tr [OPTION]... SET1 [SET2]

echo abc | tr abc xyz

echo abcd | tr abcd xyz

echo abcde | tr abcd xyz

echo abcde | tr[:lower:] [:upper:] translate lower case to upper case

tr -c "complement" use the complement of SET1

echo a1b2c3 | tr -dc [:lower:]

echo a1b2c3EF | tr -dc [:lower:]

tr -d abc "delete" delete characters in SET1, do not translate

echo a1b2c3 | tr -d [:lower:]

remark: newline is not a lower case character

tr -s "squeez and repeat" replace each input sequence of a repeated character that is listed in SET1 with a single occurrence of that character

echo aaabbca | tr abc xyz

echo "A test line" | tr -s [:space:] "\t"

echo "A test line" | tr -s [:space:] 1

remark: newline is in the "space" class

sed

sed s/thingyouwanttoreplace/thingyouwanttouse/g <fileyouRead> fileYouSaveOutput

s--substitute g--all characters, not a line

echo day | sed s/day/night/g

sed 1p hello "print the first line"

--read the first line, put it into a buffer, run the command in the buffer, then output the content in the buffer

sed 2d Hello "delete"

--put the first line into buffer, check, output buffer; put second line into buffer,check and delete it, output buffer(nothing), ...

sed -n 1p Hello "silent processing:does not print what's in the buffer"

--only print the first line

sed -n -e s/Line/Blah/g -e 2p Hello

--running two seperaste command, first after -e, second after -e for each line

12,18 from 12 to 18

1~3 jump to 3, and keep going

'/pattern/d' --only remove certain pattern --''is used to make clear

/BEGIN/, /END/ -- all contents btw the two words

!d, reverse of d, --only keep

remark: if you cannot run a command, add ''

when you use p, it prints the lien

test "check files"

test 4 -gt 3

echo $?

test 4 -lt 3

echo $?

true=0 false=1 error>1

echo $a $name--variable that name a

echo $?

expr "evaluate"

a=2

echo $a

b=$(expr $a + 3 )

echo $b

Regular expression

Alternation: [A-Za-z]

|-or Hello|World

Quantification: ? \* +

Anchors: ^(beginning) $(end)

\^-"^"

<>this will not match anything

<.+>this will

a\{2,3\} this will match aa and aaa

grep 'pattern'

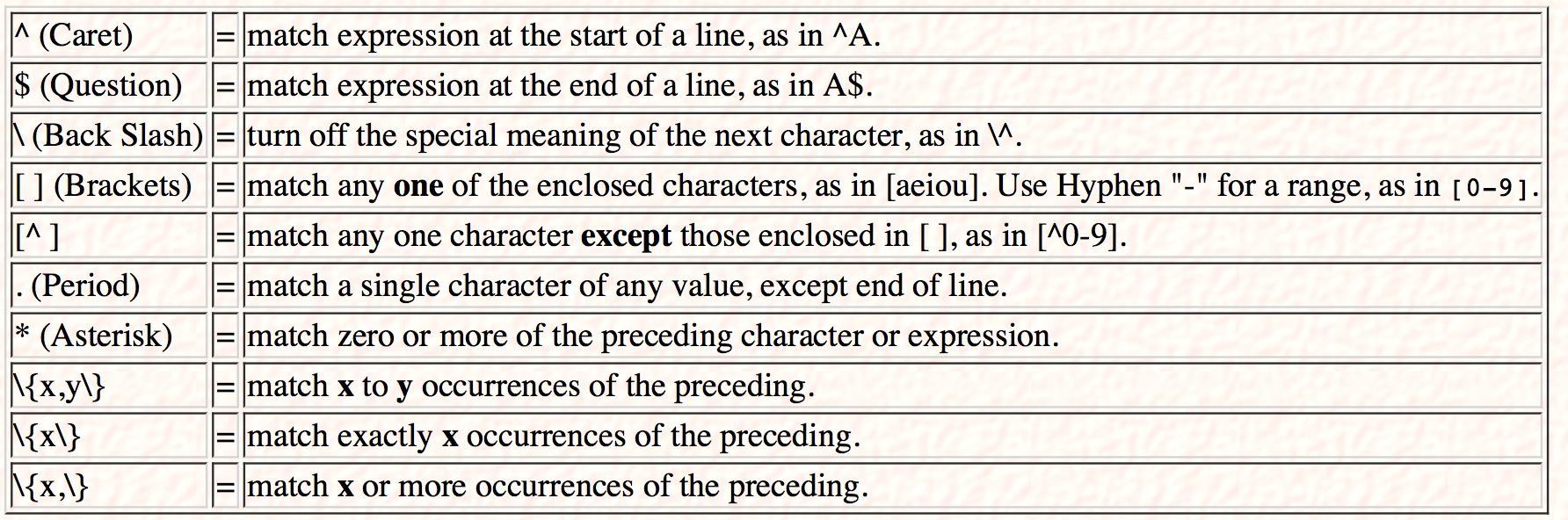
'^pattern'

'^pattern$'

'[Bb]ob'

'^$'

'[0-9][0-9]'



quoting

single quote

echo '$a'

double quote

same as ', except $, `, \

echo `ls` --evaluate and print result

echo "$a" --1

backquote --execute the command

shell scripting

Bourne sh

c bash

ksh

eg:

ls /bin/sh

/bin/sh --open the shell, and run the command

vi tmpScript

#!/bin/sh

echo Hello

-back

/bin/sh tmpScript

chmod +x tmpScript

./tmpScript

,