

# Regular expressions

## SED

Character	Description
^	Matches the beginning of the line
\$	Matches the end of the line
.	Matches any single character
*	Will match zero or more occurrences of the previous character
[ ]	Matches all the characters inside the [ ]

Regular expression	Description
/./	Will match any line that contains at least one character
/../	Will match any line that contains at least two characters
/^#/	Will match any line that begins with a '#'
/^\$/	Will match all blank lines
/}\$	Will match any lines that ends with '}' (no spaces)
/} */	Will match any line ending with '}' followed by zero or more spaces
/[abc]/	Will match any line that contains a lowercase 'a', 'b', or 'c'
/^[abc]/	Will match any line that begins with an 'a', 'b', or 'c'

## Sed examples

```
sed -i 's/Ben/Dave/g' file.txt # Replace all the words Ben for the word Dave
sed 's/Ben|ben/Dave/g' file.txt # Replace all the words Ben and ben for the word Dave
sed 's/^[^t]*//' file.txt # Delete all spaces in front of every line of file.txt
sed 's/[^t]*$//' file.txt # Delete all spaces at the end of every line of file.txt
sed -e '/^#/d' file.txt | more # View file without the commented lines
sed -e '/regexp/d' file.txt # delete the word regexp
sed 's/.../' # delete the first 3 characters on every line
```

# AWK

```
awk '!($0 in a){a[$0];print}' # Remove duplicate, nonconsecutive lines
awk '{ print $NF }' # print the last field of each line
awk -F':' '{print $3,$4;}' # show only what is on columns 3 and 4
```

## Find and replace

```
awk '{gsub(/foo/,"bar")}; 1' # if foo replace by bar
awk '/baz/{gsub(/foo/, "bar")}; 1' # ONLY for lines which contain "baz"
awk '!/baz/{gsub(/foo/, "bar")}; 1' # EXCEPT for lines which contain "baz"
```

## Grep

```
grep 'word\|logs' file # can contain 2 strings
grep "word1" file | grep "word2" # line must match the 2 strings
```

## xargs examples

```
locate file* | xargs grep "bob" # find a file and grep a string
locate file* | xargs rm # find a file a del it
```

## CUT example

```
cut -d " " -f 1 - cut everything after the first word
```

## For loop example

```
for i in {a..h}; do smartctl -i -A /dev/sd$i | grep
"Current_Pending_Sector\|Media_Wearout_Indicator\|Power_On_Hours\|Reallocated_Sector_Ct\|UDMA_
CRC_Error_Count"; done
```

```
for string in $(cat ips.txt); do ip route add blackhole $string; done
```

```
for i in `cat list.txt` ; do echo $i ; curl --user `cat user-pass.txt` -s -i -k -b
"PHPSESSID=XXXXX; JSESSIONID=XXXXXX" "https://domain.com$i" | grep -i "WORD" ; sleep 2 ; done
```

Command above will grep a web page for "domain.com/list.txt" (whatever is in list.txt), --user is for a httpasswd, PHPSESSID and JSESSIONID is used after a user is logged in, the ID can be found on chrome "inspect element >> network" (DO NOT REFRESH OR CLOSE PAGE IN CHROME OR SESSION WILL EXPIRE)

---

Revision #11

Created 26 August 2017 22:42:38 by Dave

Updated 16 December 2017 18:25:46 by Dave