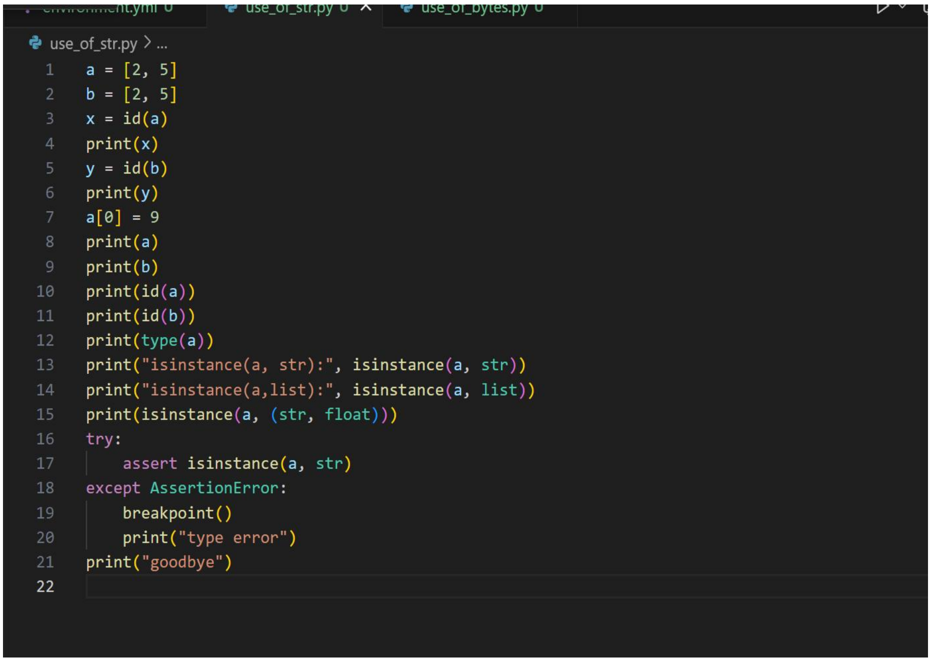
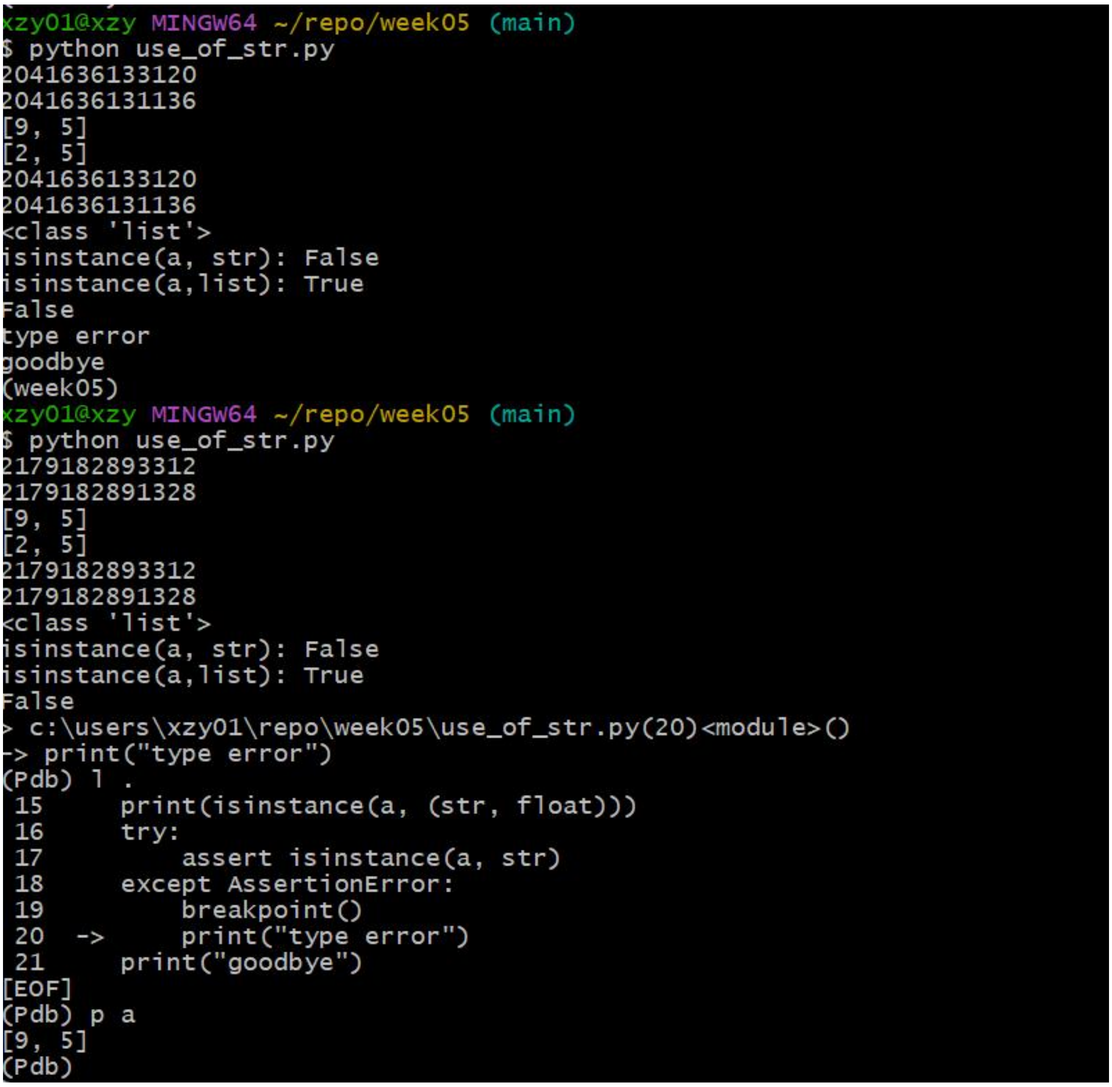
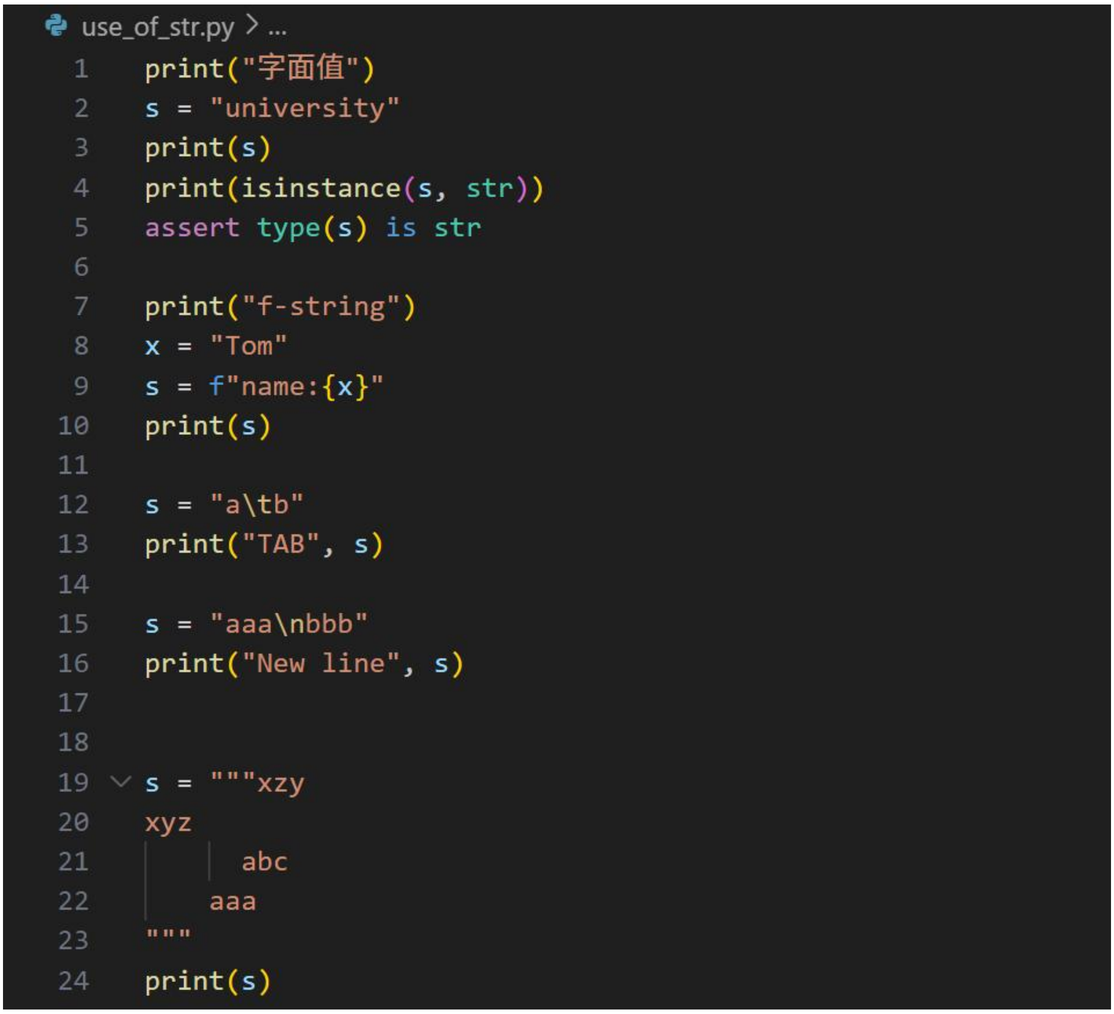
Str



```python
a = [2, 5]
b = [2, 5]
x = id(a)
print(x)
y = id(b)
print(y)
a[0] = 9
print(a)
print(b)
print(id(a))
print(id(b))
print(type(a))
print("isinstance(a, str):", isinstance(a, str))
print("isinstance(a,list):", isinstance(a, list))
print(isinstance(a, (str, float)))
try:
    assert isinstance(a, str)
except AssertionError:
    breakpoint()
    print("type error")
print("goodbye")

```

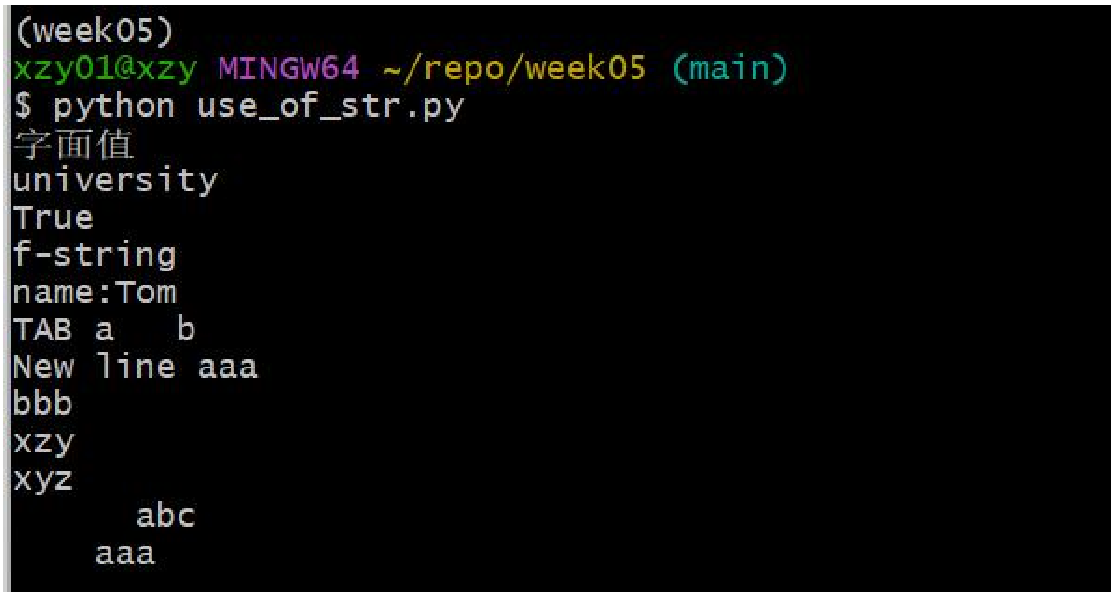


```
xzy01@xzy MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
2041636133120
2041636131136
[9, 5]
[2, 5]
2041636133120
2041636131136
<class 'list'>
isinstance(a, str): False
isinstance(a,list): True
False
type error
goodbye
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
2179182893312
2179182891328
[9, 5]
[2, 5]
2179182893312
2179182891328
<class 'list'>
isinstance(a, str): False
isinstance(a,list): True
False
> c:\users\xzy01\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) p a
[9, 5]
(Pdb)
```

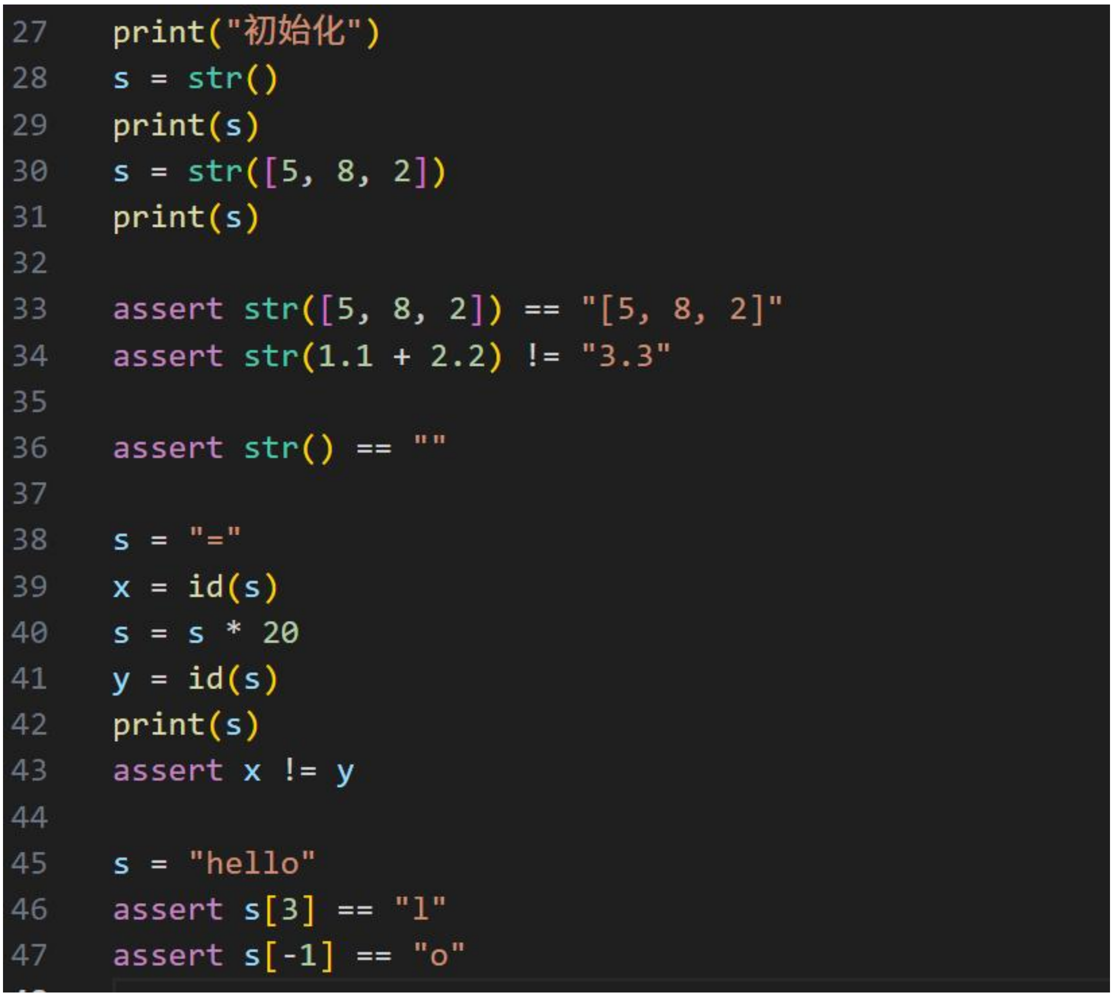


use_of_str.py > ...

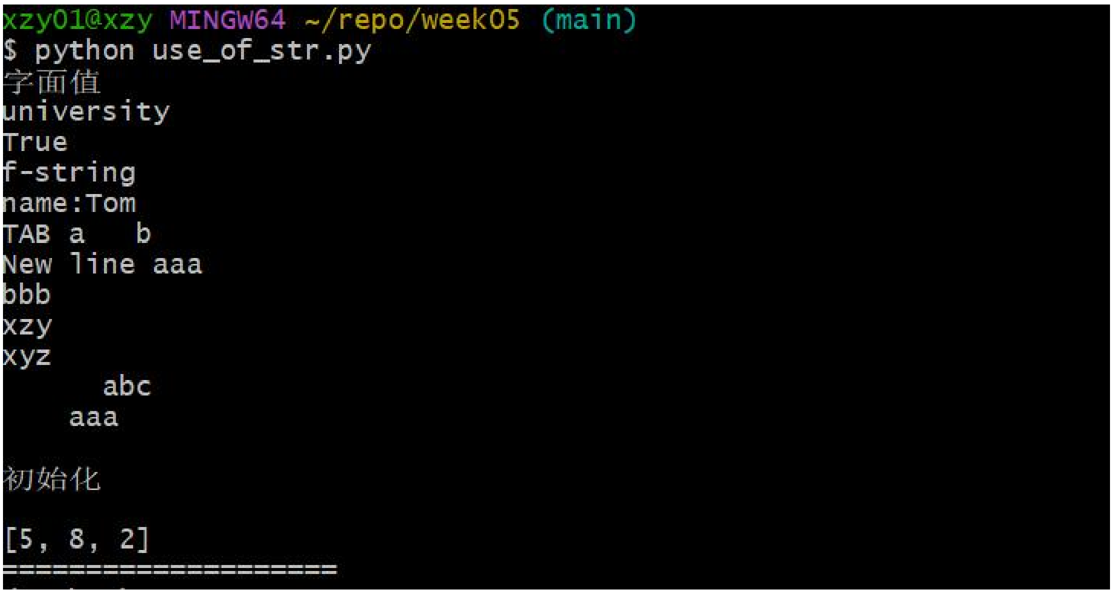
```python
print("字面值")
s = "university"
print(s)
print(isinstance(s, str))
assert type(s) is str

print("f-string")
x = "Tom"
s = f"name:{x}"
print(s)

s = "a\tb"
print("TAB", s)

s = "aaa\nbbb"
print("New line", s)


s = """xzy
xyz
      abc
    aaa
"""
print(s)
```



```
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
字面值
university
True
f-string
name:Tom
TAB a   b
New line aaa
bbb
xzy
xyz
      abc
    aaa
```

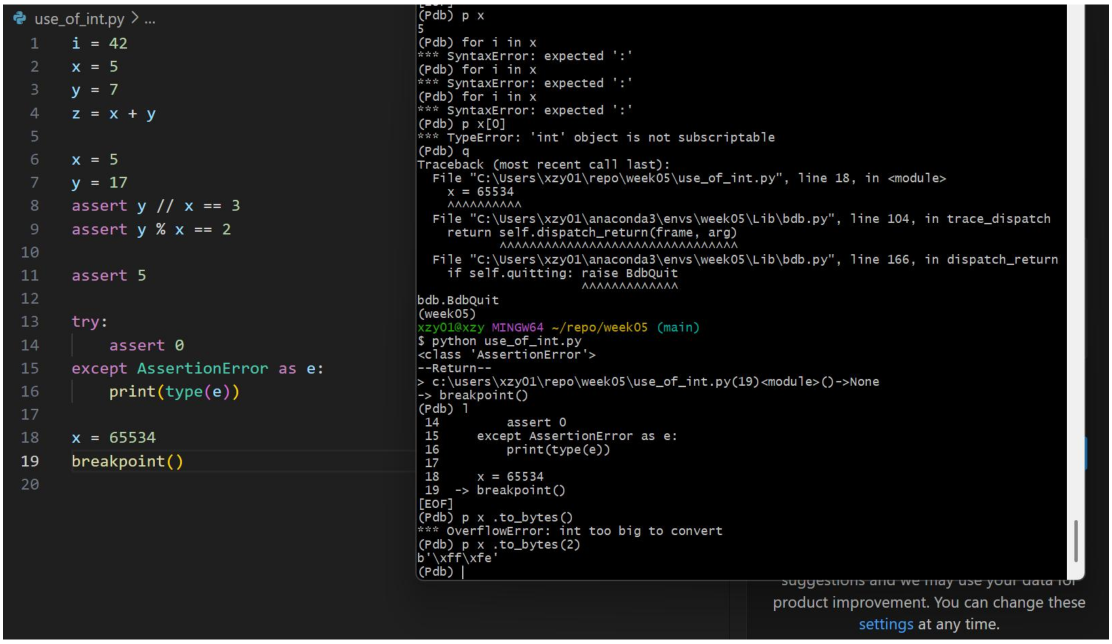


```python
    print("初始化")
    s = str()
    print(s)
    s = str([5, 8, 2])
    print(s)

    assert str([5, 8, 2]) == "[5, 8, 2]"
    assert str(1.1 + 2.2) != "3.3"

    assert str() == ""

    s = "="
    x = id(s)
    s = s * 20
    y = id(s)
    print(s)
    assert x != y

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



```
xzy01@xzy MINGW64 ~/repo/week05 (main)
$ python use_of_str.py
字面值
university
True
f-string
name:Tom
TAB a    b
New line aaa
bbb
xzy
xyz
      abc
    aaa

初始化

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

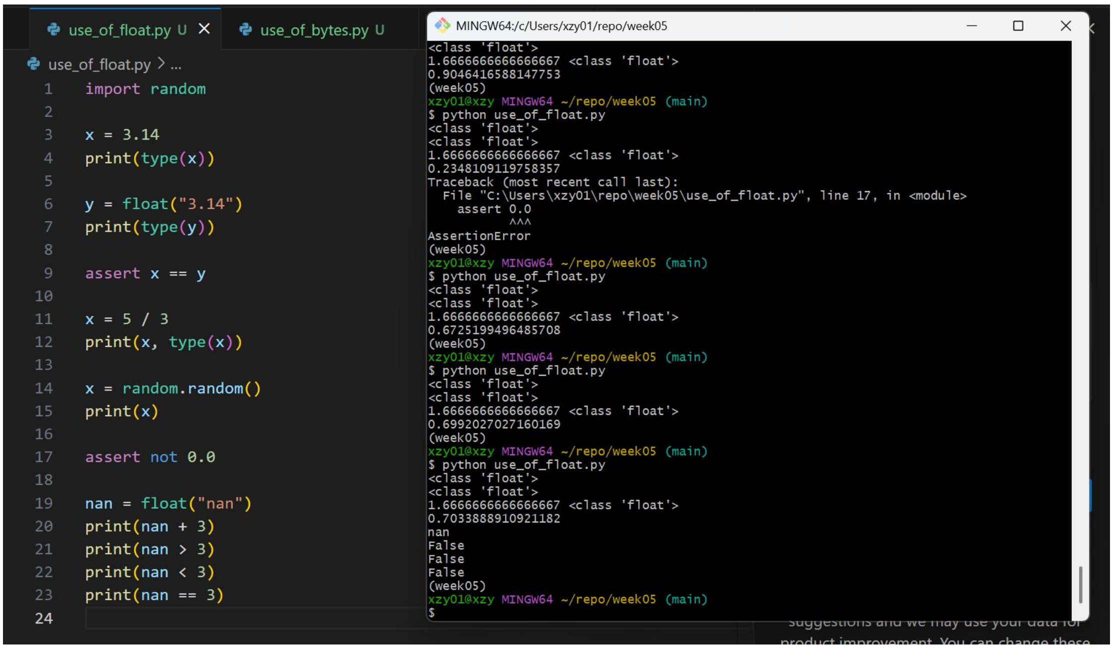
Int



```python
i = 42
x = 5
y = 7
z = x + y

x = 5
y = 17
assert y // x == 3
assert y % x == 2

assert 5

try:
    assert 0
except AssertionError as e:
    print(type(e))

x = 65534
breakpoint()

```

```
(Pdb) p x
5
(Pdb) for i in x
*** SyntaxError: expected ':'
(Pdb) for i in x
*** SyntaxError: expected ':'
(Pdb) for i in x
*** SyntaxError: expected ':'
(Pdb) p x[0]
*** TypeError: 'int' object is not subscriptable
(Pdb) q
Traceback (most recent call last):
  File "C:\Users\xzy01\repo\week05\use_of_int.py", line 18, in <module>
    x = 65534
    ^^^^^^^^^^
  File "C:\Users\xzy01\anaconda3\envs\week05\Lib\bdb.py", line 104, in trace_dispatch
    return self.dispatch_return(frame, arg)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\xzy01\anaconda3\envs\week05\Lib\bdb.py", line 166, in dispatch_return
    if self.quitting: raise BdbQuit
                      ^^^^^^^^^^^^^
bdb.BdbQuit
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$ python use_of_int.py
<class 'AssertionError'>
--Return--
> c:\users\xzy01\repo\week05\use_of_int.py(19)<module>()->None
-> breakpoint()
(Pdb) l
 14             assert 0
 15         except AssertionError as e:
 16             print(type(e))
 17
 18         x = 65534
 19     -> breakpoint()
[EOF]
(Pdb) p x .to_bytes()
*** OverflowError: int too big to convert
(Pdb) p x .to_bytes(2)
b'\xff\xfe'
(Pdb) |
```

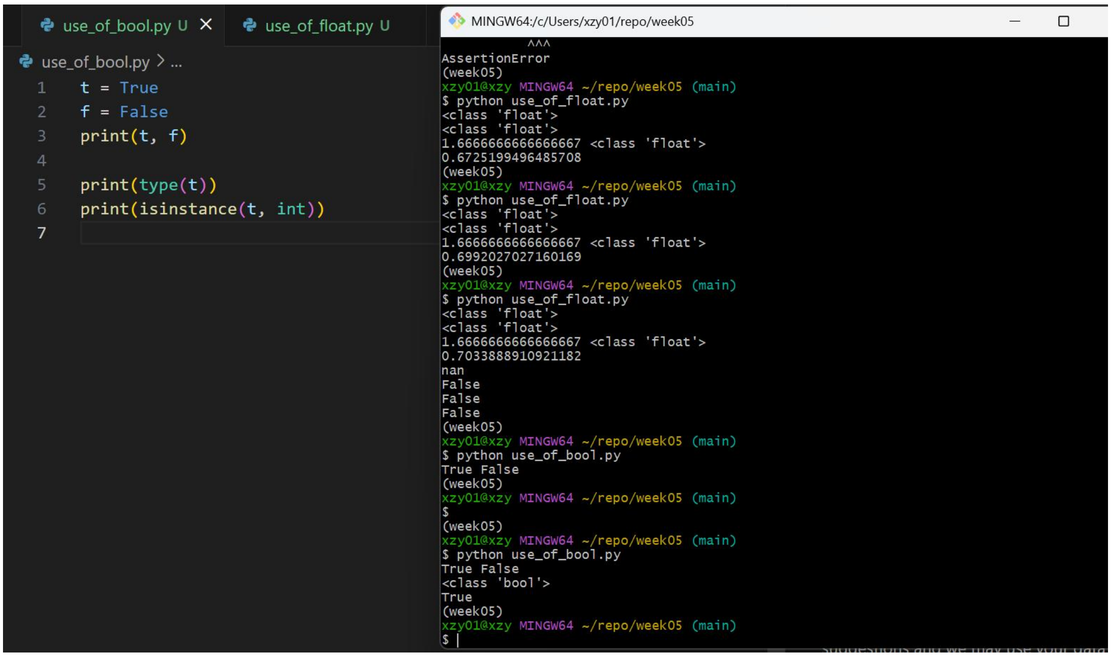
Float


use_of_float.py U ×
use_of_bytes.py U
use_of_float.py > ...
MINGW64:/c/Users/xzy01/repo/week05
product improvement. You can change these

```python
import random

x = 3.14
print(type(x))

y = float("3.14")
print(type(y))

assert x == y

x = 5 / 3
print(x, type(x))

x = random.random()
print(x)

assert not 0.0

nan = float("nan")
print(nan + 3)
print(nan > 3)
print(nan < 3)
print(nan == 3)

```

```
<class 'float'>
1.6666666666666667 <class 'float'>
0.9046416588147753
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$ python use_of_float.py
<class 'float'>
<class 'float'>
1.6666666666666667 <class 'float'>
0.2348109119758357
Traceback (most recent call last):
  File "C:\Users\xzy01\repo\week05\use_of_float.py", line 17, in <module>
    assert 0.0
           ^^^
AssertionError
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$ python use_of_float.py
<class 'float'>
<class 'float'>
1.6666666666666667 <class 'float'>
0.6725199496485708
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$ python use_of_float.py
<class 'float'>
<class 'float'>
1.6666666666666667 <class 'float'>
0.6992027027160169
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$ python use_of_float.py
<class 'float'>
<class 'float'>
1.6666666666666667 <class 'float'>
0.7033888910921182
nan
False
False
False
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$
```

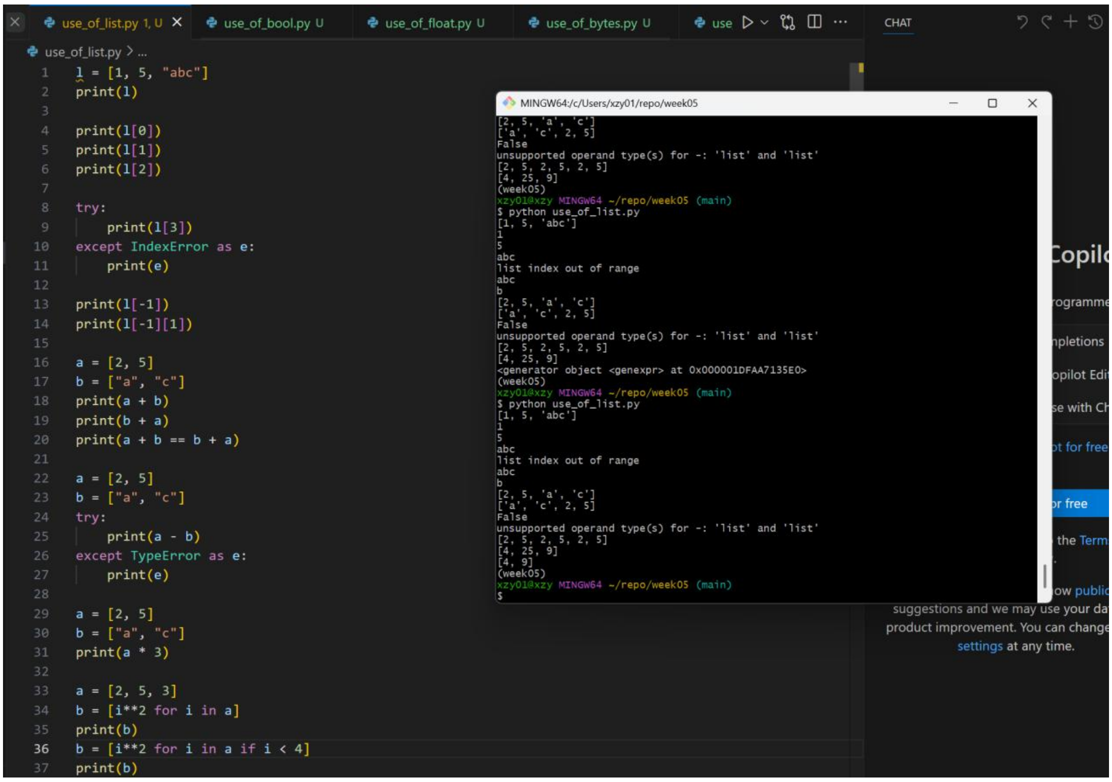
Bool



```python
t = True
f = False
print(t, f)

print(type(t))
print(isinstance(t, int))
```

```
           ^^^
AssertionError
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$ python use_of_float.py
<class 'float'>
<class 'float'>
1.6666666666666667 <class 'float'>
0.6725199496485708
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$ python use_of_float.py
<class 'float'>
<class 'float'>
1.6666666666666667 <class 'float'>
0.6992027027160169
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$ python use_of_float.py
<class 'float'>
<class 'float'>
1.6666666666666667 <class 'float'>
0.7033888910921182
nan
False
False
False
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$ python use_of_bool.py
True False
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$ python use_of_bool.py
True False
<class 'bool'>
True
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$ |
```

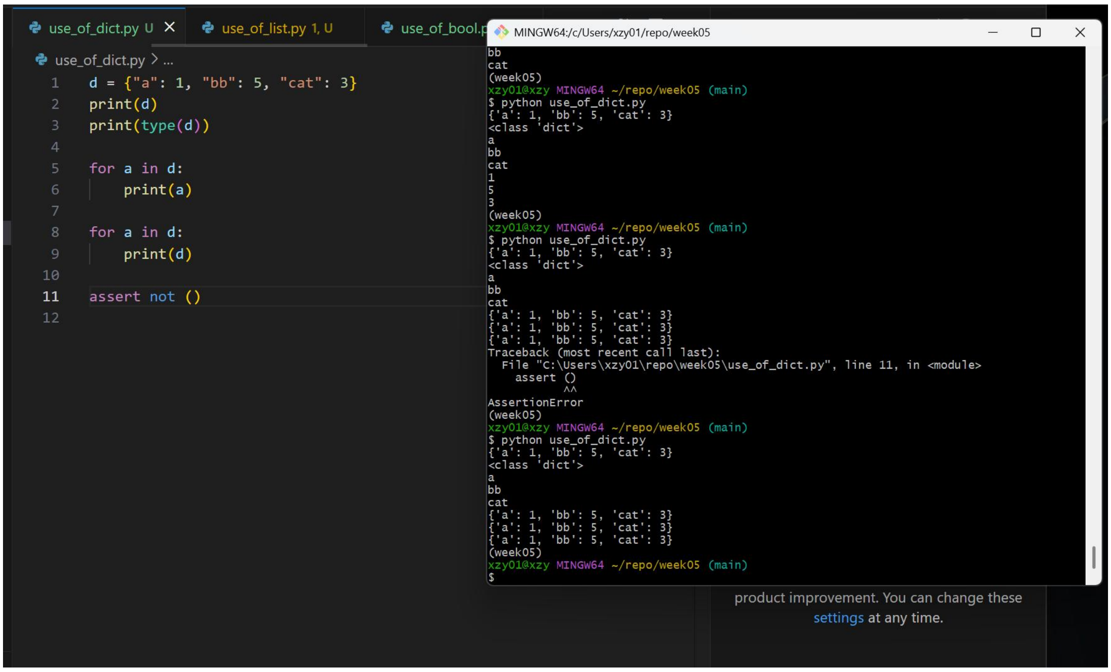
List



```python
l = [1, 5, "abc"]
print(l)

print(l[0])
print(l[1])
print(l[2])

try:
    print(l[3])
except IndexError as e:
    print(e)

print(l[-1])
print(l[-1][1])

a = [2, 5]
b = ["a", "c"]
print(a + b)
print(b + a)
print(a + b == b + a)

a = [2, 5]
b = ["a", "c"]
try:
    print(a - b)
except TypeError as e:
    print(e)

a = [2, 5]
b = ["a", "c"]
print(a * 3)

a = [2, 5, 3]
b = [i**2 for i in a]
print(b)
b = [i**2 for i in a if i < 4]
print(b)
```

```
[2, 5, 'a', 'c']
['a', 'c', 2, 5]
False
unsupported operand type(s) for -: 'list' and 'list'
[2, 5, 2, 5, 2, 5]
[4, 25, 9]
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$ python use_of_list.py
[1, 5, 'abc']
1
5
abc
list index out of range
abc
b
[2, 5, 'a', 'c']
['a', 'c', 2, 5]
False
unsupported operand type(s) for -: 'list' and 'list'
[2, 5, 2, 5, 2, 5]
[4, 25, 9]
<generator object <genexpr> at 0x000001DFAA7135E0>
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$ python use_of_list.py
[1, 5, 'abc']
1
5
abc
list index out of range
abc
b
[2, 5, 'a', 'c']
['a', 'c', 2, 5]
False
unsupported operand type(s) for -: 'list' and 'list'
[2, 5, 2, 5, 2, 5]
[4, 25, 9]
[4, 9]
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$
```

Dic

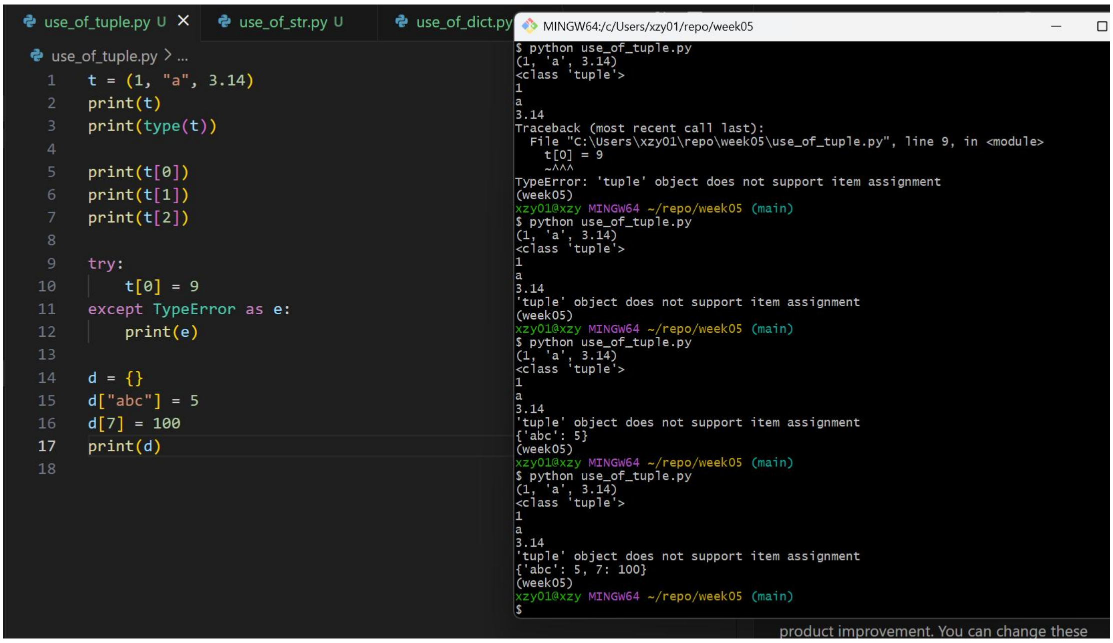


```python
d = {"a": 1, "bb": 5, "cat": 3}
print(d)
print(type(d))

for a in d:
    print(a)

for a in d:
    print(d)

assert not ()

```

```
bb
cat
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$ python use_of_dict.py
{'a': 1, 'bb': 5, 'cat': 3}
<class 'dict'>
a
bb
cat
1
5
3
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$ python use_of_dict.py
{'a': 1, 'bb': 5, 'cat': 3}
<class 'dict'>
a
bb
cat
{'a': 1, 'bb': 5, 'cat': 3}
{'a': 1, 'bb': 5, 'cat': 3}
{'a': 1, 'bb': 5, 'cat': 3}
Traceback (most recent call last):
  File "C:\Users\xzy01\repo\week05\use_of_dict.py", line 11, in <module>
    assert ()
           ^^
AssertionError
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$ python use_of_dict.py
{'a': 1, 'bb': 5, 'cat': 3}
<class 'dict'>
a
bb
cat
{'a': 1, 'bb': 5, 'cat': 3}
{'a': 1, 'bb': 5, 'cat': 3}
{'a': 1, 'bb': 5, 'cat': 3}
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$
```

product improvement. You can change these settings at any time.

Tuple

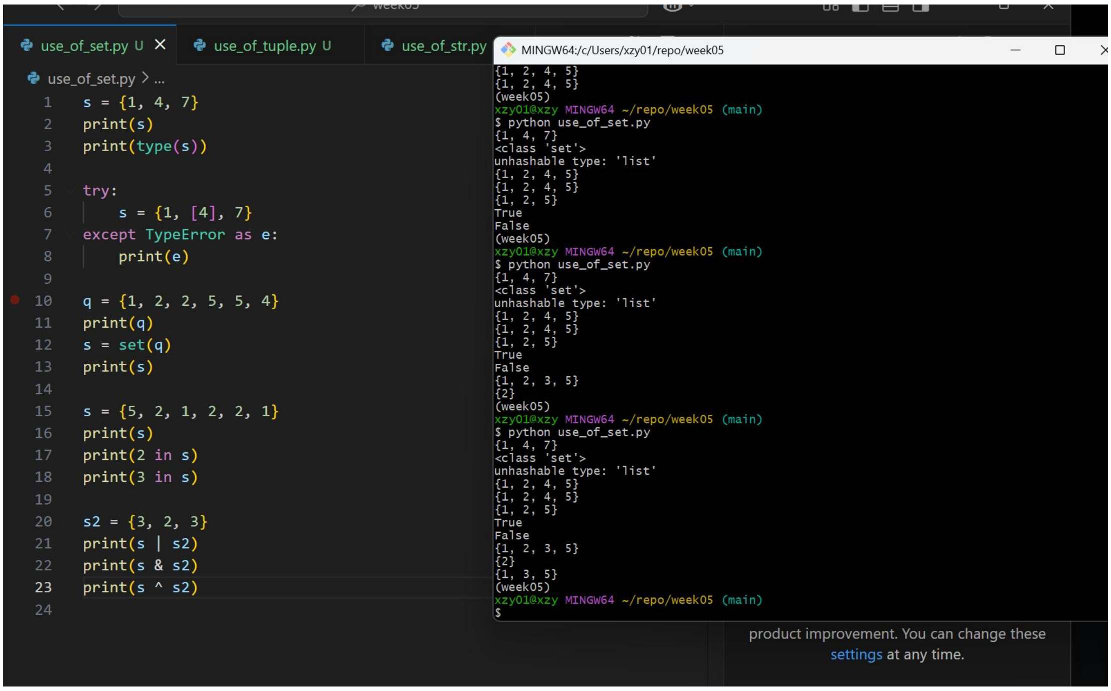


```python
t = (1, "a", 3.14)
print(t)
print(type(t))

print(t[0])
print(t[1])
print(t[2])

try:
    t[0] = 9
except TypeError as e:
    print(e)

d = {}
d["abc"] = 5
d[7] = 100
print(d)

```

```
$ python use_of_tuple.py
(1, 'a', 3.14)
<class 'tuple'>
1
a
3.14
Traceback (most recent call last):
  File "C:\Users\xzy01\repo\week05\use_of_tuple.py", line 9, in <module>
    t[0] = 9
    ~^^^
TypeError: 'tuple' object does not support item assignment
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$ python use_of_tuple.py
(1, 'a', 3.14)
<class 'tuple'>
1
a
3.14
'tuple' object does not support item assignment
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$ python use_of_tuple.py
(1, 'a', 3.14)
<class 'tuple'>
1
a
3.14
'tuple' object does not support item assignment
{'abc': 5}
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$ python use_of_tuple.py
(1, 'a', 3.14)
<class 'tuple'>
1
a
3.14
'tuple' object does not support item assignment
{'abc': 5, 7: 100}
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$
```

product improvement. You can change these

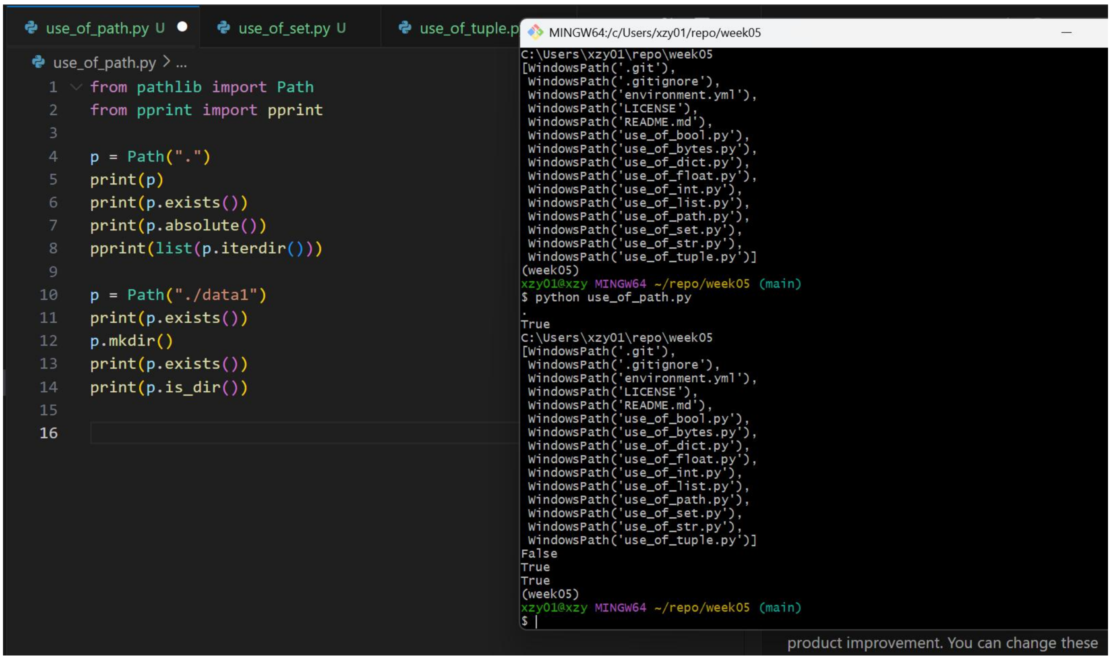
Set


use_of_set.py U
use_of_tuple.py U
use_of_str.py
use_of_set.py > ...
MINGW64:/c/Users/xzy01/repo/week05
product improvement. You can change these settings at any time.

```python
s = {1, 4, 7}
print(s)
print(type(s))

try:
    s = {1, [4], 7}
except TypeError as e:
    print(e)

q = {1, 2, 2, 5, 5, 4}
print(q)
s = set(q)
print(s)

s = {5, 2, 1, 2, 2, 1}
print(s)
print(2 in s)
print(3 in s)

s2 = {3, 2, 3}
print(s | s2)
print(s & s2)
print(s ^ s2)
```

```
{1, 2, 4, 5}
{1, 2, 4, 5}
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$ python use_of_set.py
{1, 4, 7}
<class 'set'>
unhashable type: 'list'
{1, 2, 4, 5}
{1, 2, 4, 5}
{1, 2, 5}
True
False
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$ python use_of_set.py
{1, 4, 7}
<class 'set'>
unhashable type: 'list'
{1, 2, 4, 5}
{1, 2, 4, 5}
{1, 2, 5}
True
False
{1, 2, 3, 5}
{2}
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$ python use_of_set.py
{1, 4, 7}
<class 'set'>
unhashable type: 'list'
{1, 2, 4, 5}
{1, 2, 4, 5}
{1, 2, 5}
True
False
{1, 2, 3, 5}
{2}
{1, 3, 5}
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$
```

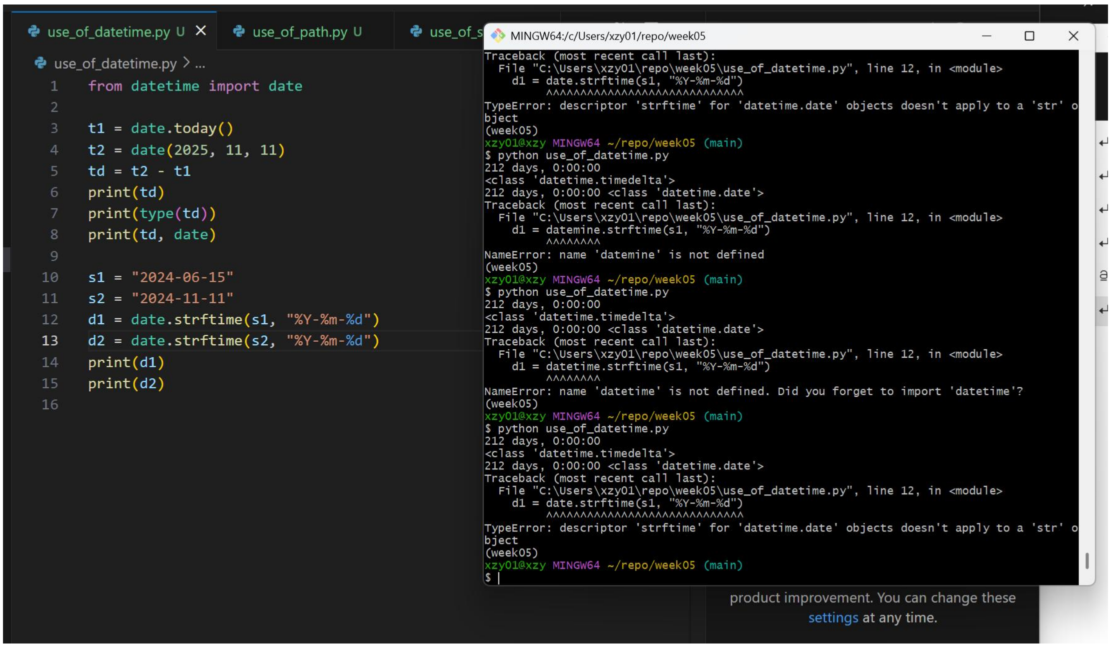
Path


use_of_path.py U
use_of_set.py U
use_of_tuple.p
use_of_path.py > ...
MINGW64:/c/Users/xzy01/repo/week05
product improvement. You can change these

```python
from pathlib import Path
from pprint import pprint

p = Path(".")
print(p)
print(p.exists())
print(p.absolute())
pprint(list(p.iterdir()))

p = Path("./data1")
print(p.exists())
p.mkdir()
print(p.exists())
print(p.is_dir())
```

```
C:\Users\xzy01\repo\week05
[WindowsPath('.git'),
 WindowsPath('.gitignore'),
 WindowsPath('environment.yml'),
 WindowsPath('LICENSE'),
 WindowsPath('README.md'),
 WindowsPath('use_of_bool.py'),
 WindowsPath('use_of_bytes.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.py'),
 WindowsPath('use_of_tuple.py')]
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$ python use_of_path.py
.
True
C:\Users\xzy01\repo\week05
[WindowsPath('.git'),
 WindowsPath('.gitignore'),
 WindowsPath('environment.yml'),
 WindowsPath('LICENSE'),
 WindowsPath('README.md'),
 WindowsPath('use_of_bool.py'),
 WindowsPath('use_of_bytes.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.py'),
 WindowsPath('use_of_tuple.py')]
False
True
True
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$ |
```

Datetime



```python
from datetime import date

t1 = date.today()
t2 = date(2025, 11, 11)
td = t2 - t1
print(td)
print(type(td))
print(td, date)

s1 = "2024-06-15"
s2 = "2024-11-11"
d1 = date.strftime(s1, "%Y-%m-%d")
d2 = date.strftime(s2, "%Y-%m-%d")
print(d1)
print(d2)

```

```
Traceback (most recent call last):
  File "C:\Users\xzy01\repo\week05\use_of_datetime.py", line 12, in <module>
    d1 = date.strftime(s1, "%Y-%m-%d")
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
TypeError: descriptor 'strftime' for 'datetime.date' objects doesn't apply to a 'str' o
bject
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$ python use_of_datetime.py
212 days, 0:00:00
<class 'datetime.timedelta'>
212 days, 0:00:00 <class 'datetime.date'>
Traceback (most recent call last):
  File "C:\Users\xzy01\repo\week05\use_of_datetime.py", line 12, in <module>
    d1 = datemine.strftime(s1, "%Y-%m-%d")
         ^^^^^^^^
NameError: name 'datemine' is not defined
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$ python use_of_datetime.py
212 days, 0:00:00
<class 'datetime.timedelta'>
212 days, 0:00:00 <class 'datetime.date'>
Traceback (most recent call last):
  File "C:\Users\xzy01\repo\week05\use_of_datetime.py", line 12, in <module>
    d1 = datetime.strftime(s1, "%Y-%m-%d")
         ^^^^^^^^
NameError: name 'datetime' is not defined. Did you forget to import 'datetime'?
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$ python use_of_datetime.py
212 days, 0:00:00
<class 'datetime.timedelta'>
212 days, 0:00:00 <class 'datetime.date'>
Traceback (most recent call last):
  File "C:\Users\xzy01\repo\week05\use_of_datetime.py", line 12, in <module>
    d1 = date.strftime(s1, "%Y-%m-%d")
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
TypeError: descriptor 'strftime' for 'datetime.date' objects doesn't apply to a 'str' o
bject
(week05)
xzy01@xzy MINGW64 ~/repo/week05 (main)
$ 
```