

bat 腳本中 For /f 中的 Delims 和 Tokens 總結

2018-03-13 254

nds 最後一行 ogr ims 愛人 輸出 總結 ech 包括

在 For 命令語句的參數 F 中，最難理解的就是 Delims 和 Tokens 兩個選項，本文簡單的做一個比較和總結。

“For/f” 常用來解析文本，讀取字符串。分工上，delims 負責切分字符串，而 tokens 負責提取字符串。如果把字符串當作蛋糕，Delims 像刀子，用來切蛋糕，tokens 像叉子，用來取切好的蛋糕。下面我們用實例來進行理解。

把以下內容保存為文本文件“歌曲列表.txt”，注意擴展名為“.txt”：

序號、歌手名-歌曲名.後綴名

- 1、饒天亮-玫瑰愛人.wma
- 2、高一首-我不願錯過.mp3
- 3、黃凱芹-傷感的戀人.MP3
- 4、黃燦-黃玫瑰.lrc
- 5、黎姿-如此這般的愛情故事.mp3

代碼 1：顯示全部內容

```
@echo off
```

```
for /f %%i in (歌曲列表.txt) do echo %%i
```

```
pause>nul
```

運行結果：

序號、歌手名-歌曲名.後綴名

- 1、饒天亮-玫瑰愛人.wma
- 2、高一首-我不願錯過.mp3
- 3、黃凱芹-傷感的戀人.MP3
- 4、黃燦-黃玫瑰.lrc
- 5、黎姿-如此這般的愛情故事.mp3

講解：

如果不使用參數“/f”，運行結果只顯示括號裏的文字字符“歌曲列表.txt”，而不能讀取文本文件“歌曲列表.txt”中的內容。可見，“/f”是解析文本字符串的好工具。

一、delims

假如只要序號，不要歌手名、歌曲名和後綴名，如何辦到？

代碼 2：默認提取第一列

```
@echo off
for /f "delims=、" %%i in (歌曲列表.txt) do echo %%i
pause>nul
```

運行結果：

序號

- 1
- 2
- 3
- 4
- 5

講解：

"delims=、"表示定義頓號“、”為分隔符，並用該分隔符“、”切分文本字符串。字符串就是“歌曲列表.txt”裏的內容，也就是文件裏的文字和標點符號。該頓號是原文中就有的。除了頓號“、”，原文中還有減號“-”和點號“.”，因此你也可以用它們來做分隔符。

代碼 3：用減號“-”做分隔符

```
@echo off
for /f "delims=-" %%i in(歌曲列表.txt) do echo %%i
pause>nul
```

運行結果：

序號、歌手名

- 1、饒天亮
- 2、高一首
- 3、黃凱芹
- 4、黃燦
- 5、黎姿

講解：

因為，當減號“-”被用做分隔符時，每行內容被減號“-”分隔成前後兩半，默認只顯示前半部分，而後半部分連同分隔符減號“-”都被忽略（省略）了。

代碼 4：用點號“.”做分隔符

```
@echo off
```

```
for /f "delims=." %%i in(歌曲列表.txt) do echo %%i
pause>nul
```

運行結果：

序號、歌手名-歌曲名

- 1、饒天亮-玫瑰愛人
- 2、高一首-我不願錯過
- 3、黃凱芹-傷感的戀人
- 4、黃燦-黃玫瑰
- 5、黎姿-如此這般的愛情故事

講解：

默認情況下，單純使用 delims 而不用 tokens 時，只顯示第一個分隔符前的內容，第一個分隔符和第一個分隔符後面的內容將被忽略。

代碼 5：定義多個分隔符

```
@echo off
for /f "delims=、-." %%i in (歌曲列表.txt) do echo %%i
pause>nul
```

運行結果：

序號

- 1
- 2
- 3
- 4
- 5

講解：

原因是，當定義頓號“、”、減號“-”和點號“.”三個標點符號為分隔符後，原文被分隔成四個部分。

如第二行“1、饒天亮-玫瑰愛人.wma”將被分隔成“1”、“饒天亮”、“玫瑰愛人”和“wma”四個部分。

從第一行到最後一行，每行的每個部分對應下來相當於一個豎列。因此，原文就有“序號”、“歌手名”、“歌曲名”、“後綴名”四列。

一般情況下，只讀取第一列的內容。後面的內容需要用 tokens 選項提取。

二、tokens

假如只要歌手名和歌曲名，不要序號和後綴名，如何辦到？

代碼 6：提取單列

```
@echo off
```

```
for /f "tokens=2 delims=、." %%i in (歌曲列表.txt) do echo %%i
pause>nul
```

運行結果將顯示：

歌手名-歌曲名

饒天亮-玫瑰愛人

高一首-我不願錯過

黃凱芹-傷感的戀人

黃燦-黃玫瑰

黎姿-如此這般的愛情故事

講解：

用 delims 定義頓號“、”和點號“.”作為分隔符，將原文分成三部分。

如第六行“5、黎姿-如此這般的愛情故事.mp3”被分割成：

第一部分（第一列）：5

第二部分（第二列）：黎姿-如此這般的愛情故事

第三部分（第三列）：mp3

“tokens=2”表示用 tokens 提取第二列的字符串，即“黎姿-如此這般的愛情故事”。

沒有被 tokens 定義提取的第一列和第三列將被忽略。

假如只要序號和歌曲名，而不要歌手名和後綴名，如何辦到？

代碼 7：提取多列

```
@echo off
```

```
for /f "tokens=1,3delims=、-." %%i in (歌曲列表.txt) do echo %%i%%j
pause>nul
```

運行結果將顯示：

序號歌曲名

1 玫瑰愛人

2 我不願錯過

3 傷感的戀人

4 黃玫瑰

5 如此這般的愛情故事

講解：

“delims=、-.”表示定義頓號、減號和點號為分隔符。

“tokens=1,3”表示只提取第一列和第三列。

“%%i %%j”對應於“token”後面的列數。有多少列就要有多少個輸出變量，並且各變量中的字母存在先後順序。

如何只提取文字，不要標點符號？

代碼 8：忽略分隔符

```
@echo off
```

```
for /f "tokens=1,2-4delims=、-." %%i in (歌曲列表.txt) do echo %%i %%j %%k%%l  
pause>nul
```

運行結果：

序號 歌手名 歌曲名後綴名

- 1 饒天亮玫瑰愛人 wma
- 2 高一首我不願錯過 mp3
- 3 黃凱芹傷感的戀人 MP3
- 4 黃燦黃玫瑰 lrc
- 5 黎姿如此這般的愛情故事 mp3

講解：

默認情況下，用做分隔符的標點符號將被忽略。

“tokens=1,2-4”中的“2-4”表示第二至第四列。

三、小結

“For/f”

一句話總結：解析文本，讀取字符串。

Delims 的語法：

```
FOR /F "Delims=符號集" %%I
```

```
IN (Command1) DO
```

```
Command2
```

一句話總結：忽略分隔符，切分字符串。

delims 的意義包括兩個方面：

第一、指定原文中的標點符號作為分隔符。使得文本被劃分為許多小部分，方便使用批處理命令讀取和編輯。

第二、讀取第一個分隔符之前的內容。忽略第一個分隔符和分隔符後面的內容，如需讀取和編輯，需要使用 tokens 等命令。

注意分隔符和標點符號的聯系和區別。

分隔符就是原文中的標點符號，可以是一個標點符號也可以是多個。但原文中的標點符號不一定是分隔符。

分隔符需要定義，即用“delims=”來指定，等於號後面跟被用來做分隔符的標點符號，該標點符號來自原文。

當有多個標點符號被定義為分隔符時，標點符號之間沒有空格。當用空格做分隔符時，空格應該放在其它用作分隔符的標點符號之後。

即使不使用 delims，默認情況下，批處理也將空格作為分隔符。如文件名

“ProgramFiles”中含有空格時，批處理一般只讀取空格前的“Program”，剩下的“Files”被忽略了。

例：

代碼 9：空格默認作為分隔符

```
@echo off
```

```
for /f "delims=" %%a in("偉大的中國人民萬歲 萬歲 萬萬歲") do echo%%a
```

```
pause>nul
```

運行結果：

偉大的中國人民萬歲 萬歲萬萬歲

講解：

“delims=”代表取消默認以空格作為分隔符。

批處理默認空格是分隔符，分隔符會隱藏第一個分隔符後面的所有內容。為了顯示完整的信息，所以有必要取消該功能。

因此，當文件名或路徑中含有空格時，需要用雙引號括住。如果不使用雙引號，括號內的內容將被當做文件名，如果文件名不存在，將提示“系統找不到……”。

如果不使用“delims=”，即使使用雙引號，也只能顯示第一個空格前面的字符串，後面的字符串將被默認忽略。

Tokens 的語法：

```
FOR /F "tokens=x,y,m-n" %%I IN (Command1) DO Command2
```

一句話總結：提取列。

Tokens 的意義就是，提取指定的列。

注意列與句的聯系與區別。

文本內容由許多文字字符串組成，它們被標點符號分隔，兩個標點符號之間的語句我們稱為“句子”。

當標點符號被“tokens=”指定為分隔符之後，文字將被分成多個部分。對應的每行的每個部分我們稱之為“列”。

一列可能是一個句子，也可能包含多個句子，視 delims 定義的標點符號而定。

其它方面：關於“行”的提取，前面做過一個很簡單的：

```
@echo off
```

```
findstr /n .* 12.txt>>122.txt
```

```
set /p m=請輸入要定位的行：
```

```
findstr /b "%m%" 122.txt
```

```
pause
```

bat 腳本中 For /f 中的 Delims 和 Tokens 總結

相關文章

- [bat 腳本中 For /f 中的 Delims 和 Tokens 總結](#)
- [批處理之 for /f 中的 delims 和 tokens](#)
- [js 中的 null 和 undefined 總結](#)
- [Shell 腳本中的 if、case、for、while、until](#)
- [理解 c# 的 Get Set 訪問器及測試腳本中的應用](#)
- [Shell 腳本中的變量](#)
- [shell 腳本中\\$的用處](#)
- [shell 腳本中的邏輯判斷、文件目錄屬性判斷、if 特殊用法、case 判斷](#)
- [shell 腳本介紹、shell 腳本結構和執行、date 命令用法、shell 腳本中的變量](#)
- [Shell 腳本中的邏輯判斷、文件目錄屬性判斷、if 的特殊用法、case 判斷](#)
- [shell 腳本中的邏輯判斷、文件目錄屬性判斷、if 特殊用法、case 判斷](#)
- [shell 腳本中的函數，shell 中的數組，shell 項目-告警系統](#)
- [shell 腳本中的邏輯判斷 文件目錄屬性判斷 if 特殊用法 case 判斷](#)
- [shell 腳本介紹 shell 腳本結構和執行 date 命令用法 shell 腳本中的變量](#)
- [Shell 腳本中的邏輯判斷、文件目錄屬性判斷、if 特殊用法、case 判斷](#)