第七周：文件和数据格式化

7.1 文件的使用

1.文件的类型

文件是数据的抽象和集合：文件是存储在辅助存储器上的数据序列；文件是数据存储的一种形式；文件展现形态有文本文件和二进制文件。最根本上是二进制文件。

（1）文本文件：由单一特定编码组成的文件，如 UTF-8 编码。由于存在文件，也被看成存储着的长字符串。

（2）二进制文件：文件直接由比特 0 和 1 组成，**没有统一字符编码**。一般存在二进制 0 和 1 的组织结构，即文件格式。

2.文件的打开和关闭

文件处理步骤：打开-操作-关闭。

在不打开文件的时候，文件处于存储状态。当我们需要对其进行操作的时候，要使文件处于占用状态，此时操作对文件具有独一的、排他的操作。处理完成后，需要将文件关闭。

python 中常用的读文件函数为 a.read(size)、a.readline(size)和 a.readlines(hint)；常用的写文件函数为 a.write(s)、a.writelines(lines)和 a.seek(offset)。

打开文件的方式：

<变量名>=open(<文件名>,<打开模式>)

文件名是指文件路径和名称，源文件如果同目录可以省略路径（**路径中的斜杠使用反斜杠/**或者**双斜杠//**）。文件路径的四种方式：1.绝对路径使用反斜杠；2.绝对路径使用双斜杠；3.若文件在程序文件所在目录的子目录中，可以使用例如**./PYE/F.txt** 的方式；4.若是相同目录，可直接使用文件名。

打开模式：文本 or 二进制、读 or 写。如表 1.1 所示。

表 1.1 打开模式

文件打开模式	描述
‘r’	只读模式，默认值，如果文件不存在，返回 FileNotFoundError。
‘w’	覆盖模式，文件不存在则创建，否则完全覆盖。
‘x’	创建写模式，文件不存在则创建，存在则返回 FileExistsError。
‘a’	追加写模式，文件不存在则创建，存在则在文件最后追加内容。
‘b’	二进制文件模式。
‘t’	文本文件模式，默认值。
‘+’	与 r/w/x/a 一同使用，在员工能基础上增加同时读写功能。

其它还可以参见：http://www.yushuai.me/2017/10/18/3494.html。该文章中部分内容摘录如下。

r 只读，r+读写，不创建 w 新建只写，w+新建读写，二者都会将文件内容清零 （以 w 方式打开，不能读出。w+可读写） 以 a,a+的方式打开文件，附加方式打开 （a：附加写方式打开，不可读；a+：附加读写方式打开）

以 'U' 标志打开文件，所有的行分割符通过 Python 的输入方法(例#如 read*()），返回时都会被替换为换行符\n. ('rU' 模式也支持 'rb' 选项) . r 和 U 要求文件必须存在 不可读的打开方式：w 和 a 若不存在会创建新文件的打开方式：a，a+，w，w+

文件关闭方式：

<变量名>.close()

3.文件内容的读取

表 1.2 文件内容的读取

操作方法	描述
a.read(size)	读入全部内容，如果给出参数 size，则只读入前 size 长度的内容。
a.readline(size)	读入一行内容，如果给出参数 size，则只读出该行前 size 长度的内容。
a.readlines(hint)	读入文件所有行，以每行为元素形成列表，如果给出参数，读入前 hint 行。

（1）文件的全文本操作

遍历全文本：方法一（一次性读入，不适合大文件）

```
fname = input("请输入要打开的文件名称：")
fo = open(fname,'r')
txt.fo.read()
#对全文操作
fo.close()
```

方法二（按数量读入，分步处理）

```
fname = input("请输入要打开的文件名称：")
fo = open(fname,'r')
txt.fo.read(2)
while txt!="":
    #对 txt 进行处理
    txt.fo.read(2)
fo.close()
```

逐行遍历文件：方法一（一次读入，分行处理）

```
fname = input("请输入要打开的文件名称：")
fo = open(fname,'r')
txt.fo.read(2)
for line in fo.readlines():
    print(line)
fo.close()
```

方法二（分行读入，逐行处理）

```
fname = input("请输入要打开的文件名称：")
fo = open(fname,'r')
txt.fo.read(2)
for line in fo:
    print(line)
fo.close()
```

4.数据的文件写入

表 1.3 数据的文件写入

操作方法	描述
a.write(s)	向文件写入一个字符串或字节流。
f.writelines(lines)	将一个元素全为字符串的列表写入文件（所有行被合并一起写入）。
f.seek(offset)	改变当前文件操作指针的位置，offset 含义如下：0 位文件开头；1 位当前位置；2 位文件结尾。

例如：

```
fo = open(fname,'w+')
ls=["中国","法国","美国"]
fo.writelines(ls)
for line in fo:
    print(line)
fo.close()
```

为什么没有任何输出呢？

```
fo = open(fname,'w+')
ls=["中国","法国","美国"]
fo.writelines(ls)
fo.seek(0)
for line in fo:
    print(line)
fo.close()
```

必须将指针恢复到文件开头才能真正输出信息，否则指针一直在文件最后，所以无法输出文件中的信息。

7.2 实例 11：自动轨迹绘制

基本思路：

步骤 1：定义数据文件格式（接口）。

步骤 2：编写程序，根据文件接口解析参数绘制图形；

步骤 3：编制数据文件。

数据接口定义是可以随意的，每个人都可以有自己的定义方式。在这里，我们用一行数字代表一次操作：其中第一个数字代表行进距离，第二个代表转向判断（0 左转，1 右转），第三个数字代表绝对转向角度，第四、五、六个数字代表 RGB 三个通道颜色（0~1 之间的浮点数）。

代码：

```
# -*- coding: utf-8 -*-
"""
Created on Tue Apr 24 21:10:49 2018

@author: davidcheung
"""
import turtle as t
t.title('自动轨迹绘制')
```

```
t.setup(800,600,0,0)
t.pencolor('red')
t.pensize(5)
#数据读取
datals=[]
f = open("data.txt")
for line in f:
    line = line.replace("\n","")
    #去掉了每一行提取后变成字符串时的那个引号
    datals.append(list(map(eval,line.split(","))))
    #将去掉逗号的每个数据都转换成数字，然后再转换成列表
    #map 是将第一个参数里面函数的功能作用于第二个参数的每一个数值
    #自动控制
    for i in range(len(datals)):
        t.pencolor(datals[i][3],datals[i][4],datals[i][5])
        t.fd(datals[i][0])
        if datals[i][1]:
            t.right(datals[i][2])
        else:
            t.left(datals[i][2])
```

7.3 一维数据的格式化和处理

1.数据组织的维度

一维数据是由对等关系的有序或无序数据构成，采用线性方式组织。它对应列表、数组和集合等概念。

二维数据是由多个一维数据构成，是一维数据的组合形式。表格是典型的二维数据。

多维数据是由一位或二维数据在新维度上扩展而成。

高维数据是仅利用最基本的二元关系展示数据间的复杂结构，例如键值对。

数据的操作周期分为三个阶段：存储<->表示<->操作。

数据存储中我们关心**存储格式**；数据表示是程序表达的方式，我们关心**数据类型**；数据操作我们关心**操作方式**。

2.一维数据的表示

如果数据间有序，使用**列表类型**。for 循环可以遍历数据，进而对每个数据进行处理。

如果数据间无序，使用**集合类型**。for 循环可以遍历数据，进而对每个数据进行处理。

3.一维数据的存储

存储方式一：空格分隔。只用空格分隔，不换行。缺点是数据中不能存在空格。

存储方式二：逗号分隔。使用英文半角逗号分隔数据，不换行。缺点是数据中不能有英文逗号。

存储方式三：其他方式。利用特殊符号进行分隔，缺点是需要根据数据特点定义，通用性差。

4.一维数据的处理

这里的处理是指将存储于表示相互转换，包括：将存储的数据读入程序，将程序表示的数据写入文件。

将数据写入文件中，文件中内容采用空格分隔方式。代码如下：

```
fname = 'country.txt'
ls = ['中国','美国','日本']
f = open(fname,'w')
f.write(' '.join(ls))
#将 join 前面的字符块分隔放入 join 参数中的字符串
f.close()
```

7.4 二维数据的格式化和处理

1.二维数据的表示

使用列表类型，是**二维列表**。这只是一种**基础**的。可以使用两层 for 循环遍历所有数据。外层列表中，每个元素可以对应一行，也可以对应一列。

2.CSV 格式与二维数据的存储

CSV，即 **Comma-Separated Values**，也就是**用逗号分隔的值**的意思。这是一种国际上通用的一二维数据存储格式，一般用.csv 扩展名。它每行一个一维数据，采用逗号分隔，无空行。CSV 是数据转换的通用标准格式。CSV 有以下约定：

（1）如果某个元素缺失，逗号仍要保留。

（2）二维数据的表头可以作为数据存储，也可以另行存储。

（3）逗号是英文半角逗号，逗号与数据之间没有空格间隔。

（4）按行存和按列存均可，具体由程序决定。但是一般索引习惯：ls[row][column]，即先行后列。

3.二维数据的处理

（1）从 CSV 格式的文件中读入数据

```
fo = open(fname)
ls = []
for line in fo:
    line = line.replace('\n','')
    ls.append(line.split(','))
fo.close()
```

（2）将数据存入 CSV 格式的文件中

```
ls=[[],[],[]]#二维列表
f = open(fname,'w')
for item in ls:
    f.write(','.join(item)+'\n')
f.close()
```

（3）二维数据的逐一处理

```
ls=[[],[],[]]#二维列表
for row in ls:
    for column in row:
        print(ls[row][column])
```

7.5 模块 6：worldcloud 库的安装和使用

1.wordcloud 库的安装

输入命令：

pip install wordcloud

2.wordcloud 使用

wordcloud 库把词云当作一个 WordCloud 对象。即 wordcloud.WordCloud()代表一个文本对应的词云，它可以根据文本中词语出现频率等参数绘制云。

它的常规方法：使用 w=wordcloud.WordCloud()生成一个词云对象，以该对象为基础，配置参数、加载文本、输出对象。方法如表 5.1 所示。

表 5.1 wordcloud 库常规方法

方法	描述
w.generate(txt)	向 WordCloud 对象 w 中加载文本 txt。
w.to_file(filename	将词云输出为图像文件，png 或者 jpg 格式。

使用方法：

（1）配置对象参数。

（2）使用 generate 方法加载词云文本。

（3）输出词云文件。

示例代码如下：

```
import wordcloud
c = wordcloud.WordCloud()
c.generate("wordcloud by Python")
c.to_file("ceshi.png")
```

在这里面，代码做了哪些事情呢？首先，以空格分隔单词。然后统计单词出现次数并过滤。再然后根据统计配置字号。最后布局颜色环境尺寸。

wordcloud 的配置对象参数格式如下：

w= wordcloud.WordCloud(<参数>)

其可配置参数如表 5.2 所示。

5.2 配置对象参数

参数	描述
width	指定词云对象生成图片的宽度，默认 400 像素。
height	指定词云对象生成图片的高度，默认 200 像素。
min_font_size	指定词云中字体的最小字号，默认 4 号。
max_font_size	指定词云中字体的最大字号，根据高度自动调节。
font_step	指定词云中字体字号的步进间隔，默认为 1。
font_path	指定字体文件的路径，默认为 None。
max_words	指定词云显示的最大单词数量，默认为 200。
stop_words	指定词云的排除词列表，即不显示的单词列表。
mask	指定词云形状，默认为长方形，需要引进 imread()函数。例如： from scipy.misc import imread mk = imread(“pic.png”) w= wordcloud.WordCloud(mask=mk)
background_color	指定词云图片的背景颜色，默认为黑色。

例程：

```
import wordcloud
import jieba
txt = "百度，全球最大的中文搜索引擎、最大的中文网站。\
1999 年底，身在美国硅谷的李彦宏看到了中国互联网及中文搜索\
引擎服务的巨大发展潜力，抱着技术改变世界的梦想，他毅然辞掉\
硅谷的高薪工作，携搜索引擎专利技术，于 2000 年 1 月 1 日\
在中关村创建了百度公司"
w = wordcloud.WordCloud(width = 1000,\
                        height = 700,font_path="msyh.ttc")
w.generate(" ".join(jieba.lcut(txt)))
w.to_file("ceshi.png")
```

图片效果如图 1 所示。

图 1

7.6 实例 12：政府报告词云

代码如下：

```
import jieba
import wordcloud
f = open("2018 年政府工作报告.txt", "r", encoding="utf-8")

t = f.read()
f.close()
ls = jieba.lcut(t)

txt = " ".join(ls)
w = wordcloud.WordCloud( \
    width = 700, height = 300,\
    background_color = "white",
    font_path = "msyh.ttc",max_words = 20
    )
w.generate(txt)
w.to_file("grwordcloud.png")
```

结果图如 2 所示。

图 2