

Week05 作业

一、常用的对象检视函数和语句

```
MINGW64/c/Users/Lenovo/repo/week05
-rw-r--r-- 1 Lenovo 197121 73 Apr 7 19:57 environment.yml
(base)
Lenovo@DESKTOP-VCHBCBH MINGW64 ~/repo (main)
$ cd week05
(base)
Lenovo@DESKTOP-VCHBCBH MINGW64 ~/repo/week05 (main)
$ ls -l
total 25
-rw-r--r-- 1 Lenovo 197121 18805 Apr 7 19:51 LICENSE
-rw-r--r-- 1 Lenovo 197121 2239 Apr 7 19:51 README.md
-rw-r--r-- 1 Lenovo 197121 91 Apr 7 20:01 environment.yml
(base)
Lenovo@DESKTOP-VCHBCBH MINGW64 ~/repo/week05 (main)
$ conda env create
Error while loading conda entry point: conda-libmamba-solver (DLL load failed while
importing bindings: 找不到指定的模块。)
Retrieving notices: done

CondaValueError: You have chosen a non-default solver backend (libmamba) but it
was not recognized. Choose one of: classic
(base)
Lenovo@DESKTOP-VCHBCBH MINGW64 ~/repo/week05 (main)
$
```

报错解决办法：重新安装 libmamba 相关依赖

```
environment.yml
1 a = [1, 5]
2 b = [2, 5]
3 x = id(a)
4 print(a)
5 y = id(b)
6 print(y)
7 a[] = 8
8 print(a)
9 print(b)
10 print(id(a)) # Is it same as x?
11 print(id(b)) # Is it same as y?
12 print(type(a))
13 print(isinstance(a, str), isinstance(a, str))
14 print(isinstance(a, list), isinstance(a, list))
15 print(isinstance(a, str, float))
16 try:
17     assert isinstance(a, str)
18 except AssertionError:
19     print('type error')
20 print('goodbye!')
```

然后你就继续练了

二、获得 str 类型实例的几种途径

```
environment.yml
1 s = 'university'
2 print(s)
3 print(isinstance(s, str))
4 assert type(s) is str
5
6 print('f-string')
7 x = 'Tom'
8 s = f'name: {x}'
9 print(s)
10
11 s = 'a/b'
12 print('TAB', s)
13
14 s = 'aaa\bbbb'
15 print('New Line', s)
16
17 # raw-string, translate, multi-line
18 s = """xyz
19 abc
20 eee
21 aaa
22 """
23 print(s)
24
25 print("初始化")
26 s = str()
27 print(s)
28 s = str([5, 8, 2])
29 print(s)
30 s = str(1.1 + 2.2)
31 print(s)
32
33 assert str([5, 8, 2]) == '[5, 8, 2]'
34 assert str(1.1 + 2.2) == '3.3'
```

三、str 类型支持的各种操作和方法

六、tuple~date 等类型

```
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
def is_file() # Whether this path is a regular file (also True for symlinks pointing...)
def is_junction() # Whether this path is a junction.
def is_mount() # Check if this path is a mount point
def is_relative_to(other, /, *_deprecated) # Return True if the path is relative to another path or False.
def is_reserved() # Return True if the path contains one of the special names reserved...
def is_socket() # Whether this path is a socket.
def is_symlink() # Whether this path is a symbolic link.
def iterdir() # Yield path objects of the directory contents...
def joinpath(*pathsegments) # Combine this path with one or several arguments, and return a...
def lchmod(mode) # Like chmod(), except if the path points to a symlink, the symlink's...
def lstat() # Like stat(), except if the path points to a symlink, the symlink's...
def match(path_pattern, *, case_sensitive=None) # Return True if this path matches the given pattern.
def mkdir(mode=511, parents=False, exist_ok=False) # Create a new directory at this given path.
def open(mode='r', buffering=-1, encoding=None, errors=None, newline=None) # Open the file pointed to by this path and return a file object, as...
def owner() # Return the login name of the file owner.
def read_bytes() # Open the file in bytes mode, read it, and close the file.
def read_text(encoding=None, errors=None) # Open the file in text mode, read it, and close the file.
def readlink() # Return the path to which the symbolic link points.
def relative_to(other, /, *_deprecated, walk_up=False) # Return the relative path to another path identified by the passed...
```