

day06 数据类型（四）

今日内容

- 集合
- 内存相关
- 深浅拷贝

内容回顾 & 补充

1. 内容回顾

2. 补充

◦ 列表

- reverse, 反转。

```
v1 = [1,2,3111,32,13]
print(v1)
v1.reverse()
print(v1)
```

- sort

```
v1 = [11,22,3111,32,13]
print(v1)

# v1.sort(reverse=False) # 从小到大（默认）
# v1.sort(reverse=True) # 从大到小
# print(v1)
```

◦ 字典

- keys/values/items
- get

```
info = {'k1':'v1','k2':'v2'}

# v1 = info['k11111']
# v2 = info.get('k1111') # None就是Python中的空
# v3 = info.get('k1111',666)
# print(v2)

# None数据类型，改类型表示空（无任何功能，专门用于提供空值）
```

- pop

```
info = {'k1': 'v1', 'k2': 'v2'}
result = info.pop('k2')
print(info, result)

del info['k1']
```

- update

```
info = {'k1': 'v1', 'k2': 'v2'}

# 不存在, 则添加/存在, 则更新
info.update({'k3': 'v3', 'k4': 'v4', 'k2': 666})
print(info)
```

- 判断一个字符串中是否有敏感字符?

- str

```
v = "Python全栈21期"

if "全栈" in v:
    print('含敏感字符')
```

- list/tuple

```
v = ['alex', 'oldboy', '藏老四', '利奇航']

if "利奇航" in v:
    print('含敏感')
```

- dict

```
v = {'k1': 'v1', 'k2': 'v2', 'k3': 'v3'}

# 默认按照键判断, 即: 判断x是否是字典的键。
if 'x' in v:
    pass

# 请判断: k1 是否在其中?
if 'k1' in v:
    pass

# 请判断: v2 是否在其中?
# 方式一: 循环判断
flag = '不存在'
for v in v.values():
    if v == 'v2':
        flag = '存在'
print(flag)

# 方式二:
if 'v2' in list(v.values()): # 强制转换成列表 ['v1', 'v2', 'v3']
```

```

pass
# 请判断: k2:v2 是否在其中?
value = v.get('k2')
if value == 'v2':
    print('存在')
else:
    print('不存在')

```

■ 练习题

```

# 让用户输入任意字符串，然后判断此字符串是否包含指定的敏感字符。

char_list = ['利奇航', '堂有光', '炸展会']
content = input('请输入内容: ') # 我叫利奇航 / 我是堂有光 / 我要炸展会

success = True

for v in char_list:
    if v in content:
        success = False
        break

if success:
    print(content)
else:
    print('包含敏感字符')

# 示例:
# 1. 昨天课上最后一题
# 2. 判断 'v2' 是否在字典的value中 v = {'k1': 'v1', 'k2': 'v2', 'k3': 'v3'} 【循环判断】
# 3. 敏感字

```

内容详细

1. 集合 set

- 无序
- 无重复

```

v = {1,2,3,4,5,6,99,100}

# 疑问: v = {}
"""
None
int
    v1 = 123
    v1 = int() --> 0
bool

```

```

v2 = True/False
v2 = bool() -> False
str
v3 = ""
v3 = str()
list
v4 = []
v4 = list()
tuple
v5 = ()
v5 = tuple()
dict
v6 = {}
v6 = dict()
set
v7 = set()
"""

```

1. 集合独有功能

- add
- discard
- update
- intersection
- union
- difference
- symmetric_difference

2. 公共功能

- len

```

v = {1,2,'李邵奇'}
print(len(v))

```

- for循环

```

v = {1,2,'李邵奇'}
for item in v:
    print(item)

```

- 索引【无】
- 步长【无】
- 切片【无】
- 删除【无】
- 修改【无】

3. 嵌套问题

```

# 1. 列表/字典/集合 -> 不能放在集合中+不能作为字典的key (unhashable)
# info = {1, 2, 3, 4, True, "国风", None, (1, 2, 3)}

```

```
# print(info)
# 2. hash -> 哈希是怎么回事?
# 因为在内部会将值进行哈希算法并得到一个数值（对应内存地址），以后用于快速查找。

# 3. 特殊情况
# info = {0, 2, 3, 4, False, "国风", None, (1, 2, 3)}
# print(info)

# info = {
#     1:'alex',
#     True:'oldboy'
# }
# print(info)
```

2. 内存相关

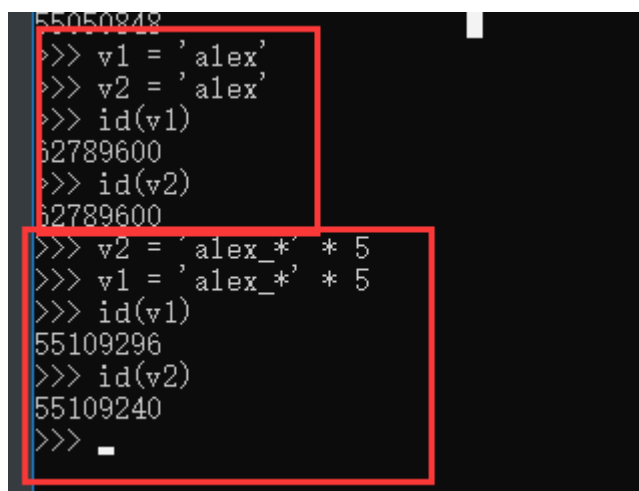
- 示例一

```
v1 = [11,22,33]
v2 = [11,22,33]

v1 = 666
v2 = 666

v1 = "asdf"
v2 = "asdf"

# 按理 v1 和 v2 应该是不同的内存地址。特殊：
1. 整型： -5 ~ 256
2. 字符串: "alex",'asfasd asdf asdf d_asdf ' ----"f_*" * 3 - 重新开辟内存。
```



```
>>> v1 = 'alex'
>>> v2 = 'alex'
>>> id(v1)
62789600
>>> id(v2)
62789600

>>> v2 = 'alex_*' * 5
>>> v1 = 'alex_*' * 5
>>> id(v1)
55109296
>>> id(v2)
55109240
>>> _
```

- 示例二：

```
v1 = [11,22,33,44]
v1 = [11,22,33]
```

- 示例三:

```
v1 = [11,22,33]
v2 = v1

# 练习1 (内部修改)
v1 = [11,22,33]
v2 = v1
v1.append(666)
print(v2) # 含 666

# 练习2: (赋值)
v1 = [11,22,33]
v2 = v1
v1 = [1,2,3,4]
print(v2)

# 练习3: (重新赋值)
v1 = 'alex'
v2 = v1
v1 = 'oldboy'
print(v2)
```

- 示例四

```
v = [1,2,3]
values = [11,22,v]

# 练习1:
"""
v.append(9)
print(values) # [11,22,[1,2,3,9]]
"""

# 练习2:
"""
values[2].append(999)
print(v) # [1, 2, 3, 999]
"""

# 练习3:
"""
v = 999
print(values) # [11, 22, [1, 2, 3]]
"""

# 练习4:
values[2] = 666
print(v) # [1, 2, 3]
```

- 示例五

```
v1 = [1,2]
v2 = [2,3]

v3 = [11,22,v1,v2,v1]
```

- 查看内存地址

```
"""
v1 = [1,2,3]
v2 = v1
v1.append(999)
print(v1,v2)
print(id(v1),id(v2))
"""

"""
v1 = [1,2,3]
v2 = v1
print(id(v1),id(v2))
v1 = 999
print(id(v1),id(v2))
"""
```

- 问题： == 和 is 有什么区别？
 - == 用于比较值是否相等。
 - is 用于比较内存地址是否相等。

```
>>>
>>> v1 = [1, 2, 3, 4]
>>> v2 = [1, 2, 3, 5, 6, 7]
>>> v1 == v2
False
>>> v1 is v2
False

>>> v1 = [1, 2]
>>> v2 = [1, 2]
>>> v1 == v2
True
>>> v1 is v2
False

>>> v1 = [1, 2, 3]
>>> v2 = v1
>>> v1 == v2
True
>>> v1 is v2
True
>>>
```

```
>>>
>>> v1 = 10
>>> v2 = 10
>>> v1 == v2
True
>>> v1 is v2
True
>>> v1 = 10000
>>> v2 = 10000
>>> id(v1)
55050864
>>> id(v2)
55050912
>>>
```

走你今儿

- 列表
 - reverse
 - sort
- 字典
 - get (*)
 - update
- 集合
 - add
 - discard
 - update
 - intersection (*)
 - union
 - difference
 -
- 特殊：
 - 嵌套：集合/字典的key
 - 空：None
 - 空集合：...
- id
- type
- 嵌套的应用： (*)
 - 赋值
 - 修改内部元素：列表/字典/集合

