**SHELL SCRIPTING**

read VAR < /etc/hostname

VAR=$(pwd)

Let, (()), [], expr, bc(only for floating math)

**MATH:**

Script execution on remote server: ssh $host\_name "/home/gfadmin/operations/kill\_pid.sh $port\_num"

let num=result+7(don’t use $ when using let)

INC – let num++

DEC- let num—

Other way:

Num=$(( result + 6 ))

Num=$[ result + 6 ]

Res=`expr $num + 5`

Res=$(expr $num + 5`)

Res=`echo “$num \* 1.9” | bc

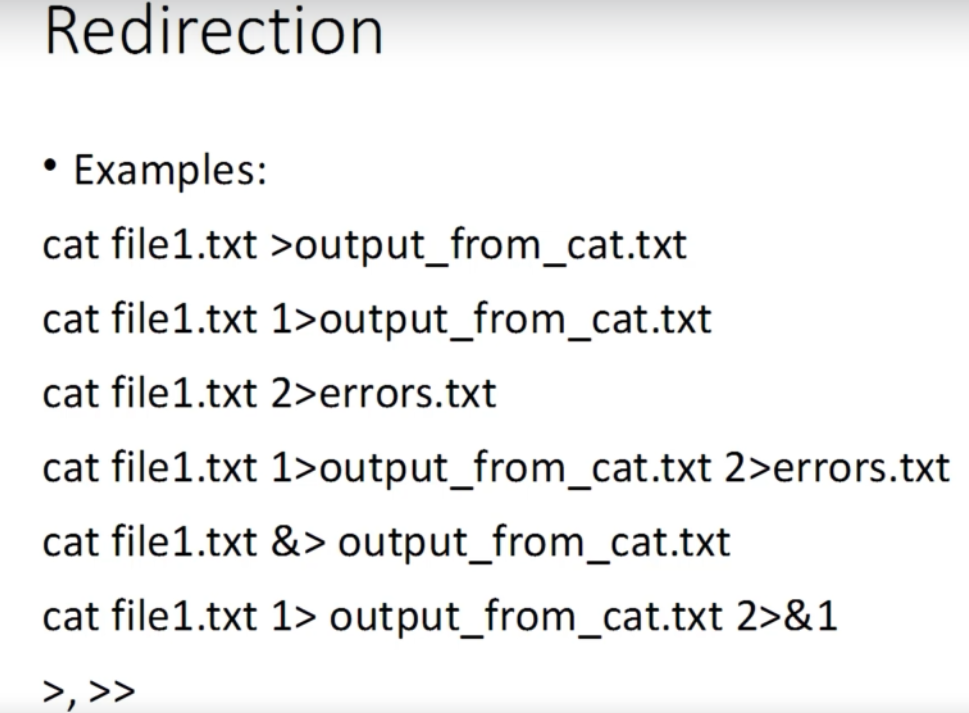
**ARGUMENTS:**

To view filed separator: set | grep ^IFS

**Redirection and Piping:**

STDIN(0) – input

STDOUT(1)-output

STDERR(2)-errorr

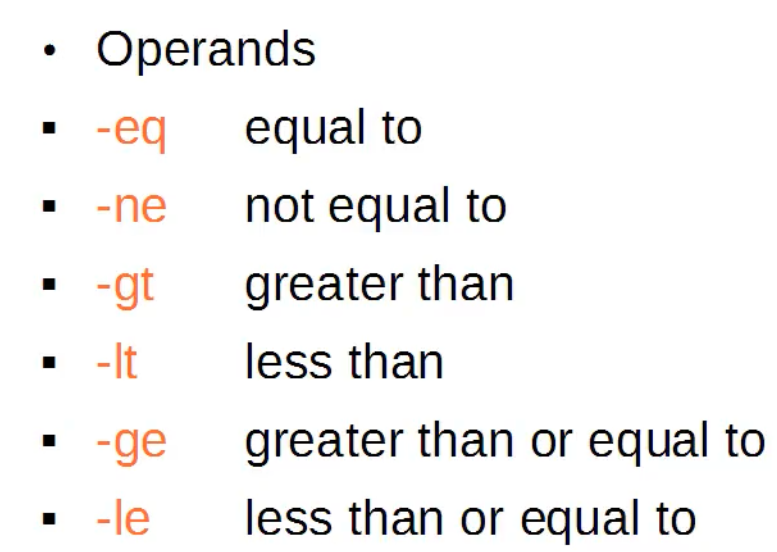
**EXIT Status:**

Echo $?

We can use exit <num> in our scripts

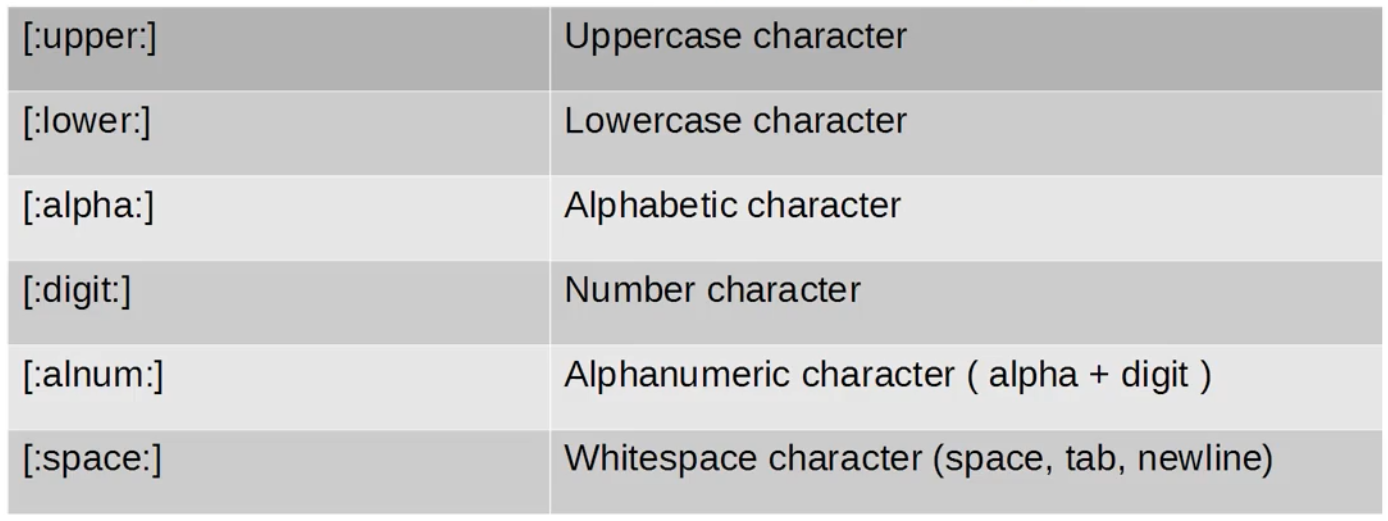
Exit status will return that number

**IF:**

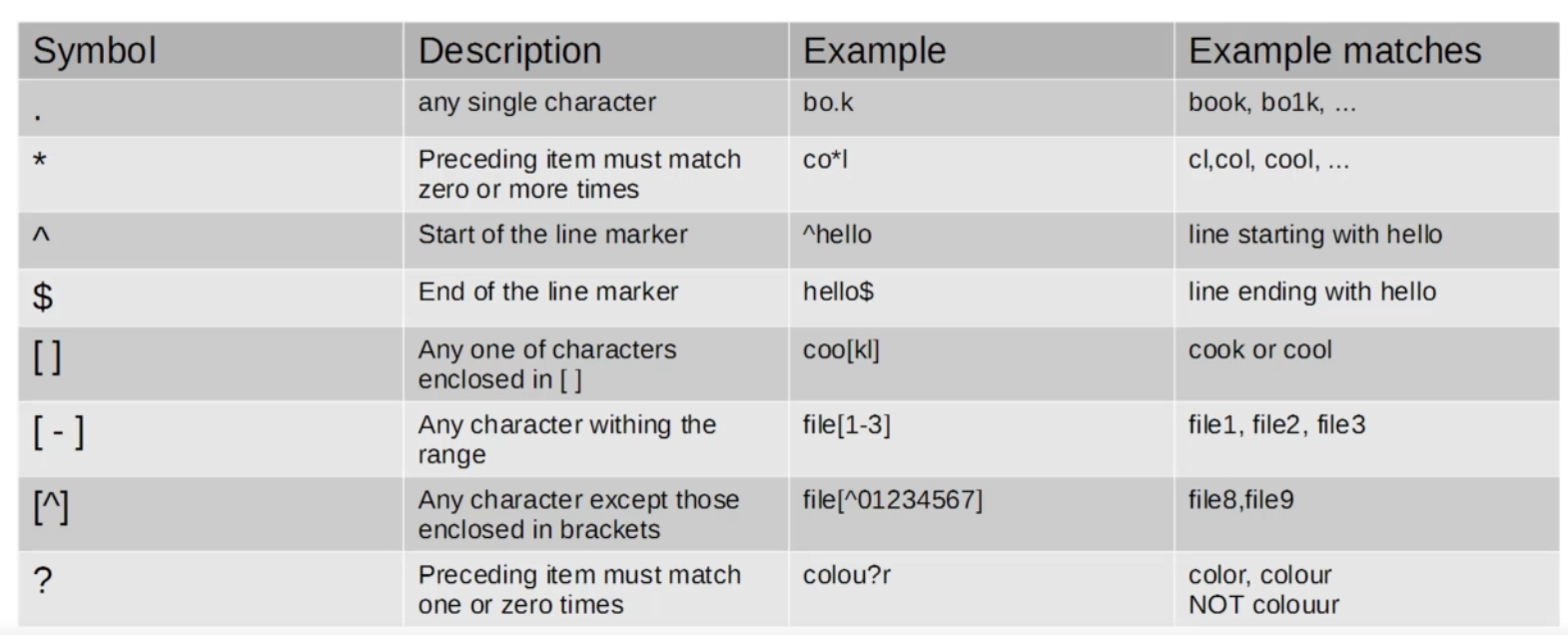


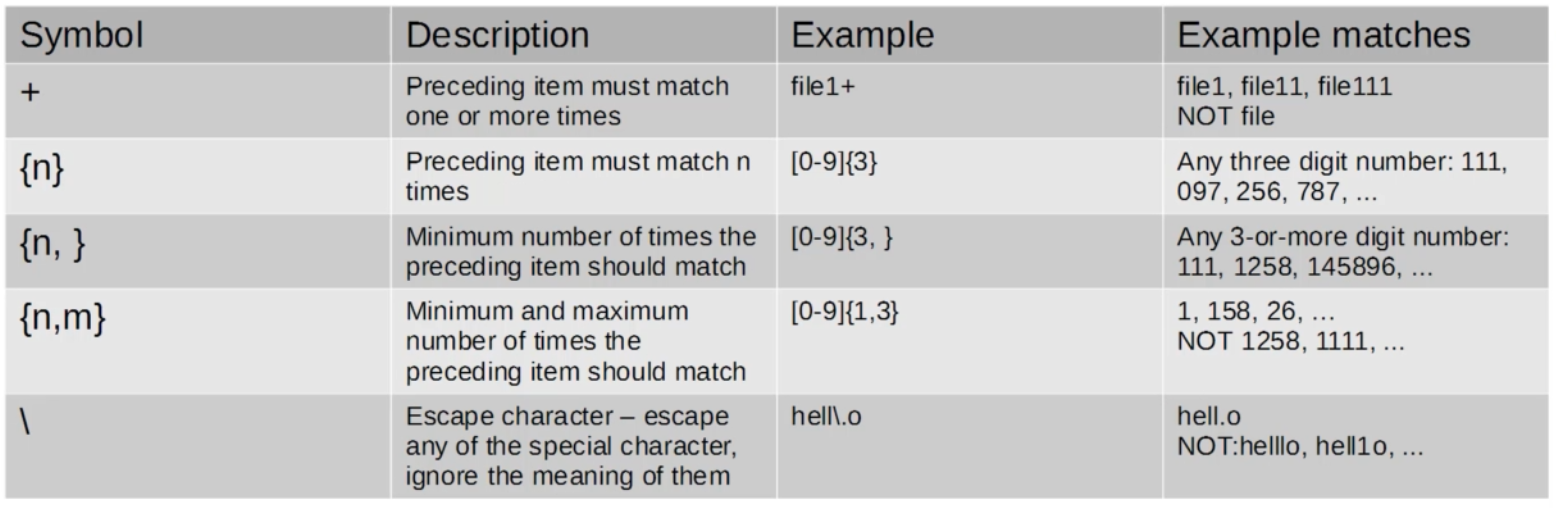
**Wildcards:**

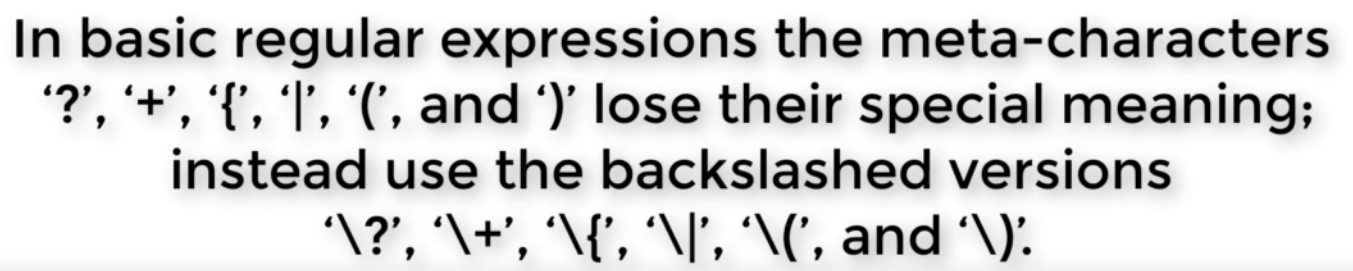




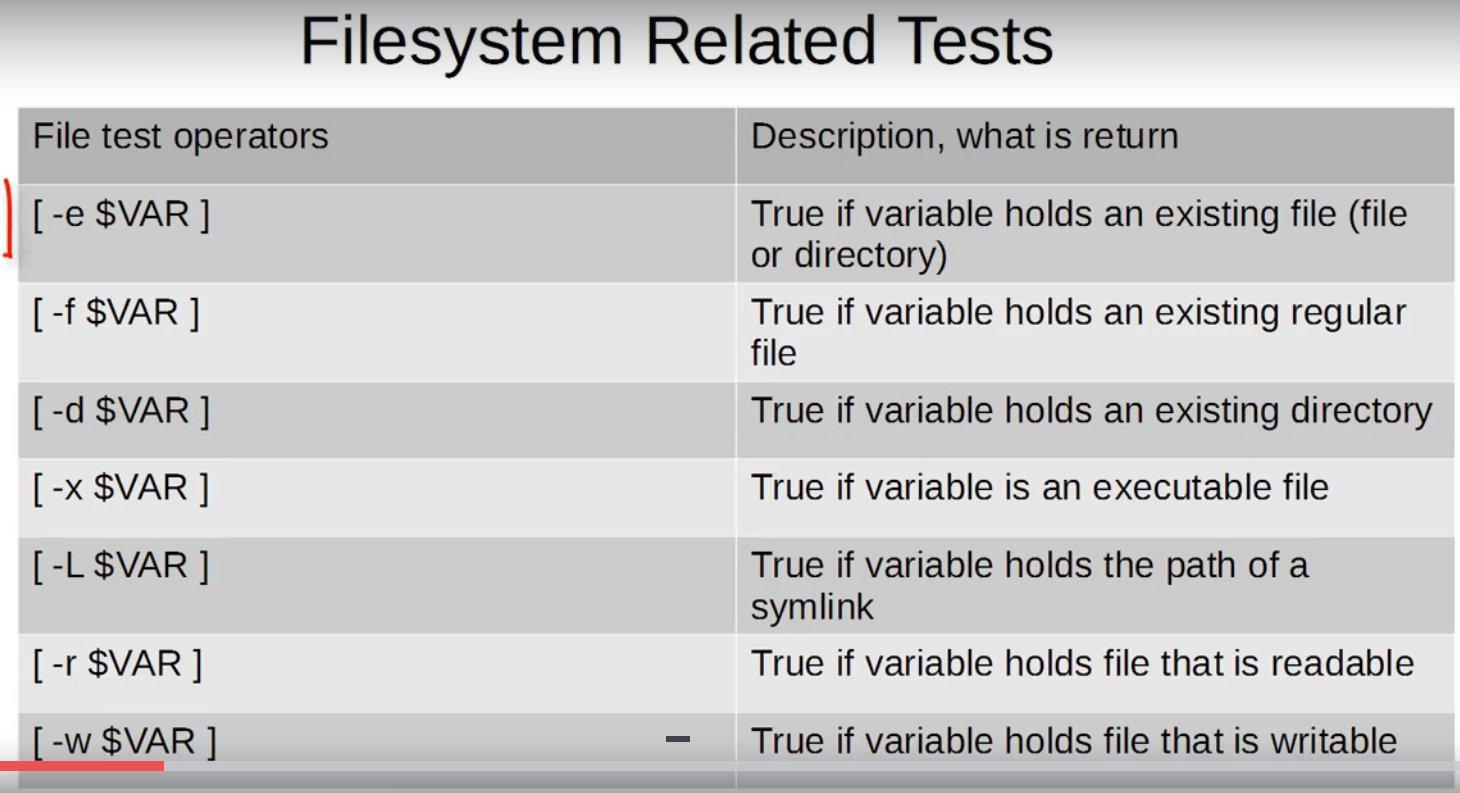
**REGEX:**







ip=${BASH\_REMATCH[0]}

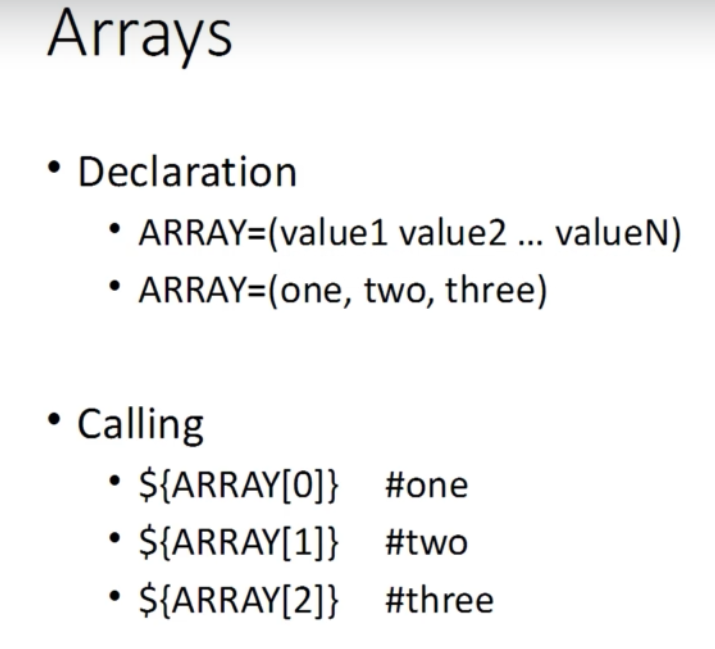


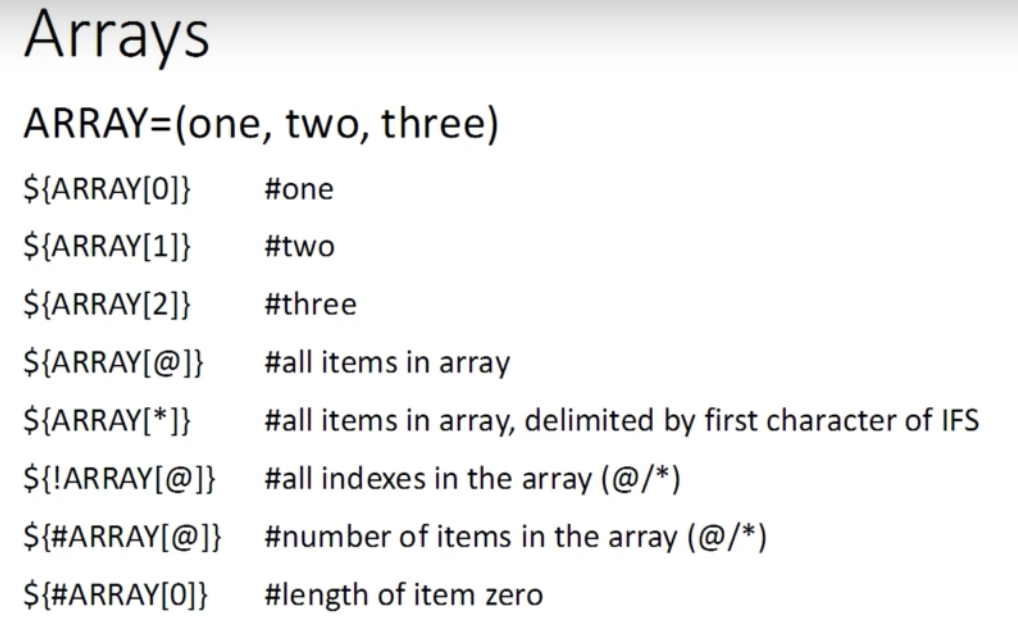
[ -f hello.txt ] && echo "exists" || echo "Doesn't exists"

[ $age -le 20 ] && { echo "You are not allowed"; exit 1 ;} || echo "welcome"

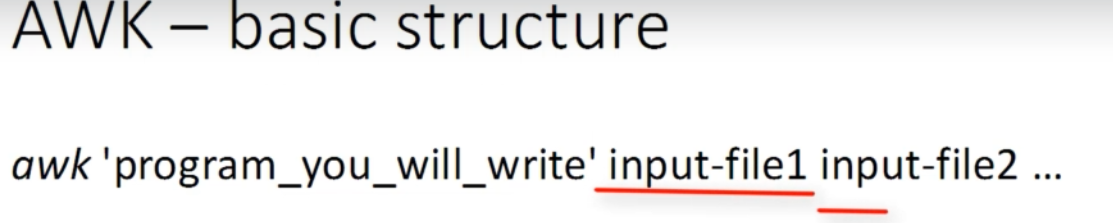
Shift command to remove argument from arg list

**ARRAYS**:





**AWK**:

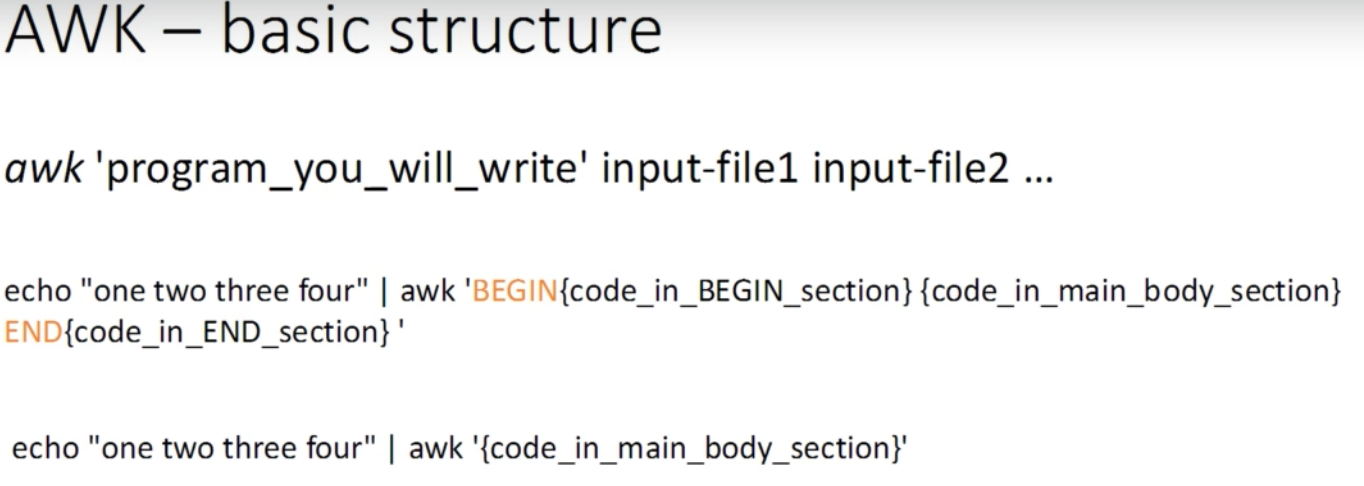


awk ‘

BEGIN{}

{MAIN}

END{}’



awk 'BEGIN{print "Hello"}'

awk '{print "Hello"}'

awk 'END{print "Hello"}'

echo "one two three four" | awk '{print "Hello"}'

awk '{print "Hello"}' awk1.sh

cat awk1.sh | awk '{print "Hello"}'

awk 'BEGIN{print "Begin Part"}{print "Hello"}END{print "End part"}' awk1.sh

echo "one two three four" | awk '{print $0}' and we can use 1,2,3,4..n after dollar

awk '{print $1 "\t" $3}' awk2.sh

awk '{print $1 "\t" $3 "\t" "Hello"}' awk2.sh

cat awk2.sh | awk '/Revanth/ {print $1 " ", $3}' 🡪 // used for searching string

cat awk2.sh | awk '/\tIT/ {print $1 " ", $3}' 🡪 \t will search for patterns where only IT is independent word other than IT in between words like UNIT

cat awk2.sh | awk ' !/name/ {print $0}' 🡪 !/name/ will remove the header if name is there in header section

cat awk2.sh | awk '$1 == "Revanth"' 🡪to get the line of Revanth

echo "one two three four"| awk '{print NF}' 🡪 NF will tell number of fields and $NF will give value of last field

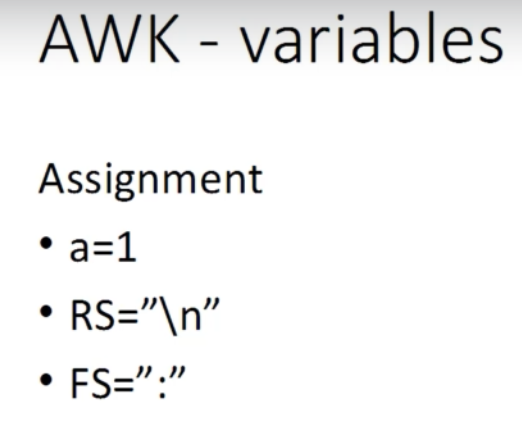
echo "one two three four"| awk '{print $(NF-2)}'

cat awk2.sh | awk '{print "Number of fields in this line: "NF}'

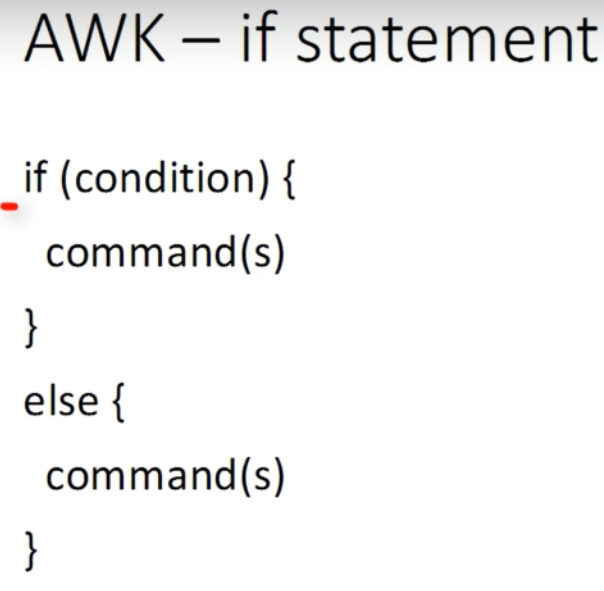
cat awk2.sh | awk '{print NR}' 🡪 prints number of records

cat awk2.sh | awk 'END{print NR}'

cat /etc/passwd | awk 'BEGIN{RS=":"}{print $0}' 🡪 Default RS is new line but we can set it using RS

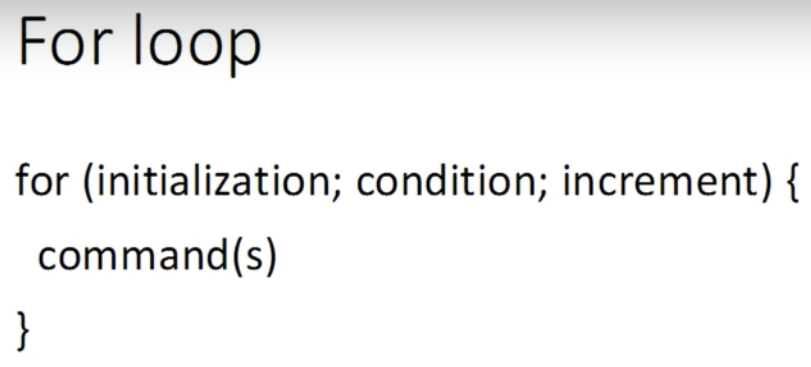


cat /etc/passwd|awk 'BEGIN{RS="\n";count=0}{count++}END{print NR,count}'



cat awk2.sh | awk '{if ($1 == "RevYuv"){ print $0}}' 🡪 to display records of Revanth

cat /etc/passwd | awk -F ":" '{if ($1=="root"){print $1,$6}}'

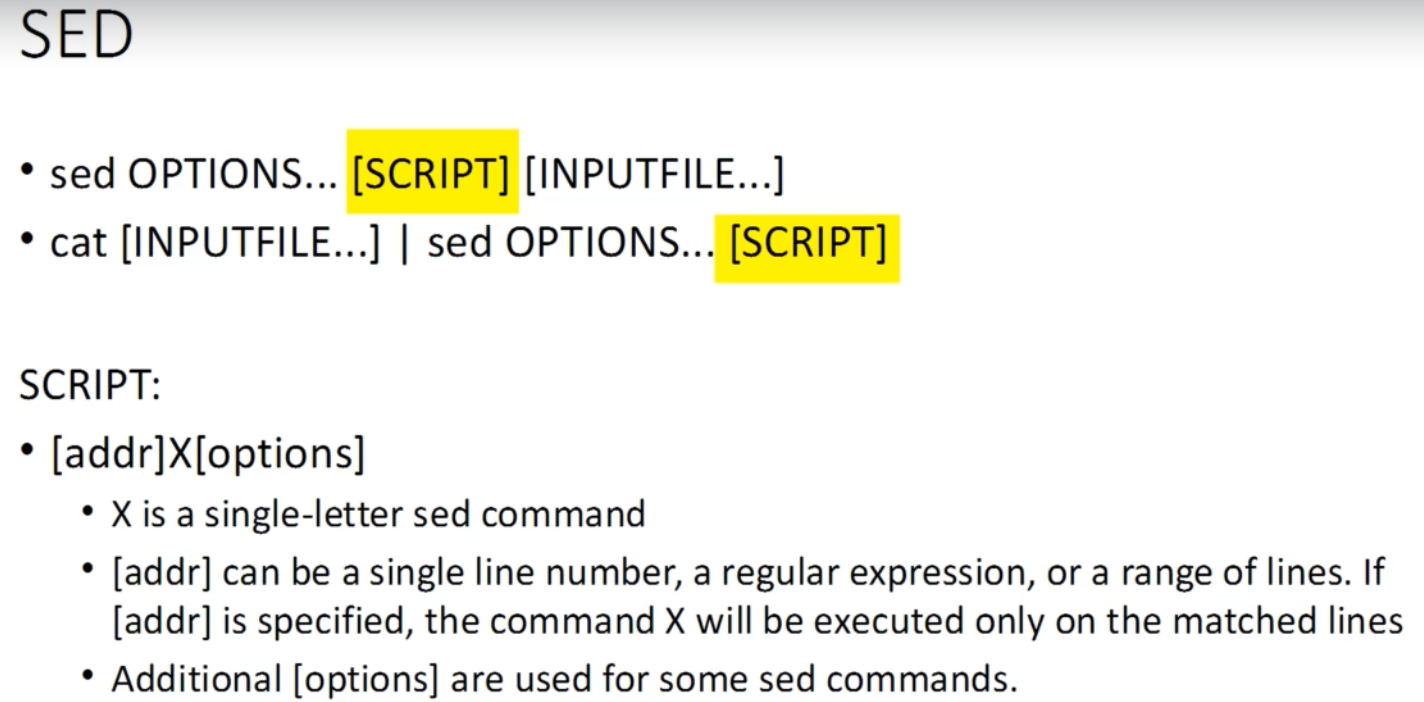


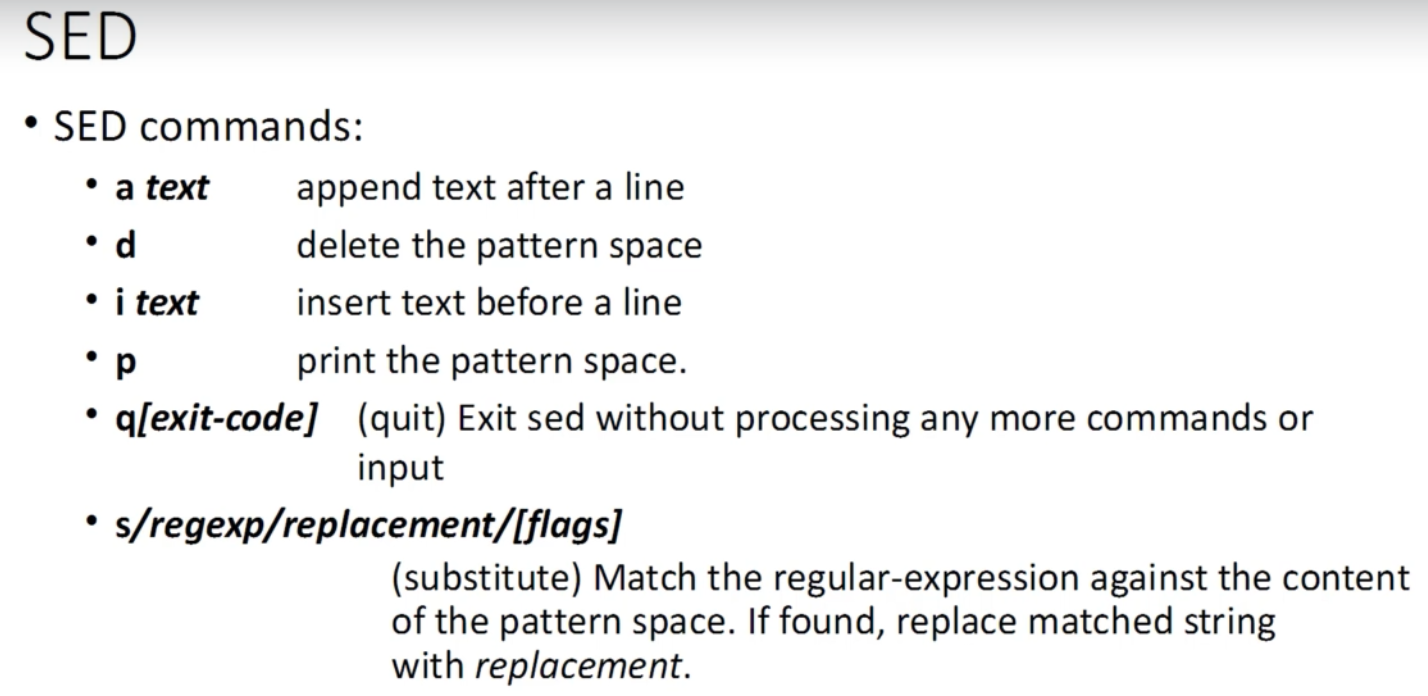
ls -l | awk ' /^-/' | awk '/\.conf$/' 🡪 lists all regular files with conf extension but exit status will be 0 always so use grep 🡪 ls -l | awk ' /^-/' | grep '\.conf$'

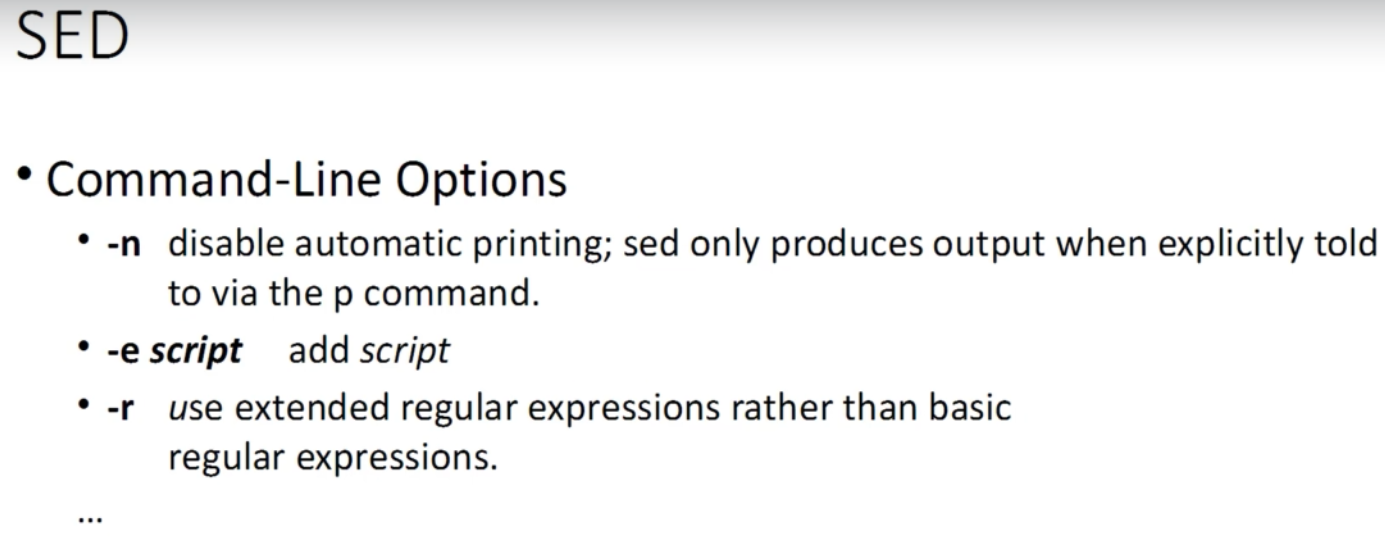
ls -l | awk '{sum=sum+$5}END{print "SUM: ",sum,"B"}' 🡪 adding 5th column which is size of all files.

ls -l $loc|awk '/^-/'|grep "\.$ext$"|awk -v st=$stats -f size.awk 🡪 Use -f to take awk ‘{script}’ from external file and file should not have ‘

**SED:**







sed -n '/Revanth/p' table.txt 🡪 for printing a particular record based on search string

sed -n '3p' table.txt 🡪 print only 3rd line

sed -n '2,7p' table.txt 🡪 print 2 to 7th line

sed -n '4,$p' table.txt 🡪 $ means last line of the file

sed '/IT/a contact: tel. 777 111 222' table.txt 🡪 append contact details after search string IT

sed '/IT/i contact: tel. 777 111 222' table.txt 🡪 insert contact details before search string IT

sed '1i Employee Details' table.txt 🡪 Insert text before 1st line

sed '1a Employee Details' table.txt 🡪 append text after 1st line

sed '/IT/d' table.txt 🡪 delete based on search pattern IT

sed '/\tIT/d' table.txt 🡪 delete only independent string pattern IT using \t

sed '/\tIT/c Hidden Info from IT ' table.txt 🡪 change line which contains IT pattern to text

sed '/Revanth/q2' table.txt 🡪 searching for a pattern and to return exit status

sed -ne '/Rashmi/p' -ne '/Rashmi/q2' table.txt 🡪 use -e to run multiple sed commands

sed -e '/Rashmi/a after Rashmi' -e '/Rashmi/i before Rashmi' table.txt

sed -i '/Ranjan/a after peter' table2.txt 🡪 use I option to change the input file itself

sed '/name/e echo -n "Date: ";date' table.txt 🡪 to execute shell commands used ; in between to execute multiple commands



sed 's/[3-9][[:digit:]]/\*\*\*/' table.txt 🡪 search and replace string

sed 's/IT/& & &/' table.txt 🡪 replace IT with 3 times IT so & will hold search string value

echo "1234 5678 9091 4657" | sed 's/[[:digit:]]\{4\}[ ]\{1\}[[:digit:]]\{4\}[ ]\{1\}[[:digit:]]\{4\}[ ]\{1\}/\*\*\*\*-\*\*\*\*-\*\*\*\*-/' 🡪 to mask cc num with regex

sed -i 's/[[:blank:]]\+$//' "$f" 🡪 to remove spaces at ending of line

sed -i 's/^[[:blank:]]\+' "$f" 🡪 to remove spaces at starting of line

echo "$line" | sed -e '/[[:space:]]\+$/q9' -e '/^[[:space:]]\+/q7' >/dev/null