链滴

python 笔记 3

作者：zhuhonglin
原文链接：https://ld246.com/article/1494639370813
来源网站：链滴

list and dict

关于list、dict的删除：

```
d = [1,2,3,4,5]
x = d.pop()
y = d.pop(0)
del d[1]
print(x,y,d)
'''
输出：
5 1 [2, 4]
'''
```

可以使用pop和del来删除。pop是类自带的函数，删除成功返回pop的值。del是全局函数，不返回值

一些特殊方法

```
import math
class A(object):
    __slots__ = ('x', 'y', 'z')
    def __init__(self, x, y, z):
        self.x = x
        self.y = y
        self.z = z
    def __del__(self):
        #这个是析构函数
        pass
    def __str__(self):
```

```
        return "(%s,%s,%s)" % (self.x, self.y, self.z)
    def  __getitem__ (self, index):
        index %= 3
        arr = {0: self.x, 1: self.y, 2: self.z }
        return arr[index];
    def __setitem__(self, index, value):
        index %= 3
        arr = {0: 'self.x = value', 1: 'self.y = value', 2: 'self.z = value'}
        exec(arr[index])
        #上面这个函数的写法我感觉不舒服，但我目前想不到更好更简洁的
    def __eq__(self, s):
        l_self = pow(self.x, 2) + pow(self.y, 2) + pow(self.z, 2)
        l_s = pow(s.x, 2) + pow(s.y, 2) + pow(s.z, 2)
        return l_self == l_s
        #类似的有lt(less than),gt,ne,le(less than or equal),ge
    def __add__(self, s):
        x = self.x + s.x
        y = self.y + s.y
        z = self.z + s.z
        return A(x,y,z)
        #类似的有sub,mul,truediv,floordiv..等，一个道理
# class A 结束
def main():
    ins = A(1,10,100)       #__init__
    print(ins)              #__str__
    print(ins[0])           #__getitem__
    ins[0] = 50             #__setitem__
```

```
    print(ins[0])
    n_ins = A(50,10,100)
    print(ins == n_ins)     #__eq__
    nn_ins = ins + n_ins    #__add__
    print(nn_ins)
if __name__ == '__main__':
    main()
'''
输出>>
(1,10,100)
1
50
True
(100,20,200)
'''
```

上面是一些常用的特殊方法，其他的应该找文档就行。一个道理。

关于序列化(json)

简单的序列化

```
import json
def SerializeToFile(d, filename):
    with open(filename, 'w') as f:
        json.dump(d, f)
def UnSerializeFromFile(filename):
    with open(filename, 'r') as f:
        info = json.load(f)
```

```
    return info
def main():
    d = {'name': 'Bob', 'age': 20, 'score': 88}
    filename = 'dump.txt'
    SerializeToFile(d, filename)
    print(UnSerializeFromFile(filename))
if __name__ == '__main__':
    main()
'''
输出>>
{'name': 'Bob', 'age': 20, 'score': 88}
'''
```

使用load和dump，用于json和python各个类型的转换

定制序列化

```
import json
class A(object):
    def __init__(self, name, age, score):
        self.name = name
        self.age = age
        self.score = score
    def __str__(self):
        return 'Name:%s Age:%s Score:%s' % (self.name, self.age, self.score)
def SerializeToFile(d, filename):
    with open(filename, 'w') as f:
        json.dump(d, f, default= lambda obj: obj.__dict__)
def UnSerializeFromFile(filename):
```

```
    with open(filename, 'r') as f:
        info = json.load(f, object_hook=DictToObject)
    return info
def DictToObject(d):
    return A(d['name'], d['age'], d['score'])
def main():
    d = A('Bob', 20, 88)
    filename = 'dump.txt'
    SerializeToFile(d, filename)
    obj = UnSerializeFromFile(filename)
    print("%r\n%s" % (obj, obj))
if __name__ == '__main__':
    main()
'''
输出>>
<__main__.A object at 0x0000020438E193C8>
Name:Bob Age:20 Score:88
'''
```

序列化时写default，参数是一个函数，说明对象转换dict(也可以用别的)的方法，用lambda简单，且可以覆盖任意类实例。

反序列化写object_hook，本例中，参数需要说明dict转换对象的方法，因为类不统一，所以转换的法不统一，使用DictToObject函数封装。