Week05

1. Str

```
use_of_str.py > ...
  1    a = "str验证,在这是一个字符串"
  2    a1 = "str验证,在这是一个字符串"
  3    b = "另一个字符串"
  4    print(a)
  5    x = id(a)
  6    print(x)
  7    print(id(a1))
  8    y = id(b)
  9    print(y)
 10    c = [100, 65]
 11    d = [55, 989]
 12    print(id(c))
 13    print(id(d))
 14
```

```
问题    输出    调试控制台    终端    端口                                    + ∨  ⊡ Python  ⊓ 🗑 ⋯ ∧ ✕

2929578060032
PS C:\Users\Zhaogs\Desktop\金融编程\week05> & E:/Anaconda3/python.exe c:/Users/Zhaogs/Desktop/金融编程/week05/use_of_str.
py
str验证,在这是一个字符串
2453522603232
2453522603232
2453527419456
2453527159744
2453530060992
```

内容相同字符串的 id 是相同的，但是内容相同列表的 id 是不一样的

```
use_of_str.py > ...
  9    print(y)
 10    c = [100, 65]
 11    d = [55, 989]
 12    print(id(c))
 13    print(id(d))
 14    print(type(a))
 15    print(type(c))
 16    print(type(x))
 17    print(isinstance(a, str))   # 如果a是str,则为true,否则为false
 18    print(isinstance(c, str))
 19    print(dir(a))
 20    print(str(c))
 21    try:
 22        assert isinstance(c, str)   # 如果报错会直接退出,未报错则直接
 23    except AssertionError:
 24        breakpoint() #进入断点调试
 25        print("报错啦")
 26    print("end")
 27
```

```
问题    输出    调试控制台    终端    端口                                    + ∨  ⊡ Python  ⊓ 🗑 ⋯ ∧ ✕

True
False
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getatt
ribute__', '__getitem__', '__getnewargs__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__', '__le__',
'__len__', '__lt__', '__mod__', '__mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__rmod__',
'__rmul__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', 'capitalize', 'casefold', 'center', 'count', 'encod
e', 'endswith', 'expandtabs', 'find', 'format', 'format_map', 'index', 'isalnum', 'isalpha', 'isascii', 'isdecimal', 'isd
igit', 'isidentifier', 'islower', 'isnumeric', 'isprintable', 'isspace', 'istitle', 'isupper', 'join', 'ljust', 'lower',
'lstrip', 'maketrans', 'partition', 'removeprefix', 'removesuffix', 'replace', 'rfind', 'rindex', 'rjust', 'rpartition',
'rsplit', 'rstrip', 'split', 'splitlines', 'startswith', 'strip', 'swapcase', 'title', 'translate', 'upper', 'zfill']
[100, 65]
报错啦
end
```

2. 通过表达式得到实例
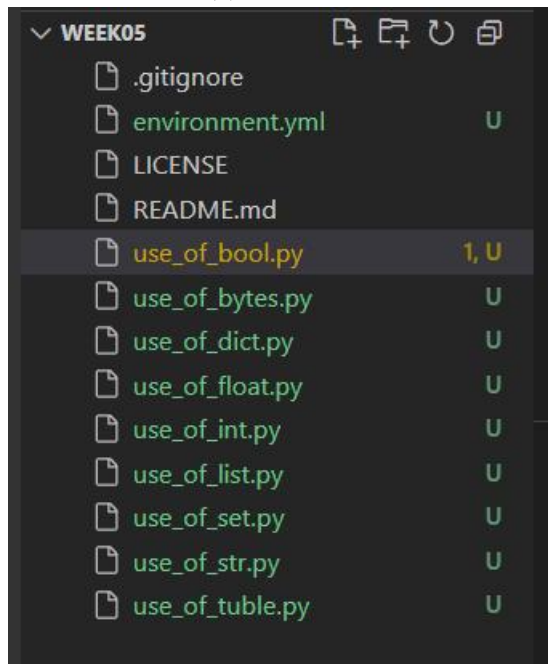
常用的都尝试操作了一下

```python
# 布尔值 (bool)
bool_example = True

# 使用内置函数检视对象
print(f"ID: {id(bool_example)}")
print(f"Type: {type(bool_example)}")
print(f"Is instance of bool: {isinstance(bool_example, bool)}")
print(f"Attributes and methods: {dir(bool_example)}")
print(f"Boolean representation: {str(bool_example)}")

# 数学运算符支持
result = bool_example + 1
print(f"Addition: {result}")

# 判断相等
assert bool_example == True, "Booleans are not equal"

# 比较运算符支持
another_bool = False
if another_bool < bool_example:
    print("another_bool is less than bool_example")
```

```
'__rand__', '__rdivmod__', '__reduce__', '__reduce_ex__', '__repr__', '__rfloordiv__', '__rlshift__', '__rmod__', '__rmu
l__', '__ror__', '__round__', '__rpow__', '__rrshift__', '__rshift__', '__rsub__', '__rtruediv__', '__rxor__', '__setattr
__', '__sizeof__', '__str__', '__sub__', '__subclasshook__', '__truediv__', '__trunc__', '__xor__', 'as_integer_ratio', '
bit_length', 'conjugate', 'denominator', 'from_bytes', 'imag', 'numerator', 'real', 'to_bytes']
Boolean representation: True
Addition: 2
another_bool is less than bool_example
True value evaluates to: True
False value evaluates to: False
'bool' object is not iterable
object of type 'bool' has no len()
'bool' object is not subscriptable
Not operation: False
```

```python
    # 数字运算符支持
12  try:
13      result = bytes_example + b" Python"
14      print(f"Concatenation: {result}")
15  except TypeError as e:
16      print(e)
17
18  # 判断相等
19  assert bytes_example == b"Hello, Bytes!", "Bytes are not equal"
20
21  # 比较运算符支持
22  another_bytes = b"Hello, Universe!"
23  if another_bytes > bytes_example:
24      print("another_bytes is greater than bytes_example")
25
26  # 布尔值判断
27  empty_bytes = b""
28  non_empty_bytes = b"Non-empty"
29  print(f"Empty bytes evaluates to: {bool(empty_bytes)}")
30  print(f"Non-empty bytes evaluates to: {bool(non_empty_bytes)}")
31
32  # 可迭代性
```

问题 1   输出   调试控制台   终端   端口   + ∨   Python   ⊓   🗑   …   ∧   ✕

```
 'removesuffix', 'replace', 'rfind', 'rindex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines', 'startsw
ith', 'strip', 'swapcase', 'title', 'translate', 'upper', 'zfill']
Bytes representation: b'Hello, Bytes!'
Concatenation: b'Hello, Bytes! Python'
another_bytes is greater than bytes_example
Empty bytes evaluates to: False
Non-empty bytes evaluates to: True
72 101 108 108 111 44 32 66 121 116 101 115 33
Length of bytes_example: 13
First byte: 72
Last byte: 33
Hexadecimal: 48656c6c6f2c20427974657321
From hexadecimal: b'Hello, Bytes!'
PS C:\Users\Zhaogs\Desktop\金融编程\week05> ▯
```

📄 use_of_dict.py › …

```python
35  for key in dict_example:
36      print(key, dict_example[key])
37
38  # 长度
39  print(f"Length of dict_example: {len(dict_example)}")
40
41  # 索引操作
42  print(f"Value of 'name': {dict_example['name']}")
43
44  # 常用方法
45  dict_example["city"] = "Wonderland"
46  print(f"After adding city: {dict_example}")
47  del dict_example["age"]
48  print(f"After deleting age: {dict_example}")
49  keys = dict_example.keys()
50  values = dict_example.values()
51  items = dict_example.items()
52  print(f"Keys: {keys}")
53  print(f"Values: {values}")
54  print(f"Items: {items}")
55
```

问题 1   输出   调试控制台   终端   端口   + ∨   Python   ⊓   🗑   …   ∧   ✕

```
unsupported operand type(s) for +: 'dict' and 'dict'
'>' not supported between instances of 'dict' and 'dict'
Empty dictionary evaluates to: False
Non-empty dictionary evaluates to: True
name Alice
age 25
Length of dict_example: 2
Value of 'name': Alice
After adding city: {'name': 'Alice', 'age': 25, 'city': 'Wonderland'}
After deleting age: {'name': 'Alice', 'city': 'Wonderland'}
Keys: dict_keys(['name', 'city'])
Values: dict_values(['Alice', 'Wonderland'])
Items: dict_items([('name', 'Alice'), ('city', 'Wonderland')])
PS C:\Users\Zhaogs\Desktop\金融编程\week05> ▯
```

```python
# 浮点数 (float)
float_example = 3.14159

# 使用内置函数检视对象
print(f"ID: {id(float_example)}")
print(f"Type: {type(float_example)}")
print(f"Is instance of float: {isinstance(float_example, float)}")
print(f"Attributes and methods: {dir(float_example)}")
print(f"Float representation: {str(float_example)}")

# 数学运算符支持
result = float_example + 1.0
print(f"Addition: {result}")

# 判断相等
assert abs(float_example - 3.14159) < 1e-9, "Floats are not equal"

# 比较运算符支持
another_float = 3.14160
if another_float > float_example:
    print("another_float is greater than float_example")
```

```
, '__repr__', '__rfloordiv__', '__rmod__', '__rmul__', '__round__', '__rpow__', '__rsub__', '__rtruediv__', '__set_format
__', '__setattr__', '__sizeof__', '__str__', '__sub__', '__subclasshook__', '__truediv__', '__trunc__', 'as_integer_ratio
', 'conjugate', 'fromhex', 'hex', 'imag', 'is_integer', 'real']
Float representation: 3.14159
Addition: 4.14159
another_float is greater than float_example
Zero float evaluates to: False
Non-zero float evaluates to: True
'float' object is not iterable
object of type 'float' has no len()
'float' object is not subscriptable
Exponentiation: 9.869587728099999
Square root: 1.7724531023414978
PS C:\Users\Zhaogs\Desktop\金融编程\week05>
```

📄 use_of_int.py ⟩ ...

```python
    # 不可迭代
    try:
        for i in int_example:
            print(i)
    except TypeError as e:
        print(e)

    # 长度
    try:
        len(int_example)
    except TypeError as e:
        print(e)

    # 索引操作
    try:
        print(int_example[0])
    except TypeError as e:
        print(e)

    # 常用方法
    print(f"Binary: {bin(int_example)}")
```

```
__', '__sizeof__', '__str__', '__sub__', '__subclasshook__', '__truediv__', '__trunc__', '__xor__', 'as_integer_ratio', '
bit_length', 'conjugate', 'denominator', 'from_bytes', 'imag', 'numerator', 'real', 'to_bytes']
Integer representation: 42
Addition: 52
another_int is greater than int_example
Zero integer evaluates to: False
Non-zero integer evaluates to: True
'int' object is not iterable
object of type 'int' has no len()
'int' object is not subscriptable
Binary: 0b101010
Hexadecimal: 0x2a
Octal: 0o52
```

```python
use_of_list.py > ...
20
21    # 比较运算符支持
22    another_list = [1, 2, 3, 4, 6]
23    if another_list > list_example:
24        print("another_list is greater than list_example")
25
26    # 布尔值判断
27    empty_list = []
28    non_empty_list = [1, 2, 3]
29    print(f"Empty list evaluates to: {bool(empty_list)}")
30    print(f"Non-empty list evaluates to: {bool(non_empty_list)}")
31
32    # 可迭代性
33    for item in list_example:
34        print(item, end=" ")
35    print()
36
37    # 长度
38    print(f"Length of list_example: {len(list_example)}")
39
40    # 索引操作
```

```
问题 1    输出    调试控制台    终端    端口                          + ∨  >_ Python  ⊞  🗑  …  ∧
']
List representation: [1, 2, 3, 4, 5]
Concatenation: [1, 2, 3, 4, 5, 6, 7]
another_list is greater than list_example
Empty list evaluates to: False
Non-empty list evaluates to: True
1 2 3 4 5
Length of list_example: 5
First element: 1
Last element: 5
After append: [1, 2, 3, 4, 5, 6]
After remove: [1, 2, 4, 5, 6]
After sort: [6, 5, 4, 2, 1]
```

```python
use_of_tuble.py > ...
 7        print(f"is instance of tuple: {isinstance(tuple_example, tuple)}")
 8        print(f"Attributes and methods: {dir(tuple_example)}")
 9        print(f"Tuple representation: {str(tuple_example)}")
10
11    # 数学运算符支持
12 ∨  try:
13        result = tuple_example + (6, 7)
14        print(f"Concatenation: {result}")
15 ∨  except TypeError as e:
16        print(e)
17
18    # 判断相等
19    assert tuple_example == (1, 2, 3, 4, 5), "Tuples are not equal"
20
21    # 比较运算符支持
22    another_tuple = (1, 2, 3, 4, 6)
23 ∨  if another_tuple > tuple_example:
24        print("another_tuple is greater than tuple_example")
25
26    # 布尔值判断
27    empty_tuple = ()
28    non_empty_tuple = (1, 2, 3)
```

```
问题 1    输出    调试控制台    终端    端口                          + ∨  >_ Python  ⊞  🗑  …  ∧  ✕
, '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mul__', '__ne__', '__new__', '__reduce__', '__reduce
_ex__', '__repr__', '__rmul__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', 'count', 'index']
Tuple representation: (1, 2, 3, 4, 5)
Concatenation: (1, 2, 3, 4, 5, 6, 7)
another_tuple is greater than tuple_example
Empty tuple evaluates to: False
Non-empty tuple evaluates to: True
1 2 3 4 5
Length of tuple_example: 5
First element: 1
Last element: 5
Count of 3: 1
Index of 4: 3
```