Thursday, March 31, 2022 10:37 AM

printf("t") ---------------- letter t is printed

usual meaning of t in "" is the letter t
we want to use another meaning of t (tab) --------------- escape t ------------ \t

usual meaning of n in "" is the letter n
we want to use another meaning of n (new line) --------------- escape n ------------ \n

what is the escape character ? = \ backslash

usual meaning of * has to be escaped *

____

Condition = if [condition that you put here]

Conditional constructs in bash !!!
Standard [ ]
Magic [[ ]]

$((4 * 5))

____

1	Shell command called as **expr**
Prompt	**expr all integer maths**
in the shell script	**use expr to do all integer maths**

We learnt how to use a command in a statement = put the command in back quotes
We learnt that expr and (( )) are only working for integer arithmetic

PIPE ----------- IPC (pipe system call) P1 | P2 output of P1 is sent as input to P2
We can use pipe in CLI also --- we use | symbol
We know that every command is a process
We can chain the commands using PIPE C1 | C2 | C3 }}} final output of C3 will be seen

Text manipulation commands ---- wc , grep , head, tail , sort , grep , cut , tr

wc	**Word count of the input**
head	**It shows the topmost lines of the input**
	yellow@prachi:~/shellprograms$ head -3 data 12 smriti 89 19 gauri 45 34 trsha 78
tail	It shows the bottommost lines
	$tail **-3** data 112 sonal 78 88 prithvi 88 $tail -1 data $tail -5 data 99 gil 45 101 nitin 99 112 sonal 78 88 prithvi 88
sort	By default sorts the input LEXICOGRAPHICALLY (alphabetically) **first column** ascending order $sort data **101** nitin 99 **112** sonal 78 **12** smriti 89 19 gauri 45 34 trsha 78 88 prithvi 88 99 gil 45 blue ball

	basically ball basically blue
Sort first col in numerical ascending	$sort **-n** data 12 smriti 89 19 gauri 45 34 trsha 78 88 prithvi 88 99 gil 45 101 nitin 99 112 sonal 78
Sort **First col** **Num** **desc**	$sort **-n -r** data 112 sonal 78 101 nitin 99 99 gil 45 88 prithvi 88 34 trsha 78 19 gauri 45 12 smriti 89
Sort **3rd col** **Alpha** **Asc**	$sort -k 3 data 19 gauri 45 99 gil 45 112 sonal 78 34 trsha 78 88 prithvi 88 12 smriti 89 101 nitin 99 $sort -k 2 data
Sort **3rd** **Num** **desc**	$sort -k 3 -r -n data 101 nitin 99 12 smriti 89 88 prithvi 88 34 trsha 78 112 sonal 78 99 gil 45 19 gauri 45
	$sort -k 3 -r -n data | head -1 **topper** 101 nitin 99 $sort -k 3 -r -n data | head -2 **first two** 101 nitin 99 12 smriti 89 $sort -k 3 -r -n data | head -2 | tail -1 **second topper** 12 smriti 89 $sort -k 3 -r -n data | head -3 | tail -1 **third topper** 88 prithvi 88 $
Grep	Searches the input for a search string , shows all lines that have the search string
	grep search-string data
$ indicates that select only the lines that have search string in the end	grep 99**$** data 101 nitin 99
^ indicates that select only the lines that have search string in the begining	grep ^99 data 99 gil 45
Show all lines that are BLANK **Nothing between ^$**	grep ^$ data
Count the number of blank lines in a file	grep ^$ data | wc -l 5
-v indicates inverted or reverse search--- show all lines that DON’T have the search string	$grep **-v** 99 data 12 smriti 89 19 gauri 45 34 trsha 78 112 sonal 78 88 prithvi 88
Show all lines that have chars between t to z small case	$grep **[t-z]** data 12 smriti 89 19 gauri 45 34 trsha 78 101 nitin 99 88 prithvi 88
	grep **[0-1a-cA-C]** data 12 smriti 89 19 gauri 45 34 trsha 78

	101 nitin 99 112 sonal 78 66 YASMINE 84
tr command	tr is used to REPLACE text in the input
Replace all lowercase chars with uppercase	tr "[a-z]" "[A-Z]" < data
Delete all occurences of letter a	tr **-d** "a" <data
cut command	cuts the char for which index is given from the line

Redirection ------------

Output redirection ---the output is sent to a file
> overwrite the file
>> append to file

<< input redirection

1	**Write a shell script to accept a string from the user and show its length**
2	**Write a shell script that accepts a string and shows a menu to the user** **1 show string in uppercase** **2 show string in lowercase** **3 show string as it is**
3	**Write a shell script that accepts a file name and search string** **Show how many lines in the file have the search string** **Show how many lines in the file don't have the search string** **Grep , grep -v , wc -l**
4	**Accept names from the user and show them in sorted order** **Ask user 1 asc , 2 desc**
5	**Accept a number from user and show as follows**
	User enters 3
For loop traditional	***** *** *** *** * ***
While loop	**3** **3 3** **3 3 3**
Until loop	**1** **1 2** **1 2 3**
	Do the example loop2.sh for shell script for loop

```
  for   var  in  LIST
  do
        Use the variable
  done
```