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 MIN
$  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 > ...

```python
import random


def guessing_game():
    # 生成 1 到 100 之间的随机整数
    secret_number = random.randint(1, 100)
    n = 0

    print("欢迎来到猜数字游戏！我已经想好了一个 1 到 100 之间的

    while True:
        n += 1
        # 获取玩家输入
        guess = input(f"(第 {n} 次尝试) 请输入你猜的数字 (输入
        guess = guess.strip()  # 去除多余空白字符

        if guess == "q":
            break

        try:
            guess = int(guess)
        except ValueError:
            print("输入无效🤦, 请输入一个整数。")
            continue

        if guess < 1 or guess > 100:
            print("输入无效🤦, 输入值应该在 1~100 之间。")
            continue
```

3. Python 流程控制语句

3.1for 循环例子

flow_controls.py > ...

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

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

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

person = {'name': 'Alice', 'age': 25, 'city': 'New York'}
for key, value in person.items():
    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
```

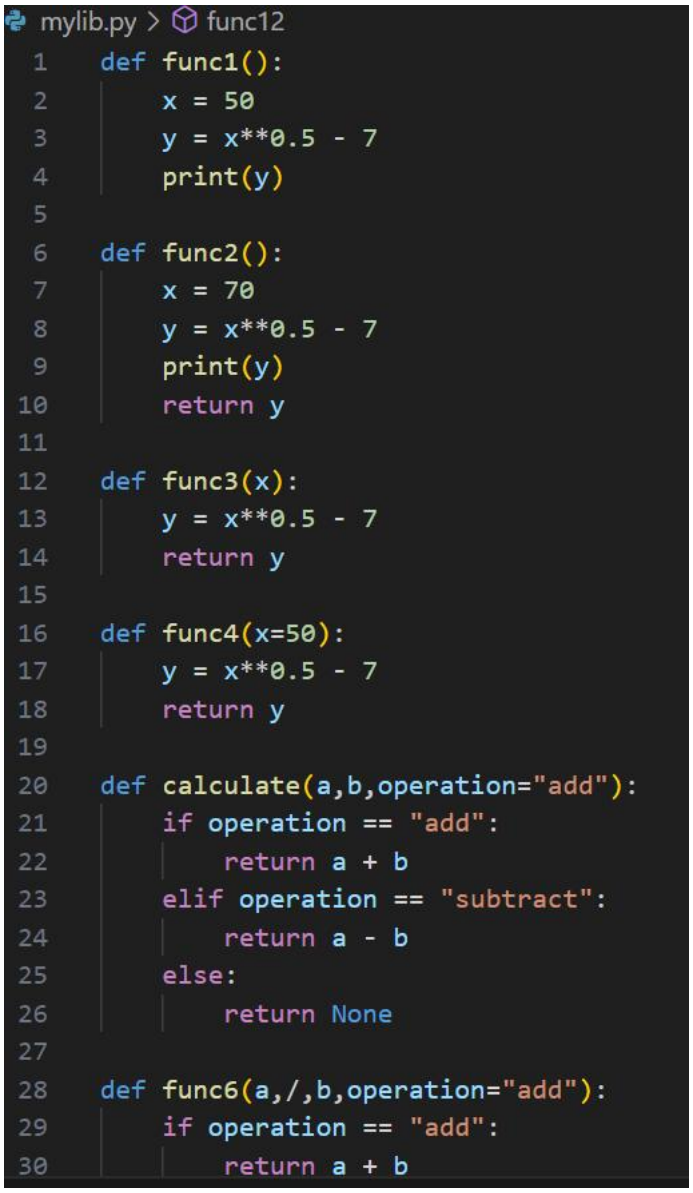
3.2 while 循环

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

numbers = [1,2,3,4,5]
while numbers:
    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



```python
def func7(a,/,b,*,operation="add"):
    if operation == "add":
        return a + b
    elif operation == "subtract":
        return a - b
    else:
        return None

def func8(*numbers):
    total=0
    for num in numbers:
        total = total + num
    return total

def func9(**user):
    for key, value in user.items():
        print(f"{key}:{value}")

def func10(arg1, arg2, named_arg="default"):
    print(f"位置实参 arg1:{arg1}")
    print(f"位置实参 arg2:{arg2}")
    print(f"命名实参 named_arg:{named_arg}")

def func12(arg1:str, arg2:int, named_arg:str = "default") -> None:
    print(f"位置实参 arg1:{arg1}")
    print(f"位置实参 arg2:{arg2}")
    print(f"命名实参 named_arg:{named_arg}")
```

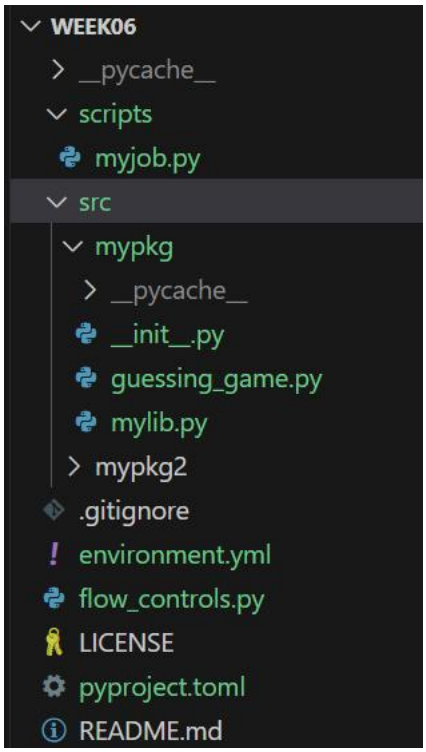
myjob.py

```python
print(mylib.calculate(10,5,"add"))
print(mylib.calculate(operation="add",b=5,a=10))
print(mylib.calculate(5,8,"subtract"))

print(mylib.func6(a=10,b=5))
print(mylib.func7(10,5,operation="subtract"))
print(mylib.func8(4,8))

mylib.func9(name="Alice",age=25,city="New York")

tuple_args = (10,20)
mylib.func10(*tuple_args)

list_args = (30,40)
mylib.func10(*list_args)

mylib.func12(7,8,9)
```

myjob.py > ...

```python
import mylib

y = mylib.func1()
print(y)

try:
    y = mylib.func1(0)
except TypeError as e:
    print(e)

y = mylib.func2()
print(y)
y = mylib.func3(45)
print(y)
y = mylib.func3(x=47)
print(y)

try:
    y = mylib.func3()
except TypeError as e:
    print(e)

try:
    mylib.func3(y=47)
except TypeError as e:
    print(e)

y = mylib.func4(x=49)
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

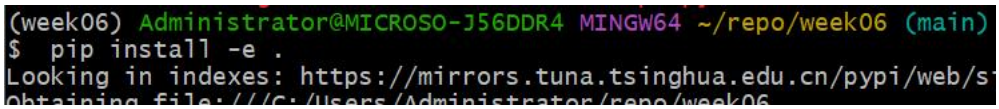


6. 填写基本的软件包信息，构建配置

pyproject.toml

```toml
[project]
name = "mypackage"
version = "2025.4.18"
dependencies = [
  "openpyxl",
]
authors = [
  {name = "Jiawei", email = "pradyun@example.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/mypkg",
]
```

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



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