

魔术方法 ***

容器相关方法

方法	意义
<code>__len__</code>	内建函数len(), 返回对象的长度 (>=0的整数), 如果把对象当做容器类型看, 就如同list或者dict。 bool()函数调用的时候, 如果没有 <code>__bool__()</code> 方法, 则会看 <code>__len__()</code> 方法是否存在, 存在返回非0为真
<code>__iter__</code>	迭代容器时, 调用, 返回一个 新的迭代器对象
<code>__contains__</code>	in 成员运算符, 没有实现, 就调用 <code>__iter__</code> 方法遍历
<code>__getitem__</code>	实现self[key]访问。序列对象, key接受整数为索引, 或者切片。对于set和dict, key为hashable。key不存在引发KeyError异常
<code>__setitem__</code>	和 <code>__getitem__</code> 的访问类似, 是设置值的方法
<code>__missing__</code>	字典或其子类使用 <code>__getitem__()</code> 调用时, key不存在执行该方法

```
1 class A(dict):
2     def __missing__(self, key):
3         print('Missing key : ', key)
4         return 0
5
6 a = A()
7 print(a['k'])
```

思考

为什么空字典、空字符串、空元组、空集合、空列表等可以等效为False?

应用

设计一个购物车, 能够方便增加商品, 能够方便的遍历

```
1 class Cart:
2     def __init__(self):
3         self.items = []
4
5     def __len__(self):
6         return len(self.items)
7
8     def additem(self, item):
9         self.items.append(item)
10
11     def __iter__(self):
12         # yield from self.items
13         return iter(self.items)
14
15     def __getitem__(self, index): # 索引访问
16         return self.items[index]
```

```
17
18     def __setitem__(self, key, value): # 索引赋值
19         self.items[key] = value
20
21     def __str__(self):
22         return str(self.items)
23
24     def __add__(self, other): # +
25         self.items.append(other)
26         return self
27
28 cart = Cart()
29 # 长度、bool
30 print(cart, bool(cart), len(cart))
31
32 cart.additem(1)
33 cart.additem('abc')
34 cart.additem(3)
35
36 # 长度、bool
37 print(cart, bool(cart), len(cart))
38
39 # 迭代
40 for x in cart:
41     print(x)
42
43 # in
44 print(3 in cart)
45 print(2 in cart)
46
47 # 索引操作
48 print(cart[1])
49 cart[1] = 'xyz'
50 print(cart)
51
52 # 链式编程实现加法
53 print(cart + 4 + 5 + 6)
54 print(cart.__add__(17).__add__(18))
```