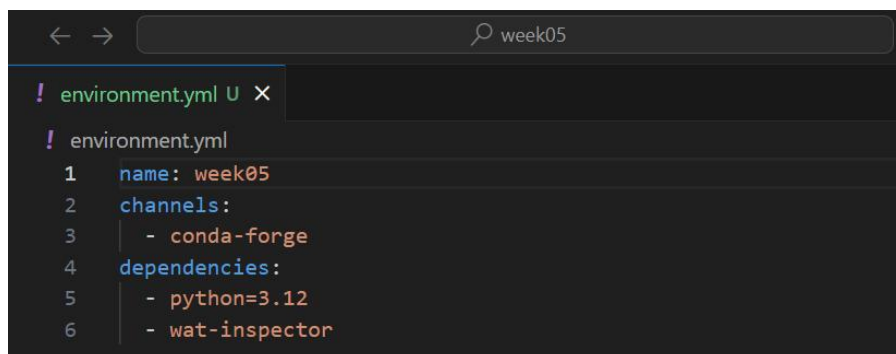


第五周作业笔记

一、Fork 第 05 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 ~/repo
$ git clone git@gitcode.com:jiemoduner/week05.git
Cloning into 'week05'...
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), 8.44 KiB | 2.81 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

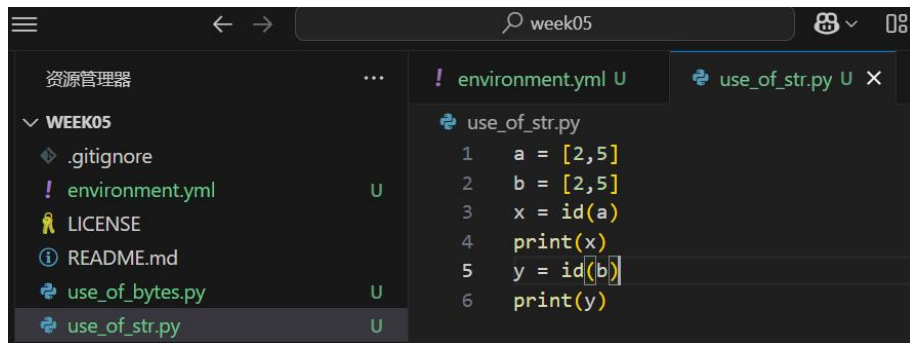


```
<  → week05
! environment.yml U X
! environment.yml
1  name: week05
2  channels:
3    - conda-forge
4  dependencies:
5    - python=3.12
6    - wat-inspector
```

```
(base) FAN0917@LAPTOP-7PDVKG40 MINGW64 ~/repo
$ cd week05
done
#
# To activate this environment, use
#
#     $ conda activate week05
#
# To deactivate an active environment, use
#
#     $ conda deactivate
```

三、逐个 创建 use_of_{name}.py 文件, 尝试一些内置函数

id() -- 返回对象在虚拟内存中的地址 (正整数), 如果 $\text{id}(a) == \text{id}(b)$, 那么 $a \text{ is } b$ (is 是个运算符)



```
(base) FAN0917@LAPTOP-7PDVKG40 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py

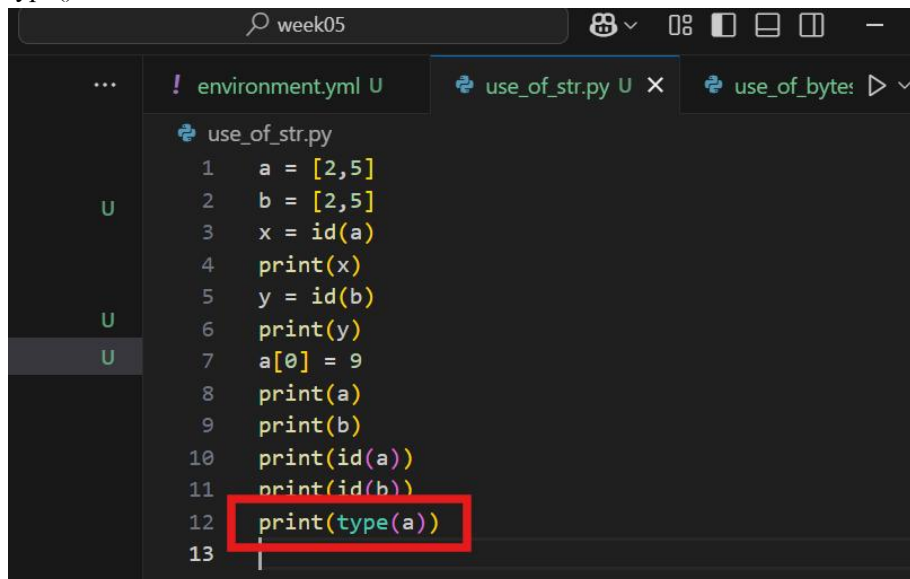
(base) FAN0917@LAPTOP-7PDVKG40 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
hello

(base) FAN0917@LAPTOP-7PDVKG40 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
2205229670896

(base) FAN0917@LAPTOP-7PDVKG40 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
2906382426608
2906382426608

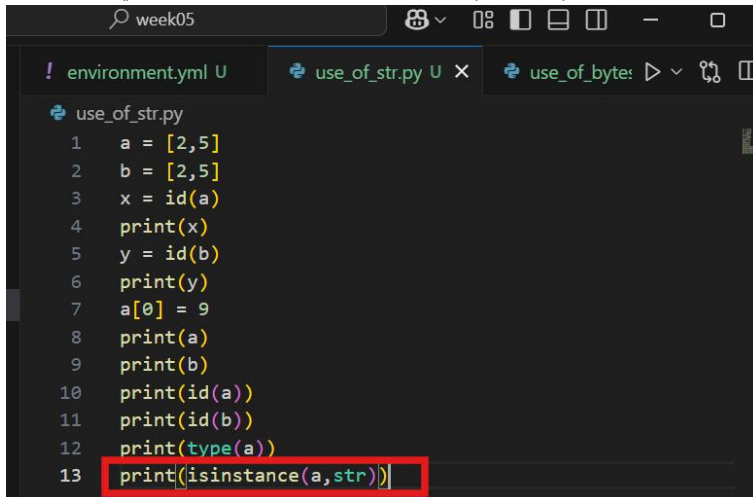
(base) FAN0917@LAPTOP-7PDVKG40 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
2956516399360
2956516397376
```

type() -- 返回对象的类型



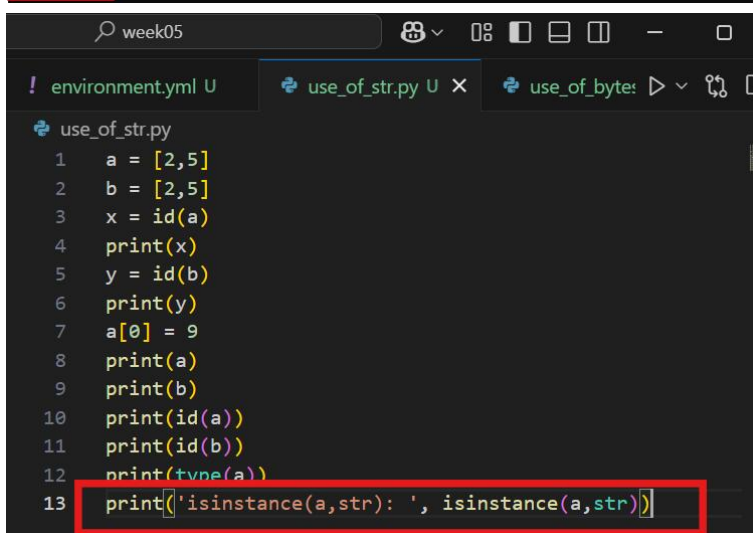
```
(base) FAN0917@LAPTOP-7PDVKG40 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
1439770286336
1439770284352
[9, 5]
[2, 5]
1439770286336
1439770284352
<class 'list'>
```

isinstance() -- 判断对象是否属于某个 (或某些) 类型



```
week05
! environment.yml U
use_of_str.py U x
use_of_byte: ▷ v ? []
use_of_str.py
1 a = [2,5]
2 b = [2,5]
3 x = id(a)
4 print(x)
5 y = id(b)
6 print(y)
7 a[0] = 9
8 print(a)
9 print(b)
10 print(id(a))
11 print(id(b))
12 print(type(a))
13 print(isinstance(a, str))
```

```
(base) FAN0917@LAPTOP-7PDVKG40 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
2565846604032
2565846602048
[9, 5]
[2, 5]
2565846604032
2565846602048
<class 'list'>
False
```

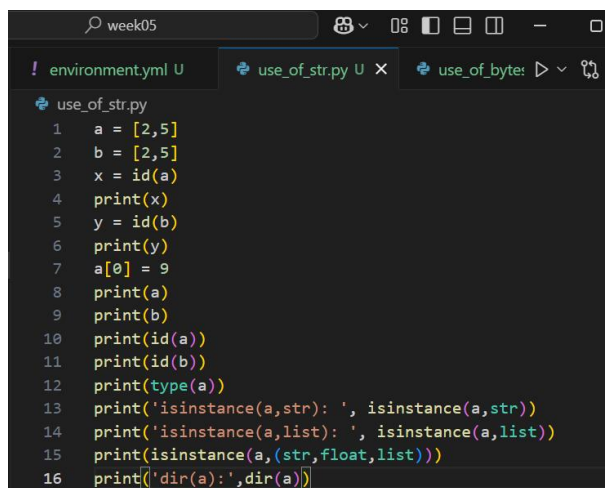


```
week05
! environment.yml U
use_of_str.py U x
use_of_byte: ▷ v ? []
use_of_str.py
1 a = [2,5]
2 b = [2,5]
3 x = id(a)
4 print(x)
5 y = id(b)
6 print(y)
7 a[0] = 9
8 print(a)
9 print(b)
10 print(id(a))
11 print(id(b))
12 print(type(a))
13 print('isinstance(a, str): ', isinstance(a, str))
```

```
(base) FAN0917@LAPTOP-7PDVKG40 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
2627073808640
2627073806656
[9, 5]
[2, 5]
2627073808640
2627073806656
<class 'list'>
instance(a,str): False
```

dir() -- 返回对象所支持的属性 (attributes) 的名称列表

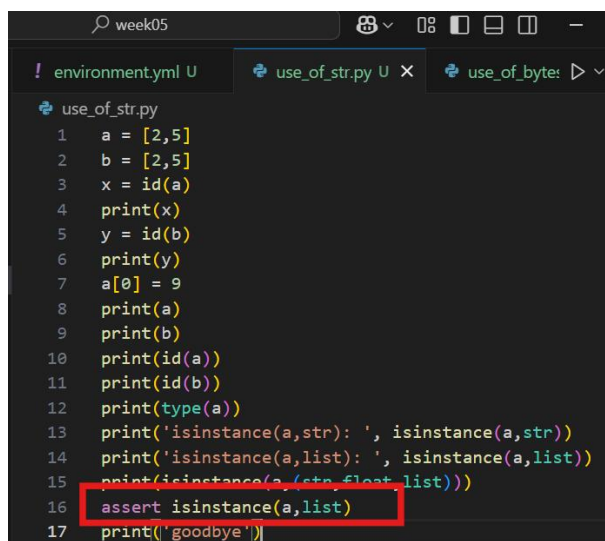
str() -- 返回对象 print 时要显示在终端的字符串



```
week05
! environment.yml U use_of_str.py X use_of_byte: D v
use_of_str.py
1 a = [2,5]
2 b = [2,5]
3 x = id(a)
4 print(x)
5 y = id(b)
6 print(y)
7 a[0] = 9
8 print(a)
9 print(b)
10 print(id(a))
11 print(id(b))
12 print(type(a))
13 print('instance(a,str): ', instance(a,str))
14 print('instance(a,list): ', instance(a,list))
15 print(instance(a,(str,float,list)))
16 print(['dir(a):',dir(a)])
```

```
(base) FAN0917@LAPTOP-7PDVKG40 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
2128776861952
2128776859968
[9, 5]
[2, 5]
2128776861952
2128776859968
<class 'list'>
instance(a,str): False
instance(a,list): True
True
dir(a): ['__add__', '__class__', '__class_getitem__', '__contains__', '__delattr__', '__delitem__', '__dir__', '__doc__',
['__eq__', '__format__', '__ge__', '__getattribute__', '__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__', '__s
tr__', '__subclasshook__', 'append', 'clear', 'copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse',
'sort']
```

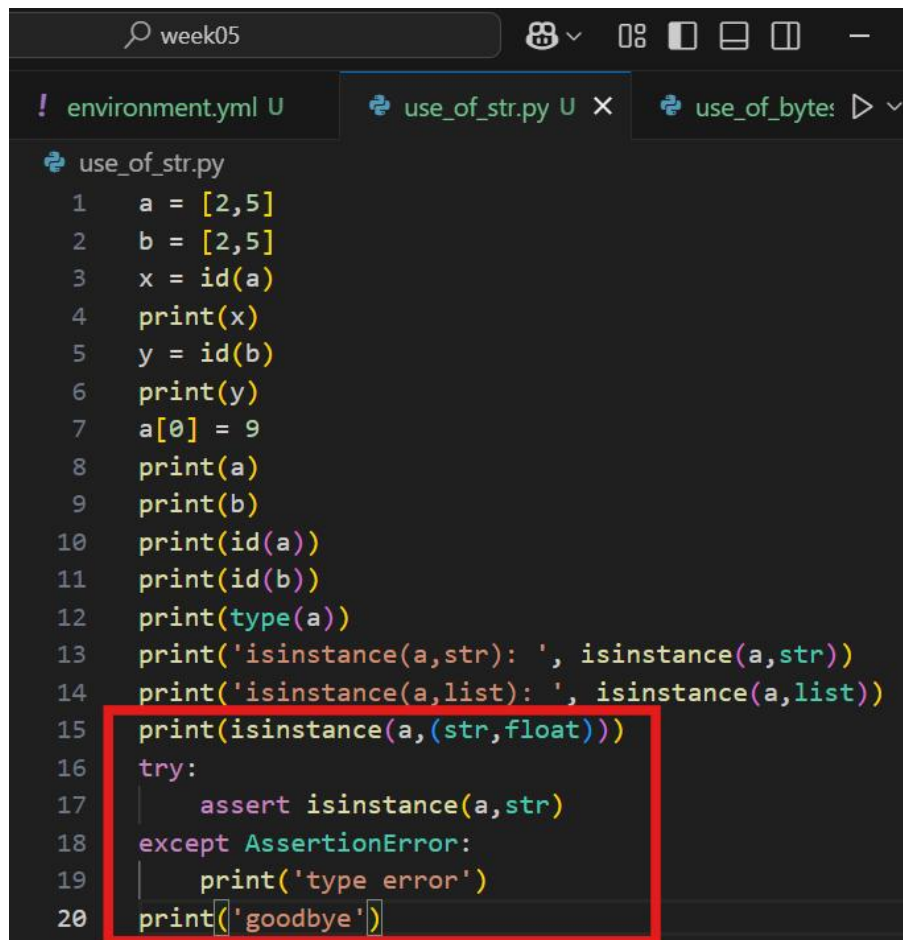
利用 assert 语句查验某个表达式 (expression) 为真, 否则报错 (AssertionError) 退出



```
week05
! environment.yml U use_of_str.py X use_of_byte: D v
use_of_str.py
1 a = [2,5]
2 b = [2,5]
3 x = id(a)
4 print(x)
5 y = id(b)
6 print(y)
7 a[0] = 9
8 print(a)
9 print(b)
10 print(id(a))
11 print(id(b))
12 print(type(a))
13 print('instance(a,str): ', instance(a,str))
14 print('instance(a,list): ', instance(a,list))
15 print(instance(a,(str,float,list)))
16 assert instance(a,list)
17 print(['goodbye'])
```

```
(base) FAN0917@LAPTOP-7PDVKG40 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
1785856006400
1785856004416
[9, 5]
[2, 5]
1785856006400
1785856004416
<class 'list'>
isinstance(a,str): False
isinstance(a,list): True
True
goodbye
```

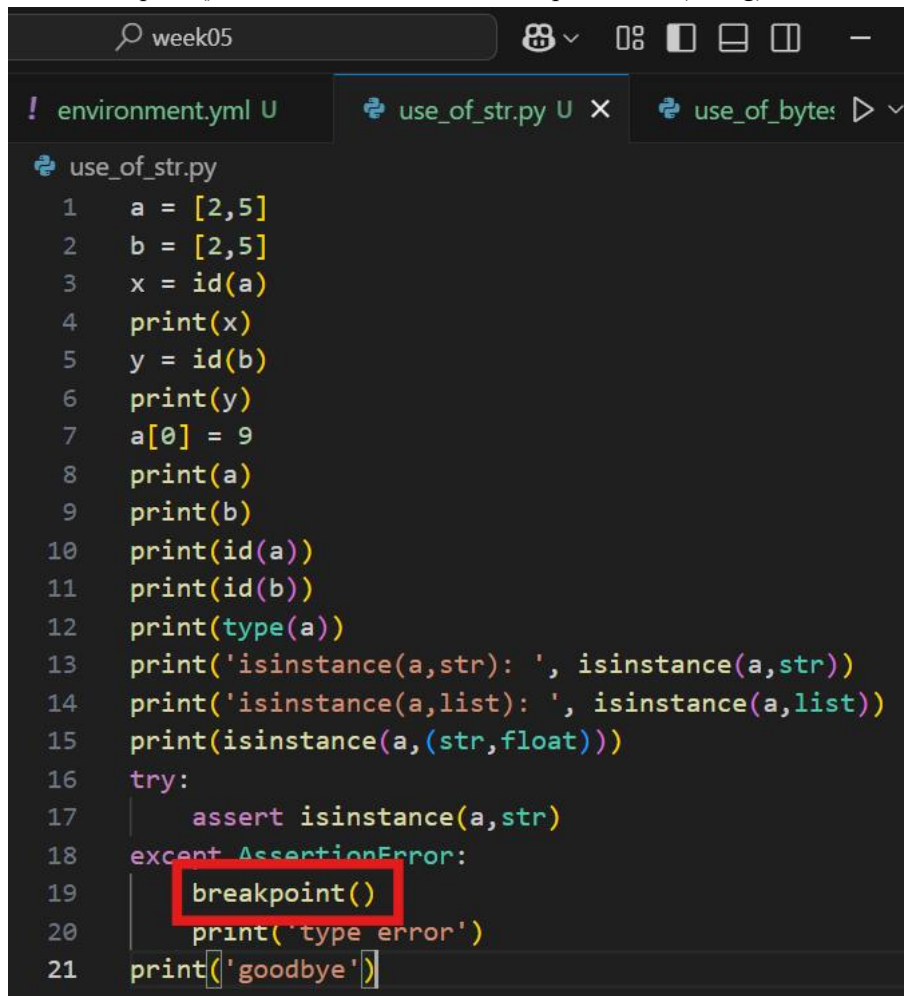
利用 try 语句拦截报错，避免退出，将流程 (flow) 转入 except 语句



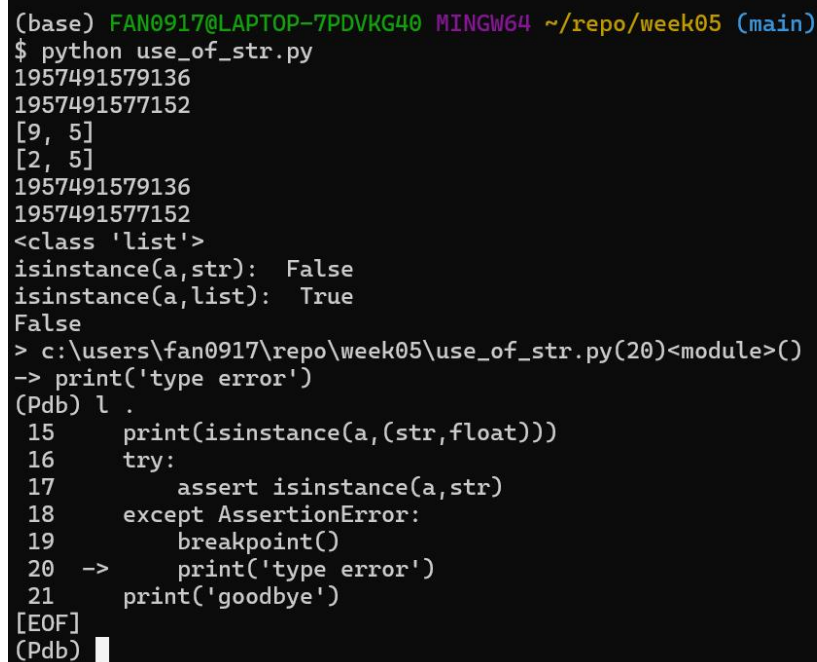
```
! environment.yml U  use_of_str.py U X  use_of_byte: D V
use_of_str.py
1  a = [2,5]
2  b = [2,5]
3  x = id(a)
4  print(x)
5  y = id(b)
6  print(y)
7  a[0] = 9
8  print(a)
9  print(b)
10 print(id(a))
11 print(id(b))
12 print(type(a))
13 print('isinstance(a,str): ', isinstance(a,str))
14 print('isinstance(a,list): ', isinstance(a,list))
15 print(isinstance(a,(str,float)))
16 try:
17     assert isinstance(a,str)
18 except AssertionError:
19     print('type error')
20 print('goodbye')
```

```
(base) FAN0917@LAPTOP-7PDVKG40 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
2906726799616
2906726797632
[9, 5]
[2, 5]
2906726799616
2906726797632
<class 'list'>
isinstance(a,str): False
isinstance(a,list): True
False
type error
goodbye
```


调用 breakpoint() 函数暂停程序运行，进入 pdb 调试 (debug) 模式



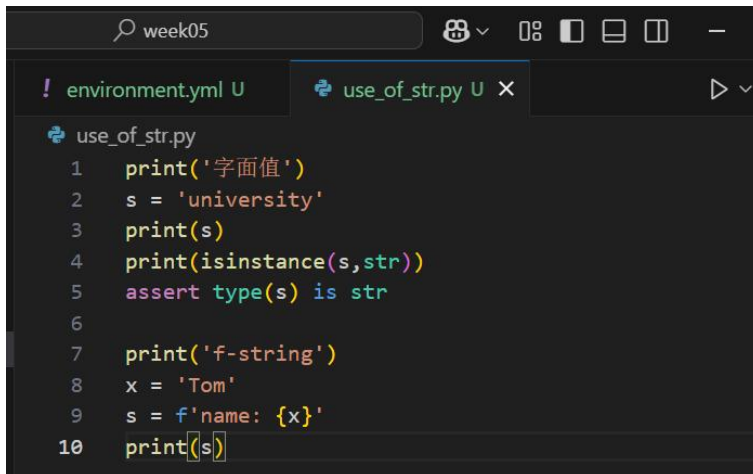
```
! environment.yml U use_of_str.py U X use_of_byte: ▷ v
use_of_str.py
1  a = [2,5]
2  b = [2,5]
3  x = id(a)
4  print(x)
5  y = id(b)
6  print(y)
7  a[0] = 9
8  print(a)
9  print(b)
10 print(id(a))
11 print(id(b))
12 print(type(a))
13 print('isinstance(a,str): ', isinstance(a,str))
14 print('isinstance(a,list): ', isinstance(a,list))
15 print(isinstance(a,(str,float)))
16 try:
17     assert isinstance(a,str)
18 except AssertionError:
19     breakpoint()
20     print('type error')
21 print('goodbye')
```



```
(base) FAN0917@LAPTOP-7PDVKG40 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
1957491579136
1957491577152
[9, 5]
[2, 5]
1957491579136
1957491577152
<class 'list'>
isinstance(a,str): False
isinstance(a,list): True
False
> c:\users\fan0917\repo\week05\use_of_str.py(20)<module>()
-> print('type error')
(Pdb) l .
15     print(isinstance(a,(str,float)))
16     try:
17         assert isinstance(a,str)
18     except AssertionError:
19         breakpoint()
20 ->     print('type error')
21     print('goodbye')
[EOF]
(Pdb)
```

四、通过 表达式 (expression) 得到他们的 实例 (instance)

字面值 (literal) (包括 f-string 语法)

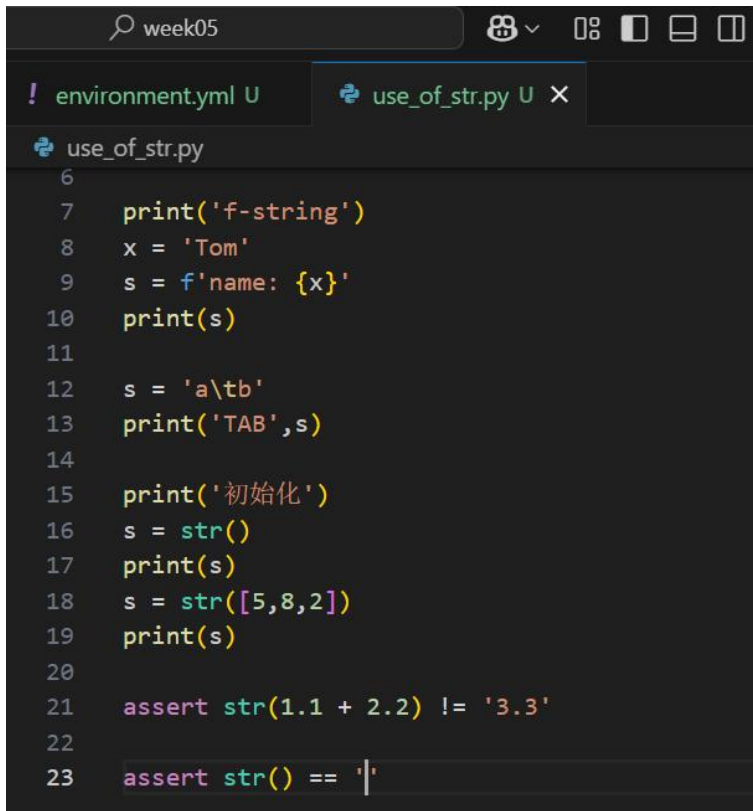


```
! environment.yml U use_of_str.py U X
use_of_str.py
1 print('字面值')
2 s = 'university'
3 print(s)
4 print(isinstance(s, str))
5 assert type(s) is str
6
7 print('f-string')
8 x = 'Tom'
9 s = f'name: {x}'
10 print(s)
```

```
(base) FAN0917@LAPTOP-7PDVKG40 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
字面值
university
True
f-string
name: Tom
```

推导式 (comprehension) (仅限 list、dict、set)

初始化 (init)



```
! environment.yml U use_of_str.py U X
use_of_str.py
6
7 print('f-string')
8 x = 'Tom'
9 s = f'name: {x}'
10 print(s)
11
12 s = 'a\tb'
13 print('TAB', s)
14
15 print('初始化')
16 s = str()
17 print(s)
18 s = str([5, 8, 2])
19 print(s)
20
21 assert str(1.1 + 2.2) != '3.3'
22
23 assert str() == ''
```

```
(base) FAN0917@LAPTOP-7PDVKG40 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
字面值
university
True
f-string
name: Tom
TAB a    b
初始化

[5, 8, 2]
```

运算值 (operator)

```
24
25 s = '='
26 s = s * 20
27 print(s)
```

```
(base) FAN0917@LAPTOP-7PDVKG40 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
字面值
university
True
f-string
name: Tom
TAB a    b
初始化

[5, 8, 2]
=====
```

索引值 (subscription)

```
28
29 s = 'hello'
30 assert s[3] == 'l'
31 assert s[-1] == 'o'
32 assert s[:3] == 'hel'
```

```
(base) FAN0917@LAPTOP-7PDVKG40 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
字面值
university
True
f-string
name: Tom
TAB a    b
初始化

[5, 8, 2]
=====
```


返回值 (return value of function/method call)

```
33
34     s = 'hello'
35     u = s.upper()
36     print(u)
37     print(s)
```

```
(base) FAN0917@LAPTOP-7PDVKG40 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
字面值
university
True
f-string
name: Tom
TAB a    b
初始化
[5, 8, 2]
=====
HELLO
hello
```

五、验证对象的属性

对数学运算符 (+、-、*、/、//、%、@) 有没有支持

```
39     s1 = 'abc'
40     s2 = 'ghi'
41     s = s1 + s2
42     assert s == 'abcghi'
43     print(s2 + s1)
44
45     s = ' *= '
46     s = s * 10
47     print(s)
```

```
(base) FAN0917@LAPTOP-7PDVKG40 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
字面值
university
True
f-string
name: Tom
TAB a    b
初始化
[5, 8, 2]
=====
HELLO
hello
ghiabc
=====
```

如何判断相等 (==)

```
49     s = 'aaa'
50     try:
51         s = s/2
52     except TypeError as e:
53         print(e)
54
55     assert s == 'aaa'
```

```
(base) FAN0917@LAPTOP-7PDVKG40 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
字面值
university
True
f-string
name: Tom
TAB a    b
初始化

[5, 8, 2]
=====
HELLO
hello
ghiabc
=====
unsupported operand type(s) for /: 'str' and 'int'
```

对于比较运算符 (>、<、>=、<=) 有没有支持

什么值被当作 True, 什么值被当作 False (先比较第一个字符, 然后第二次, 顺次类推)

```
57 print('abc' > 'ABC')
58 print('123' > 'abcd')
```

```
(base) FAN0917@LAPTOP-7PDVKG40 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
字面值
university
True
f-string
name: Tom
TAB a    b
初始化

[5, 8, 2]
=====
HELLO
hello
ghiabc
=====
unsupported operand type(s) for /: 'str' and 'int'
True
False
```

是否可迭代 (iterable), 如何做迭代 (for 循环)

```
67 for c in s:
68     print(c)
```

```
(base) FAN0917@LAPTOP-7PDVKG40 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
字面值
university
True
f-string
name: Tom
TAB a    b
初始化

[5, 8, 2]
=====
HELLO
hello
ghiabc
=====
unsupported operand type(s) for /: 'str' and 'int'
True
False
<str_ascii_iterator object at 0x000001CEC131EAA0>
b
o
o
k
```

是否支持返回长度 (len)

```
69
70 print(len(s))
71
```

是否 (如何) 支持索引操作 (subscription) ([] 运算符)

```

72 s = 'book'
73 assert s[1:3] == "oo"
74

```

拥有哪些常用方法 (method) 可供调用 () 运算符)

```

75 s = 'the book of why'
76 print(s.capitalize())
77 print(s)
78

```

```

(base) FAN0917@LAPTOP-7PDVKG40 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
字面值
university
True
f-string
name: Tom
TAB a b
初始化

[5, 8, 2]
=====
HELLO
hello
ghiabc
=====
unsupported operand type(s) for /: 'str' and 'int'
True
False
<str_ascii_iterator object at 0x00000180D8BFECB0>
b
o
o
k
4
The book of why
the book of why

```

```

(base) FAN0917@LAPTOP-7PDVKG40 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
字面值
university
True
f-string
name: Tom
TAB a b
初始化

[5, 8, 2]
=====
HELLO
hello
ghiabc
=====
unsupported operand type(s) for /: 'str' and 'int'
True
False
<str_ascii_iterator object at 0x00000171249AEB00>
b
o
o
k
4

```