

第五周打卡

将第四周文件夹中的 environment.yml 文件复制到 week05 文件夹

```
wat-inspector(base)
zhangyunfei@LAPTOP-NU6BCCKG MINGW64 ~/repo
$ cp week04/environment.yml week05/
(base)
zhangyunfei@LAPTOP-NU6BCCKG MINGW64 ~/repo
$ ls -l week05
total 25
-rw-r--r-- 1 zhangyunfei 197121 91 4月 11 09:45 environment.yml
-rw-r--r-- 1 zhangyunfei 197121 18805 4月 7 17:26 LICENSE
-rw-r--r-- 1 zhangyunfei 197121 2239 4月 7 17:26 README.md
(base)
```

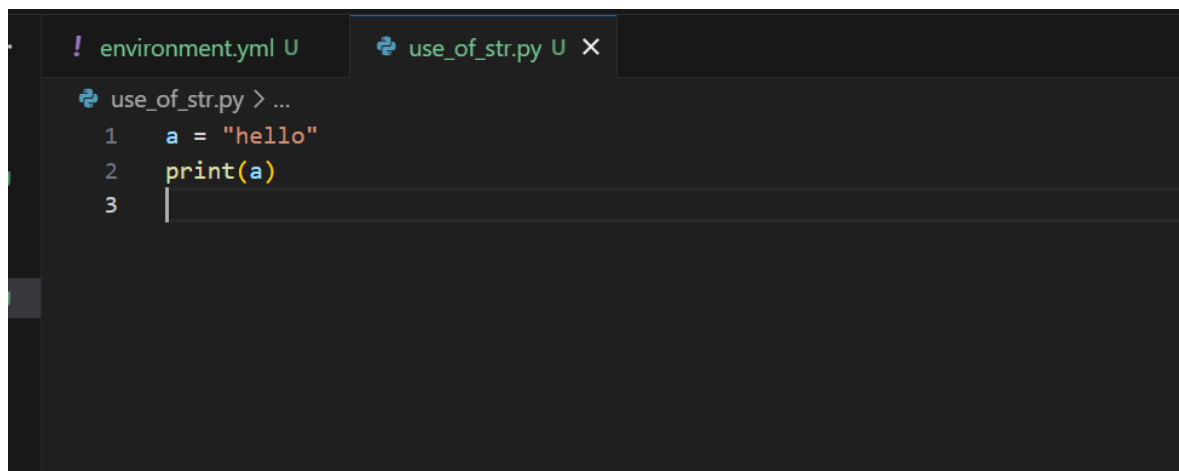
修改环境文件

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

Conda env create 创建 conda 环境

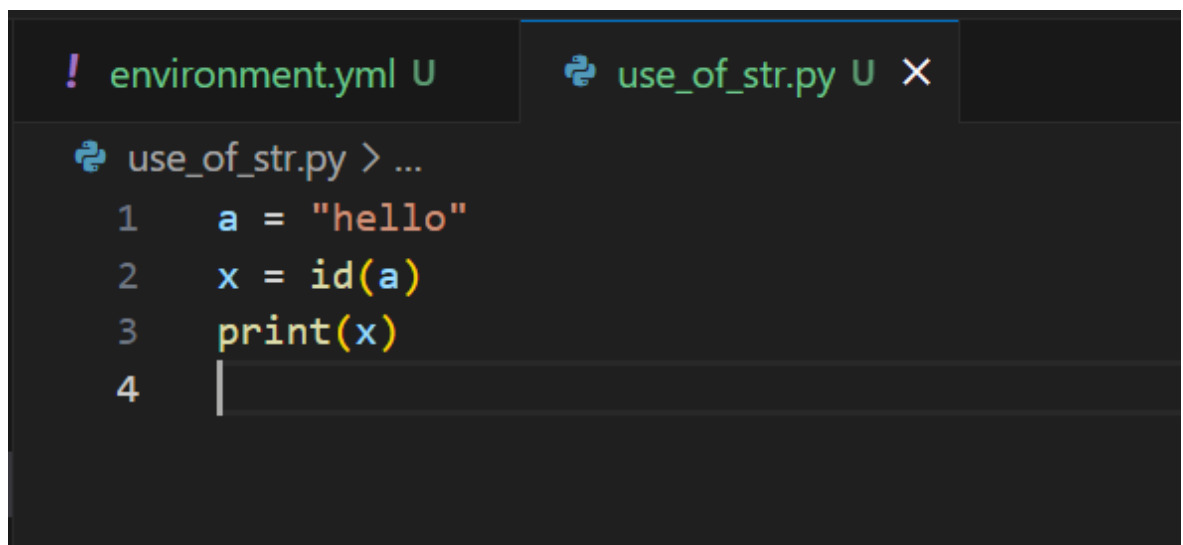
运行 python 文件

```
zhangyunfei@LAPTOP-NU6BCCKG MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
hello
(week05)
zhangyunfei@LAPTOP-NU6BCCKG MINGW64 ~/repo/week05 (main)
$ |
```



```
! environment.yml U use_of_str.py U X
use_of_str.py > ...
1 a = "hello"
2 print(a)
3 |
```

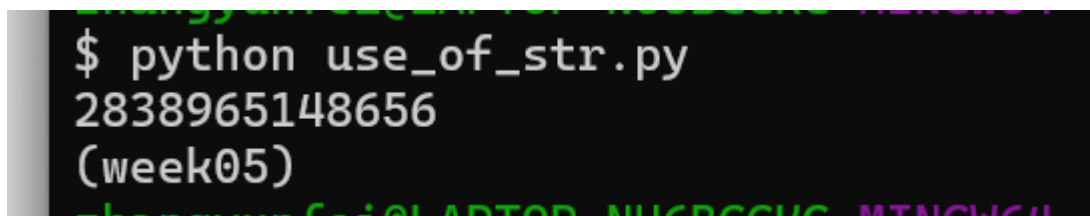
输出 a 的 id



```
! environment.yml U use_of_str.py U X
use_of_str.py > ...
1 a = "hello"
2 x = id(a)
3 print(x)
4 |
```

Id 返回对象在虚拟内存中的地址

Type 返回对象的类型



```
$ python use_of_str.py
2838965148656
(week05)
```

```
environment.yml  use_of_str.py  x
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
```

```
zhangyunfei@LAPTOP-N06BCKRG MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
2729164544256
2729164542272
[9, 5]
[2, 5]
2729164544256
2729164542272
<class 'list'>
(week05)
zhangyunfei@LAPTOP-N06BCKRG MINGW64 ~/repo/week05 (main)
```

isinstance()- 判断对象是否属于某个 (或某些) 类型

```
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("dir(a):", dir(a))
15
```

dir()-- 返回对象所支持的属性 (attributes) 的名称列表

```
zhangyunfei@LAPTOP-NU6BCKG MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
1622105069824
1622105067840
[9, 5]
[2, 5]
1622105069824
1622105067840
<class 'list'>
isinstance(a,str): False
dir(a): ['__add__', '__class__', '__class_getitem__', '__contains__', '__delattr__',
['__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__getstate__',
['__imul__', '__init__', '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__',
['__reduce__', '__reduce_ex__', '__repr__', '__reversed__', '__rmul__', '__setattr__',
['__sizeof__', '__subclasshook__', 'append', 'clear', 'copy', 'count', 'extend', 'index', 'ins
'sort']
(week05)
```

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

```
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("dir(a):", dir(a))
15 assert isinstance(a, str)
16
```

```
['__sizeof__', '__subclasshook__', 'append', 'clear', 'copy', 'count', 'extend', 'index', 'ins
'sort']
Traceback (most recent call last):
  File "C:\Users\zhangyunfei\repo\week05\use_of_str.py", line 15, in <module>
    assert isinstance(a, str)
    ^^^^^^^^^^^^^^^^^^^^^^^
AssertionError
(week05)
```

try 语句拦截报错，避免退出，将流程转入 except 语句

```
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("dir(a):", dir(a))
15 try:
16     assert isinstance(a, str)
17 except AssertionError:
18     print("type error")
19 print("hello")
20
```

breakpoint()函数暂停程序运行，进入 pdb 调试模式

```
zhangyuntai@LAPTOP-N06BCKRG MINGW64 ~/Repo/week05 (main)
$ python use_of_str.py
2208613800192
2208613798208
[9, 5]
[2, 5]
2208613800192
2208613798208
<class 'list'>
isinstance(a,str): False
dir(a): ['__add__', '__class__', '__class_getitem__', '__contains__',
 '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__',
 '__imul__', '__init__', '__init_subclass__', '__iter__', '__le__',
 '__reduce__', '__reduce_ex__', '__repr__', '__reversed__', '__rmul__',
 '__str__', '__subclasshook__', 'append', 'clear', 'copy', 'count', 'exte
'sort']
type error
hello
(week05)
```

\n 换行显示

```
use_of_str_1.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)
11
12 s = "a\tb"
13 print("TAB", s)
14
15 s = "aaa\nbbb"
16 print("New Line", s)
17
```

```
$ python use_of_str_1.py
字面值
university
True
f-string
name: Tom
TAB a    b
New Line aaa
bbb
(week05)
```

初始化获得实例

```
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)
11
12 s = "a\tb"
13 print("TAB", s)
14
15 s = "aaa\nbbb"
16 print("New Line", s)
17
18
19 print("初始化")
20 s = str()
21 print(s)
22 s = str([5, 8, 2])
23 print(s)
24
25 assert str(1 + 2) == "3"
26 assert str(1.1 + 2.2) == "3.3"
27
```

```
$ python use_of_str_1.py
字面值
university
True
f-string
name: Tom
TAB a   b
New Line aaa
bbb
初始化

[5, 8, 2]
Traceback (most recent call last):
  File "C:\Users\zhangyunfei\repo\week05\use_of_str_1.py", line 26, in <module>
    assert str(1.1 + 2.2) == "3.3"
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
AssertionError
```

运算值、索引值、返回值

```

print("运算值")
s = "="
s = s * 20
print(s)

print("索引值")
s = "hello"
print(s[3])
assert s[3] == "l"
assert s[:3] == "hel"

print("返回值")
s = "hello"
s = s.upper()
print(s)

t = "name:{},age:{"
print(t)
t1 = t.format("Jack", 21)
print(t1)

```

字符串运算符 +、*支持

```

$ python use_of_str_1.py
字面值
university
True
f-string
name: Tom
TAB a    b
New Line aaa
bbb
初始化
[5, 8, 2]
运算值
=====
索引值
l
返回值
HELLO
name: {}, age: {}
name: Jack, age: 21
(week05)

```

字符串比较大小


```
use_of_str_2.py
1  print("abc" > "ABC")
2  print("123" > "abcd")
3
```

```
zhangyunfei@LAPTOP-NU6BCCKG MINGW
$ python use_of_str_2.py
True
False
(week05)
```

Assert 空字符串报错

```
$ python use_of_str_2.py
True
False
Traceback (most recent call last):
  File "C:\Users\zhangyunfei\repo\week05\use_of_str_2.py", line 3, in <module>
    assert ""
    ^^
AssertionError
```

use_of_str_2.py

```
1  print("abc" > "ABC")
2  print("123" > "abcd")
3  assert "book"
4  assert ""
```

Print(iter(s)) 查看 s 是否可迭代 for 函数的使用 以 book 为例, b 为 0 位, o 为 1 位 【1: 3】表示第一位到第三位, 包含第一位不包含第三位 即为“oo”

```
use_of_str_2.py > ...
1 print("abc" > "ABC")
2 print("123" > "abcd")
3 assert "book"
4
5 s = "book"
6 print(iter(s))
7 for c in s:
8     print(c)
9
10 print(len(s))
11 s = "book"
12 assert s[1:3] == "bo"
13

$ python use_of_str_2.py
True
False
<str_ascii_iterator object at 0x00000238BA3E7760>
b
o
o
k
4
Traceback (most recent call last):
  File "C:\Users\zhangyunfei\repo\week05\use_of_str_2.py", line 12, in <module>
    assert s[1:3] == "bo"
    ^^^^^^^^^^^^^^^^^
AssertionError
(week05)
zhangyunfei@LAPTOP-NU6BCCKG MINGW64 ~/repo/week05 (main)
$ python use_of_str_2.py
True
False
<str_ascii_iterator object at 0x00000162899F7760>
b
o
o
k
4
```

join 的用法

```
9
10 print(len(s))
11 s = "book"
12 assert s[1:3] == "oo"
13
14 q = ["rose", "jack", "bob"]
15 print(":".join(q))
16
b
o
o
k
4
rose:jack:bob
(week05)
zhangyunfei@LAPTOP-NU6BCCKG MINGW64 ~/repo/week05
```

字节串编解码 字符串可以编码变成字节串

```
EXPLORER
WEEK05
.gitignore
environment.yml
LICENSE
README.md
use_of_bytes.py
use_of_str_1.py
use_of_str_2.py

tr_2.py U
use_of_bytes.py U
use_of_str_1.py U
use_of_str_2.py U

use_of_bytes.py > ...
1 s = b"hello"
2 print(s)
3 print(s[0])
4
5 s = "你好"
6 b = s.encode()
7 print(b)

MINGW64/c/Users/zhangyur X + v
zhangyunfei@LAPTOP-NU6BCCKG MINGW64 ~/repo/week05 (main)
$ python use_of_str_2.py
True
False
<str_ascii_iterator object at 0x000001E87ECB7910>
b
o
o
k
4
rose:jack:bob
(week05)
zhangyunfei@LAPTOP-NU6BCCKG MINGW64 ~/repo/week05 (main)
$ python use_of_bytes.py
b'hello'
104
(week05)
zhangyunfei@LAPTOP-NU6BCCKG MINGW64 ~/repo/week05 (main)
$ python use_of_bytes.py
b'hello'
104
(week05)
zhangyunfei@LAPTOP-NU6BCCKG MINGW64 ~/repo/week05 (main)
$ python use_of_bytes.py
b'hello'
104
b'\xe4\xbd\xa0\xe5\xa5\xbd'
(week05)
```

整数

```
(week05)
zhangyunfei@LAPTOP-NU6BCKG MINGW64 ~/repo/week0
5 (main)
$ python use_of_bytes.py
b'hello'
104
(week05)
zhangyunfei@LAPTOP-NU6BCKG MINGW64 ~/repo/week0
5 (main)
$ python use_of_bytes.py
b'hello'
104
b'\xe4\xbd\xa0\xe5\xa5\xbd'
(week05)
zhangyunfei@LAPTOP-NU6BCKG MINGW64 ~/repo/week0
5 (main)
$ python use_of_bytes.py
b'hello'
104
b'\xe4\xbd\xa0\xe5\xa5\xbd'
整数
5 1 6
(week05)
zhangyunfei@LAPTOP-NU6BCKG MINGW64 ~/repo/week0
5 (main)
```

浮点数

```
use_of_float.py > ...
1 import random
2
3 x = 3.14159
4 print(type(x))
5
6
7 y = float("3.14159")
8 print(type(y))
9 assert x == y
10
11
12 x = 10 / 3
13 print(x, type(x))
14
15 x = random.random()
16 print(x)
17
18 assert not 0.0
19
20 f = float("nan") ##特殊值
21 print(f)
22
23 print(3.14e2) ##3.14乘10的平方
24 inf = float("inf")
25 print(inf == inf) ##正无穷等于正无穷
26 ##浮点数不精确，支持各种运算

314.0
(week05)
zhangyunfei@LAPTOP-NU6BCKG MINGW64 ~/re
$ python use_of_float.py
<class 'float'>
<class 'float'>
3.3333333333333335 <class 'float'>
0.6056613900692682
nan
314.0
Traceback (most recent call last):
  File "C:\Users\zhangyunfei\repo\week05
    print(pinf == pinf)  ##正无穷等于正无穷
    ^^^^^
NameError: name 'pinf' is not defined. D
(week05)
zhangyunfei@LAPTOP-NU6BCKG MINGW64 ~/re
$ python use_of_float.py
<class 'float'>
<class 'float'>
3.3333333333333335 <class 'float'>
0.8629931557005736
nan
314.0
True
(week05)
zhangyunfei@LAPTOP-NU6BCKG MINGW64 ~/re
```

布尔值

```
zhangyunfei@LAPTOP-NU6BCKG MINGW64
$ python use_of_bool.py
True False
<class 'bool'>
(week05)
zhangyunfei@LAPTOP-NU6BCKG MINGW64
```

列表

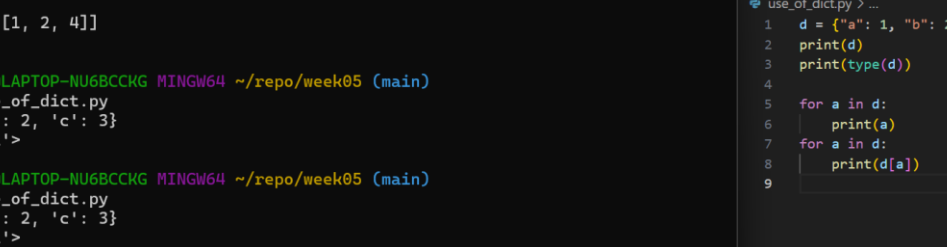
The screenshot shows a code editor with a file explorer on the left, a central editor window, and a terminal at the bottom. The file explorer shows a project named 'WEEK05' with files: .gitignore, environment.yml, LICENSE, README.md, use_of_bool.py, use_of_bytes.py, use_of_float.py, use_of_int.py, use_of_list.py (selected), use_of_str_1.py, use_of_str_2.py, and use_of_str.py. The central editor shows the content of 'use_of_list.py':

```
1 l = [1, 5, "abc"]
2 print(l)
3
4 print(l[0])
5
6 try:
7     print(l[3])
8 except IndexError as e:
9     print(e)
10
11 a = [2, 3]
12 print(1 + a)
13
14 print(1 + a == a + 1)
15 ##不支持-运算 支持*运算
16 print(a * 2)
17
18 b = a * 2
19 print(b)
20 a[0] = 3
21 print(a)
22 print(b)
23
24 a = [1, 2]
25 b = [a] * 2
26 print(b)
27 a[0] = 3
28 print(a)
29 print(b)
30
31 a = [2, 5, 3]
32 b = [i**2 for i in a] ##运算式
33 print(b)
34
```

The terminal at the bottom shows the output of running 'python use_of_list.py':

```
zhangyunfei@LAPTOP-NU6BCKG MINGW64 ~/repo/week05 (ma:
$ python use_of_list.py
[1, 5, 'abc']
1
list index out of range
[1, 5, 'abc', 2, 3]
False
[2, 3, 2, 3]
[2, 3, 2, 3]
[3, 3]
[2, 3, 2, 3]
[[1, 2], [1, 2]]
[3, 2]
[[3, 2], [3, 2]]
[4, 25, 9]
[[1, 2], [1, 2]]
[1, 2, 4]
[[1, 2, 4], [1, 2, 4]]
None
(week05)
zhangyunfei@LAPTOP-NU6BCKG MINGW64 ~/repo/week05 (ma:
$
```

字典



```
MINGW64~/c:/Users/zhangyur x + use_of_dict.py U x y 1.0 use_of_dict.py U x
1, 2, 4]
[1, 2, 4], [1, 2, 4]]
one
week05)
hangyunfei@LAPTOP-NU6BCCKG MINGW64 ~/repo/week05 (main)
python use_of_dict.py
{'a': 1, 'b': 2, 'c': 3}
class 'dict'>
week05)
hangyunfei@LAPTOP-NU6BCCKG MINGW64 ~/repo/week05 (main)
python use_of_dict.py
{'a': 1, 'b': 2, 'c': 3}
class 'dict'>
week05)
hangyunfei@LAPTOP-NU6BCCKG MINGW64 ~/repo/week05 (main)
python use_of_dict.py
{'a': 1, 'b': 2, 'c': 3}
class 'dict'>
```

集合以及集合的运算

```
MINGW64/c/Users/zhangyur x + - □ × use_of_set.py U x use_of_set.py > ...
<class 'set'>
unhashable type: 'list'
(week05)
zhangyunfei@LAPTOP-NU6BCCKG MINGW64 ~/repo/week05 (main)
$ python use_of_set.py
<class 'set'>
unhashable type: 'list'
[1, 2, 1, 2, 4, 5, 6]
{1, 2, 4, 5, 6}
(week05)
zhangyunfei@LAPTOP-NU6BCCKG MINGW64 ~/repo/week05 (main)
$ python use_of_set.py
<class 'set'>
unhashable type: 'list'
[1, 2, 1, 2, 4, 5, 6]
{1, 2, 4, 5, 6}
False
(week05)
zhangyunfei@LAPTOP-NU6BCCKG MINGW64 ~/repo/week05 (main)
$ python use_of_set.py
<class 'set'>
unhashable type: 'list'
[1, 2, 1, 2, 4, 5, 6]
{1, 2, 4, 5, 6}
False
{1, 2, 3, 4, 5, 6}
{1, 2}
(week05)
zhangyunfei@LAPTOP-NU6BCCKG MINGW64 ~/repo/week05 (main)
1 s = {1, 2, 3}
2 print(type(s))
3
4
5 try:
6     s = {1, [4], 7}
7 except TypeError as e:
8     print(e)
9
10 q = [1, 2, 1, 2, 4, 5, 6]
11 print(q)
12 s = set(q)
13 print(s)
14
15 print(3 in s) ##判断
16
17 s1 = {1, 2, 3}
18 print(s | s1)
19 print(s & s1)
```

Datetime

```
04/07 19:08:58 | 4 个文件 607.0 / MB | 157 516 378

MINGW64/c/Users/zhangyur x + v

(useweek05)
zhangyunfei@LAPTOP-NU6BCCWG MINGW64 ~/repo/week05 (main)
$ python use_of_set.py
<class 'set'>
unhashable type: 'list'
[1, 2, 1, 2, 4, 5, 6]
{1, 2, 4, 5, 6}
False
{1, 2, 3, 4, 5, 6}
{1, 2}
(useweek05)
zhangyunfei@LAPTOP-NU6BCCWG MINGW64 ~/repo/week05 (main)
$ python use_of_datetime.py
2025-04-11
173 days, 0:00:00
173
(useweek05)
zhangyunfei@LAPTOP-NU6BCCWG MINGW64 ~/repo/week05 (main)
$ python use_of_datetime.py
2025-04-11
173 days, 0:00:00
173
3 days, 0:00:00
(useweek05)
zhangyunfei@LAPTOP-NU6BCCWG MINGW64 ~/repo/week05 (main)

use_of_datetime.py x
use_of_datetime.py > ...
1 from datetime import date
2
3
4 print(date.today())
5
6 t1 = date.today()
7 t2 = date(2025, 10, 1)
8 td = t2 - t1
9 print(td)
10 print(td.days)
11
12 s1 = date(2024, 12, 31)
13 s2 = date(2025, 1, 3)
14 sd = s2 - s1
15 print(sd)
```

Pathlib 的使用

```
$ python use_of_path.py
True
C:\Users\zhangyunfei\repo\week05
[WindowsPath('.git'),
 WindowsPath('.gitignore'),
 WindowsPath('environment.yml'),
 WindowsPath('LICENSE'),
 WindowsPath('README.md'),
 WindowsPath('ues_of_bool.py'),
 WindowsPath('use_of_bytes.py'),
 WindowsPath('use_of_datetime.py'),
 WindowsPath('use_of_dict.py'),
 WindowsPath('use_of_float.py'),
 WindowsPath('use_of_int.py'),
 WindowsPath('use_of_list.py'),
 WindowsPath('use_of_path.py'),
 WindowsPath('use_of_set.py'),
 WindowsPath('use_of_str_1.py'),
 WindowsPath('use_of_str_2.py'),
 WindowsPath('use_of_tuple.py')]
False
True
True
README.md
C:\Users\zhangyunfei\repo\week05\README.md
(useweek05)
zhangyunfei@LAPTOP-NU6BCCWG MINGW64 ~/repo/week05 (main)

EXPLORER
WEEK05
> data
.gitignore
environment.yml U
LICENSE
README.md
ues_of_bool.py U
use_of_bytes.py U
use_of_datetime.py U
use_of_dict.py U
use_of_float.py U
use_of_int.py U
use_of_list.py U
use_of_path.py U
use_of_set.py U
use_of_str_1.py U
use_of_str_2.py U
use_of_str.py U
use_of_tuple.py U

use_of_path.py x
use_of_path.py > ...
1 from pathlib import Path # noqa:
2 from pprint import pprint
3
4 p = Path(".")
5 print(p)
6 print(p.exists())
7 print(p.absolute())
8 pprint(list(p.iterdir()))
9
10 p = Path("./data")
11 print(p.exists())
12 p.mkdir(exist_ok=True)
13 print(p.exists())
14 print(p.is_dir())
15
16
17 p = Path(".")
18 p1 = p / "README.md"
19 print(p1)
20 p2 = p1.absolute()
21 print(p2)
22
```