

www.muppix.co explore directories [begin end last days minutes size greater]

```
mount          ## names & sizes of all connected hard-drives on this version of linux. TIP: goto using these harddrive names
df             ## current hard-drive (mydrive) name, size & available
find -type d   ## select all subdirectory names
find -type d -ls | fgrep -i 'mydir'          ## select all directories & subdirectories with 'mydir' somewhere (in between ) in its name , include the times la
find -type d -name "mydir*" -ls            ## select all directories & subdirectories beginning with 'mydir' , include the saved times and access
find -type d -name "*mydir" -ls           ## select all directories & subdirectories ending in 'mydir' , include the saved times and access
find -type d -name ".*"                    ## select all hidden subdirectory names
du . | sort -n                             ## sort by size each directory (mydir) and subdirectory ie: you've run out of space & need to delete stuff (try doing t
du . | awk '{if($1+0 > 2)print $0}'          ## directory (mydir) and subdirectory sizes greater than 2
du . | awk '{if($1+0 < 2)print $0}'          ## directory (mydir) and subdirectory sizes less than 2
find -type d -mtime -2                      ## select directory and subdirectory with files saved in last 2 days (2 months = 62 days, 2 years = 730 days)
find -type d -mtime -2 -ls                 ## select directory and subdirectory with files saved in last 2 days, include size, saved dates
find -type d -mmin -2 -ls                  ## select directory and subdirectory with files saved in last 2 (second) minutes
```

www.muppix.co explore filenames [begin end filename hidden myextension last days minutes size]

```
find -exec ls -al {} \;                     ## select all files and subdirectories , include so-called hidden files
find -type f -print                          ## selects all files names from all subdirectories
find -iname "*" -ls                           ## select all filenames (include size/date information) in all subdirectories. TIP: to just show the filenames, also a
find -type f -printf '%s %t %p\n' | sort -rk 1 -n ## select all files in subdirectories, sizes and saveddates, sorted by size
find -type f -exec ls -s {} \; | sort -n      ## select all files in subdirectories, sorted by size
find -name *.pdf -print                       ## filenames, myextension is pdf, in all subdirectories, include just the filenames & subdirectory name/path
find -name "*myextension" -ls                ## filenames with myextension in all subdirectories, include size, date, path
find -iname "*myfile*" -ls                  ## select filenames with " in between/somewhere in the filename (include size/date & subdirectory/path inform
find -iname "myfile*" -ls                    ## select only filenames beginning with " (ignore case)
find -iname "*myfile" -ls                    ## select only filenames ending in " (ignore case)
find -type f -printf '%T@ %s %t %p\n' | sort -rk 1 -n | awk '{S1="" ; print $0}' ## select all files in subdirectories, sizes and saved dates. sorted by date
ls -lgo --sort=extension                     ## Sorting files by extension/version, only this directory
find -exec ls -al {} \; | rev | sed 's/\./#/1' | rev | sort -t'#' -k2 | tr '#.' '!' ## Sorting files by extension , subdirectories
find -type f -print | egrep '\.(jpg,gif,png)' ## select jpg or gif or png (myextension) files in any subdirectory
find -type f -print | fgrep -i myfile         ## select files with the name " somewhere/between in it (ignore case) in all directories . show full path
find -size +2 -print                          ## files greater 2K in size
find -size -2 -ls                             ## files less 2K in size
find -size +2k -size -2M -print              ## select files in all subdirectories between 2,000 & 2Megs
find -size +2k -size -2M -ls                 ## select all files in all subdirectories, size between 2K & 2Mb. Each k is 1024 bytes, include sizes, saved dat
find -mtime -2 -name "*" -ls                 ## files in subdirectories saved in last 2 days
find -mtime +2 -print                          ## files saved in last 2 days or older
find -mtime 0 -ls                             ## select files and directories saved between now and 1 days ago (last 24 hours)
find -mmin +2 -mmin -10                      ## files in subdirectories saved between last 2 minutes and 10 minutes ago
find -newer myfile.txt -ls                   ## select all files saved after the time that .txt was saved
find -mtime +2 -size +8 -print                ## files saved in last 2 days or more aswell as greater 8K in size)
find -mtime +2 -size +8 -ls                  ## files saved last 2 days or more aswell as greater 8K in size) include date, size
```

select lines with 'mytext' in files [filename begin end ignore case number aswell mysecondtext]

TIP: goto the top level of the data using 'cd', use the command below and save it to a temporary file in a directory where you can save it. ie: fgrep -rai 'mytext'

```
fgrep -rai 'mytext' *                        ## select lines with 'mytext' in all files in all subdirectories (entire hard drive / network), ignore case, include filename
egrep -rai 'mytext|mysecondtext|mythirdtext' * ## select lines with 'mytext' or 'mysecondtext' or 'mythirdtext', ignore case, in all directories
fgrep -raic mytext m | fgrep -v '0'          ## how many times is 'mytext' is found in files in all subdirectories, ignore case
find -type f -print0 | xargs -0 grep -ai "mytext" ## select 'mytext' (ignore case) from all subdirectories & select the directories, filenames & the lines if 'my
find -name "myfile[ABC]*" -print | xargs grep -aHE 'mytext' ## select files with 'A.myfileB or myfileC somewhere in the filename in subdirectories, and only
find -name "myfile" -print0 | xargs -0 grep -aHE 'mytext' ## select only filenames ending in 'name' (myextension) and select 'mytext'
find -name "*myextension" -exec cat {} \;     ## select all lines in all the files with myextension. ie: glue together all *.txt files into one single file
find -name "*myextension" -exec grep -H " {} \; ## select lines with myextension in all subdirs & include filename. ie: glue together all *.txt files into o
find -name "myfile*" -print0 | xargs -0 grep -aHE '^mytext' ## select 'mytext' at beginning of the line, but only in files with " between/in its filename.
find -name "*myextension" | xargs fgrep -a 'mytext' ## select 'mytext' in files with myextension
find -name 'myfile*mytext' -print0 | xargs -0 grep -a [alnum] ## all lines of filenames beginning with " aswell as 'mytext' in the filename in all subdirs
find -name "*.txt" -exec awk '{if ($1 ~/[A-Z][A-Z]) print $0}' {} \; ## in any subdirectory for files with 'txt' extension, select lines if there are atleast 2 cons
find -name "*.txt" -exec egrep -H "\b[0-9]{2,}\b" {} \; ## select lines in any subdirectory if there is an integer of atleast 2 consecutive number /digits ie: 09
find -type f -print0 | xargs -0 egrep ('mytext|mysecondtext|mythirdtext') ## select lines in subdirectories with 'mytext' or 'mysecondtext' or 'mythirdtext'
find -type f -print0 | xargs -0 grep -o "mytext.*mysecondtext" ## select lines with 'mytext' aswell as 'mysecondtext'
find -print | xargs grep -iaC2 'mytext'      ## select 'mytext' from all files/subdirectories, and also select 2 (second) lines above and below 'mytext' (ignor
find -print | xargs grep -aB2 'mytext'       ## select 'mytext' in all files & subdirectories, and also select 2 (second) lines above 'mytext' Address Pattern
find -print | xargs grep -aA2 'mytext'       ## select 'mytext' in all files & subdirectories, and also select 2 (second) lines below 'mytext' Address Pattern
find -print | xargs grep -aC2 'mytext'       ## select 'mytext' in all files/ subdirectories, and also select 2 (second) lines above and below 'mytext' Address
fgrep -H 'mytext' */**/*                     ## select lines with 'mytext' in all files but only the third level subdirectories below this one, include filenames
find -mtime -2 -size -2k -name 'myfile*' -print | xargs grep -ias 'mytext' ## select 'mytext' (ignore case) in files saved in last 2 days, size less than 2K (not great
find -mtime +2 -size +2k -name '*myfile' -print | xargs grep -ias 'mytext' ## select 'mytext' (ignore case) in files saved in last 2 days or more, size greater than 2
find -mtime -2 -print | xargs grep -ias 'mytext' ## lines containing 'mytext' in files saved in last 2 days
find -mmin -2 -print | xargs grep -ias 'mytext' ## lines containing 'mytext' in files saved in last 2 minutes
find -mmin -2 -size -10k -print | xargs grep -ias 'mytext' ## files saved today's date, in last 2 minutes, size less than 10K (not greater than 10K)
fgrep -rif /cygdrive/c/muppix/mylist.txt *   ## select lines from a list of text/words in the file mylist.txt, (mytext or mysecondtext or ThirdText etc) all
find -exec grep -if /cygdrive/c/muppix/mylist.txt {} \; ## select lines from a list of text/words in the file c:/mytext/mylist.txt, (mytext or mysecondtext or Third
find -name "*.txt" -exec grep -IHf /cygdrive/c/muppix/mylist.txt {} \; ## select lines from a list of text/words in the file c:/muppix.mylist.txt, (mytext or mysec
TIP: on a Windows PC, ensure you run dos2unix dos2unix on mylist.txt!
```

select line with 'mytext' [begin end before after aswell or mysecondtext mythirdtext word ignore]

```
TIP: now you have your temporary file such as 'myfile.txt', type 'cat myfile.txt' followed by a command shown below. ie: cat myfile.txt | fgrep -i 'mytext' | w
fgrep -i 'mytext' ## select line with 'mytext' ignore case, ie: could match MytEXT mytext or MYTEXT etc
fgrep 'mytext' ## select if 'mytext' anywhere on the line
fgrep 'mytext' | fgrep 'mysecondtext' ## select line with both 'mytext' aswell as 'mysecondtext' in any order on the line
fgrep -i 'mytext' | fgrep -i 'mysecondtext' ## select line with 'mytext' aswell as 'mysecondtext' on the line (ignore case)
fgrep -i 'mythirdtext' | fgrep -i 'mysecondtext' | fgrep -i 'mytext' aswell as 'mysecondtext' aswell as 'mythirdtext' in any order (ignore cas
fgrep -e 'mytext' -e 'mysecondtext' ## select either 'mytext' or 'mysecondtext'
egrep -i 'mytext|mysecondtext|mythirdtext' ## select line with 'mytext' or 'mysecondtext' or 'mythirdtext', ignore case
fgrep -if mylist.txt ## select any of the texts in the file mylist.txt 'mytext' or 'mysecondtext' or 'mythirdtext' etc TIP: in Windows ensure
fgrep 'mytext' ## select line that begin with 'mytext' TIP:may first want to ensure there are no leading spaces
fgrep '^mytext[ABCD]' ## select line that begin with (range) 'mytextA' or 'mytextB' or 'mytextC' or 'mytextD'
fgrep 'mytext$' ## select line ending with 'mytext'
egrep '[.,\?]' ## select the punctuation character '.', and '?' These are special characters. To use the character as literal character, it ha
egrep '[ABCD]' ## select line that begin with character 'A','B','C' or 'D' (range)
egrep '^[ABCD]' ## delete line that begin with character 'A','B','C' or 'D' (range)
grep '([A-Za-z][A-Za-z])' ## select line with 2 character US state names, surrounded by round braces. mychar '(', range A-Za-z, range
fgrep -iw 'myword' ## select line with the word 'myword', ignore case. so will select 'MYWord' or 'myWORD', but wont select 'myw
awk '$0 ~mytext.*mysecondtext/' ## select line where 'mytext' is before 'mysecondtext', 'mysecondtext' after 'mytext'
awk '$0 ~mytext.*mytext/' ## select line where 'mytext' appears twice or more often - second occurrence
egrep 'mytext(?:.*)mysecondtext' ## select line if the text 'mytext' is then followed/before/after somewhere on the same line with 'mysecondtext'
awk '$0 ~mytext.*mysecondtext.*mythirdtext/' ## line with 'mytext' before 'mysecondtext' aswell as followed after by 'mythirdtext'. ie: line with many
egrep '[.,\?]' ## select the punctuation character '.', and '?' These are special characters. To use the character as literal character, it ha
egrep 'bmytextw\b' ## select line with word/column beginning with 'mytext' ie: 'mytextualisation'
egrep 'b(w+)$s+1\b' ## select line with word/column ending in 'mytext'. ie: find words/columns ending in 'ion' 'ing'
awk 'if ($2 ~/[A-Z]{2,}) print $0' ## select line if second column begins with 2 consecutive uppercase characters (range) after / before
grep '^b[[:upper:]]{2,}' ## select line that begin with a word of atleast 2 uppercase characters (range)
awk 'if ($0 ~/[A-Z][A-Z][0-9][0-9]) print $0' ## select line with 2 consecutive uppercase characters and 2 numbers anywhere on the line (range)
egrep 'mytext$' ## select line with words that end with 'mytext' ie: egrep 'ing$' will find 'beginning' (starting) '++[going]'
egrep 'b(w+)$s+1\b' ## select line with words that are duplicated, any word immediately followed/after by same second word
egrep 'mytext(?:.*)mytext' ## select line if 'mytext' appears atleast twice duplicated - also for a second time on each line
sed -n '/[a-zA-Z][a-zA-Z]*\|1/p' ## select line if any text is duplicated, any text after/before the same second text
egrep '[A-F][g-k][0-9]' ## select words/text anywhere on line beginning with a range of characters A or B,C,D,E,F followed/after by a
fgrep -w 'myword' ## select line with the exact word 'myword' on the line, so wont select line with 'mywords' or 'allmyword' for exam
```

select a section of lines [lines above below mytext after blankline between mysecondtext]

```
awk 'IGNORECASE=1; {print $0;if (match($0,"mytext"))exit}' ## select begin lines above and including 1st occurrence of 'mytext'.(ignore case) delete all line
awk '{print $0; if (match($0,"mytext")) exit}' ## select begin lines above and including 'mytext', delete all lines below
awk '{if(match($0,"mytext"))exit;print $0}' ## select begin lines above 'mytext' (but not including mytext) delete 'mytext' and all lines below/end
awk '{print $0; if (match($0,"mytext")) exit}' ## select begin lines above and including 'mytext', if 'mytext' is at beginning of line. delete all lines below
awk '{print $0; if (length($0)==0) exit}' ## select the beginning lines (above) upto the beginning blankline, delete lines below beginning blankline T
sed 'mytext/./</p>/d' ## select beginning lines above 'mytext', delete all lines below 'mytext'
sed -n 'mytext/.$p' ## select lines below 'mytext' to end of file, including 'mytext'. delete beginning lines above 'mytext'
sed '/^mytext/./</p>/d' ## if 'mytext' at the beginning of the line, delete this line & lines below. select beginning lines above 'mytext'
sed -n '!G;h;$p' | awk '{print $0;if (length($0)==0) exit}' | sed -n '!G;h;$p' ## select all lines below the final ending blankline, delete all lines up to the final e
awk '/^$/ {p=1;next}; p==1 {print $0}' ## delete all lines up to the beginning blankline, select all lines below beginning blankline TIP:may need t
awk '/^$/ {p++;next}; p>1 {print $0}' ## select all lines below the second blankline. (delete all lines above the second blankline)
awk 'mytext/{p++;next}; p>1 {print $0}' ## select all lines below the second 'mytext'. (delete all lines above the second 'mytext')
awk '{print $0; if (match($2,"mytext")) exit}' ## select begin lines above and including 'mytext', if 'mytext' is in second column. delete all lines below
awk 'match($0,"mytext"),match($0,"mysecondtext")' ## select section of lines between / below 'mytext' and 'mysecondtext'
awk '$2=="mytext",$2=="mysecondtext"' ## select section of lines between the beginning line with 'mytext' in second column to 'mysecondtext' in
awk 'NR>=2&&NR<=20' ## select section lines between second (fixed) line to line 20 (fixed) ie: cat -n |awk 'NR>=2&&NR<=20'
awk '{if ((p==0)&&match($0,"mytext")){p=3} else {if(p<=2) {print $0} else {if ((p==3)&&match($0,"mysecondtext")) p=1}}}' ## delete section lines bet
tr '\n' 'E' | sed 's/mytext.*mysecondtext/g' | tr 'E' '\n' ## delete section lines between begin 'mytext' and end occurrence of 'mysecondtext'
```

delete lines [begin end above below duplicate blanklines]

```
touch myfile.txt >myfile.txt ## empty-out entire contents/delete all lines in .txt
sed '1 d' ## delete just the beginning (fixed) line, select below beginning line
sed 'd' ## delete just the end (fixed) line, select all lines above
fgrep -iv 'mytext' ## delete line if 'mytext' is somewhere on the line (ignore case) TIP: first dble check which line will be deleted by ru
fgrep -v 'mytext' ## delete line if 'mytext' is somewhere on the line TIP: dont ignore case & also first check which lines will be delete
grep -v '^mytext' ## delete lines that begin with 'mytext'
grep -v 'mytext$' ## delete lines that end with 'mytext'
egrep -v 'mytext|mysecondtext' ## delete lines with 'mytext' or 'mysecondtext'
egrep -iv 'mytext|mysecondtext' ## delete line with 'mytext' or 'mysecondtext' anywhere on the line (ignore case)
fgrep -vif mylist.txt ## delete lines if any of the texts in the file mylist.txt are found, 'mytext' etc TIP: in Windows ensure you run dos2
awk 'BEGIN {} {print l; l=$0} END {if($0 !~mytext/){print $0}}' | sed '1 d' ## if 'mytext' somewhere in the end line, delete the line
awk '{if(NR=1)&&($0 ~mytext/)} else {print $2}' ## if 'mytext' somewhere in the begin line, delete the line
awk '{if($0 ~mytext/&&mysecondtext/){}'} else {print $0}' ## delete line with 'mytext' aswell 'mysecondtext' anywhere on the line
egrep -v 'mytext(?:.*)mysecondtext' ## delete lines with 'mytext' before 'mysecondtext' ('mysecondtext' after 'mytext'
awk NF ## truly delete all blanklines which may have some spaces or tabs or no spaces at all
sort -u ## sort & delete duplicate lines (dont maintain the original order & is a lot faster)
awk 'x{S0}++' ## select only the duplicate lines, ie: those lines that occur twice or more
sed 'mytext/./mysecondtext/d' ## delete duplicate lines, but maintain the original order ( without sorting) select begin occurrence of each line
awk '!($2 in a){a[$2];print $0}' ## delete lines that begin with 'mytext' aswell as end 'mysecondtext' ie: xml tags: sed '/./</myxml>/d'
egrep -v 'mytext(?:.*)mytext' ## delete duplicate lines, based on duplicates in second column only, select begin occurrence of 2nd column,p
sed 'mytext/./</p>/d' ## delete all lines below 'mytext', select beginning lines above 'mytext'
sed '/^mytext/./</p>/d' ## delete all lines below 'mytext' is in the beginning of the line, select beginning lines above 'mytext'
sed '/./^$/d' ## delete multiple/duplicate/consecutive blanklines except the beginning line; also deletes blanklines from beginning a
sed '/^$/N;/\n$/N;/D' ## delete all multiple/duplicate/consecutive blanklines except the beginning and second
sed '1,2d' ## delete the (fixed) beginning and second lines, select lines below second line, to the end line
sed '2,8d' ## delete between second line to eighth line : (fixed) lines 2 3 4 5 6 7 8
sed -e 'a -e $d;N;2,3ba' -e 'P;D' ## delete the end (fixed) 3 lines, including second line, select all lines above the end 3 lines
sort -n | uniq -c ## how many/occurrence of duplicate lines - pivot table
```

delete 'mytext' in the line [begin end before after between number second mychar mydelimiter wor

```

sed 's/mytext/g'          ## delete 'mytext' on the line if found
awk '{S2="" ; print $0}'  ## delete second column / word (delimiter is spaces by default)
awk '{S1=S2=S3=$NF="" ; print $0}' ## delete begin , second , third, end column
sed 's/mytext/2'        ## delete only the second occurrence of 'mytext' on each line
sed 's/mytext.*//g'     ## select everything before 'mytext' on the line, (delete 'mytext' & everything after)
sed 's/*mytext/g'       ## delete everything before 'mytext' on the line, (select all text after 'mytext')
awk '{SNF="" ; print $0}' ## delete end word / end column
awk -v v="" , "BEGIN {FS=OFS=v} {SNF="" ; print $0}' ## delete end word / end column with comma ',' as mydelimiter
cut -c 3-                ## delete beginning and second characters (fixed) from each line (delete before 3rd & select after second characters)
sed 's/./'              ## delete end character (fixed) on each line
sed 's/./'              ## delete the end 2 (fixed) characters on each line. (end & second from end character)
sed 's/./'              ## delete end character if its a comma ( is mychar)
awk '{S1="" ; print $0}' ## delete beginning word / column
sed -e 's/^/[ \t]*//g'   ## left align /justify, delete beginning/leading spaces and or tabs on each line
cut -c 3-                ## delete beginning and second characters (fixed) from each line (delete before 3rd & select after second characters)
sed 's/./'              ## delete end character (fixed) on each line
sed 's/./'              ## delete end character if its a comma ( is mychar)
awk '{S1="" ; print $0}' ## delete beginning word / column
sed 's/[ \t]*$//g'      ## delete spaces or tabs at end of each line. right align. also deletes extra spaces on blanklines
sed 's/[ \t]*//;s/[ \t]*$//g' ## delete leading/beginning space aswell as ending/trailing spaces on the line(left align, trim )
sed -n -e 's/*mytext(.*)mysecondtext.*\1/p' ## select between 'mytext' to end occurrence of 'mysecondtext' on each line. will find 'anymytext' to the so
sed -n -e 's/*mytext(b(.*))bmysecondtext.b.*\1/p' ## select text between the exact words 'mytext' and 'mysecondtext' on the same line, delete before 'myte
grep -o "mytext.*mysecondtext" ## select text between 'mytext' and 'mysecondtext' on the line. delete before 'mytext' aswell as after 'mysecond
sed 's/mytext.*mysecondtext//g' ## delete the text between 'mytext' and 'mysecondtext'
sed 's/mytext##2' sed 's/##.*//g' ## delete everything after second occurrence of 'mytext', select everything before 2nd occurrence of 'mytext'
sed 's/mytext##2' sed 's/##.*//g' ## delete everything before second occurrence of 'mytext', select everything after 2nd occurrence of 'mytext'
sed 's/[^ ]*mytext*[ \t]*//g' ## delete words/columns anywhere on the line, with 'mytext' somewhere inside/between the word ie: will delete
awk -v OFS="" ' $1=$1' ## delete/replace all multiple/duplicate/consecutive spaces with single space/blank
tr -d 'a'                ## delete 'a' characters ('a' is mychar/mytext)
tr -d 'abc'              ## delete all occurrence of any of these 3 single (mychar) character 'a','b' or 'c' (ie: also delete 'abc', dan' etc) delete mult
tr -d '"'                ## delete double quote character (mychar/mytext)
tr -d "'"                ## delete single quote character (mychar/mytext)
sed 's/*mytext//g'      ## delete everything on the line before the end occurrence of 'mytext' including 'mytext' . select everything after end
rev | cut -d '/' -f2 | rev ## delete everything on the line after the end occurrence of '/' (mychar) , select everything before end mydelimiter (/)
rev | sed 's/*mychar/@/g' | cut -@ '/' -f2 | rev ## delete everything on the line after the end occurrence of 'mychar' , select everything before end 'mychar'
tr -dc '\11\12\40-\176' ## delete all non-printable punctuation characters TIP: initially use this when you get "Binary file (standard input)
sed 's/[^\a-z0-9A-Z]/ /g' ## replace punctuation with space (delete all punctuation)
tr '[:punct:]' ' '      ## replace punctuation with space (delete all punctuation)

```

www.muppix.co select / delete columns [mytext begin end second or delete mychar mydelimiter

```

awk '{print $1}'          ## select beginning column only
awk '{print $2}'          ## select second column
awk '{print $2}' FS=","   ## select second column, but using ',' comma as mydelimiter
awk '{print $NF}'        ## select only the end column, delete all columns before the end column
awk '{print $2,$NF}'     ## select second column and end column
cut -d ' ' -f2-8         ## select between second column and 8th column
awk '{if($1 == "mytext") print $0}' ## select line if begin column is 'mytext'
awk '{if($NF == "mytext") print $0}' ## select line if end column is 'mytext'
awk '{if($2 == "mytext") print $0}' ## select line if second column is 'mytext'
awk -v v="" "BEGIN {FS=OFS=v} {if($2=="mytext")print$0}' ## select line if second column is 'mytext', but column mydelimiter is '/'
awk '{if ($2 ~mytext/mysecondtext/mythirdtext) print $0}' ## select whole line if 'mytext' or 'mysecondtext' or 'mythirdtext' is somewhere in the second column
awk '{if ($2 ~mytext) print $0}' ## select line if second column begins with 'mytext'
awk '{if ($2 ~mytext$) print $0}' ## select line if second column ends with 'mytext'
awk '{print $(NF-1)}'     ## select only the second from end column , delete all other columns
awk '{S2=S3=$5="" ; print $0}' ## delete second aswell as third fifth (2 3 5) columns , regardless how many columns there are
awk '{if ($2 == "mytext" ) print $1,$2,$3,$4 }' ## select column 1,2,3,4 if second column is 'mytext'
awk 'BEGIN {z="mytext"; if (substr($0,2,length(z))==z) print $0}' ## select line if (fixed) character columns 2-7 is 'mytext' (from second character, for 6 ch
awk '{if ($2 !~mytext)print$0}' ## delete line if second column is 'mytext'
sed -e 's/^[[:alnum:]]*[Aa]/>/&/g' ## delete words/columns ending in 'A' or 'a' (range)
awk 'NF > 2'             ## select line with more than/greater 2 columns length (delete lines with begin and second columns) length
awk '{S0 ~mytext} || ({S1 ~mysecondtext} && {S2 ~mythirdtext})' ## select if ( the whole line contains the word 'mytext' ) or ( the beginning colu
cut -d ' ' -f2-         ## select second column (using mydelimiter ',') & all columns after 2, (split lines)
sed 's/*mytext//g'      ## delete 'mytext' if it is at the beginning of the line
sed 's/mytext$/'        ## delete 'mytext' if it is at the end of line
head -2                 ## select the beginning (fixed) begin and second lines (above), delete lines below second line
tail -2                 ## select (fixed) end line and second from end line , delete beginning/above lines. ie: tail -100 , end 100 lines TIP:useful
tail -1000f myfile.txt ## select the ending 1000 lines of .txt and continue showing any new updates to the file
awk 'NR>=2'             ## select the second (fixed) lines & below , delete lines above second line
sed '2,881d'            ## select fixed line, between second line to 88th line, useful in splitting up a file

```

research: select lines with 'mytext' and also lines above or below

```

fgrep -B2 'mytext'      ## select the line with mytext, aswell as the beginning and second lines above each mytext - near Address Pattern
fgrep -A2 'mytext'      ## select the line with mytext, aswell as the beginning and second lines below each mytext - near Address Pattern i
fgrep -C2 'mytext'      ## select 'mytext', aswell as the beginning and second fixed lines above & below 'mytext' - near Address Pattern
awk 'length > 2'        ## select line greater than (fixed) 2 characters length (second) , delete lines smaller than 1 2 ( < less than)
awk 'length>max{max=length;lin=$0} END {print lin;}' ## select the longest line
egrep '<w{2}'            ## select lines with a word/column of length of 2 characters (second)
egrep '<w{2,}'           ## select lines with a words/column of length of 2 characters or more (second)
egrep '<w{2,8}'          ## select lines with word/column of length of 2 to 8 characters (second)

```

numbers or values [greater smaller equals number end begin second column delete]

```

egrep [0-9]             ## select lines with a number (range) somewhere on the line
grep -v [0-9]           ## delete lines with a number (range) somewhere on the line
awk '{for(i=1;i<=NF;i++){if(($i+0) > 2.0){print $0;i=NF}}}' ## if a number on the line is greater than 2.0 ,select whole line. range TIP: number must be 1234
awk '{for(i=1;i<=NF;i++){if(($i+0) < 2.0){print $0;i=NF}}}' ## if a number on the line is less than 2.0 ,select whole line. range TIP: number must be 1234 &
awk '{if($1+0) > 2.0} print $0}' ## select line if begin column has a number/value : is greater than 2.0
awk '{if($2+0) > 2.0} print $0}' ## if second column has a number/value : is greater than 2.0, select whole line. TIP: '> 0' is the same as select
awk '{if($1+0) < 2.0} print $2,$3,$1 }' ## begin column has a number/value : is smaller than 2.0, select second,third and third column
awk '{(SNF+0) >= 2.0}' ## select line if end column has a number/value : is greater or equals than 2.0
egrep [0-9]{2}'        ## select lines with 2 consecutive numbers
egrep 'b(100|1-9)?[0-9]b)' ## select lines if there's a number between 0-100, greater than 0 TIP: wont find number '2.0' TIP: use awk exa
egrep 'b[0-9]{2,}b)'   ## lines have a numbers with length of 2 or consecutive more/greater numbers (second), somewhere on the line
sed 's/^/ /; s/ *(\{7,\})\1/' ## right align numbers / format
grep [0-9]{2,}'        ## lines with atleast 2 consecutive numbers/digits, or more (length)
awk '{if (($2+0) > 2.0) {$2="mytext" $2;print $0} else print $0}' ## insert 'mytext' before second column, if 2nd column is greater than number/value 2.0
tr -d '[digit]'        ## delete all numbers on the line (range of characters 0-9)
sed 's/[^\0-9].*/g'     ## select numbers before characters , delete characters after the numbers
egrep '[^\0-9]'        ## delete lines with just numbers (lines beginning with just single integer amount) (can select the range/character set
sed 's/[0-9]/g'         ## delete all numbers/digits
egrep [0-9]{5}'        ## select US zip codes (5 fixed numbers ) anywhere on the line
awk '{($2 + 0) != 2.0}' ## if second column is NOT equals to 2.0 ie: column could show 10.000, 10, 010, delete that whole line
awk '{($2 + 0) == 2.0}' ## if second column has a number/value : is exactly equals to 2.0 ie: column could select 10.000, 10, 010, select
grep [0-9]{2,}mytext' ## lines with atleast 2 numbers before mytext. mytext is after atleast 2 numbers

```

replace or convert text [mysecondtext beginning ignore case mythirdtext begin end line mychar du

```

sed 's/mytext/mysecondtext/g'          ## replace every 'mytext' with 'mysecondtext'
sed 's/mytext/mysecondtext/g'         ## replace every 'mytext' with 'mysecondtext', ignore case of 'mytext'
sed '/mytext/c/mysecondtext'          ## if 'mytext' is found anywhere on line, replace whole line with 'mysecondtext'
sed 's/(.*)mytext\1/mysecondtext/g'   ## if 'mytext' is at the end on the line, replace with 'mysecondtext'
sed 's/mytext/mysecondtext/1'        ## replace only the beginning occurrence of 'mytext' on each line with 'mysecondtext'
sed 's/mytext/mysecondtext/2'        ## replace only the second occurrence of 'mytext' on each line with 'mysecondtext'
rev | sed 's/mychar/mysecondchar/1' | rev ## replace end occurrence of 'mychar' with 'mysecondchar'
sed 's/mytext/E/g' | rev | sed 's/E/#/1' | rev | sed 's/#/mysecondtext/1' ## replace end occurrence of 'mytext' with 'mysecondtext' TIP: ensure chars 'ÁÊ'
awk '/mythirdtext/{gsub(/mytext/, "mysecondtext");}' ## replace 'mytext' with 'mysecondtext' only on lines containing 'mythirdtext'
awk '/mythirdtext/{gsub(/mytext/, "mysecondtext");}' ## replace 'mytext' with 'mysecondtext' only on those lines NOT containing 'mythirdtext'
sed 's/mytext/(.*)mysecondtext/mytext\1/mythirdtext/' ## select 'mytext' on the line, then for the remainder of the line replace (the end occurrence of) 'mysec
sed '2 c mytext'                       ## replace second (fixed) line with 'mytext'
sed 's c mytext'                       ## replace end line with 'mytext'
sed -e 's/mytext.*mysecondtext/'      ## replace everything after 'mytext' with 'mysecondtext', replacing 'mytext' and everything after 'mytext'
sed 's/^/$mytext/g'                   ## replace blanklines with 'mytext', insert 'mytext' TIP: may need to ensure is truly blankline
sed 's/^.*[0-9]mytext[AB]/ /g'        ## delete/replace words beginning or ending with a range fixed number/text. ie: 8mytextA or 3mytextB anywhere
awk '{gsub("mytext",w+*,");print}'    ## delete/replace all words beginning with 'mytext'
awk 'mytext$/{sub(/mytext$/, ""); getline t; print $0 t; next}; 1' ## if 'mytext' at end of line, glue the line below after this line
awk -v OFS=" " '$1=$1'                ## replace all multiple/duplicate/consecutive spaces with single space, delete spaces, compress text
awk '{if ($1 ~ /^mytext/) $1="mysecondtext"; print $0}' ## if begin column is 'mytext', replace with 'mysecondtext'
awk '{if ($2 ~ /^mytext/) $2="mysecondtext"; print $0}' ## if second column is 'mytext', replace with 'mysecondtext'
awk '{if ($2 ~ /^mytext/) mysecondtext mythirdtext;}' print $0 "myfourthtext"; else print $0' ## if 'mytext' or 'mysecondtext' or 'mythirdtext' is found in beginn
awk '{if ($NF ~ /^mytext/) $NF="mysecondtext"; print $0}' ## if end column is 'mytext', replace with 'mysecondtext'
awk '{gsub("mytext", "mysecondtext", $2); print $0}' ## if 'mytext' is anywhere in second column, replace with 'mysecondtext' ($NF if 'mytext' is in end col
awk '{gsub("[a-zA-Z0-9]*[aA])>", "mysecondtext"); print}' ## replace words/columns ending in character 'a' or 'A' with 'mysecondtext'
sed 's0 ~ mytext/[n+1]{if (n==2){sub("mytext", "mysecondtext", $0); print}}' ## replace only the second instance of 'mytext' in the whole file with 'mysecond
sed -f /cygdrive/c/muppix/myreplacelist.txt ## replace or delete or insert mytext or mysecondtext (many texts) using a list of multiple/duplicate texts
awk '{gsub(/,/,"mytext", $2); print $0}' ## replace comma ',' (mychar) with 'mytext' in second column 2
tr -c [:alnum:] ' ' tr '\n'          ## replace punctuation characters with spaces, then replaces spaces with newlines, split text to a long list of words/
awk '{gsub("mytext", "mysecondtext", $2); print}' ## replace 'mytext' anywhere inside the second column with 'mysecondtext'
awk -v v="mydelimitier" 'BEGIN {FS=OFS=v} {gsub("mytext", "mysecondtext", $2); print $0}' ## replace 'mytext' anywhere inside the second column with my

```

insert lines / append text [begin end between before after mysecondtext blankline file]

```

sed 'i\n'                               ## insert blankline above beginning of all the lines
sed '/mytext/{x;p;x;}'                  ## insert a blankline above a line with 'mytext' on it
sed '/mytext/G'                        ## insert a blankline below lines with 'mytext' on the line
sed 'i mytext'                          ## insert 'mytext' above all the lines/above beginning of lines
awk 'S0 ~ mytext/{print "mysecondtext " S0}' ## if 'mytext' on line, insert word/column 'mysecondtext' at beginning of line
sed 'S a mytext'                       ## insert 'mytext' below end of all lines
sed 'S a\ '                             ## insert blankline below the end of all the lines
awk '{if (pre=="$2") {print "mytext" S0} else {if (pre=="") {print "mytext" S0} else {print S0}}; pre=$2}' ## insert 'mytext' at the beginning of all paragraphs
echo "mytext" | cat - myfile.txt        ## take the text results of some command, insert below the file '.txt', and then continue with other commands.
sed '/mytext/i mysecondtext'            ## if 'mytext' is found anywhere on line, insert 'mysecondtext' on line above
sed '/mytext/a mysecondtext'            ## if 'mytext' is found anywhere on line, insert 'mysecondtext' on line below
awk '{if ($0 ~ mytext) {printf("%s\n%s\n", "mysecondtext", S0); printf("%s\n%s\n", S0, "mythirdtext");} else {print S0}}' ## if 'mytext' is found, insert 'myseco
sed 's/mytext/n mytext/g'               ## insert newline before 'mytext', split the line before mytext so every mytext is at the beginning of the line
sed 's/mytext/mytext\n/g'               ## insert newline after 'mytext', split the line after mytext
awk '{if ($2 ~ mytext$mysecondtext$mythirdtext$) {print "myfourthtext" S0} else print S0}' ## if 'mytext' or 'mysecondtext' or 'mythirdtext' is found in end of
sed -e 'mytext/r myfile.txt' -e 'x;SG'   ## insert file '.txt' above a line with 'mytext' on it
sed 'mytext/r myfile.txt'               ## insert file '.txt' below a line with 'mytext' on it

```

insert text on the line [mytext before after column blankline]

```

sed 's/^/mytext /'                    ## insert 'mytext' / column before beginning of the line ie: sed 's/^/ /' #indent lines
sed 's/.*&mytext/'                    ## insert 'mytext' or column after the end of the line
sed 's/mytext/mysecondtextmytext/g'   ## insert 'mysecondtext' before 'mytext'
sed 's/mytext/mytextmysecondtext/g'   ## insert 'mysecondtext' after 'mytext'
awk '{print substr(S0,"mytext",1,2)}'   ## insert upto 2 (fixed) characters (ie spaces) after end of each line - pad out lines to be length 2
awk '{S2=$2 mytext; print S0}'         ## insert 'mytext' after second column. TIP: to insert a new column use 'mytext'
awk '{S2="mytext" S2; print S0}'       ## insert 'mytext' before second column TIP: to insert a new column use 'mytext'
awk '{if (match(S0,"mytext")) {print "mysecondtext" S0} else {print S0}}' ## insert 'mysecondtext'/column at beginning of line if line has 'mytext'
awk '{if (match(S0,"mytext")) {print S0 "mysecondtext";} else {print S0}}' ## insert 'mysecondtext'/column at end of line if line has 'mytext'
sed 's/mytext[AB]mysecondtext&/g'     ## insert 'mysecondtext' before 'mytextA' or 'mytextB (range)'
awk '{if ($2 ~ mytext) {$2="mysecondtext" S2; print S0} else print S0}' ## if 'mytext' is in second column, insert 'mysecondtext' before the second column
awk '{if ($2 ~ mytext) {$2=$2 mysecondtext; print S0} else print S0}' ## if 'mytext' is in second column, insert 'mysecondtext' after the second column
awk '{getline addf < myfile.txt" } {$2=$2 addf; print S0}' ## insert file '.txt' after second column TIP: if myfile has less lines, it will repeat the last line. (befo
sed -e 's/<[:alnum:]*[mytextmysecondtext]>/mythirdtext&/g' ## insert 'mythirdtext' before words/columns ending in 'mytext' or 'mysecondtext'
nl -ba                                  ## insert linenumbers at the beginning of each line ie: find out linenumbers with 'mytext' : cat .txt | nl -ba | fgrep 'mytext'
fgrep -n 'mytext'                      ## select lines with 'mytext' include linenumbers (usefull for large files & can delete section of lines, from fixed line

```

sort & rearrange order [sort second column delimiter split]

```

sort                                     ## sort lines
sort -f                                 ## sort, but ignore case, uppercase or lowercase
sort -n                                 ## sort by numbers ie: look at beginning column as numeric values and sort TIP: if there are punctuation characters, s
sort -r                                 ## sort in reverse order
sort -k2                                ## sort on the second column TIP: beware of multiple spaces between columns
sort -t":" -k2                          ## sort text by second column, ":" is mydelimitier
sort -rk2                                ## sort on second column but in reverse order
sort -k2,2n                             ## sort on second column of numbers
sort -u                                  ## sort lines and then delete duplicate lines
rev                                      ## reverse/rotate each character on the line, end char becomes begin characer
cut -d ' ' -f2                          ## select second column only using each space character ' ' as a column mydelimitier. split TIP: shld delete multiple sp
cut -c 2-                                ## select fixed text after the second character onwards, delete beginning 2 characters
awk '{print substr(S0,length(S0)-2,length(S0))}' ## select 2 (fixed) characters from the end of line, delete before the second from end character
cut -d '#' -f2 | cut -d '.' -f2         ## select all text after the 1st '#' mydelimitier character on the line, and then all text after the next '.' character split

```

convert /split / change structure of lines

```

tr ' ' '\n'                            ## replace spaces with newlines, convert/split text to a long list of words/products TIP: may need to replace punctuatio
tr '\n' ' '                             ## replace newlines with spaces, convert list into a long single line TIP: if windows, use \r (carriage return (13)) instead o
tr ', ' '\n'                            ## replace all commas / mydelimitier = ' ' with a newline ie: split all text with commas into a table of words/columns (str
awk 1 ORS=" "                            ## convert whole text into 1 single line. replace newline with space
awk '{temp = $1; $1 = $2; $2 = temp; print}' ## select second column and then beginning column, and then all the other columns (swap columns 1
sed 's/mytext/n/g'                       ## split up line everytime it finds 'mytext' ie: insert newline when it finds 'mytext' (structure)
pr -t2                                   ## convert single list (one column) into 2 columns (filling the 1st column going down, then second column etc)
tr '[:punct:]' ' ' | tr ' ' '\n'         ## convert text into single list of words
diff -w myfile mysecondfile              ## select differences in 2 files, but ignore differences of extra spaces or tabs (white space) TIP: "<" in the out

```

loop, repeat muppix commands [mycommand mysecondcommand]

```

mylist ; do mycommand ; mysecondcommand ; done ## loop trough a list of text ( mylist) do some Unix commands and repeat ie:
find -name \*.txt -print | while read f ; do echo "$f" ; u2d "$f" ; done ## take a list of .txt files (myextension), display the filename, convert to DOS, to be rea
while sleep 2; do mycommand ;done        ## run mycommand every 2 seconds. ie: select current date & time every 2 seconds: while sleep 2; do dat
sed 's/a;SAB[0-9]{3}\>/, & , /ta'       ## format numbers : insert commas to all numbers, changing '1234567' to '1,234,567' (GNU sed)
pdftotext -layout myfile.pdf            ## generates a new file .txt in the current directory. TIP: with cygwin need to include pdftotext package when

```

reading in websites as text ie: twitter [mywebsite]

```
w3m -dump 'www.mywebsite.com'          ## select 'www.mywebsite' as text ie: w3m -dump 'www.muppix.co' | fgrep 'mytext'
wget http://www.mywebsite.com/         ## download html of mywebsite, saved as a file called index.html,& also creates a directory 'www.mywebsi
w3m -dump 'https://duckduckgo.com/?q=mytext' ## search web for 'mytext' using duckduckgo search engine
w3m -dump 'https://duckduckgo.com/?q=mytext+mysecondtext' ## search web for 'mytext' aswell as 'mysecondtext'
```

save / append files [directory extension database insert]

```
TIP: dont ever cat a file or search a file and then save it with the same name again. ie: dont : cat myfile.txt | mycommand >myfile.txt !! ##### ##
>myfile.txt                          ## save results to .txt in this directory (TIP: pls note there is no "|" with this command ) ie: ls -al >myfile.txt
>>myfile.txt                          ## insert results below end of .txt and save (even if it doesnt exist yet) ie: grep mytext * >>myfile.txt
>myfile.dat                          ## save as text file for viewing in notepad *.dat
>/cygdrive/c:/muppix/myspreadsheet.csv ## save results to excell/spreadsheet or msaccess database in this directory. TIP: ensure the columns hav
cat myfile.txt >>mysecondfile.txt      ## insert all .txt lines at end/below mysecondfile.txt and mysecondfile.txt (even if mysecondfile doesnt exis
paste myfile mysecondfile | sed 's/ / /' ## insert/glue mysecondfile after each, insert some spaces in between
pr -tmd --sep-string="|" myfile mysecondfile ## insert mysecondfile after(to right of) . side by side as 2 columns with '|' as mydelimiter between fil
join <(cat myfile.txt|sed -e 's/^[ \t]*//|sort) <(cat mysecondfile.txt|sed -e 's/^[ \t]*//|sort) ## insert after columns from mysecondfile, based on the begin colum
join <(cat myfile.txt|sed -e 's/^[ \t]*//|sort) <(cat mysecondfile.txt|sed -e 's/^[ \t]*//|sort) -a1 ## insert after columns from mysecondfile, based on the begin col
dos2unix                               ## TIP: may need to run unix2dos or u2d , before looking at the file in Windows say notepad
unix2dos
```



The Muppix Team provides innovative solutions and **Training** to make sense of large scale data
Backed by years of industry experience, the Muppix Team have developed a **Free Unix Data Science
Toolkit** to extract and analyse multi-structured information from diverse data sources

[Company](#)

[Training](#)

[Professional Services](#)

[Get Started](#)

[Blog](#)