

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
du . | sort -n ## sort by size each directory (mydir) and subdirectory ie: you've run out of space & need to delete stuff (try doing that in Windows..)
```

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

```
find . -type f -exec ls -s {} \; | sort -n ## select all files in subdirectories, sorted by size
find . -iname "*"myfile*" -ls ## select filenames with 'myfile' in between/somewhere in the filename (include size/date & subdirectory/path information) in all
find . -type f -print | egrep '(.jpg|.gif|.png)' ## select jpg or gif or png (myextension) files in any subdirectory
find . -size +2k -size -2M -ls ## select all files in all subdirectories , size between 2K & 2Mb. Each k is 1024 bytes, include sizes, saved date, directory path
find . -mtime -2 -name "*" -ls ## files in subdirectories saved in last 2 days
find . -mmin +2 -mmin -10 ## files in subdirectories saved between last 2 minutes and 10 minutes ago
```

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

```
fgrep -rai 'mytext' * ## select lines with 'mytext' in all files in all subdirectories (entire hard drive / network), ignore case, include filename TIP: cant use wildcard
find . -type f -print0 | xargs -0 grep -ai 'mytext' ## select 'mytext' (ignore case) from all subdirectories & select the directories, filenames & the lines if 'mytext' a
find . -name 'myfile*mytext' -print0 | xargs -0 grep -a [alnum] ## all lines of filenames beginning with 'myfile' aswell as 'mytext' in the filename in all subdirs
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 grea
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
```

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

```
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 'mytext' | fgrep -i 'mysecondtext' | fgrep -i 'mythirdtext' ## select line with 'mytext' aswell as 'mysecondtext' aswell as 'mythirdtext' in any order (ignore case
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 you run dos2unix on
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'
awk 'SO ~/mytext.*mysecondtext/' ## select line where 'mytext' is before 'mysecondtext', 'mysecondtext' after 'mytext'
awk 'SO ~/mytext.*mytext/' ## select line where 'mytext' appears twice or more often - second occurrence
egrep 'bmytextw*b' ## select line with word/column beginning with 'mytext' ie: 'mytextualisation'
egrep 'b*w*mytext*b' ## select line with word/column ending in 'mytext'. ie: find words/columns ending in 'ion' 'ing'
```

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 lines
sed '/mytext/,</p>/d' ## select beginning lines above 'mytext', delete all lines below 'mytext'
sed -n '/mytext/,Sp' ## select lines below 'mytext' to end of file, including 'mytext'. delete beginning lines above 'mytext'
awk 'match($0,"mytext"),match($0,"mysecondtext")' ## select section of lines between / below 'mytext' and 'mysecondtext'
awk 'S2=="mytext",S2=="mysecondtext"' ## select section of lines between the beginning line with 'mytext' in second column to 'mysecondtext' in second colu
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 betw
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]

```
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 running: fgrep -i 'mytext'
fgrep -v 'mytext' ## delete line if 'mytext' is somewhere on the line TIP: dont ignore case & also first check which lines will be deleted by running: fgrep 'mytext'
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)
awk 'BEGIN {i=1} 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/)}' ## 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)
sort | uniq -d ## select only the duplicate lines, ie: those lines that occur twice or more
sed '/mytext/,</p>/d' ## delete all lines below 'mytext', select beginning lines above 'mytext'
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
```

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)
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
sed 's/./S/' ## delete the end 2 (fixed) characters on each line. (end & second from end character)
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
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]*$/[ \t]*$//g' ## delete leading/beginning space aswell as ending/trailing spaces on the line(left align, trim)
grep -o 'mytext.*mysecondtext' ## select text between 'mytext' and 'mysecondtext' on the line. delete before 'mytext' aswell as after 'mysecondtext', include mytext
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/[^\t]*mytext*[^ \t]*//g' ## delete words/columns anywhere on the line, with 'mytext' somewhere inside/between the word ie: will delete words such as 'allm
awk -v OFS=" " 'S1=$1' ## delete/replace all multiple/duplicate/consecutive spaces with single space/blank
sed 's/[^\a-z0-9A-Z]/ /g' ## 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/) print}' # select line if second column begins with 'mytext'
awk '{if($2 ~ /mytext$/) print}' # select line if second column ends with 'mytext'
awk '{print $(NF-1)}' # select only the second from end column, delete all other columns
awk '{if($2 !~ /mytext/)print}' # delete line if second column is 'mytext'
awk 'NF > 2' # select line with more than/greater 2 columns length (delete lines with begin and second columns) length
sed 's/^mytext/' # delete 'mytext' if it is at the beginning of the line
sed 's/mytext$/' # 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 for selecting mytext on liv
awk 'NR>=2' # select the second (fixed) lines & below, delete lines above second line
sed '2,88!d' # 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 ie: 1st 2 lines after wegsite
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)
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> 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< 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 & no
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 $0}' # if second column has a number/value : is greater than 2.0, select whole line. TIP: '> 0' is the same as selecting the whole line
awk '{($NF+0) >= 2.0}' # select line if end column has a number/value : is greater or equals than 2.0
egrep '\b[0-9]{2,}' # lines have 2 or numbers with length of 2 or containing more/greater numbers (second), somewhere on the line
grep '[0-9]{2,}' # lines with atleast 2 consecutive numbers/digits, or more (length)
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
sed 's/[0-9]/g' # delete all numbers/digits
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/gi' # replace every 'mytext' with 'mysecondtext', ignore case of 'mytext'
sed 's/mytext/Mysecondtext' # if 'mytext' is found anywhere on line, replace whole line with 'mysecondtext'
sed 's/(.*)mytext/\1mysecondtext/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 -e 's/mytext.*mysecondtext/' # replace everything after 'mytext' with 'mysecondtext'. replacing mytext and everything after mytext
sed 's/^s/mytext/g' # replace blanklines with 'mytext'. insert 'mytext' TIP:may need to ensure is truly blankline
awk '{if ($1 ~ /^mytext/) $1="mysecondtext";print $0}' # if begin column is 'mytext', replace with 'mysecondtext'
awk '{if ($2 ~ /^mytext/) $2="mysecondtext";print $0}' # if second column is 'mytext', replace with 'mysecondtext'
awk '{if($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' (SNF if mytext is in end column)
awk '{gsub("[<a-zA-Z0-9]*[aA]>","mysecondtext");print}' # replace words/columns ending in character 'a' or 'A' with 'mysecondtext'
awk 's0 ~ /mytext/{n+=1; if (n==2){sub("mytext","mysecondtext",$0);print}' # replace only the second instance of 'mytext' in the whole file with 'mysecondt
awk '{gsub(/,,"mytext",$2);print $0}' # replace comma ',' (mychar) with 'mytext' in second column 2
tr -c [alnum:] '\t' tr '\n' # replace punctuation characters with spaces, then replaces spaces with newlines, split text to a long list of words/products

```

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 " $0}' # 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
sed 's/mytext/\nmytext/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

```

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 '{ $2=$2 mytext';print $0}' # insert 'mytext' after second column. TIP: to insert a new column use ' mytext '
awk '{ $2="mytext" $2;print $0}' # insert 'mytext' before second column TIP: to insert a new column use 'mytext '
awk '{if ($2 ~ /mytext/){ $2="mysecondtext" $2;print $0}else print $0}' # if 'mytext' is in second column, insert 'mysecondtext' before the second column
awk '{if ($2 ~ /mytext/){ $2=$2 mysecondtext';print $0}else print $0}' # if 'mytext' is in second column, insert 'mysecondtext' after the second column
nl -ba # insert linenumbers at the beginning of each line ie: find out linenumbers with 'mytext': cat myfile.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 linenumbers)

```

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, sort may not work & delet
sort -u # sort lines and then delete duplicate lines

```

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 punctuation with spaces first
tr '\n' ' ' # replace newlines with spaces, convert list into a long single line TIP: if windows, us '\r' (carriage return (13)) instead of '\n' (10)
tr ', '\n' # replace all commas / mydelimiter = ',' with a newline ie: split all text with commas into a table of words/columns (structure)

```

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.mywebsite.com', can be loa
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]

```
>myfile.txt ## save results to myfile.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 myfile.txt and save (even if it doesnt exist yet) ie: grep mytext * >>myfile.txt
>/cygdrive/c:/muppix/myspreadsheet.csv ## save results to excell/spreadsheet or msaccess database in this directory. TIP: ensure the columns have a delimiter s
unix2dos ## TIP: may need to run unix2dos or u2d , before looking at the file in Windows say notepad
```

webversion of Muppix toolkit (450 commands listed, out of 1480)



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

[Blog](#)

Training

Professional Services

Get Started