Python快速入门

嵩 天

Python基本编程解析(下)

嵩 天

Python基本编程解析(下)

Python快速入门

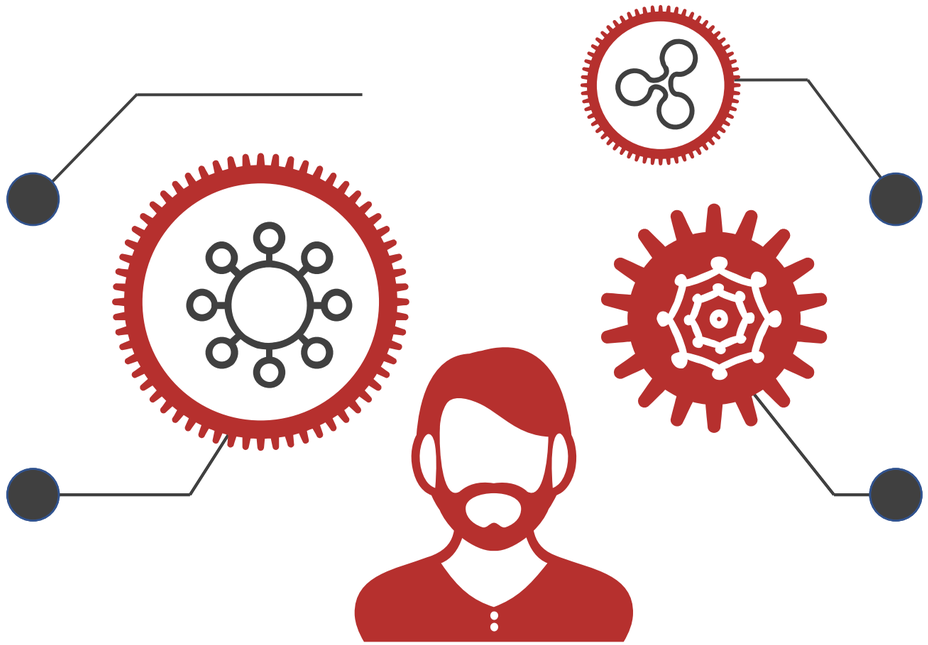
单元开篇

Python123

单元开篇

(1) import的三种用法

(2) jieba中文分词库



(3) 计算生态编程

(4) “中文词语统计”代码分析

Python基本编程解析(下)

单元开篇

目的：概要了解Python编程的基本知识
学习编写10行的Python代码，开启编程之旅

1. 知道 Python编程的基本知识
2. 理解 Python中文词语统计程序
3. 能独立编写 一个10行左右类似功能的Python程序

Python基本编程解析(下)

Python快速入门

import的三种用法

Python123

代码演示

```
#WordsCount.py
import jieba
f = open("2018年一号文件.txt", "r", encoding="utf-8")
txt = f.read()
f.close()
ls = jieba.lcut(txt)
d = {}
for w in ls:
    d[w] = d.get(w, 0) + 1
for k in d:
    if d[k] >= 50 and k != "\n":
        print('"{}"出现{}次'.format(k, d[k]))
```

引入外部功能库

import的三种用法

import：引用功能库的保留字，有三种使用方法

方法一：

```
import <库名>

<库名>.<函数名>(<函数参数>)
```

或

```
import <库名1>, <库名2>
```

代码演示


```
#WordsCount.py
import jieba
f = open("2018年一号文件.txt", "r", encoding="utf-8")
txt = f.read()
f.close()
ls = jieba.lcut(txt)
d = {}
for w in ls:
    d[w] = d.get(w, 0) + 1
for k in d:
    if d[k] >= 50 and k != "\n":
        print('"{}"出现{}次'.format(k, d[k]))
```

import的三种用法

import：引用功能库的保留字，有三种使用方法

方法二：

```
from <库名> import <函数名
```

或 >

```
from <库名> import *
<函数名>(<函数参数>)
```

代码演示

```
#WordsCount.py
from jieba import lcut
f = open("2018年一号文件.txt", "r", encoding="utf-8")
txt = f.read()
f.close()
ls = lcut(txt)
d = {}
for w in ls:
    d[w] = d.get(w, 0) + 1
for k in d:
    if d[k] >= 50 and k != "\n":
        print('"{}"出现{}次'.format(k, d[k]))
```

import的三种用法

import：引用功能库的保留字，有三种使用方法

方法三：

```
import <库名> as <库别名>
<库别名>.<函数名>(<函数参数>)
```

import三种用法比较

方法一：

```
import <库名>
<库名>.<函数名>(<函数参数>)
```

适合简单库名情况

方法二：

```
from <库名> import *
<函数名>(<函数参数>)
```

混合命名空间，适合极少库使用情况

方法三：

```
import <库名> as <库别名>
<库别名>.<函数名>(<函数参数
>)
```

适合复杂库名情况

代码演示

```
#WordsCount.py
import jieba as ja
f = open("2018年一号文件.txt", "r", encoding="utf-8")
txt = f.read()
f.close()
ls = ja.lcut(txt)
d = {}
for w in ls:
    d[w] = d.get(w, 0) + 1
for k in d:
    if d[k] >= 50 and k != "\n":
        print('"{}"出现{}次'.format(k, d[k]))
```

Python快速入门

jieba中文分词库

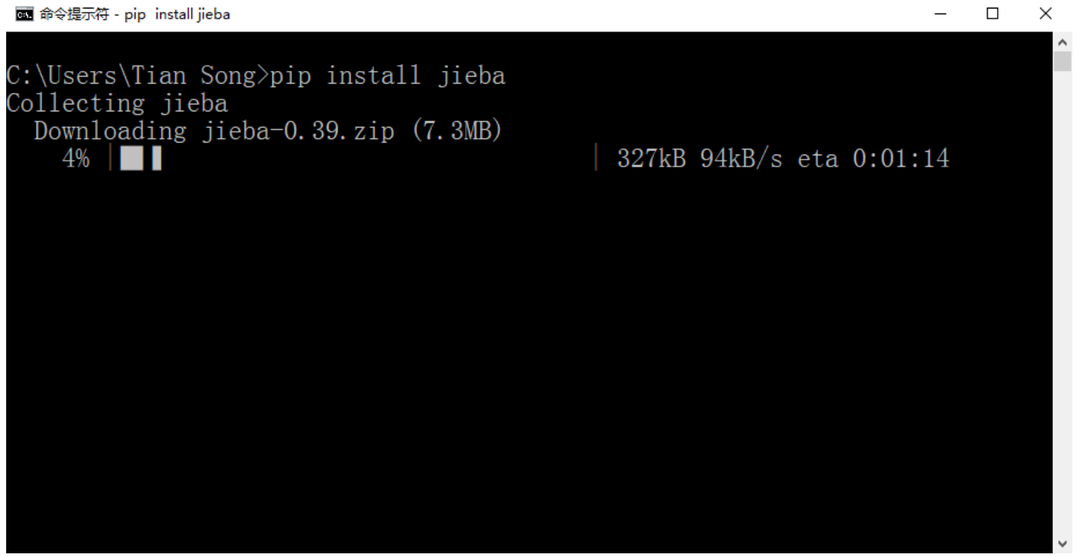
jieba库概述

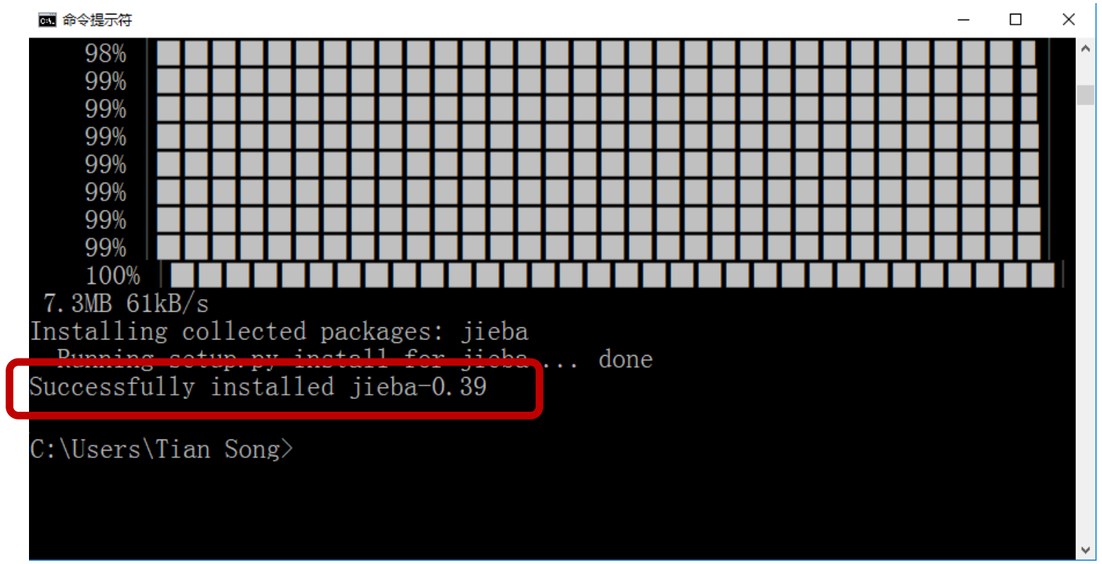
jieba是优秀的中文分词第三方库

- 对中文文本进行分词操作，产生包含产生词语的列表
- jiaba是第三方库，需要额外安装

jieba库的安装

(cmd命令行)　pip install jieba





jieba库使用

函数	描述
jieba.lcut(s)	精确模式，返回字符串s对应的一个列表类型分词结果 >>>jieba.lcut("中国是一个伟大的国家") ['中国', '是', '一个', '伟大', '的', '国家']
jieba.lcut(s, cut_all=True)	全模式，返回字符串s对应的一个列表类型分词结果，存在冗余 >>>jieba.lcut("中国是一个伟大的国家", cut_all=True) ['中国', '国是', '一个', '伟大', '的', '国家']
jieba.add_word(w)	向分词词典增加新词w **>>>jieba.add_word("蟒蛇语言")**

代码演示

```
#WordsCount.py
import jieba
f = open("2018年一号文件.txt", "r", encoding="utf-8")
txt = f.read()
f.close()
ls = jieba.lcut(txt)
d = {}
for w in ls:
    d[w] = d.get(w, 0) + 1
for k in d:
    if d[k] >= 50 and k != "\n":
        print('"{}"出现{}次'.format(k, d[k]))
```

Python快速入门

计算生态编程

Python123

计算生态编程

之一：利用Python庞大的计算生态提高编程产量

- 除了Python语法外，要数量掌握一批Python库的使用
- 对于某些“通用问题”，学会去寻找Python库
- https://pypi.org

计算生态编程

之二：围绕Python计算生态完成编程功能

- 结合Python计算生态中较重要的框架，完成编程任务
- 例如：结合PyTorch开展深度学习应用
- 例如：结合Scrapy框架编写爬虫应用

计算生态编程

之三：构建Python库，丰富Python计算生态

- 对于新的理解和认识，构建Python计算生态
- 底层可以利用C/C++等语言实现，给予Python接口

Python快速入门

"中文词语统计"需求分析

Python123

代码分析

```
#WordsCount.py
import jieba
f = open("2018年一号文件.txt", "r", encoding="utf-8")
txt = f.read()
f.close()
ls = jieba.lcut(txt)
d = {}
for w in ls:
    d[w] = d.get(w, 0) + 1
for k in d:
    if d[k] >= 50 and k != "\n":
        print('"{}"出现{}次'.format(k, d[k]))
```

代码分析

```
#WordsCount.py
import jieba
f = open("2018年一号文件.txt", "r", encoding="utf-8")
txt = f.read()
f.close()
ls = jieba.lcut(txt)
d = {}
for w in ls:
    d[w] = d.get(w, 0) + 1
for k in d:
    if d[k] >= 50 and k != "\n":
        print('"{}"出现{}次'.format(k, d[k]))
```

注释

代码分析

→

```
#WordsCount.py
import jieba
f = open("2018年一号文件.txt", "r", encoding="utf-8")
txt = f.read()
f.close()
ls = jieba.lcut(txt)
d = {}
for w in ls:
    d[w] = d.get(w, 0) + 1
for k in d:
    if d[k] >= 50 and k != "\n":
        print('"{}"出现{}次'.format(k, d[k]))
```

import
方法一
引入外部
功能库

代码分析

```
#WordsCount.py
import jieba
f = open("2018年一号文件.txt", "r", encoding="utf-8")
txt = f.read()
f.close()
ls = jieba.lcut(txt)
d = {}
for w in ls:
    d[w] = d.get(w, 0) + 1
for k in d:
    if d[k] >= 50 and k != "\n":
        print('"{}"出现{}次'.format(k, d[k]))
```

打开文件

关闭文件

代码分析

```
#WordsCount.py
import jieba
f = open("2018年一号文件.txt", "r", encoding="utf-8")
txt = f.read()
f.close()
ls = jieba.lcut(txt)
d = {}
for w in ls:
    d[w] = d.get(w, 0) + 1
for k in d:
    if d[k] >= 50 and k != "\n":
        print('"{}"出现{}次'.format(k, d[k]))
```

把文件内容以文本形式读入

代码分析

```
#WordsCount.py
import jieba
f = open("2018年一号文件.txt", "r", encoding="utf-8")
txt = f.read()
f.close()
ls = jieba.lcut(txt)
d = {}
for w in ls:
    d[w] = d.get(w, 0) + 1
for k in d:
    if d[k] >= 50 and k != "\n":
        print('"{}"出现{}次'.format(k, d[k]))
```

中文分词产生结果保存为列表类型

代码分析

```
#WordsCount.py
import jieba
f = open("2018年一号文件.txt", "r", encoding="utf-8")
txt = f.read()
f.close()
ls = jieba.lcut(txt)
d = {}
for w in ls:
    d[w] = d.get(w, 0) + 1
for k in d:
    if d[k] >= 50 and k != "\n":
        print('"{}"出现{}次'.format(k, d[k]))
```

创建一个空字典键值对的集合

字典类型

字典是映射的结合，体现为键值对的组合

- 映射是一种键(索引)和值(数据)的对应

字典类型

字典是映射的结合，体现为键值对的组合

- 采用大括号{}创建，键值对之间用冒号:表示

{<键1>:<值1>, <键2>:<值2>, ... , <键n>:<值n>}

字典操作：通过键检索值

d = {<键1>:<值1>, <键2>:<值2>, ... , <键n>:<值n>}

d[<键1>]　　结果是：<值1>

d.get(<键1>, 0)　　结果是：<值1>

d.get(<未知>, 0)　　结果是：0

字典操作：增加元素

d = {<键1>:<值1>, <键2>:<值2>, … , <键n>:<值n>}

d[<键n+1>] = <值n+1>

代码分析

```
#WordsCount.py
import jieba
f = open("2018年一号文件.txt", "r", encoding="utf-8")
txt = f.read()
f.close()
ls = jieba.lcut(txt)
d = {}
for w in ls:
    d[w] = d.get(w, 0) + 1
for k in d:
    if d[k] >= 50 and k != "\n":
        print('"{}"出现{}次'.format(k, d[k]))
```

建立每个单词与出现次数的键值对

代码分析


```
#WordsCount.py
import jieba
f = open("2018年一号文件.txt", "r", encoding="utf-8")
txt = f.read()
f.close()
ls = jieba.lcut(txt)
d = {}
for w in ls:
    d[w] = d.get(w, 0) + 1
for k in d:
    if d[k] >= 50 and k != "\n":
        print('"{}"出现{}次'.format(k, d[k]))
```

遍历结果
设置条件
打印输出

Python基本编程解析(下)

Python快速入门

单元小结

Python123


Python 123
Thank you