目录

[Python入门 2](#_Toc534700998)

[1 Why Python? 2](#_Toc534700999)

[2 语法 2](#_Toc534701000)

[3 变量 3](#_Toc534701001)

[4 数字 3](#_Toc534701002)

[5 强制转换 4](#_Toc534701003)

[6 字符串 4](#_Toc534701004)

[7 运算符 5](#_Toc534701005)

[8 列表List 7](#_Toc534701006)

[9 元组Tuple 10](#_Toc534701007)

[10 集合Set 10](#_Toc534701008)

[11 字典Dictionary 11](#_Toc534701009)

[12 If…Else 11](#_Toc534701010)

[13 While Loops 12](#_Toc534701011)

[14 For Loops 12](#_Toc534701012)

[15 Recursion 17](#_Toc534701013)

[16 Functions 17](#_Toc534701014)

[17 项目 18](#_Toc534701015)

Python入门

<https://www.w3schools.com/python/python_lists.asp>

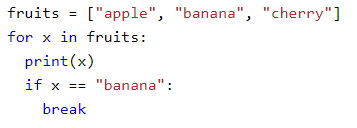
# Why Python?

1. 比起其他编程语言，Python更容易上手
2. Python功能强大，不需要考虑如何管理你的程序使用的内存等底层细节，并且，Python有很丰富的库，包括官方的，也有第三方开发的。
3. 在职场中，使用Python工作的主要有这样几类人：
   1. 网站后端程序员：使用Python搭建网站、后台服务会比较容易维护
   2. 自动化运维：越来越多的运维开始倾向于自动化，批量处理大量的运维任务。
   3. 数据分析师：Python能快速开发的特性可以让你迅速验证你的想法，而不是浪费在程序本身上，帮你节省时间
   4. 游戏开发者：一般是作为游戏脚本内嵌在游戏中
   5. 自动化测试
   6. 用爬虫爬取或处理信息
   7. 使用混合编译的方式使用c/c++/Java等语言的库

# 语法

1. 缩进

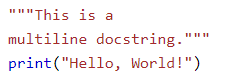
Python不用括号来表示语种块，也不用开始或结束标志符来表示，而是靠缩进，用四个空格表示（在PyCharm中，可以定义**Tab**为四个空格）。



1. 注释

单行注释：采用**#**开头

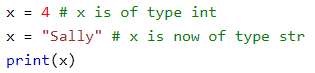
多行注释（docString）：采用三个双引号**”””**开头，三个双引号结束

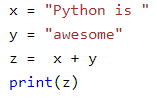
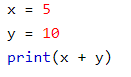
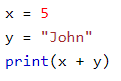
1. 防止中文注释报错
   1. 在文件开头加一行注释：
   2. 也可在设置里找到”File Encodings”设置为**UTF-8**
2. 过长的代码段可以使用**\**来进行换行，属于一行的代码同时会有一个缩进代表是一行的。

# 变量

Python中不需要声明变量，当赋值时即创建了变量。并且在程序中，可改变变量类型。



经常使用**print**输出变量。符号**+**可以连接两个字符串，可以把两个数相加，但不能将一个数与一个字符串相加。

🗶

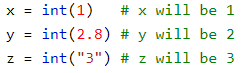
# 数字

主要有两种数字类型：**int**和**float**，可用**type()**函数来查看变量类型

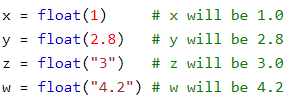
  

# 强制转换

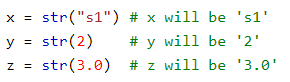
1. 转为整型：int()



1. 转为浮点型：float()



1. 转为字符串：str()



# 字符串

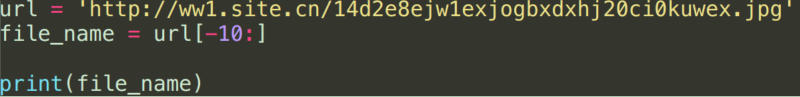
1. 字符串可用单引号**’**或双引号**”**表示，’hello’和”hello”相同
2. 字符串作为数组存在。

下标从0开始。

b[2:5]表示下标2至5（不包括）的字符，即表示b[2], b[3], b[4]的字符。

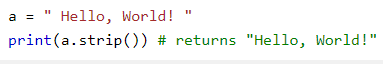
 

输出：e 输出：llo



输出：

1. strip()：移除开始和末尾的空格



1. len(): 字符串长度



1. lower()：转小写字母



1. upper()：转大写字母



1. replace(): 用另一字符串替代一个指定的字符串



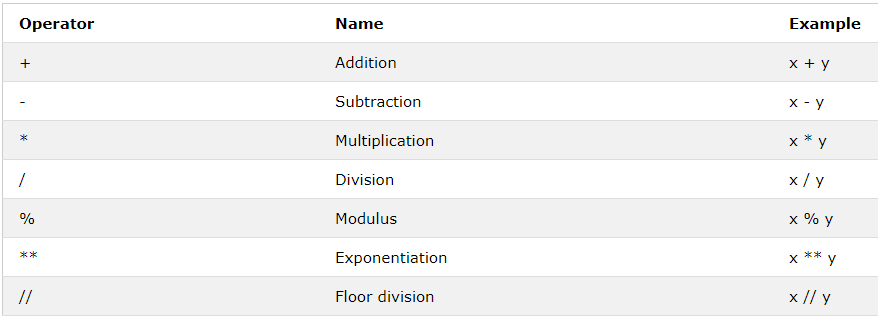
输出：heaao

1. split(): 分开字符串

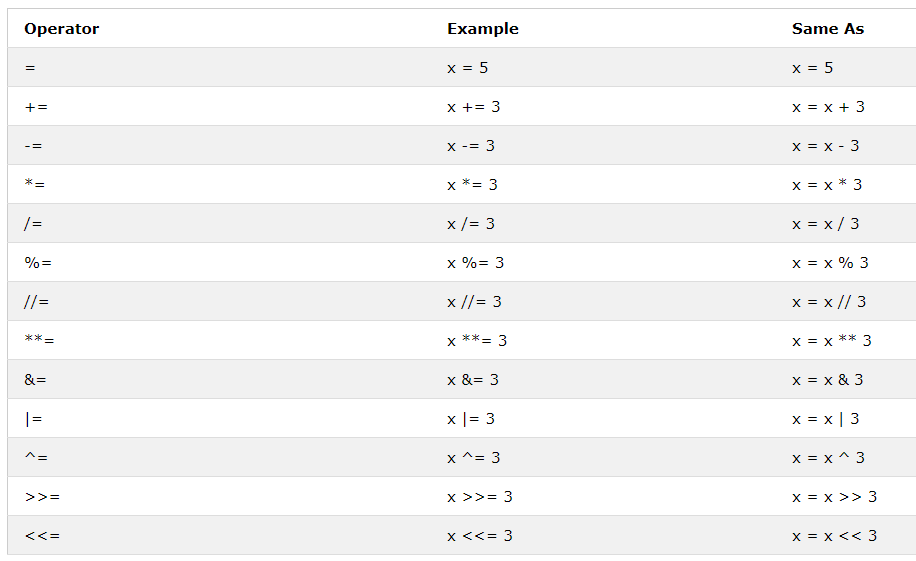


# 运算符

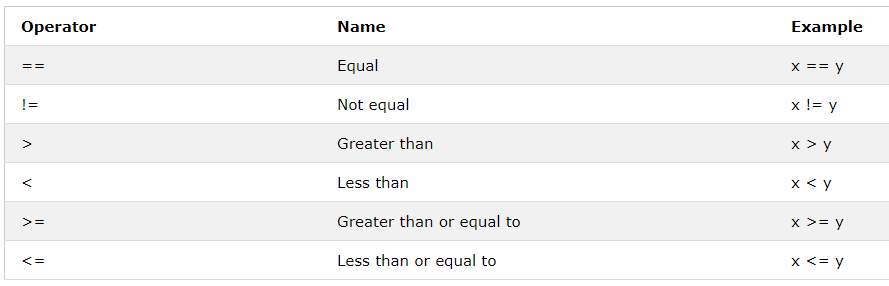
1. **算术运算符**



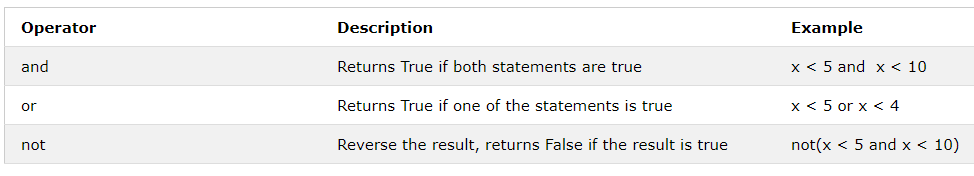
1. **赋值运算符**



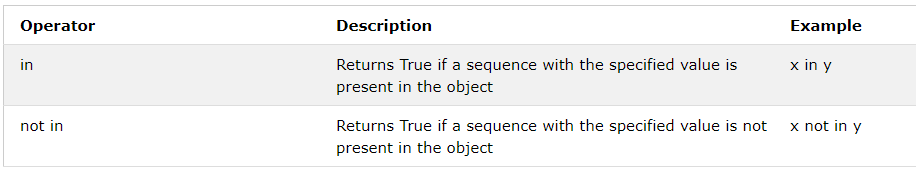
1. **比较运算符**

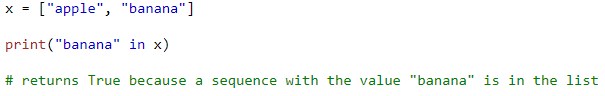


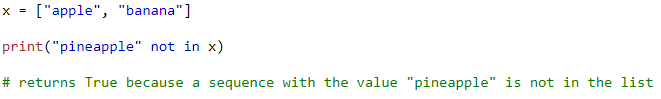
1. **逻辑运算符**



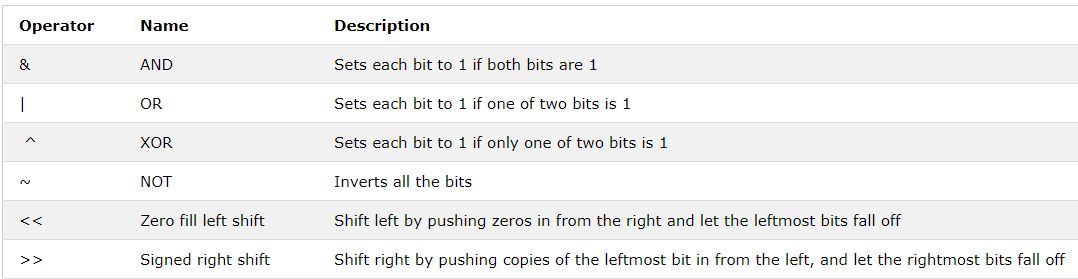
1. **属于运算符**







1. **位运算**



# 列表List

1. **特征：**
   1. 列表中每一元素都是可变的
   2. 列表中的元素是有序的，每一个元素都有一个位置
   3. 列表可以容纳Python中的任何对象
2. **创建List**



1. **使用list() Constructor创建List**



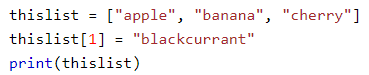
输出：['apple', 'banana', 'cherry']

1. **得到List中元素**

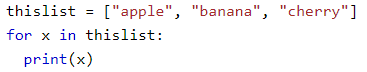


输出：banana

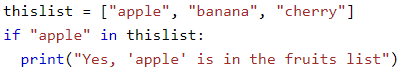
1. **改变List中元素值**



1. **通过for循环遍历List中元素**



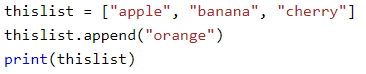
1. **检验是否有某个元素**



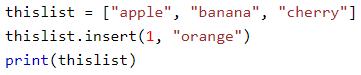
1. **List的长度**



1. **增加元素**

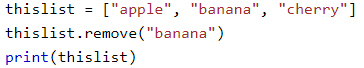


在后面增加元素

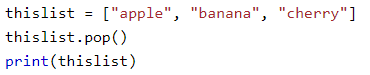


输出：['apple', 'orange', 'banana', 'cherry']

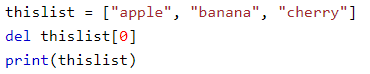
1. **移除元素**



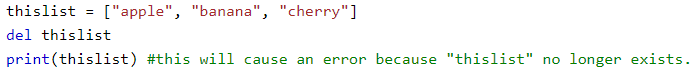
移除指定元素



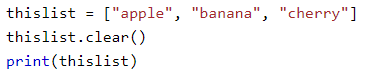
移除指定编号的元素或最后一个元素（没有指定编号）



移除指定编号的元素

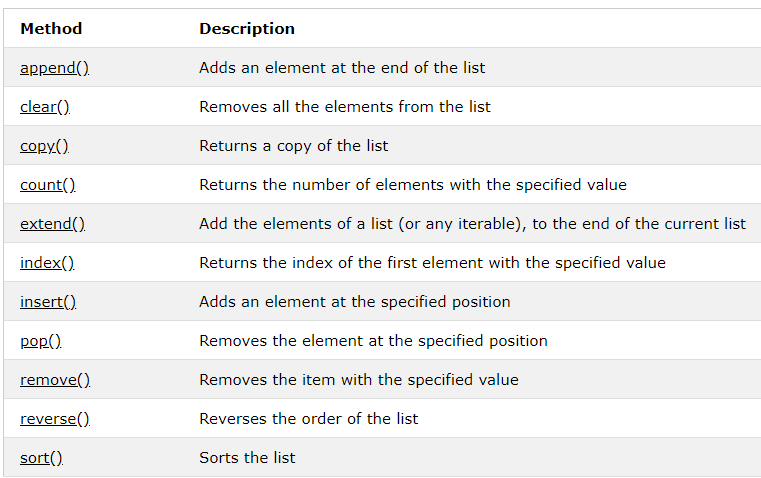


删除列表本身



清空列表

1. **List的方法**



# 元组Tuple

1. Tuple定义：一组有序但不可改变其值的元素集合，稳固版的列表。
2. 创建Tuple



输出：('apple', 'banana', 'cherry')

# 集合Set

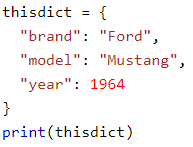
1. 定义：一组无序并且没有编号的元素集合，接近于数学上集合的概念。
2. 创建集合：



输出：{'apple', 'banana', 'cherry'}

# 字典Dictionary

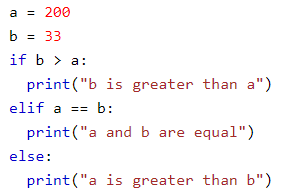
1. **特征：**
   1. 字典中的数据必须是以键值对的形式出现
   2. 逻辑上讲，键是不能重复的，而值可以重复
   3. 字典中的键（key）是不可变的，也就是无法修改的；而值（value）是可变的，可修改的，可以是任何对象。
2. **创建Dictionary**



输出：{'brand': 'Ford', 'model': 'Mustang', 'year': 1964}

# If…Else

1. **注意缩进**



1. **简易格式**

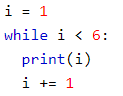




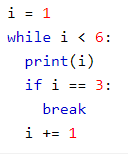


# While Loops

1. **While**

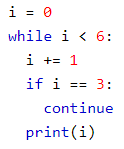


1. **break**



输出：1,2,3

1. **continue**

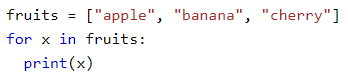


输出：1,2,4,5,6

# For Loops

A for loop is used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string).

1. **Print each fruit in a fruit list:**



输出：

1. **Loop through the letters in the word "banana":**



输出：

1. **range()函数：starting from 0 by default, and increments by 1 (by default),** **and ends at a specified number**



输出：

1. **range(): Using the start parameter**



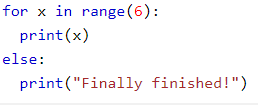
输出：

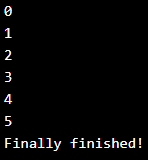
1. **Increment the sequence with 3 (default is 1):**



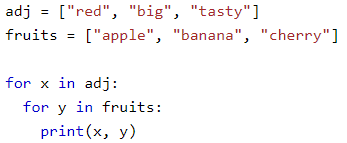
输出：

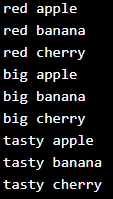
1. **Else in For Loop**



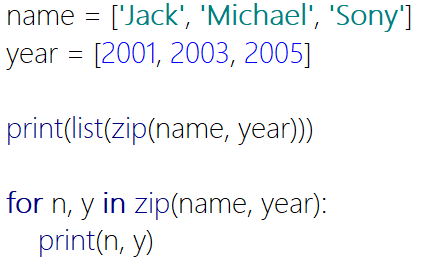
输出：

1. **Nested Loops**

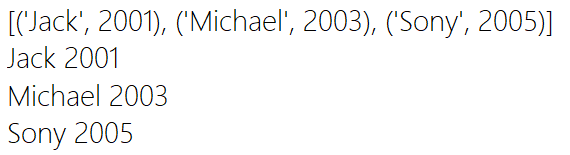


输出：

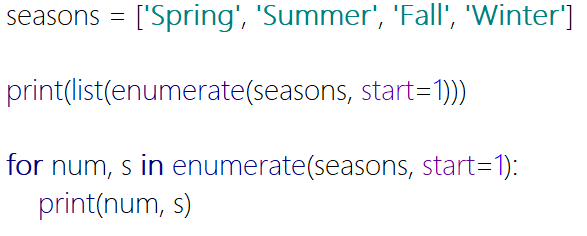
1. **zip()函数**



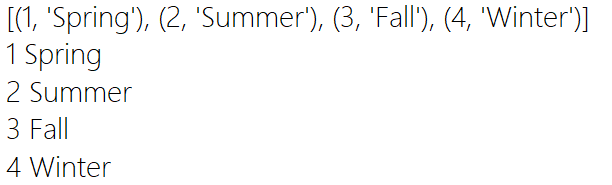
输出：



1. **enumerate()函数**



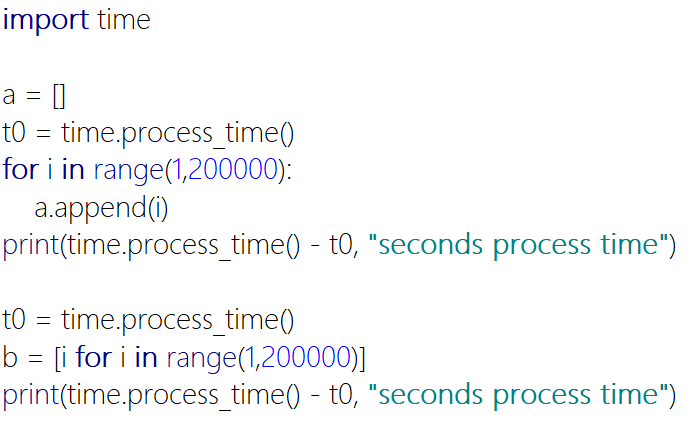
输出：



1. **推导式-列表解析式**

列表解析式不仅方便，并且在执行效率上要高于普通方。

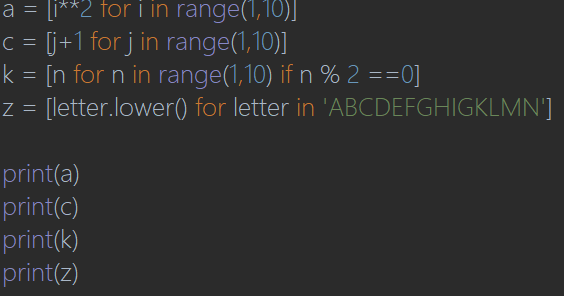
* 1. 例1



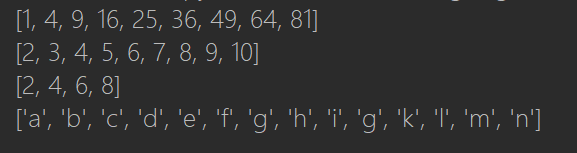
输出：



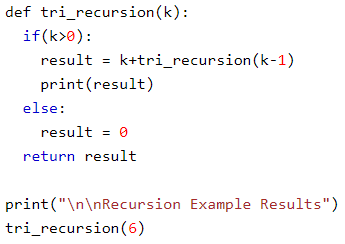
* 1. 例2

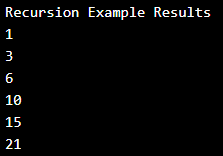


输出：



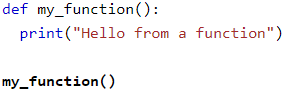
# Recursion



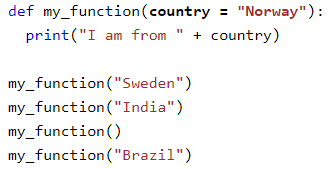
输出：

# Functions

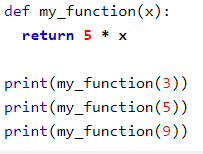
1. Creating and calling a Function



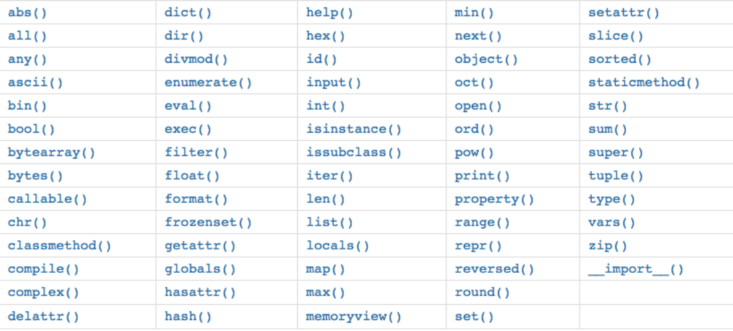
1. Default Parameter Value



1. Return Values

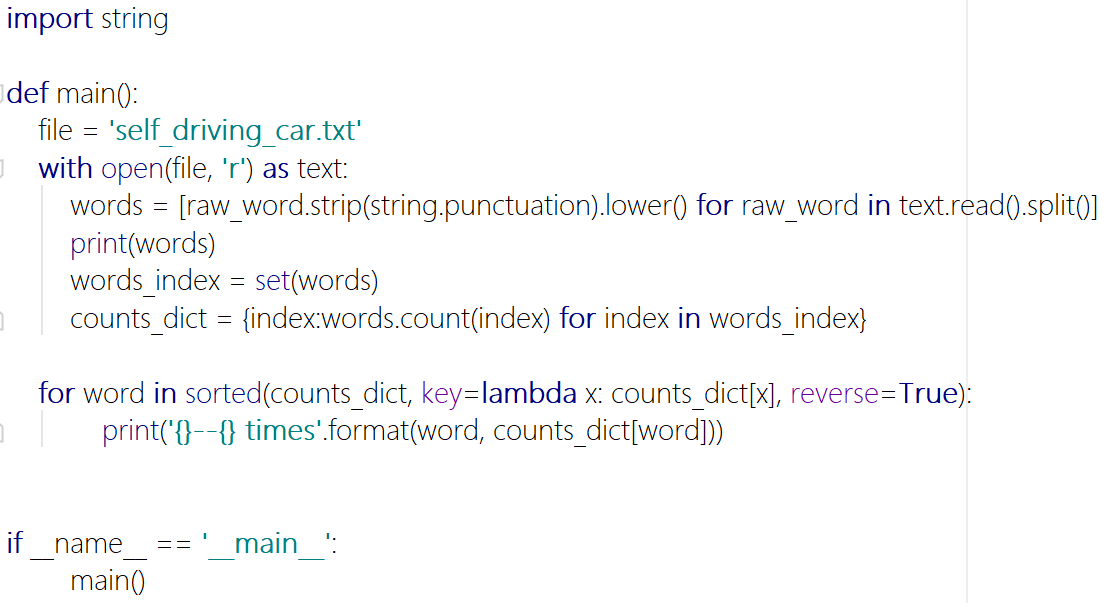


1. Built-in Functions

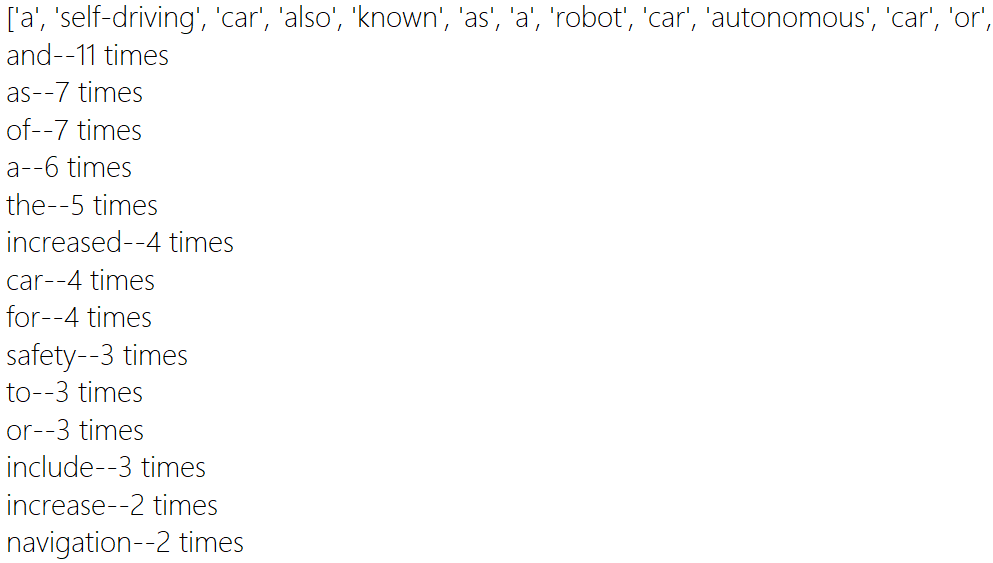


# 项目

从一个txt文本中读取单词，统计每个单词出现的频率，按词频从大到小排列。



输出：



说明：

1. 第1行，引入模块string，其中string.punctuation表示：!”#$%&’()\*+,-./:;<=>?@[\]^\_`{|}~
2. 第4行，在文字中去掉了标点符号，并把大写的字母转化成小写
3. 第5行，将列表用set函数转换成集合，自动去除了其中所有重复的元素
4. 第6行，创建了一个以单词为键(key)出现频率为值(value)的字典
5. 第7~8行，打印整理后的字典，其中叫做lambda表达式，可以理解为以字典中的值为排序的函数。