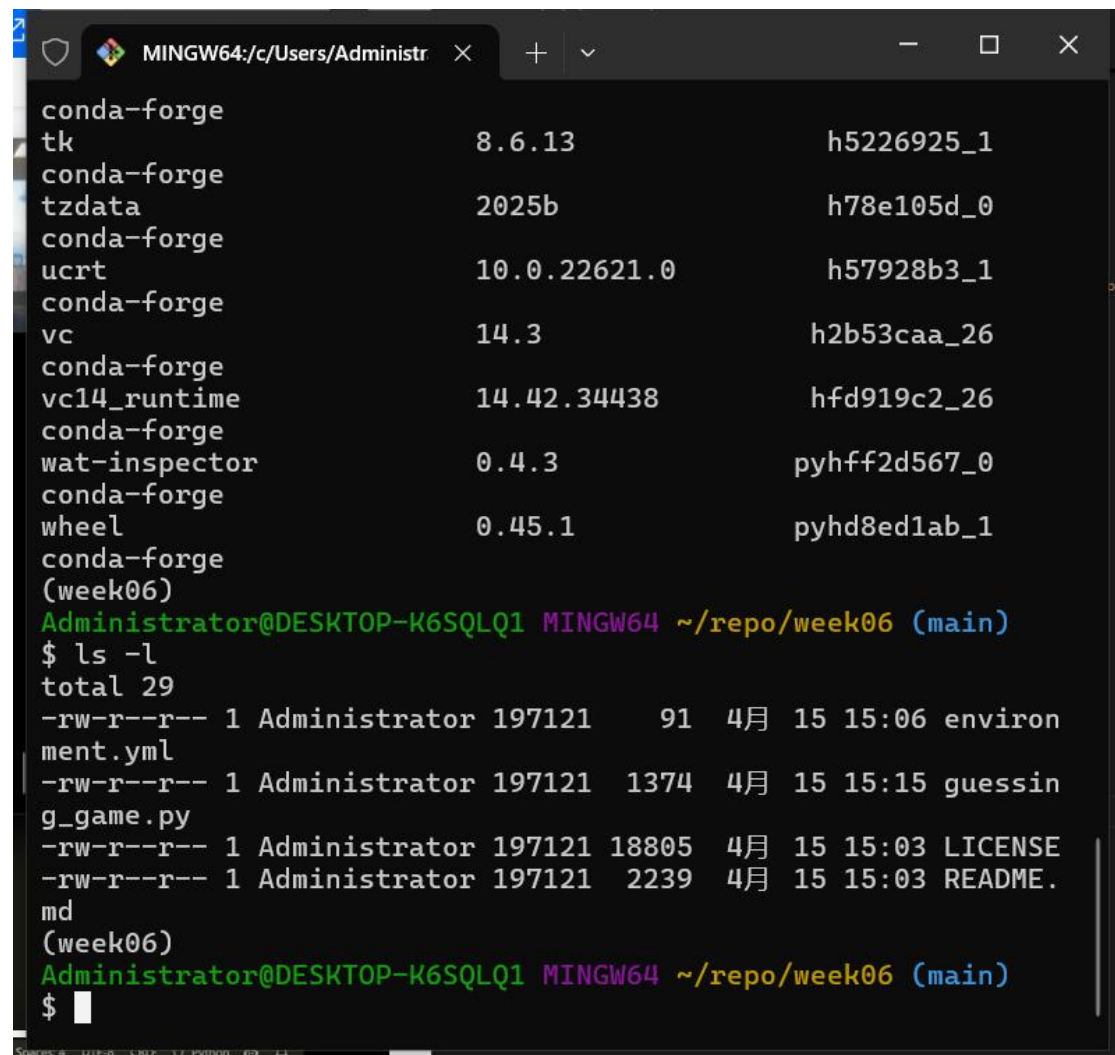


第六周笔记作业

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
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
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：



```
MINGW64:/c/Users/Administr
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)
Administrator@DESKTOP-K6SQLQ1 MINGW64 ~/repo/week06 (main)
$ ls -l
total 29
-rw-r--r-- 1 Administrator 197121   91  4月 15 15:06 environ
ment.yml
-rw-r--r-- 1 Administrator 197121 1374  4月 15 15:15 guessin
g_game.py
-rw-r--r-- 1 Administrator 197121 18805  4月 15 15:03 LICENSE
-rw-r--r-- 1 Administrator 197121  2239  4月 15 15:03 README.
md
(week06)
Administrator@DESKTOP-K6SQLQ1 MINGW64 ~/repo/week06 (main)
$
```

```
MINGW64/c/Users/Administr
+ v
Try 'python -h' for more information.
(week06)
Administrator@DESKTOP-K6SQLQ1 MINGW64 ~/repo/week06 (main)
$ python -m pdb guessing_game.py
> c:\users\administrator\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\administrator\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)
```

```
50         guessing_game()
[EOF]
(Pdb) n
> c:\users\administrator\repo\week06\guessing_game.py(50)<module>()
-> guessing_game()
(Pdb) q
(week06)
Administrator@DESKTOP-K6SQLQ1 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): 30
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): 80
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): 60
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): 70
猜的数字太大了，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): 65
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): 63
恭喜你 🎉，猜对了！
游戏结束，再见 🙋。
(week06)
Administrator@DESKTOP-K6SQLQ1 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，体会理解以下 Python 流程控制语句：

```
1  # for语句
2  fruits = ["pear", "grape", "lemon"]
3  for fruit in fruits:
4      print(fruit)
5
6  fruits = ["pear", "grape", "lemon"]
7  for fruit in fruits:
8      fruit += ", good"
9      print(fruit) # 遍历列表
10
11 message = "Python"
12 for char in message:
13     print(char) # 遍历字符串
14
15 for i in range(7):
16     print(i) # 结合range函数
17
18 student = {"name": "Bob", "age": 22, "grade": "B"}
19 for key, value in student.items():
20     print(f"{key}: {value}") # 遍历字典
21
22 student = {"name": "Bob", "age": 22, "grade": "B"}
23 for value in student.values():
24     print(value) # 遍历字典
25
26 # while语句
27 count = 3
28 while count < 8:
29     print(count)
30     count = count + 1 # 简单计数
31
32 numbers = [6, 7, 8, 9, 10]
33 while numbers:
```

Ln 210, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.12.7 ('base': conda)

```
Administrator@DESKTOP-K6SQLQ1 MINGW64 ~/repo/week06 (main)
$ python -m pdb flow_controls.py
> c:\users\administrator\repo\week06\flow_controls.py(2)<module>()
-> fruits = ["pear", "grape", "lemon"]
(Pdb) l
1      # for语句
2  ->  fruits = ["pear", "grape", "lemon"]
3      for fruit in fruits:
4          print(fruit)
5
6      fruits = ["pear", "grape", "lemon"]
7      for fruit in fruits:
8          fruit += ", good"
9          print(fruit) # 遍历列表
10
11     message = "Python"
(Pdb) n
> c:\users\administrator\repo\week06\flow_controls.py(3)<module>()
-> for fruit in fruits:
(Pdb) l
1      # for语句
2      fruits = ["pear", "grape", "lemon"]
3  ->  for fruit in fruits:
4          print(fruit)
5
6      fruits = ["pear", "grape", "lemon"]
7      for fruit in fruits:
8          fruit += ", good"
9          print(fruit) # 遍历列表
```

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 def func1():
2     x = 60 # 修改 x 的值为 60
3     y = x**0.8 - 82
4     print(y)
5
6
7 def func2():
8     x = 120 # 修改 x 的值为 120
9     y = x**0.8 - 82
10    print(y)
11    return y
12
13
14 def func3(x):
15     y = x**0.8 - 82
16     return y
17
18
19 def func4(x=800): # 修改默认参数值为 800
20     y = x**0.8 - 82
21     return y
22
23
24 # 定义一个函数, 包含位置形参和命名形参
25 def calculate_area(length, width=15): # 修改 width 默认值为 15
26     return length * width
27
28
29 # 调用函数, 使用位置实参和命名实参
30 area1 = calculate_area(6) # 使用位置实参传递 length 为 6, width 使用默认
31 area2 = calculate_area(length=8, width=4) # 使用命名实参传递参数
32
33
```

```
1 import mylib # noqa: F401
2
3 y = mylib.func1()
4 print(y)
5
6
7 try:
8     y = mylib.func1(1) # 修改传入的参数为 1
9 except TypeError as e:
10    print(e)
11
12 y = mylib.func2()
13 print(y)
14
15 y = mylib.func3(60) # 修改传入的参数为 60
16 print(y)
17 y = mylib.func3(x=80) # 修改传入的参数为 80
18 print(y)
19 try:
20     y = mylib.func3()
21 except TypeError as e:
22     print(e)
23
24 y = mylib.func4(25) # 修改传入的参数为 25
25 print(y)
26 y = mylib.func4(x=90) # 修改传入的参数为 90
27 print(y)
28 y = mylib.func4()
29 print(y)
30
31 print(mylib.area1)
32 print(mylib.area2)
33
```

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

```
packages = ["src/mypkg"](week06)
Administrator@DESKTOP-K6SQLQ1 MINGW64 ~/repo/week06 (main)
$ ls -l
total 34
-rw-r--r-- 1 Administrator 197121 124 4月 15 15:57 environment.yml
-rw-r--r-- 1 Administrator 197121 5362 4月 15 15:45 flow_controls.py
-rw-r--r-- 1 Administrator 197121 18805 4月 15 15:03 LICENSE
-rw-r--r-- 1 Administrator 197121 273 4月 15 16:26 pyproject.toml.de
l
-rw-r--r-- 1 Administrator 197121 2239 4月 15 15:03 README.md
drwxr-xr-x 1 Administrator 197121 0 4月 15 16:08 scripts/
(week06)
Administrator@DESKTOP-K6SQLQ1 MINGW64 ~/repo/week06 (main)
$ pip install --editable .
Looking in indexes: https://mirrors.tuna.tsinghua.edu.cn/pypi/web/simpl
e
Obtaining file:///C:/Users/Administrator/repo/week06
ERROR: file:///C:/Users/Administrator/repo/week06 does not appear to be
a Python project: neither 'setup.py' nor 'pyproject.toml' found.
(week06)
Administrator@DESKTOP-K6SQLQ1 MINGW64 ~/repo/week06 (main)
$ mv pyproject.toml.del pyproject.toml
(week06)
Administrator@DESKTOP-K6SQLQ1 MINGW64 ~/repo/week06 (main)
$ ls -l
total 34
-rw-r--r-- 1 Administrator 197121 124 4月 15 15:57 environment.yml
-rw-r--r-- 1 Administrator 197121 5362 4月 15 15:45 flow_controls.py
-rw-r--r-- 1 Administrator 197121 18805 4月 15 15:03 LICENSE
-rw-r--r-- 1 Administrator 197121 273 4月 15 16:26 pyproject.toml
-rw-r--r-- 1 Administrator 197121 2239 4月 15 15:03 README.md
drwxr-xr-x 1 Administrator 197121 0 4月 15 16:08 scripts/
(week06)
```

The screenshot shows the Visual Studio Code editor interface. On the left, the Explorer sidebar displays the project structure for 'WEEK06', including files like .vscode, mypackage.egg-info, scripts, src, mypkg, mylib.py, mypkg2, myjob.py, .gitignore, environment.yml, flow_controls.py, LICENSE, pyproject.toml, and README.md. The main editor window is open to 'pyproject.toml', showing the following content:

```
1 [project]
2 name = "mypackage"
3 version = "2025.4.14"
4 dependencies = [
5     "openpyxl",
6 ]
7 authors = [
8     {name = "Ting", email = "2141619615@qq.com"},
9 ]
10 description = "测试一下"
11
12 [project.optional-dependencies]
13 dev = [
14     "pytest",
15 ]
16
```

The bottom status bar indicates the time is 16:43 on 2025/4/15.

The screenshot shows a terminal window with the following output:

```
Successfully built mypackage
Installing collected packages: mypackage
Successfully installed mypackage-2025.4.14
(week06)
Administrator@DESKTOP-K6SQLQ1 MINGW64 ~/repo/week06 (main)
$ conda list
# packages in environment at D:\haozhiting\Anaconda3\envs\week06:
#
# Name                    Version           Build    Channel
bzip2                     1.0.8             h2466b09_7  conda-forge
ca-certificates           2025.1.31         h56e8100_0  conda-forge
et-xmlfile                2.0.0             pypi_0     pypi
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
mypackage                 2025.4.14         pypi_0     pypi
openpyxl                  3.1.5             pypi_0     pypi
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)
Administrator@DESKTOP-K6SQLQ1 MINGW64 ~/repo/week06 (main)
$
```