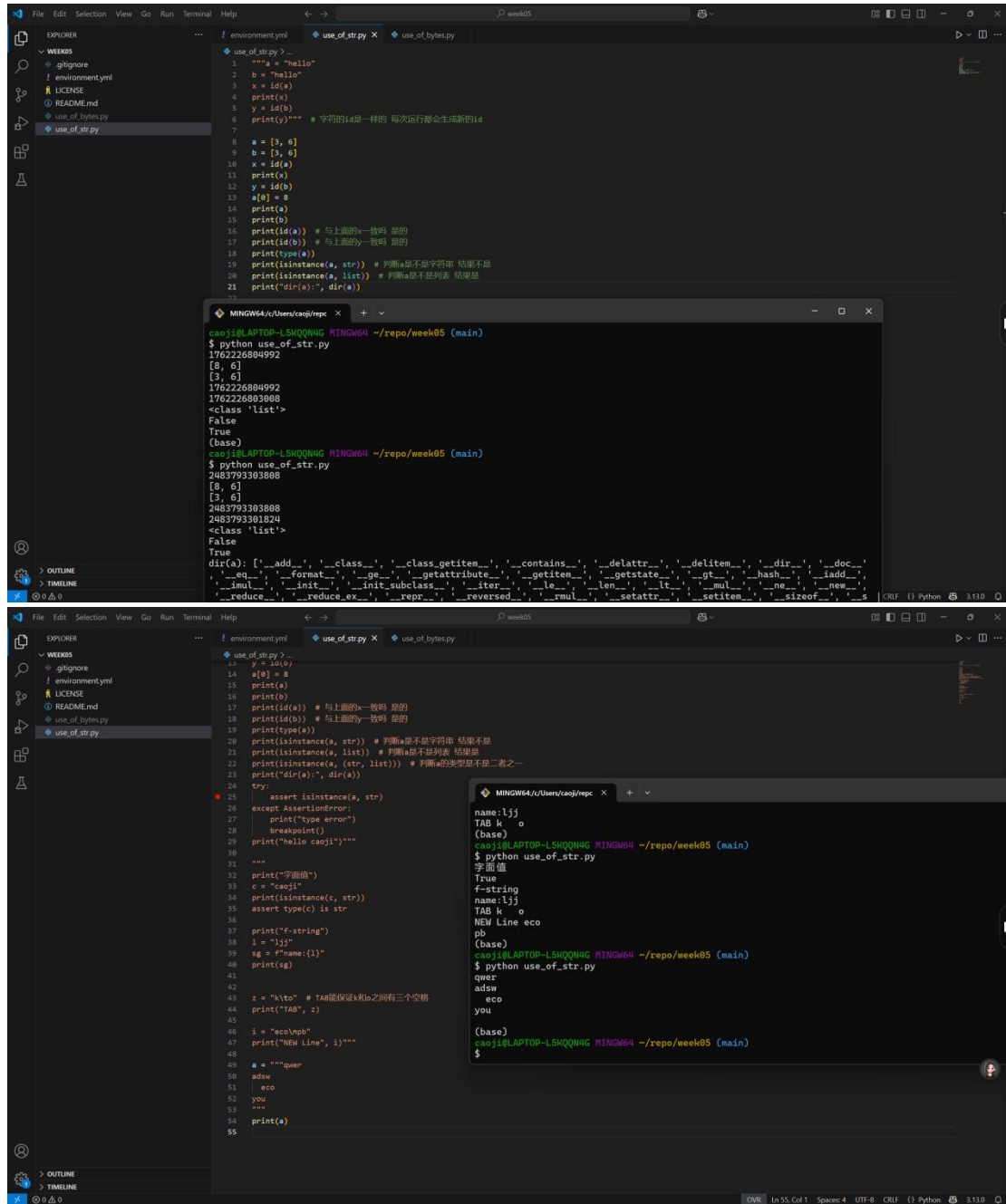


第五周学习笔记



```
environment.yml
use_of_str.py
use_of_bytes.py

WEEDS
  .gitignore
  environment.yml
  LICENSE
  README.md
  use_of_bytes.py
  use_of_str.py

use_of_str.py
1 """a = "hello"
2 b = "hello"
3 x = id(a)
4 print(x)
5 y = id(b)
6 print(y)""" # 字符串的id是一样的 每次运行都会生成新的id
7
8 a = [3, 6]
9 b = [3, 6]
10 x = id(a)
11 print(x)
12 y = id(b)
13 a[0] = 8
14 print(a)
15 print(b)
16 print(id(a)) # 与上面的x一致吗 是的
17 print(id(b)) # 与上面的y一致吗 是的
18 print(type(a))
19 print(isinstance(a, str)) # 判断a是不是字符串 结果不是
20 print(isinstance(a, list)) # 判断a是不是列表 结果是
21 print("dir(a):", dir(a))
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55

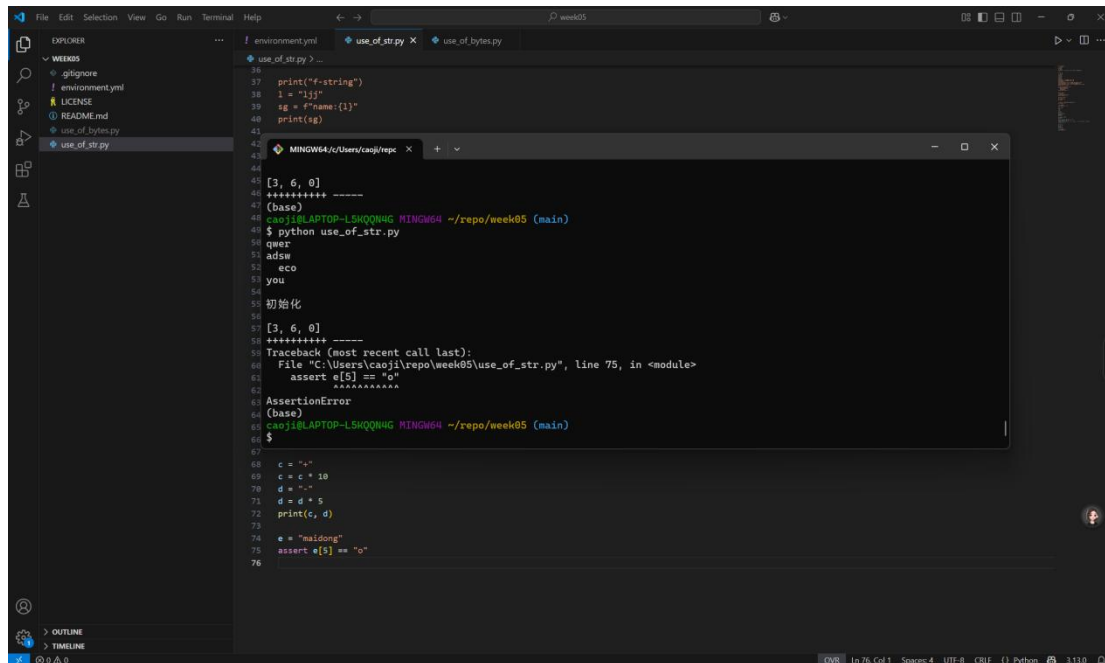
MINGW64~/Users/caoj/repco
caoj@LAPTOP-LSHQQWNG MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
1762226804992
[3, 6]
1762226804992
1762226803088
<class 'list'>
False
True
(base)
caoj@LAPTOP-LSHQQWNG MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
2483793303808
[3, 6]
2483793303808
2483793301024
<class 'list'>
False
True
dir(a): ['__add__', '__class__', '__class_getitem__', '__contains__', '__delattr__', '__delitem__', '__dir__', '__doc__',
__eq__', '__format__', '__ge__', '__getattr__', '__getitem__', '__getstate__', '__gt__', '__hash__', '__iadd__', '__imul__',
__init__', '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mul__', '__ne__', '__new__', '__reduce__',
__reduce_ex__', '__repr__', '__reversed__', '__rmul__', '__setattr__', '__setitem__', '__sizeof__', '__str__', '__subclasshook__']

environment.yml
use_of_str.py
use_of_bytes.py

WEEDS
  .gitignore
  environment.yml
  LICENSE
  README.md
  use_of_bytes.py
  use_of_str.py

use_of_str.py
11 y = id(b)
14 a[0] = 8
15 print(a)
16 print(b)
17 print(id(a)) # 与上面的x一致吗 是的
18 print(id(b)) # 与上面的y一致吗 是的
19 print(type(a))
20 print(isinstance(a, str)) # 判断a是不是字符串 结果不是
21 print(isinstance(a, list)) # 判断a是不是列表 结果是
22 print(isinstance(a, (str, list))) # 判断a的类型是不是二者之一
23 print("dir(a):", dir(a))
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55

MINGW64~/Users/caoj/repco
name:lj
TAB k o
(base)
caoj@LAPTOP-LSHQQWNG MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
字面值
True
f-string
name:lj
TAB k o
NEW Line eco
pb
(base)
caoj@LAPTOP-LSHQQWNG MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
qwer
adsw
eco
you
(base)
caoj@LAPTOP-LSHQQWNG MINGW64 ~/repo/week05 (main)
$
```



```
37 print(f"string")
38 l = "ljj"
39 sg = f"name:{l}"
40 print(sg)
41
42
43
44 [3, 6, 0]
45 ++++++ -----
46 (base)
47 caoji@LAPTOP-LSKQQNHG MINGW64 ~/repo/week05 (main)
48 $ python use_of_str.py
49
50 quer
51 adsw
52 eco
53 you
54
55 初始化
56
57 [3, 6, 0]
58 ++++++ -----
59 Traceback (most recent call last):
60   File "C:\Users\caoji\repo\week05\use_of_str.py", line 75, in <module>
61     assert e[5] == "o"
62     ^^^^^^^^^
63 AssertionError
64 (base)
65 caoji@LAPTOP-LSKQQNHG MINGW64 ~/repo/week05 (main)
66 $
67
68 c = "+"
69 c = c * 10
70 d = "-"
71 d = d * 5
72 print(c, d)
73
74 e = "maidong"
75 assert e[5] == "o"
76
```

与老师演示不同，演示中“Hello”的检索第三个字母对应的应该是第四个，但因为第三个和第四个都是“l”，所以老师的演示并未报错。

但自己操作的时候发现检索并不是检索第五个字母，反而是第六个，并对其进行了验证，发现确实在 Python 中，字符串的索引从 0 开始，所以 e[5] 指的是字符串 "maidong" 中的第 6 个字符，也就是 "n"。

而 assert 语句检查 e[5] 是否等于 "o"，由于 e[5] 实际上是 "n"，条件 e[5] == "o" 为 False，所以会触发 AssertionError 异常。

第二个视频