

Forum:[Hack Forums](#)

Thread: [syntax error unexpected end of file](#)

Exsanity - November 30, 2016, 12:59 am

So i have asked on discords for help as well and figured i'd take it to the forums i read that it might be due to mismatch if and fi or done without a for and all that but i tried putting fis and din't work might of put them in the wrong place
but the exact error i am getting is:
line 38: syntax error: unexpected end of file
I am new to batch (like legit 2 days) so excuse my beginner mistake(s)
My Code:

```
#!/bin/bash
echo "This program is meant only for simple and educational/personal uses"
echo "choose what you want!"
:menu
echo "[1] Check for Certain Port"
echo "[2] Scan all Open Ports"
echo "[3] Scan Tool"
echo "[4] Close the Program"
echo
set /p mc="What would you like to do: "
if %mc%==1 goto 1
if %mc%==2 goto 2
if %mc%==3 goto 3
if %mc%==4 goto 4
if %mc% > 4 || %mc% < 1
    echo Invalid Number Entered you entered %mc% not 1 - 4 try agin
    goto menu
:1
echo Not Set Up Yet
goto menu
:2
echo Not Set Up Yet
goto menu
:3
echo work or some shit
if exist %userprofile%\Desktop\%NSL%_info.txt del %userprofile%\Desktop\%NSL%_info.txt
set /p NSL="Enter IP: "
echo Scanning %NSL% > %userprofile%\Desktop\%NSL%_info.txt
nslookup %NSL%
echo All of this info will be placed on your desktop
wait 9

goto menu
:4
echo Closing...
exit
```

SIMMS7400 - November 30, 2016, 3:32 am

You have bash and batch code in the same file. What is your purpose?

To declare a variable in batch you use:

```
SET VAR1=Hi  
ECHO %HI%
```

In Shell:

```
VAR1=Hi  
ECHO $VAR1
```

I'm going to bet you haven't the slightest clue.

b00st3d - November 30, 2016, 4:06 am

First, put all code in a tag as seen below (remove the space) please.

[code]your script here[/code]

Remove #!/bin/bash this is just for bash scripts in linux

Top line should be as follows to remove the commands and just output what you have echo'd

Code @echo off

This line wont work. Just remove it. If it gets to the line, then an invalid selection was made.

Code if %mc% > 4 || %mc% < 1

instead of wait 9, use timeout 9

Feel free to ask more questions and hang around the forums. Someone here can help you out.

Exsanity - November 30, 2016, 5:49 am

for some reason quotes aren't working on the forums but i did both of your solutions

Indeed i did not know, i guess i was reading the wrong book! Thank you for your correction and excuse my mistake! After doing this change and making all the %mc% and %NSL% etc into just \$mc \$NSL etc i still receive the syntax error

for some reason quotes aren't working on the forums but i did both of your solutions

Done this as well as read above^ and the error persists

New code:

```
@echo off  
echo "This program is meant only for simple and educational/personal uses"  
echo "choose what you want!"  
goto menu  
:menu  
echo "[1] Check for Certain Port"  
echo "[2] Scan all Open Ports"  
echo "[3] Scan Tool"
```

```

echo "[4] Close the Program"
echo
set /p mc="What would you like to do: "
if $mc==1 goto 1
if $mc==2 goto 2
if $mc==3 goto 3
if $mc==4 goto 4

:1
echo Not Set Up Yet
goto menu
:2
echo Not Set Up Yet
goto menu
:3
echo work or some shit
if exist $userprofile\Desktop\NSL$_info.txt del $userprofile\Desktop\NSL$_info.txt
set /p NSL="Enter IP: "
echo Scanning NSL > $userprofile\Desktop\NSL_info.txt
nslookup $NSL
echo All of this info will be placed on your desktop
timeout 9

goto menu
:4
echo Closing...
exit

```

For some reason quotes and anything BB such as spoilers and code tags etc won't work!

b00st3d - November 30, 2016, 4:34 pm

(Nov 30, 2016 06:49 AM)Exsanity Wrote: [\[CITE\]](#) ...

You are mixing up your scripting enviroments. Is this for linux or windows?

Nevermind the error you mention is definitely a linux error. I'm downloading linux mint to test on but I re-wrote your script as a shell script. It doesn't want to run in the windows 10 bash for some reason.

OK this is verified working on linux. Take note of the comments I used and the layout. Shell scripting likes to work similar to programming languages with functions called by the main script. In this case I wrote everything into it's own function then called the menu at the bottom of the script. All functions point back to the menu when the complete. There are some logic errors in the scan tool that I left in there. Logic errors means it will still run but it probably wont function as expected ie. you check for the existance of a file with a variable in the filename that is not defined. It will of course not exist unless you run the scan tool twice during the same instance of running the script.

```
Code #! /bin/bash
```

```
echo "This program is meant only for simple and educational/personal uses"
```

```
echo "choose what you want!"
```

```

#menu
menu() {
clear
echo "[1] Check for Certain Port"
echo "[2] Scan all Open Ports"
echo "[3] Scan Tool"
echo "[4] Close the Program"
echo
echo -en "What would you like to do: "

#read keyboard input into the variable mc
read mc

case $mc in
    1) singlePort ;;
    2) allPorts ;;
    3) scanTool ;;
    4) exit ;;
    *) echo;echo "Invalid selection made";sleep 2;clear;menu ;;
esac
}

singlePort() {
    echo "Not set up yet"
    sleep 3
    menu
}

allPorts() {
    echo "Not set up yet"
    sleep 3
    menu
}

scanTool() {
    echo "work or some shit"
    #PROBLEM HERE IS NSL IS NOT DEFINED AT RUNTIME FOR THIS FUNCTION
    if [ -f ~/Desktop/${NSL}_info.txt ]; then rm ~/Desktop/${NSL}_info.txt
    fi
    echo -en "Enter IP: "
    read NSL

    #WHY ECHO SCANNING NSL TO A FILE?
    echo "Scanning ${NSL}" > ~/Desktop/${NSL}_info.txt

    #WHY NOT ECHO THIS TO A FILE?
    nslookup $NSL
    sleep 9
    menu
}

#launch the menu

```

