第五周作业

Python 对象类型

一、常见检查所用的工具和命令

1、将 week04 的 environment.yml 复制到 week05，并创建环境

```
cm36 — -zsh — 80×24
(base) sjy@sunjiayideMacBook-Pro cm36 % cat week04/environment.yml
name: week04
channels:
  - conda-forge
dependencies:
  - python=3.12
  - wat-inspector
%
(base) sjy@sunjiayideMacBook-Pro cm36 % cp week04/environment.yml week05/
(base) sjy@sunjiayideMacBook-Pro cm36 % ls -l
total 8
-rw-r--r--@  1 sjy  staff  546  3  6 23:24 11
drwxr-xr-x   5 sjy  staff  160  3 13 19:02 mywork
drwxr-xr-x   9 sjy  staff  288  3  9 00:58 week01
drwxr-xr-x   8 sjy  staff  256  3 19 11:48 week02
drwxr-xr-x   7 sjy  staff  224  3 19 11:49 week03
drwxr-xr-x  12 sjy  staff  384  3 27 22:44 week04
drwxr-xr-x   7 sjy  staff  224  4  6 18:19 week05
(base) sjy@sunjiayideMacBook-Pro cm36 % ls -l week05
total 56
-rw-r--r--  1 sjy  staff  18411  4  6 18:17 LICENSE
-rw-r--r--  1 sjy  staff   2216  4  6 18:17 README.md
-rw-r--r--@ 1 sjy  staff     89  4  6 18:19 environment.yml
(base) sjy@sunjiayideMacBook-Pro cm36 %
```

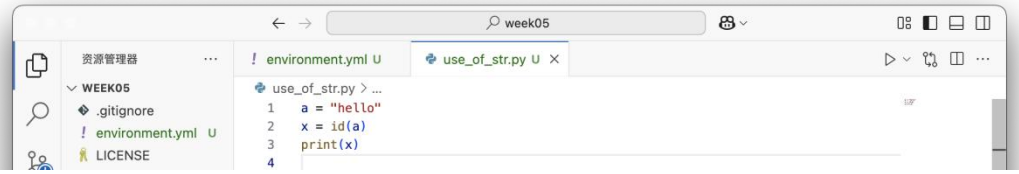
2、用 print()函数将表达式输出到终端

```
week05
资源管理器
WEEK05
.gitignore
environment.yml U
LICENSE
environment.yml U
use_of_str.py U
use_of_str.py > ...
1  a = "hello"
2  print(a)
3
```

```
(base) sjy@sunjiayideMacBook-Pro week05 % conda activate week05
(week05) sjy@sunjiayideMacBook-Pro week05 % python use_of_str.py
(week05) sjy@sunjiayideMacBook-Pro week05 % python use_of_str.py
hello
```

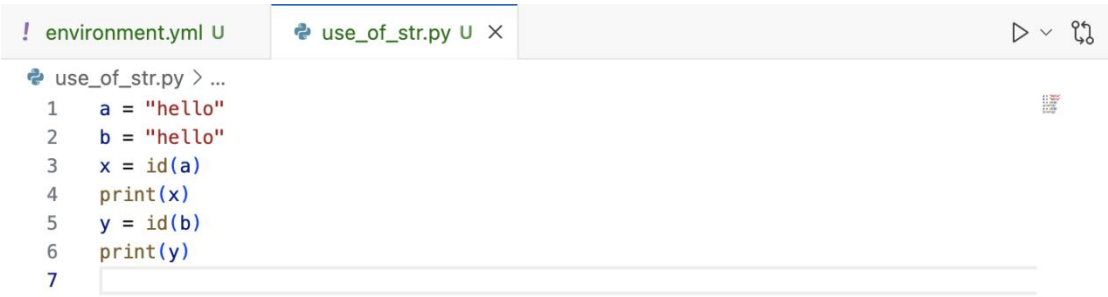
3、id()：显示对象在虚拟内存中的地址

（1）



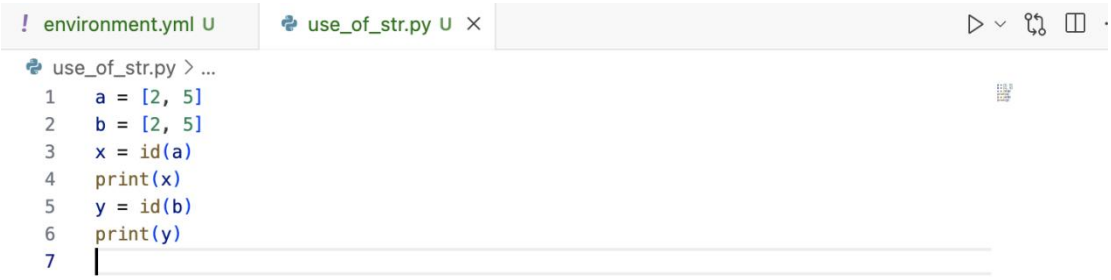

```
(week05) sjy@sunjiayideMacBook-Pro week05 % python use_of_str.py
4313849152
(week05) sjy@sunjiayideMacBook-Pro week05 % python use_of_str.py
4401781840
(week05) sjy@sunjiayideMacBook-Pro week05 % python use_of_str.py
4359249216
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

（2）两者概念一样时，输出的是一样的




```
(week05) sjy@sunjiayideMacBook-Pro week05 % python use_of_str.py
4330740816
4330740816
(week05) sjy@sunjiayideMacBook-Pro week05 % python use_of_str.py
4388953360
4388953360
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

（3）概念不同时，输出结果不同

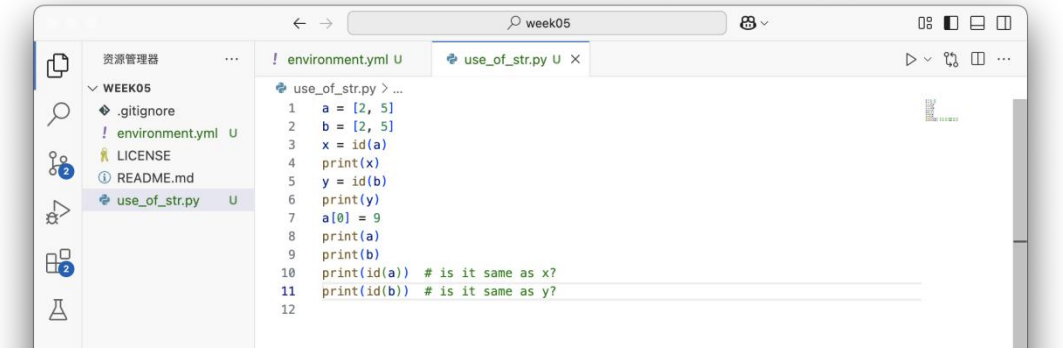



```
(week05) sjy@sunjiayideMacBook-Pro week05 % python use_of_str.py
4495378752
4495380608
(week05) sjy@sunjiayideMacBook-Pro week05 % python use_of_str.py
4406905152
4406907008
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

(4)


```
(week05) sjy@sunjiayideMacBook-Pro week05 % python use_of_str.py
4515416384
4515418240
[9, 5]
[2, 5]
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

(5)




```
(week05) sjy@sunjiayideMacBook-Pro week05 % python use_of_str.py
4559358272
4559360128
[9, 5]
[2, 5]
4559358272
4559360128
```

4、type()--返回对象的类型

```
print(type(a))
```

```
<class 'list'>
```

5、isinstance()--判断对象是否属于某个/某些类型（只要有就行）

```
print(isinstance(a, str))
```

```
False
```

```
print(isinstance(a, (str, list)))
```

```
True
```

6、dir()--返回对象所支持的属性(attributes)的名称列表

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

dir(a): ['__add__', '__class__', '__class_getitem__', '__contains__', '__delattr
__', '__delitem__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__g
etattribute__', '__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) sjy@sunjiayideMacBook-Pro week05 %
```

7、()--返回对象 print 时要显示在终端的字符串

8、可用 assert 查验某个表达式为真假，否则报错退出

（1）错误就会报错并退出，后面的程序就不走了

```
assert isinstance(a, str)

AssertionError
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

（2）正确则不显示，会继续往下走

```
assert isinstance(a, list)
print("good")

dir(a): ['__add__', '__class__', '__class_getitem__', '__contains__', '__delattr
__', '__delitem__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__g
etattribute__', '__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']
good
```

9、可用 try 语句拦截报错，避免退出，将流程转入 except 语句

```
try:
    assert isinstance(a, str)
except AssertionError:
    print('type error')
```

```
(week05) sjy@sunjiayideMacBook-Pro week05 % python use_of_str.py
4486138176
4486140032
[9, 5]
[2, 5]
4486138176
4486140032
<class 'list'>
False
True
dir(a): ['__add__', '__class__', '__class_getitem__', '__contains__', '__delattr
__', '__delitem__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__g
etattribute__', '__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']
type error
good
```

10、调用 breakpoint()函数暂停程序运行，进入 pdb 调试模式

```python
try:
    assert isinstance(a, str)
except AssertionError:
    breakpoint()
    print("type error")

print("good")
```

```
(Pdb) l .
 15     print("dir(a):", dir(a))
 16     try:
 17         assert isinstance(a, str)
 18     except AssertionError:
 19         breakpoint()
 20  ->     print("type error")
 21
 22     print("good")
[EOF]
(Pdb)
```

```
(Pdb) p a
[9, 5]
(Pdb) p isinstance(a,str)
False
(Pdb)
```

二、得到字符串实例

1、字符串字面值

use_of_str.py > ...

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

```
bdb.BdbQuit
(week05) sjy@sunjiayideMacBook-Pro week05 % python use_of_str.py
字面值
True
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

➢ f-string 语法

```python
print('f-string')
y='baseball'
s=f'name:{y}'
print(s)
```

```
(week05) sjy@sunjiayideMacBook-Pro week05 % python use_of_str.py
字面值
True
f-string
name:baseball
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

➢ \b：空三个格再输出下一个；\n：换行输出下一个

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

s = "a111\nb111"
print("22222", s)
```

```
(week05) sjy@sunjiayideMacBook-Pro week05 % python use_of_str.py
字面值
True
f-string
name:baseball
TAB a    b
22222 a111
b111
```

➢ 多行实例：空格和换行都会保留

```python
s = """111222333
aaabbbccc
  444
333333
"""
print(s)
```

```
111222333
aaabbbccc
  444
333333

(week05) sjy@sunjiayideMacBook-Pro week05 %
```

2、初始化——用类型得到字符串

```
print("初始化")
s=str()
print(s)
s=str([1,2,3])
print(s)
```

```
初始化

[1, 2, 3]
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

➢ 初始化和字面值得到的字符串相等吗?

✖ 问题出现：在运行这步时，发现不匹配，通过询问大模型得知 Python 的 str()函数在将列表转换为字符串时，元素之间会有一个空格，所以 str([1,2,3])得到的字符串是'[1 ,2 ,3]'，这和'[1,2,3]'不相等，从而导致 assert 语句的条件为 False，抛出 AssertionError。

```
assert str([1, 2, 3])=='[1,2,3]'
```

```
Traceback (most recent call last):
  File "/Users/sjy/cm36/week05/use_of_str.py", line 31, in <module>
    assert str([1, 2, 3]) == "[1,2,3]"
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^
AssertionError
```

✓ 解决方案：将后面的字符串加上空格即可

```
assert str([1, 2, 3]) == "[1, 2, 3]"
```

```
(week05) sjy@sunjiayideMacBook-Pro week05 % python use_of_str.py
字面值
True
f-string
name:baseball
TAB a    b
22222 a111
b111
111222333
aaabbbccc
  444
333333

初始化

[1, 2, 3]
```

➢ 为什么不相等：因为时小数点后好几位，不是单纯的 3.3

```
assert str(1.1 + 2.2) == "3.3"
```

```
[1, 2, 3]
Traceback (most recent call last):
  File "/Users/sjy/cm36/week05/use_of_str.py", line 32, in <module>
    assert str(1.1 + 2.2) == "3.3"
           ^^^^^^^^^^^^^^^^^^^^^^^
AssertionError
```

```
assert str(1.1 + 2.2) != "3.3"
```

```
(week05) sjy@sunjiayideMacBook-Pro week05 % python use_of_str.py
字面值
True
f-string
name:baseball
TAB a    b
22222 a111
b111
111222333
aaabbbccc
  444
333333

初始化

[1, 2, 3]
```

3、运算值：用运算符算出来的字符串

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

```
运算值
====================
```

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

```
aaabbbccc
  444
333333

初始化

[1, 2, 3]
运算值
====================
```

4、索引值：对某一值索引后产生的新对象

✖ 问题出现：开始认为第二个是 p，运行显示 AssertionError。

```
print("索引值")
s='sport'
assert s[2]=='p'
```

✓ 解决方法：通过不断尝试，发现因此第二个应该是 0。通过大模型解答发现，在 Python 里，字符串的索引是从 0 开始的。

```
[1, 2, 3]
运算值
====================
索引值
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

➢ -从后面开始

```
assert s[-1] == "t"
```

```
====================
索引值
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

```
assert s[4] == s[-1]
```

```
====================
索引值
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

➢ :从头选几个字母

```
assert s[:3] == "spo"
```

```
====================
索引值
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

```
try:
    s[5]
except IndexError as e:
    print(e)
```

```
====================
索引值
string index out of range
```

5、返回值：不修改原字符串，只是返回新值

➢ upper 命令：大写

```
print("返回值")
s = "sport"
x = s.upper()
print(x)
```

```
返回值
SPORT
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

➢ format 命令

```
y = "name:{},age{}"
print(y)
yy = y.format("hahah", 66)
print(yy)
```

```
返回值
SPORT
name:{},age{}
name:hahah,age66
```

三、验证属性

1、对数学运算符(+、-、*、/、//、%、@)有没有支持

+：可以

```
s = "abc"
ss = "222"
sss = s + ss
assert sss == "abc222"
```

```
索引值
string index out of range
返回值
SPORT
name:{},age{}
name:hahah,age66
```

```
print(sss + ss)
```

```
abc222222
```

- ：不支持

```
print(s - sss)
```

```
abc222222
Traceback (most recent call last):
  File "/Users/sjy/cm36/week05/use_of_str.py", line 71, in <module>
    print(s - sss)
          ~~^~~~~
TypeError: unsupported operand type(s) for -: 'str' and 'str'
```

```
try:
    print(s - sss)
except TypeError as aaaaa:
    print(aaaaa)
```

```
abc222222
unsupported operand type(s) for -: 'str' and 'str'
```

*：可以

```
a = "===*===" 
a = a * 6
print(a)
```

```
unsupported operand type(s) for -: 'str' and 'str'
===*======*======*======*======*===
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

/：不支持

```
a = "zzzzz"
a1 = a / 3
```

```
unsupported operand type(s) for -: 'str' and 'str'
===*======*======*======*======*===
Traceback (most recent call last):
  File "/Users/sjy/cm36/week05/use_of_str.py", line 82, in <module>
    a1 = a / 3
         ~~^~~
TypeError: unsupported operand type(s) for /: 'str' and 'int'
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

```
a = "zzzzz"
try:
    a1 = a / 3
except TypeError as c:
    print(c)
```

```
===*======*======*======*======*======*===
unsupported operand type(s) for /: 'str' and 'int'
```

- //：整除

```
x = 2
y = 9
assert y // x == 4
```

```
(week05) sjy@sunjiayideMacBook-Pro week05 % python use_of_int.py
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

- %：余除

```
x = 2
y = 9
assert y // x == 4
assert y % x == 1
```

```
(week05) sjy@sunjiayideMacBook-Pro week05 % python use_of_int.py
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

2、如何判断相等(=)

（1）字符串完全一模一样即相等

```
assert a == "zzzzz"
```

```
name:{},age{}
name:hahah,age66
abc222222
unsupported operand type(s) for -: 'str' and 'str'
===*======*======*======*======*======*===
unsupported operand type(s) for /: 'str' and 'int'
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

3、能不能比较大小(>、<、>=、<=)

✖ 问题出现：不可以随意比较大小

```
print("比较大小")
print("aaa" > "bbb")
```

```
比较大小
False
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

（2）小写字母比大写字母大

```python
print("比较大小")
print("aaa" > "bbb")
print("abc" > "ABC")
```

```
比较大小
False
True
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

大小可以参考 ASCII 表

编号	八进制	十六进制	二进制	符号	HTML 编号	HTML 实体名称	描述
0	000	00	00000000	NUL	&#000;		Null char
1	001	01	00000001	SOH	&#001;		Start of Heading
2	002	02	00000010	STX	&#002;		Start of Text
3	003	03	00000011	ETX	&#003;		End of Text

参照表中选择的一些字符进行比较（若第一个相同，则会比较下一个字符）

```
print("比较大小")
print("aaa" > "bbb")
print("abc" > "ABC")
print("!" < "1")
print("?" > ":")
print("+1" < ":a")
print("111" < "22a")
```

```
比较大小
False
True
True
True
True
True
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

4、什么值被当作 True，什么值被当作 False

字符串长度不为 0 则是 true，为 0 就是 false

```
print("是和否")
assert "sport"
assert ""
```

```
是和否
Traceback (most recent call last):
  File "/Users/sjy/cm36/week05/use_of_str.py", line 100, in <module>
    assert ""
           ^^
AssertionError
```

5、是否可迭代(iterable)，如何做迭代(for 循环)

（1）是否可迭代

```
print("迭代")
a = "sport"
print(iter(a))
```

```
迭代
<str_ascii_iterator object at 0x107114b20>
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

（2）for 循环做迭代：字符串做循环就是把每个字符串跑一遍

```
for b in a:
    print(b)
```

```
迭代
<str_ascii_iterator object at 0x1105ecbb0>
s
p
o
r
t
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

6、是否能返回长度(len)

```
print("长度")
print(len(a))
```

```
长度
5
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

7、索引操作([])：前包后不包

```
print("索引操作")
a = "sport"
assert a[0:4] == "spor"
```

```
索引操作
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

8、常用方法调用

```
week05 — python use_of_str.py — 80×24
-> breakpoint()
(Pdb) p a
'sport'
(Pdb) import wat
(Pdb) wat/a

value: 'sport'
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 comparis
ons.
  def center(width, fillchar=' ', /) # Return a centered string of length width.
…
  def count(…) # S.count(sub[, start[, end]]) -> int…
  def encode(encoding='utf-8', errors='strict') # Encode the string using the co
dec registered for encoding.…
  def endswith(…) # S.endswith(suffix[, start[, end]]) -> bool…
  def expandtabs(tabsize=8) # Return a copy where all tab characters are expande
```

translate：变换内容

```
(Pdb) p a.translate({ord('t'):ord('b')})
'sporb'
(Pdb)
```

capitalize 首字母大写

```
a = "who are you"
print(a.capitalize())
print(a)
```

```
Who are you
who are you
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

count：字符串中指定内容的不重复的个数

```
print(a.count("o") == 2)
```

```
Who are you
who are you
True
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

是否以……结尾

```
(Pdb) p a
'who are you'
(Pdb) p a.endswith('are')
False
(Pdb) p a.endswith('you')
True
(Pdb) p a.endswith('u')
True
(Pdb)
```

出现位置，如果有多个，则显示第一个的位置

```
(Pdb) p a.index('r')
5
(Pdb) p a.index('o')
2
(Pdb) p a.index('are')
4
(Pdb)
```

检验字符串中是否有除字母和数字以外的其他

```
print("aaa111".isalnum())
print("aaa!!111".isalnum())
```

```
True
True
False
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

检验能不能做标志符

```python
print("aaa111".isidentifier())
print("222aaa111".isidentifier())
print("aaa-111".isidentifier())
print("aaa_111".isidentifier())
```

```
True
False
False
True
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

链接列表

```python
a = ["aaa", "bbb", "ccc"]
print(":".join(a))
```

```
aaa:bbb:ccc
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

抹掉内容

```python
print("移除内容")
text = "  Hello, World!  "
stripped_text = text.strip()
print(stripped_text)

text = "---Hello, World!---"
stripped_text = text.strip("-")
print(stripped_text)

text = "***Hello, World!!!***"
stripped_text = text.strip("*!")
print(stripped_text)
```

```
Hello, World!
Hello, World!
Hello, World
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

拆成字符串

```
p = "aaa:bbb:ccc"
print(p.split(":"))
```

```
['aaa', 'bbb', 'ccc']
```

拆分

```
p = "aaa:bbb:ccc"
assert p.partition(":") == ("aaa", ":", "bbb:ccc")
```

```
Hello, World!
Hello, World
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

四、字节串

1、什么是字节串:里面是二进制的东西，不是字符

use_of_bytes.py > ...

```
s = b"sport"
print(s)
```

```
(week05) sjy@sunjiayideMacBook-Pro week05 % python use_of_bytes.py
b'sport'
```

```
print(s[0])
```

```
b'sport'
115
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

```
from pathlib import Path

s = b"sport"
print(s)
print(s[0])
p = Path("/opt/anaconda3/envs/week05/bin/python")
breakpoint()
```

```
(Pdb) p p.is_file()
True
(Pdb) p p.is_dir()
False
(Pdb) p p.exists()
True
(Pdb)
```

```
a = p.read_bytes()
print(len(a))
```

```
7030608
```

```
p = Path("environment.yml")
s = p.read_bytes()
print(s[0])
```

```
110
```

字符串编码得到字节串

```
p = Path("environment.yml")
b = p.read_bytes()
print(b[0])

s = b.decode()
breakpoint()
```

```
(week05) sjy@sunjiayideMacBook-Pro week05 % python use_of_bytes.py
b'sport'
115
7030608
110
--Return--
> /Users/sjy/cm36/week05/use_of_bytes.py(19)<module>()->None
-> breakpoint()
(Pdb)
```

⭐ 中文能否编码变成二进制呢？可以

```
s = "哈嘿啊嘿"
b = s.encode()
```

```
(week05) sjy@sunjiayideMacBook-Pro week05 % python use_of_bytes.py
b'sport'
115
7030608
110
--Return--
> /Users/sjy/cm36/week05/use_of_bytes.py(24)<module>()->None
-> breakpoint()
(Pdb) p b
b'\xe5\x93\x88\xe5\x98\xbf\xe5\x95\x8a\xe5\x98\xbf'
(Pdb)
```

编解码器的差异

```python
s = "哈嘿啊嘿"
b1 = s.encode("utf-8")
print(b1)
b2 = s.encode("gbk")
print(b2)
```

```
b'\xe5\x93\x88\xe5\x98\xbf\xe5\x95\x8a\xe5\x98\xbf'
b'\xb9\xfe\xba\xd9\xb0\xa1\xba\xd9'
```

emoji 也可以编码

```python
sss = "abc我是一个📅"
print(sss)
bbb = sss.encode()
```

```
(week05) sjy@sunjiayideMacBook-Pro week05 % python use_of_bytes.py
b'sport'
115
7030608
110
b'\xe5\x93\x88\xe5\x98\xbf\xe5\x95\x8a\xe5\x98\xbf'
b'\xb9\xfe\xba\xd9\xb0\xa1\xba\xd9'
abc我是一个📅
--Return--
> /Users/sjy/cm36/week05/use_of_bytes.py(31)<module>()->None
-> breakpoint()
(Pdb) p bbb
b'abc\xe6\x98\x91\xe6\x98\xaf\xe4\xb8\x80\xe4\xb8\xaa\xf0\x9f\x93\x85'
(Pdb)
```

```
(Pdb) p bbb[9:].decode()
'一个📅'
(Pdb)
```

五、整数

use_of_int.py > ...

```python
a = 66
b = 9
c = 10
abc = a + b + c


x = 2
y = 9
assert y // x == 4
assert y % x == 1

assert 5
try:
    assert 0
except AssertionError as e:
    print(type(e))

x = 65535
breakpoint()

```

```
(week05) sjy@sunjiayideMacBook-Pro week05 % python use_of_int.py
<class 'AssertionError'>
--Return--
> /Users/sjy/cm36/week05/use_of_int.py(19)<module>()->None
-> breakpoint()
```

六、浮点数

✖ 浮点数最好不判断相等，会有四舍五入的误差

use_of_float.py > ...

```python
import random

x = 66.669
print(type(x))

a = float("66.669")
print(type(a))

assert x == a


bb = 18 / 5
print(bb, type(bb))


x = random.random()
print(x)

assert not 0.0
```

```
(week05) sjy@sunjiayideMacBook-Pro week05 % python use_of_float.py
<class 'float'>
<class 'float'>
3.6 <class 'float'>
0.7040563720174875
(week05) sjy@sunjiayideMacBook-Pro week05 % python use_of_float.py
<class 'float'>
<class 'float'>
3.6 <class 'float'>
0.45246352812746427
```

nan 缺失值：特殊浮点数

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

```
nan
False
False
False
```

无穷大

```python
pinf = float("inf")
print(3.21e-2)
print(pinf > 1e200)
print(pinf > pinf)
print(pinf == pinf)

ninf = float("-inf")
print(ninf)
```

```
0.0321
True
False
True
-inf
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

七、布尔值

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

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

```
(week05) sjy@sunjiayideMacBook-Pro week05 % python use_of_bool.py
True False
<class 'bool'>
True
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

八、列表：通过序号排队找到值

environment.yml U　　use_of_list.py 1, U

use_of_list.py > ...

```python
l = [1, "badui", 3]
print(l)

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

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

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

a = [3, 9]
b = ["a", "d"]

print(a + b)
print(b + a)
print(a + b == b + a)

a = [3, 9]
b = [7]
try:
    print(a - b)
except TypeError as e:
    print(e)
```

```python
a = [3, 9]
print(a * 3)

a = [3, 9]
b = a * 3
a[0] = 10
print(a)
print(b)

a = [2, 9, 200]
b = [i**2 for i in a]
print(b)
b = [i**2 for i in a if i < 66]
print(b)

a = [3, 9]
b = a * 3
x = a.append(8)
print(x)
print(a)
print(b)

breakpoint()
```

```
(week05) sjy@sunjiayideMacBook-Pro week05 % python use_of_list.py
[1, 'badui', 3]
1
badui
3
list index out of range
badui
a
[3, 9, 'a', 'd']
['a', 'd', 3, 9]
False
unsupported operand type(s) for -: 'list' and 'list'
[3, 9, 3, 9, 3, 9]
[10, 9]
[3, 9, 3, 9, 3, 9]
[4, 81, 40000]
[4, 81]
None
[3, 9, 8]
[3, 9, 3, 9, 3, 9]
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

九、字典：没有顺序，是散列表

```
-> breakpoint()
(Pdb) l
  1     d = {"p": 22, "pad": 66, "caper": 99}
  2     print(d)
  3     print(type(d))
  4  -> breakpoint()
[EOF]
(Pdb) p hash('a')
5953040388617670922
(Pdb) p hash('a')
5953040388617670922
(Pdb) p hash('a')
5953040388617670922
(Pdb) p hash(1)
1
(Pdb) p hash(2)
2
(Pdb) p hash('2')
8220323933377837711
(Pdb) p hash('1')
-462573377566774
(Pdb)
```

✖ 空字典会报 false

environment.yml U | use_of_list.py 1, U | use_of_dict.py 1, U

use_of_dict.py > ...

```python
d = {"p": 22, "pad": 66, "caper": 99}
print(d)
print(type(d))


for a in d:
    print(a)

for a in d:
    print(d[a])

for a in d.values():
    print(a)

l = [a for a in d.items()]
print(l)

for k, v in d.items():
    print(k, v)

breakpoint()
```

```
(week05) sjy@sunjiayideMacBook-Pro week05 % python use_of_dict.py
{'p': 22, 'pad': 66, 'caper': 99}
<class 'dict'>
p
pad
caper
22
66
99
22
66
99
[('p', 22), ('pad', 66), ('caper', 99)]
p 22
pad 66
caper 99
--Return--
> /Users/sjy/cm36/week05/use_of_dict.py(21)<module>()->None
-> breakpoint()
(Pdb)
```

十、元组

✓ 只有两个命令

```
(Pdb) import wat
(Pdb) wat /a
value: (1, 'b', 33669)
type: tuple
len: 3

Public attributes:
  def count(value, /) # Return number of occurrences of value.
  def index(value, start=0, stop=9223372036854775807, /) # Return first index of
 value.…
```

- 元祖是不可变的对象，不支持修改
- 可变的对象是不能做键

vironment.yml U | use_of_list.py U | use_of_dict.py 1, U | use_of_tuple.py U

use_of_tuple.py > ...

```python
a = (1, "b", 33669)
print(a)
print(type(a))

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

try:
    a[0] = 20
except TypeError as e:
    print(e)

d = {}
d["abc"] = 5
d[77] = 20177
q = [5, 992]

try:
    d[q] = 443
except TypeError as e:
    print(e)

dd = (3, 1)
d[dd] = 366
print(d)
print(d[3, 1])

aaa = 2, 4, 7, 271
print(aaa)
print(type(aaa))
```

```
(week05) sjy@sunjiayideMacBook-Pro week05 % python use_of_tuple.py
(1, 'b', 33669)
<class 'tuple'>
1
b
33669
'tuple' object does not support item assignment
unhashable type: 'list'
{'abc': 5, 77: 20177, (3, 1): 366}
366
(2, 4, 7, 271)
<class 'tuple'>
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

十一、集合

use_of_set.py U

use_of_set.py > ...

```python
a = {477, 293, 626}
print(a)
print(type(a))

try:
    a = {477, [293], 626}
except TypeError as e:
    print(e)

aa = [66, 44, 33, 88, 99, 33, 99]
print(aa)
bb = set(aa)
print(bb)

aa = {66, 44, 33, 88, 99, 33, 99}
print(aa)
print(44 in aa)
print(688 in aa)
```

```
(week05) sjy@sunjiayideMacBook-Pro week05 % python use_of_set.py
{477, 626, 293}
<class 'set'>
unhashable type: 'list'
[66, 44, 33, 88, 99, 33, 99]
{33, 66, 99, 44, 88}
{33, 66, 99, 88, 44}
True
False
```

- 只有键，没有键值对
- 集合内的数是没有顺序的，是随便放的
- 并集，交集，对称差

```python
cc = {2, 987, 99, 88}
print(cc | aa)
print(cc & aa)
print(cc ^ aa)
```

```
{33, 2, 99, 66, 44, 88, 987}
{88, 99}
{33, 66, 2, 987, 44}
(week05) sjy@sunjiayideMacBook-Pro week05 %
```

十二、pathlib

use_of_set.py U　use_of_path.py U

use_of_path.py > ...

```python
from pathlib import Path

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

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

p = Path(".")
pp = p / "README,md"
print(pp)
ppp = pp.absolute()
print(ppp)
breakpoint()
```

```
(week05) sjy@sunjiayideMacBook-Pro week05 % python use_of_path.py
.
True
/Users/sjy/cm36/week05
[PosixPath('use_of_float.py'), PosixPath('use_of_bool.py'), PosixPath('LICENSE')
, PosixPath('environment.yml'), PosixPath('use_of_tuple.py'), 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'), PosixPat
h('.git')]
False
True
True
README,md
/Users/sjy/cm36/week05/README,md
--Return--
> /Users/sjy/cm36/week05/use_of_path.py(20)<module>()->None
-> breakpoint()
```

十三、datetime

use_of_datetime.py > ...

```python
from datetime import date, datetime, timedelta  # noqa: F401

a = date.today()
aa = date(2025, 11, 11)
b = aa - a
print(b)
print(type(b))
print(b.days)

a = "2024-09-01"
aa = "2024-11-11"
b = datetime.strptime(a, "%Y-%m-%d")
bb = datetime.strptime(aa, "%Y-%m-%d")
print(b)
print(bb)
breakpoint()
```

```
(week05) sjy@sunjiayideMacBook-Pro week05 % python use_of_datetime.py
218 days, 0:00:00
<class 'datetime.timedelta'>
218
2024-09-01 00:00:00
2024-11-11 00:00:00
--Return--
> /Users/sjy/cm36/week05/use_of_datetime.py(16)<module>()->None
-> breakpoint()
(Pdb) p b
datetime.datetime(2024, 9, 1, 0, 0)
(Pdb) p format(b)
'2024-09-01 00:00:00'
(Pdb) p format(b,"%a")
'Sun'
(Pdb) p format(bb,"%a")
'Mon'
(Pdb) p format(bb,"%B")
'November'
(Pdb) p format(b,"%B")
'September'
(Pdb) p format(b,"%B,%d,%A")
'September,01,Sunday'
(Pdb)
```