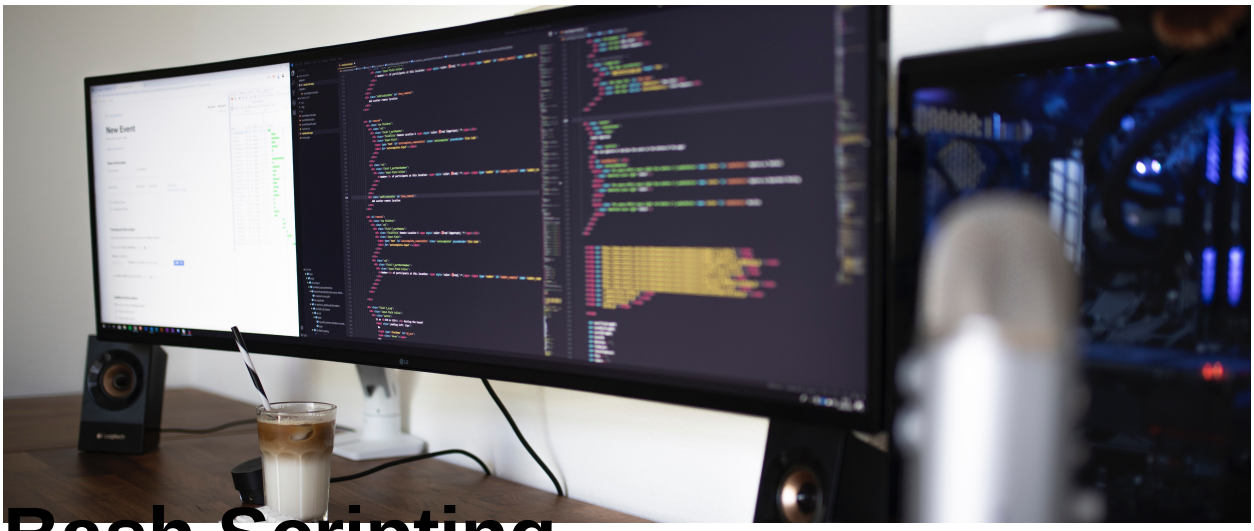


# Bash Scripting

- stands for Bourne Again Shell.
- easy commands
- most used shell language
- stands the test of time
- lightweight and always available
- is not mutually exclusive

### Disadvantages

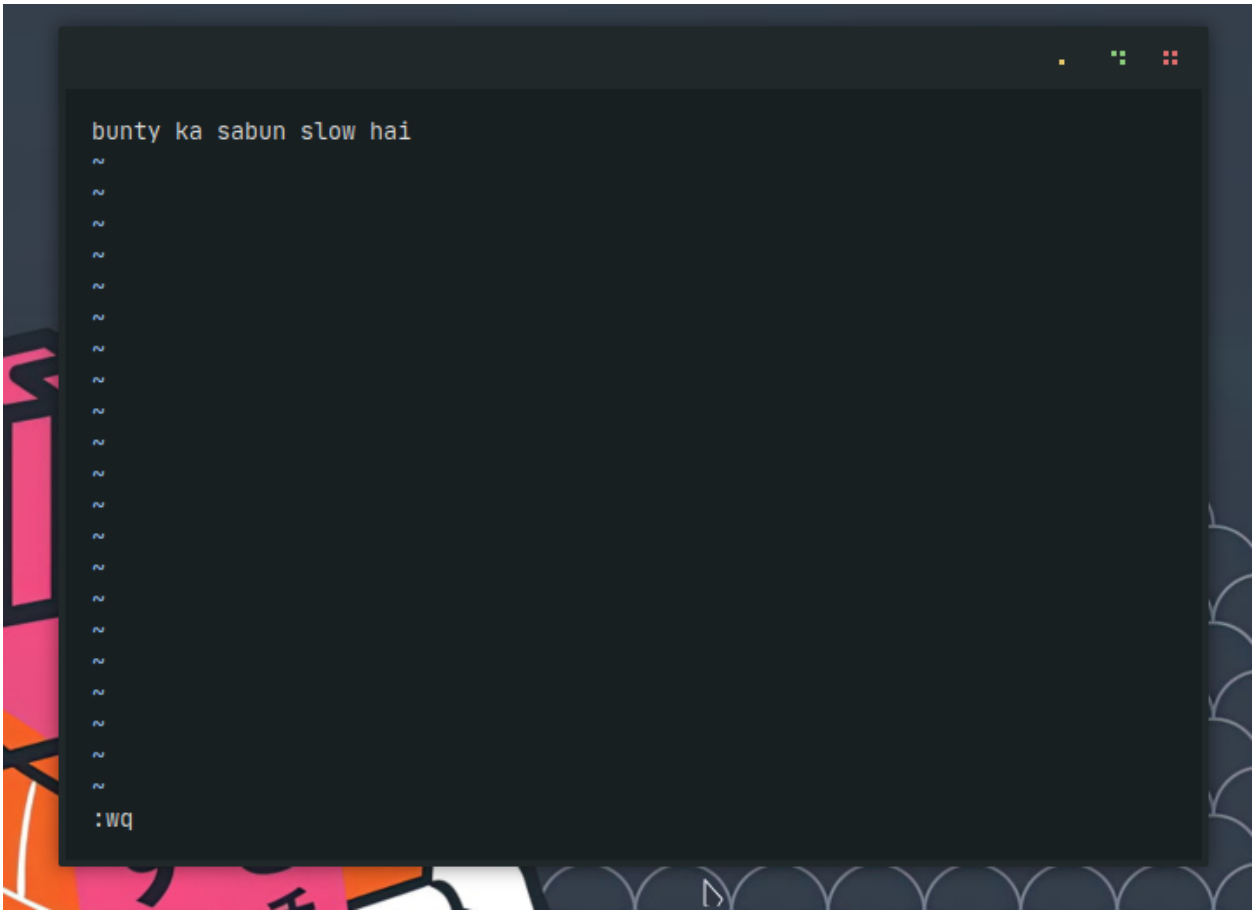
- lacks some features
- no OOP
- difficult syntax compared to python
- newer tools like Ansible

## Basic Commands

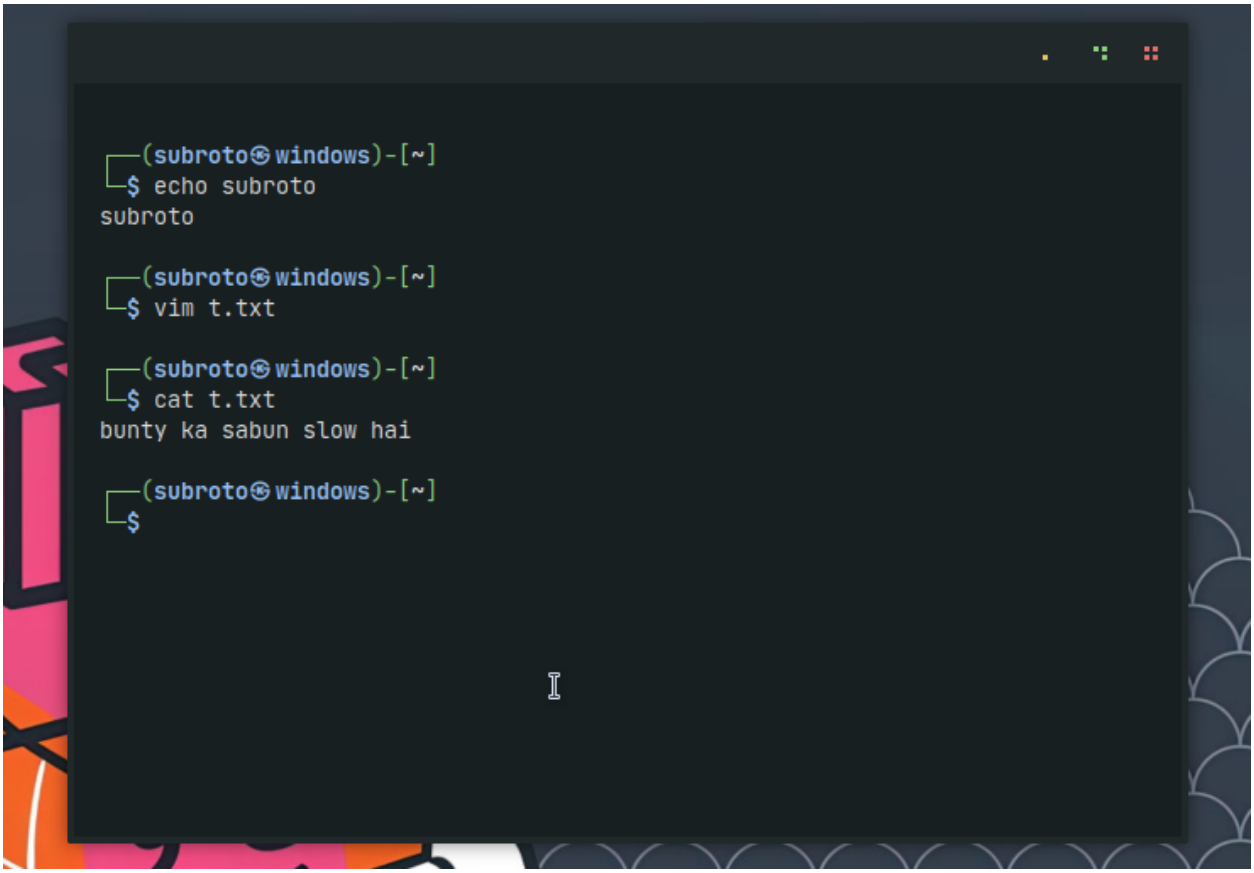
```
┌──(subroto㉿windows)-[~]
└─$ echo subroto
subroto

┌──(subroto㉿windows)-[~]
└─$ vim t.txt_
```

echo command prints it argument , here subroto is the argument so it is returned.

then we have created a text file in vim editor using vim t.txt



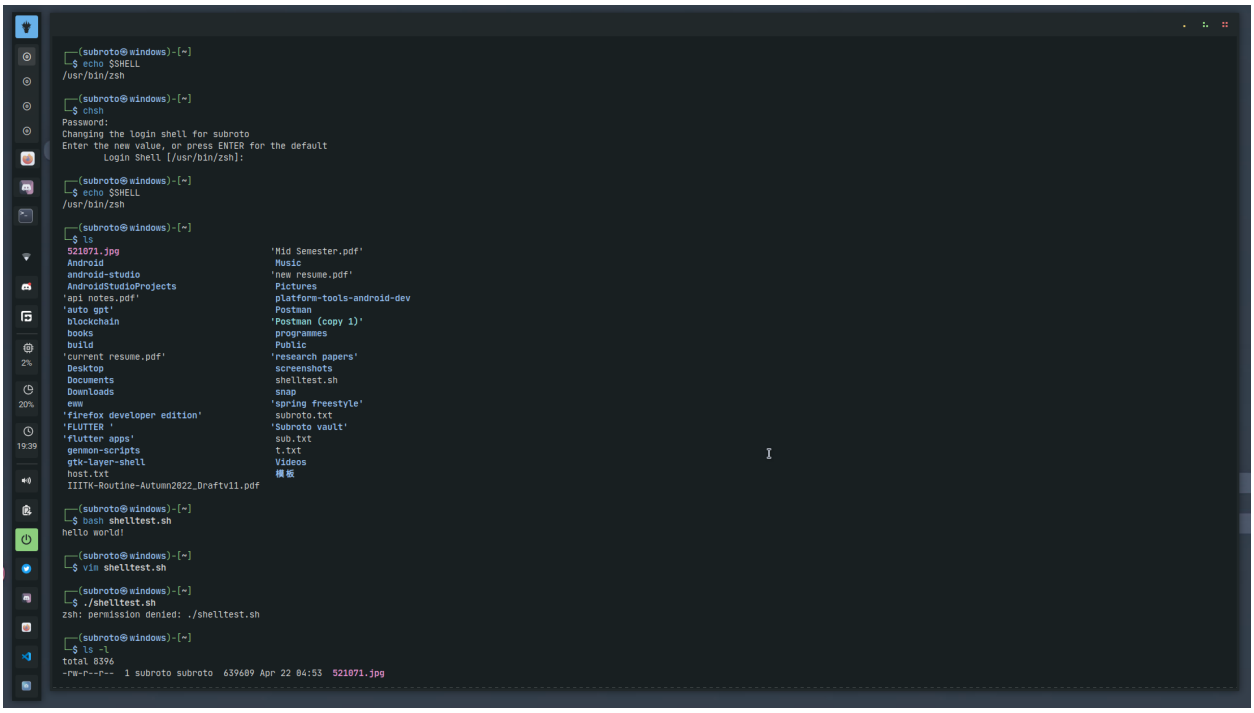
inside the vim editor press “ i ”  to enter the writing mode, then press escape to come back to command mode, then type :wq to save and exit

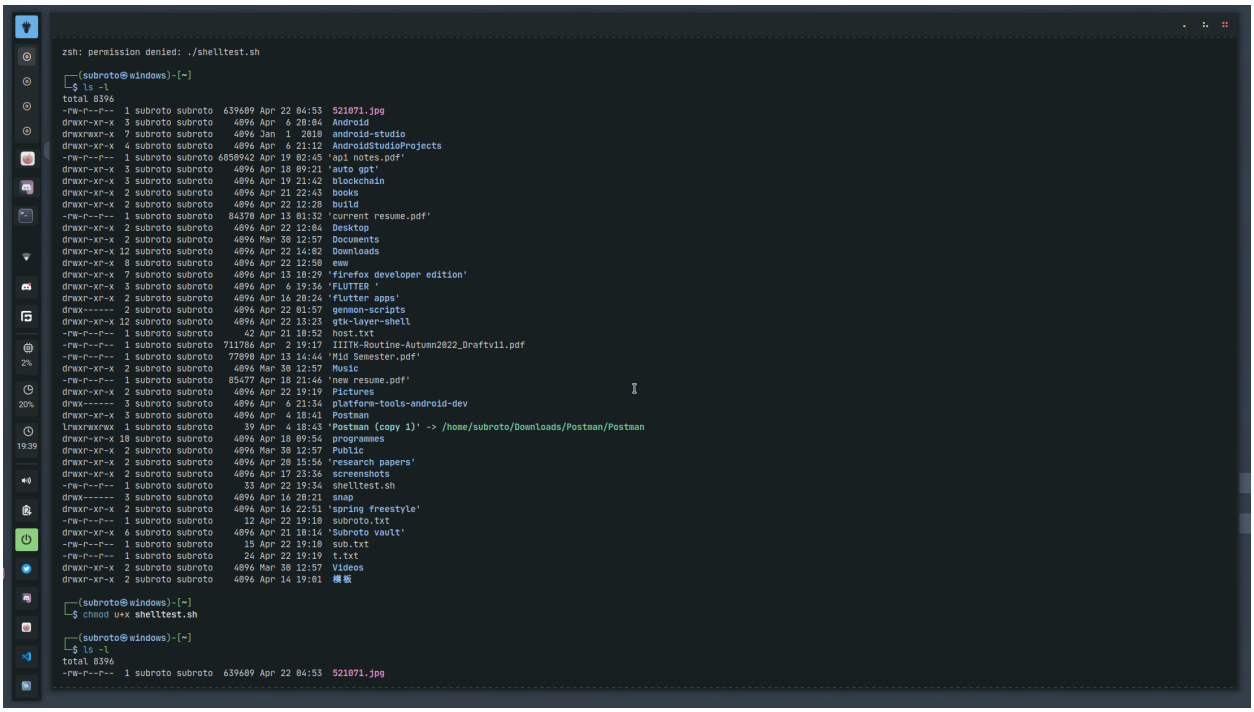


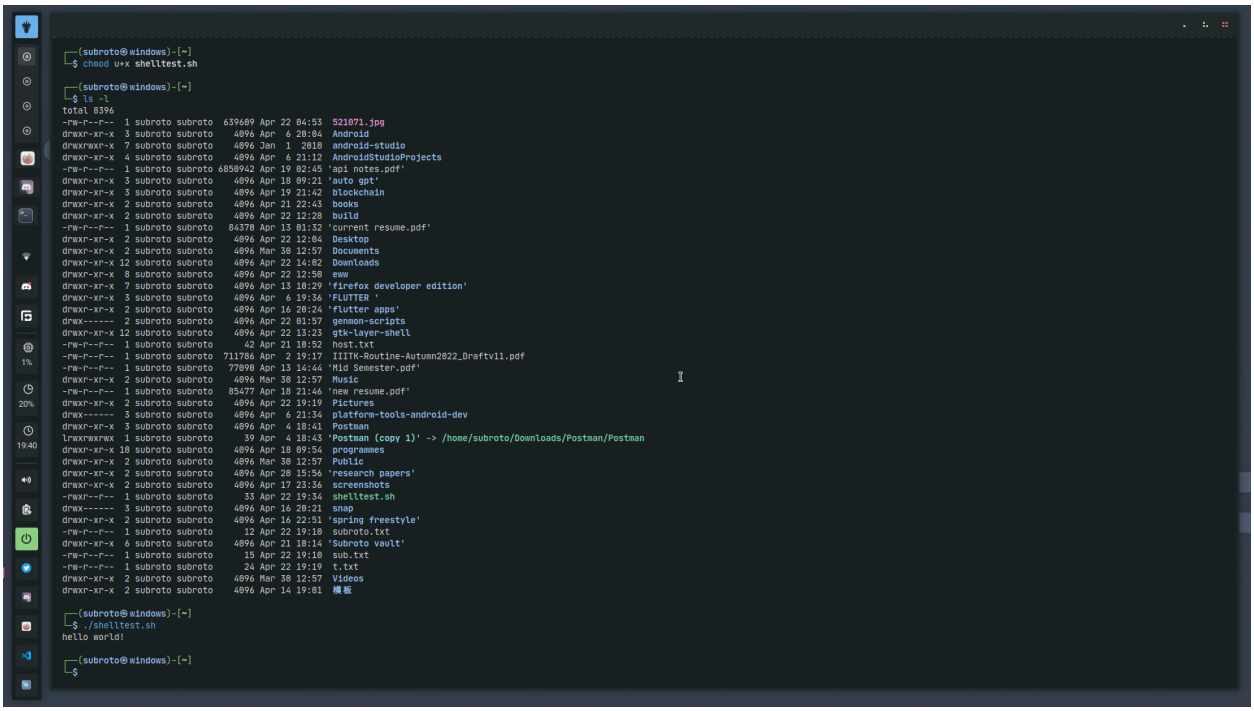

then after our textfile is saved , we use the cat command to print the contents of the file.

to give our script executable permission

```
chmod u+x filename.sh
```


(subroto㉿windows)-[~]
$ echo $SHELL
/usr/bin/zsh
(subroto㉿windows)-[~]
$ chsh
Password:
Changing the login shell for subroto
Enter the new value, or press ENTER for the default
Login Shell [/usr/bin/zsh]:
(subroto㉿windows)-[~]
$ echo $SHELL
/usr/bin/zsh
(subroto㉿windows)-[~]
$ ls
521071.jpg
Android
android-studio
AndroidStudioProjects
'api notes.pdf'
'auto gpt'
blockchain
books
build
'current resume.pdf'
Desktop
Documents
Downloads
eww
'firefox developer edition'
'FLUTTER '
'flutter apps'
genmon-scripts
gtk-layer-shell
host.txt
IIITK-Routine-Autumn2022_Draftv11.pdf
'Mid Semester.pdf'
Music
'new resume.pdf'
Pictures
platform-tools-android-dev
Postman
'Postman (copy 1)'
programmes
Public
'research papers'
screenshots
shelltest.sh
snap
'spring freestyle'
subroto.txt
'Subroto vault'
sub.txt
t.txt
Videos
模板
(subroto㉿windows)-[~]
$ bash shelltest.sh
hello world!
(subroto㉿windows)-[~]
$ vim shelltest.sh
(subroto㉿windows)-[~]
$ ./shelltest.sh
zsh: permission denied: ./shelltest.sh
(subroto㉿windows)-[~]
$ ls -l
total 8396
-rw-r--r-- 1 subroto subroto 639609 Apr 22 04:53 521071.jpg


zsh: permission denied: ./shelltest.sh
(subroto㉿windows)-[~]
$ ls -l
total 8396
-rw-r--r-- 1 subroto subroto 639609 Apr 22 04:53 521071.jpg
drwxr-xr-x 3 subroto subroto 4096 Apr 6 20:04 Android
drwxrwxr-x 7 subroto subroto 4096 Jan 1 2010 android-studio
drwxr-xr-x 4 subroto subroto 4096 Apr 6 21:12 AndroidStudioProjects
-rw-r--r-- 1 subroto subroto 6858942 Apr 19 02:45 'api notes.pdf'
drwxr-xr-x 3 subroto subroto 4096 Apr 18 09:21 'auto gpt'
drwxr-xr-x 3 subroto subroto 4096 Apr 19 21:42 blockchain
drwxr-xr-x 2 subroto subroto 4096 Apr 21 22:43 books
drwxr-xr-x 2 subroto subroto 4096 Apr 22 12:28 build
-rw-r--r-- 1 subroto subroto 84378 Apr 13 01:32 'current resume.pdf'
drwxr-xr-x 2 subroto subroto 4096 Apr 22 12:04 Desktop
drwxr-xr-x 2 subroto subroto 4096 Mar 30 12:57 Documents
drwxr-xr-x 12 subroto subroto 4096 Apr 22 14:02 Downloads
drwxr-xr-x 8 subroto subroto 4096 Apr 22 12:50 eww
drwxr-xr-x 7 subroto subroto 4096 Apr 13 10:29 'firefox developer edition'
drwxr-xr-x 3 subroto subroto 4096 Apr 6 19:36 'FLUTTER '
drwxr-xr-x 2 subroto subroto 4096 Apr 16 20:24 'flutter apps'
drwx------ 2 subroto subroto 4096 Apr 22 01:57 genmon-scripts
drwxr-xr-x 12 subroto subroto 4096 Apr 22 13:23 gtk-layer-shell
-rw-r--r-- 1 subroto subroto 42 Apr 21 18:52 host.txt
-rw-r--r-- 1 subroto subroto 711786 Apr 2 19:17 IIITK-Routine-Autumn2022_Draftv11.pdf
-rw-r--r-- 1 subroto subroto 77098 Apr 13 14:44 'Mid Semester.pdf'
drwxr-xr-x 2 subroto subroto 4096 Mar 30 12:57 Music
-rw-r--r-- 1 subroto subroto 85477 Apr 18 21:46 'new resume.pdf'
drwxr-xr-x 2 subroto subroto 4096 Apr 22 19:19 Pictures
drwx------ 3 subroto subroto 4096 Apr 6 21:34 platform-tools-android-dev
drwxr-xr-x 3 subroto subroto 4096 Apr 4 18:41 Postman
lrwxrwxrwx 1 subroto subroto 39 Apr 4 18:43 'Postman (copy 1)' -> /home/subroto/Downloads/Postman/Postman
drwxr-xr-x 10 subroto subroto 4096 Apr 18 09:54 programmes
drwxr-xr-x 2 subroto subroto 4096 Mar 30 12:57 Public
drwxr-xr-x 2 subroto subroto 4096 Apr 20 15:56 'research papers'
drwxr-xr-x 2 subroto subroto 4096 Apr 17 23:36 screenshots
-rw-r--r-- 1 subroto subroto 33 Apr 22 19:34 shelltest.sh
drwx------ 3 subroto subroto 4096 Apr 16 20:21 snap
drwxr-xr-x 2 subroto subroto 4096 Apr 16 22:51 'spring freestyle'
-rw-r--r-- 1 subroto subroto 12 Apr 22 19:18 subroto.txt
drwxr-xr-x 6 subroto subroto 4096 Apr 21 18:14 'Subroto vault'
-rw-r--r-- 1 subroto subroto 15 Apr 22 19:18 sub.txt
-rw-r--r-- 1 subroto subroto 24 Apr 22 19:19 t.txt
drwxr-xr-x 2 subroto subroto 4096 Mar 30 12:57 Videos
drwxr-xr-x 2 subroto subroto 4096 Apr 14 19:01 模板
(subroto㉿windows)-[~]
$ chmod u+x shelltest.sh
(subroto㉿windows)-[~]
$ ls -l
total 8396
-rw-r--r-- 1 subroto subroto 639609 Apr 22 04:53 521071.jpg



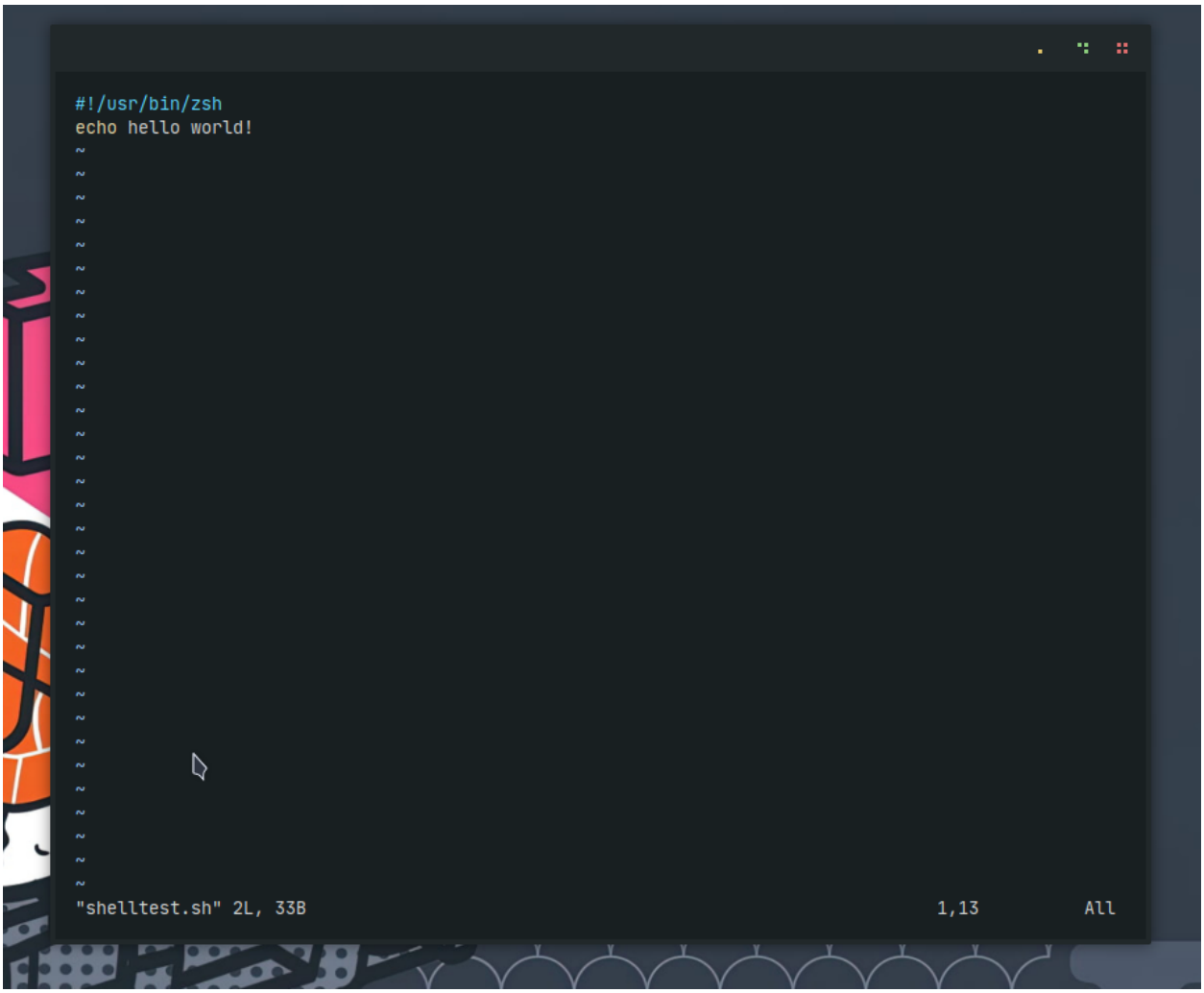

the command ls -l

lists the files with the permissions it is granted

after performing all this we finally can run the ./filename.sh command to run our scripts

also we have to put the location obtained from the command echo $SHELL

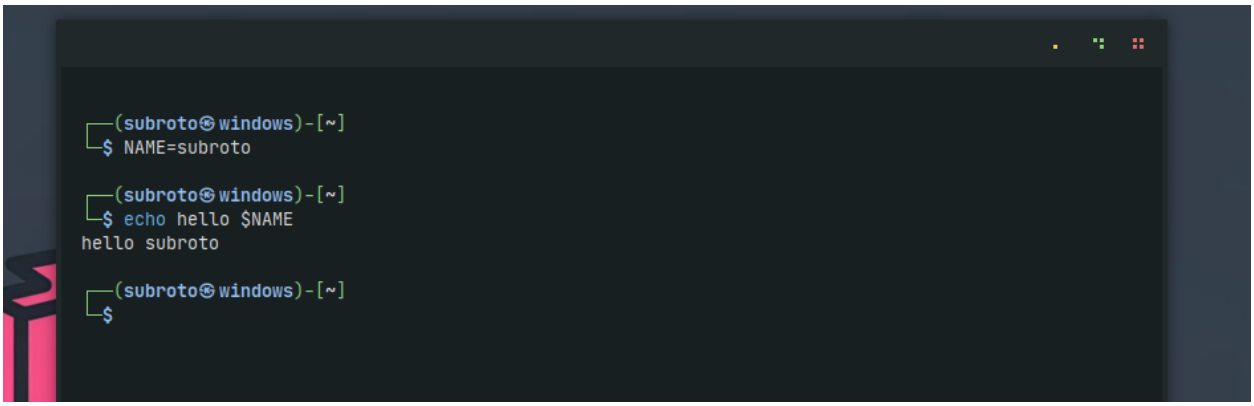
in the file.



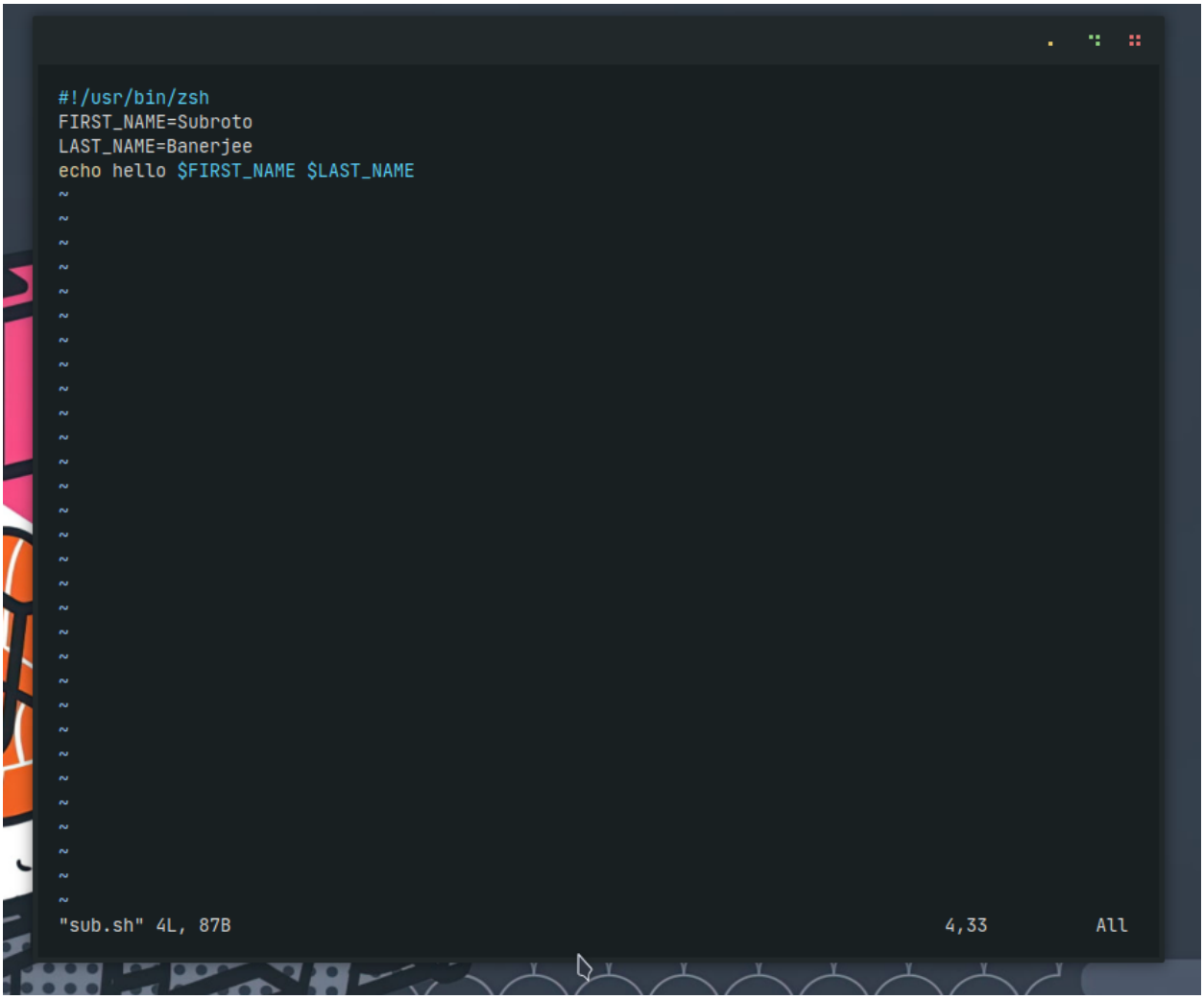

the topmost command after # is the one we're talking about. It shows from where the file has to be run.

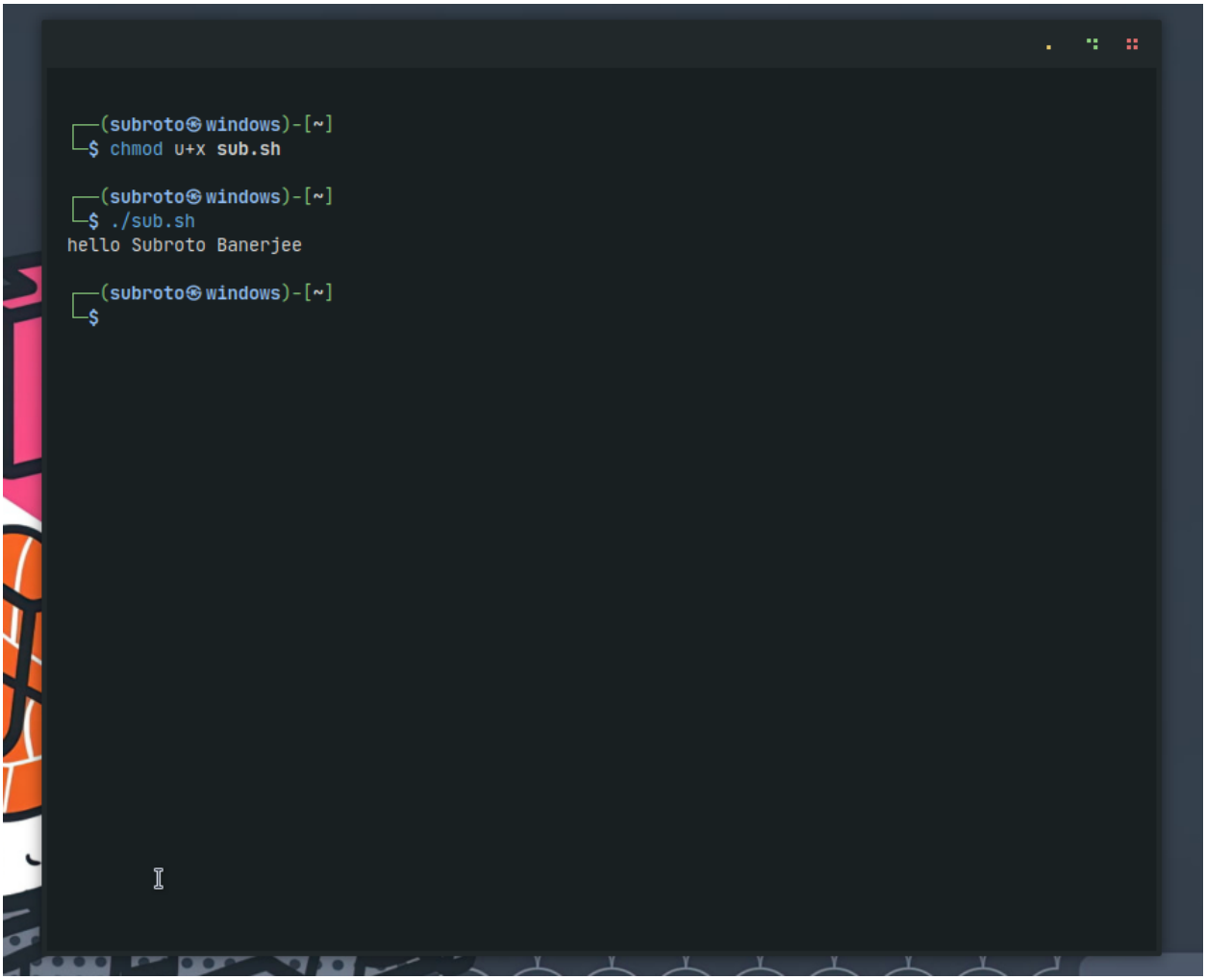
## Variables

simple variables



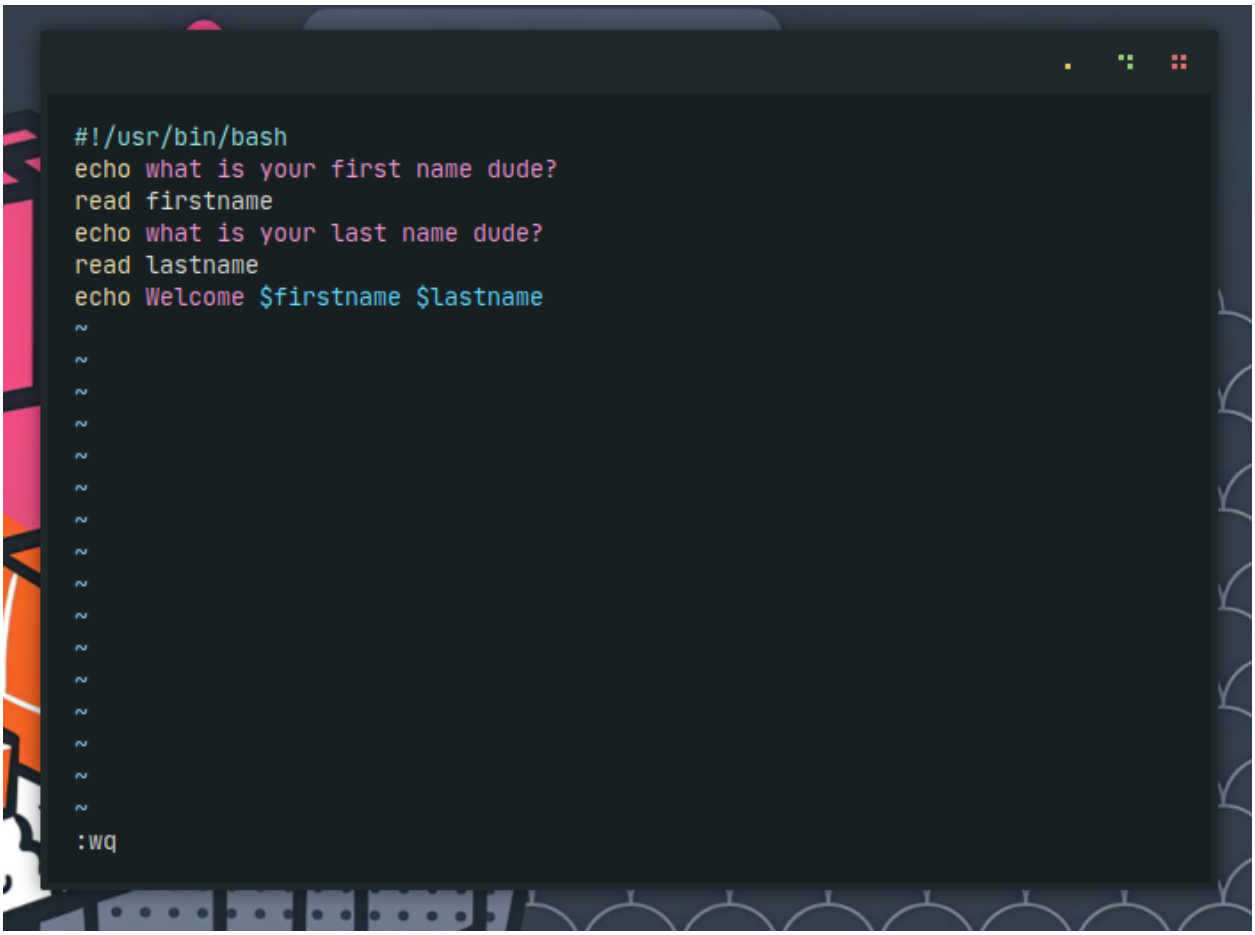

using variables in a script and printing them

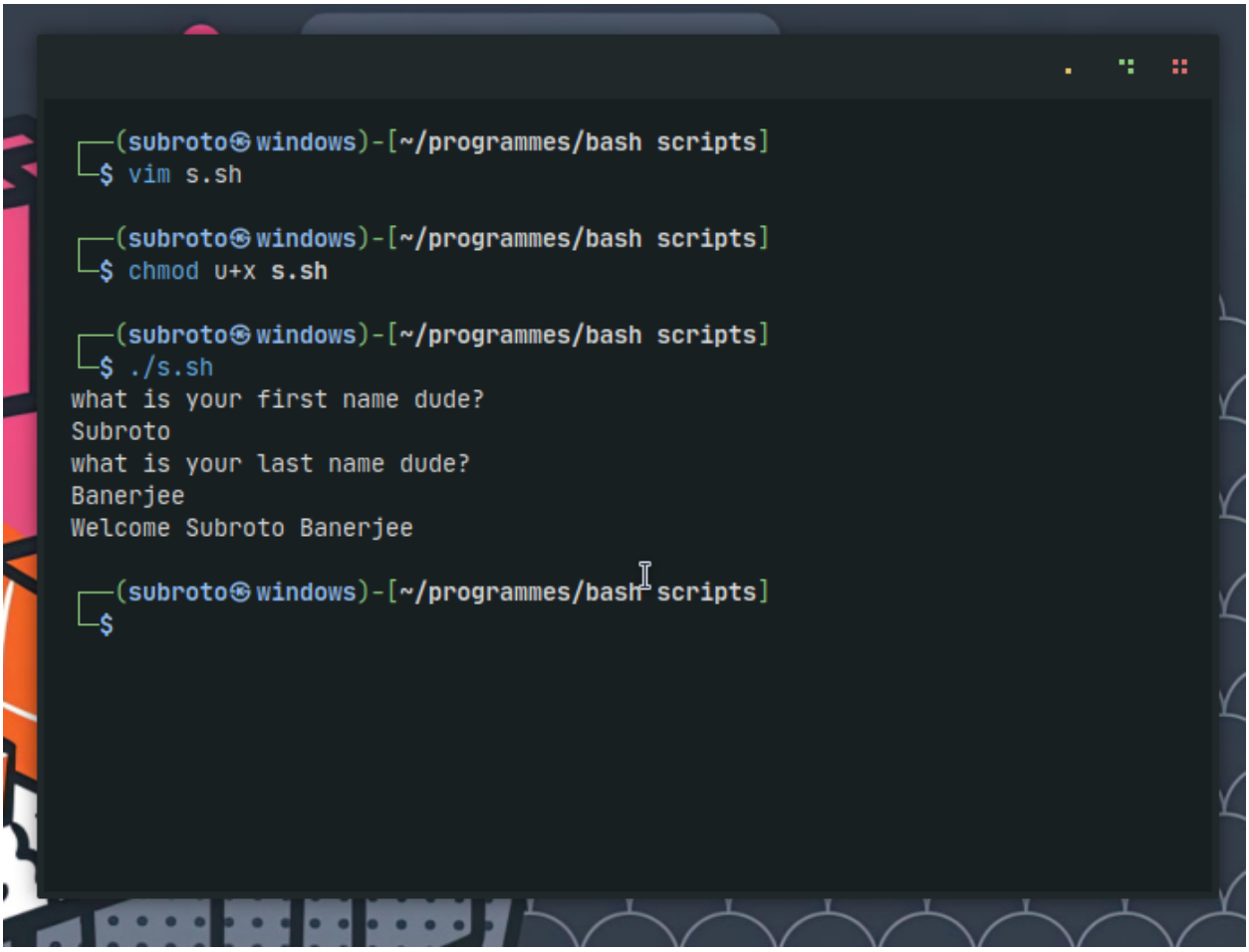





## Getting input from user

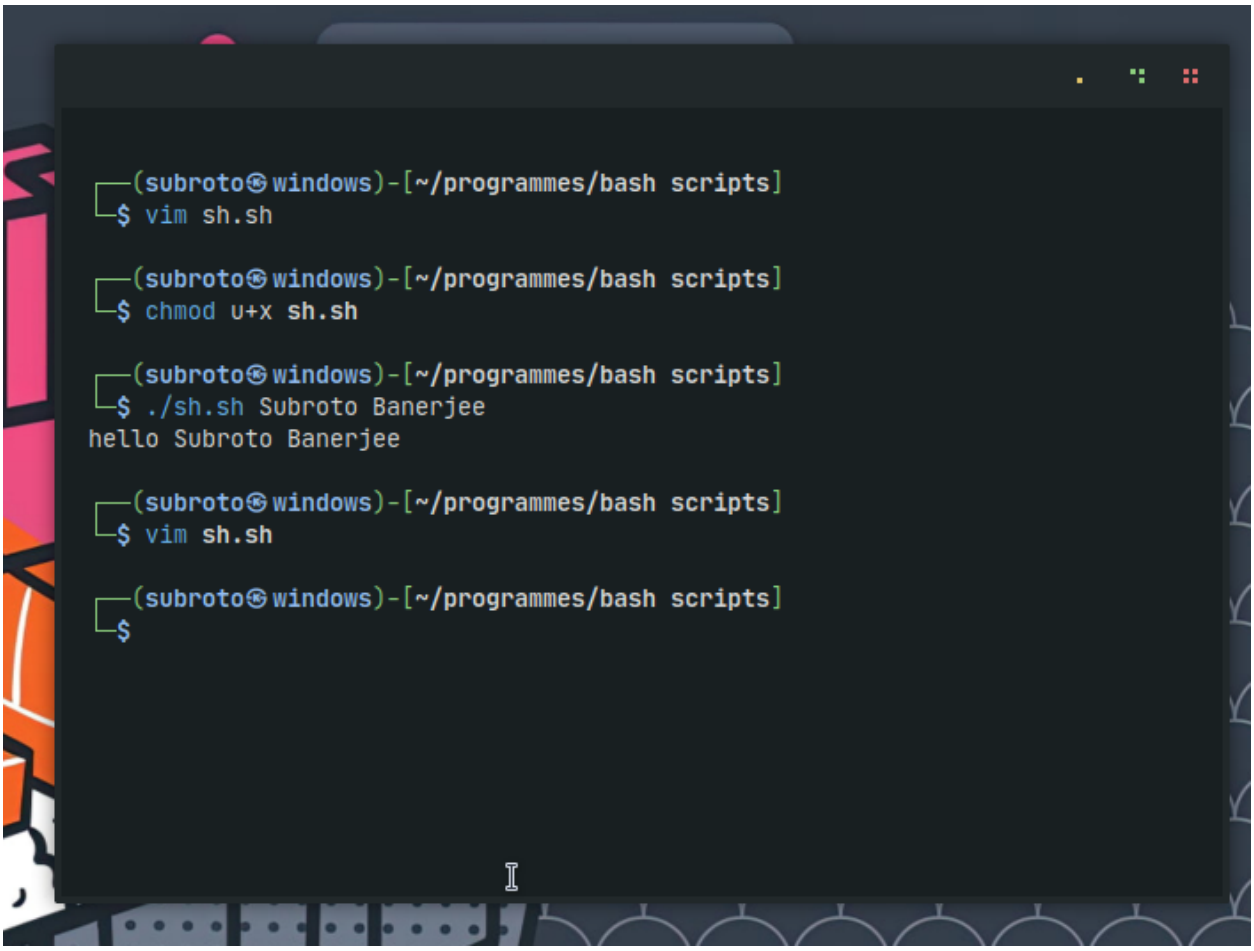
first create a script and :


#!/usr/bin/bash
echo what is your first name dude?
read firstname
echo what is your last name dude?
read lastname
echo Welcome $firstname $lastname
:wq




## Positional Arguements

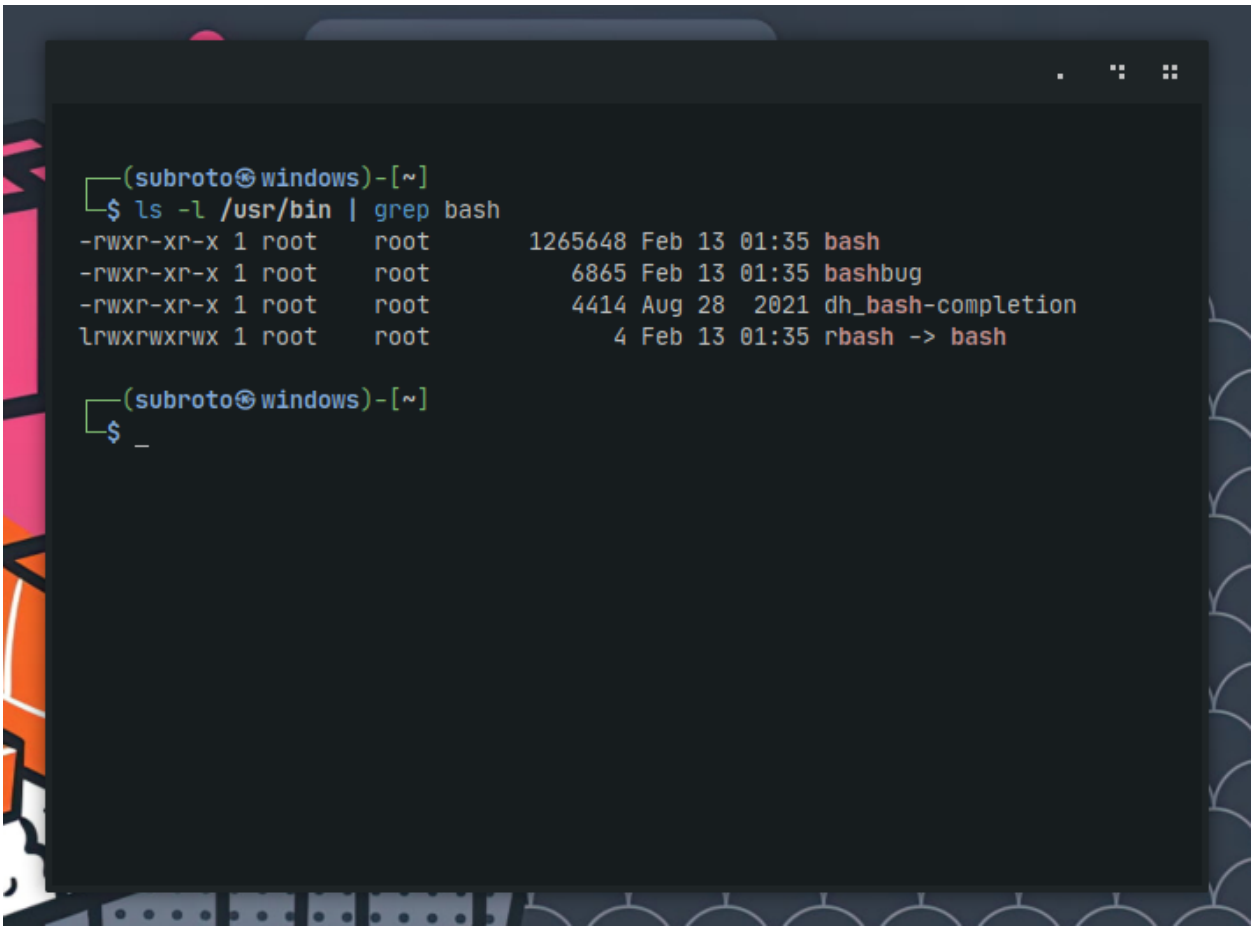

#!/usr/bin/bash
echo hello $1 $2
"sh.sh" 2L, 33B
2,16
All




## Piping

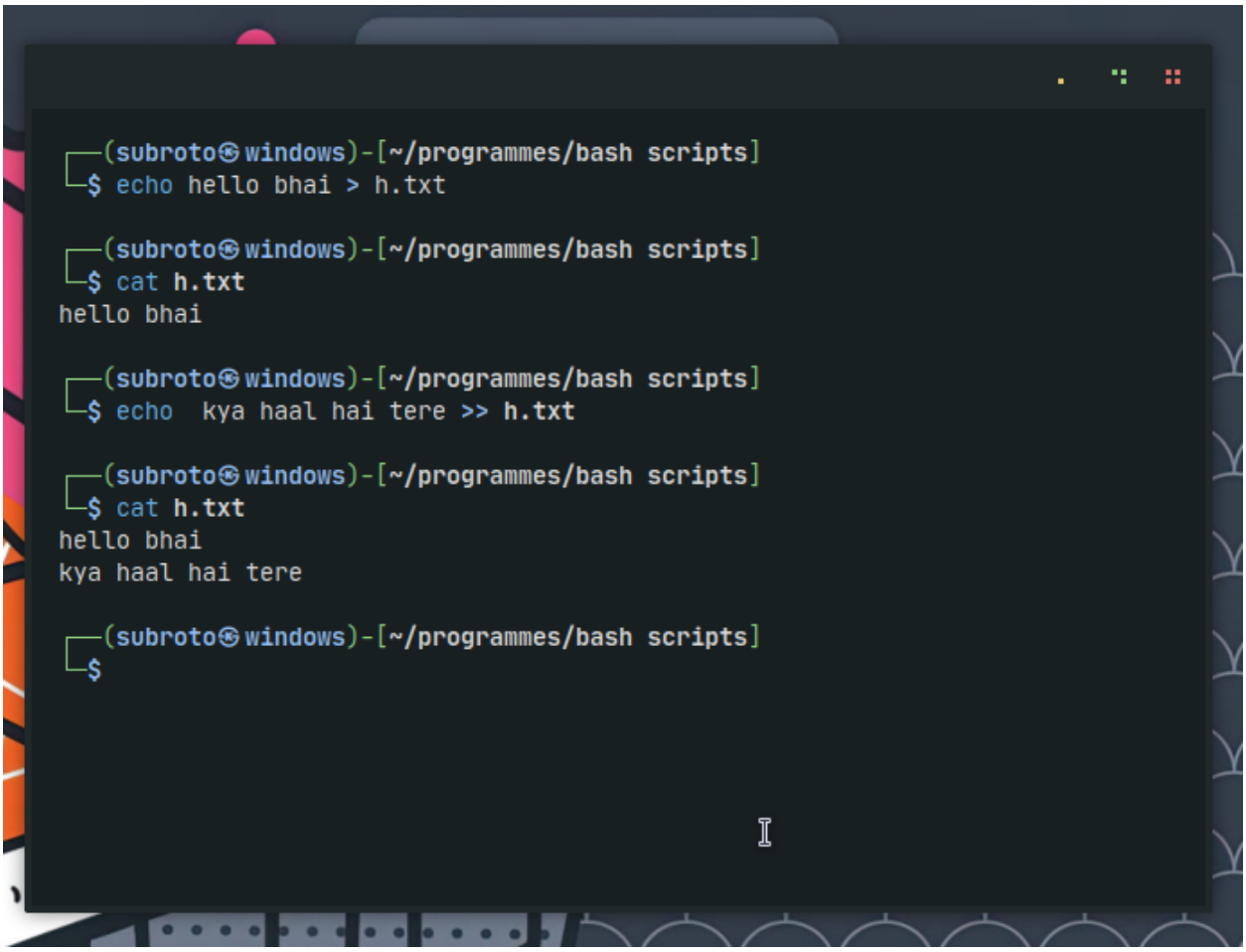
doing something with the output

eg: filtering the output




this filters out only the files having bash term in them in the directory /usr/bin

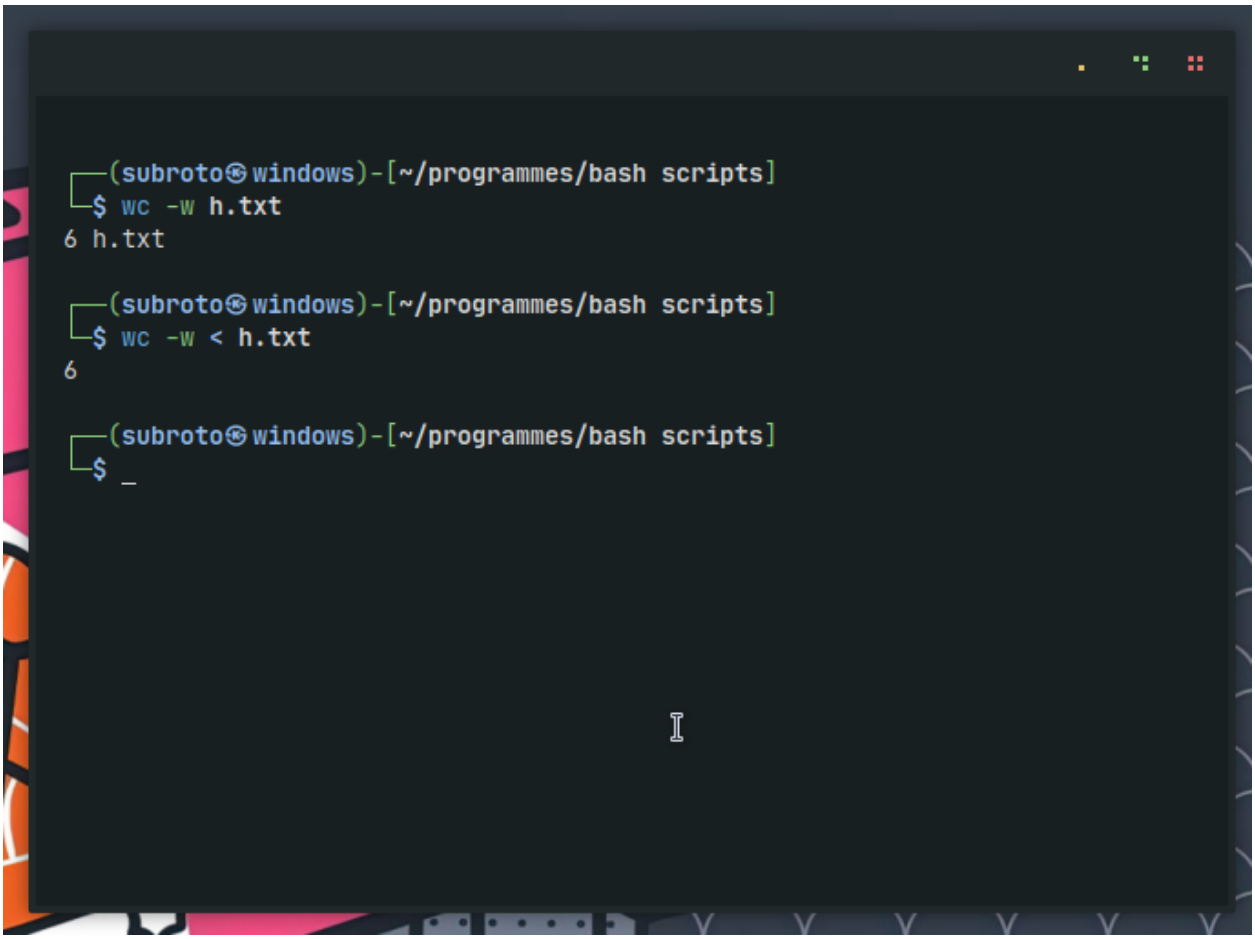
## Writing to a file




so > command overwrites whereas >> appends

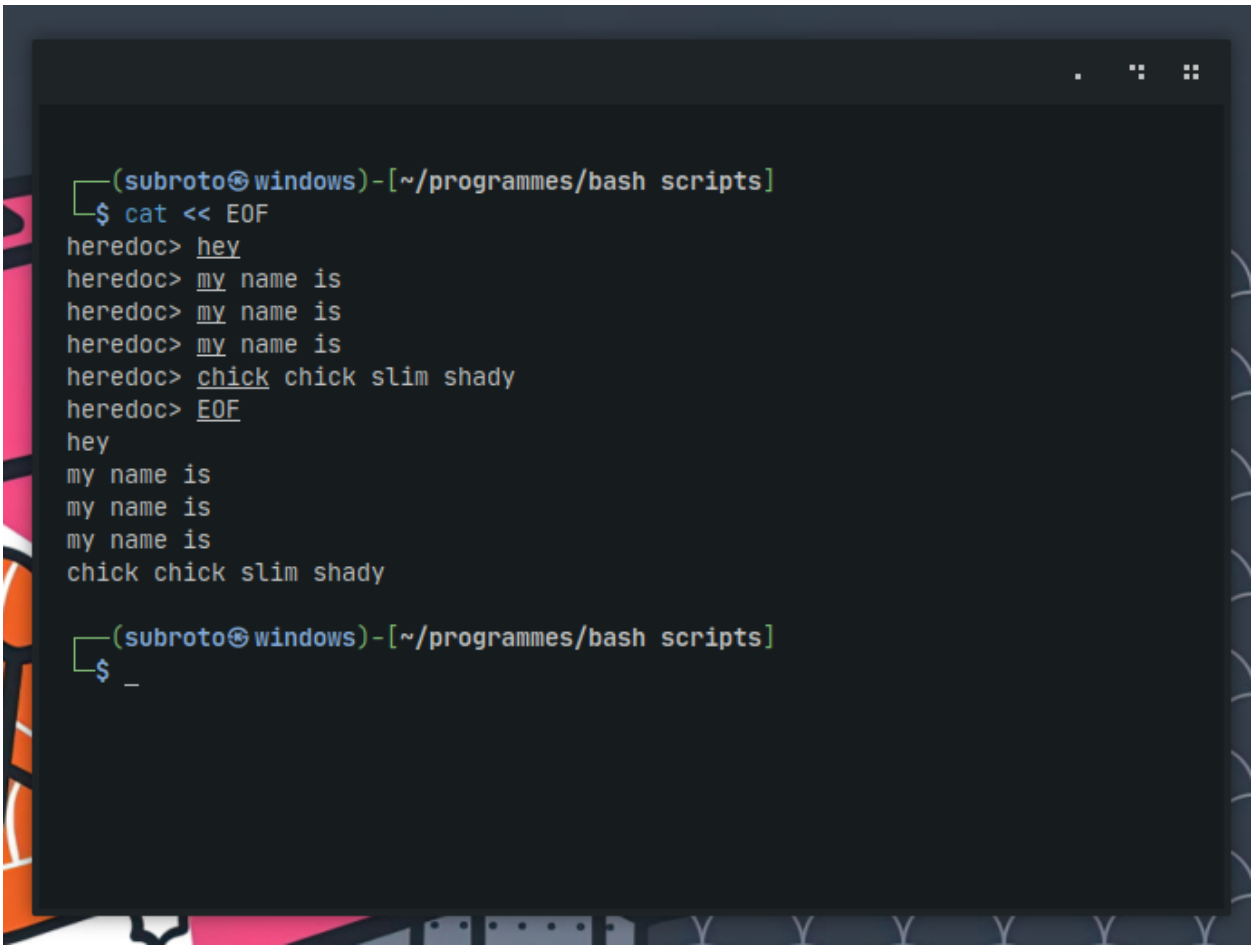
## Getting input from text files

< command



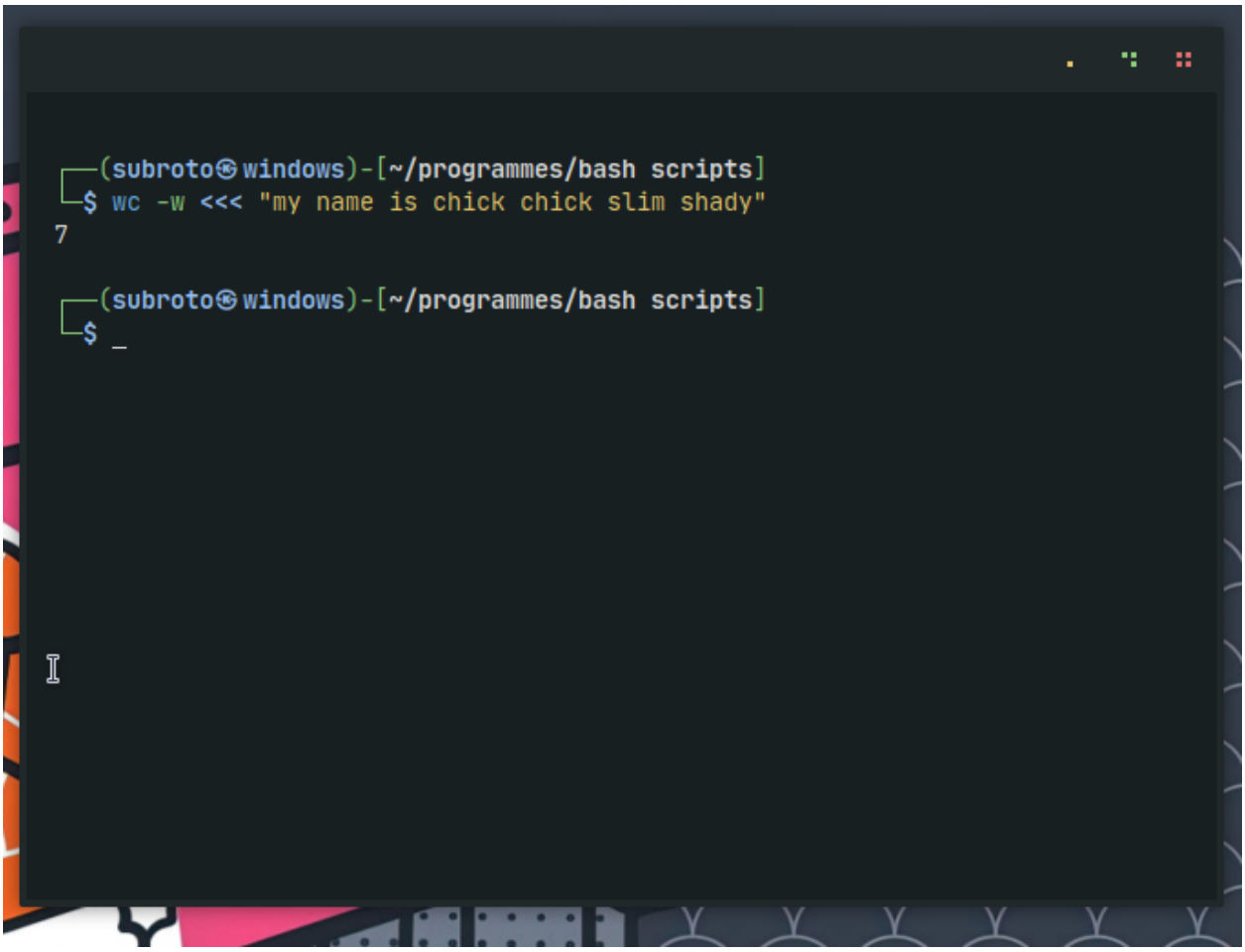

<< command :

gives the text between first occurence and last occurrence of the entered word




<<< command:

feeds a string into a command and the string should be between double quotes only “ ”.

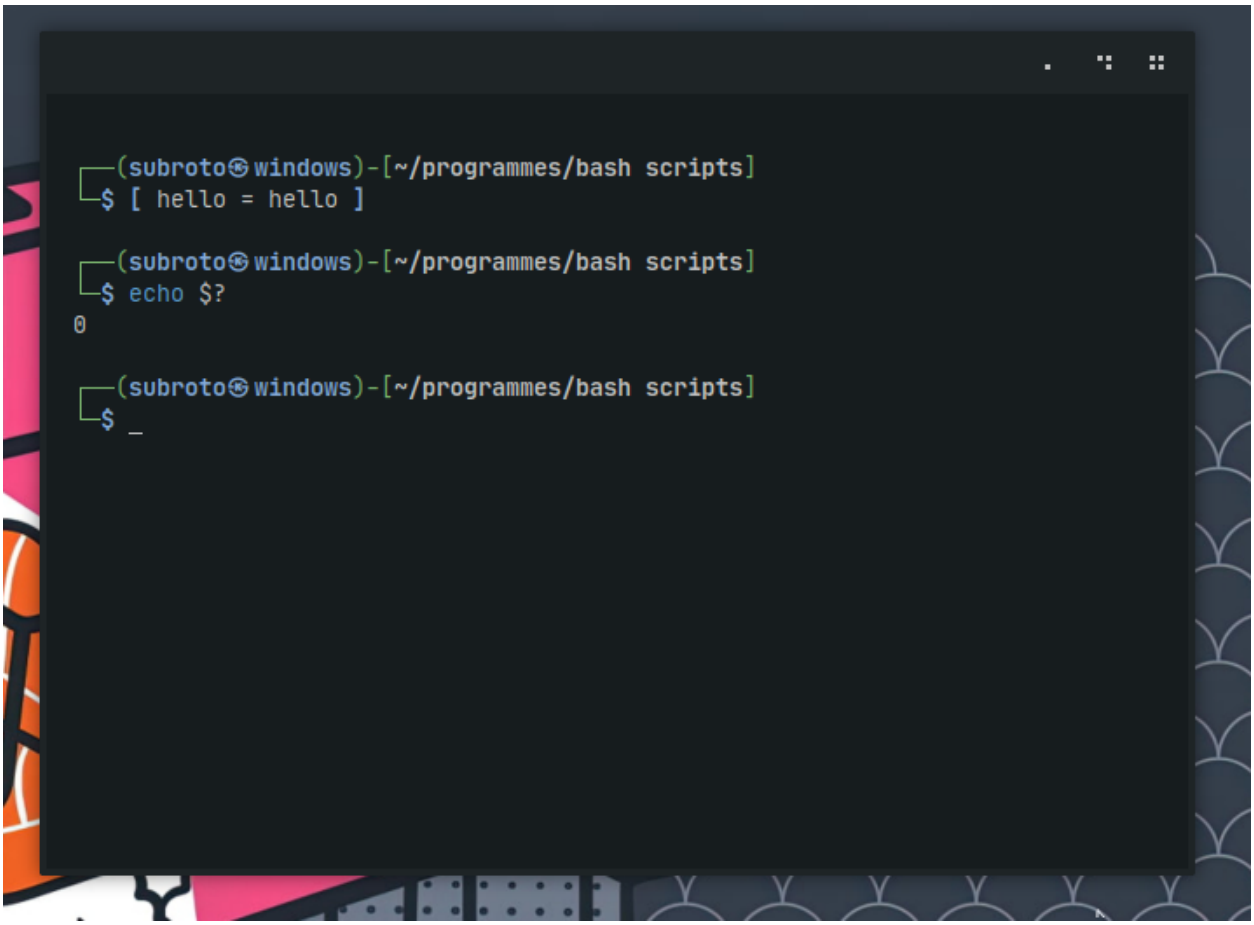



here it returned the word count of the entered string as we used the command wc -w.

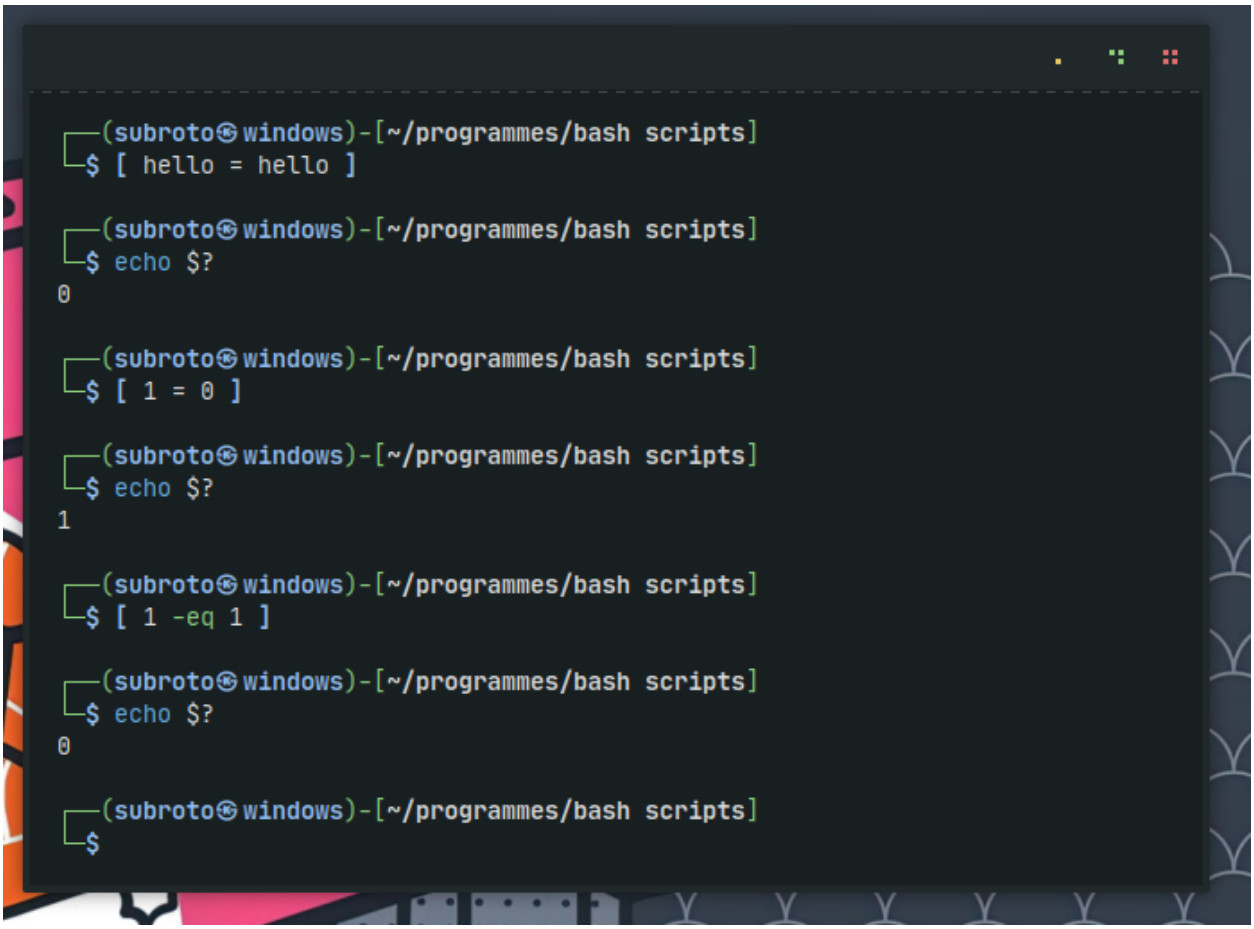
## Test Operators

helps finding whether something is equal or not, comparing different values etc etc.

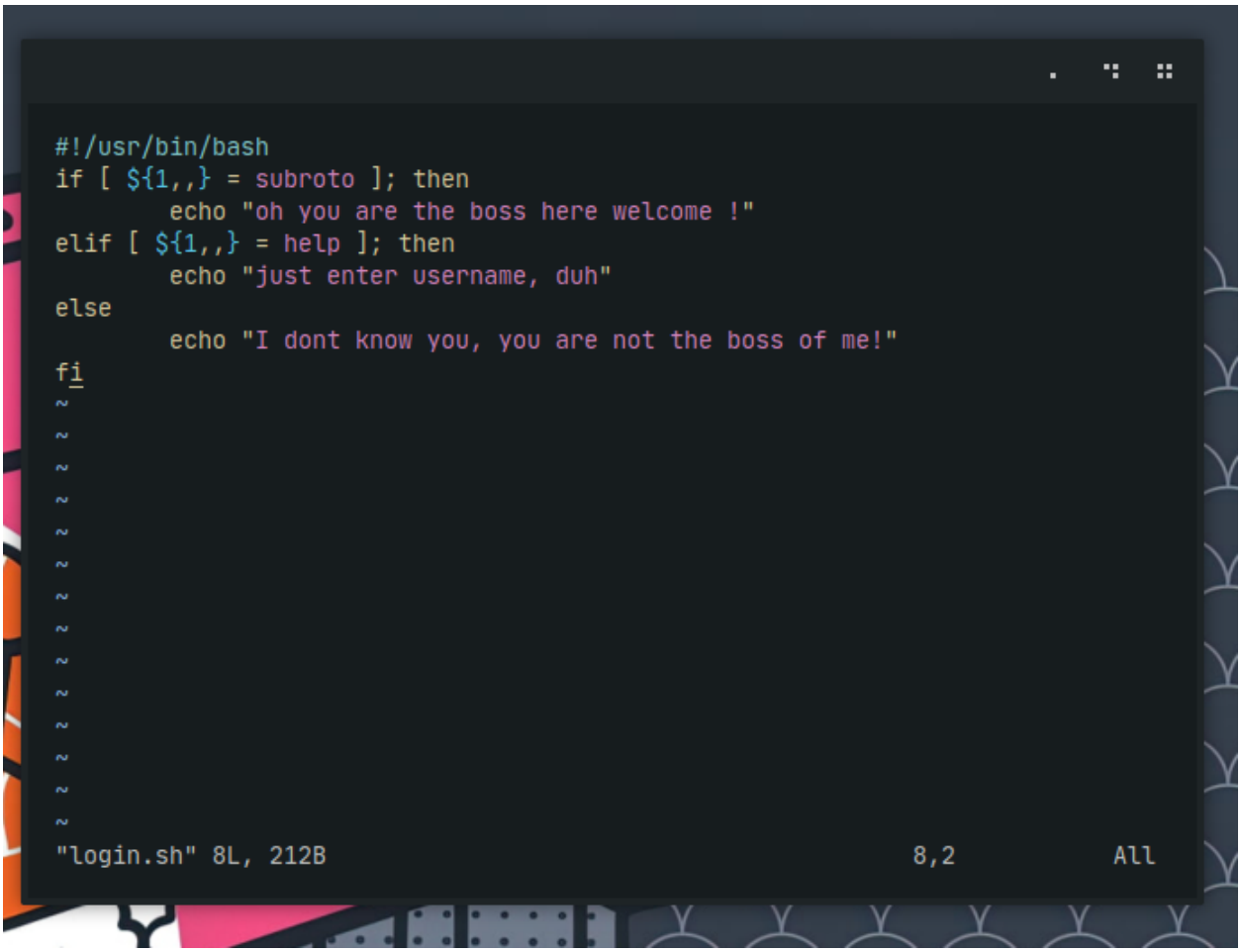
eg:

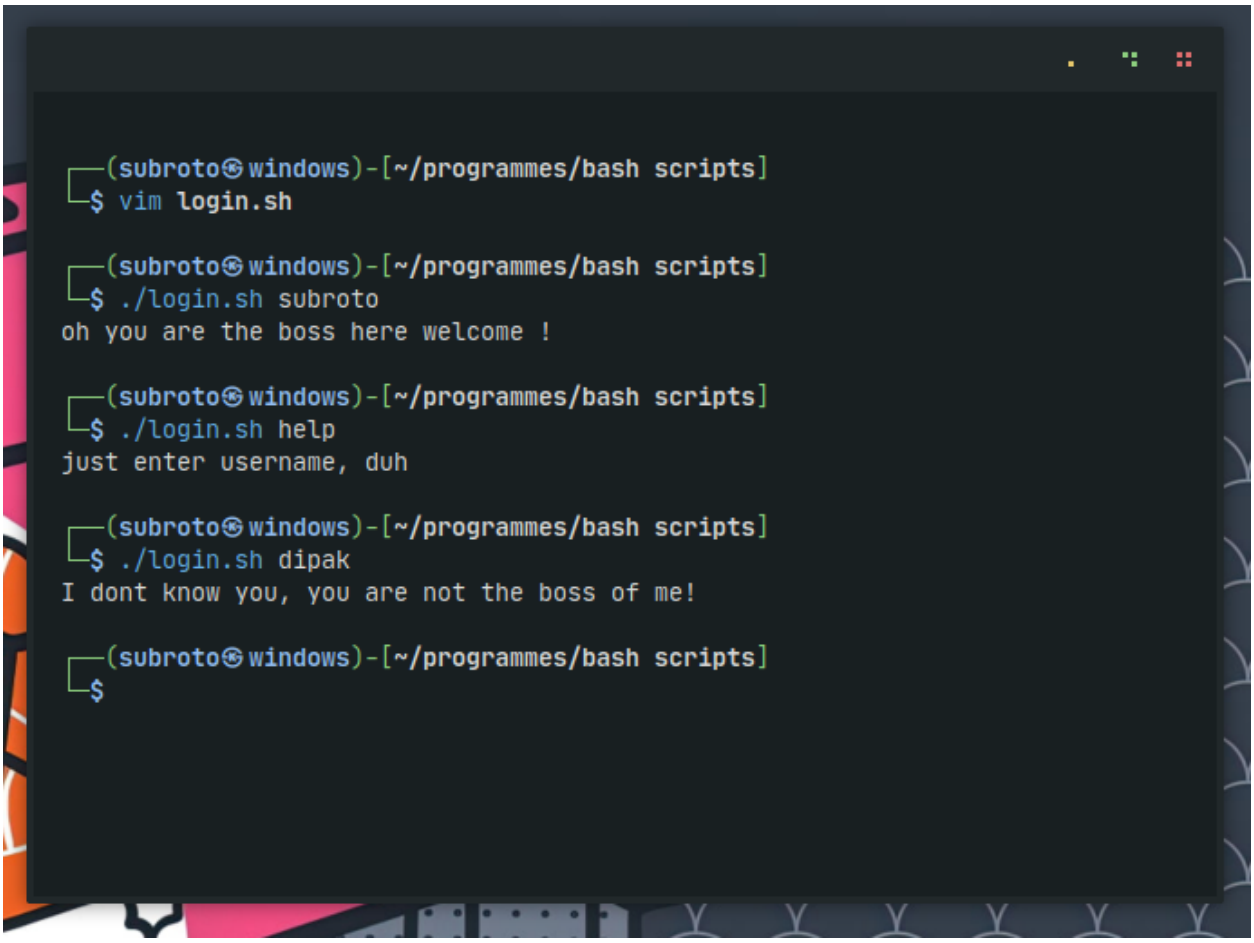



here it returned zero because both strings are equal

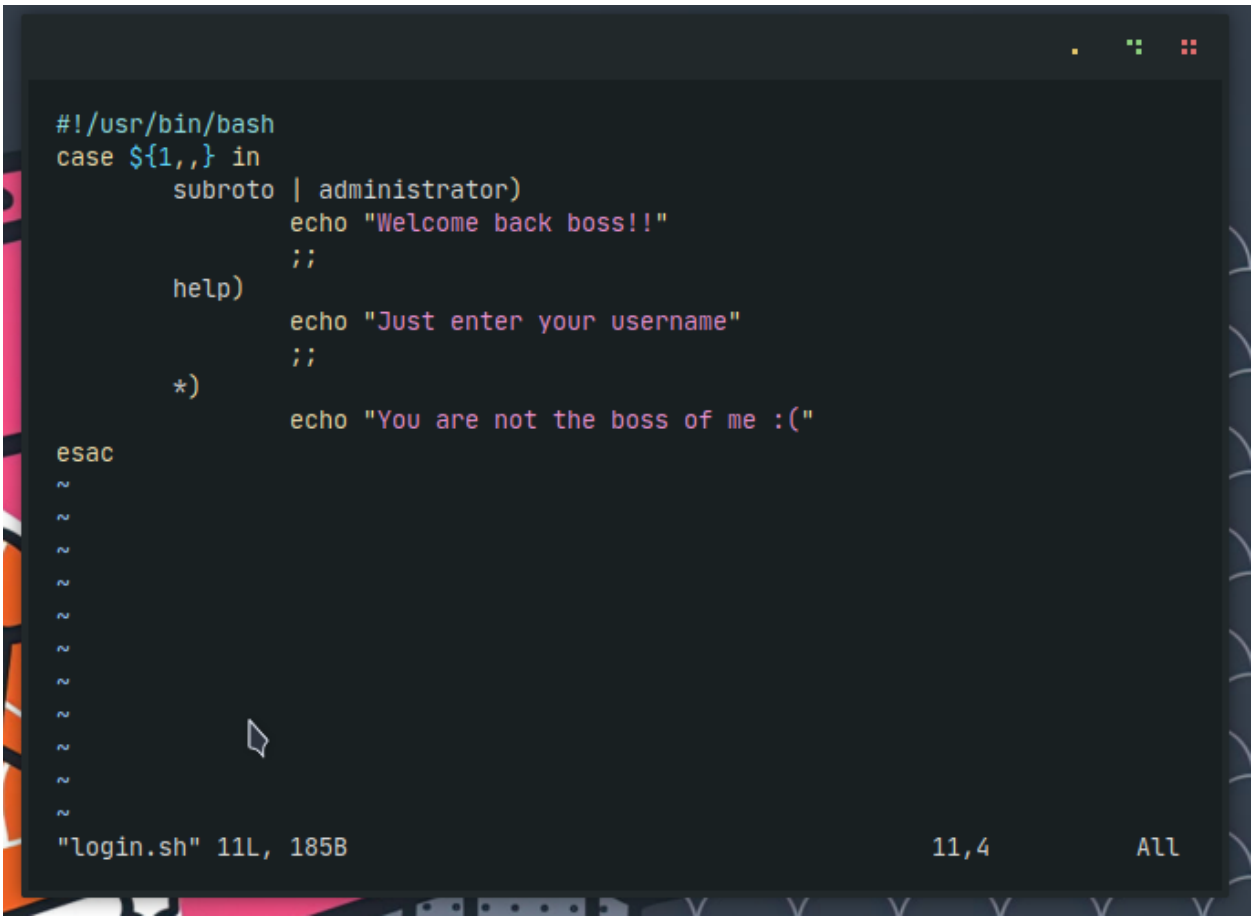

(subroto㉿windows)-[~/programmes/bash scripts]
$ [ hello = hello ]
(subroto㉿windows)-[~/programmes/bash scripts]
$ echo $?
0
(subroto㉿windows)-[~/programmes/bash scripts]
$ [ 1 = 0 ]
(subroto㉿windows)-[~/programmes/bash scripts]
$ echo $?
1
(subroto㉿windows)-[~/programmes/bash scripts]
$ [ 1 -eq 1 ]
(subroto㉿windows)-[~/programmes/bash scripts]
$ echo $?
0
(subroto㉿windows)-[~/programmes/bash scripts]
$

# If/Else/Elif


#!/usr/bin/bash
if [ ${1,,} = subroto ]; then
echo "oh you are the boss here welcome !"
elif [ ${1,,} = help ]; then
echo "just enter username, duh"
else
echo "I dont know you, you are not the boss of me!"
fi
"login.sh" 8L, 212B
8,2
All




## Case Statements

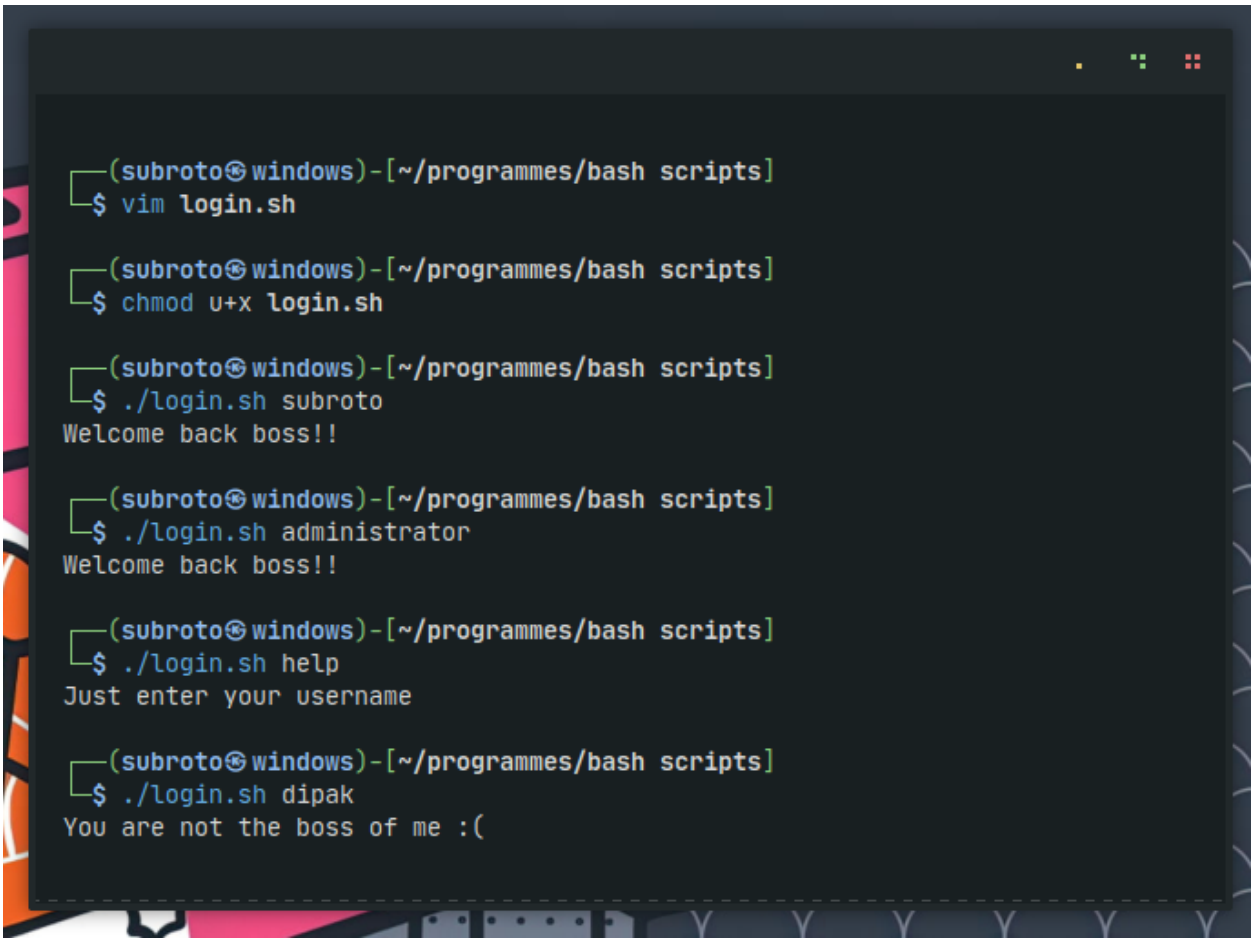



here first case is for either subroto or administrator

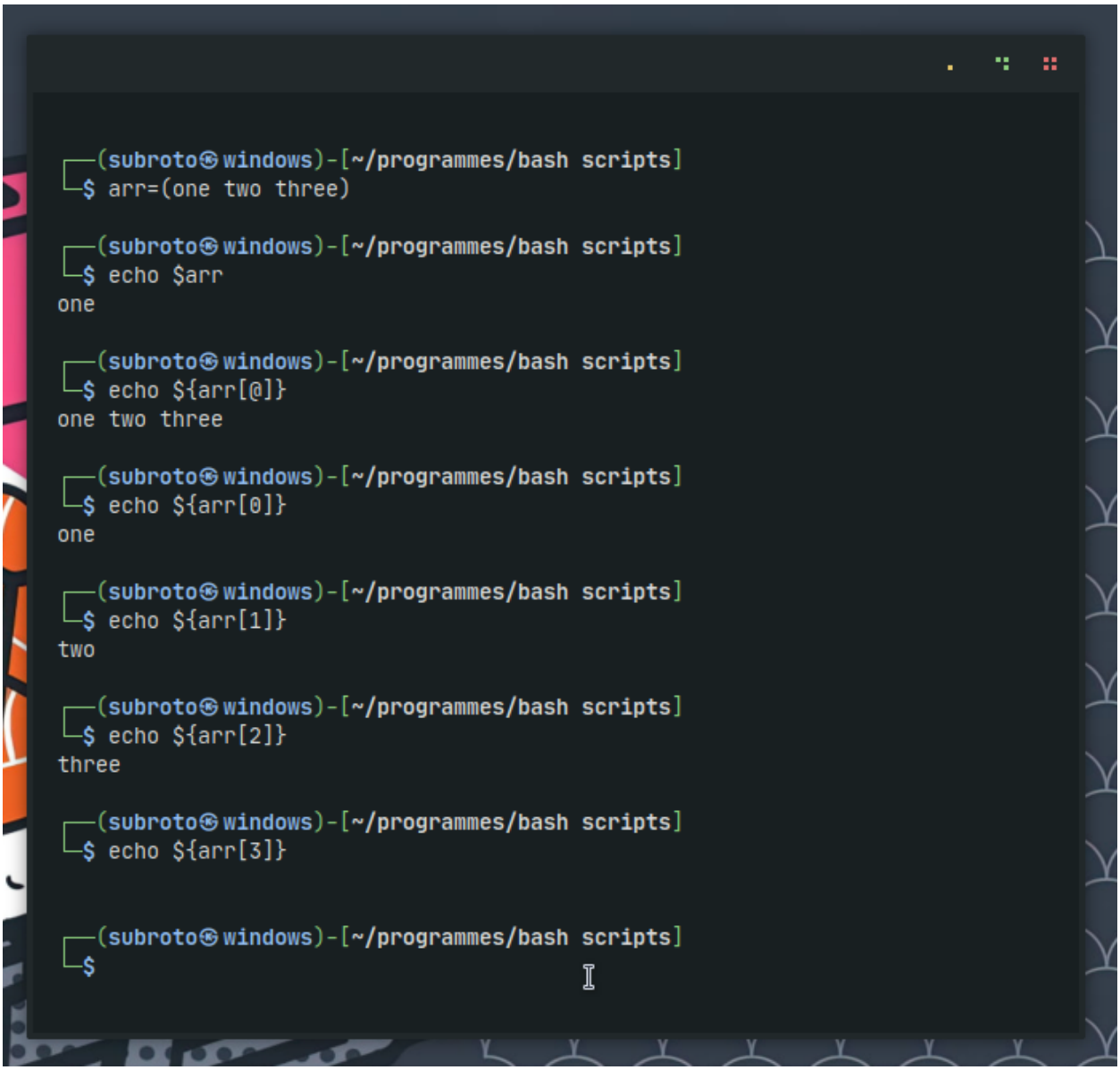
second case for help

and * is for default/any other word

esac is to end the cases.


(subroto㉿windows)-[~/programmes/bash scripts]
$ vim login.sh
(subroto㉿windows)-[~/programmes/bash scripts]
$ chmod u+x login.sh
(subroto㉿windows)-[~/programmes/bash scripts]
$ ./login.sh subroto
Welcome back boss!!
(subroto㉿windows)-[~/programmes/bash scripts]
$ ./login.sh administrator
Welcome back boss!!
(subroto㉿windows)-[~/programmes/bash scripts]
$ ./login.sh help
Just enter your username
(subroto㉿windows)-[~/programmes/bash scripts]
$ ./login.sh dipak
You are not the boss of me :(

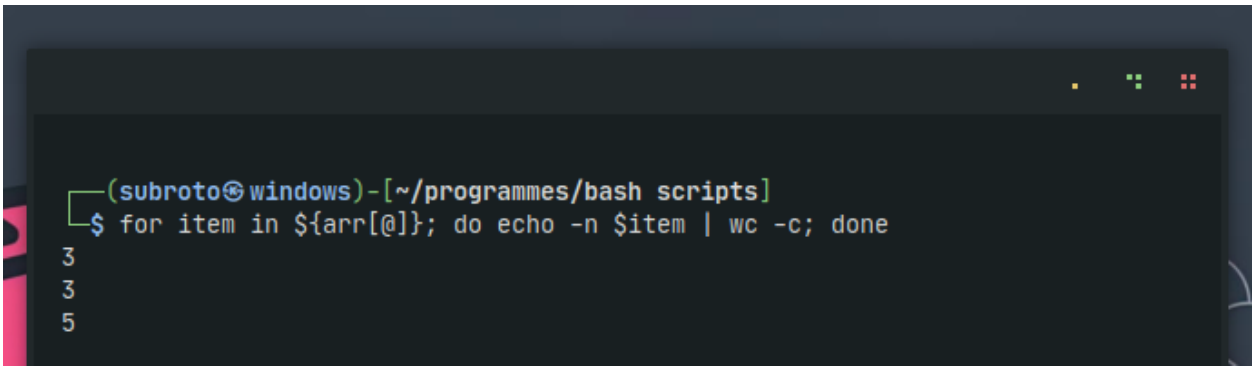
## Arrays




first line shows how to create array

then next shows how to print it and its elements.

## For LOOP

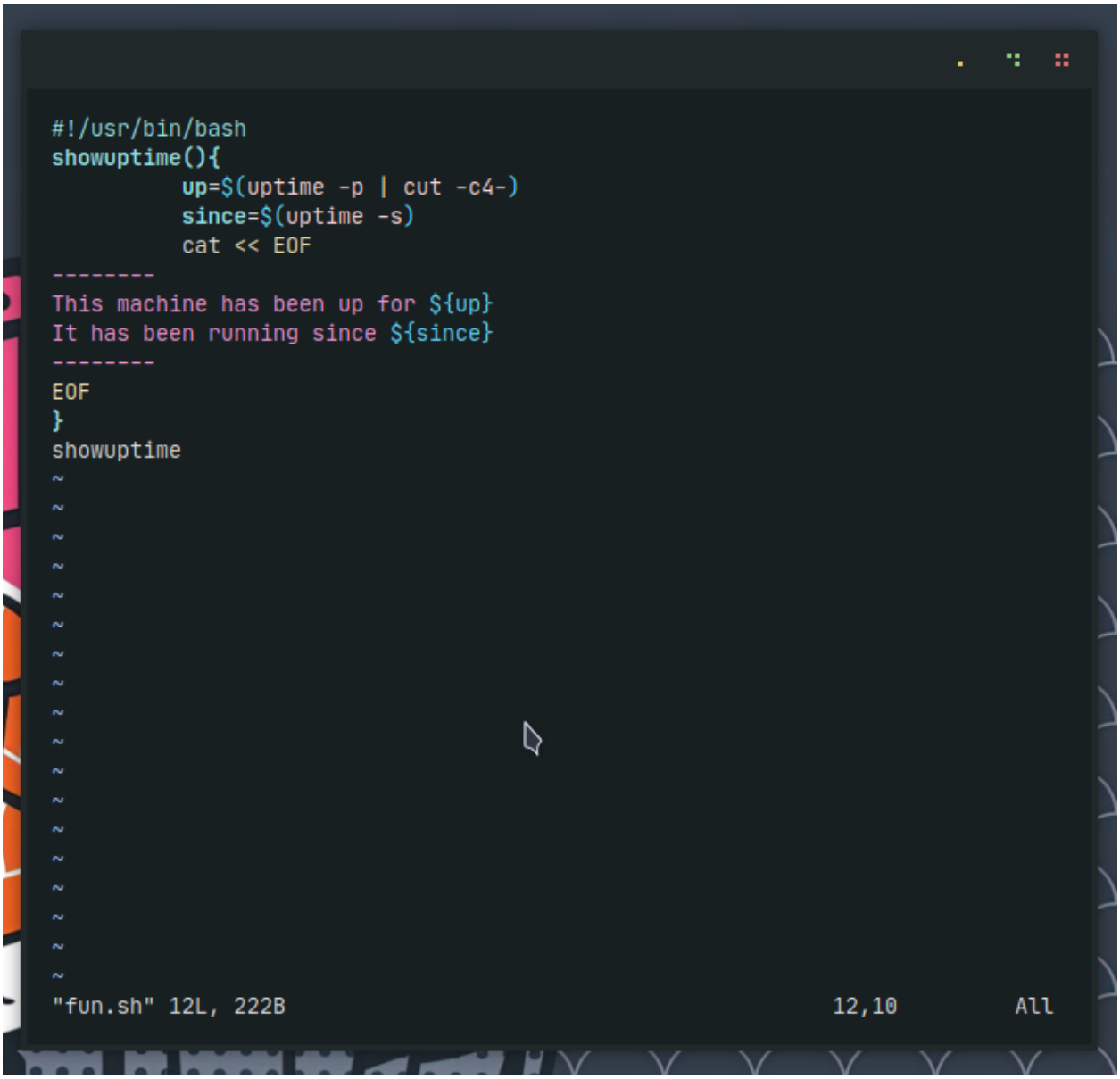


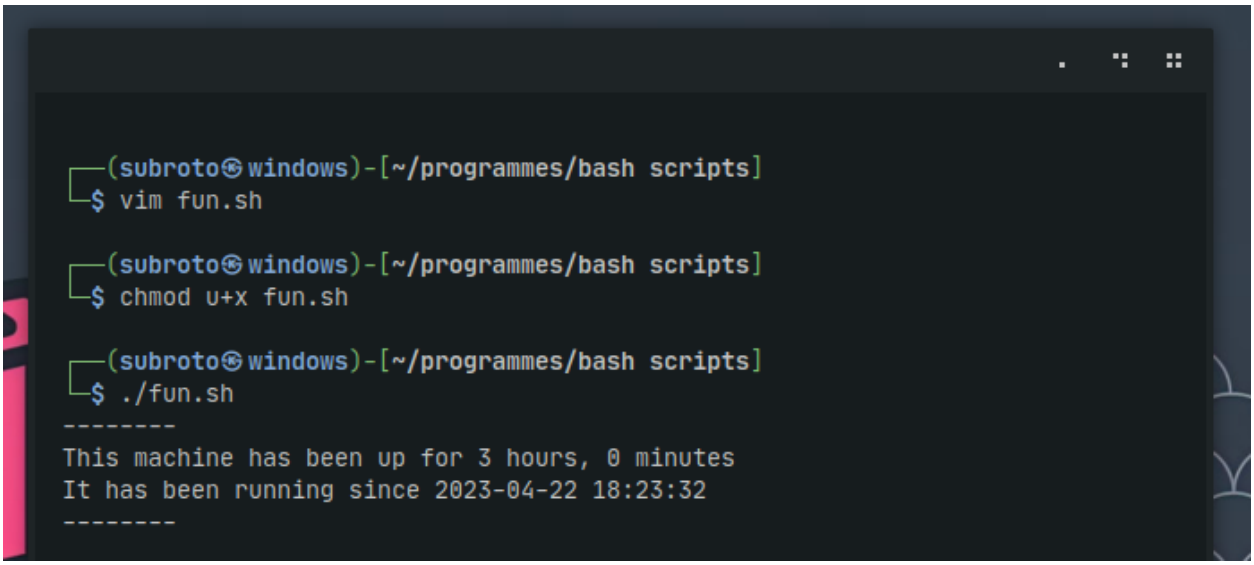

here , do is for instructing what to do

-n means ignore the newline characters

wc -c is to return word counts

## FUNCTIONS

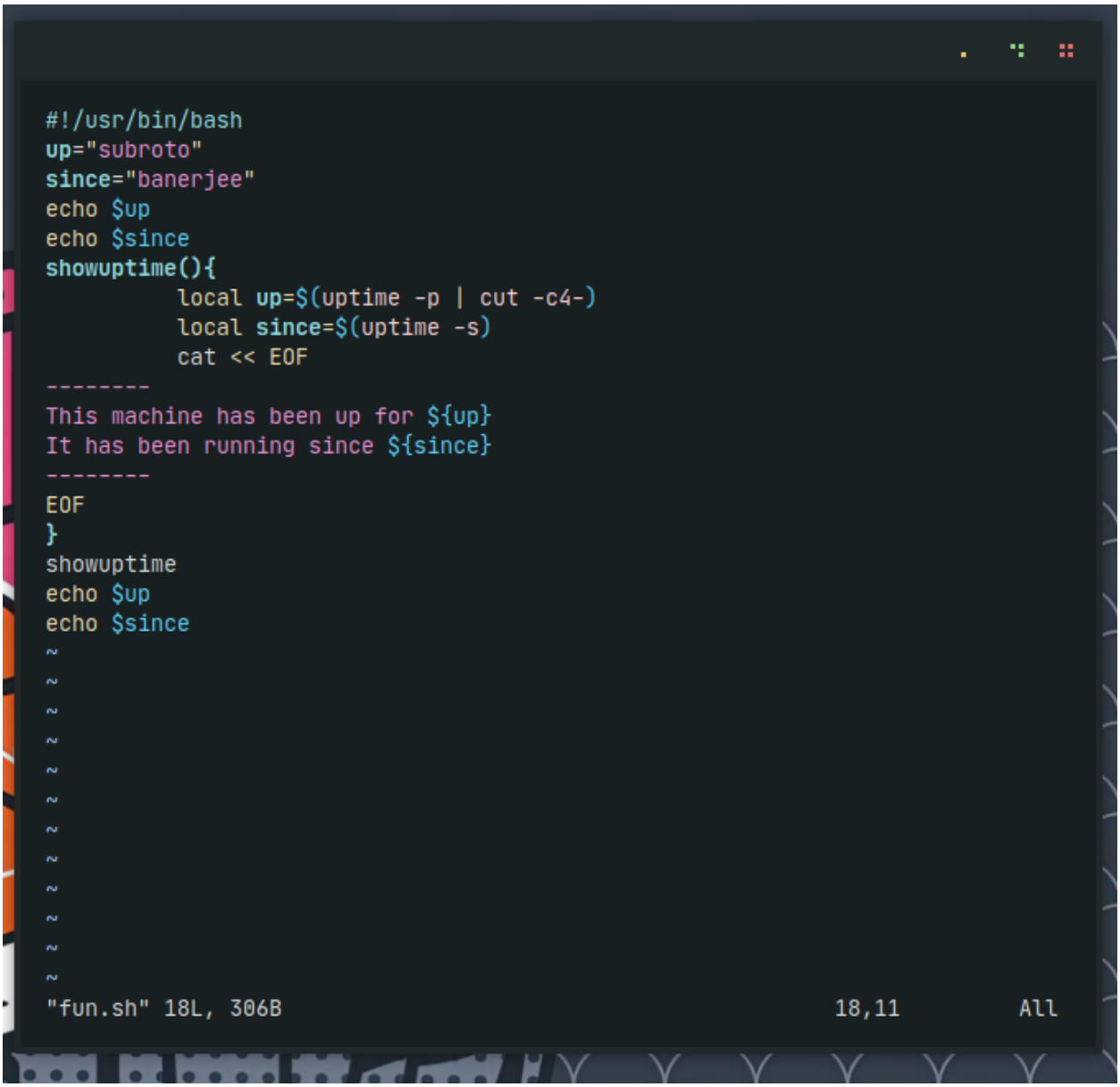

#!/usr/bin/bash
showuptime(){
up=$(uptime -p | cut -c4-)
since=$(uptime -s)
cat << EOF
--------
This machine has been up for ${up}
It has been running since ${since}
--------
EOF
}
showuptime
"fun.sh" 12L, 222B
12,10
All



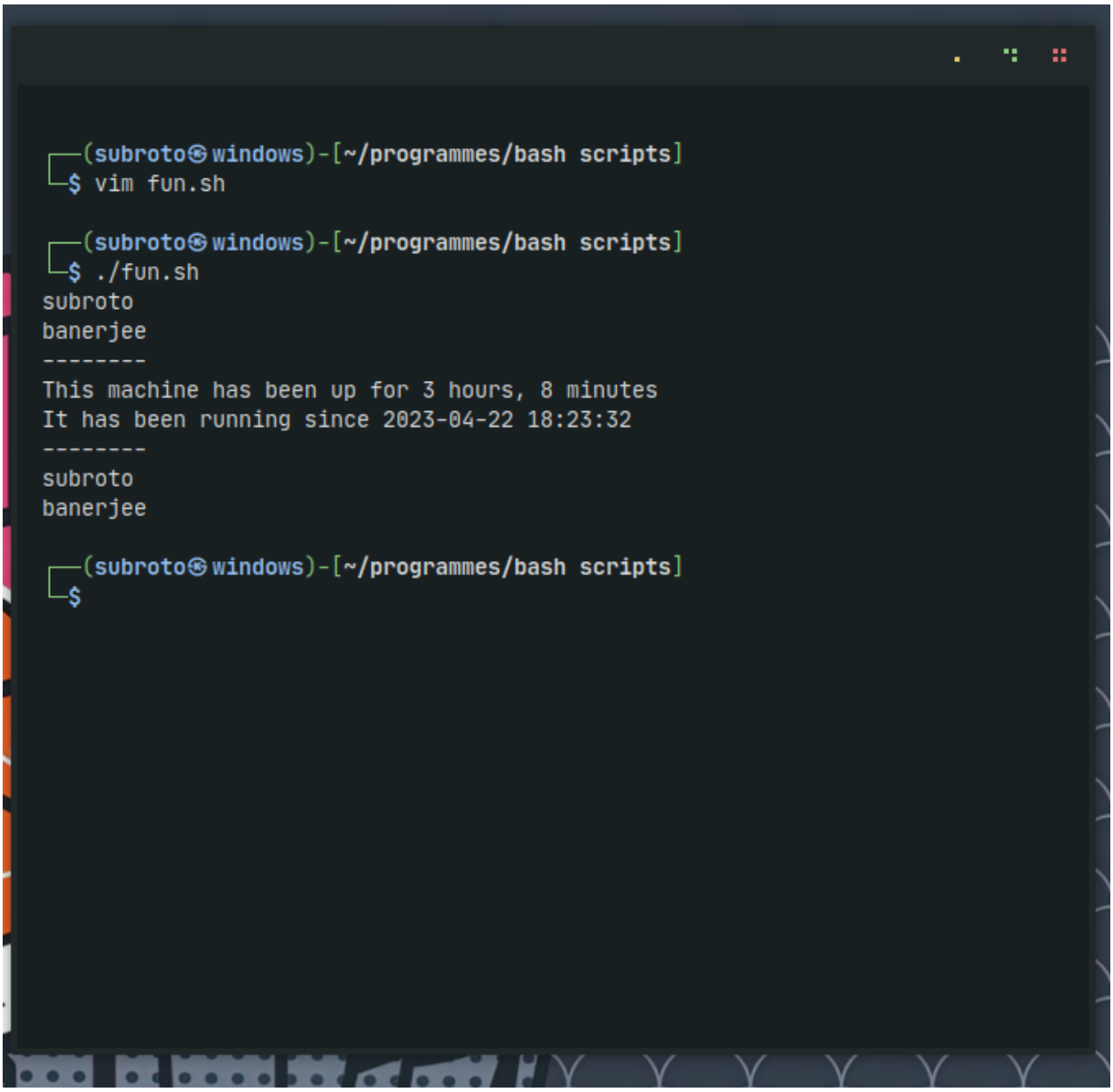

Creating local variables:

if we do not specify the variables to be local in the function, then they would be overwritten by the function

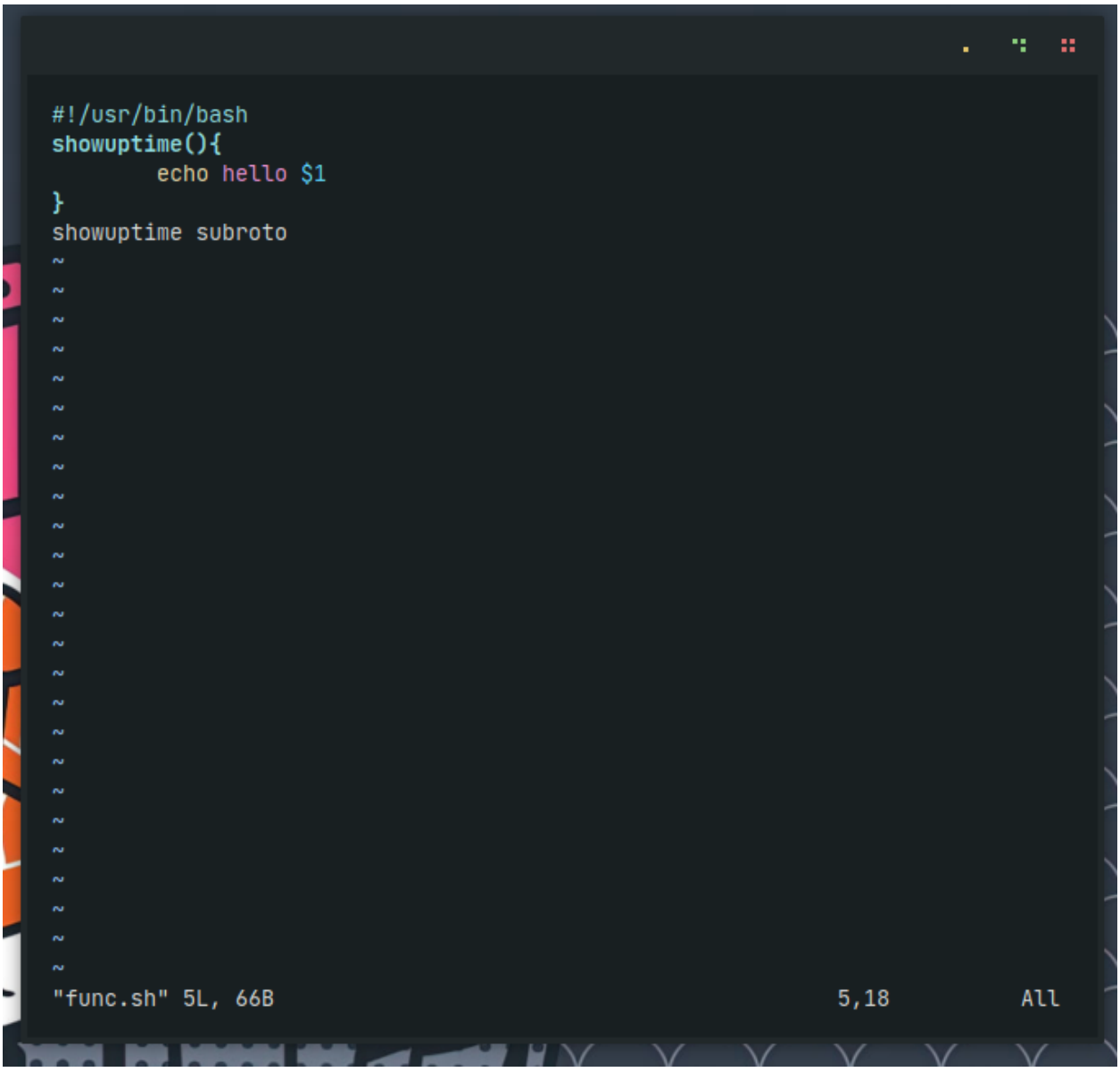
so use local to make them local to the function block only

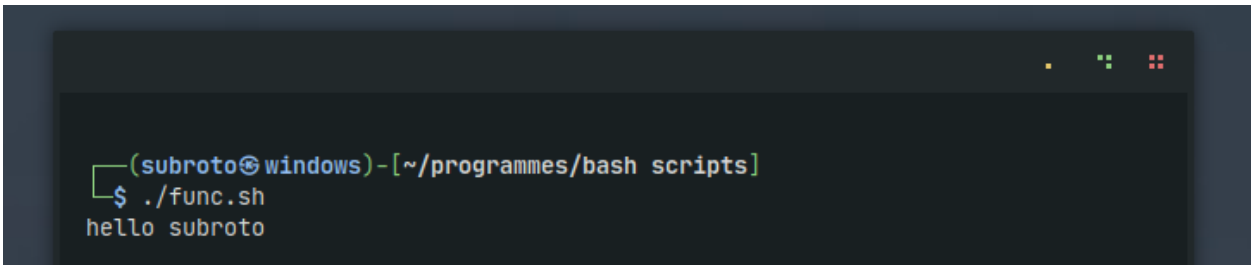


```
#!/usr/bin/bash
up="subroto"
since="banerjee"
echo $up
echo $since
showuptime(){
          local up=$(uptime -p | cut -c4-)
          local since=$(uptime -s)
          cat << EOF
--------
This machine has been up for ${up}
It has been running since ${since}
--------
EOF
}
showuptime
echo $up
echo $since
~
~
~
~
~
~
~
~
~
~
~
~
"fun.sh" 18L, 306B                                    18,11         All
```



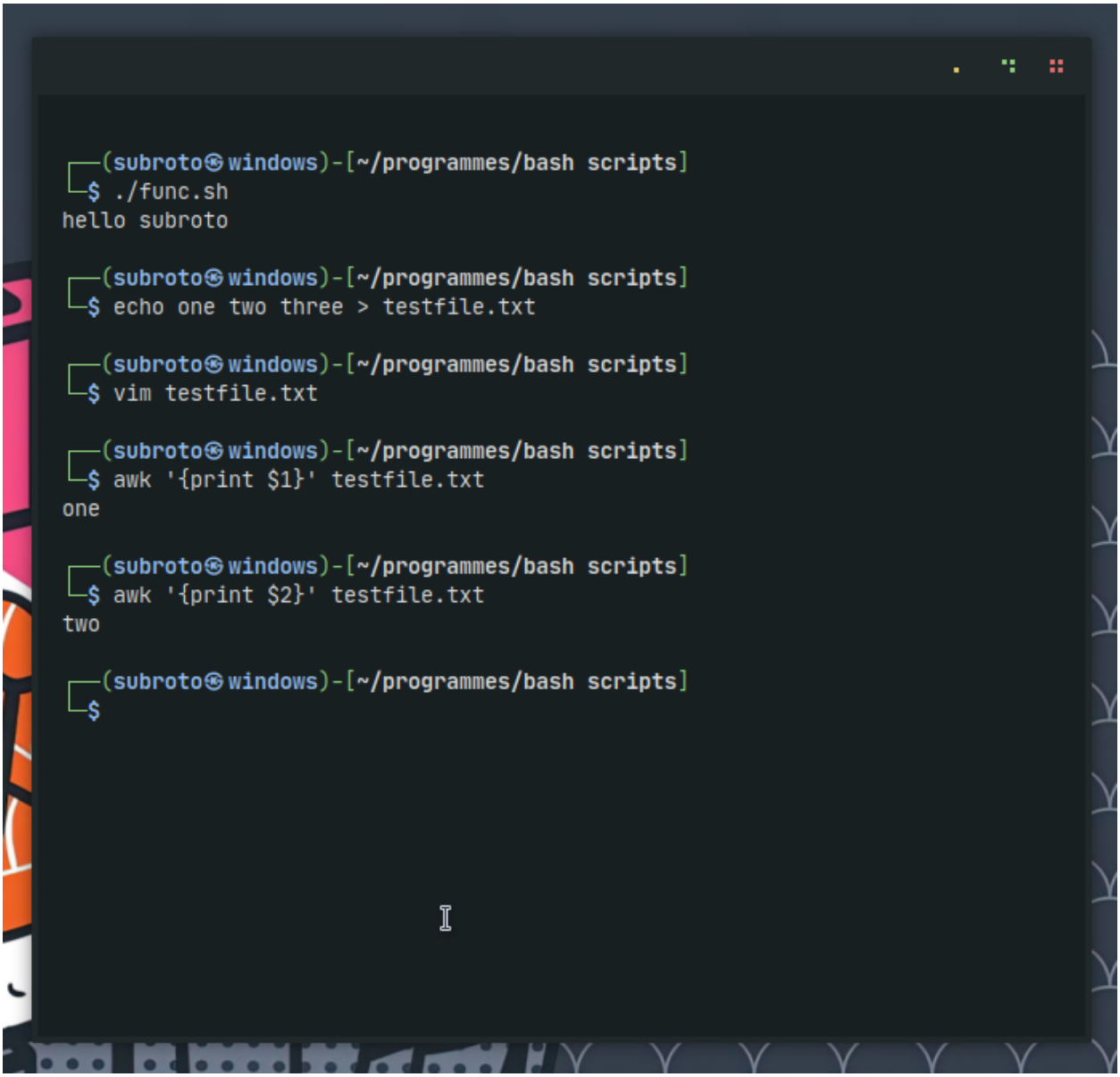

## Arguements in function


#!/usr/bin/bash
showuptime(){
echo hello $1
}
showuptime subroto
"func.sh" 5L, 66B
5,18
All


(subroto㉿windows)-[~/programmes/bash scripts]
$ ./func.sh
hello subroto

## AWK

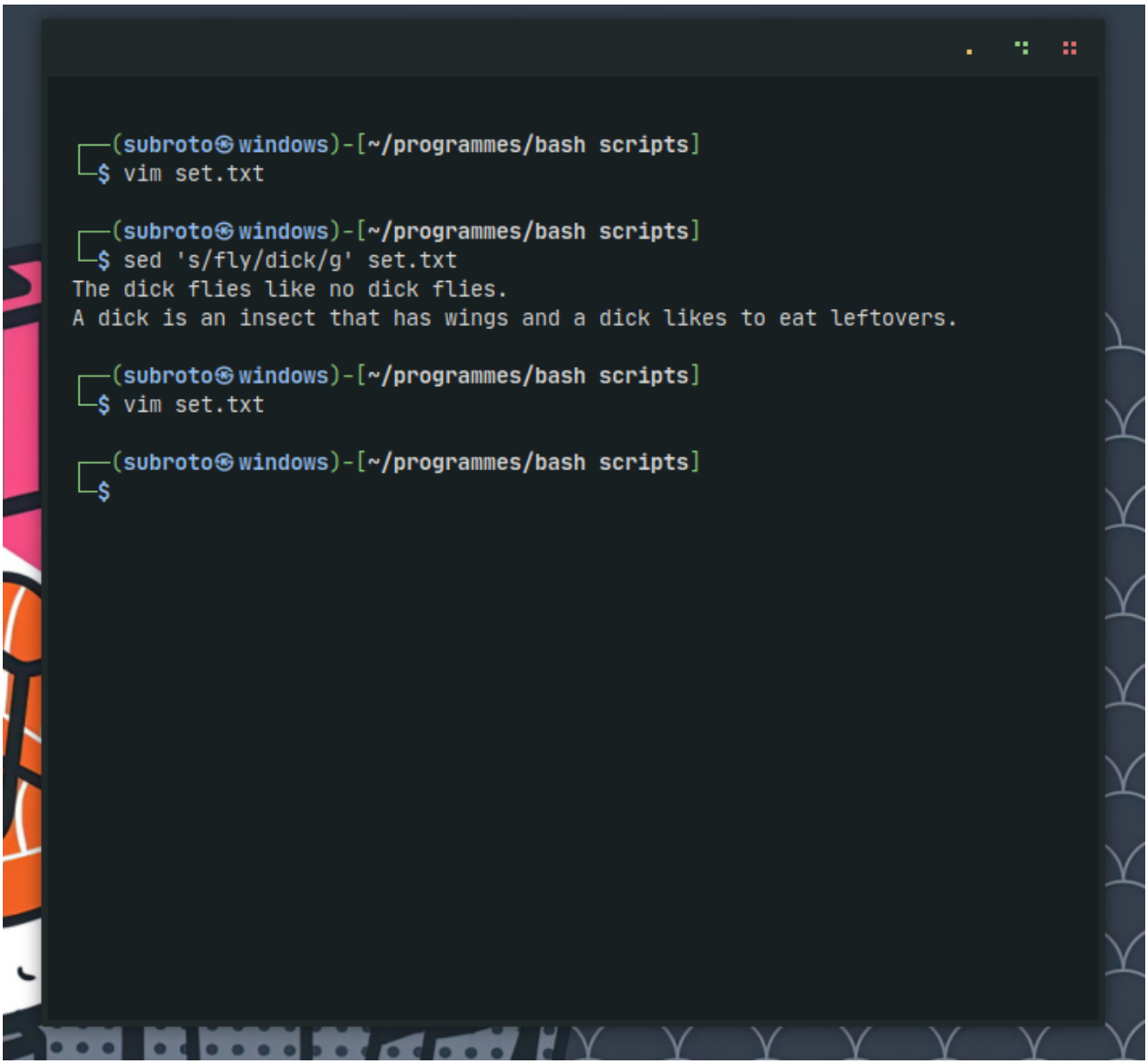
can be used to filter output of a file or a command output.




## SED

it is a command line tool that allows you to modify textfiles using simple commands


The fly flies like no fly flies.
A fly is an insect that has wings and a fly likes to eat leftovers.
"set.txt" 2L, 101B
2,67
All




s means substitute

then we provide the word for which we wanna substitute, then the word which is to be substituted then g means globally , we want this to change globally in the whole file.