

第 6 周学习笔记

1. 创建环境

```
(base) Administrator@MICROSO-J56DDR4 MINGW
$ cat environment.yml
name: week05
channels:
  - conda-forge
dependencies:
  - python=3.12
  - wat-inspector
(base) Administrator@MICROSO-J56DDR4 MINGW
$ cat environment.yml
name: week06
channels:
done
#
# To activate this environment, use
#
#     $ conda activate week06
#
# To deactivate an active environment, use
#
#     $ conda deactivate
```

2. 猜数字游戏

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

3. Python 流程控制语句

3.1 for 循环例子

```
flow_controls.py > ...
1  fruits = ['apple', 'banana', 'cherry']
2  for fruit in fruits:
3      fruit = fruit + ", ok"
4      print(fruit)
5
6  message = "Hello"
7  for char in message:
8      print(char)
9
10 for i in range(5):
11     print(i)
12
13 person = {'name': 'Alice', 'age': 25, 'city': 'New York'}
14 for key, value in person.items():
15     print(f"{key}: {value}")
```

```
(week06) Administrator@MICR
$ python flow_controls.py
apple, ok
banana, ok
cherry, ok
H
e
l
l
o
0
1
2
3
4
name: Alice
age: 25
city: New York
```

3.2 while 循环

```
17 count = 0
18 while count < 5:
19     print(count)
20     count = count + 1
21
22 numbers = [1,2,3,4,5]
23 while numbers:
24     print(numbers.pop())
```

```
0
1
2
3
4
5
4
3
2
1
```

4. 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:

`mylib.py`

```
mylib.py > func12
1  def func1():
2      x = 50
3      y = x**0.5 - 7
4      print(y)
5
6  def func2():
7      x = 70
8      y = x**0.5 - 7
9      print(y)
10     return y
11
12 def func3(x):
13     y = x**0.5 - 7
14     return y
15
16 def func4(x=50):
17     y = x**0.5 - 7
18     return y
19
20 def calculate(a,b,operation="add"):
21     if operation == "add":
22         return a + b
23     elif operation == "subtract":
24         return a - b
25     else:
26         return None
27
28 def func6(a,/,b,operation="add"):
29     if operation == "add":
30         return a + b
```

```

31 def func7(a,/,b,*,operation="add"):
32     if operation == "add":
33         return a + b
34     elif operation == "subtract":
35         return a - b
36     else:
37         return None
38
39 def func8(*numbers):
40     total=0
41     for num in numbers:
42         total = total + num
43     return total
44
45 def func9(**user):
46     for key, value in user.items():
47         print(f"{key}:{value}")
48
49 def func10(arg1, arg2, named_arg="default"):
50     print(f"位置实参 arg1:{arg1}")
51     print(f"位置实参 arg2:{arg2}")
52     print(f"命名实参 named_arg:{named_arg}")
53
54 def func12(arg1:str, arg2:int, named_arg:str = "default") -> None:
55     print(f"位置实参 arg1:{arg1}")
56     print(f"位置实参 arg2:{arg2}")
57     print(f"命名实参 named_arg:{named_arg}")
58

```

myjob.py

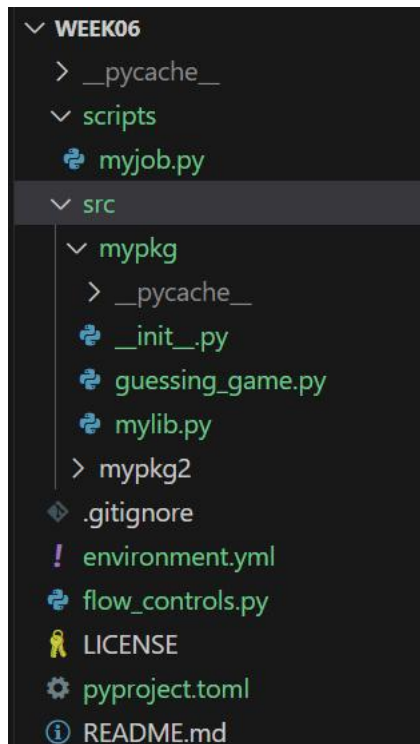
```

31 print(mylib.calculate(10,5,"add"))
32 print(mylib.calculate(operation="add",b=5,a=10))
33 print(mylib.calculate(5,8,"subtract"))
34
35 print(mylib.func6(a=10,b=5))
36 print(mylib.func7(10,5,operation="subtract"))
37 print(mylib.func8(4,8))
38
39 mylib.func9(name="Alice",age=25,city="New York")
40
41 tuple_args = (10,20)
42 mylib.func10(*tuple_args)
43
44 list_args = (30,40)
45 mylib.func10(*list_args)
46
47 mylib.func12(7,8,9)

```

```
myjob.py > ...
1  import mylib
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
11 y = mylib.func2()
12 print(y)
13 y = mylib.func3(45)
14 print(y)
15 y = mylib.func3(x=47)
16 print(y)
17
18 try:
19     y = mylib.func3()
20 except TypeError as e:
21     print(e)
22
23 try:
24     mylib.func3(y=47)
25 except TypeError as e:
26     print(e)
27
28 y = mylib.func4(x=49)
29 print(y)
```

5. 把 myjob.py 脚本移动至 scripts/myjob.py, 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



6. 填写基本的软件包信息，构建配置

```
pyproject.toml
1  [project]
2  name = "mypackage"
3  version = "2025.4.18"
4  dependencies = [
5  |   "openpyxl",
6  | ]
7  authors = [
8  |   {name = "Jiawei", email = "pradyun@example.com"},
9  | ]
10 description = "测试用的软件包"
11
12 [project.optional-dependencies]
13 dev = [
14 |   "pytest",
15 | ]
16
17 [build-system]
18 requires = ["hatchling"]
19 build-backend = "hatchling.build"
20
21 [tool.hatch.build.targets.wheel]
22 packages = [
23 |   "src/mypkg",
24 | ]
```


7. 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) Administrator@MICROSO-J56DDR4 MINGW64 ~/repo/week06 (main)
$ pip install -e .
Looking in indexes: https://mirrors.tuna.tsinghua.edu.cn/pypi/web/s
Obtaining file:///C:/Users/Administrator/repo/week06
```

8. 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 ."
```

```
(base) Administrator@MICROSO-J56D
$ conda env list
# conda environments:
#
```