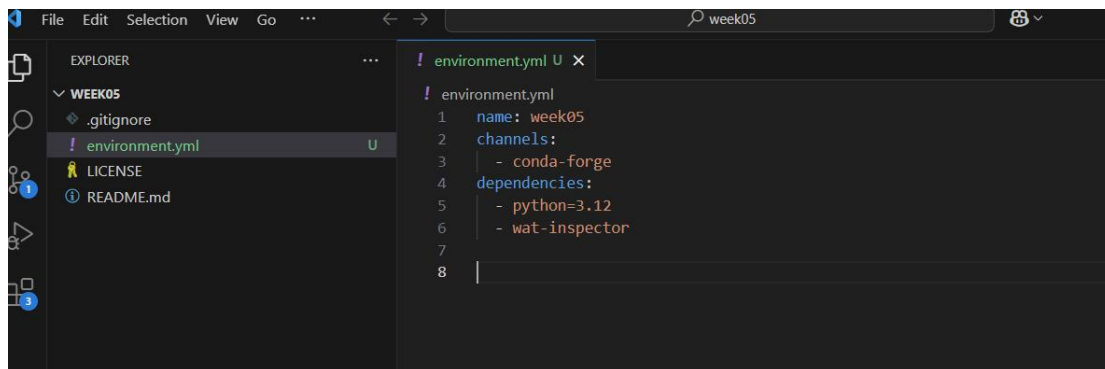


1、用 VS Code 打开项目目录，新建一个 environment.yml 文件，指定安装 Python 3.12，然后运行 conda env create 命令创建 Conda 环境



The screenshot shows the VS Code interface with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'WEEK05' with files like '.gitignore', 'environment.yml', 'LICENSE', and 'README.md'. The code editor shows the content of 'environment.yml'.

```
environment.yml
1 name: week05
2 channels:
3   - conda-forge
4 dependencies:
5   - python=3.12
6   - wat-inspector
7
8
```

```
done
#
# To activate this environment, use
#
#     $ conda activate week05
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(base) ABCD@DESKTOP-SUC708H MINGW64 ~/repo/week05 (main)
$
```

2、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

```
(base) ABCD@DESKTOP-SUC708H MINGW64 ~/repo/week05 (main)
$ conda activate week05
(week05)
ABCD@DESKTOP-SUC708H MINGW64 ~/repo/week05 (main)
$ python
Python 3.12.9 | packaged by conda-forge | (main, Mar 4 2025, 22:37:18) [MSC v.1943
Type "help", "copyright", "credits" or "license" for more information.
>>> quit()
(week05)
ABCD@DESKTOP-SUC708H MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
1770988198848
(week05)
ABCD@DESKTOP-SUC708H MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
2230387628992
(week05)
```

3、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)

```
ABCD@DESKTOP-SUC708H MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
1662201584576
1662201584576
(week05)
ABCD@DESKTOP-SUC708H MINGW64 ~/repo/week05 (main)
$
```

`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)

```
! environment.yml U  use_of_str.py U x  use_of_bytes.py U
use_of_str.py > ...
1  a = "hello"
2  b = "hello"
3  x = id(a)
4  print(x)
5  y = id(b)
6  print(y)
7  |
```

```
ABCD@DESKTOP-SUC708H MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
1662201584576
1662201584576
(week05)
ABCD@DESKTOP-SUC708H MINGW64 ~/repo/week05 (main)
$
```

一样的

```
! environment.yml U  use_of_str.py U x  use_of_bytes.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  |
```

```
ABCD@DESKTOP-SUC708H MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
1903792888064
1903792886080
(week05)
```

不一样的两个对象

```

ABCD@DESKTOP-SUC708H MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
1700371241216
1700371239232
[9, 5]
[2, 5]
(week05)
ABCD@DESKTOP-SUC708H MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
2211764181248
2211764179264
[9, 5]
[2, 5]
2211764181248
2211764179264
(week05)
ABCD@DESKTOP-SUC708H MINGW64 ~/repo/week05 (main)
$

```

type() -- 返回对象的类型输出类型

```

$ python use_of_str.py
3042080659712
3042080657728
[9, 5]
[2, 5]
3042080659712
3042080657728
<class 'list'>
(week05)

```

isinstance() -- 判断对象是否属于某个 (或某些) 类型

```

print("isinstance(a, str):", isinstance(a, str))

```

```

$ python use_of_str.py
3081229244672
3081229242688
[9, 5]
[2, 5]
3081229244672
3081229242688
<class 'list'>
isinstance(a, str): False
(week05)

```

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

```

isinstance(a, str): 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']
(week05)

```

```

<class 'list'>
isinstance(a, str): False
isinstance(a, list): True
dir(a): ['__add__', '__class__', '__class_getitem__', '__contains__', '__eq__', '__format__', '__ge__', '__getattr__', '__getattribute__', '__getitem__', '__iadd__', '__imul__', '__init__', '__init_subclass__', '__iter__', '__len__', '__lt__', '__mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__reversed__', '__setattr__', '__setitem__', '__sizeof__', '__str__', '__subclasshook__', 'append', 'clear', 'copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse', 'sort']
(week05)
ABCD@DESKTOP-SUC708H MINGW64 ~/repo/week05 (main)

```

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)

```

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

```

```

(week05)
ABCD@DESKTOP-SUC708H MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
字面值
True
(week05)

```

```

7 print("f-string")
8 x = "Tom"
9 s = f"name:{x}"
10 print(s)
11 |

```

```

$ python use_of_str.py
字面值
university
True
f-string
name:Tom
(week05)

```

\t 中间空 3 格

```
11
12 s = "a\tb"
13 print("TAB", s)
14
```

```
ABCD@DESKTOP-SUC708H MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
字面值
university
True
f-string
name:Tom
TAB a    b
(week05)
```

\n 换行

```
14
15 s = "aaaaa\nbbbbbb"
16 print("New Line", s)
17
```

```
ABCD@DESKTOP-SUC708H MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
字面值
university
True
f-string
name:Tom
TAB a    b
(week05)
```

```
AssertionError
Uncaught exception. Entering post mortem debugging
Running 'cont' or 'step' will restart the program
> c:\users\abcd\repo\week05\use_of_str.py(32)<module>()
-> assert str([1.1 + 2.2]) == "3.3"
(Pdb) p srt(1.1+2.2)
*** NameError: name 'srt' is not defined
(Pdb) p str(1.1+2.2)
'3.3000000000000003'
(Pdb)
```

```
s = "hello"
assert s[3] == "l"
assert s[-1] == "o"
assert s[:3] == "hel"
```

```
(week05)
ABCD@DESKTOP-SUC708H MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
字面值
university
True
f-string
name:Tom
TAB a    b
New Line aaaaa
bbbbbb
xyz
abc
    eee
aaa

初始化

[5, 8, 2]
=====
string index out of range
HELLO
hello
(week05)
ABCD@DESKTOP-SUC708H MINGW64 ~/repo/week05 (main)
```