cmd

cmd 常见内部命令

cmd 批处理常用符号详解

@

一般在它之后紧跟一条命令或一条语句，则此命令或语句本身在执行的时候不会显示在屏幕上。

```
echo a
@pause
@echo b
@pause
```

%、%%

百分号在不同的场合，有不同的含义：

1、当百分号成对出现，并且其间包含非特殊字符时，一般做变量引用处理，比如：%var%、%str%。

```
@echo off
set str=abc
echo 变量 str 的值是:  %str%
pause
```

2、百分号作为变量引用还有一种特殊形式，那就是对形式参数的引用，此时，单个百分号后面紧跟 0～9 这10个数字，如 %0、%1，其中，%0 为脚本本身的名称，%1 至 %9 为第二至九个参数，最多支持%0～%9。%10 以后就是变量引用了，即 %15 为 %1 的值接上 5。

```
@echo off

if defined str goto next

set str=
set /p str=请把文件拉到本窗口后回车:
call "%~0" %str%
pause
exit

:next
cls
echo 本批处理文件完整路径为:"%~0"
echo 拖到本窗口的文件完整路径为:"%~1"

goto :eof
```

3、%% 出现在 set /a 语句中时，表示两数相除取余数，也就是所谓的模运算。它在命令行窗口和批处理文件中的写法略有差异。

- 在命令行窗口中，只需要单个的%。例如：在命令行窗口中，运行 set /a num=4%2 ，则结果将显示0，因为4除以2的余数为0。
- 在批处理文件中，需要连续两个百分号，写成%%

```
@echo off
set /a num=4%%2
echo 4除以2的余数为 %num%
pause
```

4、转义符号，如果要显示 % 本身时，需要在前面用%来转义。

```
@echo off
echo 一个百分号：%%
echo 两个百分号：%%%%
echo 三个百分号：%%%%%%
pause
```

:、::

1、以 : 打头的单个的 : 表示该行是一个标签，它之后的内容是一个标签段，如 :test，则表示 :test 之下的内容是标签段，而 test 是这个标签段的名，可以用 goto test、goto :test 跳转到该标签段或用 call :test 调用该子过程

2、连续两个冒号 :: 打头表示该行内容为注释内容，实际上，:: 是个无效的标签名，: 加上空格同样可以起到注释的作用，此时，:: 的功能和注释命令 rem 相同；但是，rem 注释语句中的某些命令符号如重定向符号和管道符号还是会执行，而如果用 :: 来注释的时候，与 :: 同处一行的所有命令或符号直接被命令解释器忽略掉，无形中提高了注释的兼容性和整个程序的执行效率，并且在众多的命令语句中更显得醒目，所以，注释语句推荐使用 :: 的格式。

3、在 set 语句中，: 和 ~ 同时使用时，: 起到截取字符串的功能。假设 set str=abcde，那么，set var=%str:~0,1% 表示截取字符串abcde的第一个字符。

4、: 和 = 同时使用时，起到替换字符串的功能。假设：set str=abc:de，那么，set var=%str:a=1% 则表示把字符串 abc:de 中的 a 替换为 1，set var=%str::=2% 则表示把字符串 abc:de 中的 : 替换为 2。

~

1、用在 set 语句中，和 : 同时使用时，起到截取字符串的功能。

2、用在 set /a 语句中时，它是一元运算符号，表示将操作数字按位取反，例如，set /a num=~1的执行结果是-2，set /a num=~0的结果是-1

3、用在 for 语句中，表示增强 for 的功能，能够提取到更多的信息。

例如：在批处理文件的 for 语句中：

- %%~i 表示去掉第一对外侧引号
- %%~zi 表示获取文件的大小(以字节为单位)
- %%~ni 表示获取文件名
- %%~xi 表示获取扩展名(带点号)
- 它们可以组合使用，如%%~nxi表示获取文件名和后缀名

>、>>

- > 表示用新内容覆盖原文件内容
- >> 表示向原文件追加内容，此时，它们以重定向符号的身份出现
- 如果用在 set /a 语句中，则 > 表示分组，>> 表示逻辑移位

|

一般而言，它以管道符号的身份出现，表示把在它之前的命令或语句的执行结果作为在它之后的命令或语句的处理对象，简而言之，就是把它之前的输出作为它之后的输入。

例如：

- echo abcd | findstr "b"，表示把 echo abcd 的执行结果作为 findstr "b" 的执行对象，也就是在字符串 abcd 中查找 b 字符
- 如果 test.txt 中有 abcd 字符串，则语句 findstr "b" test.txt 也是在字符串 abcd 中查找 b 字符

^

一般而言，^ 以转义字符的身份出现。因为在 cmd 环境中，有些字符具备特殊功能，如 >、>> 表示重定向，| 表示管道，&、&&、|| 表示语句连接。它们都有特定的功能，如果需要把它们作为字符输出的话，需要用 echo ^>、echo ^|、echo ^|^|、echo ^^ 之类的格式来处理

&

一般而言，& 表示两条命令或语句同时执行的意思。如 echo a & echo b，将在屏幕上同时显示 a 和 b 字符。当几条语句含义近似或作用相同且没有先后的顺序之别时，启用 & 符号连接这些语句将会增加程序的可读性

&&、||

这是一对含义截然相反的命令符

- && 表示如果它之前的语句成功执行，将执行它之后的语句
- || 则表示如果它之前的语句执行失败，将执行它之后的语句；在某些场合，它们能替代 if \ else 语句。

例如：

```
@echo off
md test && echo 成功创建文件夹test || echo 创建文件夹test失败
pause
```

效果等同于如下代码：

```
@echo off
md test
if "%errorlevel%"=="0" (echo 成功创建文件夹test) else echo 创建文件夹test失败
pause
```

()

小括号对经常出现在 for 语句和 if 语句中，还有一些特定场合。

在 for 和 if 句中属于语句格式的要求，例如：

1、for %%i in (语句1) do (语句2)

在这条语句中，语句1必须用括号对包围，而语句2的括号对则可视情况予以抛弃或保留：如果语句2是单条语句或用 &、& &、|| 等连接符号连接的多条语句，括号对可以抛弃，如果语句2是有逻辑先后关系的多条语句集合，则必须保留括号对，并且，多条语句必须断行书写；例如：

```
@echo off
for %%i in (a b c) do echo %%i & echo --------
pause
```

也可以改写为：

```
@echo off

for %%i in (a b c) do (
    echo %%i
        &echo --------
)
pause
```

2、if 条件 (语句1) else (语句2)：如果没有 else 部分，则语句1的括号对可有可无；如果有 else 部分，则语句1中的括号对必须保留，此时，语句2中的括号对保留与否，和上一点类似。例如：

```
@echo off
if exist test.txt echo 当前目录下有 test.txt
pause

@echo off
if exist test.txt (echo 当前目录下有 test.txt) else echo 当前目录下没有 test.txt
pause

@echo off
if exist test.txt (echo 当前目录下有test.txt) else (
    echo 当前目录下没有 test.txt
    pause
        cls
        echo 即将创建test.txt文件
    cd . > test.txt && echo 成功创建 test.txt
)
pause
```

3、特定场合下使用括号对，不但可以使代码逻辑清晰，增强可读性，还可能会减少代码量。比如用 echo 语句构造多行文本内容的时候：

```
@echo off
(echo 第一行
 echo 第二行
 echo 第三行
) > test.txt
start test.txt
```

如果不使用括号对的话，则需要使用如下代码：

```
@echo off
echo 第一行 > test.txt
echo 第二行 >> test.txt
echo 第三行 >> test.txt
start test.txt
```

+、-、*、/

- set /a 语句中，这些符号的含义分别为：加、减、乘、除。例如：set /a num=1+2-3*4/5。需要注意的是，这些运算符号遵循数学运算中的优先级顺序：先乘除后加减，有括号的先算括号，并且，直接忽略小数点，因此，刚 才那个算式的结果是1而不是0或0.6。
- set /a num+=1、set /a num-=1、set /a num=*1 和 set /a num/=1，这些表示累加、累减、累乘、累除，步长都是1，展开后的完整写法为：set /a num=num+1、set /a num=num-1、set /a num=num*1 和 set /a num=num/1
- et /a 语句中，变量引用可以忽略百分号对或感叹号对，set /a num=%num%+1 与 set /a num=num+1 等同

equ、neq、lss、leq、gtr、geq

这几个命令符是if语句中常用到的数值比较符号，取自英文的关键字母，具体的含义为

- EQU 等于
- NEQ 不等于
- LSS 小于
- LEQ 小于或等于
- GTR 大于
- GEQ 大于或等于

脚本示例

```
@echo off
rem 默认构建命令

echo 正在编译解决方案...
::编译解决方案（RDC环境变量中指定了msbuild路径，可以直接使用）
msbuild /m "src\二开整体解决方案.sln" /tv:15.0 /p:VisualStudioVersion=15.0
if errorlevel 1 goto fail

echo 正在拷贝配置文件...
xcopy "config\wwwroot"  "src\00-ERP站点\"  /E /Y /R /C /Q

echo 正在编译前端代码...
pushd src\00-ERP站点\node\ :: 切换到 src\00-ERP站点\node\ 目录，并且记录现在目录
call "node_modules_update.cmd"
call "01_产品构建.cmd"
call "02_二开构建.cmd"
popd  :: 切换到之前的目录

exit /b 0

:fail
exit /b 1
```