

# 批处理BAT脚本中set和if命令的使用



Batcher

2015-08-07

# 在线视频分享约定

1. 专心听就行了，不用着急做笔记。视频和文档会放到网上供大家[免费下载](#)。
2. 请不要使用音箱等外放设备，以免有回声影响其他人，[最好使用耳机](#)。
3. 理论知识介绍完之后，随机从论坛上找几个帖子[现场演示](#)分析问题、解决问题的过程。

# 内容概要

- set

- 赋值
- 数学运算
- 读取输入
- 字符串截取 ★
- 字符串替换

- if

- 判断上一条命令的退出代码
- 判断文件是否存在
- 字符串比较 ★

# 使用 set 命令进行赋值

注意事项	😊	😞
➤ 等号两边不要有空格	<pre>C:\&gt;set str=BatHome  C:\&gt;echo %str% BatHome</pre>	<pre>C:\&gt;set str = BatHome  C:\&gt;echo %str% %str%</pre>
➤ 变量值包含特殊字符需用双引号	<pre>C:\&gt;set "str=Bat&amp;Home"  C:\&gt;echo "%str%" "Bat&amp;Home"</pre>	<pre>C:\&gt;set str=Bat&amp;Home 'Home' 不是内部或外部命令，也 不是可运行的程序或批处理文件。  C:\&gt;echo "%str%" "Bat"</pre>

- 变量值设置为空可以清空变量值，实现变量初始化。
- 避免使用系统环境变量同名的自定义变量，例如：  
path , date , errorlevel 等

语法可行但不推荐使用：

- 使用连等把多个变量赋值成数字 `set /a x=y=100`
- 逗号分隔把多个变量赋值成数字 `set /a x=1,y=2`

# 使用 set /a 命令进行数学运算（1）

- 等号右侧的变量可以省略百分号

```
C:\>set x=1
C:\>set y=2
C:\>set /a n=%x%+%y%
3
```

```
C:\>set x=1
C:\>set y=2
C:\>set /a n=x+y
3
```

- 八进制和十六进制转换成十进制很简单

```
C:\>set /a 012
10
```

```
C:\>set /a 0xA
10
```

- BAT文件里面进行“取余”操作的时候需要使用两个百分号

```
@echo off
set /a n=3%%2
echo %n%
pause
```

- 逻辑运算的特殊字符需要使用双引号

```
@echo off
set /a "m=1<<10"
set /a n=1"<<"10
echo %m% %n%
pause
```

# 使用 set /a 命令进行数学运算（2）

➤ 不支持直接计算超大数（例如磁盘空间的计算）

```
C:\>set /a n=2147483646+1  
2147483647
```

```
C:\>set /a n=2147483647+1  
-2147483648
```

```
C:\>set /a n=2147483648+1  
无效数字。数字精确度限为 32 位。
```

```
C:\>set /a n=-2147483646-1  
-2147483647
```

```
C:\>set /a n=-2147483647-1  
-2147483648
```

```
C:\>set /a n=-2147483648-1  
无效数字。数字精确度限为 32 位。
```

➤ 不支持直接计算浮点数

批处理通过自己编写算法实现浮点计算

<http://bbs.bathome.net/thread-3322-1-1.html>

<http://bbs.bathome.net/thread-3372-1-1.html>

➤ 合理使用位运算可以简化代码、提高效率

位运算相关概念及批处理演示

<http://bbs.bathome.net/thread-1844-1-1.html>

<http://bbs.bathome.net/thread-5884-1-1.html>

➤ 对于超大数、浮点数的计算，BAT调用VBS或PowerShell是相对简单的解决方法。

➤ 截取日期、时间之后进行 set /a 计算时08和09会被看做非法八进制数字，参考：

<http://bbs.bathome.net/thread-29062-1-1.html>

<http://bbs.bathome.net/thread-29253-1-1.html>

# 使用 set /p 命令读取输入

➤ 交互获取用户输入

```
@echo off
set /p input=请输入一个字符串:
echo,%input%
pause
```

➤ 读取文件的第一行内容

```
C:\>echo BatHome>1.txt
C:\>set /p input=<"1.txt"
C:\>echo %input%
BatHome
```

➤ 输出字符串之后不产生换行

```
@echo off
for %%i in (bat home) do (
    echo,%%i
)
pause
```

```
@echo off
for %%i in (bat home) do (
    set /p =%%i<nul
)
pause
```

批处理光标回退（非cls清屏）动画效果演示

<http://bbs.bathome.net/viewthread.php?tid=11266#pid159704>

# 使用 set 命令进行字符串截取

原始字符串	字符串截取 set sub=%str:~x,y%	结果
set str=12345	截取第一个到第三个字符 set sub=%str:~0,3%	<div>str</div> → <div>12345</div> → <div>sub</div>
set str=12345	截取第二个到最后一个字符 set sub=%str:~1%	<div>str</div> → <div>12345</div> → <div>sub</div>
set str=12345	截取倒数第四个到倒数第二个字符 set sub=%str:~-4,3% set sub=%str:~-4,-1%	<div>str</div> → <div>12345</div> → <div>sub</div>
set str=12345	截取最后两个字符 set sub=%str:~-2%	<div>str</div> → <div>12345</div> → <div>sub</div>

- 当x为正数时，表示从左到右截取；当x为负数时，表示从右到左截取。
- 当y为正数时，表示要截取后字符串的长度；当y为负数时，表示要抛弃的最后几个字符长度。
- 当y及其之前的逗号不存在时，表示截取的是第x+1个字符及其之后的所有字符。



# 使用 set 命令进行字符串替换

**set StrNew=%StrOld:SubOld=SubNew%**

➤ 把字符串 bbs 替换成 www

```
set StrOld=http://bbs.bathome.net
set StrNew=%StrOld:bbs=www%
```

➤ 把字符串 bbs 删掉

```
set StrOld=http://bbs.bathome.net
set StrNew=%StrOld:bbs=%
```

➤ 把第一字符 t 及其左侧所有字符全部删掉（统配符 \* 的使用）

```
set StrOld=http://bbs.bathome.net
set StrNew=%StrOld:*t=%
```

**待替换的子字符串不区分大小写（可以利用这个特性实现字母大小写转换）**

```
@echo off
setlocal enabledelayedexpansion
REM 全部转换成大写字母
set str=http://bbs.BATHOME.net
set up=A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
for %%i in (%up%) do (
    set str=!str:%%i=%%i!
)
echo %str%
pause
```

```
@echo off
setlocal enabledelayedexpansion
REM 全部转换成小写字母
set str=http://bbs.BATHOME.net
set low=a b c d e f g h i j k l m n o p q r s t u v w x y z
for %%i in (%low%) do (
    set str=!str:%%i=%%i!
)
echo %str%
pause
```

# set命令知识点(1)把命令结果赋值给变量

UNIX/Linux Shell 支持命令替换	Windows BAT 借助重定向或者for语句来实现
<pre>PacketLoss=`ping -c 4 127.0.0.1   grep '%` echo \$PacketLoss</pre>	<pre>@echo off REM 利用重定向把命令结果（单行字符串）赋值给变量 ping 127.0.0.1   findstr "%%" &gt; "%temp%\ping.txt" set /p PacketLoss=&lt;"%temp%\ping.txt" echo %PacketLoss% pause</pre>
<pre>PacketLoss=\$(ping -c 4 127.0.0.1   grep '%') echo \$PacketLoss</pre>	<pre>@echo off REM 利用for语句把命令结果（单行字符串）赋值给变量 for /f "delims=" %%i in ('ping 127.0.0.1 ^  findstr "%%"') do (     set "PacketLoss=%%i" ) echo %PacketLoss% pause</pre>

# set命令知识点(2)显示以某字符开头的变量

- 直接执行 set 命令可以显示系统环境变量
- 命令 set xxx 可以显示所有以 xxx 开头的变量
- 显示结果按照变量名进行排序

```
C:\>set xxx1=A
C:\>set xxx3=C
C:\>set xxx2=B
C:\>set xxx
xxx1=A
xxx2=B
xxx3=C
```

- 利用“排序”特性的完整代码演示

```
@echo off
for %%i in (a.txt c.txt b.txt e.txt) do (
    set #%%i=%%i
)
for /f "tokens=2 delims==" %%i in ('set #') do (
    echo %%i
)
pause
```

在解决实际问题中的应用

<http://bbs.bathome.net/viewthread.php?tid=33539#pid159774>

<http://bbs.bathome.net/viewthread.php?tid=35443#pid167675>

<http://bbs.bathome.net/viewthread.php?tid=34927#pid165547>

<http://bbs.bathome.net/viewthread.php?tid=35705#pid168703>

# 内容概要

- set

- 赋值
- 数学运算
- 读取输入
- 字符串截取 ★
- 字符串替换

- if

- 判断上一条命令的退出代码
- 判断文件是否存在
- 字符串比较 ★

# 使用 if 命令判断上一条命令的退出代码

- 左括号的左边需要有空格
- else 右边需要有空格左边建议有空格
- 右括号、else、左括号需要放在同一行

```
@echo off
echo bbs.bathome.net | findstr "bat"
if %errorlevel% equ 0 (
    echo 找到指定字符串
) else (
    echo 没有找到指定字符串
)
pause
```

- if errorlevel 0 判断的是 errorlevel 是否大于等于 0

把 if errorlevel 0 放在最前面在逻辑上是错误的

```
if errorlevel 0 (
    echo 找到指定字符串
) else if errorlevel 1 (
    echo 没有找到指定字符串
)
```



```
@echo off
echo bbs.bathome.net | findstr "bat"
if errorlevel 1 (
    echo 没有找到指定字符串
) else (
    echo 找到指定字符串
)
pause
```

# 使用 if 命令判断文件是否存在

不使用反斜线的情况下 if exist 命令无法直接区分文件和文件夹

- 判断文件夹的时候在最后加上反斜线

```
@echo off
if exist "C:\Program Files\" (
    echo 文件夹存在
) else (
    echo 文件夹不存在
)
pause
```

- 判断文件的时候先排除文件夹
- if not 表示条件不成立的情况下执行命令

```
@echo off
if not exist "1.txt\" (
    if exist "1.txt" (
        echo 文件存在
    ) else (
        echo 文件不存在
    )
)
pause
```

# 使用 if 命令进行字符串比较

➤ 使用 == 判断两个字符串或数字是否相等

```
if 1 == 1 echo yes
if "a" == "a" echo yes
```

➤ 使用 /i 可以忽略字母大小写

```
if "a" == "A" echo yes
if /i "a" == "A" echo yes
```

➤ 数字比较不要使用双引号

➤ 使用双引号之后会变成字符串比较

```
if 2 gtr 10 echo yes
if "2" gtr "10" echo yes
```

➤ 比较操作符

equ	equal	等于
neq	not equal	不等于
lss	less than	小于
leq	less than or equal	小于或等于
gtr	greater than	大于
geq	greater than or equal	大于或等于

批处理if命令字符串比较和ASCII的关系

- 数字小于字母。
- 同一个字母，小写字母小于大写字母。
- 不同的字母，按照字母表中的顺序排列。

扩展阅读

<http://bbs.bathome.net/thread-6851-1-1.html>  
<http://bbs.bathome.net/thread-12314-1-1.html>  
<http://bbs.bathome.net/thread-12329-1-1.html>

# if命令知识点(1)变量延迟扩展

## 复合语句需要启用变量延迟扩展

```
@echo off
set VAR=before
if "%VAR%" == "before" (
    set VAR=after
    if "%VAR%" == "after" (
        echo 重新赋值成功
    ) else (
        echo 重新赋值失败
    )
)
pause
```



```
@echo off
setlocal enabledelayedexpansion
set VAR=before
if "%VAR%" == "before" (
    set VAR=after
    if "!VAR!" == "after" (
        echo 重新赋值成功
    ) else (
        echo 重新赋值失败
    )
)
pause
```



```
@echo off
for %i in ("C:\Windows\*.exe") do (
    set list=%list% %i
)
echo %list%
pause
```



```
@echo off
setlocal enabledelayedexpansion
for %i in ("C:\Windows\*.exe") do (
    set list=!list! %i
)
echo %list%
pause
```



批处理中的变量延迟扩展、变量嵌套  
<http://bbs.bathome.net/thread-2899-1-1.html>



# if命令知识点(2)删除重复行

使用 if defined 命令删除文本中的重复行

```
@echo off
for /f "delims=" %%i in (1.txt) do (
    if not defined _%%i (
        echo,%%i
        set "_%%i=1"
    )
)
pause
```

```
C:\>type 1.txt
bbs
bathome
net
bathome
```

```
C:\>test.bat
bbs
bathome
net
请按任意键继续. . .
```

这个方法不适合用来处理非常大的文件，而且无法处理包含等号的行，参考：  
<http://bbs.bathome.net/thread-4690-1-1.html>

THANKS  
&  
GOOD LUCK

请在自愿的前提下支持论坛的发展  
论坛唯一专用支付宝账号 [batcher.li@gmail.com](mailto:batcher.li@gmail.com)