

SED EXAMPLES

```
# delete leading whitespace (spaces, tabs) from front of each line. aligns all text flush left
sed 's/^[ \t]*//' # type tab key, not \t

# delete trailing whitespace (spaces, tabs) from end of each line
sed 's/[ \t]*$//' # type tab key, not \t

# delete BOTH leading and trailing whitespace from each line
sed 's/^[ \t]*//;s/[ \t]*$//' # type tab key, not \t

# insert 5 blank spaces at beginning of each line (make page offset)
sed 's/^/     /'

# align all text flush right on a 79-column width
sed -e :a -e 's/^\.{1,78}$/ &;ta' # set at 78 plus 1 space

# center all text in the middle of 79-column width.
# In method 1, spaces at the beginning of the line are significant, and trailing spaces are appended at the end
# In method 2, spaces at the beginning of the line are discarded in centering the line, and no trailing spaces appear
# at the end of lines.
sed -e :a -e 's/^\.{1,77}$/ & /;ta' # method 1
sed -e :a -e 's/^\.{1,77}$/ &;ta' -e 's/( *\)\1/\1/' # method 2

# substitute (find and replace) "foo" with "bar" on each line
sed 's/foo/bar/' # replaces only 1st instance in a line
sed 's/foo/bar/4' # replaces only 4th instance in a line
sed 's/foo/bar/g' # replaces ALL instances in a line
sed 's/(.*\)foo\(. *foo\)\/\1bar\/2/' # replace the next-to-last case
sed 's/(.*\)foo\/\1bar/' # replace only the last case

# substitute "foo" with "bar" ONLY for lines which contain "baz"
sed '/baz/s/foo/bar/g'

# substitute "foo" with "bar" EXCEPT for lines which contain "baz"
sed '/baz/!s/foo/bar/g'

# change "scarlet" or "ruby" or "puce" to "red"
sed 's/scarlet/red/g;s/ruby/red/g;s/puce/red/g'

# reverse order of lines (emulates "tac")
sed '1!G;h;$!d' # method 1
sed -n '1!G;h;$p' # method 2

# reverse each character on the line (emulates "rev")
sed '/\n/!G;s/(. \(.*\n\))/&\2\1/;/D;s/.//'

# join pairs of lines side-by-side (like "paste")
sed '$!N;s/\n/ /'

# if a line ends with a backslash, append the next line to it
sed -e :a -e '/\$$/N; s/\\n//; ta'

# if a line begins with an equal sign, append it to the previous line and replace the "=" with a single space
sed -e :a -e '$!N;s/\n=/ /;ta' -e 'P;D'

# add commas to numeric strings, changing "1234567" to "1,234,567"
sed -e :a -e 's/(.*[0-9])\([0-9]\{3\})\/\1,\2/;ta'

# print only lines which match regular expression (emulates "grep")
sed -n '/regexp/p' # method 1
sed '/regexp/!d' # method 2

# print only lines which do NOT match regexp (emulates "grep -v")
sed -n '/regexp/!p' # method 1, corresponds to above
sed '/regexp/d' # method 2, simpler syntax
```

```

# print the line immediately before a regexp, but not the line containing the regexp
sed -n '/regexp/{g;1!p;};h'

# print the line immediately after a regexp, but not the line containing the regexp
sed -n '/regexp/{n;p;}'

# grep for AAA and BBB and CCC (in any order)
sed '/AAA/!d; /BBB/!d; /CCC/!d'

# grep for AAA and BBB and CCC (in that order)
sed '/AAA.*BBB.*CCC/!d'

# grep for AAA or BBB or CCC (emulates "egrep")
sed -e '/AAA/b' -e '/BBB/b' -e '/CCC/b' -e d

# print section of file between two regular expressions (inclusive)
sed -n '/Iowa/,/Montana/p' # case sensitive

# print all of file EXCEPT section between 2 regular expressions
sed '/Iowa/,/Montana/d'

# delete duplicate, consecutive lines from a file (emulates "uniq").
# First line in a set of duplicate lines is kept, rest are deleted.
sed '$!N; /\^(.*\)\n\1$/!P; D'

# delete lines matching pattern
sed '/pattern/d'

# delete ALL blank lines from a file (same as "grep '! '")
sed '/^$/d' # method 1
sed '/./!d' # method 2

# remove most HTML tags (accommodates multiple-line tags)
sed -e :a -e 's/<[^>]*>/g;/</N;/ba'

```