

第五周学习报告

- 1.仓库 Clone 到本地计算机

```
MINGW64/c/Users/hp/repo  x  +  v
-rw-r--r-- 1 hp 197121      0  6月 30  2021  untitled.txt
drwxr-xr-x 1 hp 197121      0  4月  1  18:57  'WPS Cloud Files'/
drwxr-xr-x 1 hp 197121      0  3月  3  17:46  WPSDrive/
(base)
hp@LAPTOP-L5E04S06 MINGW64 ~
$ cd repo
(base)
hp@LAPTOP-L5E04S06 MINGW64 ~/repo
$ git clone git@gitcode.com:crazy77/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)
hp@LAPTOP-L5E04S06 MINGW64 ~/repo
$ ls -l
total 16
drwxr-xr-x 1 hp 197121 0   3月 20 20:19 ccprj/
drwxr-xr-x 1 hp 197121 0   3月 14 11:52 mywork/
drwxr-xr-x 1 hp 197121 0   3月 19 22:14 prj1/
drwxr-xr-x 1 hp 197121 0   3月  7 21:41 week01/
drwxr-xr-x 1 hp 197121 0   3月 14 12:25 week02/
drwxr-xr-x 1 hp 197121 0   3月 20 20:50 week03/
drwxr-xr-x 1 hp 197121 0   3月 28 18:06 week04/
drwxr-xr-x 1 hp 197121 0   4月  6 19:14 week05/
(base)
hp@LAPTOP-L5E04S06 MINGW64 ~/repo
$
```

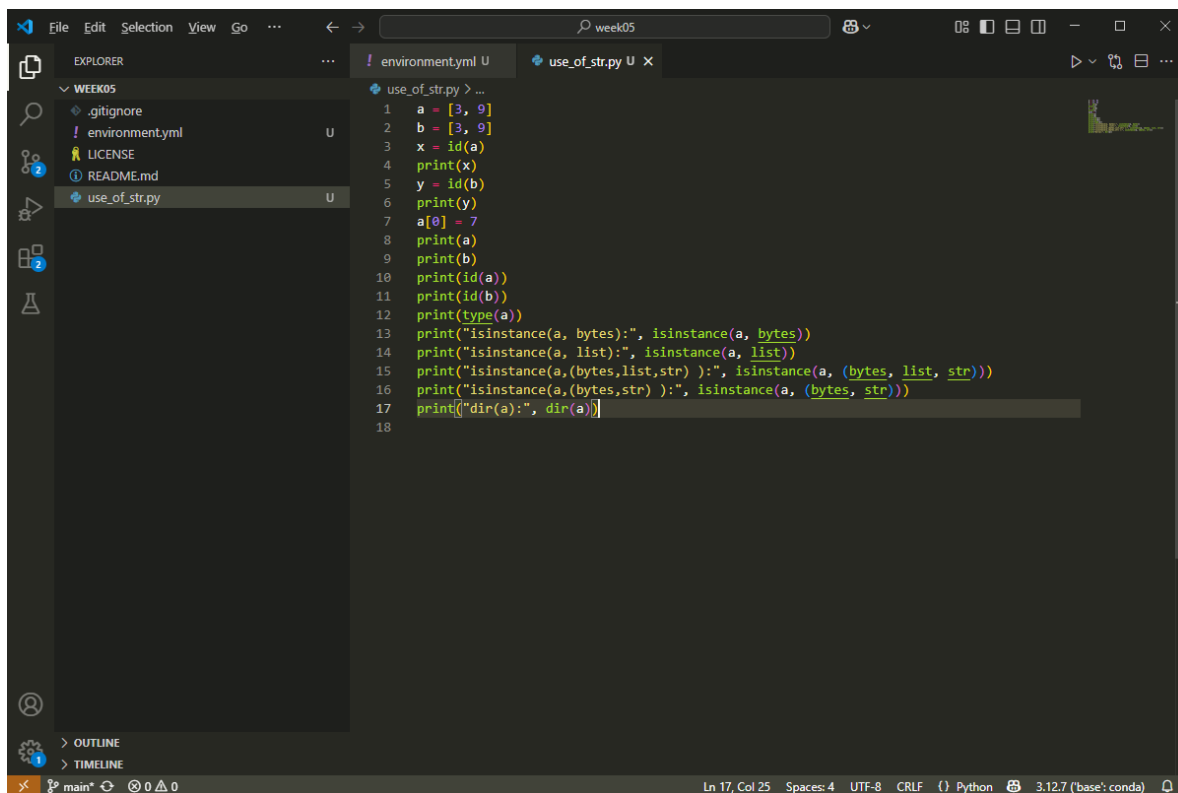
- 2.新建一个 environment.yml 文件

```
File Edit Selection View Go ...  week05
EXPLORER
WEEK05
  .gitignore
  ! environment.yml U
  LICENSE
  README.md
! environment.yml x
! environment.yml
1 name: week05
2 channels:
3   - conda-forge
4 dependencies:
5   - python=3.12
6   - wat-inspector
7
```

```
MINGW64/c/Users/hp/repo/v x + v
(base)
hp@LAPTOP-L5E04S06 MINGW64 ~/repo
$ cat week04/environment.yml
name: week04
channels:
  - conda-forge
dependencies:
  - python=3.12
  - wat-inspector
(base)
hp@LAPTOP-L5E04S06 MINGW64 ~/repo
$ cp week04/environment.yml week05/
done
#
# To activate this environment, use
#
#     $ conda activate week05
#
# To deactivate an active environment, use
#
#     $ conda deactivate
#
(base)
hp@LAPTOP-L5E04S06 MINGW64 ~/repo/week05 (main)
$
```

• 3.创建 use_of_str.py 文件，验证概念

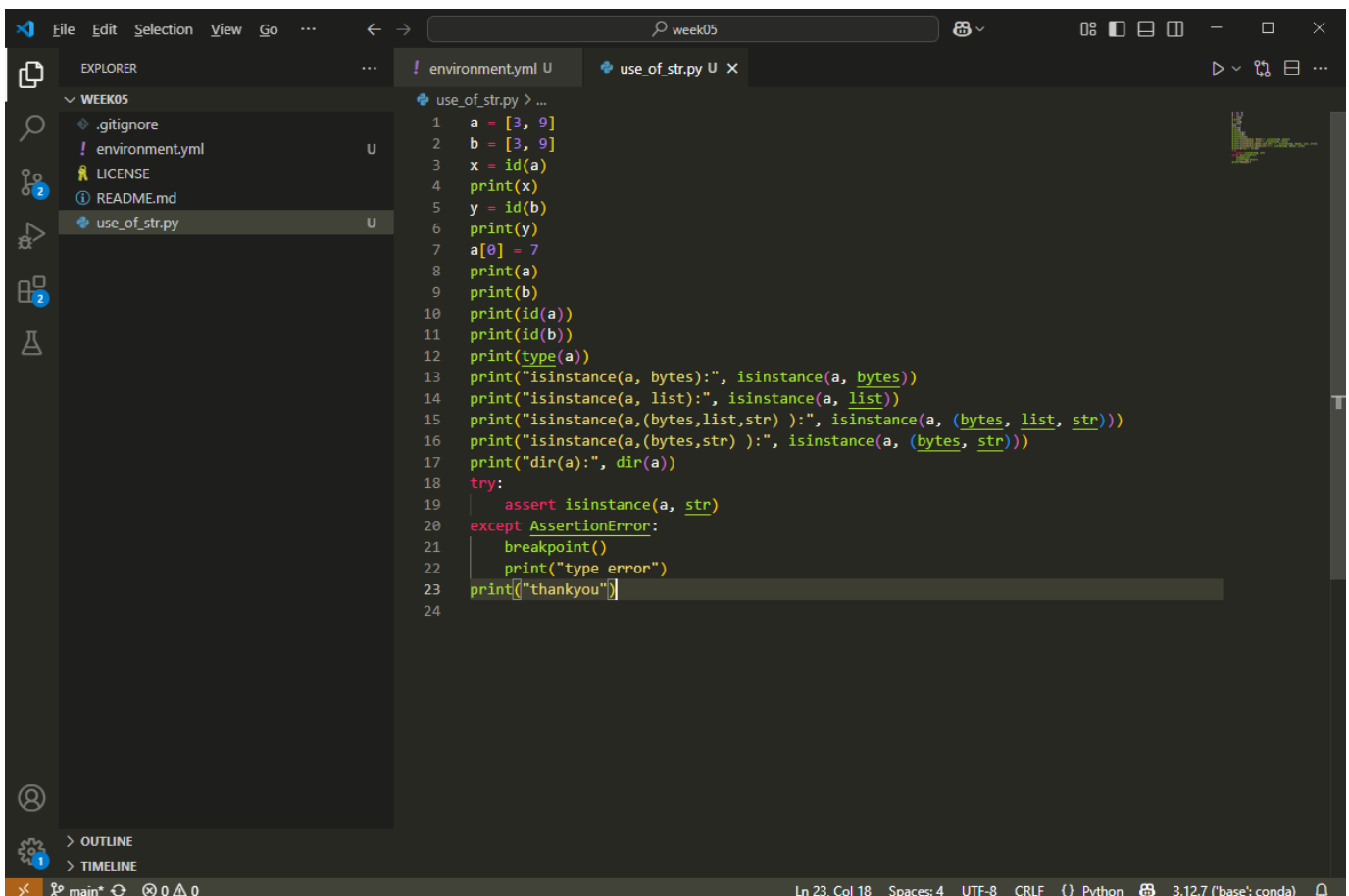
- ❖ id() -- 返回对象在虚拟内存中的地址 (正整数)
- ❖ type() -- 返回对象的类型
- ❖ isinstance() -- 判断对象是否属于某个 (或某些) 类型
- ❖ dir() -- 返回对象所支持的属性 (attributes) 的名称列表
- ❖ str() -- 返回对象 print 时要显示在终端的字符串



```
File Edit Selection View Go ... week05
EXPLORER
WEEK05
  .gitignore
  ! environment.yml
  LICENSE
  README.md
  use_of_str.py
! environment.yml U
use_of_str.py U
use_of_str.py > ...
1 a = [3, 9]
2 b = [3, 9]
3 x = id(a)
4 print(x)
5 y = id(b)
6 print(y)
7 a[0] = 7
8 print(a)
9 print(b)
10 print(id(a))
11 print(id(b))
12 print(type(a))
13 print("isinstance(a, bytes):", isinstance(a, bytes))
14 print("isinstance(a, list):", isinstance(a, list))
15 print("isinstance(a, (bytes, list, str)):", isinstance(a, (bytes, list, str)))
16 print("isinstance(a, (bytes, str)):", isinstance(a, (bytes, str)))
17 print("dir(a):", dir(a))
18
```

```
MINGW64/c/Users/hp/repo/v x + v
isinstance(a,(bytes,list,str)): True
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)
hp@LAPTOP-L5E04S06 MINGW64 ~/repo/week05(main)
$ python use_of_str.py
2231942584576
2231942582592
[7, 9]
[3, 9]
2231942584576
2231942582592
<class 'list'>
isinstance(a, bytes): False
isinstance(a, list): True
isinstance(a,(bytes,list,str)): True
isinstance(a,(bytes,str)): 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)
hp@LAPTOP-L5E04S06 MINGW64 ~/repo/week05 (main)
$
```

- ❖ print() 函数将表达式 (expression) 输出到终端，查看结果是否符合预期
- ❖ assert 语句查验某个表达式 (expression) 为真，否则报错(AssertionError) 退出
- ❖ try 语句拦截报错，避免退出，将流程 (flow) 转入 except 语句
- ❖ breakpoint() 函数暂停程序运行，进入 pdb 调试 (debug) 模式



```
File Edit Selection View Go ... week05
EXPLORER
WEEK05
  .gitignore
  ! environment.yml U
  LICENSE
  README.md
  use_of_str.py U
use_of_str.py > ...
1 a = [3, 9]
2 b = [3, 9]
3 x = id(a)
4 print(x)
5 y = id(b)
6 print(y)
7 a[0] = 7
8 print(a)
9 print(b)
10 print(id(a))
11 print(id(b))
12 print(type(a))
13 print("isinstance(a, bytes):", isinstance(a, bytes))
14 print("isinstance(a, list):", isinstance(a, list))
15 print("isinstance(a,(bytes,list,str)):", isinstance(a, (bytes, list, str)))
16 print("isinstance(a,(bytes,str)):", isinstance(a, (bytes, str)))
17 print("dir(a):", dir(a))
18 try:
19     assert isinstance(a, str)
20 except AssertionError:
21     breakpoint()
22     print("type error")
23 print("thankyou")
24
```



```
22 print(a)
23
24 a = str()
25 print(a)
26 b = str([2, 3, 4])
27 print(id(a))
28 print(id(b))
29
30 assert str([3, 4, 5]) == "[3, 4, 5]"
31 assert str(1 + 2) == "3"
32 assert str(2.1 + 2.1) == "4.2"
33 print(2.1 + 2.1)
34 assert str(1.1 + 2.1) == "3.2"
35
36
37 print("运算值")
38 a = "+"
39 x = id(a)
40 b = a * 10
41 y = id(b)
42 print(b)
43 assert x != y
44
45 print("索引值")
46 s = "wonderful"
47 assert s[4] == "e"
48 assert s[-1] == "l"
49 assert s[4] == "wond"
50 assert s[8] == s[-1]
51
52 print("返回值")
53 a = "goodbye"
54 b = a.upper()
55 print(a)
56 print(b)
57
```

● 5.验证字符串属性

```
1 a1 = "asd"
2 a2 = "fgh"
3 a = a1 + a2
4 assert a == "asdfgh"
5 print(a2 + a1)
6
7 try:
8     print(a1 - a2)
9 except TypeError as e:
10    print(e)
11
12 a = "+++"
13 s = a * 5
14 print(s)
15
16 a = "asd"
17 b = "fgh"
18 try:
19     s = a * b
20 except TypeError as e:
21     print(e)
22
23 s = "dfh"
24 try:
25     s = s / 6
26 except TypeError as e:
27     print(e)
28 assert s == "dfh"
29
30 print("dfg" > "DFG")
31 print("345" > "asd")
32 print("6" > ".")
33 print("god" > "good")
34 print("bus" > "[")
35
```

```
MINGW64/c/Users/hp/repo/v x + v
(base)
hp@LAPTOP-L5E04S06 MINGW64 ~
$ repo week05
bash: repo: command not found
(base)
hp@LAPTOP-L5E04S06 MINGW64 ~
$ cd repo
(base)
hp@LAPTOP-L5E04S06 MINGW64 ~/repo
$ cd week05
(base)
hp@LAPTOP-L5E04S06 MINGW64 ~/repo/week05 (main)
$ conda activate week05
(world05)
hp@LAPTOP-L5E04S06 MINGW64 ~/repo/week05 (main)
$ python use_of_attributes_str.py
fgghasd
unsupported operand type(s) for -: 'str' and 'str'
+++++
can't multiply sequence by non-int of type 'str'
unsupported operand type(s) for /: 'str' and 'int'
True
False
True
False
True
(world05)
hp@LAPTOP-L5E04S06 MINGW64 ~/repo/week05 (main)
$
```

- ❖ 字符串支持比较运算符、字符串不支持减法、除法运算
- ❖ ASCII字符编码大小关系总结：
控制字符的编码值小于可打印字符。
在可打印字符中，空格 < 数字 < 大写字母 < 小写字母。
同一类连续字符（如数字、大写字母、小写字母）内，编码值按字符顺序递增。
- ❖ 字符串的字典序：
对于字符串来说，字典序是按照字母表顺序来比较的。具体规则如下：
从两个字符串的第一个字符开始比较，如果对应位置字符不同，则字符编码值小的字符所在的字符串排在前面。
如果对应位置字符相同，则继续比较下一个位置的字符，直到出现不同字符或者其中一个字符串结束。
如果一个字符串是另一个字符串的前缀，那么较短的字符串排在前面。

```
35
36 assert "apple"
37 assert ""
38
True
(world05)
hp@LAPTOP-L5E04S06 MINGW64 ~/repo/week05 (main)
$ python use_of_attributes_str.py
fgghasd
unsupported operand type(s) for -: 'str' and 'str'
+++++
can't multiply sequence by non-int of type 'str'
unsupported operand type(s) for /: 'str' and 'int'
True
False
True
False
True
Traceback (most recent call last):
  File "C:\Users\hp\repo\week05\use_of_attributes_str.py", line 37, in <module>
    assert ""
    ^^
AssertionError
(world05)
```

❖ 字符串长度为0，会被当做False。

```
38
39 a = "apple"
40 print(iter(a))
```

```
hp@LAPTOP-L5E04S06 MINGW64 ~/repo/week05 (main)
$ python use_of_attributes_str.py
fghasd
unsupported operand type(s) for -: 'str' and 'str'
+++++
can't multiply sequence by non-int of type 'str'
unsupported operand type(s) for /: 'str' and 'int'
True
False
True
False
True
<str_ascii_iterator object at 0x000001C79415C280>
```

显示此则为可迭代。

```
print(len(a))
```

```
MINGW64/c/Users/hp/repo/v x + v
True
<str_ascii_iterator object at 0x00000153EFC3C310>
a
p
p
l
e
e
(week05)
hp@LAPTOP-L5E04S06 MINGW64 ~/repo/week05 (main)
$ python use_of_attributes_str.py
fghasd
unsupported operand type(s) for -: 'str' and 'str'
+++++
can't multiply sequence by non-int of type 'str'
unsupported operand type(s) for /: 'str' and 'int'
True
False
True
False
True
False
True
<str_ascii_iterator object at 0x000001C9CDF4C370>
a
p
p
l
e
5
(week05)
hp@LAPTOP-L5E04S06 MINGW64 ~/repo/week05 (main)
$
```

显示字符串长度为5

```
a = "apple"
assert a[2:4] == "pl"
```

字符串支持索引操作（：前包含：后不包含）

查看字符串支持的方法：

```
MINGW64/c:/Users/hp/repo/s x + v
value: 'apple'
type: str
len: 5

Public attributes:
def capitalize() # Return a capitalized version of the string...
def casefold() # Return a version of the string suitable for caseless comparisons...
def center(width, fillchar=' ') # Return a centered string of length width...
def count(sub[, start[, end]]) # Return the number of non-overlapping occurrences of substring sub in string s...
def encode(encoding='utf-8', errors='strict') # Encode the string using the codec registered for encoding...
def endsuffix(suffix[, start[, end]]) # Return True if the string ends with the given suffix...
def expandtabs(tabsize=8) # Return a copy where all tab characters are expanded using spaces...
def find(sub[, start[, end]]) # Return the lowest index in s where substring sub is found...
def format(*args, **kwargs) # Return a formatted string using the given arguments...
def format_map(mapping) # Return a formatted string using the given mapping...
def index(sub[, start[, end]]) # Return the lowest index in s where substring sub is found...
def isalnum() # Return True if the string is an alphanumeric string, False otherwise...
def isalpha() # Return True if the string is an alphabetic string, False otherwise...
def isascii() # Return True if all characters in the string are ASCII, False otherwise...
def isdecimal() # Return True if the string is a decimal string, False otherwise...
def isdigit() # Return True if the string is a digit string, False otherwise...
def isidentifier() # Return True if the string is a valid Python identifier, False otherwise...
def islower() # Return True if the string is a lowercase string, False otherwise...
def isnumeric() # Return True if the string is a numeric string, False otherwise...
def isprintable() # Return True if the string is printable, False otherwise...
def isspace() # Return True if the string is a whitespace string, False otherwise...
def istitle() # Return True if the string is a title-cased string, False otherwise...
def isupper() # Return True if the string is an uppercase string, False otherwise...
def join(iterable) # Concatenate any number of strings...
def ljust(width, fillchar=' ') # Return a left-justified string of length width...
def lower() # Return a copy of the string converted to lowercase...
def lstrip(chars=None) # Return a copy of the string with leading whitespace removed...
def maketrans(table) # Return a translation table usable for str.translate()...
def partition(sep) # Partition the string into three parts using the given separator...
def removeprefix(prefix) # Return a str with the given prefix string removed if present...
def removesuffix(suffix) # Return a str with the given suffix string removed if present...
def replace(old, new, count=-1) # Return a copy with all occurrences of substring old replaced by new...
def rfind(sub[, start[, end]]) # Return the highest index in s where substring sub is found...
def rindex(sub[, start[, end]]) # Return the highest index in s where substring sub is found...
def rjust(width, fillchar=' ') # Return a right-justified string of length width...
def rpartition(sep) # Partition the string into three parts using the given separator...
def rsplit(sep=None, maxsplit=-1) # Return a list of the substrings in the string, using sep as the separator string...
defrstrip(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[, end]]) # Return True if the string starts with the given prefix...
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...
```

```
s = "i miss you"
print(s.capitalize())
print(s)
```

```
I miss you
i miss you
```

❖ capitalize: 将第一个字母大写

```
53
54 print("789hui".isalnum())
55 print("789_hui".isalnum())
56
```

- ❖ isalnum: 主要作用是检查字符串中的所有字符是否都是字母或数字（即字母 a-z、A-Z 或者数字 0-9），并且字符串至少要有有一个字符。如果满足这些条件，isalnum() 方法会返回 True；反之，则返回 False。
- ❖ isidentifier: 它会对调用该方法的字符串进行检查，若该字符串是有效的 Python 标识符，则返回 True；反之，返回 False。
- ❖ join: 其作用是将可迭代对象（像列表、元组、集合等）中的元素连接成一个字符串。


```

numbers = ("1", "2", "3", "4")
result = ", ".join(numbers)
print(result)

```

```

(MINGW64/c/Users/hp/repo/v)
hp@LAPTOP-L5E04S06 MINGW64 ~/repo/week05 (main)
$ python use_of_attributes_str.py
fghasd
unsupported operand type(s) for -: 'str' and 'str'
+++++
can't multiply sequence by non-int of type 'str'
unsupported operand type(s) for /: 'str' and 'int'
True
False
True
False
True
<str_ascii_iterator object at 0x000001A885A3CB50>
a
p
p
l
e
5
I miss you
i miss you
True
False
True
False
1, 2, 3, 4
(MINGW64/c/Users/hp/repo/v)
hp@LAPTOP-L5E04S06 MINGW64 ~/repo/week05 (main)
$

```

6. 字节串 (bytes)

```

use_of_bytes.py > ...
1  from pathlib import Path
2
3  s = b"hello"
4  print(s)
5  print(s[0])
6
7  p = Path("D:\\ANACONDA\\envs\\week05\\python.exe")
8  breakpoint()

```

```

(MINGW64/c/Users/hp/repo/v)
hp@LAPTOP-L5E04S06 MINGW64 ~/repo/week05 (main)
$ python use_of_bytes.py
b'hello'
104
--Return--
> c:\users\hp\repo\week05\use_of_bytes.py(8)<module>()->None
-> breakpoint()
(Pdb) p p
WindowsPath('D:/ANACONDA/envs/week05/python.exe')
(Pdb) p p.exists()
True
(Pdb) p p.is_file()
*** SyntaxError: invalid syntax
(Pdb) p p.is_file()
True
(Pdb) p p.is_dir()
False
(Pdb)

```

```
p = Path("D:\\ANACONDA\\envs\\week05\\python.exe")
s = p.read_bytes()
print(len(s))
breakpoint()
```

```
hp@LAPTOP-L5E04S06 MINGW64 ~/repo/week05 (main)
$ python use_of_bytes.py
b'hello'
104
93184
--Return--
> c:\users\hp\repo\week05\use_of_bytes.py(10)<module>()->None
-> breakpoint()
(Pdb)
```

查看字节串的字节数

```
s = b.decode()
assert isinstance(s, str)
b2 = s.encode()
assert isinstance(b2, bytes)
```

字符串编码后为字节串，字节串解码后为字符串

```
22 s = "再见"
23 b1 = s.encode("utf-8")
24 print(b1)
25 b2 = s.encode("gbk")
26 print(b2)
27 breakpoint()
28
```

```
b'\xe5\x86\x8d\xe8\xa7\x81'
b'\xd4\xd9\xbc\xfb'
```

由于 utf - 8 和 gbk 采用了不同的编码规则和字节表示方式，所以对相同的字符串进行编码后，得到的字节序列是不同的。

7. 整数 (int)

```
use_of_int.py > ...
1  x = 89
2  y = 9
3  z = x * y
4  print(z)
5
6  x = 8
7  y = 20
8  assert y // x == 2
9  assert y % x == 4
10
11 assert 90
12 try:
13     assert 0
14 except AssertionError as e:
15     print(type(e))
16
```

0会被当做False

```
(week05)
hp@LAPTOP-L5E04S06 MINGW64 ~/repo/week05 (main)
$ python use_of_int.py
801
<class 'AssertionError'>
```

```
(week05)
hp@LAPTOP-L5E04S06 MINGW64 ~/repo/week05 (main)
$ python use_of_int.py
801
<class 'AssertionError'>
--Return--
> c:\users\hp\repo\week05\use_of_int.py(17)<module>()->None
-> breakpoint()
(Pdb) l
12     try:
13         assert 0
14     except AssertionError as e:
15         print(type(e))
16
17 -> breakpoint()
[EOF]
(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 iter(x)
*** TypeError: 'int' object is not iterable
(Pdb)
```

❖ 整数不可迭代

```
(Pdb) p len(x)
*** TypeError: object of type 'int' has no len()
(Pdb)
```

❖ 整数不能求长度

```
*** TypeError: object of type 'int' has no len()
(Pdb) p x[0]
*** TypeError: 'int' object is not subscriptable
(Pdb)
```

❖ 整数不支持索引操作

❖ 查看整数支持的方法：

```
(Pdb) import wat
(Pdb) wat /x

value: 8
type: int

Public attributes:
  denominator: int = 1
  imag: int = 0
  numerator: int = 8
  real: int = 8

def as_integer_ratio() # Return a pair of integers, whose ratio is equal to the original int...
def bit_count() # Number of ones in the binary representation of the absolute value of self...
def bit_length() # Number of bits necessary to represent self in binary...
def conjugate(...) # Returns self, the complex conjugate of any int.
def from_bytes(bytes, byteorder='big', *, signed=False) # Return the integer represented by the given array of bytes...
def is_integer() # Returns True. Exists for duck type compatibility with float.is_integer.
def to_bytes(length=1, byteorder='big', *, signed=False) # Return an array of bytes representing an integer...
```

8.浮点数 (float)

```
use_of_float.py > ...
1  import random
2
3  print("字面值")
4  x = 5.78
5  print(type(x))
6
7  print("初始化")
8  y = float("5.78")
9  print(type(y))
10
11 assert x == y
12
13 x = 6 / 7
14 print(x, type(x))
15
16 x = random.random()
17 print(x)
18
```

```

(week05)
hp@LAPTOP-L5E04S06 MINGW64 ~/repo/week05 (main)
$ python use_of_float.py
字面值
<class 'float'>
初始化
<class 'float'>
0.8571428571428571 <class 'float'>
0.3991457113453797
(week05)
hp@LAPTOP-L5E04S06 MINGW64 ~/repo/week05 (main)
$ python use_of_float.py
字面值
<class 'float'>
初始化
<class 'float'>
0.8571428571428571 <class 'float'>
0.8089006161978205

```

函数的返回值可以是浮点数

```

18
19 assert 0.0
20

```

0.0会被当做False

```

20
21 s = float("nan")
22 print(s + 10)
23 print(s > 10)
24 print(s < 10)
25 print(s == 10)
26

```

```

nan
False
False
False

```

缺失值运算后还是缺失值

```

a = float("inf")
print(3.14e2)
print(a > 10e100)
print(a > a)
print(a == a)

b = float("-inf")
print(b)
print(b > b)
print(b == b)

```

```

314.0
True
False
True
-inf
False
True
(week05)

```

9.布尔值 (bool)

不能循环、没有长度、不支持索引操作

```
use_of_bool.py > ...
1  print("字面值")
2  t = True
3  f = False
4  print(t, f)
5
```

```
(week05)
hp@LAPTOP-L5E04S06 MINGW64 ~/repo/week05 (main)
$ python use_of_bool.py
字面值
True False
```

布尔是特殊的整数

```
5
6  print(type(t))
7  print(isinstance(t, int))
8
True False
<class 'bool'>
True
```

10.列表 (list)

```
use_of_list.py > ...
1  print("字面值")
2  a = [2, 5, "ghjk"]
3  print(a)
4
5  print(a[0])
6  print(a[1])
7  print(a[2])
8  print(a[-1])
9  print(a[-1][2])
10
11 try:
12     print(a[3])
13 except IndexError as e:
14     print(e)
15
```

列表支持索引操作、可以比较大小

```

16 a = [2, 5]
17 b = [5, 2]
18 print(a + b)
19 print(b + a)
20 print(b + a == a + b)
21
22 a = [8, 9]
23 b = [9]
24 try:
25     print(a - b)
26 except TypeError as e:
27     print(e)
28
29 a = [3, 8]
30 print(a * 4)

```

```

[2, 5, 5, 2]
[5, 2, 2, 5]
False
unsupported operand type(s) for -: 'list' and 'list'
[3, 8, 3, 8, 3, 8, 3, 8]

```

列表支持加法、乘法，不支持减法

11.字典 (dict)

```

use_of_dict.py > ...
1  print("字典值")
2  a = {"f": 7, "g": 8, "j": 9}
3  print(a)
4  print(type(a))
5
6  for m in a:
7      print(m)
8
9  for m in a:
10     print(a[m])
11
12 for m in a.values():
13     print(a)
14
15 l = [m for m in a.items()]
16 print(l)
17
18 for k, v in a.items():
19     print(k, v)
20

```

```

(week05)
hp@LAPTOP-L5E04S06 MINGW64 ~/repo/week05 (main)
$ python use_of_dict.py
字典值
{'f': 7, 'g': 8, 'j': 9}
<class 'dict'>
f
g
j
7
8
9
{'f': 7, 'g': 8, 'j': 9}
{'f': 7, 'g': 8, 'j': 9}
{'f': 7, 'g': 8, 'j': 9}
[('f', 7), ('g', 8), ('j', 9)]
f 7
g 8
j 9

```

```
20
```

```
21 assert {}
```

```
22
```

空字典会被当做False
字典里的键是不可变的对象

12.元组 (tuple)

```
use_of_tuple.py > ...
1 print("字面值")
2 a = (3, "asd", 5.77)
3 print(type(a))
4 print(a)
5
6 print(a[0])
7 print(a[1])
8 print(a[2])
9
```

```
(week05)
hp@LAPTOP-L5E04S06 MINGW64 ~/repo/week05 (main)
$ python use_of_tuple.py
字面值
<class 'tuple'>
(3, 'asd', 5.77)
3
asd
5.77
```

- ❖ 列表和元组的区别：
元组是不可变的对象，不支持赋值
- ❖ 元组可以作为键
- ❖ 不会造成语法歧义时，圆括号可以省掉

```
try:
    a[1] = 9
except TypeError as e:
    print(e)

d = {}
d["abc"] = 9
d[8] = 200
q = [2, 5]

try:
    d[q] = 32
except TypeError as e:
    print(e)

t = (5, 8)
d[t] = 32
print(d)
print(d[5, 8])
```



```
'tuple' object does not support item assignment
unhashable type: 'list'
{'abc': 9, 8: 200, (5, 8): 32}
32
```

13.集合 (set)

```
use_of_set.py > ...
1  print("字面值")
2  s = {1, 9, 0}
3  print(s)
4  print(type(s))
5
6  try:
7      s = {1, [9], 0}
8  except TypeError as e:
9      print(e)
10
11 print("初始化")
12 a = [2, 3, 4, 4, 3, 5]
13 print(a)
14 s = set(a)
15 print(s)
16
17 s = {2, 3, 4, 4, 3, 5}
18 print(s)
19 print(2 in s)
20 print(9 in s)
21
22 s2 = {1, 2, 3}
23 print(s | s2)
24 print(s & s2)
25 print(s ^ s2)
26
```

```
(week05)
hp@LAPTOP-L5E04S06 MINGW64 ~/repo/week05 (main)
$ python use_of_set.py
字面值
{0, 1, 9}
<class 'set'>
unhashable type: 'list'
初始化
[2, 3, 4, 4, 3, 5]
{2, 3, 4, 5}
{2, 3, 4, 5}
True
False
{1, 2, 3, 4, 5}
{2, 3}
{1, 4, 5}
(week05)
```

- ❖ 有并集、交集、对称差
- ❖ 没有补集