

Python及其应用

主讲人：钱惠敏

E-mail: amandaqian@hhu.edu.cn

第6讲 数据文件的操作

6.1 文件的基础

6.2 文件的基本处理

6.1 文件的基础

➤文件

- ✓ 存储在外部介质上的数据或信息的集合
- ✓ 有序的数据序列

➤编码

- ✓ 信息从一种形式转换为另一种形式的过程

➤常用编码

- ✓ ASCII码 (7个二进制位, 128个字符) `ord()`, `chr()`
- ✓ Unicode (2个字节二进制位, 65536个字符)
- ✓ UTF-8 (1-4个字节二进制位)
- ✓ GBK (双字节二进制位) 《汉字内码扩展规范》

6.1 文件的基础(续1)

➤ Python中的字符串类型未编码

✓ encode()

✓ decode()

```
s="中文字符编码"  
bs=s.encode("utf-8")  
print(bs)  
b'\xe4\xb8\xad\xe6\x96\x87\xe5\xad\x97\xe7\xac\xa6\xe7\xbc\x96\xe7\xa0\x81'
```

```
bs.decode("utf-8")  
'中文字符编码'
```

```
s="中文字符编码"  
bs=s.encode("gbk")  
print(bs)  
b'\xd6\xd0\xce\xca\xd7\xd6\xb7\xfb\xbl\xe0\xc2\xeb'
```

```
bs.decode("gbk")  
'中文字符编码'
```

6.1 文件的基础(续2)

➤文件数据的类型

✓ 文本文件 （以ASCII码方式存储的文件）

多行文本 `\n`表示换行

✓ 二进制文件 （照片、音乐、视频、计算机程序等）

```
print("Hello \nWorld \n \nGoodbye")
```

Hello

World

Goodbye

6.2 文件的基本处理

➤ 文件的读取

✓ 打开并读取文件

```
f=open(name[, mode])
```

name是文件名，mode指明打开文件的模式，返回文件对象f。基于f的方法可实现对文件内数据的操作。



```
content=f.read()
```

```
f.close()
```

```
f=open('HW.txt','r')
```

```
content=f.read()
```

```
print(content)
```

```
f.close()
```

Hello World!

每次用完文件后，都要关闭文件。
否则，文件会一直被占用，不能被
其他进程使用。

6.2文件的基本处理(续1)

打开模式	描述	打开模式	描述
r	以只读方式打开文件	r+	以读写方式打开文件
w	以只写方式打开文件	w+	以读写方式打开文件
a	以追加只写方式打开文件	a+	以追加读写方式打开文件
rb	以二进制只读方式打开文件	rb+	以二进制读写方式打开文件
wb	以二进制只写方式打开文件	wb+	以二进制读写方式打开文件
ab	以二进制追加只写方式打开文件	ab+	以二进制追加读写方式打开文件

6.2文件的基本处理（续2）

➤打开并读取文件（续）

`with open(name[, mode]) as f`

操作完成后自动关闭文件。

```
with open('HW.txt','r') as f:  
    content=f.read()  
    print(content)
```

Hello World!

➤在read()函数中加入参数，可指定读取的字符数

```
with open('HW.txt','r') as f:  
    content=f.read(5)  
    print(content)
```

Hello

6.2 文件的基本处理（续3）

➤ 打开并写入文件

```
f=open(name[, mode])
```

```
f.write()
```

```
f.close()
```

```
f=open('HW_w.txt','w')
```

```
f.write('Hello World!')
```

```
f.close()
```



6.2 文件的基本处理（续4）

➤ 常见的数据文件包括 txt 格式或者 csv（逗号分隔值文件格式）格式文件

➤ 文本文件的读取(1)

`f=open()`#打开文件

`content=f.readline()`#一次读入一条数据

或

`content=f.readlines()`#一次读入所有数据

`f.close()`

#	Name	Type 1	Type 2	Total	HP	Attack	Defense	SP.Atk	Sp.Def	Speed	Generation	Legendary
1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	False
1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	False
2	Ivysaur	Grass	Poison	358	58	69	39	67	45	38	1	False
2	Ivysaur	Grass	Poison	358	58	69	39	67	45	38	1	False
3	Venusaur	Grass	Poison	543	68	60	59	37	95	58	1	False
3	Venusaur	Grass	Poison	543	68	60	59	37	95	58	1	False

➤ 文本文件的读取(2)

```
path='Pokemon.txt'
```

```
f=open(path,'r')
```

```
content=f.readlines()
```

```
f.close()
```

```
print(content[0])
```

```
print(content[1])
```

注: `readlines()` 读取后得到的是每行数据组成的列表, 但是一行样本数据全部存储为一个字符串。

```
# Name Type 1 Type 2 Total HP Attack Defense SP.Atk Sp.Def Speed Generation Legendary
1 Bulbasaur Grass Poison 318 45 49 49 65 65 45 1 False
```

```
print(content)
```

```
['#\tName\tType 1\tType 2\tTotal\tHP\tAttack\tDefense\tSP. Atk\tSp. Def\tSpeed\tGeneratio
n\tLegendary\n', '1\tBulbasaur\tGrass\tPoison\t318\t45\t49\t49\t65\t65\t45\t1\tFalse
\n', '1\tBulbasaur\tGrass\tPoison\t318\t45\t49\t49\t65\t65\t45\t1\tFalse\n', '2\tIvysau
r\tGrass\tPoison\t358\t58\t69\t39\t67\t45\t38\t1\tFalse\n', '2\tIvysaur\tGrass\tPoison
\t358\t58\t69\t39\t67\t45\t38\t1\tFalse\n', '3\tVenusaur\tGrass\tPoison\t543\t68\t60\t5
9\t37\t95\t58\t1\tFalse\n', '3\tVenusaur\tGrass\tPoison\t543\t68\t60\t59\t37\t95\t58\t1
\tFalse']
```

Pokemon - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

#	Name	Type 1	Type 2	Total	HP	Attack	Defense	SP.Atk	Sp.Def	Speed	Generation	Legendary
1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	False
1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	False
2	Ivysaur	Grass	Poison	358	58	69	39	67	45	38	1	False
2	Ivysaur	Grass	Poison	358	58	69	39	67	45	38	1	False
3	Venusaur	Grass	Poison	543	68	60	59	37	95	58	1	False
3	Venusaur	Grass	Poison	543	68	60	59	37	95	58	1	False

➤将字符串转换为列表

```
path='Pokemon.txt'
```

```
f=open(path,'r')
```

```
content=f.readlines()
```

```
f.close()
```

```
content_new=[]
```

```
for con in content:
```

```
    tmp=con.split() #对字符串进行切片，默认分隔符包括空格、换行符、制表符
```

```
    content_new.append(tmp) #在列表末尾添加新的对象
```

```
print (content_new[0])
```

```
print (content_new[1])
```

```
['#', 'Name', 'Type', '1', 'Type', '2', 'Total', 'HP', 'Attack', 'Defense', 'SP.Atk', 'Sp.Def',  
'Speed', 'Generation', 'Legendary']
```

```
['1', 'Bulbasaur', 'Grass', 'Poison', '318', '45', '49', '49', '65', '65', '45', '1', 'False']
```

```
print (content_new)
```

```
[['#', 'Name', 'Type', '1', 'Type', '2', 'Total', 'HP', 'Attack', 'Defense', 'SP.Atk',  
'Sp.Def', 'Speed', 'Generation', 'Legendary'], ['1', 'Bulbasaur', 'Grass', 'Poison', '3  
18', '45', '49', '49', '65', '65', '45', '1', 'False'], ['1', 'Bulbasaur', 'Grass', 'Po  
ison', '318', '45', '49', '49', '65', '65', '45', '1', 'False'], ['2', 'Ivysaur', 'Gras  
s', 'Poison', '358', '58', '69', '39', '67', '45', '38', '1', 'False'], ['2', 'Ivysau  
r', 'Grass', 'Poison', '358', '58', '69', '39', '67', '45', '38', '1', 'False'], ['3',  
'Venusaur', 'Grass', 'Poison', '543', '68', '60', '59', '37', '95', '58', '1', 'Fals  
e'], ['3', 'Venusaur', 'Grass', 'Poison', '543', '68', '60', '59', '37', '95', '58',  
'1', 'False']]
```

6.2 文件的基本处理（续7）

➤ 文本文件的写入

```
path='Pokemon_write.txt'
f=open(path,'w')
#写入Pokemon_write的数据
content=[[ '#', 'Name', 'Type 1', 'Type 2', 'Total', 'HP', 'Attack', 'Defense', 'SP.Atk',
'Sp.Def', 'Speed', 'Generation', 'Legendary'],
['1', 'Bulbasaur', 'Grass', 'Poison', '318', '45', '49', '49', '65', '65', '45', '1', 'False'],
['1', 'Bulbasaur', 'Grass', 'Poison', '318', '45', '49', '49', '65', '65', '45', '1', 'False'],
['2', 'Ivysaur', 'Grass', 'Poison', '358', '58', '69', '39', '67', '45', '38', '1', 'False'],
['2', 'Ivysaur', 'Grass', 'Poison', '358', '58', '69', '39', '67', '45', '38', '1', 'False'],
['3', 'Venusaur', 'Grass', 'Poison', '543', '68', '60', '59', '37', '95', '58', '1', 'False'],
['3', 'Venusaur', 'Grass', 'Poison', '543', '68', '60', '59', '37', '95', '58', '1', 'False']]
f.writelines(str(content)) #将一个列表里的所有数据一次性写入文件中

f.close()
```

```
path='Pokemon_write0.txt'
f=open(path,'w')
#写入Pokemon_write0的数据
content=[[ '#', 'Name', 'Type 1', 'Type 2', 'Total', 'HP', 'Attack', 'Defense', 'SP.Atk',
'Sp.Def', 'Speed', 'Generation', 'Legendary'],
['1', 'Bulbasaur', 'Grass', 'Poison', '318', '45', '49', '49', '65', '65', '45', '1', 'False'],
['1', 'Bulbasaur', 'Grass', 'Poison', '318', '45', '49', '49', '65', '65', '45', '1', 'False'],
['2', 'Ivysaur', 'Grass', 'Poison', '358', '58', '69', '39', '67', '45', '38', '1', 'False'],
['2', 'Ivysaur', 'Grass', 'Poison', '358', '58', '69', '39', '67', '45', '38', '1', 'False'],
['3', 'Venusaur', 'Grass', 'Poison', '543', '68', '60', '59', '37', '95', '58', '1', 'False'],
['3', 'Venusaur', 'Grass', 'Poison', '543', '68', '60', '59', '37', '95', '58', '1', 'False']]
#循环content中的内容
for con in content:
    #print(con)
    con='\t'.join(con) #是用.join()将列表中每个数值用\t联合组成一个字符串
    #print(con)
    con=con+'\r\n' #每行后添加换行符
    #print(con)
    f.write(con) #逐次写入
f.close()
```

6.2 文件的基本处理（续8）

➤逗号分割值文件的读取

```
import csv
path='Pokemon.csv'
f=open(path,'r')
reader=csv.reader(f)#生成reader对象
content=[]
#for循环，从reader中逐个添加存储
for con in reader:
    content.append(con)
f.close()
print (content[0])
print (content[1])
```

```
['#', 'Name', 'Type 1', 'Type 2', 'Total', 'HP', 'Attack', 'Defense', 'SP.Atk', 'Sp.Def', 'Speed', 'Generation', 'Legendary']
['1', 'Bulbasaur', 'Grass', 'Poison', '318', '45', '49', '49', '65', '65', '45', '1', 'False']
```


➤逗号分割值文件的写入

```
import csv
path='Pokemon_write.csv'
#写入Pokemon的数据
content=[['#','Name','Type 1','Type 2','Total','HP','Attack','Defense','SP.Atk',
'Sp.Def','Speed','Generation','Legendary'],
          ['1','Bulbasaur','Grass','Poison','318','45','49','49','65','65','45','1','False'],
          ['1','Bulbasaur','Grass','Poison','318','45','49','49','65','65','45','1','False'],
          ['2','Ivysaur','Grass','Poison','358','58','69','39','67','45','38','1','False'],
          ['2','Ivysaur','Grass','Poison','358','58','69','39','67','45','38','1','False'],
          ['3','Venusaur','Grass','Poison','543','68','60','59','37','95','58','1','False'],
          ['3','Venusaur','Grass','Poison','543','68','60','59','37','95','58','1','False']]
f=open(path,'w')
writer=csv.writer(f) #生成writer对象
#循环写入content中的内容
for con in content:
    writer.writerow(con)
f.close()
```

例：文件合并

已知有两个文件，电话号码簿TelAddBk.txt，邮箱地址EmailAddBk.txt。要求：将两个文件合并为AddBk.txt。

分析**IPO**模式：

I：电话簿、邮箱地址簿文件

P：合并两个文件的内容

O：合并后的电话和邮箱地址簿文件

程序步骤:

1. 打开文件，读取文件；
2. 分别获取两个文件的信息，并存放至列表1和列表2；
3. 建立合并用的空列表3，将列表1和2的信息合并到列表3.
 - 3-1 生成信息表头（姓名，电话，邮箱）
 - 3-2 遍历列表1的人，创立列表3的信息（此时应该处理列表2中与列表1中相同姓名的信息）
 - 3-3 处理列表2中剩余人的信息，合并到列表3
4. 将列表3的信息写入到新文件
5. 关闭所有打开的文件

程序步骤:

1. 打开文件，读取文件;

```
fadd1=open('EmailAddBk.txt','rb')
```

```
fadd2=open('TeleAddBk.txt','rb')
```

#第一行数据不要

```
fadd1.readline()
```

```
fadd2.readline()
```

#读取第一行后的数据

```
lines1=fadd1.readlines()
```

```
lines2=fadd2.readlines()
```

2. 分别获取两个文件的信息，并存放至列表1和列表2；

```
l1_name=[] #建立空列表存储姓名、电话、Email
```

```
l1_tele=[]
```

```
l2_name=[]
```

```
l2_email=[]
```

```
for line in lines1:    #读取TeleAddBk中的信息
```

```
    elements=line.split()
```

```
    l1_name.append(str(elements[0].decode('gbk')))
```

```
    l1_tele.append(str(elements[1].decode('gbk')))
```

```
for line in lines2:    #读取EmailAddBk中的信息
```

```
    elements=line.split()
```

```
    l2_name.append(str(elements[0].decode('gbk')))
```

```
    l2_email.append(str(elements[1].decode('gbk')))
```

3. 建立合并用的空列表3，将列表1和2的信息合并到列表3.

3-1 生成信息表头（姓名，电话，邮箱）

#生成新的数据

```
lines=[]
```

```
lines.append('姓名\t 电话  \t 邮箱\n')
```

3. 建立合并用的空列表3，将列表1和2的信息合并到列表3.

3-2 遍历列表1的人，创立列表3的信息（此时应该处理列表2中与列表1中相同姓名的信息）

#按索引方式遍历姓名列表

```
for i in range(len(l1_name)):
```

```
    s=""
```

```
    if l1_name[i] in l2_name:
```

```
        j=l2_name.index(l1_name[i])
```

```
        s='\t'.join([l1_name[i],l1_tele[i],l2_email[j]])
```

```
        s+='\n'
```

```
    else:
```

```
        s='\t'.join([l1_name[i],l1_tele[i],str(' ---- ')])
```

```
        s+='\n'
```

```
    lines.append(s)
```

3. 建立合并用的空列表3，将列表1和2的信息合并到列表3.

3-3 处理列表2中剩余人的信息，合并到列表3

#处理姓名列表2中的剩余姓名

```
for i in range(len(l2_name)):
```

```
    s=""
```

```
    if l2_name[i] not in l1_name:
```

```
        s='\t'.join([l2_name[i],str(' ---- '),l2_email[i]])
```

```
        s+='\n'
```

```
    lines.append(s)
```


4. 将列表3的信息写入到新文件

#将生成的合并数据写入新的文件

```
fadd3=open('AddBk.txt','w')
```

```
fadd3.writelines(lines)
```

5. 关闭所有打开的文件

#关闭文件

```
fadd3.close()
```

```
fadd2.close()
```

```
fadd1.close()
```

```
print('The AddressBooks are merged.')
```