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}'                  ## 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 '{$2=$3=$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}'                    ## delete line if second column is 'mytext'
sed -e 's/\<[[:alnum:]]*[A|a]\>/&/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 '($0 ~/mytext/) || (($1 ~/mysecondtext/) && ($2 ~/mythirdtext/)) {print $0}' ## select if ( the whole line contains the word 'mytext' ) or ( the beginning colu
cut -d ',' -f2-                                   ## select second column (using mydelimiter ',') & all columns after 2, (split lines)
sed 's/^mytext//'                                 ## delete 'mytext' if it is at the beginning of the line
sed 's/mytext$//'                                 ## 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 '2,88!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 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 selec
awk '{if(($1+0) < 2.0) print $2,$3,$1 }'          ## begin column has a number/value : is smaller than 2.0, select second,third and third column
awk '($NF+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 or 010, selec
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 '/mytext/c\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 's/mytext/£/g' |rev|sed 's/£/#/1' |rev|sed 's/£/mytext/g' | sed 's/#/mysecondtext/1' ## replace end occurrence of 'mytext' with mysecondtext TIP:ensure chars 'Â£'
awk '/mythirdtext/{gsub(/mytext/,"mysecondtext")};1'      ## replace 'mytext' with 'mysecondtext' only on lines containing 'mythirdtext'
awk '!/mythirdtext/{gsub(/mytext/,"mysecondtext")};1'     ## replace 'mytext' with 'mysecondtext' only on those lines NOT containing 'mythirdtext'
sed 's/mytext\(.*\)mysecondtext/mytext\1mythirdtext/'     ## 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 '$ 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/^$/mytext/g'                              ## replace blanklines with 'mytext'. insert 'mytext' TIP:may need to ensure is truly blankline
sed 's/^.*[0-9]mytext[AB]/ /g'                   ## delete/replace words beginning or ending with a range fixed number/text. ie: 8mytextA or 3mytextB anywhe
awk '{gsub("mytext\\w+ *","");print}'            ## delete/replace all words beginning with mytext
awk '/mytext$/ {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]*[a|A]\\>", "mysecondtext" );print}' ## replace words/columns ending in character 'a' or 'A' with 'mysecondtext'
awk '$0 ~/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="mydelimiter" '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 '1i\\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 '1i\mytext'                                  ## insert 'mytext' above all the lines/above beginning of lines
awk '$0 ~/mytext/{print "mysecondtext " $0}'     ## if 'mytext' on line, insert word/column 'mysecondtext ' at beginning of line
sed '$ a\mytext'                                 ## insert 'mytext' below end of all lines
sed '$ a\'                                       ## insert blankline below the end of all the lines
awk '{if (""==$2) {print "mytext" $0} else {if (pre==""){print "mytext" $0} else {print $0}} ; 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", $0);printf("%s\n%s\n",$0,"mythirdtext")} else {print $0}}' ## if 'mytext' is found, insert 'myseco
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
awk '{if ($2 ~/mytext$|mysecondtext$|mythirdtext$/){print "myfourthtext" $0} else print $0}' ## if 'mytext' or 'mysecondtext' or 'mythirdtext' is found in end of
sed -e '/mytext/r myfile.txt' -e 'x;$G'          ## 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($0 "mytext",1,2)}'            ## insert upto 2 (fixed) characters (ie spaces) after end of each line - pad out lines to be length 2
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(match($0,"mytext")){print "mysecondtext" $0} else {print $0}}' ## insert mysecondtext/column at beginning of line if line has 'mytext'
awk '{if(match($0,"mytext")){print $0 "mysecondtext"} else {print $0}}' ## 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" $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
awk '{getline addf <"myfile.txt"} {$2=$2 addf;print $0}'  ## insert file '.txt' after second column TIP: if myfile has less lines, it will repeat the last line. (befo
sed -e 's/\<[[:alnum:]]*[mytext|mysecondtext]\>/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 mydelimiter
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 mydelimiter. split TIP: shld delete multiple sp
cut -c 2-                                        ## select fixed text after the second character onwards, delete beginning 2 characters
awk '{print substr($0,length($0)-2,length($0))}' ## 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 '#' mydelimiter 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, us \r (carriage return (13)) instead o
tr ',' '\n'                                      ## replace all commas / mydelimiter = ',' 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 ':a;s/\B[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 -tmJ --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
join -f '|' <(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 begi
unix2dos                                         ## TIP: may need to run unix2dos or u2d , before looking at the file in Windows say notepad
```