**Daily Report**

26th January, 2018

1. **Len()函数**

len() 函数可以计算任意集合的大小

1. **dict作用**

将两个量联结起来，组成一个查找表。例如：

d = {

'Adam': 95,

'Lisa': 85,

'Bart': 59

}

1. **访问dict**

使用 d[key] 的形式来查找对应的 value

通过 key 访问 dict 的value，只要 key 存在，dict就返回对应的value。如果key不存在，会直接报错：KeyError

要避免 KeyError 发生，有两个办法

1. 先判断一下 key 是否存在，用 in 操作符。例如：

if 'Paul' in d:

print d['Paul']

1. 使用dict本身提供的一个 get 方法，在Key不存在的时候，返回None。例如：

>>> print d.get('Bart')

59

>>> print d.get('Paul')

None

1. **Dict的特点**
2. 查找速度快，占用内存大
3. Key不能重复
4. 存储的内容无序
5. 作为 key 的元素必须不可变，Python的基本类型如字符串、整数、浮点数都是不可变的，都可以作为 key。但是list是可变的，就不能作为 key
6. **更新dict**

Dict是可变的，可以将新的值赋予给dict。赋值语句如：

>>> d['Paul'] = 72

如果 key 已经存在，则赋值会用新的 value 替换掉原来的 value

1. **遍历dict**

可以通过 for 循环实现

1. **Set**

set 持有一系列元素，这一点和 list 很像，但是set的元素没有重复，而且是无序的，这点和 dict 的 key很像

创建 set 的方式是调用 set() 并传入一个 list，list的元素将作为set的元素：

>>> s = set(['A', 'B', 'C'])

>>> s = set(['A', 'B', 'C', 'C'])

>>> print s

set(['A', 'C', 'B'])

>>> len(s)

3

1. **访问set**

访问 set中的某个元素实际上就是判断一个元素是否在set中，可以使用in操作符进行判断

1. **更新set**

添加元素时，用set的add()方法

删除set中的元素时，用set的remove()方法