set,E文翻译过来就是“设置”的意思，相当于数学里的“令”。   
如：set X=5，就是令X=5的意思。   
语法形式：  
SET [variable=[string]]   
SET /P variable=[promptString]    
SET /A expression

一、SET [variable=[string]]

示例1:  
@echo off    
set  
pause    
显示所有的变量的值

示例2:

@echo off    
set var=我是值    
echo %var%    
pause    
请看 set var=我是值 ,这就是BAT直接在批处理中设置变量的方法!    
set 是命令 var是变量名 =号右边的"我是值"是变量的值    
在批处理中我们要引用这个变量就把var变量名用两个%(百分号)扩起来,如%var%

二、SET /P variable=[promptString]

有时候我们需要提供一个交互界面,让  用户自己输入变量的值,然后我们在来根据这个值来做相应操作,现在我就来说说这SET的这  种语法,只需要加一个"/P"参数就可以了!    
实例1:    
@echo off    
set /p var=请输入你的名字:     
echo 您的名字是:%var%  
pause      
set /p 是命令语法 var是变量名 =号右边的"请输入变量的值: ",这个是提示语,不是变    
量的值了!    
运行后,我们在提示语后面直接输入robin,就会显示一行您” 您的名字是:robin”

三、SET /A expression

/A 命令行开关指定等号右边的字符串为被评估的数字表达式。  
该表达式解析很简单并以递减的优先权顺序支持下列操作:   
() - 分组    
! ~ - - 一元运算符    
\* / % - 算数运算符    
+ - - 算数运算符    
<< >> - 逻辑移位    
& - 按位“与”    
^ - 按位“异”    
| - 按位“或”    
= \*= /= %= += -= &= ^= |= <<= >>= - 赋值   
, - 表达式分隔符    
set的/A参数就是让SET可以支持数学符号进行加减等一些数学运算!    
现在开始举例子介绍这些数学符号的用法:    
看例子 这里的例子请直接在CMD下拷贝命令运行,不需要保存为BAT!    
set /a var=1 + 1     
set /a 语法, var变量名 1 + 1 数学式子    
  拷贝运行后会直接显示一个2,或者运行完后我们输入echo %var%,也是二,这就是    
一个简单的加法运算!

复制代码代码如下:

set /a var=2 - 1 结果是多少呢?如果你看不到结果就echo %var%.....    
set /a var=2 \* 2 乘法运算    
set /a var=2 / 2 除法运算    
set /a var=(1+1) + (1+1) 结果等于4 看得懂吧!

实例1：

复制代码代码如下:

@echo off  
set /a a=1+1,b=2+1,c=3+1   
echo %a% %b% %c%

实例2：

复制代码代码如下:

@echo off  
set a=100  
set b=1000  
set /a c=a+b  
echo the result is :%c%  
pause

有时候我们需要直接在原变量进行加减操作就可以用这种语法    
set /a var+=1 这样的语法对应原始语法就是set /a var = %var% + 1    
都是一样的结果,在原变量的值上在进行数学运算,不过这样写简单一点    
在来一个:     
set /a var\*=2     
其他都这么用,只要帮助里有这个语法!    
另外还有一些用逻辑或取余操作符,这些符号,按照上面的使用方法会报错的    
比如我们在CMD里输入set /a var=1 & 1 "与运算",他并不会显示为1,而是报错,    
为什么?对于这样的"逻辑或取余操作符",我们需要把他们用双引号引起来,看例子

复制代码代码如下:

set /a var= 1 "&" 1 这样结果就显示出来了,其他逻辑或取余操作符用法    
set /a var= 1 "^" 1 异运算    
set /a var= 1 "%" 1 取模运算    
set /a var= 2 "<<" 2 位左移    
set /a var= 4 ">>" 2 位右移

实例3：

复制代码代码如下:

@echo off  
set a=100  
set b=1000  
set /a c=a+b  
set /a c= c "<<" 2  
echo the result is :%c%  
pause

这些符号也可以用&= ^= |= <<= >>= 这样的简单用法如    
set /a var"&=" 1 等于set /a var = %var% "&" 1 注意引号    
set的通常用法示例： 

**1、set 显示环境变量：**   
如：在命令行下输入：   
set 会显示当前所有的系统环境变量   
set c 会显示所有的以c开头或C开头的变量（也即不区分大小写）；   
set com 会显示所有的以com开头或COM开头的变量（不区分大小写）； 

**2、set 设置环境变量，用法示例：set 表达式，**   
如：set str=ok   
就是将变量str的值设置成字符串“OK”   
注意：set str=ok>   
这个在批处理运行过程中会出错，系统会提示：“命令语法不正确。”   
原因是，因为 变量str 中的值“ok>”的字符">"为一特殊字符，会被系统当成 定向 符处理，所以会运行出错，那如果确实要设置这样的变量str，值为 "ok>"怎么办呢？解决办法有二：将特殊字符转义，转义方法在该特殊字符前加一个“^”，如：set str=ok^> (这种方法比较少用，因为没有多少应用价值)   
常见的解决办法，在set后面将整个表达式用 引号括起来，如：set "str=>" 

**3、带参数 /a的set 数值计算**set的/A参数就是让SET可以支持数学符号进行加减等一些数学运算! 支持的数学运算包含（优先权递减）:   
() - 分组   
! ~ - - 一元运算符   
\* / % - 算数运算符   
+ - - 算数运算符   
<< >> - 逻辑移位   
& - 按位“与”   
^ - 按位“异”   
| - 按位“或”   
= \*= /= %= += -= - 赋值   
&= ^= |= <<= >>=   
, - 表达式分隔符   
用法示例：set /a num=2\*10,这样变量num的值将变为：20

复制代码代码如下:

set /a var=2 - 1 结果是多少呢?如果你看不到结果就echo %var%.....   
set /a var=2 \* 2 乘法运算   
set /a var=2 / 2 除法运算   
set /a var=(1+1) + (1+1) 结果等于4 看得懂吧!   
set /a a=1+1,b=2+1,c=3+1 运行后会显示一个4,但我们用   
echo %a% %b% %c%后看结果,会发现其他数学运算也有效果!,这就是"斗"号的   
作用!

有时候我们需要直接在原变量进行加减操作就可以用这种语法   
set /a var+=1 这样的语法对应原始语法就是set /a var = %var% + 1   
都是一样的结果。   
对于"逻辑或取余操作符",我们需要把他们用双引号引起来,看例子

复制代码代码如下:

set /a var= 1 "&" 1 这样结果就显示出来了,其他逻辑或取余操作符用法   
set /a var= 1 "+" 1 异运算   
set /a var= 1 "%" 1 取模运算   
set /a var= 2 "<<" 2 次方运算   
set /a var= 4 ">>" 2 这个不太记得数学里的叫法....

这些符号也可以用&= ^= |= <<= >>= 这样的简单用法如   
set /a var"&=" 1 等于set /a var = %var% "&" 1 注意引号   
  
**4、带参数 /p 的set 等待用户输入**windows 2000 和windows 是没有choice这个命令，choice的等待用户输入功能就要通过set /p来实现   
演示如下：

复制代码代码如下:

@echo off   
set /p str=请输入你想输入的字符：   
echo 你输入的字符串为：%str%   
pause>nul

set /p的特殊应用（set /p=<nul相当于输出字符并不带回车，利用这个性质，可以做出比较漂亮的动画），用法示例：

复制代码代码如下:

@echo off   
for /l %%i in (1 1 70) do (   
set /p=O<nul   
for /l %%a in (1 1 50) do ver>nul   
)   
pause>nul

注：代码中用到了一句for /l %%a in (1 1 50) do ver>nul，它的目的是利用for循环运行某一命令来达到延时的目的。 

**5、利用set截取字符 保存下面的代码为test.bat运行查看即可。**

复制代码代码如下:

@echo off&color 1f&rem num=15 nums=30   
mode con: cols=80 lines=35   
set "var=1234567890"   
set "num= "   
set "nums= "   
rem 第一屏 舍弃   
cls&echo 第一屏 舍弃   
echo\&echo %num% var=%var%&echo\   
for /l %%i in (1 1 3) do call :dan %%i   
echo\   
for /l %%i in (0 1 1) do call :yi %%i   
echo\   
for /l %%i in (0 1 2) do call :\_yi %%i   
echo ~ 波浪号后面的数字：为正数表示舍弃变量的前位;   
echo , 逗号后面的数字为正数表示取变量的舍弃以后的前几位;   
echo , 逗号后面为负数表示舍弃变量的后几位   
echo\&echo\&pause&color 1e   
rem 第二屏 取   
cls&echo 第二屏 取   
echo\&echo %num% var=%var%&echo\   
for /l %%i in (-1 -1 -3) do call :dan %%i   
echo\   
for /l %%i in (-4 -1 -6) do call :er %%i   
echo\   
call :\_er -8   
echo ~ 波浪号后面直接跟负数表示取变量的后几位   
echo , 逗号后面的数字为正数表示取变量的舍弃以后的前几位;   
echo , 逗号后面为负数表示舍弃变量的后几位   
echo\&echo\&pause&color 1f   
rem 第三屏 替换 删除   
cls&echo 第三屏 替换 删除   
set "var=abc:\123\ef:g\123\456"hij"789"klm\"   
echo\&echo %num%var=%var%   
echo\   
call :san   
echo = 号左边为要替换掉的字符, 右边为需替换来的字符   
echo = 号左边要替换掉的字符前面加个\*号,代表第一次出现的字符及其前面的所有字符   
echo\   
echo ========= 演示完毕 ========= 按任意键退出 ==========   
pause>nul   
exit   
:dan   
set shu=%1   
if %shu% lss 0 (set aa=取&set ji=最后) else set aa=舍弃&set ji=前   
if "%shu:~0,1%"=="-" set shu=%shu:~1%   
set str1=%%var:~%1%%%num%   
set str2=%aa%var的%ji%%shu%位%nums%   
call set str3=%%var:~%1%%%num%   
call echo %%str1:~0,15%% %%str2:~0,30%% %%str3:~0,15%%   
goto :eof   
:yi   
set shu=%1   
if %shu% lss 0 (set aa=取&set ji=最后) else set aa=舍弃&set ji=前   
for /l %%i in (1 1 3) do (   
set str1=%%var:~%1,%%i%%%num%   
set str2=%aa%var的%ji%%shu%位之后取%%i位%nums%   
call set str3=%%var:~%1,%%i%%%num%   
call echo %%str1:~0,15%% %%str2:~0,26%% %%str3:~0,15%%   
if %%i==3 echo\   
)   
goto :eof   
:\_yi   
for /l %%i in (-1 -1 -3) do (   
set str1=%%var:~%1,%%i%%%num%   
set str2=舍弃var的前%1位和%%i位%nums%   
call set str3=%%var:~%1,%%i%%%num%   
call echo %%str1:~0,15%% %%str2:~0,28%% %%str3:~0,15%%   
if %%i==-3 echo\   
)   
goto :eof   
:er   
set sss=%1   
set sss=%sss:~1%   
for /l %%i in (1 1 3) do (   
set str1=%%var:~%1,%%i%%%num%   
set str2=从var的倒数%sss%位开始取%%i位%nums%   
call set str3=%%var:~%1,%%i%%%num%   
call echo %%str1:~0,15%% %%str2:~0,24%% %%str3:~0,15%%   
if %%i==3 echo\   
)   
goto :eof   
:\_er   
set sss=%1   
set sss=%sss:~1%   
for /l %%i in (-1 -1 -3) do (   
set str1=%%var:~%1,%%i%%%num%   
set str2=从var的倒数%sss%位开始并舍弃最后%%i位%nums%   
call set str3=%%var:~%1,%%i%%%num%   
call echo %%str1:~0,15%% %%str2:~0,20%% %%str3:~0,15%%   
if %%i==-3 echo\   
)   
goto :eof   
:san   
set str1=%%var:"=%%%num%   
set str2=删除var中的所有^"号%num%   
set str3=%var:"=%%nums%   
call echo %%str1:~0,13%% %%str2:~0,23%% %%str3:~0,30%%   
set str1=%%var:\=%%%num%   
set str2=删除var中的所有\号%num%   
set str3=%var:\=%%nums%   
call echo %%str1:~0,13%% %%str2:~0,23%% %%str3:~0,30%%   
set str1=%%var::=%%%num%   
set str2=删除var中的所有:号%num%   
set str3=%var::=%%nums%   
call echo %%str1:~0,13%% %%str2:~0,23%% %%str3:~0,30%%   
echo\   
set str1=%%var:"=好%%%num%   
set str2=替换var中的所有^"号为好字%num%   
set str3=%var:"=好%%nums%   
call echo %%str1:~0,12%% %%str2:~0,20%% %%str3:~0,30%%   
set str1=%%var:\=好%%%num%   
set str2=替换var中的所有\号为好字%num%   
set str3=%var:\=好%%nums%   
call echo %%str1:~0,12%% %%str2:~0,20%% %%str3:~0,30%%   
set str1=%%var::=好%%%num%   
set str2=替换var中的所有:号为好字%num%   
set str3=%var::=好%%nums%   
call echo %%str1:~0,12%% %%str2:~0,20%% %%str3:~0,30%%   
echo\   
set str1=%%var:\*"=%%%num%   
set str3=%var:\*"=%%nums%   
echo 删除var中的第一次出现的^"号及其前面的所有字符   
call echo %%str1:~0,15%% %%str3:~0,30%%   
echo.   
set str1=%%var:\*\=%%%num%   
set str3=%var:\*\=%%nums%   
echo 删除var中的第一次出现的\号及其前面的所有字符   
call echo %%str1:~0,15%% %%str3:~0,30%%   
echo.   
set str1=%%var:\*:=%%%num%   
set str3=%var:\*:=%%nums%   
echo 删除var中的第一次出现的:号及其前面的所有字符   
call echo %%str1:~0,15%% %%str3:~0,30%%   
echo\   
set str1=%%var:\*"=ppp%%%num%   
set str3=%var:\*"=ppp%%nums%   
echo 替换var中的第一次出现的^"号及其前面的所有字符为ppp   
call echo %%str1:~0,15%% %%str3:~0,30%%   
echo.   
set str1=%%var:\*\=ppp%%%num%   
set str3=%var:\*\=ppp%%nums%   
echo 替换var中的第一次出现的\号及其前面的所有字符为ppp   
call echo %%str1:~0,15%% %%str3:~0,30%%   
echo.   
set str1=%%var:\*:=ppp%%%num%   
set str3=%var:\*:=ppp%%nums%   
echo 替换var中的第一次出现的:号及其前面的所有字符为ppp   
call echo %%str1:~0,15%% %%str3:~0,30%%   
echo\   
goto :eof

批处理Set命令详解 补充篇  
**一、设置自定义变量**

复制代码代码如下:

@echo off   
set var=我是值   
echo %var%   
pause

**二、设置自定义变量，并接受用户输入作为值**

复制代码代码如下:

@echo off   
@echo off   
set /p var=请输入变量的值:   
if %var% == 1 echo 您输入了 1 ~\_~   
pause

**三、SET /A expression**   
set的/A参数就是让SET可以支持数学符号进行加减等一些数学运算! 支持的数学运算包含（优先权递减）:

复制代码代码如下:

() - 分组   
! ~ - - 一元运算符   
\* / % - 算数运算符   
+ - - 算数运算符   
<< >> - 逻辑移位   
& - 按位“与”   
^ - 按位“异”   
| - 按位“或”   
= \*= /= %= += -= - 赋值   
&= ^= |= <<= >>=   
, - 表达式分隔符

Batch代码

复制代码代码如下:

set /a var=2 - 1 结果是多少呢?如果你看不到结果就echo %var%.....   
set /a var=2 \* 2 乘法运算   
set /a var=2 / 2 除法运算   
set /a var=(1+1) + (1+1) 结果等于4 看得懂吧!   
set /a a=1+1,b=2+1,c=3+1 运行后会显示一个4,但我们用   
echo %a% %b% %c%后看结果,会发现其他数学运算也有效果!,这就是"斗"号的作用!

有时候我们需要直接在原变量进行加减操作就可以用这种语法   
set /a var+=1 这样的语法对应原始语法就是set /a var = %var% + 1   
都是一样的结果。   
对于"逻辑或取余操作符",我们需要把他们用双引号引起来,看例子   
Batch代码

复制代码代码如下:

set /a var= 1 "&" 1 这样结果就显示出来了,其他逻辑或取余操作符用法   
set /a var= 1 "+" 1 异运算   
set /a var= 1 "%" 1 取模运算   
set /a var= 2 "<<" 2 次方运算   
set /a var= 4 ">>" 2 这个不太记得数学里的叫法....

这些符号也可以用&= ^= |= <<= >>= 这样的简单用法如   
set /a var"&=" 1 等于set /a var = %var% "&" 1 注意引号   
下面是一个具体的例子：

复制代码代码如下:

@echo off   
set p=123   
title 密码是%p%   
set n=1   
:in   
if %n% gtr 3 goto wrong   
set /p a=请输入您需要登陆的密码:   
if "%a%" equ "%p%" goto right   
set/a n+=1   
goto in   
:right   
echo 猜对了，新年快乐！   
goto end   
:wrong   
echo 已经输过3次了，还不正确，退出！   
:end   
pause

**四、替换变量值的内容**   
语法：set newVar="oldVar:str1=str2%   
将变量oldVar的值中的str1替换成str2,并将替换后的变量值赋给newVar.   
看例子

复制代码代码如下:

@echo off   
set a=bbs.verybat.cn   
echo 替换前的值: "%a%"   
set var=%a:bbs=sss%   
echo 替换后的值: "%var%"   
pause

对比一下,我们发现他把变量%a%的bbs给替换掉了,换成了sss.   
**五、字符串截取**   
1、%Var:~start,len%   
注意：从0开始。   
例如：

复制代码代码如下:

@echo off   
set a=www.jb51.net   
set var=%a:~1,2%   
echo %var%   
pause

2、%Var:~-len%   
截取倒数len位   
3、%Var:~0,-len%   
截取从第一位开始，倒数第len位结束的值。   
例子

复制代码代码如下:

@echo off   
set a=www.jb51.net   
set var=%a:~0,-3%   
echo %var%   
pause

执行后,我们发现显示的是"www.jb51.net",少了"et"   
从结果分析,很容易分析出,这是把变量a的值从0位开始,   
到倒数第三位之间的值全部赋予给var

[上一页1](http://www.jb51.net/article/18973.htm)[**2**](http://www.jb51.net/article/18973_2.htm)[阅读全文](http://www.jb51.net/article/18973_all.htm)

**您可能感兴趣的文章:**