

## 第五周学习报告

1、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

```
MINGW64: c:/Users/86153/rep  X + v
drwxr-xr-x 1 86153 197609 0 3月 9 11:57 week01/
drwxr-xr-x 1 86153 197609 0 3月 9 12:00 week01_3492/
drwxr-xr-x 1 86153 197609 0 3月 16 13:28 week02/
drwxr-xr-x 1 86153 197609 0 3月 20 20:57 week03/
drwxr-xr-x 1 86153 197609 0 3月 28 14:06 week04/

(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~/repo
$ pwd
/c:/Users/86153/repo

(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~/repo
$ git clone git@gitcode.com:gossamer/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 | 4.22 MiB/s, done.

(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~/repo
$ ls -l
total 21
drwxr-xr-x 1 86153 197609 0 3月 20 20:34 prj1/
-rw-r--r-- 1 86153 197609 221 3月 15 23:07 tkconch-script.py
drwxr-xr-x 1 86153 197609 0 3月 9 11:57 week01/
drwxr-xr-x 1 86153 197609 0 3月 9 12:00 week01_3492/
drwxr-xr-x 1 86153 197609 0 3月 16 13:28 week02/
drwxr-xr-x 1 86153 197609 0 3月 20 20:57 week03/
drwxr-xr-x 1 86153 197609 0 3月 28 14:06 week04/
drwxr-xr-x 1 86153 197609 0 4月 6 18:10 week05/
```

2、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

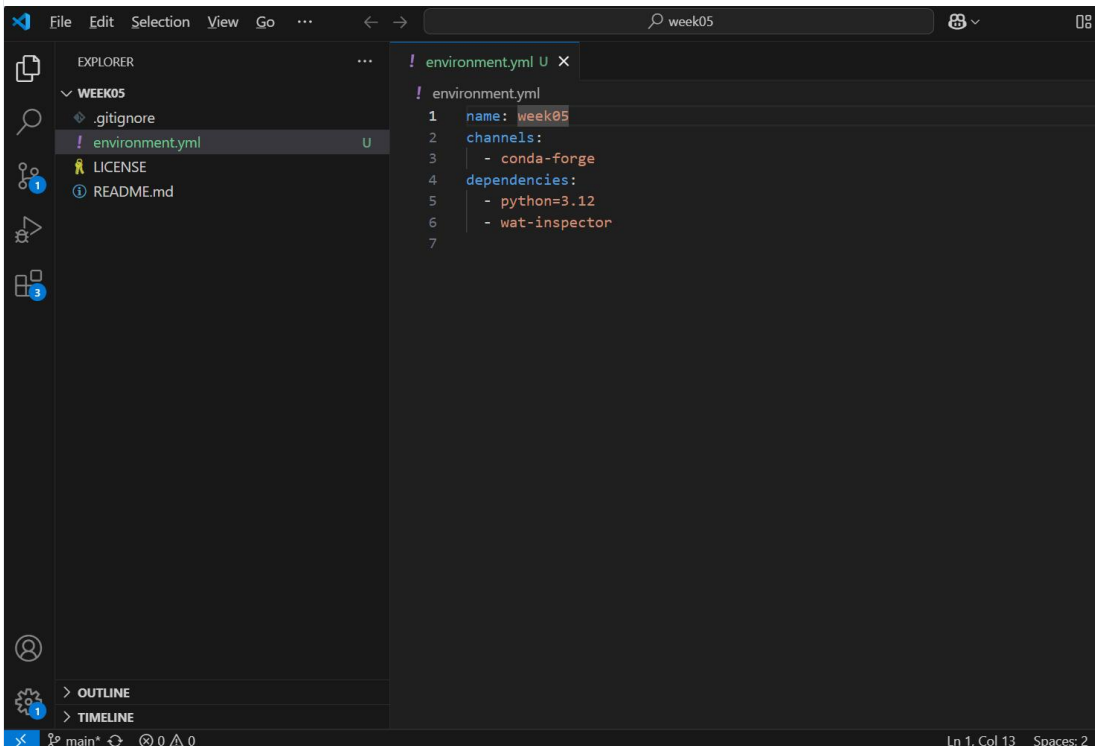
```
MINGW64:/c/Users/86153/rep  X + v
-rw-r--r-- 1 86153 197609 221 3月 15 23:07 tkconch-script.py
drwxr-xr-x 1 86153 197609 0 3月 9 11:57 week01/
drwxr-xr-x 1 86153 197609 0 3月 9 12:00 week01_3492/
drwxr-xr-x 1 86153 197609 0 3月 16 13:28 week02/
drwxr-xr-x 1 86153 197609 0 3月 20 20:57 week03/
drwxr-xr-x 1 86153 197609 0 3月 28 14:06 week04/
drwxr-xr-x 1 86153 197609 0 4月 6 18:10 week05/

(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~/repo
$ cat week04/environment.yml
name: week04
channels:
  - conda-forge
dependencies:
  - python=3.12
  - wat-inspector

(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~/repo
$ cp cat week04/environment.yml week05/
cp: cannot stat 'cat': No such file or directory

(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~/repo
$ cp week04/environment.yml week05/

(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~/repo
$ ls -l week05
total 25
-rw-r--r-- 1 86153 197609 93 4月 6 18:18 environment.yml
-rw-r--r-- 1 86153 197609 18805 4月 6 18:10 LICENSE
-rw-r--r-- 1 86153 197609 2239 4月 6 18:10 README.md
```



```
File Edit Selection View Go ... week05
! environment.yml U X
! environment.yml
1 name: week05
2 channels:
3   - conda-forge
4 dependencies:
5   - python=3.12
6   - wat-inspector
7
```

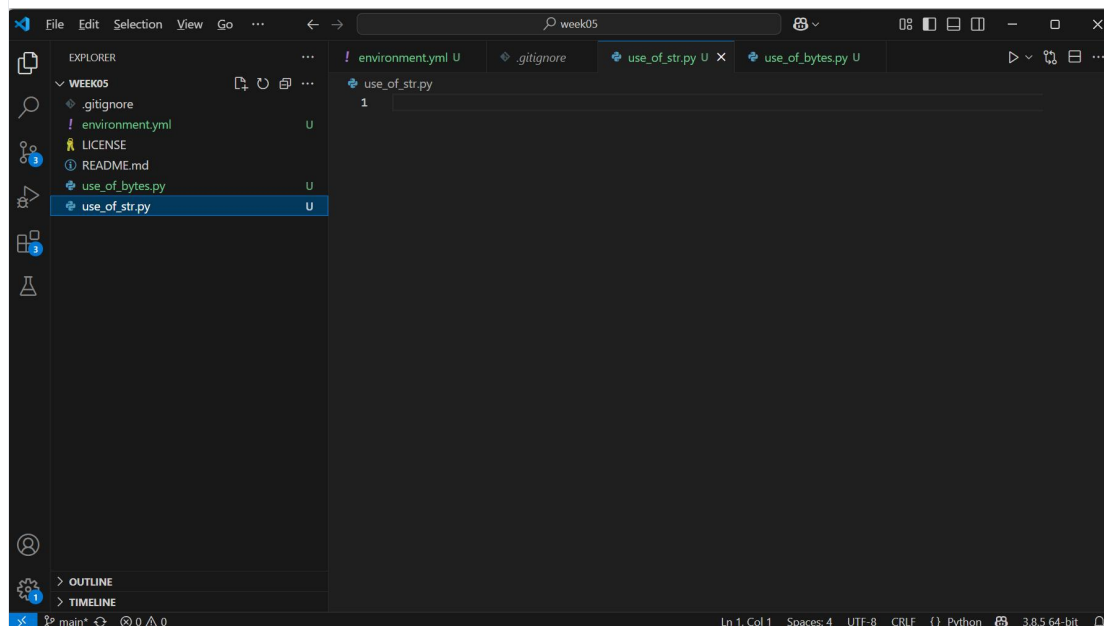
```
MINGW64~/c/Users/86153/rep x + v
$ ls -l week05
total 25
-rw-r--r-- 1 86153 197609 93 4月 6 18:18 environment.yml
-rw-r--r-- 1 86153 197609 18805 4月 6 18:10 LICENSE
-rw-r--r-- 1 86153 197609 2239 4月 6 18:10 README.md

(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~/repo
$ cd week05/

(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week05 (main)
$ ls -l
total 25
-rw-r--r-- 1 86153 197609 93 4月 6 18:20 environment.yml
-rw-r--r-- 1 86153 197609 18805 4月 6 18:10 LICENSE
-rw-r--r-- 1 86153 197609 2239 4月 6 18:10 README.md

(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week05 (main)
$ conda env create
C:\Users\86153\anaconda3\Lib\argparse.py:2006: FutureWarning: 'remote_definition' is deprecated and will be removed in a future version. Use 'conda env create --file=URL' instead.
  action(self, namespace, argument_values, option_string)
Retrieving notices: ...working... done
Channels:
 - conda-forge
 - defaults
 - https://repo.anaconda.com/pkgs/main
 - https://repo.anaconda.com/pkgs/r
 - https://repo.anaconda.com/pkgs/msys2
Platform: win-64
Collecting package metadata (repodata.json): | |
```

3、逐个 创建 `use_of_{name}.py` 文件，其中 `{name}` 替换为上述要求掌握的对象类型，例如 `use_of_str.py`:



```
MINGW64/c/Users/86153/rep x + v
# $ conda activate week05
#
# To deactivate an active environment, use
#
# $ conda deactivate

(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week05 (main)
$ conda env list
# conda environments:
#
base                * C:\Users\86153\anaconda3
myenv               C:\Users\86153\anaconda3\envs\myenv
prj2                C:\Users\86153\anaconda3\envs\prj2
week04              C:\Users\86153\anaconda3\envs\week04
week05              C:\Users\86153\anaconda3\envs\week05

(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week05 (main)
$ conda activate week05
(week05)
86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
(week05)
86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
hello
(week05)
```

4、在全局作用域 (global scope) 内尝试键入 (活学活用) Python 代码, 亲手验证概念 (Proof of Concept, PoC)

```
MINGW64/c/Users/86153/rep x + v

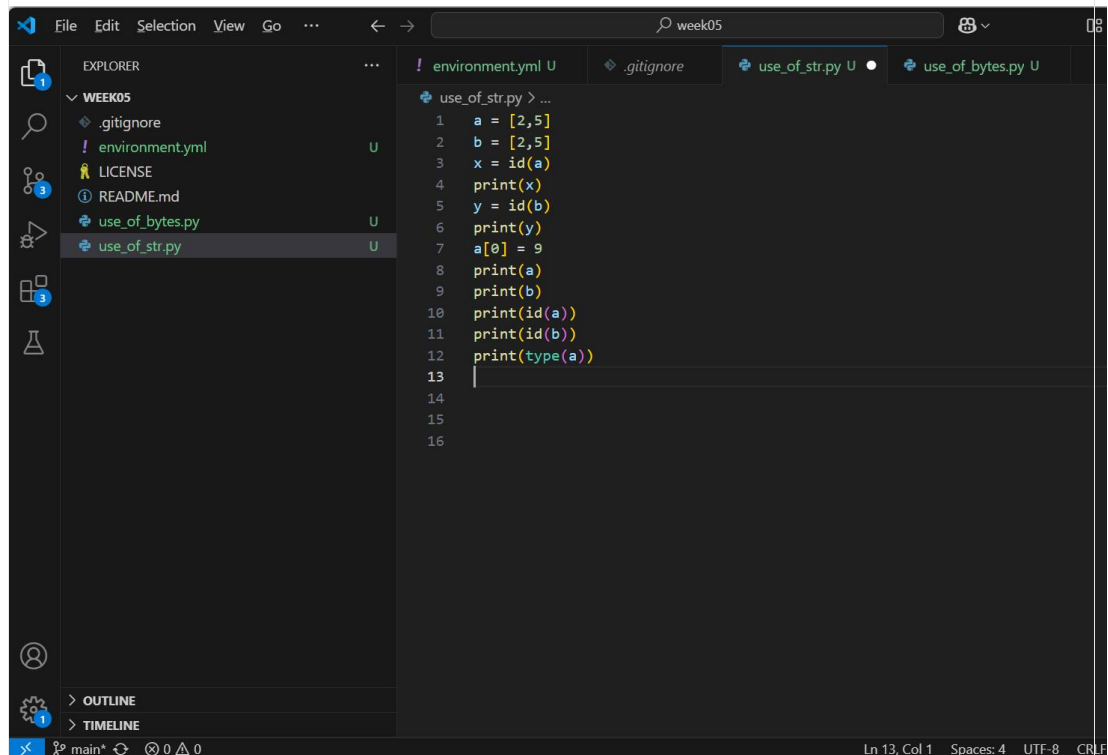
(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week05 (main)
$ conda env list
# conda environments:
#
base                * C:\Users\86153\anaconda3
myenv               C:\Users\86153\anaconda3\envs\myenv
prj2                C:\Users\86153\anaconda3\envs\prj2
week04              C:\Users\86153\anaconda3\envs\week04
week05              C:\Users\86153\anaconda3\envs\week05

(base) 86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week05 (main)
$ conda activate week05
(week05)
86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
(week05)
86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
hello
(week05)
86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
2111554411696
(week05)
86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
3044487355568

main*  Ln 3, Col 9  Spaces: 4  UTF-8  CR
```

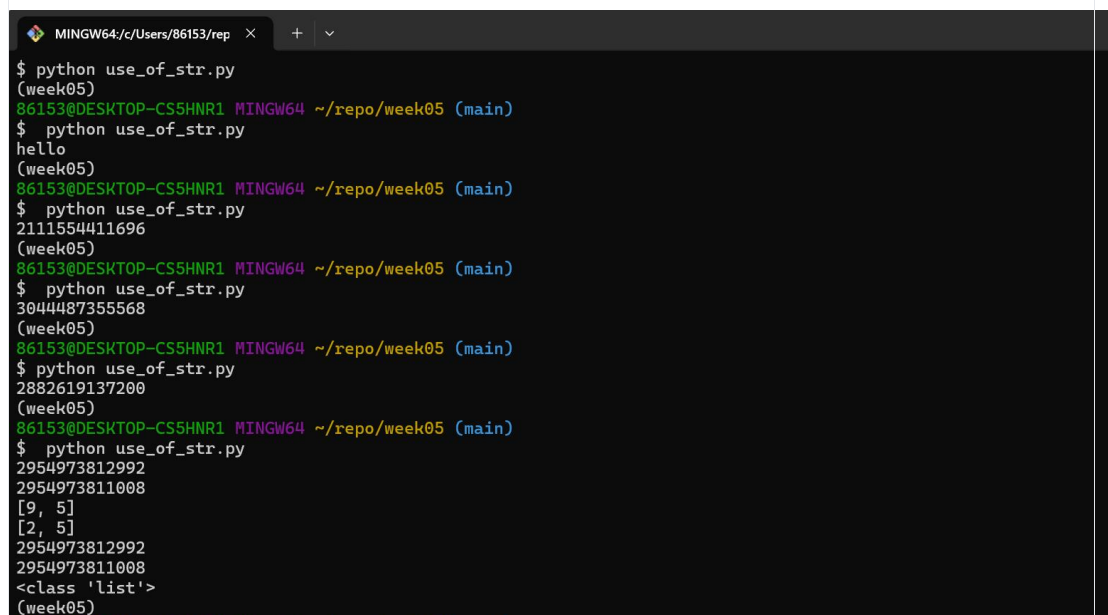
5、对于任何对象, 都可以传给以下内置函数 (built-in function) 用于检视 (inspect):

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



```
File Edit Selection View Go ... week05
EXPLORER
WEEK05
  .gitignore
  ! environment.yml U
  LICENSE
  README.md
  use_of_bytes.py U
  use_of_str.py U
  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
14
15
16
Ln 13, Col 1 Spaces: 4 UTF-8 CRLF
```

`type()` -- 返回对象的类型



```
MINGW64/c/Users/86153/rep x + v
$ python use_of_str.py
86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
hello
86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
2111554411696
86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
3044487355568
86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
2882619137200
86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
2954973812992
2954973811008
[9, 5]
[2, 5]
2954973812992
2954973811008
<class 'list'>
86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week05 (main)
```

`isinstance()` -- 判断对象是否属于某个（或某些）类型

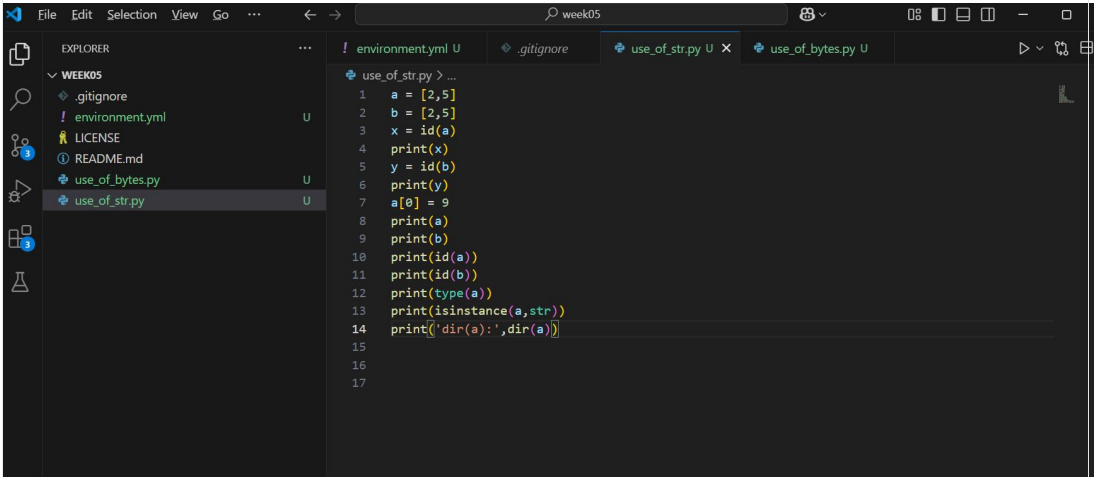
The image shows a terminal window and a VS Code editor. The terminal window displays the execution of a Python script named `use_of_str.py`. The output shows the creation of a list `a` with elements `[9, 5]` and `[2, 5]`, and the printing of its `id` and `type`. The VS Code editor shows the source code of `use_of_str.py`, which defines a list `a` and prints its `id`, `type`, and `isinstance` information.

```
$ python use_of_str.py
3044487355568
(week05)
86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
2882619137200
(week05)
86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
2954973812992
2954973811008
[9, 5]
[2, 5]
2954973812992
2954973811008
<class 'list'>
(week05)
86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
2121109084416
2121109082432
[9, 5]
[2, 5]
2121109084416
2121109082432
<class 'list'>
False
(week05)
```

```
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))
14
15
16
17
```

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

```
MINGW64: c:/Users/86153/rep x + v
86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
2121109084416
2121109082432
[9, 5]
[2, 5]
2121109084416
2121109082432
<class 'list'>
False
(week05)
86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
1328029767936
1328029765952
[9, 5]
[2, 5]
1328029767936
1328029765952
<class 'list'>
False
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']
(week05)
```



`str()` -- 返回对象 `print` 时要显示在终端的字符串

可以调用 `print()` 函数将表达式 (expression) 输出到终端，  
查看结果是否符合预期

可以利用 `assert` 语句查验某个表达式 (expression) 为真，否  
则报错 (`AssertionError`) 退出



```
MINGW64~/Users/86153/rep x + v
, '__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__', '__str__',
['__subclasshook__', 'append', 'clear', 'copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse',
'sort']
'week05')
86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
2296622291200
2296622289216
[9, 5]
[2, 5]
2296622291200
2296622289216
<class 'list'>
False
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__', '__str__',
['__subclasshook__', 'append', 'clear', 'copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse',
'sort']
Traceback (most recent call last):
  File "C:\Users\86153\repo\week05\use_of_str.py", line 15, in <module>
    assert isinstance(a,str)
    ^^^^^^^^^^^^^^^^^^^^^
AssertionError
'week05')
86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week05 (main)
```

```
File Edit Selection View Go ... week05
EXPLORER
WEEK05
  .gitignore
  ! environment.yml U
  LICENSE
  README.md
  use_of_bytes.py U
  use_of_str.py U
  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))
14 print('dir(a):',dir(a))
15 assert isinstance(a,str)
16
17
18
19
Ln 16, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.8.5 64-bit
```

```
MINGW64~/Users/86153/rep x + v
, '__imul__', '__init__', '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mul__', '__ne__', '__new__',
['__reduce__', '__reduce_ex__', '__repr__', '__reversed__', '__rmul__', '__setattr__', '__setitem__', '__sizeof__', '__str__',
['__subclasshook__', 'append', 'clear', 'copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse',
'sort']
Traceback (most recent call last):
  File "C:\Users\86153\repo\week05\use_of_str.py", line 15, in <module>
    assert isinstance(a,str)
    ^^^^^^^^^^^^^^^^^^^^^
AssertionError
'week05')
86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
1977596844288
1977596842304
[9, 5]
[2, 5]
1977596844288
1977596842304
<class 'list'>
False
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__', '__str__',
['__subclasshook__', 'append', 'clear', 'copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse',
'sort']
goodbye
'week05')
```



```
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))
14 print('dir(a):',dir(a))
15 assert isinstance(a,list)
16 print('goodbye')
17
18
19
20
```

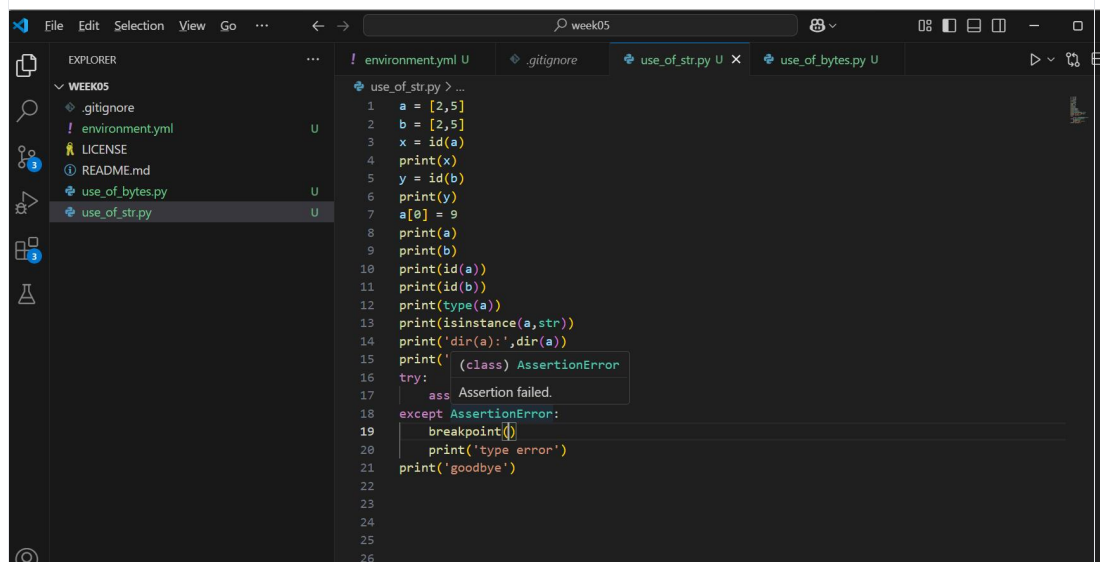
可以利用 `try` 语句拦截报错，避免退出，将流程（flow）转入 `except` 语句

```
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))
14 print('dir(a):',dir(a))
15 print('goodbye')
16 try:
17     assert isinstance(a,str)
18 except AssertionError:
19     print('goodbye')
20
21
22
23
24
```

```
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__', '__str__',
['__subclasshook__', 'append', 'clear', 'copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse', 'sort']
goodbye
(week05)
86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
2990007458048
2990007456064
[9, 5]
[2, 5]
2990007458048
2990007456064
<class 'list'>
False
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__', '__str__',
['__subclasshook__', 'append', 'clear', 'copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse', 'sort']
goodbye
type error
goodbye
(week05)
86153@DESKTOP-CS5HNR1 MINGW64 ~/repo/week05 (main)
```

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

```
MINGW64~/c:/Users/86153/rep x + v
1744180156672
1744180154688
[9, 5]
[2, 5]
1744180156672
1744180154688
<class 'list'>
False
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__', 'append', 'clear', 'copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse', 'sort']
goodbye
> c:\users\86153\repo\week05\use_of_str.py(20)<module>()
-> print('type error')
(Pdb) l
15     print('goodbye')
16     try:
17         assert isinstance(a,str)
18     except AssertionError:
19         breakpoint()
20 ->     print('type error')
21     print('goodbye')
22
23
24
25
```

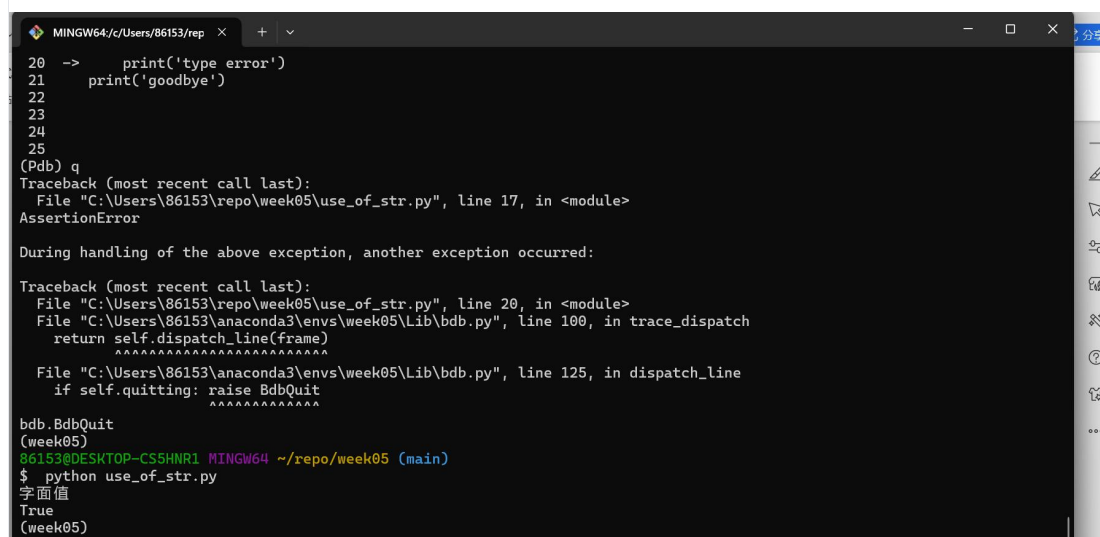
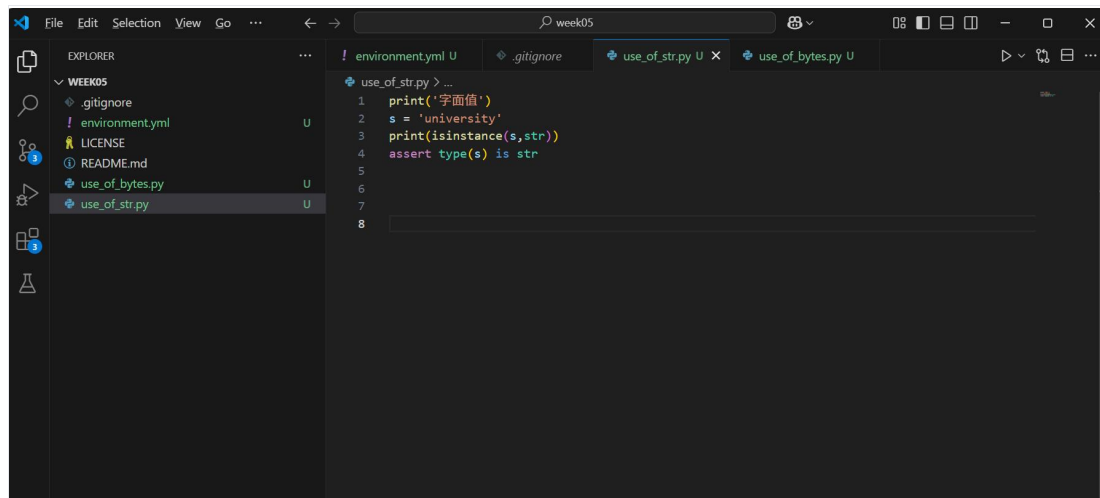


```
File Edit Selection View Go ... week05
EXPLORER
WEEK05
  .gitignore
  environment.yml
  LICENSE
  README.md
  use_of_bytes.py
  use_of_str.py
  use_of_str.py U
  use_of_bytes.py U
  use_of_str.py U

! environment.yml U
.gitignore
use_of_str.py U
use_of_bytes.py U
use_of_str.py U

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))
14 print('dir(a):',dir(a))
15 print(' (class) AssertionError
16 try:
17     ass AssertionError:
18 except AssertionError:
19     breakpoint()
20     print('type error')
21 print('goodbye')
22
23
24
25
26
```

5、对于 每一个 上述要求掌握的对象类型（将来遇到新的对象类型也应该如此），我们首先应该熟悉如何通过 表达式 (expression) 得到他们的 实例 (instance)，一般包括以下途径：字面值 (literal)（包括 f-string 语法）



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

初始化 (init)

运算值 (operator)

索引值 (subscription)

返回值 (return value of function/method call)

7、对于 每一个 上述要求掌握的对象类型（将来遇到新的对象类型也应该如此），我们也要尝试验证其以下几个方面的 属性（attributes）：

对数学运算符（+、-、\*、/、//、%、@）有没有支持

如何判断相等（==）

对于比较运算符（>、<、>=、<=）有没有支持

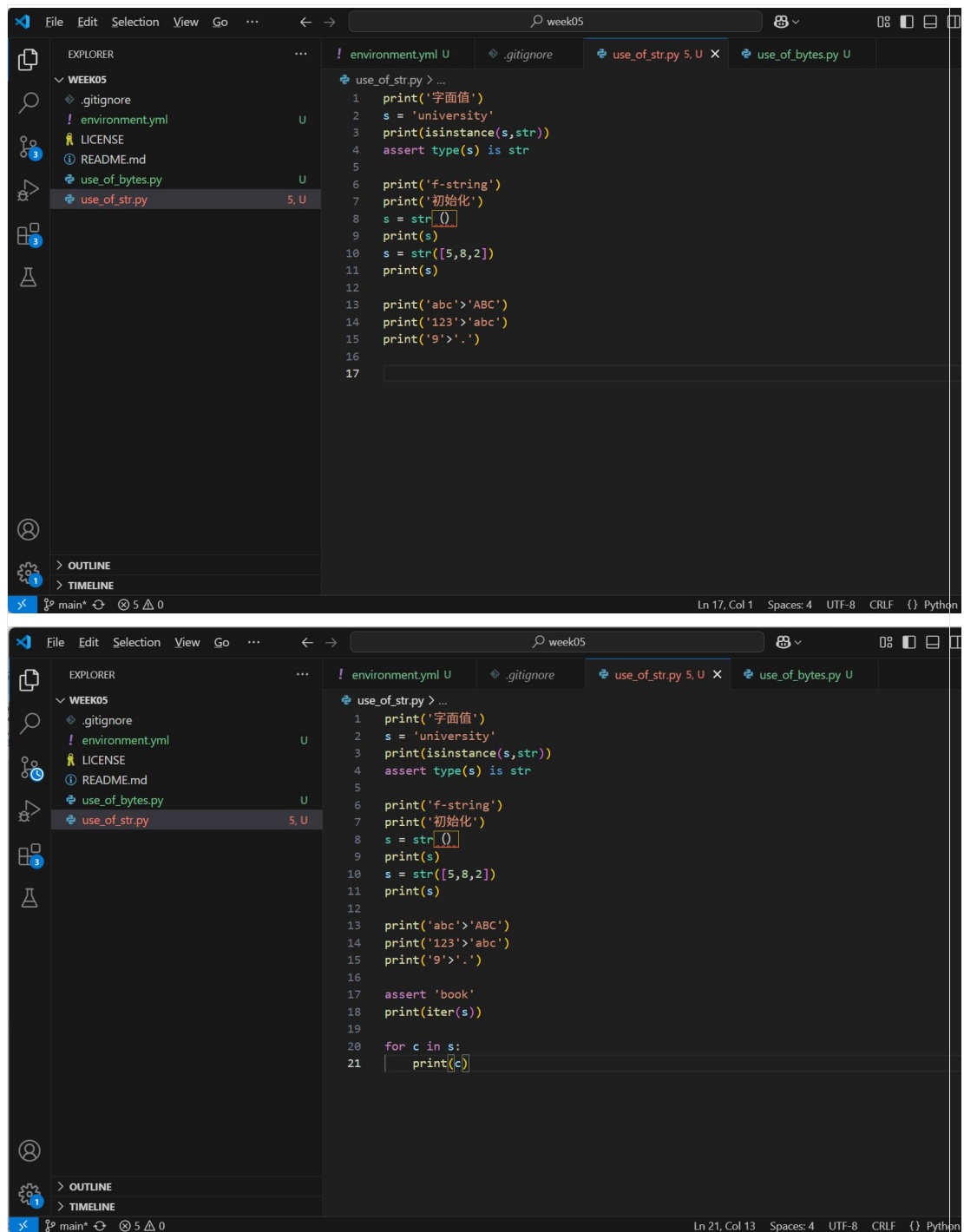
什么值被当作 True，什么值被当作 False

是否可迭代（iterable），如何做迭代（for 循环）

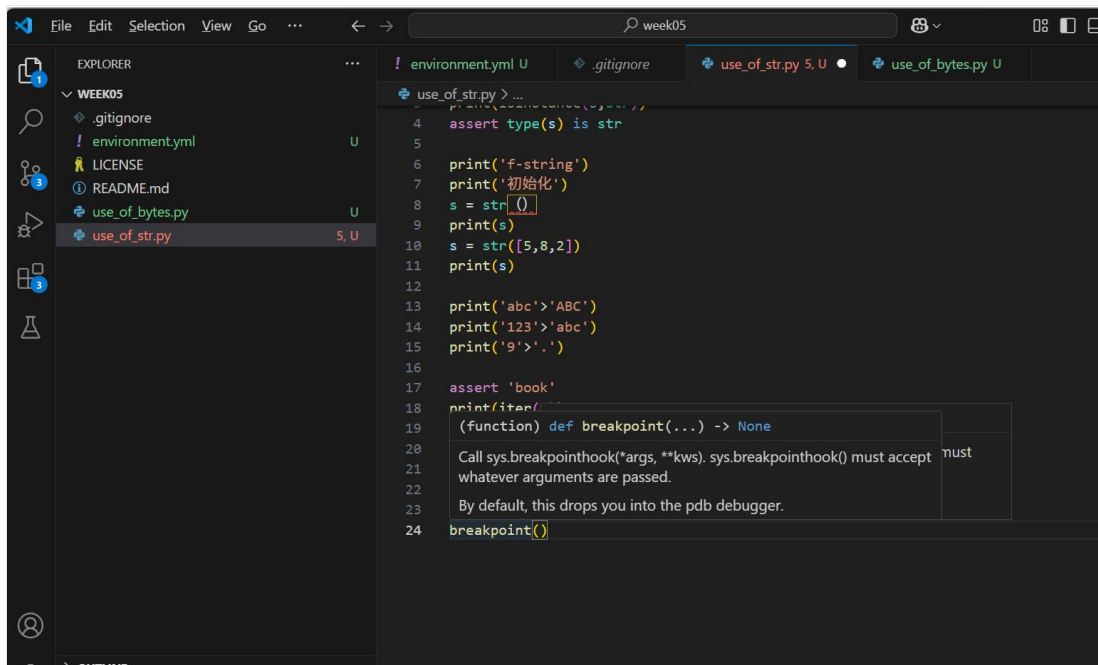
是否支持返回长度（len）

是否（如何）支持索引操作（subscription）（[] 运算符）

拥有哪些常用方法（method）可供调用（() 运算符）

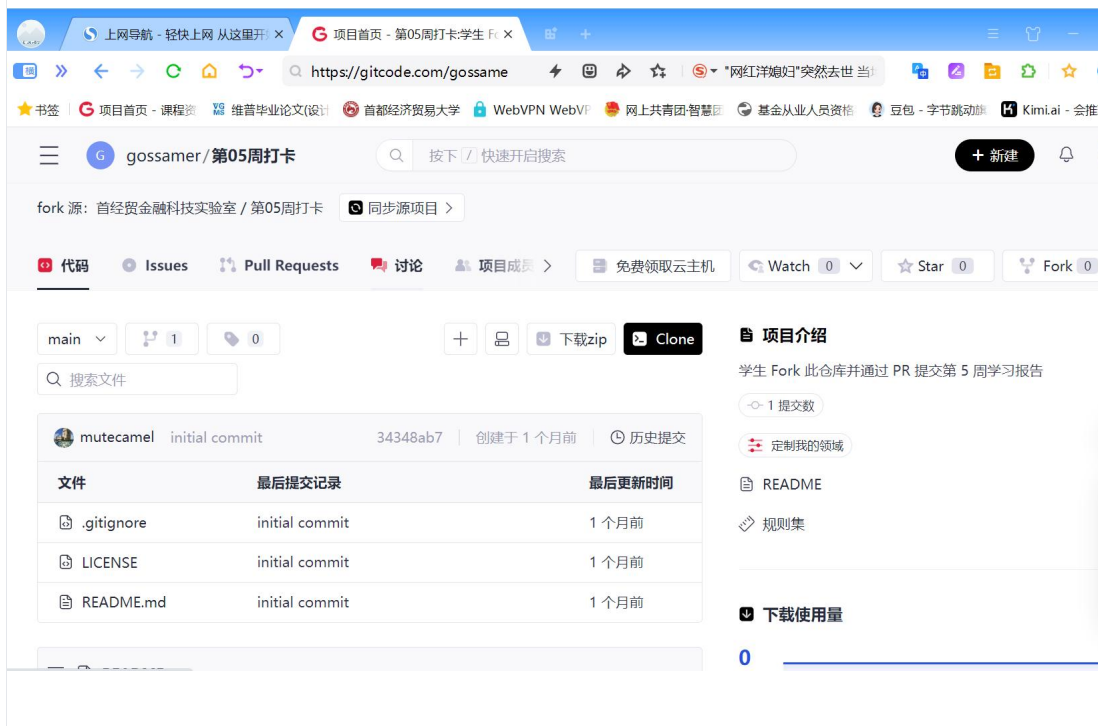


8、建议先在 `pdb` 里试验，然后把确定能够运行的代码写在 `use_of_{name}.py` 文件



```
1 print('f-string')
2 print('初始化')
3 s = str(0)
4 assert type(s) is str
5
6 print('f-string')
7 print('初始化')
8 s = str(0)
9 print(s)
10 s = str([5,8,2])
11 print(s)
12
13 print('abc'>'ABC')
14 print('123'>'abc')
15 print('9'>'.')
16
17 assert 'book'
18 print(iter(s))
19 (function) def breakpoint(...) -> None
20
21 Call sys.breakpointhook(*args, **kws). sys.breakpointhook() must accept
22 whatever arguments are passed.
23 By default, this drops you into the pdb debugger.
24 breakpoint()
```

9、将你学习理解实践这些概念所产生的笔记，以及运行用的 .py 代码，都 add、commit、push 到 GitCode 平台你名下的仓库里，最后提交 PR



gossamer/第05周打卡

fork 源: 曾经金融科技实验室 / 第05周打卡

代码 Issues Pull Requests 讨论 项目成员

main 1 0 + 下载zip Clone

文件	最后提交记录	最后更新时间
.gitignore	initial commit	1 个月前
LICENSE	initial commit	1 个月前
README.md	initial commit	1 个月前

项目介绍

学生 Fork 此仓库并通过 PR 提交第 5 周学习报告

1 提交数

定制我的领域

README

规则集

下载使用量

0