SSH client and SCP client
Prompt ($ versus #)
$HOME versus home
ls (list)
    -a
    -i
    -l
    patterns: * or *.* or ?
pwd (path to working directory
man (manual)
. (current directory) versus .. (parent directory)
absolute path (it starts by /) versus relative path (it doesn't start by /)
cd (change directory)
mkdir (make a directory)
rmdir (remove a directory)
cat (concatenate)
cp (copy)
mv (move)
rm (remove)
echo
clear
ps (process status)
bash (to create a new process)
exit (exit)

Predefined variables such as HOME, PATH or $?, $#, $@, $0, $i (for shell scripts).
env (environment)
Defining variables considering "", '', `` or conditional or reading (read variable)
Export
Defining complex commands: ;, &, || and &&
Redirections
    > output
    >> output adding
    < input
    2> error
Pipe |
shift (for shell scripts)
wc (word count)
    -w
    -l
    -c

i-node concept. Each i-node has an id --> ls -i. It allows to reconstruct the file system.
ln (hard link. Physically same file (same i-node). Increase the number of hard links (second column when ls -l).
ln -s (soft link. New file containing an alternative path (new i-node). first character when ls -l --> l)
Using ls -l --> attributes
chmod (change modus --> change permissions)
    with character
    octal
chown (change owner. Only root)
chgrp (change group. Only root)
umask (666-mask=new permissions)
gunzip
tar

more and les (paging)
sort
    -t (delimiter or separator)
    -k (field)
    -o (output)
Cut

-d (delimiter or separator)
-f (field)
-b (byte)
-c (column)

Paste

-d (delimiter)

Join (files must be sorted)

-t (delimiter)
-n m (number of file (n) and the corresponding key field (m))

REGEX

- . whatever char.
- * 0 or more repetitions of previous char
- ^ starts by next …
- $ ends by previous …
- \ to literality
- [] options
- [^] no options

Grep pattern file

Find where? What(criteria)? What(to do)?

{} result of find
\; end of find command