# BAT脚本

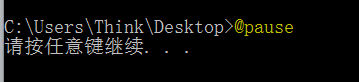
1. 注释：

rem：注释信息。 rem 这是个注释符

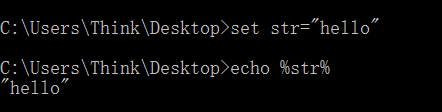
1. @

@: 后面跟一条命令和语句。此命令或语句本身在执行的时候不会显示在屏幕上。

@pause: 控制台屏幕提示：请按任意键继续. . .



1. cls: 清楚屏幕信息
2. %% 和 %
3. %%： 中间带有参数则是变量的引用。%var%



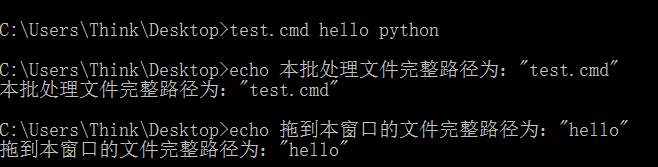
1. %： 参数的引用：%0:本身，%1：表示传入参数的第一位。最大为：%9

执行如下命令：

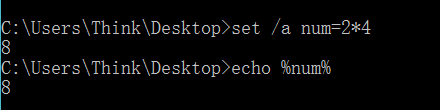
echo 本批处理文件完整路径为："%0"

echo 拖到本窗口的文件完整路径为："%1"

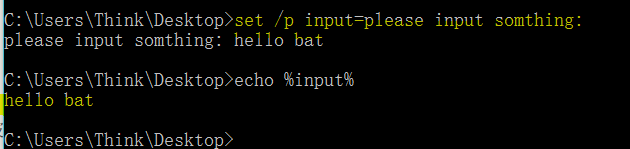
输出的结果:



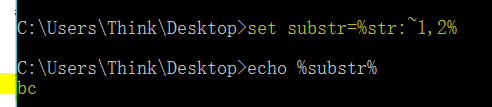
1. /的用法:
2. set /a 后面跟着表达式: set /a 1+2+3

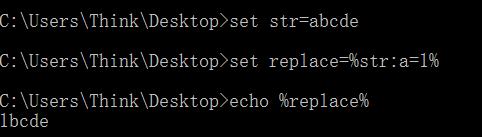


1. set /p: 接收用户输入的信息： set /p info=Please input something:



1. : 和::
2. ：搭配~使用：起到截取的功能，搭配=使用：起到替换的功能.





1. ：： 起到注释的作用，特点是注释所有的内容，不管是重定向或则管道符。

与rem比较：rem注释的语句比如重定向或则管道符还是会被执行。

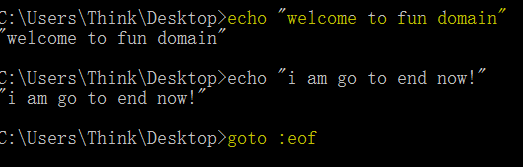
1. :与goto ： 定义作用域和跳转

:fun

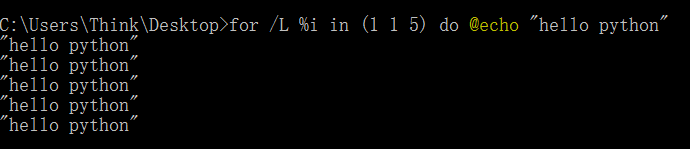
echo “welcome to fun”

goto :eof (如果不加的划，会一直输出信息)

goto :fun (或则: goto fun)



1. 循环
2. for \l %%i in (1,1,5) do @echo “hello python”



1. for \l %%i in (1,1,5) do echo “hello python”

注意：两者输出的区别。

@echo “Hello Python”: 命令在执行的时候都不会显示在屏幕上 (适用于循环)

echo “Hello Python”: 命令在执行的语句都会显示在屏幕上



1. 还有个更好玩的：

For one: 同时打开多个控制台,以下是打开5个。

For \l %%i in (1,1,5) do start cmd

pause

for two: 创建多个文件夹

for \l %%i in (1,1,10) do md %%i

1. @echo 与 @echo off的比较：

相同点： 两者都可以是的执行的命令不在控制台显示。

不同点： @echo: 适用于循环语句中: for /l %%i in (1,25) do @echo %%i

@echo off: 适用于行的声明： @echo off (全局声明)

echo hello world