

for 迴圈

批次檔的 for 迴圈很重要，但有些特性比較隱晦，不容易駕馭，下面稍加整理需注意使用之處

1. for 迴圈初始化變數，在撰寫為批次檔時，請使用 %%variable，而在命令列執行時要改用 %variable。
2. for 迴圈初始化變數有大小寫的區分，所以 %%i 不同於 %%I。
3. for 迴圈內的變數會有取值異常的情形。

以下針對第 3 點取值異常的情形作一說明，迴圈的敘述通常是以下列格式撰寫，左括弧與 do 同一行，右括弧放在最後面獨立成行

```
for %%i in (*) do (  
    echo %%i  
    timeout 1  
)
```

但其實批次檔會把迴圈內的敘述集結成一行變成

```
for %%i in (*) DO echo %%i & timeout 1
```

但就因為集結成一行的關係，批次檔在命令解譯器進行直譯時，會對每一行敘述中的變數進行預先賦值的動作，因此會造成同一行敘述中，對變數改變其值卻無效的情形，請猜一下底下這個範例會顯示的值是甚麼？

```
@echo off  
set var=Peter  
set var=John & echo %var%  
timeout 6
```

結果仍是 Peter，因為針對第 3 行，命令解譯器會預先賦值給 %var%，然後才進行 set var = John，若要正確取出此變數異動後的值 John，必須 [啟用變數延遲展開\(SETLOCAL ENABLEDELAYEDEXPANSION\)](#)的功能，在開啟用變數延遲展開功能之後，取用變數的方式要由 %var% 更改為 !var!，將範例修改如下

```
@echo off  
SETLOCAL ENABLEDELAYEDEXPANSION
```

```
set var=Peter
set var=John & echo !var!
timeout 6
```

批次檔中只要是利用括弧()分成多行撰寫的指令，實際上都看成一行，在括弧()裡面取用變數時都會遇到需要開啟用延遲環境變數擴充功能的問題，一定要特別注意。

以下直接說明範例不特別解釋語法，細節請透過 `for /?` 學習

找出符合條件之檔案的 `for` 迴圈

1. 顯示批次檔存在的目錄中所有符合 `.mp4 .avi *.mpg` 的檔案名稱
2. `for %%i in (*.mp4 *.avi *.mpg) DO @echo %%i`
3. 顯示使用者目錄中的所有檔案名稱 此例必須直接在命令列輸入(注意變數名稱的差別)、環境變數 `userprofile` 代表使用者目錄
4. `for %i in (%userprofile%\*) DO @echo %i`

找出符合條件之目錄的 `for /D` 迴圈

- 顯示使用者目錄中的所有目錄名稱 此例必須直接在命令列輸入(注意變數名稱的差別)、環境變數 `userprofile` 代表使用者目錄
- `for /D %i in (%userprofile%\*) DO @echo %i`

遞迴搜尋指定的路徑下所有符合檔案的 `for /R` 迴圈

- 將 `c:\temp\` 目錄與所有子目錄下的 `*.bak` 刪除
- `for /R c:\temp\ %%G in (*.bak) do del "%%G"`

可以設定開始數值、增/減數值、停止數值的 `for /L` 迴圈

1. 顯示 0-100 的數字 (從 0 開始、遞增 1、終止值 100)
2. `for /L %%i in (0 1 100) do echo %%i`
3. 計算從 1 累加至 100 的和
4. `set sum=0`
5. `for /L %%i IN (100, -1, 1) DO set /a sum+=%%i`
6. `echo %sum%`
7. 啟用延遲環境變數擴充功能範例
8. `SETLOCAL ENABLEDELAYEDEXPANSION`
9. `for /L %%i in (1 1 5) do (`
10. `set var=%%i`

```
11.echo !var!  
12.)  
13.timeout 6
```

逐行讀取文字檔的 for /F 迴圈

for /F 是迴圈中最重要的應用，因為它可以讀取檔案、讀取字串與讀取命令，分別說明如下：

1. 讀取檔案：逐行讀取指定的檔案，然後依照分隔符號將內容賦值給指定變數
範例：逐行讀取 test.ini 文字檔內容，以等號(=)為分隔，左邊給%%i，右邊給%%j
2. FOR /F "tokens=1,2 delims==" %%i IN (test.ini) DO set %%i=%%j

test.ini 內容如下

```
account=john  
passwd=a1234567890
```

上述指令執行完畢之後會有下列效果，將原來必須放在批次檔內部的變數設定，放在外部的 ini 檔之後進行讀取，將程式碼與設定檔分離，可減少原始檔被亂改的機會

```
set account=john  
set passwd=a1234567890
```

3. 讀取字串：讀取字串或變數，然後依照分隔符號將內容賦值給指定變數
範例：讀取變數%date%，以斜線(/)為分隔，依序給%%a、%%b、%%c 三變數，再依照年/月/日的格式，儲存到 mydate 變數
4. For /f "tokens=1-3 delims=/ " %%a in ("%date%") do set mydate=%%a/%%b/%%c
5. 讀取命令：依照分隔符號將內容賦值給指定變數

命令執行的結果，成為 for /F 迴圈讀取資料的來源，須注意此命令須包含在單引號之間，若命令中包含了管線 |，則須在管線前方加上逸脫字元 ^

範例 1：將命令 date /t 的執行結果，以斜線(/)為分隔，依序給%%a、%%b、%%c 三變數，輸出年-月-日的格式

```
for /F "tokens=1-3 delims=/ " %%a in ('date /t') do echo %%a-%%b-%%c
```

範例 2：將命令 `sc query` 的執行結果透過管線輸出給 `find` 指令，尋找含有 `SERVICE_NAME` 字串的列，然後將第 2 欄的內容存到 `%i` 變數並顯示出來

```
rem 此例必須直接在命令列輸入(注意變數名稱的差別)
for /f "tokens=2" %i in ('sc query ^| find /i "SERVICE_NAME"')
do @echo %i
```

`sc query` 命令會顯示目前系統所有的服務

範例 3：利用 `findstr` 過濾文字檔(.csv)包含有井字號(#)的列,然後以逗號(,)為分隔，依序給`%%I`、`%%J`、`%%K`、`%%L` 4 變數，然後再分別設定到有意義的變數中儲存

```
rem 設定參數檔名與批次檔相同(.csv)
set cfg=%~n0.csv
FOR /F "tokens=1,2,3,4 delims=," %%I IN ('findstr /V [#] %cfg%') DO (
    set remoteDIR=%%I
    set localBackupFolder=%%J
    set RetentionDay=%%K
    set isCallCheckspace=%%L
)
```

與批次檔同名的 csv 檔，其內容如下

```
#遠端備份名稱,本地端備份路徑,備份保留天數,是否檢查硬碟剩餘空間
myweb,C:\backup,20,Y
```