

## 第六周作业

一、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

```
(base) FAN0917@LAPTOP-7PDVKG40 MINGW64 ~  
$ cd repo  
  
(base) FAN0917@LAPTOP-7PDVKG40 MINGW64 ~/repo  
$ ll  
total 25  
drwxr-xr-x 1 FAN0917 197121 0 3月 18 23:39 myproject/  
drwxr-xr-x 1 FAN0917 197121 0 3月 14 21:03 mywork/  
drwxr-xr-x 1 FAN0917 197121 0 3月 18 22:52 prj1/  
-rw-r--r-- 1 FAN0917 197121 374 3月 9 11:42 script1.py  
drwxr-xr-x 1 FAN0917 197121 0 3月 9 20:53 week01/  
drwxr-xr-x 1 FAN0917 197121 0 3月 14 21:27 week02/  
drwxr-xr-x 1 FAN0917 197121 0 3月 19 00:02 week03/  
drwxr-xr-x 1 FAN0917 197121 0 3月 28 13:12 week04/  
drwxr-xr-x 1 FAN0917 197121 0 4月 8 10:44 week05/  
  
(base) FAN0917@LAPTOP-7PDVKG40 MINGW64 ~/repo  
$ git clone git@gitcode.com:jiemoduner/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 | 1.41 MiB/s, done.
```

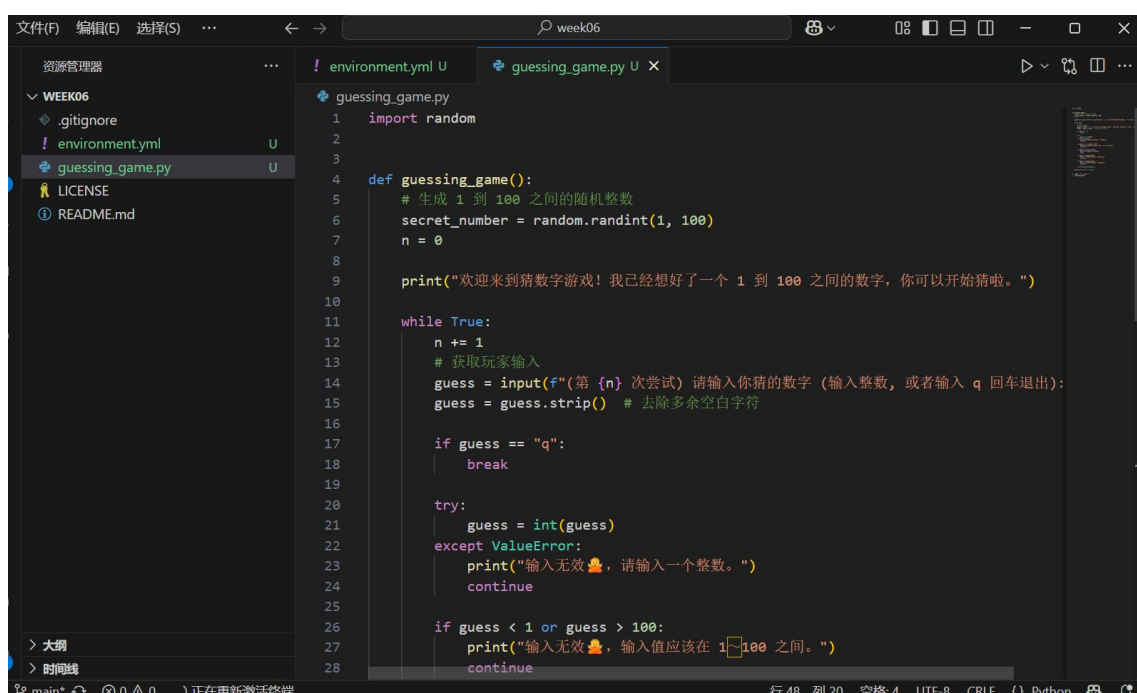
## 二、用 VS Code 打开项目目录,新建一个 environment.yml 文件,指定安装 Python 3.12, 然后运行 conda env create 命令创建 Conda 环境

```
(base) FAN0917@LAPTOP-7PDVKG40 MINGW64 ~/repo/week06 (main)
$ cp ../week05/environment.yml ./

(base) FAN0917@LAPTOP-7PDVKG40 MINGW64 ~/repo/week06 (main)
$ ll
total 25
-rw-r--r-- 1 FAN0917 197121 91 4月 15 19:52 environment.yml
-rw-r--r-- 1 FAN0917 197121 18805 4月 15 19:51 LICENSE
done
#
# To activate this environment, use
#
#     $ conda activate week06
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(base) FAN0917@LAPTOP-7PDVKG40 MINGW64 ~/repo/week06 (main)
$ conda env list
# conda environments:
#
myproject          C:\Users\FAN0917\.conda\envs\myproject
prj2               C:\Users\FAN0917\.conda\envs\prj2
week04            C:\Users\FAN0917\.conda\envs\week04
week05            C:\Users\FAN0917\.conda\envs\week05
week06            C:\Users\FAN0917\.conda\envs\week06
base              * D:\GQ\anaconda3
```

## 三、创建一个 guessing\_game.py 文件,运用 pdb 调试器理解其运行流程



```
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
10    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。")
11
12    while True:
13        n += 1
14        # 获取玩家输入
15        guess = input(f"({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        guess = guess.strip() # 去除多余空白字符
17
18        if guess == "q":
19            break
20
21        try:
22            guess = int(guess)
23        except ValueError:
24            print("输入无效, 请输入一个整数。")
25            continue
26
27        if guess < 1 or guess > 100:
28            print("输入无效, 输入值应该在 1-100 之间。")
29            continue
```

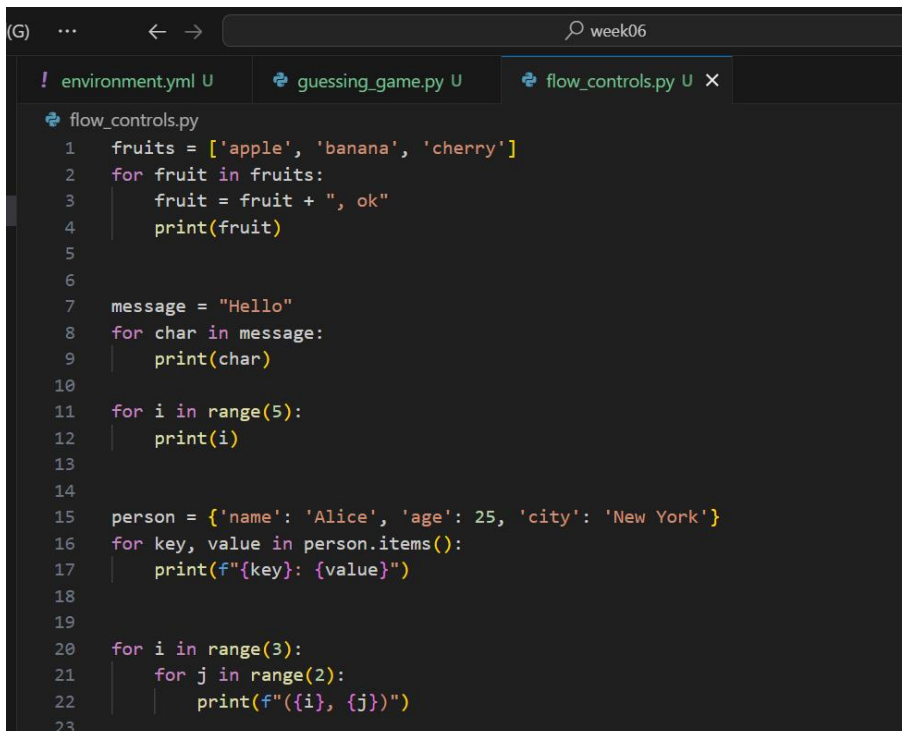
```

FAN0917@LAPTOP-7PDVKG40 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): b
输入无效 🍌，请输入一个整数。
(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): aaa
输入无效 🍌，请输入一个整数。
(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): 33
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): 56
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): 78
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): q
游戏结束，再见 🍌。
(week06)
FAN0917@LAPTOP-7PDVKG40 MINGW64 ~/repo/week06 (main)
$

```

#### 四、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（此处以 for, while 为例）

for 迭代循环 (iteration loop)



```

(G) ... week06
! environment.yml U guessing_game.py U flow_controls.py U X
flow_controls.py
1  fruits = ['apple', 'banana', 'cherry']
2  for fruit in fruits:
3      fruit = fruit + ", ok"
4      print(fruit)
5
6
7  message = "Hello"
8  for char in message:
9      print(char)
10
11 for i in range(5):
12     print(i)
13
14
15 person = {'name': 'Alice', 'age': 25, 'city': 'New York'}
16 for key, value in person.items():
17     print(f"{key}: {value}")
18
19
20 for i in range(3):
21     for j in range(2):
22         print(f"({i}, {j})")
23

```

while 条件循环 (conditional loop)

```

25 count = 0
26 while count < 5:
27     print(count)
28     count = count + 1
29
30
31 numbers = [1, 2, 3, 4, 5]
32 while numbers:
33     print(numbers.pop())
34
35
36 valid_input = False
37 while not valid_input:
38     user_input = input("请输入一个大于 10 的数字: ")
39     try:
40         num = int(user_input)
41         if num > 10:
42             valid_input = True
43             print("输入有效! ")
44         else:
45             print("输入的数字不大于 10, 请重新输入。")
46     except ValueError:
47         print("输入不是有效的数字, 请重新输入。")
48
49
50 while True:
51     user_choice = input("输入 'q' 退出: ")
52     if user_choice == 'q':
53         break
54     print("你输入的不是 'q', 请继续。")

```

运行结果:

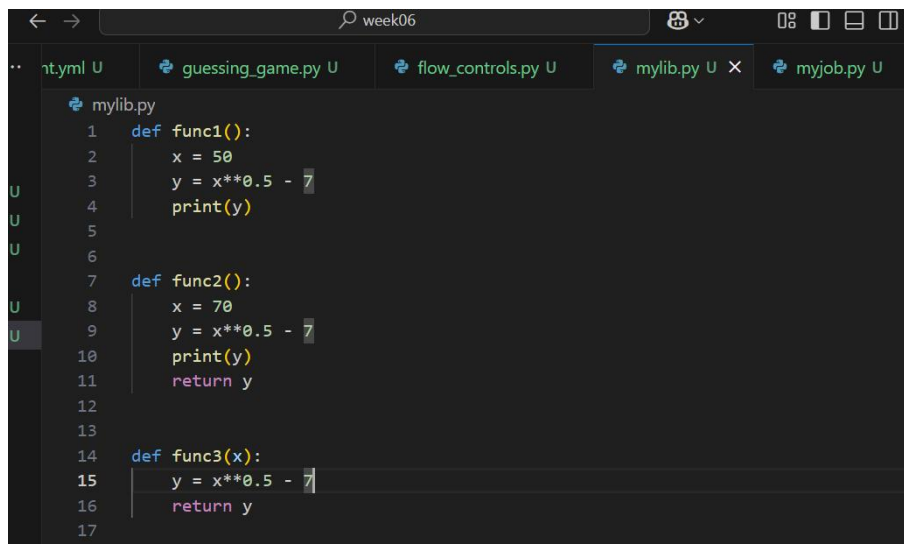
```

FAN0917@LAPTOP-7PDVKG40 MINGW64 ~/repo/week06 (main)
$ 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
(0, 0)
(0, 1)
(1, 0)
(1, 1)
(2, 0)
(2, 1)
0
1
2
3
4
5
4
3
2
1
请输入一个大于 10 的数字: 12
输入有效!
输入 'q' 退出: q
(week06)
FAN0917@LAPTOP-7PDVKG40 MINGW64 ~/repo/week06 (main)
$

```

五、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

```
2
3 y = mylib.func1(0)
4 print(y)
5
6 try:
7     y = mylib.func1(0)
8 except TypeError as e:
9     print(e)
10
11
12 y = mylib.func2()
13 print(y)
14
15 y = mylib.func3(45)
16 print(y)
17
18 y = mylib.func3(x=47)
19 print(y)
20
21
22 try:
23     y = mylib.func3()
24 except TypeError as e:
25     print(e)]
```



The screenshot shows a code editor with a dark theme. At the top, there's a search bar with 'week06' and several window tabs: 'nt.yml U', 'guessing\_game.py U', 'flow\_controls.py U', 'mylib.py U x', and 'myjob.py U'. The 'mylib.py' tab is active, showing the following code:

```
1 def func1():
2     x = 50
3     y = x**0.5 - 7
4     print(y)
5
6
7 def func2():
8     x = 70
9     y = x**0.5 - 7
10    print(y)
11    return y
12
13
14 def func3(x):
15     y = x**0.5 - 7
16     return y
17
```

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

