**Daily Report**

29th January, 2018

1. **切片（Slice）操作符**

操作如：

>>> L[0:3]

['Adam', 'Lisa', 'Bart']

如果第一个索引是0，还可以省略：

>>> L[:3]

['Adam', 'Lisa', 'Bart']

只用一个 : ，表示从头到尾：

>>> L[:]

['Adam', 'Lisa', 'Bart', 'Paul']

切片操作还可以指定第三个参数：

>>> L[::2]

['Adam', 'Bart']

第三个参数表示每N个取一个，上面的 L[::2] 会每两个元素取出一个来，也就是隔一个取一个。

把list换成tuple，切片操作完全相同，只是切片的结果也变成了tuple。

对于list，既然Python支持L[-1]取倒数第一个元素，那么它同样支持倒数切片

1. **Range（）数列**

range()函数可以创建一个数列：

>>> range(1, 101)

[1, 2, 3, ..., 100]

1. **Upper()**

upper() 可以把字符变成大写字母

1. **索引迭代**

想在 for 循环中拿到索引，方法是使用 enumerate() 函数

>>> L = ['Adam', 'Lisa', 'Bart', 'Paul']

>>> for index, name in enumerate(L):

... print index, '-', name

...

0 - Adam

1 - Lisa

2 - Bart

3 - Paul

1. **Zip（）函数**

zip()函数可以把两个 list 变成一个 list：

>>> zip([10, 20, 30], ['A', 'B', 'C'])

[(10, 'A'), (20, 'B'), (30, 'C')]

1. **获取dict对象的value**

dict 对象有一个 values() 方法，这个方法把dict转换成一个包含所有value的list，这样，我们迭代的就是 dict的每一个 value

还有一个 itervalues() 方法，用 itervalues() 方法替代 values() 方法，迭代效果完全一样

1. **Values（）和itervalues（）的区别**

1. values() 方法实际上把一个 dict 转换成了包含 value 的list。

2. 但是 itervalues() 方法不会转换，它会在迭代过程中依次从 dict 中取出 value，所以 itervalues() 方法比 values() 方法节省了生成 list 所需的内存。

3. 打印 itervalues() 发现它返回一个 <dictionary-valueiterator> 对象，这说明在Python中，for 循环可作用的迭代对象远不止 list，tuple，str，unicode，dict等，任何可迭代对象都可以作用于for循环，而内部如何迭代我们通常并不用关心。

1. **Items（）**

items() 方法把dict对象转换成了包含tuple的list，我们对这个list进行迭代，可以同时获得key和value

1. **isinstance(x, str) 可以判断变量 x 是否是字符串**
2. **Random（）函数示例**

import random

for i in range(10):

x = random.random()

print(x)

1. **Randint（）函数**

需要输入上下限，并且能输出上下限内（包含上下限）的整数

>>> random.randint(5, 10)

5

>>> random.randint(5, 10)

9

1. **Choice（）函数**

通过choice函数选择list等中的一个结果

>>> t = [1, 2, 3]

>>> random.choice(t)

2

>>> random.choice(t)

3