

第五周 金融编程与计算作业

0Python 常用值类型:

类型	描述	说明
数字 (Number)	支持 <ul style="list-style-type: none">整数 (int)浮点数 (float)复数 (complex)布尔 (bool)	整数 (int), 如: 10、-10
		浮点数 (float), 如: 13.14、-13.14
		复数 (complex), 如: 4+3j, 以j结尾表示复数
		布尔 (bool) 表达现实生活中的逻辑, 即真和假, True表示真, False表示假。 True本质上是一个数字记作1, False记作0
字符串 (String)	描述文本的一种数据类型	字符串 (string) 由任意数量的字符组成
列表 (List)	有序的可变序列	Python中使用最频繁的数据类型, 可有序记录一堆数据
元组 (Tuple)	有序的不可变序列	可有序记录一堆不可变的Python数据集合
集合 (Set)	无序不重复集合	可无序记录一堆不重复的Python数据集合
字典 (Dictionary)	无序Key-Value集合	可无序记录一堆Key-Value型的Python数据集合

bytes: 是一个不可变 (immutable) 的序列类型, 用于表示字节序列 (即二进制数据)

关键特性: ①不可变性: bytes 对象创建后不能被修改 (类似于 str 或 tuple)。

②范围: 每个元素是 0 到 255 之间的整数 (即一个字节的数值)。

③字面量表示: 前缀 b 表示字节串, 例如: b'hello'; 非 ASCII 字符需要用转义形式 (如 b'\xff\x00')。

1Fork 第05周打卡 仓库至你的名下, 然后将你名下的这个仓库 Clone 到你的本地计算机

```
(base) 86157@LAPTOP-GMTRB58B MINGW64 ~/repo
$ pwd
/c/Users/86157/repo

(base) 86157@LAPTOP-GMTRB58B MINGW64 ~/repo
$ git clone git@gitcode.com:Typing_lqqqq/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-reuse
Receiving objects: 100% (5/5), 8.44 KiB | 2.81 MiB/s, don

(base) 86157@LAPTOP-GMTRB58B MINGW64 ~/repo
$ ls -l
total 25
drwxr-xr-x 1 86157 197609 0 3月 15 21:53 mywork/
drwxr-xr-x 1 86157 197609 0 3月 20 20:47 prj1/
-rw-r--r-- 1 86157 197609 13 3月 8 22:36 script1.py
drwxr-xr-x 1 86157 197609 0 3月 9 14:47 week01/
drwxr-xr-x 1 86157 197609 0 3月 9 12:12 week01_9806/
drwxr-xr-x 1 86157 197609 0 3月 16 21:40 week02/
drwxr-xr-x 1 86157 197609 0 3月 23 20:39 week03/
drwxr-xr-x 1 86157 197609 0 3月 30 19:51 week04/
drwxr-xr-x 1 86157 197609 0 3月 30 20:17 week04_7861/
drwxr-xr-x 1 86157 197609 0 4月 6 19:00 week05/
```

2用 VS Code 打开项目目录, 新建一个 environment.yml 文件, 指定安装 Python 3.12, 然后运行 conda env create 命令创建 Conda 环境

```
! environment.yml U X
! environment.yml
1  name: week05
2  channels:
3    - conda-forge
4  dependencies:
5    - python=3.12
6    - wat-inspector
7
```

3 逐个 创建 use_of_{name}.py 文件, 其中 {name} 替换为上述要求掌握的对象类型, 例如 use_of_str.py:

```
! environment.yml U
🔑 LICENSE
📖 README.md
🔗 use_of_bool.py U
🔗 use_of_bytes.py U
🔗 use_of_datetime.py U
🔗 use_of_dict.py U
🔗 use_of_float.py U
🔗 use_of_int.py U
🔗 use_of_list.py U
🔗 use_of_path.py U
🔗 use_of_set.py U
🔗 use_of_str.py U
🔗 use_of_tuple.py U
```

4Str 以下是对 str 作为例子操作的截图

```
use_of_str.py > ...
1  a = [2, 5]
2  b = [2, 5]
3  x = id(a)
4  print(a)
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):", isinstance(a, str))
14

(base) 86157@LAPTOP-GMTRB58B MINGW64 ~/repo/week05
$ conda activate list

EnvironmentNameNotFound: Could not find conda environment.
You can list all discoverable environments with 'conda env list'.

(base) 86157@LAPTOP-GMTRB58B MINGW64 ~/repo/week05
$ conda activate week05
(week05)
86157@LAPTOP-GMTRB58B MINGW64 ~/repo/week05 (m
$ python use_of_str.py
(week05)
86157@LAPTOP-GMTRB58B MINGW64 ~/repo/week05 (m
$ python use_of_str.py
[2, 5]
2081631179072
[9, 5]
[2, 5]
2081631181056
2081631179072
<class 'list'>
isinstance(a,str): False
(week05)
86157@LAPTOP-GMTRB58B MINGW64 ~/repo/week05 (m
```

```

use_of_str.py > ...
1 print("字面值")
2 s = "university"
3 print(s)
4 print(isinstance(s, str))
5 assert type(s) is str
6

```

tion` is deprecated and will be removed in
 --file=URL` instead.
 action(self, namespace, argument_values,
 Retrieving notices: ...working... done
 CondaValueError: prefix already exists: D:

(base) 86157@LAPTOP-GMTRB58B MINGW64 ~/re
 \$ conda env activate
 usage: conda-script.py env [-h] command ..
 conda-script.py env: error: argument comma
 e' (choose from 'config', 'create', 'expor
 e')

(base) 86157@LAPTOP-GMTRB58B MINGW64 ~/re
 \$ conda activate
 (base)
 86157@LAPTOP-GMTRB58B MINGW64 ~/repo/week0
 \$ conda activate week05
 (week05)
 86157@LAPTOP-GMTRB58B MINGW64 ~/repo/week0
 \$ python use_of_str.py
 字面值
 university
 True
 (week05)

```

use_of_str.py > ...
1 print("字面值")
2 s = "university"
3 print(s)
4 print(isinstance(s, str))
5 assert type(s) is str
6
7 print("f-string")
8 x = "Tom"
9 s = f"name: {x}"
10 print(s)
11

```

(base)
 86157@LAPTOP-GMTRB58B MINGW64
 \$ conda activate week05
 (week05)
 86157@LAPTOP-GMTRB58B MINGW64
 \$ python use_of_str.py
 字面值
 university
 True
 (week05)
 86157@LAPTOP-GMTRB58B MINGW64
 \$ python use_of_str.py
 字面值
 university
 True
 (week05)
 86157@LAPTOP-GMTRB58B MINGW64
 \$
 (week05)
 86157@LAPTOP-GMTRB58B MINGW64
 \$ python use_of_str.py
 字面值
 university
 True
 f-string
 name: Tom
 (week05)
 86157@LAPTOP-GMTRB58B MINGW64
 \$

```

11
12 s = "a\tb"
13 print("pig", s)
14
15 s = "ccc\nddd"
16 print("cat", s)
17

```

university
 True
 f-string
 name: Tom
 (week05)
 86157@LAPTOP-GMTRB58B MINGW64 ~/re
 \$ python use_of_str.py
 字面值
 university
 True
 f-string
 name: Tom
 pig a b
 cat ccc
 ddd
 (week05)
 86157@LAPTOP-GMTRB58B MINGW64 ~/re

```

k
4
--Return--
> c:\users\86157\repo\week05\use_of_str.py(128)<module>()->None
-> breakpoint()
(Pdb) l
123     print(len(s))
124
125     # 是否（如何）支持索引操作（subscription）（[] 运算符）
126     s = "book"
127     assert s[1:3] # 1是包含的，3是不包含的，结果是oo
128     -> breakpoint()
129
130     # 拥有哪些常用方法（method）可供调用（() 运算符）
[EOF]
(Pdb) p s
'book'
(Pdb) import wat
(Pdb) wat /s

value: 'book'
type: str
len: 4

Public attributes:
  def capitalize() # Return a capitalized version of the string...

```

```

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(...) # S.count(sub[, start[, end]]) -> int...
  def encode(encoding='utf-8', errors='strict') # Encode the string using the codec registered for en
  def endswith(...) # S.endswith(suffix[, start[, end]]) -> bool...
  def expandtabs(tabsize=8) # Return a copy where all tab characters are expanded using spaces...
  def find(...) # S.find(sub[, start[, end]]) -> int...
  def format(...) # S.format(*args, **kwargs) -> str...
  def format_map(...) # S.format_map(mapping) -> str...
  def index(...) # S.index(sub[, start[, end]]) -> int...
  def isalnum() # Return True if the string is an alpha-numeric 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...

```

查看翻译属性（但是不常用）

```

(Pdb) wat / s.translate

value: <built-in method translate of str object at 0x0000021C3EF33450>
type: builtin_function_or_method
signature: def translate(table, /)
"""
Replace each character in the string using the given translation table.

    table
        Translation table, which must be a mapping of Unicode ordinals to
        Unicode ordinals, strings, or None.

The table must implement lookup/indexing via __getitem__, for instance a
dictionary or list. If this operation raises LookupError, the character is
left untouched. Characters mapped to None are deleted.
"""

```

```
(Pdb) p s.translate({ord('o'):ord('t')})  
'bttk'
```

```
value: <built-in method maketrans of type object at 0x00007FF87A73A900>  
type: builtin_function_or_method  
signature: def maketrans(...)  
"""  
Return a translation table usable for str.translate().  
  
If there is only one argument, it must be a dictionary mapping Unicode  
ordinals (integers) or characters to Unicode ordinals, strings or None.  
Character keys will be then converted to ordinals.  
If there are two arguments, they must be strings of equal length, and  
in the resulting dictionary, each character in x will be mapped to the  
character at the same position in y. If there is a third argument, it  
must be a string, whose characters will be mapped to None in the result.  
"""
```

常用的命令:

capitalize() -- 大写首字母

```
4  
The book of why  
the book of why  
(week05)
```

count() -- 数字符串

```
The book of why water you thoo nooo  
True  
the book of why water you thoo nooo  
(week05)
```

endswith -- 以什么字母结尾

```
(Pdb) p s  
'book'  
(Pdb) p s.endswith('ok')  
True  
(Pdb) p s.endswith('coo ok')  
False  
(Pdb) p s.endswith('coook')  
False  
(Pdb) p s.endswith('ook')  
True  
(Pdb) █
```

Format

Index -- 字母出现的序号

```
True
(Pdb) p s.index('o')
1
(Pdb)
```

字节串

```
$ python use_of_bytes.py
b'hello'
(week05)
```

```
$ python use_of_bytes.py
104
(week05)
```

判断是不是文件还是文件夹

```
86157@LAPTOP-GMTRB58B MINGW64 ~/repo/week05 (main)
$ python use_of_bytes.py
104
--Return--
> c:\users\86157\repo\week05\use_of_bytes.py(9)<module>()->None
-> breakpoint()
(Pdb) p p
WindowsPath('D:/86157/anaconda3/envs/week05/python.exe')
(Pdb) p p.exists()
True
(Pdb) p p.is_file()
True
(Pdb) p p.is_dir()
False
(Pdb)
```

```
False
(Pdb) import wat
(Pdb) wat \p
*** SyntaxError: unexpected character after line continuation character
(Pdb) wat /p

str: D:\86157\anaconda3\envs\week05\python.exe
repr: WindowsPath('D:/86157/anaconda3/envs/week05/python.exe')
type: pathlib.WindowsPath
parents: pathlib.Path, pathlib.PureWindowsPath, pathlib.PurePath

Public attributes:
  anchor: str = 'D:\'
  drive: str = 'D:'
  name: str = 'python.exe'
  parent: pathlib.WindowsPath = D:\86157\anaconda3\envs\week05
  parents: pathlib._PathParents = <WindowsPath.parents>
  parts: tuple = ('D:\\', '86157', 'anaconda3', 'envs', 'week05', 'python.exe')
  root: str = '\
  stem: str = 'python'
  suffix: str = '.exe'
  suffixes: list = [...]
```

```
(week05)
86157@LAPTOP-GMTRB58B MINGW64 ~/repo/week05
$ python use_of_bytes.py
104
93184
--Return--
> c:\users\86157\repo\week05\use_of_bytes.p
-> breakpoint()
(Pdb) █
```

```
(Pdb) p s
b'name: week05\r\nchannels:\r\n - conda-forge\r\ndependencies:\r\n - python=3.12\r\n - wat-inspector\r\n '
(Pdb) p s.decode
<built-in method decode of bytes object at 0x000001E2ECADB360>
(Pdb) p s.decode()
'name: week05\r\nchannels:\r\n - conda-forge\r\ndependencies:\r\n - python=3.12\r\n - wat-inspector\r\n '
(Pdb)
```

可见字符和不可见字符

```
[EOF]
(Pdb) p b
b'\xe4\xbd\xa0\xe5\xa5\xbd'
(Pdb)
```

```
(Pdb) p '\x5a'
'Z'
(Pdb)
```

```
(Pdb) p s
'你好'
(Pdb) p b
b'\xe4\xbd\xa0\xe5\xa5\xbd'
(Pdb) p b[0]
228
(Pdb) p b[1]
189
(Pdb)
```

```
5
104
b'\xe4\xbd\xa0\xe5\xa5\xbd'
b'\xc4\xe3\xba\xc3'
--Return--
> c:\users\86157\repo\week05\use_o-
> breakpoint()
b'\xe4\xbd\xa0\xe5\xa5\xbd'
b'\xc4\xe3\xba\xc3'
abc你好 😊
--Return--
> c:\users\86157\repo\week05\use_o-
```



```

abc你好 😊
--Return--
> c:\users\86157\repo\week05\use_of_bytes.py(28)<module>()->None
-> breakpoint()
(Pdb) p b
b'abc\xe4\xbd\xa0\xe5\xa5\xbd\xf0\x9f\x98\x84'
(Pdb) p b[3:]
b'\xe4\xbd\xa0\xe5\xa5\xbd\xf0\x9f\x98\x84'
(Pdb) p b[3:].decode()
'你好 😊'
(Pdb) p b[3:9].decode()
'你好'
(Pdb) p b[9:]
b'\xf0\x9f\x98\x84'
(Pdb) p b[9:].decode()
'😊'
(Pdb)

```

```

86157@LAPTOP-GMTRB58B MINGW64 ~
$ python use_of_int.py
(week05)

```

```

$ python use_of_int.py
(week05)

```

```

*** SyntaxError: expected ':'
(Pdb) for i in x:
*** IndentationError: expected an indented block after 'for' statement on line 1
(Pdb) for i in x:print(x)
*** TypeError: 'int' object is not iterable
(Pdb) p iter(x)
*** TypeError: 'int' object is not iterable
(Pdb)

```

```

(Pdb) px
*** NameError: name 'px' is not defined
(Pdb) p x
3
(Pdb) p x.to_bytes
<built-in method to_bytes of int object at 0x0000024957B7B070>
(Pdb) p x.to_bytes()
b'\x03'
(Pdb)

```

```

(Pdb) p x
8837
(Pdb) p x.to_bytes
<built-in method to_bytes of int object at 0x0000024957B7B070>
(Pdb) p x.to_bytes()
*** OverflowError: int too big to convert
(Pdb) p x.to_bytes(2)
b'\x85'
(Pdb)

```