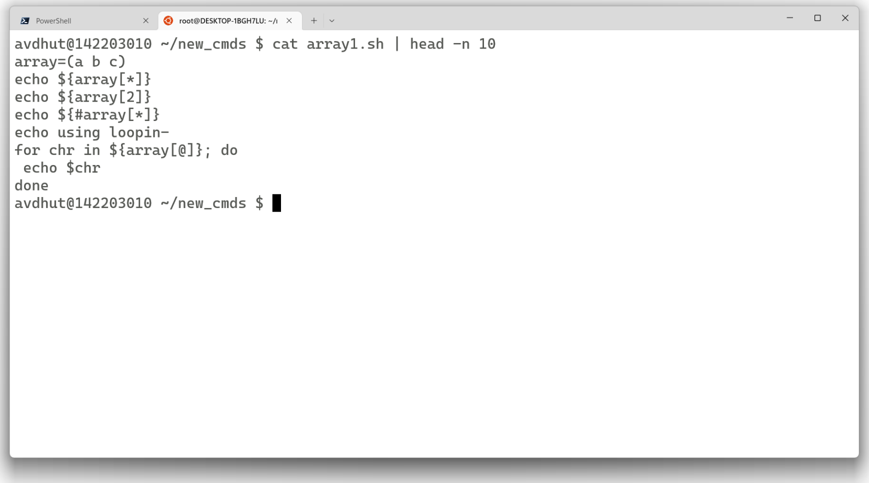
DTL Assignment - Shell Linux Commands

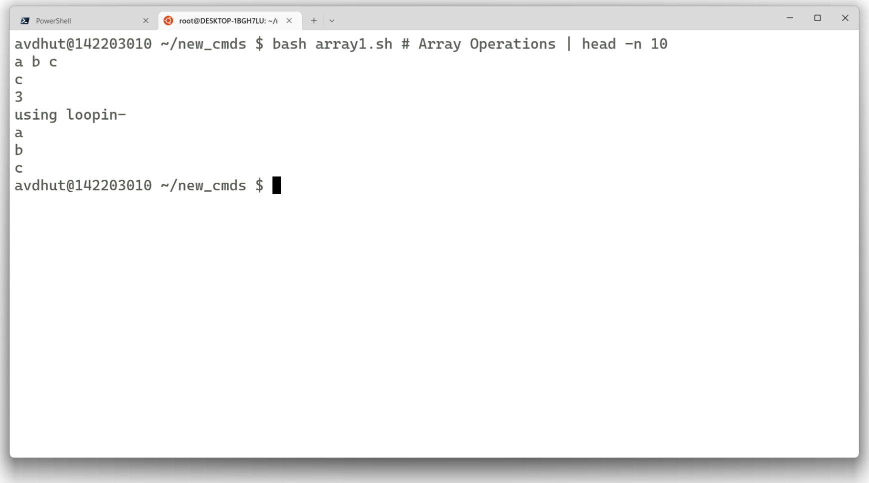
Avdhut Kamble
142203010

January 18, 2023



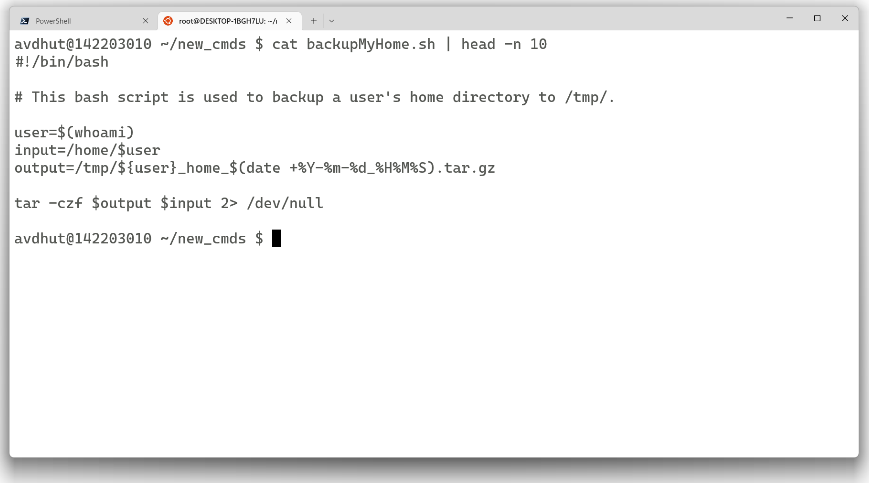
```
avdhut@142203010 ~/new_cmds $ cat array1.sh | head -n 10
array=(a b c)
echo ${array[*]}
echo ${array[2]}
echo ${#array[*]}
echo using loopin-
for chr in ${array[@]}; do
 echo $chr
done
avdhut@142203010 ~/new_cmds $
```

Figure 1: ”cat array1.sh” command



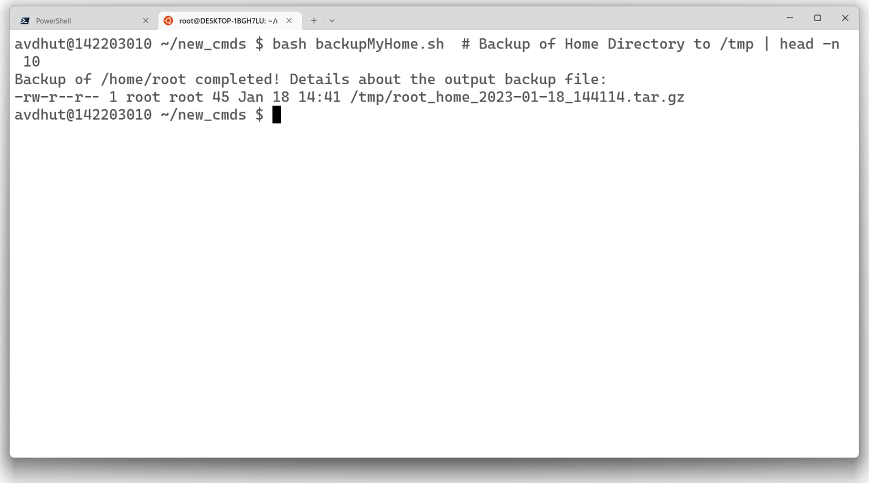
```
avdhut@142203010 ~/new_cmds $ bash array1.sh # Array Operations | head -n 10
a b c
c
3
using loopin-
a
b
c
avdhut@142203010 ~/new_cmds $
```

Figure 2: ”bash array1.sh Array Operations” command



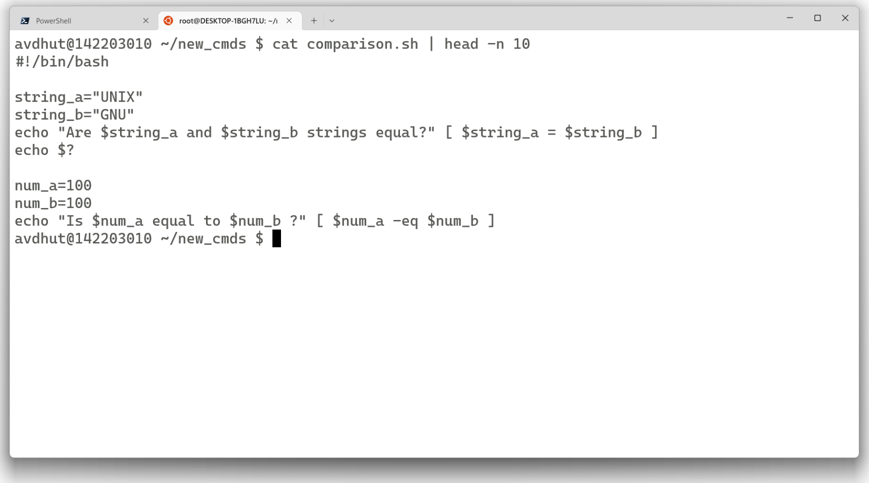
```
avdhut@142203010 ~/new_cmds $ cat backupMyHome.sh | head -n 10
#!/bin/bash

# This bash script is used to backup a user's home directory to /tmp/.

user=$(whoami)
input=/home/$user
output=/tmp/${user}_home_$(date +%Y-%m-%d_%H%M%S).tar.gz

tar -czf $output $input 2> /dev/null

avdhut@142203010 ~/new_cmds $
```

Figure 3: "cat backupMyHome.sh" command



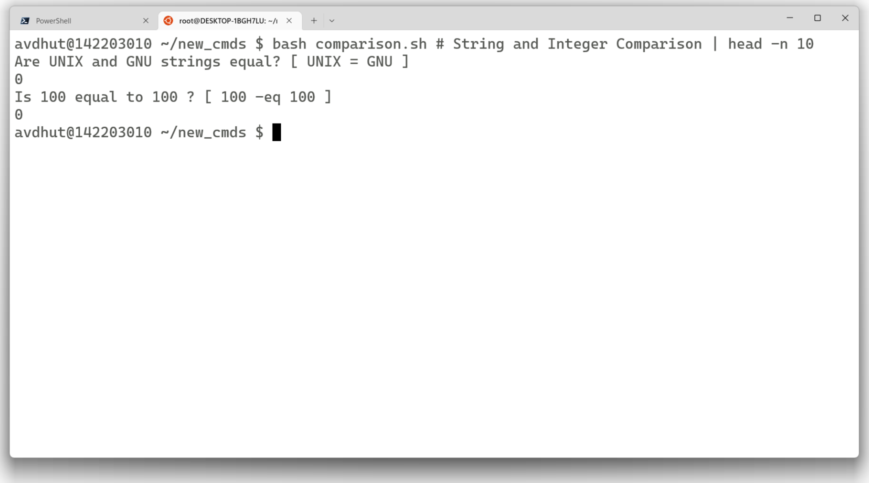
```
avdhut@142203010 ~/new_cmds $ bash backupMyHome.sh  # Backup of Home Directory to /tmp | head -n
 10
Backup of /home/root completed! Details about the output backup file:
-rw-r--r-- 1 root root 45 Jan 18 14:41 /tmp/root_home_2023-01-18_144114.tar.gz
avdhut@142203010 ~/new_cmds $
```

Figure 4: "bash backupMyHome.sh Backup of Home Directory to tmp" command



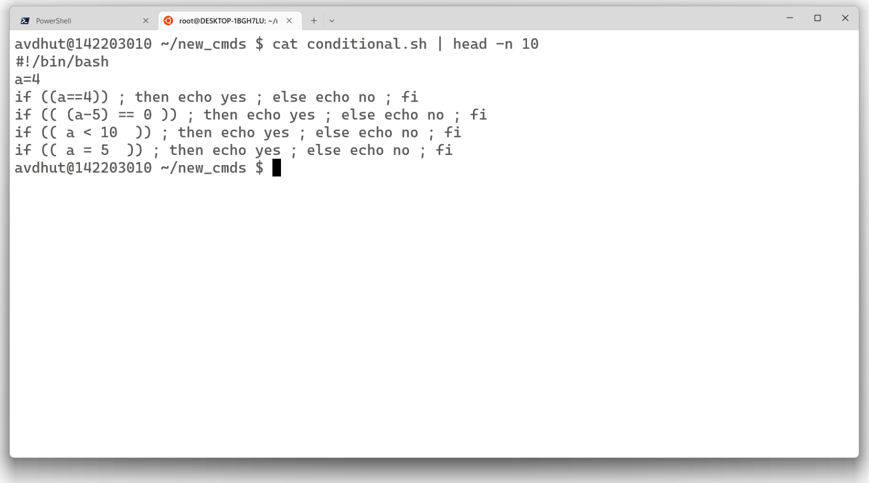
```
avdhut@142203010 ~/new_cmds $ cat comparison.sh | head -n 10
#!/bin/bash

string_a="UNIX"
string_b="GNU"
echo "Are $string_a and $string_b strings equal?" [ $string_a = $string_b ]
echo $?

num_a=100
num_b=100
echo "Is $num_a equal to $num_b ?" [ $num_a -eq $num_b ]
avdhut@142203010 ~/new_cmds $
```

Figure 5: "cat comparison.sh" command



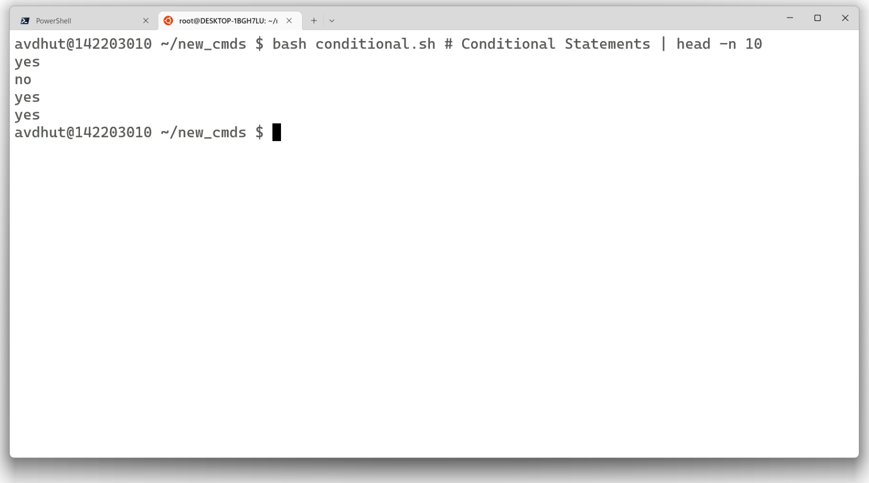
```
avdhut@142203010 ~/new_cmds $ bash comparison.sh # String and Integer Comparison | head -n 10
Are UNIX and GNU strings equal? [ UNIX = GNU ]
0
Is 100 equal to 100 ? [ 100 -eq 100 ]
0
avdhut@142203010 ~/new_cmds $
```

Figure 6: "bash comparison.sh" Comparison command



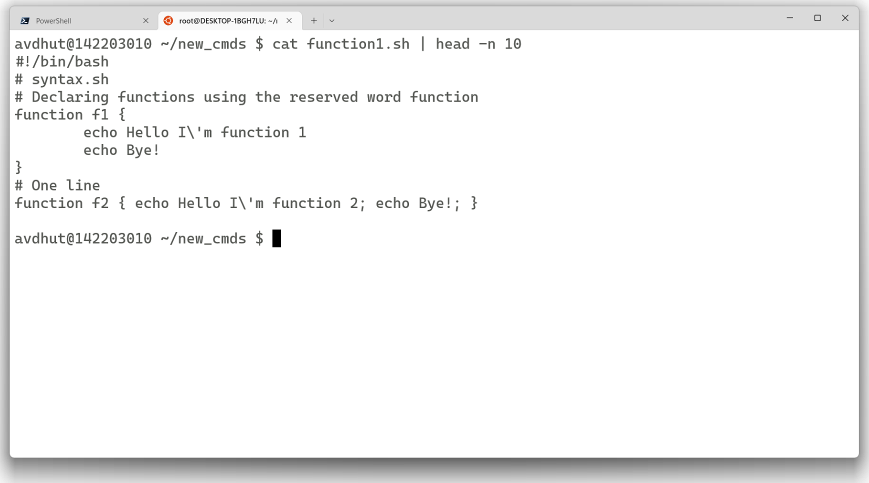
```
avdhut@142203010 ~/new_cmds $ cat conditional.sh | head -n 10
#!/bin/bash
a=4
if ((a==4)) ; then echo yes ; else echo no ; fi
if (( (a-5) == 0 )) ; then echo yes ; else echo no ; fi
if (( a < 10  )) ; then echo yes ; else echo no ; fi
if (( a = 5  )) ; then echo yes ; else echo no ; fi
avdhut@142203010 ~/new_cmds $ 
```

Figure 7: ”cat conditional.sh” command



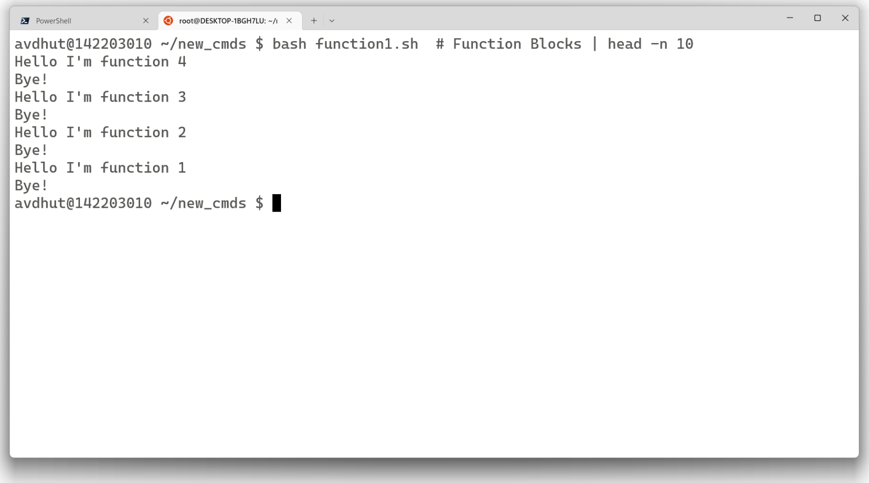
```
avdhut@142203010 ~/new_cmds $ bash conditional.sh # Conditional Statements | head -n 10
yes
no
yes
yes
avdhut@142203010 ~/new_cmds $ 
```

Figure 8: ”bash conditional.sh Conditional Statements” command



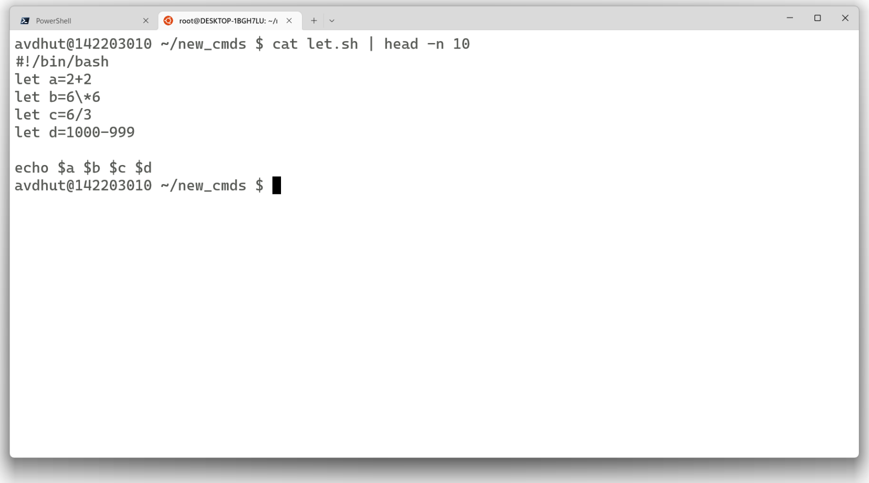
```
avdhut@142203010 ~/new_cmds $ cat function1.sh | head -n 10
#!/bin/bash
# syntax.sh
# Declaring functions using the reserved word function
function f1 {
        echo Hello I\'m function 1
        echo Bye!
}
# One line
function f2 { echo Hello I\'m function 2; echo Bye!; }

avdhut@142203010 ~/new_cmds $
```

Figure 9: ”cat function1.sh” command



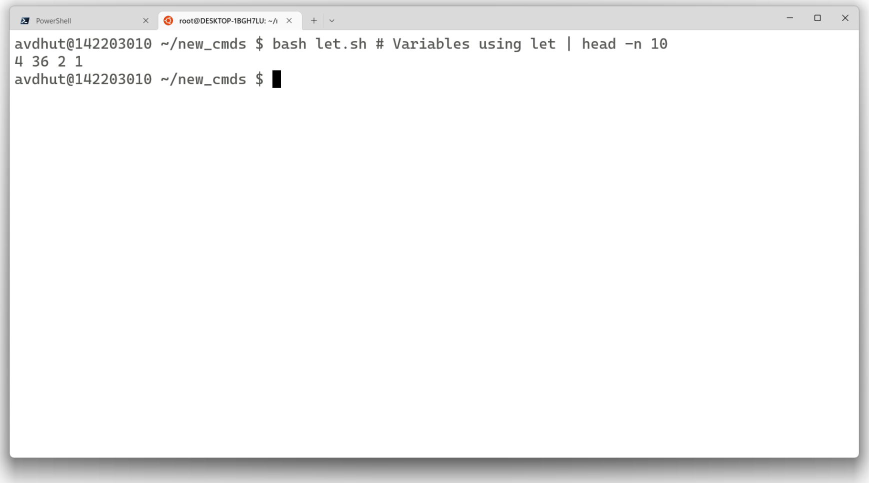
```
avdhut@142203010 ~/new_cmds $ bash function1.sh  # Function Blocks | head -n 10
Hello I'm function 4
Bye!
Hello I'm function 3
Bye!
Hello I'm function 2
Bye!
Hello I'm function 1
Bye!
avdhut@142203010 ~/new_cmds $
```

Figure 10: ”bash function1.sh Function Blocks” command



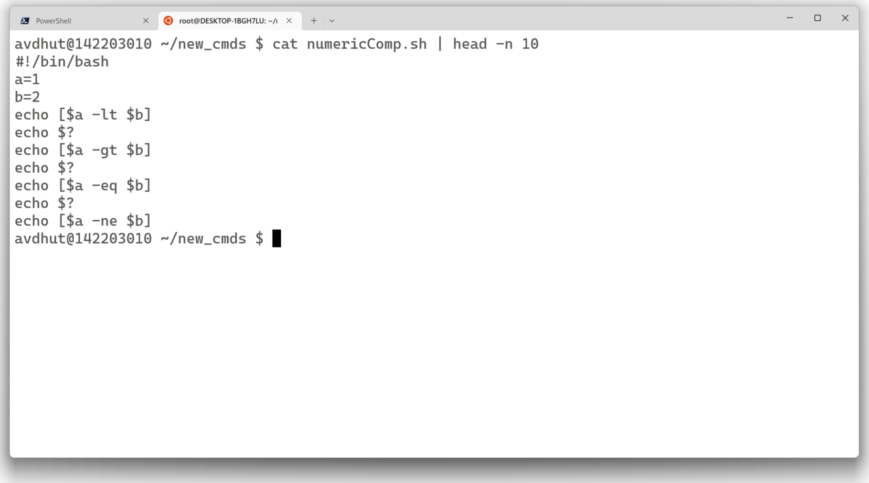
```
avdhut@142203010 ~/new_cmds $ cat let.sh | head -n 10
#!/bin/bash
let a=2+2
let b=6\*6
let c=6/3
let d=1000-999

echo $a $b $c $d
avdhut@142203010 ~/new_cmds $ 
```

Figure 11: "cat let.sh" command



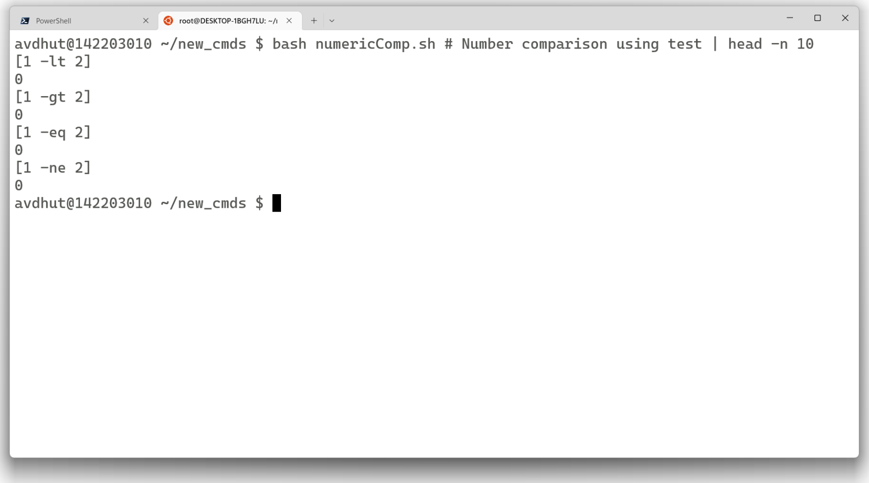
```
avdhut@142203010 ~/new_cmds $ bash let.sh # Variables using let | head -n 10
4 36 2 1
avdhut@142203010 ~/new_cmds $ 
```

Figure 12: "bash let.sh Variables using let" command



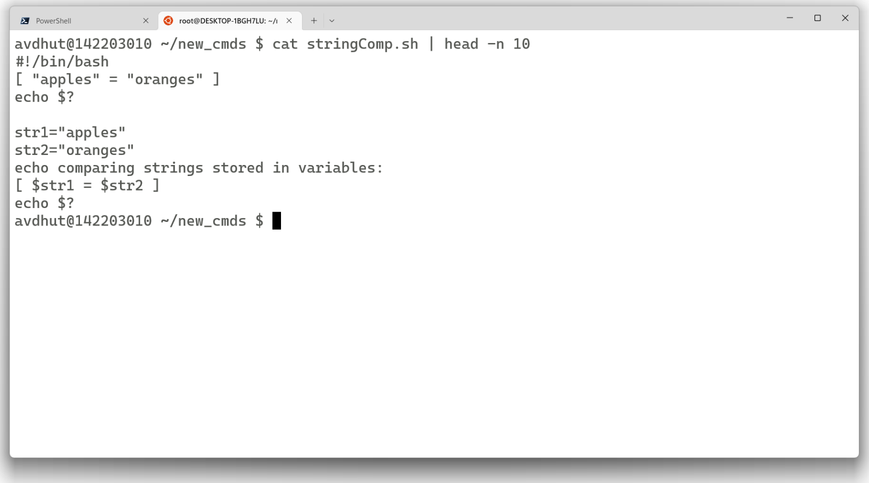
```
avdhut@142203010 ~/new_cmds $ cat numericComp.sh | head -n 10
#!/bin/bash
a=1
b=2
echo [$a -lt $b]
echo $?
echo [$a -gt $b]
echo $?
echo [$a -eq $b]
echo $?
echo [$a -ne $b]
avdhut@142203010 ~/new_cmds $
```

Figure 13: ”cat numericComp.sh” command



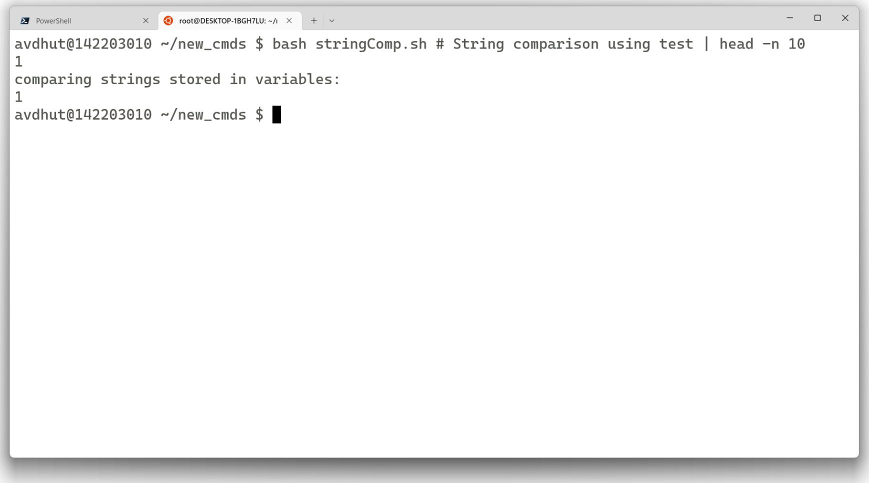
```
avdhut@142203010 ~/new_cmds $ bash numericComp.sh # Number comparison using test | head -n 10
[1 -lt 2]
0
[1 -gt 2]
0
[1 -eq 2]
0
[1 -ne 2]
0
avdhut@142203010 ~/new_cmds $
```

Figure 14: ”bash numericComp.sh Number comparison using test” command



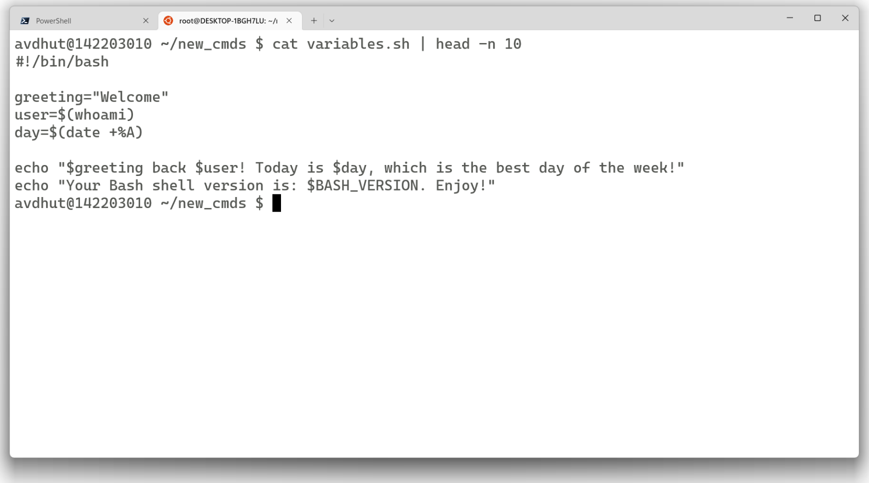
```
avdhut@142203010 ~/new_cmds $ cat stringComp.sh | head -n 10
#!/bin/bash
[ "apples" = "oranges" ]
echo $?

str1="apples"
str2="oranges"
echo comparing strings stored in variables:
[ $str1 = $str2 ]
echo $?
avdhut@142203010 ~/new_cmds $
```

Figure 15: ”cat stringComp.sh” command



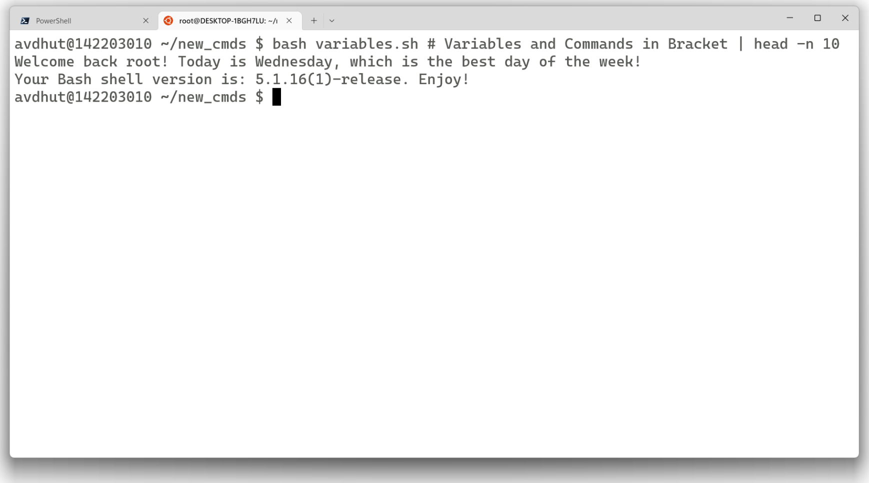
```
avdhut@142203010 ~/new_cmds $ bash stringComp.sh # String comparison using test | head -n 10
1
comparing strings stored in variables:
1
avdhut@142203010 ~/new_cmds $
```

Figure 16: ”bash stringComp.sh String comparison using test” command



```
avdhut@142203010 ~/new_cmds $ cat variables.sh | head -n 10
#!/bin/bash

greeting="Welcome"
user=$(whoami)
day=$(date +%A)

echo "$greeting back $user! Today is $day, which is the best day of the week!"
echo "Your Bash shell version is: $BASH_VERSION. Enjoy!"
avdhut@142203010 ~/new_cmds $
```

Figure 17: ”cat variables.sh” command



```
avdhut@142203010 ~/new_cmds $ bash variables.sh # Variables and Commands in Bracket | head -n 10
Welcome back root! Today is Wednesday, which is the best day of the week!
Your Bash shell version is: 5.1.16(1)-release. Enjoy!
avdhut@142203010 ~/new_cmds $
```

Figure 18: ”bash variables.sh Variables and Commands in Bracket” command