# 1.第一个大坑

Python2中接收输入用x=raw\_input(“please input name”)

Python3中接收输入用x=input(“please input name”)

# print()返回值为None

# for 循环后面的变量是全局变量

**for** takes **in** range(1,10):  
 print(**'takes'**,takes)  
print(**'globle takes='**,takes)

结果\*\*\*\*\*\*\*\*\*\*\*\*\*\*

takes 7

takes 8

takes 9

globle takes= 9

spam = ['cat', 'bat', 'rat', 'elephant']

# 4.列表加、乘字符串

>>> spam = 'Hello'

>>> spam += ' world!'

>>> spam

'Hello world!'

列表加一个字符串，相当于往后边添加一个列表元素

>>> bacon = ['Zophie']

>>> bacon \*= 3

>>> bacon

['Zophie', 'Zophie', 'Zophie']

列表乘以数字相当于复制

# 列表的insert方法

insert() 方法 的 第一个 参数 是 新 值 的 下标， 第二个 参数 是要 插入 的 新 值。

>>> spam = ['cat', 'dog', 'bat']

>>> spam. insert( 1, 'chicken')

>>> spam

['cat', 'chicken', 'dog', 'bat']

# 列表可以用的方法

Spam=[‘ants’,’cats’,’dogs’]

Spam.append(‘badgers’)

Spam.insert(1,’chicken’)

Spam.remove(‘cat’)

Spam.sort() #按照ASCII

Spam.sort(key=str.lower) #统一按照小写字母排序

# 一行代码太长需要换行，在行末加一个\,

# 下一行缩不缩进都行，都会被看成是上一行的延续

print(' Four score and seven ' + \

'years ago...')

# 概念知识点，其实字符串也可以看成是单个文本的列表，当然也可以切片与及用于in与not in的操作，但字符串不能改变，例如不能给name[2]=’the’会报错

>>> name = 'Zophie'

>>> name[ 0]

'Z'

>>> name[- 2]

'i'

>>> name[ 0: 4]

'Zoph'

>>> 'Zo' in name

True

>>> 'z' in name

False

>>> 'p' not in name

False

>>> for i in name: print('\* \* \* ' + i + ' \* \* \*')

\* \* \* Z \* \* \*

\* \* \* o \* \* \*

\* \* \* p \* \* \*

\* \* \* h \* \* \*

\* \* \* i \* \* \*

\* \* \* e \* \* \*

# 元组Tuple的值不能被修改，添加或者删

注意：当元组只有一个值里，需要加**逗号**\*\*\*

>>> type((' hello',))

< class 'tuple'

> >>> type((' hello'))

< class 'str'>

# 类型转换

>>> tuple([' cat', 'dog', 5])

('cat', 'dog', 5)

>>> list((' cat', 'dog', 5))

['cat', 'dog', 5]

>>> list(' hello')

['h', 'e', 'l', 'l', 'o']

# 11.列表变量附值，把一个列表变量的值给另一个列表变量，只是把列表变量对应ID给了另一个列表变量，改变一个列表变量里面的值，两个列表变量都会变

列表、字典（值可变）python存的是引用（类似id），

字符串、 整型 或 元 组，（值不可变） Python 变量 就 保存 值 本身。

# 如何复制列表跟字典呢？

（import copy 然后用cheese=copy.copy(spam)）

>>> import copy

>>> spam = ['A', 'B', 'C', 'D']

>>> cheese = copy. copy( spam)

>>> cheese[ 1] = 42

>>> spam ['A', 'B', 'C', 'D']

>>> cheese ['A', 42, 'C', 'D']

# 字典，概念知识点（1.无序2.整形数值可为键）

>>> spam = ['cats', 'dogs', 'moose']

>>> bacon = ['dogs', 'moose', 'cats']

>>> spam == bacon False

>>> eggs = {'name': 'Zophie', 'species': 'cat', 'age': '8'}

>>> ham = {'species': 'cat', 'age': '8', 'name': 'Zophie'}

>>> eggs == ham True

# 字典的方法keys(),values()和items()--(返回的是元组)

数据 类型（ 分别 是 dict\_ keys、 dict\_ values 和 dict\_ items） 可以 用于 for 循环。

>>> spam = {'color': 'red', 'age': 42}

>>> for v in spam. values(): print( v)

red

42

这里， for 循环 迭代 了 spam 字典 中的 每个 值。 for 循环 也可以 迭代 每个 键， 或者 键- 值 对：

>>> for k in spam. keys(): print( k)

Color

age

>>> for i in spam. items(): print( i)

('color', 'red')

('age', 42)

## 遍历字典也可以用多重赋值的技巧

>>> spam = {'color': 'red', 'age': 42}

>>> for k, v in spam. items():

print(' Key: ' + k + ' Value: ' + str( v))

Key: age Value: 42

Key: color Value: red

## 技巧1：因为字典中如果没有对应的键值会报错，所以可以用get()方法，并给定默认值的方法

>>> picnicItems = {'apples': 5, 'cups': 2}

>>> 'I am bringing ' + str( picnicItems. get(' cups', 0)) + ' cups.'

'I am bringing 2 cups.'

>>> 'I am bringing ' + str( picnicItems. get(' eggs', 0)) + ' eggs.'

'I am bringing 0 eggs.'

## 技巧2：试探一个字典中有没有某一个键值，如果有则不做任何操作，如果没有，就给这个键附一个值，用一个方法

## spam. setdefault(' color', 'white')

>>> spam = {'name': 'Pooka', 'age': 5}

>>> spam. setdefault(' color', 'black')

'black'

>>> spam

{'color': 'black', 'age': 5, 'name': 'Pooka'}

>>> spam. setdefault(' color', 'white')

'black'

>>> spam

{'color': 'black', 'age': 5, 'name': 'Pooka'}

## 可以用到循环中，给变量附初始值，因为没有初始化，所以只执行一次

for character in message:

count. setdefault( character, 0)

count[ character] = count[ character] + 1