

第五周学习报告

一、前期准备

1、fork 第五周打卡仓库到自己名下，选择 clone SSH 地址到 repo 目录下

2、将第四周文件夹中 environment.yml 文件复制到 week05 下，并创建 conda 环境

```
Last login: Mon Mar 31 16:53:56 on ttys000
(base) donghengxu@donghengxudeMacBook-Air ~ % cd repo
(base) donghengxu@donghengxudeMacBook-Air repo % ls
myproject  prj3      week01     week03
mywork     script1.py week02     week04
(base) donghengxu@donghengxudeMacBook-Air repo % git clone git@github.com:hengxudong/week05.git
Cloning into 'week05'...
Enter passphrase for key '/Users/donghengxu/.ssh/id_ed25519':
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) donghengxu@donghengxudeMacBook-Air repo % ls
myproject  prj3      week01     week03      week05
mywork     script1.py week02     week04
(base) donghengxu@donghengxudeMacBook-Air repo % cat week04/environment.yml
name: week04
channels:
  - conda-forge
dependencies:
  - python=3.12
  - matplotlib
(base) donghengxu@donghengxudeMacBook-Air repo % cp week04/environment.yml week05/
(base) donghengxu@donghengxudeMacBook-Air repo % ls -l week05
total 56
-rw-r--r--  1 donghengxu  staff  18411  4  7 18:56 LICENSE
-rw-r--r--  1 donghengxu  staff  2216  4  7 18:56 README.md
-rw-r--r--  1 donghengxu  staff    86  4  7 18:58 environment.yml
(base) donghengxu@donghengxudeMacBook-Air repo % cd week05
(base) donghengxu@donghengxudeMacBook-Air week05 % ls -l
total 56
-rw-r--r--  1 donghengxu  staff  18411  4  7 18:56 LICENSE
-rw-r--r--  1 donghengxu  staff  2216  4  7 18:56 README.md
-rw-r--r--  1 donghengxu  staff    86  4  7 19:00 environment.yml
(base) donghengxu@donghengxudeMacBook-Air week05 % conda env create
Retrieving notices: done
Channels:
 - conda-forge
 - defaults
 - https://repo.anaconda.com/pkgs/main
 - https://repo.anaconda.com/pkgs/r
Platform: osx-arm64
Collecting package metadata (repodata.json): done
Solving environment: done

Downloading and Extracting Packages:

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#   $ conda activate week05
#
# To deactivate an active environment, use
#
#   $ conda deactivate
#
(base) donghengxu@donghengxudeMacBook-Air week05 %
```

二、对于任何对象，都可以传给以下内置函数用于检视

对于字符串对象的操作：

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)

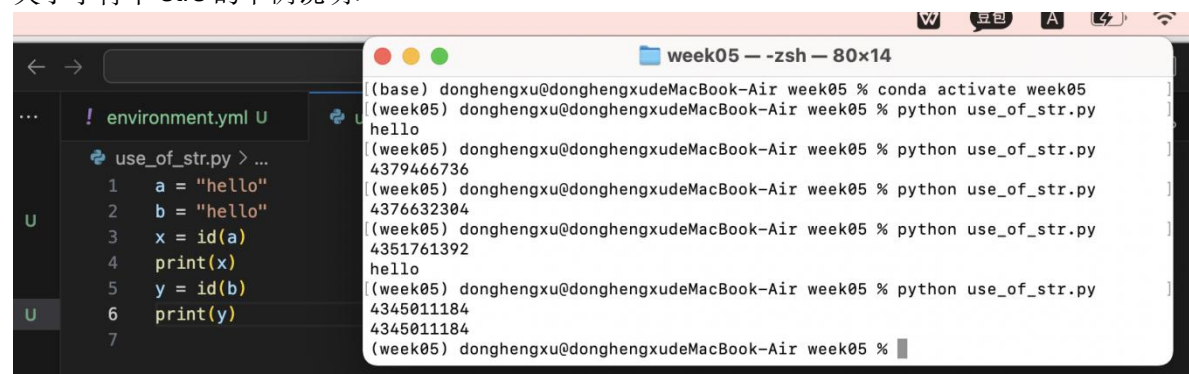
type() -- 返回对象的类型

isinstance() -- 判断对象是否属于某个（或某些）类型

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

关于字符串 strs 的举例说明：



```
week05 -- -zsh -- 80x14
(base) donghengxu@donghengxudeMacBook-Air week05 % conda activate week05
(base) donghengxu@donghengxudeMacBook-Air week05 % python use_of_str.py
hello
(base) donghengxu@donghengxudeMacBook-Air week05 % python use_of_str.py
4379466736
(base) donghengxu@donghengxudeMacBook-Air week05 % python use_of_str.py
4376632304
(base) donghengxu@donghengxudeMacBook-Air week05 % python use_of_str.py
4351761392
hello
(base) donghengxu@donghengxudeMacBook-Air week05 % python use_of_str.py
4345011184
4345011184
(base) donghengxu@donghengxudeMacBook-Air week05 %
```

```
environment.yml U 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 print(isinstance(a, str))
14 print(isinstance(a, list))
15 print(dir(a))
16 print(str(a))
17 print(isinstance(a, (str, list)))
18 print(isinstance(a, (str, float)))

(week05) donghengxu@donghengxudeMacBook-Air week05 % python use_of_str.py
4376183232
4376185088
[9, 5]
[2, 5]
4376183232
4376185088
<class 'list'>
False
True
['__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']
[9, 5]
True
False
(week05) donghengxu@donghengxudeMacBook-Air week05 %
```

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

```
18 print(isinstance(a, (str, float)))
19 assert isinstance(a, str)
20 print("good")
21

False
Traceback (most recent call last):
  File "/Users/donghengxu/repo/week05/use_of_str.py", line 19, in <module>
    assert isinstance(a, str)
    ^^^^^^^^^^^^^^^^^^^^^^^
AssertionError
(week05) donghengxu@donghengxudeMacBook-Air week05 %
```

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

```
19 try:
20     assert isinstance(a, str)
21 except AssertionError:
22     print("type error")
23     print("good")
24

type error
good
(week05) donghengxu@donghengxudeMacBook-Air week05 %
```

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

```
20 assert isinstance(a, str)
21 except AssertionError:
22     breakpoint()
23     print("type error")
24     print("good")
25

True
False
> /Users/donghengxu/repo/week05/use_of_str.py(23)<module>()
-> print("type error")
(Pdb)
```

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)

```

13 print(s)
14
15 s = "a\tb"
16 print("TAB", s)
17
18 s = "abc\nxyz"
19 print("new words", s)
20
21 s = """xyz
22 aaa
23 bbb
24 | ccc"""
25 print(s)
26
27 print("初始化")
28 s = str()
29 print(s)
30 s = str([5, 8, 2])
31 print(s, type(s))
32
33 assert str([5, 8, 2]) == "[5, 8, 2]"
34 assert str(1.1 + 2.2) != "3.3"
35
36
37 s = "abc"
38 s1 = s * 5
39 print(s1)
40
41 s = "hello world"
42 assert s[-1] == "d"
43 assert s[5] == " " # 注意索引值字面值的第一位应该是第0个, 最后一个为-1
44

```

初始化

```

[5, 8, 2] <class 'str'>
abcbcabcbcabcb
Traceback (most recent call last):
  File "/Users/donghengxu/repo/week05/use_of_str.py", line 43, in <module>
    assert s[6] == " "
    ^^^^^^^^^^^^^
AssertionError
[(week05) donghengxu@donghengxudeMacBook-Air week05 % python use_of_str.py
字面值
university
True
<class 'str'>
good
f-string
name: Tom
TAB a b
new words abc
xyz
xyz
aaa
bbb
ccc
初始化
[5, 8, 2] <class 'str'>
abcbcabcbcabcb
> /Users/donghengxu/repo/week05/use_of_str.py(44)<module>()
-> assert s[6] == " "
[(Pdb) p s[6]
'w'
[(Pdb) p s[5]
' '
(Pdb) p s[5]

```

```

def rstrip(chars=None, /) # Return a copy of the string with trailing whitespace removed...
def split(sep=None, maxsplit=-1) # Return a list of the substrings in the string, using sep as the separator string...
def splitlines(keepends=False) # Return a list of the lines in the string, breaking at line boundaries...
def startswith(prefix, start=0, end=None) # bool
def strip(chars=None, /) # Return a copy of the string with leading and trailing whitespace removed...
def swapcase() # Convert uppercase characters to lowercase and lowercase characters to uppercase...
def title() # Return a version of the string where each word is titlecased...
def translate(table, /) # Replace each character in the string using the given translation table...
def upper() # Return a copy of the string converted to uppercase...
def zfill(width, /) # Pad a numeric string with zeros on the left, to fill a field of the given width...

```

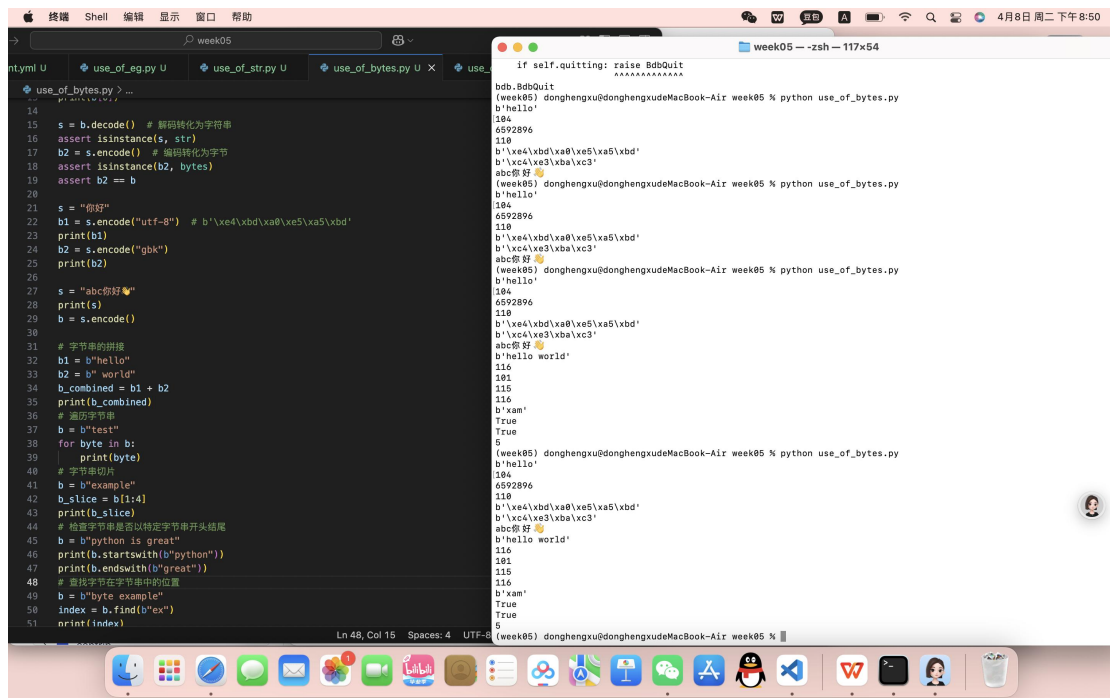
(Pdb) q
(week05) donghengxu@donghengxudeMacBook-Air week05 % python use_of_str.py

```

字面值
university
True
<class 'str'>
good
f-string
name: Tom
TAB a b
new words abc
xyz
xyz
aaa
bbb
ccc
初始化
[5, 8, 2] <class 'str'>
abcbcabcbcabcb
error
HELLO PYTHON
abc
fruit:apple,price2
abcbxyz
xyzabc
unsupported operand type(s) for -: 'str' and 'str'
unsupported operand type(s) for /: 'str' and 'int'
True
False
True
False
<str_ascii_iterator object at 0x103835060>
b
o
k
18
110
Hello world hello python
Hello World Hello Python
HELLO WORLD HELLO PYTHON
3
True
False
True
True
False
apple:peach:berry
['apple', 'peach', 'berry']
(week05) donghengxu@donghengxudeMacBook-Air week05 %

```

关于字节 bytes 的举例说明（代码运行结果）



```
if self.quitting: raise BdbQuit
AAAAAAAAAAAA
bdb.BdbQuit
(week05) donghengxu@donghengxudeMacBook-Air week05 % python use_of_bytes.py
b'hello'
104
6592896
118
b'\xe4\xbd\xa0\xe5\xa5\xbd'
b'\xc4\xe3\xba\xc3'
abc你好
(week05) donghengxu@donghengxudeMacBook-Air week05 % python use_of_bytes.py
b'hello'
104
6592896
118
b'\xe4\xbd\xa0\xe5\xa5\xbd'
b'\xc4\xe3\xba\xc3'
abc你好
(week05) donghengxu@donghengxudeMacBook-Air week05 % python use_of_bytes.py
b'hello'
104
6592896
118
b'\xe4\xbd\xa0\xe5\xa5\xbd'
b'\xc4\xe3\xba\xc3'
abc你好
b'hello world'
116
102
115
116
b'xam'
True
True
5
(week05) donghengxu@donghengxudeMacBook-Air week05 % python use_of_bytes.py
b'hello'
104
6592896
118
b'\xe4\xbd\xa0\xe5\xa5\xbd'
b'\xc4\xe3\xba\xc3'
abc你好
b'hello world'
116
102
115
116
b'xam'
True
True
5
```

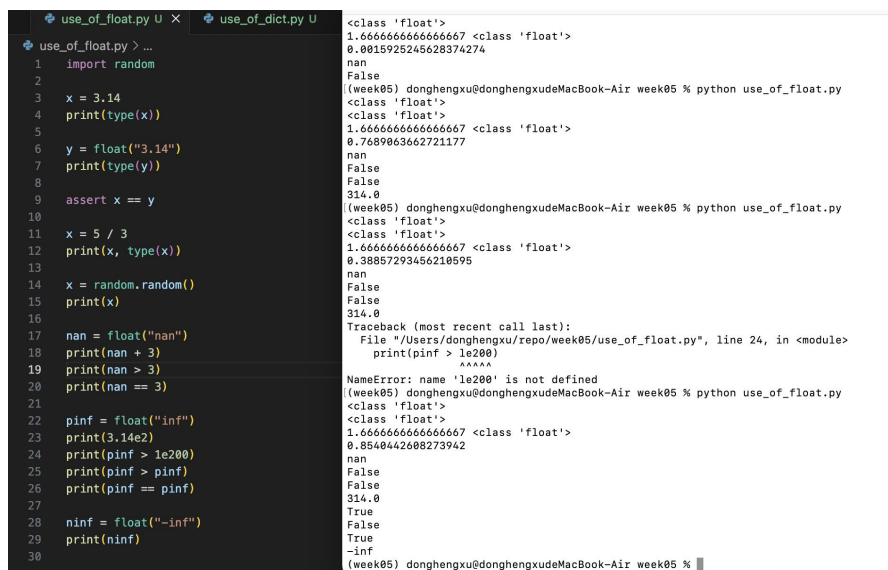
关于整数对象 int 的举例说明（代码运行结果）

整数不能迭代，提取索引，返回长度，但可以转化为字节（数字太大不可：



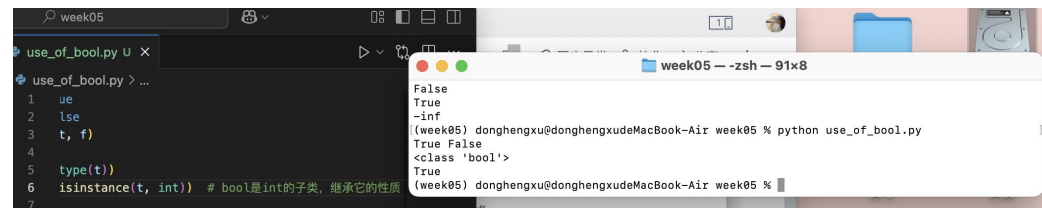
```
(Pdb) for i in x
*** SyntaxError: expected ':'
(Pdb) for i in x:
*** IndentationError: expected an indented block after 'for' statement on line 1
(Pdb) for i in x:print(i)
*** TypeError: 'int' object is not iterable
(Pdb) p len(x)
*** TypeError: object of type 'int' has no len()
(Pdb) p x.to_bytes()
b'\x05'
(Pdb) q
(week05) donghengxu@donghengxudeMacBook-Air week05 % python use_of_int.py
33
3
3.4
2
0b1010
0b12
0xa
10 10 10
3.0
5
4
3
(week05) donghengxu@donghengxudeMacBook-Air week05 %
```

关于浮点数对象 float 的举例说明（代码运行结果）



```
<class 'float'>
1.6666666666666667 <class 'float'>
0.0015925245628374274
nan
False
(week05) donghengxu@donghengxudeMacBook-Air week05 % python use_of_float.py
<class 'float'>
<class 'float'>
1.6666666666666667 <class 'float'>
0.7689063662721177
nan
False
False
314.0
(week05) donghengxu@donghengxudeMacBook-Air week05 % python use_of_float.py
<class 'float'>
<class 'float'>
1.6666666666666667 <class 'float'>
0.38857293456218595
nan
False
False
314.0
Traceback (most recent call last):
  File "/Users/donghengxu/repo/week05/use_of_float.py", line 24, in <module>
    print(pinf > 1e200)
    ^^^^^
NameError: name '1e200' is not defined
(week05) donghengxu@donghengxudeMacBook-Air week05 % python use_of_float.py
<class 'float'>
1.6666666666666667 <class 'float'>
0.8540442688273942
nan
False
False
314.0
True
False
True
-inf
(week05) donghengxu@donghengxudeMacBook-Air week05 %
```

关于布尔值对象 bool 的说明

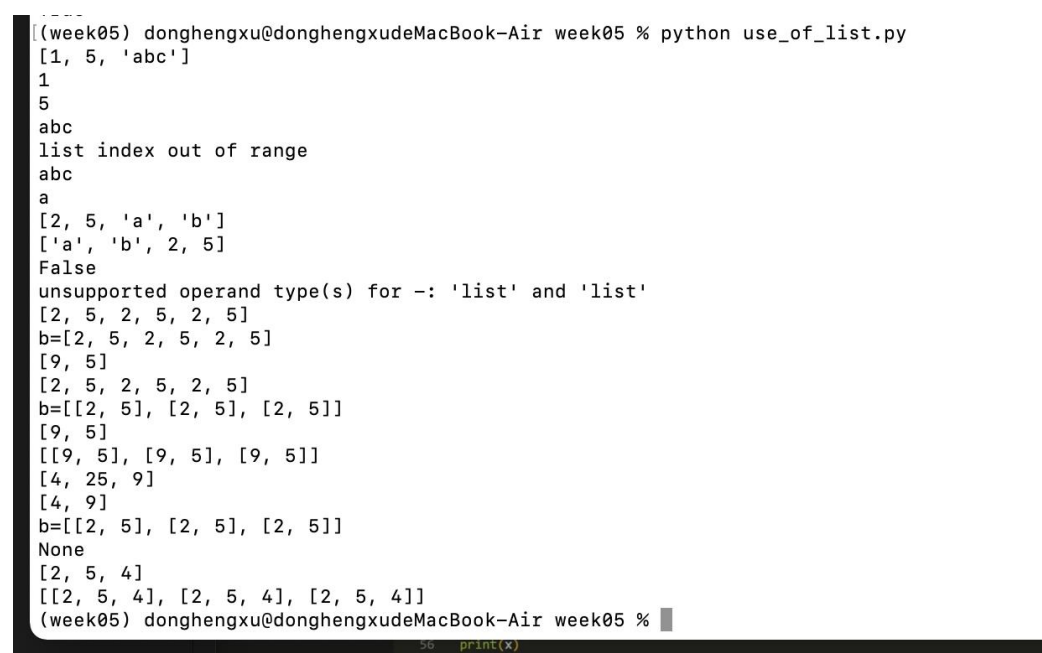


The screenshot shows a code editor with a file named `use_of_bool.py` and a terminal window. The code in the editor defines a function `use` that takes two arguments, `t` and `f`, and returns `t` if `f` is `True`, otherwise it returns `f`. It also checks if `t` is an instance of `int` and prints a comment. The terminal output shows the function being called with `False` and `True`, and the result of the `isinstance` check.

```
1 def use(t, f):
2     if f:
3         return t
4     else:
5         return f
6
7 type(t)
8 isinstance(t, int) # bool是int的子类, 继承它的性质
```

```
False
True
-inf
(True) donghengxu@donghengxudeMacBook-Air week05 % python use_of_bool.py
True False
<class 'bool'>
True
(True) donghengxu@donghengxudeMacBook-Air week05 %
```

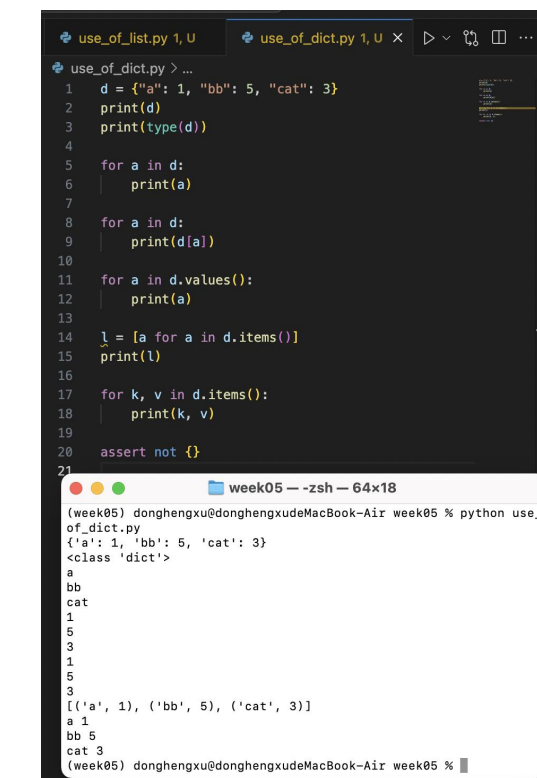
关于列表对象 list 的说明



The screenshot shows a terminal window with the following commands and output:

```
(week05) donghengxu@donghengxudeMacBook-Air week05 % python use_of_list.py
[1, 5, 'abc']
1
5
abc
list index out of range
abc
a
[2, 5, 'a', 'b']
['a', 'b', 2, 5]
False
unsupported operand type(s) for -: 'list' and 'list'
[2, 5, 2, 5, 2, 5]
b=[2, 5, 2, 5, 2, 5]
[9, 5]
[2, 5, 2, 5, 2, 5]
b=[[2, 5], [2, 5], [2, 5]]
[9, 5]
[[9, 5], [9, 5], [9, 5]]
[4, 25, 9]
[4, 9]
b=[[2, 5], [2, 5], [2, 5]]
None
[2, 5, 4]
[[2, 5, 4], [2, 5, 4], [2, 5, 4]]
(week05) donghengxu@donghengxudeMacBook-Air week05 %
```

关于字典 dict 对象的说明



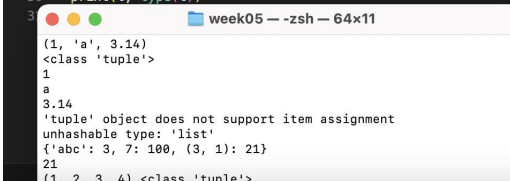
The screenshot shows a code editor with a file named `use_of_dict.py` and a terminal window. The code in the editor defines a dictionary `d` with keys 'a', 'bb', and 'cat'. It prints the dictionary, its type, and iterates over its keys, values, and items. The terminal output shows the dictionary being created and iterated over.

```
1 d = {'a': 1, 'bb': 5, 'cat': 3}
2 print(d)
3 print(type(d))
4
5 for a in d:
6     print(a)
7
8 for a in d:
9     print(d[a])
10
11 for a in d.values():
12     print(a)
13
14 l = [a for a in d.items()]
15 print(l)
16
17 for k, v in d.items():
18     print(k, v)
19
20 assert not {}
21
```

```
(week05) donghengxu@donghengxudeMacBook-Air week05 % python use_of_dict.py
{'a': 1, 'bb': 5, 'cat': 3}
<class 'dict'>
a
bb
cat
1
5
3
1
5
3
[('a', 1), ('bb', 5), ('cat', 3)]
a 1
bb 5
cat 3
(week05) donghengxu@donghengxudeMacBook-Air week05 %
```

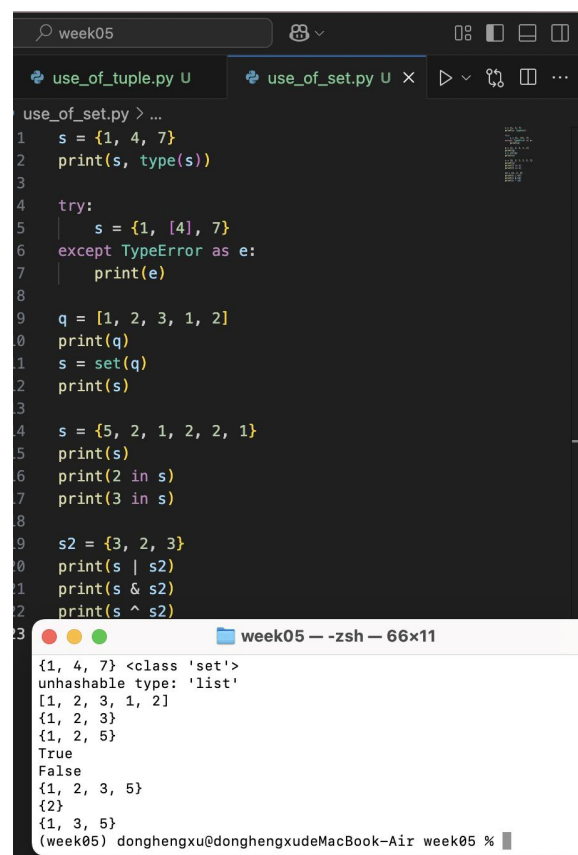
关于元组对象 tuple 的说明

```
1 t = (1, "a", 3.14)
2 print(t)
3 print(type(t))
4
5 print(t[0])
6 print(t[1])
7 print(t[2])
8
9 try:
10     t[0] = 9
11 except TypeError as e:
12     print(e)
13
14 d = {}
15 d["abc"] = 3
16 d[7] = 100
17 q = [3, 1]
18
19 try:
20     d[q] = 21
21 except TypeError as e:
22     print(e)
23
24 t = (3, 1)
25 d[t] = 21
26 print(d)
27 print(d[3, 1])
28
29 t = 1, 2, 3, 4
30 print(t, type(t))
31
```



```
{1, 'a', 3.14}
<class 'tuple'>
1
a
3.14
'tuple' object does not support item assignment
unhashable type: 'list'
{'abc': 3, 7: 100, (3, 1): 21}
21
(1, 2, 3, 4) <class 'tuple'>
```

关于集合 set 对象的说明



```
1 s = {1, 4, 7}
2 print(s, type(s))
3
4 try:
5     s = {1, [4], 7}
6 except TypeError as e:
7     print(e)
8
9 q = [1, 2, 3, 1, 2]
10 print(q)
11 s = set(q)
12 print(s)
13
14 s = {5, 2, 1, 2, 2, 1}
15 print(s)
16 print(2 in s)
17 print(3 in s)
18
19 s2 = {3, 2, 3}
20 print(s | s2)
21 print(s & s2)
22 print(s ^ s2)
23
```

```
{1, 4, 7} <class 'set'>
unhashable type: 'list'
[1, 2, 3, 1, 2]
{1, 2, 3}
{1, 2, 5}
True
False
{1, 2, 3, 5}
{2}
{1, 3, 5}
(week05) donghengxu@donghengxudeMacBook-Air week05 %
```


Python 标准库 (standard library) 里的 `pathlib` 和 `datetime` 模块 (module) 提供了用于处理 路径 和 日期时间 的类型

```
use_of_path.py > ...
1 from pathlib import Path
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("./data1")
11 print(p.exists())
12 p.mkdir(exist_ok=True)
13 print(p.exists())
14 print(p.is_dir())
15
16 p = Path(".")
17 p2 = p / "README.md"
18 print(p2)
19 p3 = p2.absolute()
20 print(p3)
21
```

```
week05 -- -zsh -- 108x26
(week05) donghengxu@donghengxudeMacBook-Air week05 % python use_of_path.py
.
True
/Users/donghengxu/repo/week05
[PosixPath('use_of_float.py'),
PosixPath('use_of_bool.py'),
PosixPath('LICENSE'),
PosixPath('environment.yml'),
PosixPath('use_of_tuple.py'),
PosixPath('data1'),
PosixPath('use_of_str.py'),
PosixPath('use_of_list.py'),
PosixPath('README.md'),
PosixPath('use_of_bytes.py'),
PosixPath('use_of_dict.py'),
PosixPath('use_of_set.py'),
PosixPath('.gitignore'),
PosixPath('use_of_int.py'),
PosixPath('use_of_path.py'),
PosixPath('use_of_eg.py'),
PosixPath('.git')]
True
True
True
README.md
/Users/donghengxu/repo/week05/README.md
```

```
use_of_datetime.py > ...
1 from datetime import date, datetime, timedelta # noqa:F401
2
3 t1 = date.today()
4 t2 = date(2025, 11, 11)
5 td = t2 - t1
6
7 print(td, type(td))
8 print(td.days)
9
10 s1 = "2024-5-23"
11 s2 = "2024-12-04"
12 d1 = datetime.strptime(s1, "%Y-%m-%d")
13 d2 = datetime.strptime(s2, "%Y-%m-%d")
14 print(d1)
15 print(d2)
16
```

```
week05 -- -zsh -- 79x6
(week05) donghengxu@donghengxudeMacBook-Air week05 % python use_of_datetime.py
212 days, 0:00:00 <class 'datetime.timedelta'>
212
2024-05-23 00:00:00
2024-12-04 00:00:00
(week05) donghengxu@donghengxudeMacBook-Air week05 %
```