貴州大學
GUIZHOU UNIVERSITY

Lecture 6 文本读写技术

- 读取文本文件
- 写入文本文件

- 认识常见的文本读写技术的特点
- 掌握读取文件、写入文件、连接数据库的方法等

文本读写技术实现

- 本堂课内容涉及到文本读写的技术实现，都是采用Python语言完成的

- 用到了Python语言中的Pandas库，请大家在学习本课前安装Python及其Pandas库

读取文本文件

- 用户访问在线商城的系统日志数据，存储在一个文本文件中，源数据的前几行：

	A	B	C	D	E	F
1	USER_IP	ACCESS_DATE	ACCESS_WEEKDAY	ACCESS_TIME	PERIOD	PRODUCT_ID
2	211.83.110.102	2015/12/15 0:00	2	1900/1/1 13:29	2	CNAB20001016014
3	211.83.110.102	2015/12/15 0:00	2	1900/1/1 13:51	2	CNAB20001216001
4	211.83.110.102	2015/12/15 0:00	2	1900/1/1 13:55	2	CNAB20001216006
5	211.83.110.102	2015/12/15 0:00	2	1900/1/1 14:24	2	CNAB20001216002
6	211.83.110.102	2015/12/15 0:00	2	1900/1/1 18:19	3	CNAB20002115012
7	211.83.110.102	2015/12/15 0:00	2	1900/1/1 18:19	3	CNAB20002115012
8	211.83.110.102	2015/12/15 0:00	2	1900/1/1 18:19	3	CNAB20002115008
9	211.83.110.102	2015/12/15 0:00	2	1900/1/1 18:19	3	CNAB20001316012
10	211.83.110.102	2015/12/15 0:00	2	1900/1/1 18:19	3	CNAB20002115008
11	211.83.110.102	2015/12/15 0:00	2	1900/1/1 18:19	3	CNAB20001316012
12	211.83.110.102	2015/12/15 0:00	2	1900/1/1 18:19	3	CNAB20001316001
13	211.83.110.102	2015/12/15 0:00	2	1900/1/1 21:19	4	CNAB20002115013
14	210.41.101.17	2015/12/15 0:00	2	1900/1/1 22:38	4	CNAB20001216004
15	210.41.101.17	2015/12/15 0:00	2	1900/1/1 22:38	4	CNAB20001216007
16	210.41.101.17	2015/12/15 0:00	2	1900/1/1 22:38	4	CNAB20001216002
17	210.41.101.17	2015/12/15 0:00	2	1900/1/1 22:45	4	CNAB20001216006
18	210.41.101.17	2015/12/16 0:00	3	1900/1/1 0:13	5	CNAB20001016009
19	223.104.9.193	2015/12/16 0:00	3	1900/1/1 8:30	1	CNAB20002115011
20	171.213.61.185	2015/12/16 0:00	3	1900/1/1 13:58	2	CNAB20001016014
21	117.136.70.52	2015/12/16 0:00	3	1900/1/1 16:41	3	CNAB20002115011
22	117.136.70.52	2015/12/16 0:00	3	1900/1/1 16:41	3	CNAB20002115009
23	117.136.70.52	2015/12/16 0:00	3	1900/1/1 16:41	3	CNAB20002115008

读取txt文件

- 读取txt文件的步骤
  - 打开test.txt文档需要的操作是

    >>> fp = open('test.txt','r')
  - 当我们输入*>>> fp*
  - Python Shell中会显示

    <open file ‘test.txt’, mode ‘r’ at 0x02BB0E38>

读取txt文件

- 读取txt文件的说明
  - *<open file 'test.txt', mode 'r' at 0x02BB0E38>*表示txt文件已经成功打开
  - open函数的第一个参数是需要打开文本的存储路径，第二个参数‘r’指open函数采用的模式为“读取模式”。

读取test.txt文档中的某几行

- 读取txt文件的某几行
  - 我们想要读取一个文档中的某一行或几行，可以采用下面的一组命令：

    >>> fp.readline() 显示一行

    >>> fp.readlines() 显示所有行

读取文本常用函数

- open()函数中的第一个参数是打开文本文件的路径，第二个参数r代表读取模式，w代表写入模式，a代表追加模式，r+代表读写模式
- read()表示读取到文件尾，size表示读取大小。

读取文本常用函数

- seek(0)表示跳到文件开始位置。
- readline()逐行读取文本文件。
- readlines()读取所有行到列表中，通过for循环可以读出数据。
- close()关闭文件。

读取csv文件

- 可以采用上节中读取txt文件的所有常用函数。
- 还可以采用pandas中提供的一些函数，将csv等表格型文件直接读取到一个Python的DataFrame对象里面。
- 最常用的函数包括read_csv和read_table函数

read_csv函数

- 首先导入pandas库>>> *import pandas as pd*

- 然后通过>>> *df=pd.read_csv('test.csv')* 将test.csv存储df这个DataFrame面。

read_csv函数

- 查看d的命令>>> df

```
>>> df
<class 'pandas.core.frame.DataFrame'>
Int64Index: 619 entries, 0 to 618
Data columns:
USER_IP          619  non-null values
ACCESS_DATE      619  non-null values
ACCESS_WEEKDAY   619  non-null values
ACCESS_TIME      619  non-null values
PERIOD           619  non-null values
PRODUCT_ID       619  non-null values
dtypes: int64(3), object(5)
```

read_table函数

- 首先导入pandas库*>>> import pandas as pd*

- 然后通过

  >>> df=pd.read_table('test.csv', sep=',')

  将test.csv存储df这个DataFrame面。

read_csv函数

- 查看d的命令>>> df

```
>>> df
<class 'pandas.core.frame.DataFrame'>
Int64Index: 619 entries, 0 to 618
Data columns:
USER_IP          619  non-null values
ACCESS_DATE      619  non-null values
ACCESS_WEEKDAY   619  non-null values
ACCESS_TIME      619  non-null values
PERIOD           619  non-null values
PRODUCT_ID       619  non-null values
dtypes: int64(3), object(5)
```

逐块读取文本文件

- 如果只想读取其中的几行（避免读取整个文件），就可以通过nrows来进行指定：

```
>>> df5=pd.read_table('test.csv', nrows=5)
```

逐块读取文本文件

- 当我们要逐块读取文件时，还有一种办法是设置chunksize（行数）：

```
>>>
chunck=pd.read_csv('test.csv',chunksize=5)
```

逐块读取文本文件

- 迭代处理test.csv文件，统计该文件中的行数，我们可以用下面的操作：

```
>>> tot=0
>>> for piece in chunck:
        tot=tot+1
```

- 统计每一块的数据行数后，迭代求出整个test.csv文件中的数据总行数。

写入文本文件

- 要把数据写入txt文件，我们就必须先创建 file 对象。

- 在这情况下，必须用 ‘w’ 模式标记指定要写入的文件。

写入文本文件

- 我们创建一个名叫myfile的文件

```
>>> mydata = ['Date', 'Time']
>>> myfile = open('testit.txt', 'w')
>>> for line in mydata:
        myfile.write(line + '\n')
>>> myfile.close()
```

写入文本文件

- 我们首先把 mydata list的内容写入文件，关闭文件，然后重新打开文件，就可以读取文件内容了。

```
>>> myfile = open("testit.txt")
>>> myfile.read()
    'Date\nTime\n'
>>> myfile.close()
```

同时读取和写入文件

- 我们用‘r+’模式重新打开了文件。

```
>>> myfile = open("testit.txt", "r+a")
>>> myfile.read()  显示'Date\nTime\n'
>>> for line in myfile:
        myfile.write(line + '\n')
```

同时读取和写入文件

```
>>> myfile.seek(0)
>>> myfile.read()  显示
'Date\nTime\nDate\nTime\n'
>>> myfile.close()
```

数据库的连接

- 引入数据处理模块之后，我们就需要和数据库进行连接了。
- db = MySQLdb.connect ("localhost","root","123456","myciti")

上述代码中四个关键的参数：第一个参数是服务器的地址；第二个参数是用户名；第三个参数是dbms密码；第四个参数是需要访问的数据库名称。

执行sql语句

- 连接上数据库之后，我们就需要开始执行sql语句了。
- *import MySQLdb*
  db = MySQLdb.connect("localhost","root","123456","myciti")
  cursor = db.cursor()

执行sql语句

```
sql="""insert into article values
(0,"woainimahah","http://www.aa.com
")"""
try:
    cursor.execute(sql)
    db.commit()
except:
    db.rollback()
db.close
```

选择和打印

- 连接数据库后，获取数据库中的数据信息，并对信息进行展示和打印。

```
import MySQLdb
db =
MySQLdb.connect("localhost","root","12
3456","myciti" )
cursor = db.cursor()
cursor.execute("select * from article“)
```

选择和打印

```
datas = cursor.fetchall()
for data in datas:
    print data[1]
print cursor.rowcount,"rows in tatal"
db.close
```

Fetchall是取出数据库表中的所有行数据

rowcount是读出数据库表中的行数

动态插入

- 动态插入是用占位符来实现的

```
import MySQLdb
title = "title"
url = "urlofwebpage "
db =
MySQLdb.connect("localhost","root","12
3456","myciti" )
cursor = db.cursor()
```

动态插入

```
sql = """insert into article values
(0,"%s","%s","2012-9-
8","wo","qq","skjfasklfj","2019","up")"""
try:
    cursor.execute(sql%(title,url))
    db.commit()
except:
    db.rollback()
db.close
```

update操作

- 在update的操作中，占位符的使用和上面是一样的

```
import MySQLdb
title = "title"
id=11
db =
MySQLdb.connect("localhost","root","123456","myciti" )
```

貴州大學
GUIZHOU UNIVERSITY

update操作

```
cursor = db.cursor()
sql = """update article set title = "%s"
where id = "%d" """
try:
    cursor.execute(sql%(title,id))
    db.commit()
except:
    db.rollback()
db.close
```