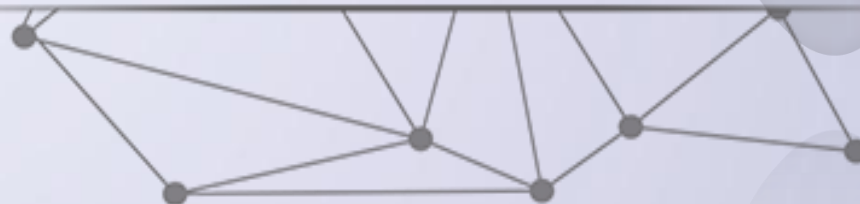




字典实例一



黄天羽

北京理工大学





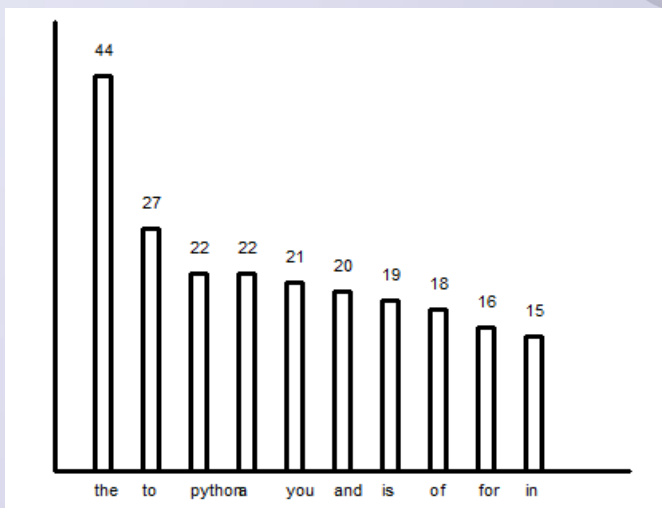
统计词频

- “统计词频” 问题
 - 统计文章其中多次出现的词语
 - 概要分析文章内容
 - 搜索引擎



统计词频IPO描述

- 输入：从文件中读取一篇英文文章
- 处理：统计文件中每个单词的出现频率
- 输出：输出最常出现10个单词及次数图像





统计词频

- 第一步：输入英文文章
- 第二步：建立用于词频计算的字典
- 第三步：对文本的每一行计算词频
- 第四步：从字典中获取数据对到列表中
- 第五步：对列表中的数据对交换位置，并从大到小进行排序
- 第六步：输出结果

最后用Turtle库绘制统计词频结果图表





统计一行词频processLine()

```
def processLine(line, wordCounts):  
    #用空格替换标点符号  
    line = replacePunctuations(line)  
    #从每一行获取每个词  
    words = line.split()  
    for word in words:  
        if word in wordCounts:  
            wordCounts[word] += 1  
        else:  
            wordCounts[word] = 1
```



符号替换repleacePunctuations()


```
def replacePunctuations(line):  
    for ch in line:  
        if ch in "~@#$$%^&*()_-=<>?/,.:;{}[]|\'\"":  
            line = line.replace(ch, "")  
    return line
```



统计词频主程序

- 输入英文文本名称

```
filename = input("enter a filename:").strip()  
infile = open(filename, "r")
```

- 
- 建立一个空字典

```
wordCounts = {}
```

- 对每一行进行统计

```
for line in infile:  
    processLine(line.lower(), wordCounts)
```





■ 词频排序

```
pairs = list(wordCounts.items())
```

■ 交换列表数据项排序

```
items = [[x,y]for (y,x)in pairs]  
items.sort()
```





■ 绘制柱状图

- 初始化窗口、画笔
- 调用drawGraph()进行绘制

```
turtle.title('词频结果柱状图')  
turtle.setup(900, 750, 0, 0)  
t = turtle.Turtle()  
t.hideturtle()  
t.width(3)  
drawGraph(t)
```



定义全局变量

#词频排列显示个数

```
count = 10
```

#单词频率数组-作为y轴数据

```
data = []
```

#单词数组-作为x轴数据

```
words = []
```


#y轴显示放大倍数-可以根据词频数量进行调节

```
yScale = 6
```

#x轴显示放大倍数-可以根据count数量进行调节

```
xScale = 30
```



- 
- drawLine()绘制线段
 - drawText()输出文字

```
#从点 (x1,y1) 到 (x2,y2) 绘制线段
def drawLine(t, x1, y1, x2, y2):
    t.penup()
    t.goto (x1, y1)
    t.pendown()
    t.goto (x2, y2)

# 在坐标 (x,y) 处写文字
def drawText(t, x, y, text):
    t.penup()
    t.goto (x, y)
    t.pendown()
    t.write(text)
```

- 
- drawRectangel()绘制矩形
 - drawBar()绘制多个柱体

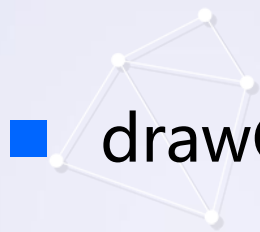
#绘制一个柱体

```
def drawRectangle(t, x, y):  
    x = x*xScale  
    y = y*yScale#放大倍数显示  
    drawLine(t, x-5, 0, x-5, y)  
    drawLine(t, x-5, y, x+5, y)  
    drawLine(t, x+5, y, x+5, 0)  
    drawLine(t, x+5, 0, x-5, 0)
```

#绘制多个柱体

```
def drawBar(t):  
    for i in range(count):  
        drawRectangle(t, i+1, data[i])
```





■ drawGraph()绘制统计图

```
def drawGraph(t):  
    #绘制x/y轴线  
    drawLine(t, 0, 0, 360, 0)  
    drawLine(t, 0, 300, 0, 0)  
  
    #x轴: 坐标及描述  
    for x in range(count):  
        x=x+1 #向右移一位,为了不画在原点上  
        drawText(t, x*xScale-4, -20, (words[x-1]))  
        drawText(t, x*xScale-4, data[x-1]*yScale+10, data[x-1])  
    drawBar(t)
```

程序运行结果

```
>>>
```

```
enter a filename: README.txt
```

```
the      44
```

```
to       27
```

```
python   22
```

```
a        22
```

```
you      21
```

```
and      20
```

```
is       19
```

```
of       18
```

```
for      16
```

```
in       15
```

