https://blog.csdn.net/A\_BlackMoon/article/details/79998515

**windows 批处理 for循环语句**

2018年04月18日 23:08:52 [寒夕若梦](https://me.csdn.net/A_BlackMoon) 阅读数：234

命令格式：for{%variable |%% variable } in (集合) do

command [options]

%variable |%% variable: 代表可替换参数。使用%variable通过命令提示符执行for命令。使用%% variable在批处理文件中执行for命令；

这个变量可以是26个英文字母任意一个，也可以是其他；这些变量会区分大小写，%%x 和%%X代表不同的变量；避免与bat的%0~%9发生冲突，最好不要定义成这个形参。

集合: 文件，目录，字符串或者范围数值的集合；可以是一个，也可以是多个，每一个字符串或变量，我们称之为一个元素，每个元素之间，用空格键、跳格键、逗号、分号或等号分隔

|  |
| --- |
| *@echo off*  *for %%x in (nicky=rebecia=jady) do (*  *echo %%x*  *)*  *pause* |

Command: 要执行的命令

Options: 参数选项

|  |
| --- |
| *@echo off*  *for %%x in (\*.txt) do (*  *echo %%x content listed as follow:*  *type %%x \n*  *)*  *pause* |

for /f 是以行为基本操作单位的，类似于Linux awk命令

**二、切分字符串的利器：delims=x**

rem 以：做为分隔符，且取得：之前的数据，后面的数据不见了。

for /f "delims=:" %%i in (1.txt) do

|  |
| --- |
| *@echo off*  *for /f "delims=:" %%i in (1.txt) do (*  *echo %%i*  *)*  *pause* |

**如果不指定，默认是以空格或者tab作为分割的。**

你也可以一次性指定多个分割符号，即所谓的符号列表，这样只要满足符号列表的数据，都会把符号前面的数据提取出来

语法：for /f"delims=,.: " %%i in (1.txt) do 四个符号作为分割

**三、定点提取：tokens= 数字 指定第几个token输出**

Tokens后面一般跟数字，可以是多个数字，但是每一个数字之间用逗号分开，如tokens=3,5,7 含义是：提取第3,5,8节字符串

for /f "delims=: tokens=2" %%i in (1.txt)do

1.txt 内容如下：

|  |
| --- |
| A:B:C:D:E  AA:BB:CC:DD  AAA;BBB;CCC;DDD  AAAA,BBBB,CCCC,DDDD |

|  |
| --- |
| *@echo off*  *for /f "delims=,;: tokens=1" %%i in (1.txt) do (*  *echo %%i*  *)*  *Pause*  *Results:*  *A*  *AA*  *AAA*  *AAAA* |

既然如此，那么我指定多个tokens, 怎么接收呢？

可以再指定一个变量，来接受第二个token,语法如下：

for /f "delims=,;: tokens=1,2" %%i in(1.txt) do

其中tokens=1,N 也可以表示为tokens=1-N,那么之间连续的数都会被使用：

|  |
| --- |
| *@echo off*  *for /f "delims=,;: tokens=1,2" %%i in (1.txt) do (*  *echo %%i %%j*  *)*  *Pause*  *结果：*  *A B*  *AA BB*  *AAA BBB*  *AAAA BBBB* |

**四：跳过无关内容：skip=n 跳过无关的行，从N+1行开始处理**

|  |
| --- |
| *@echo off*  *for /f "skip=1 delims=,;: tokens=1,2" %%i in (1.txt) do (*  *echo %%i %%j*  *)*  *Pause*  *结果：*  *AA BB*  *AAA BBB*  *AAAA BBBB* |

把第一行给去掉了

**五、忽略指定字符开头的行 eol=**

语法格式：for /f"eol=A,B delims=,;: tokens=1,2" %%i in (1.txt) do

忽略以A和B开始的行

|  |
| --- |
| *@echo off*  *for /f "eol=A,B delims=,;: tokens=1,2" %%i in (1.txt) do (*  *echo %%i %%j*  *)*  *pause* |

**六：集合中语法**

|  |
| --- |
| *6.1 如果集合是读取文件作为内容：*  *for –f %%x in (\*.txt) do*  *6.2 如果集合是命令执行结果,圆括号内容需要用单引号括起来：*  *for –f %%x in ('dir /a-d /b \*test\*.txt') do*  *6.2 如果集合是字符串，应该用双引号括起来：*  *for –f %%x in ("AA.BB,CC,DD") do* |

**七：usebackq 增强性质的参数**，那么for语句第一个括号内的写法有所改变，即集合的写法有所改变，怎么改变？

|  |
| --- |
| *7.1 如果以前是用单引号括起来的命令结果作为参数，那么单引号改为后引号``;*  *for –f %%x in (·dir /a-d /b \*test\*.txt·) do*  *7.2 如果之前是字符串的话，那么双引号改为单引号*  *for –f %%x in (‘AA.BB,CC,DD’) do*  *7.3 如果是文件名的话，就需要用双引号括起来*  *for –f %%x in (“\*.txt”) do* |

我们还必须要注意一件事情，就是路径问题：如果路径包含特殊符号我们需要用双引号把路径括起来

**八：变量延迟和扩展**

例子：

|  |
| --- |
| @echo off  set var=nicky && echo %var%  pause |

显示结果echo is off

|  |
| --- |
| @echo off  set var=nicky  if %var%==nicky (      set var=belly      echo %var%  ) |

一般新手会认为var是belly,但其结果显示的是nicky

为什么呢？这就和变量延迟有关系了。

windows在解释执行此代码段时，在遇到if,for语句后的括号后，只把它当一条语句处理而不是两条语句,那么既然如此，那么在if语句内的var变量虽然被赋值了，但是这条语句相当于只是做了预处理，并没有真正影响到执行时的var值。即这条语句还没有被执行，那怎样才会得到想要结果呢，首先容易想到的是在if语句后面去输出var值。

**方法一：**

|  |
| --- |
| *@echo off*  *set var=nicky*    *if %var%==nicky (*  *set var=belly*  *)*  *echo %var%*  *pause*  *结果：belly* |

**方法二：开启变量延迟**

*setlocal enabledelayedexpansion用于开启变量延迟*

告诉解释器，在遇到复合语句的时候，不要将其作为一条语句同时处理，而仍然一条一条地去解释。但是这时必须用!var!来引用变量，如果仍然用%var%引用是无效的

|  |
| --- |
| *@echo off*  *setlocal enabledelayedexpansion*  *set var=nicky*    *if %var%==nicky (*  *set var=belly*  *echo !var!*  *)*  *Pause*  *结果：belly* |

开启变量延迟之前的变量和之后的变量的值包含感叹号，是有一点区别的

|  |
| --- |
| *@echo off*  *setlocal enabledelayedexpansion*  *set var=nicky*    *if %var%==nicky (*  *set var=belly!*  *echo !var!*  *)*  *Pause*  *结果还是belly,不是belly!,windows给去掉了* |

应对这种办法只有暂时关闭延迟了，或者在延迟之前set 变量的值

|  |
| --- |
| *@echo off*  *set var=nicky*  *if %var%==nicky (*  *set var=belly!*  *setlocal enabledelayedexpansion*  *echo !var!*  *)*  *pause* |

**十：FOR /L%variable IN (start,step,end) DO command [command-parameters]**

|  |
| --- |
| *@echo off*  *for /L %%x in (2,2,10) do echo %%x*  *pause*  *result:*  *2*  *4*  *6*  *8*  *10* |

**十一：FOR /D%variable IN (set) DO command [command-parameters]**

如果Set (也就是我上面写的"相关文件或命令")包含通配符（\*和?），将对与Set相匹配的每个目录（而不是指定目录中的文件组）执行指定的Command。

|  |
| --- |
| *@echo off*  *：：列举出包含e和i的目录*  *for /D %%a in (\*e\* \*i\*) do echo %%a*  *pause*  *结果：*  *Cache*  *deploy*  *LogsAndDiagnostics* |

**十二：递归FOR /R[[drive:]path] %variable IN (set) DO command [command-parameters]**

进入根目录树[Drive:]Path，在树的每个目录中执行for语句。如果在/R后没有指定目录，则认为是当前目录。如果Set只是一个句点(.)，则只枚举目录树。

|  |
| --- |
| *@echo off*    *rem 遍历当前目录下所有文件*  *for /r %%x in (\*) do (echo %%x)*  *pause* |