

# LinuxDays



[openclipart.org](http://openclipart.org)

## The Linux Toolkit

# Goals of this course

- Get acclimatized to the console
- Learn how to navigate in your folders
- Basics of running commands
- Simple file and folder manipulations
- Managing software
- Dealing with storage devices
- ...getting you ready for the next course! :-)



Slides

# Why the console?

---

- because every semester, people asked us for more of it
- because it's the cleanest way to work on your router / NAS
- because it's the last thing alive when you break your system
- because that's the way to go for advanced tasks

# Look, a console!

## The default console under OpenSUSE:

The user you are currently logged in as

name of the machine

current location (~ means "home folder")

```
sandro@linux-pmcq: ~>
```

The interpreter (here: bash) is ready for your input.

Put your favourite background here

# Other console styles

## Bash under Arch Linux:

The user you are currently logged in as

name of the machine

current location (~ means "home folder")

```
[sandro@myMachine ~]$
```

The interpreter (here: bash) is ready for your input, and you are an unprivileged user.

Put your favourite background here

# Other console styles

**Fish is another interpreter, designed for beginner-friendliness:**

```
Welcome to fish, the friendly interactive shell  
Type help for instructions on how to use fish  
sandro@myMachine ~>
```

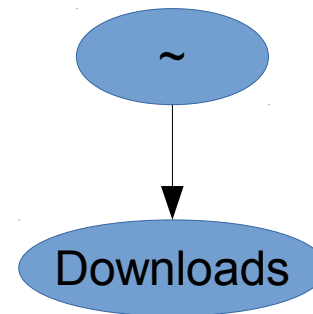
# Navigating in the file hierarchy

## cd: change directory

```
sandro@linux-pmcq:~> cd Downloads  
sandro@linux-pmcq:~/Downloads>
```



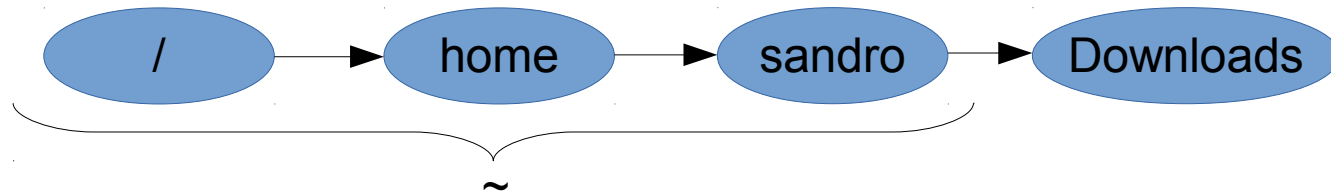
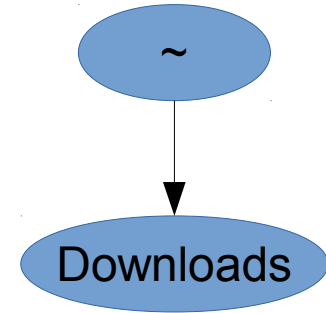
current location has changed!



# Navigating in the file hierarchy

## pwd: Where am I?

```
sandro@linux-pmcq:~> cd Downloads  
sandro@linux-pmcq:~/Downloads> pwd  
/home/sandro/Downloads  
sandro@linux-pmcq:~/Downloads>
```

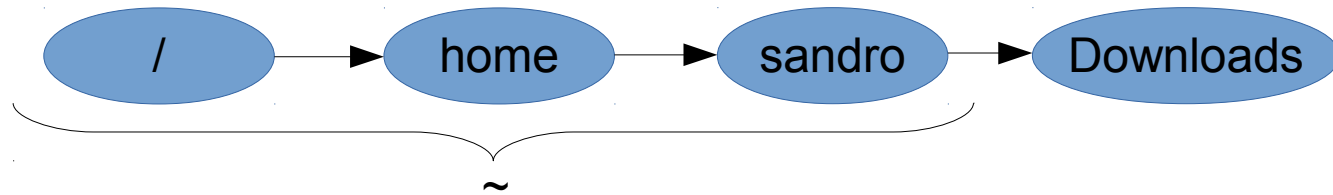
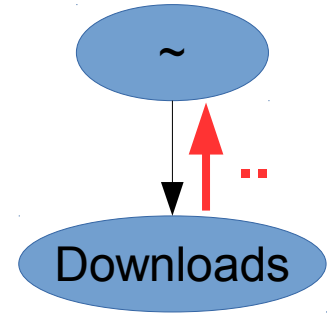




# Navigating in the file hierarchy

## .. : the directory above

```
sandro@linux-pmcq:~> cd Downloads
sandro@linux-pmcq:~/Downloads> pwd
/home/sandro/Downloads
sandro@linux-pmcq:~/Downloads> cd ..
sandro@linux-pmcq:~> pwd
/home/sandro
sandro@linux-pmcq:~>
```



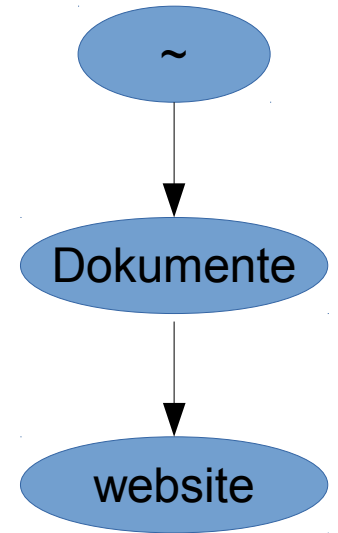
# Navigating in the file hierarchy

## Going down two folders at once...

```
sandro@linux-pmcq:~> cd Dokumente/website/  
sandro@linux-pmcq:~/Dokumente/website>
```

## ...and up again

```
sandro@linux-pmcq:~/Dokumente/website> cd ../..  
sandro@linux-pmcq:~>
```



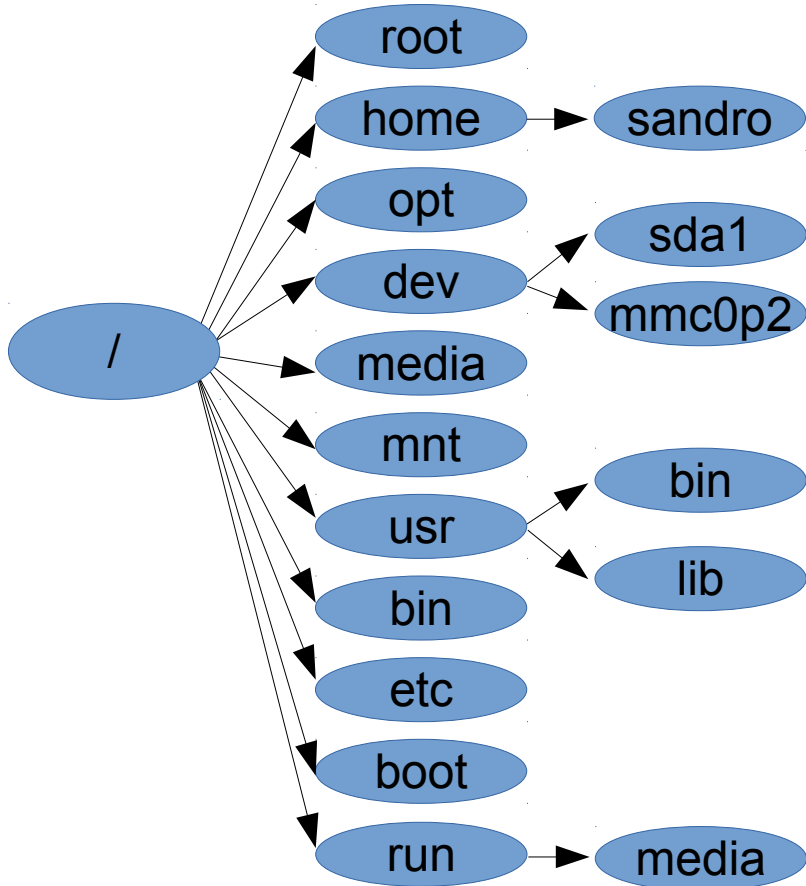
# Navigating in the file hierarchy

## Bring me home

```
sandro@linux-pmcq:~/Dokumente/website> cd  
sandro@linux-pmcq:~>
```

# Navigating in the file hierarchy

## A typical Linux file hierarchy



root:	home folder of the superuser
home:	home folder of all regular users
opt:	manually installed software
dev:	devices on this computer
sda1:	1 <sup>st</sup> hard disk ("a"), 1 <sup>st</sup> partition ("1")
mmc0p2:	1 <sup>st</sup> SD card, 2 <sup>nd</sup> partition
media:	mounted (active) storage, visible
mnt:	as above, but hidden by file manager
usr:	user software packages
bin:	basic system software
etc:	system-wide configuration files
boot:	files needed for the system start
run/media:	some file managers place USB sticks there

# Navigating in the file hierarchy

## ls: look around

```
sandro@linux-gttg:~> ls
bin                Documents          Pictures            Templates
Desktop           Downloads          Public              Videos
document 1.odt     Music              public_html
sandro@linux-gttg:~>
```

# Using commands with arguments

**If cd gets an argument, it will interpret it as a file path and go there. Else, it goes to the home folder of the user that runs it.**

command to run

argument given to that command

```
sandro@linux-pmcq:~> cd Downloads  
sandro@linux-pmcq:~/Downloads>
```

# Using commands with arguments

## ls called with argument(s):

relative path, meaning: “within the current directory”

absolute path (starts with “/”)

```
sandro@linux-gttg:~> ls Documents /home/sandro
```

```
Documents:
miau.txt
```

```
/home/sandro:
```

```
bin           Documents    Pictures     Templates
Desktop       Downloads    Public       Videos
document 1.odt  Music       public_html
```

```
sandro@linux-gttg:~>
```

# Using commands with options

**Options are special arguments altering the behavior of a command.**

Typically, single-letter options are preceded by “-”,  
multi-letter options are preceded by “--”

```
sandro@linux-gttg:~/Documents> ls
miau.txt
sandro@linux-gttg:~/Documents> ls -a
.  ..  .hidden_file  miau.txt
sandro@linux-gttg:~/Documents> ls --all
.  ..  .hidden_file  miau.txt
sandro@linux-gttg:~/Documents>
```

## Notes:

- In Linux, files / directories starting with a “.” are hidden.
- The --all (or -a) option makes ls show hidden files too, as well as two directories:
  - “.” is the current directory (here: Documents)
  - “..” is the parent folder (here: ~)



# Using commands with options

**ls -l** : Show as a list, with sizes

**ls -h**: Show numbers in a more human-readable format

**Concatenation (“-lh”) not possible with multi-letter options (--all, ...)**

```
sandro@linux-gttg:~/Documents> ls -l
total 212088
-rw-r--r-- 1 sandro users 217178112 Jan 21 17:10 miau.txt
sandro@linux-gttg:~/Documents> ls -lh
total 208M
-rw-r--r-- 1 sandro users 208M Jan 21 17:10 miau.txt
sandro@linux-gttg:~/Documents>
```

permissions

owner / group

size

last modified

# Using arguments with spaces

**You can use spaces in file names etc.**

**How to tell them apart from spaces between arguments?**

**=> Escape spaces: put a backslash (“\”) in front of each space**

**=> Or use quotes:**

```
sandro@linux-gttg:~> ls document\ 1.odt
document 1.odt
sandro@linux-gttg:~> ls "document 1.odt"
document 1.odt
sandro@linux-gttg:~>
```

# Get help (“RTFM”)

**man [command]**

**man stands for “manual”**

```
sandro@linux-gttg:~> man ls
Man: find all matching manual pages (set MAN_POSIXLY_CORRECT to
avoid this)
* ls (1)
  ls (1p)
Man: What manual page do you want?
Man:
```



just hit Enter

# Get help

[ ] means that you can omit this (ls: can omit everything and just go “ls”)

```
LS(1)                                User Commands                                LS(1)

NAME
  ls - list directory contents

SYNOPSIS
  ls [OPTION]... [FILE]...

DESCRIPTION
  List information about the FILES (the current directory by default).
  Sort entries alphabetically if none of -cftuvSUX nor --sort is speci-
  fied.

  Mandatory arguments to long options are mandatory for short options
  too.

  -a, --all
        do not ignore entries starting with .

  -A, --almost-all
        do not list implied . and ..

Manual page ls(1) line 1 (press h for help or q to quit)
```

- q to quit
- / to search
- Arrows to scroll
- PgUp / PgDown to scroll fast
- Mouse wheel scrolls, too!

# Get help

```
do not list implied entries matching shell PATTERN

-k, --kibibytes
    default to 1024-byte blocks for disk usage

-l      use a long listing format

-L, --dereference
    when showing file information for a symbolic link, show information for the file the link references rather than for the link itself

-m      fill width with a comma separated list of entries

-n, --numeric-uid-gid
    like -l, but list numeric user and group IDs

-N, --literal
    print raw entry names (don't treat e.g. control characters specially)

-o      like -l, but do not list group information
```

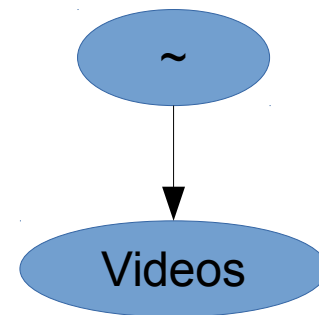
**/list**

To search on for “list”, just type “/” and hit Enter, the last item searched will automatically be searched again.

# Be efficient!

## Tab completion

```
sandro@linux-gttg:~> cd V<tab!>
```

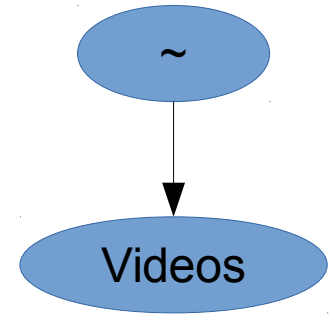


# Be efficient!

## Tab completion

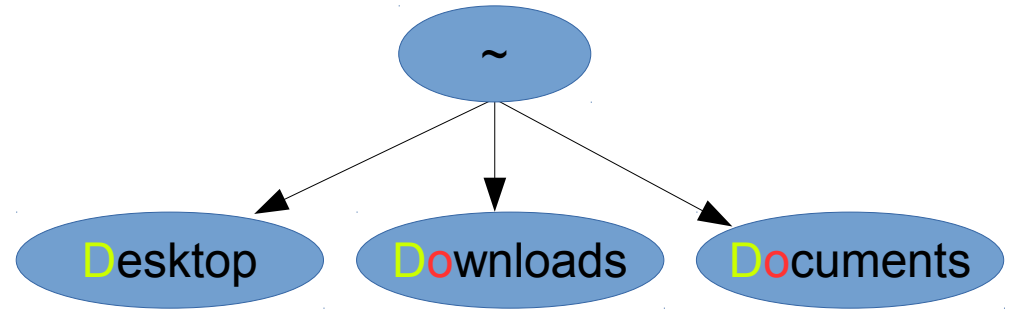
optional "/" to indicate  
a directory

```
sandro@linux-gttg:~> cd Videos/
```



# Be efficient!

## Tab completion

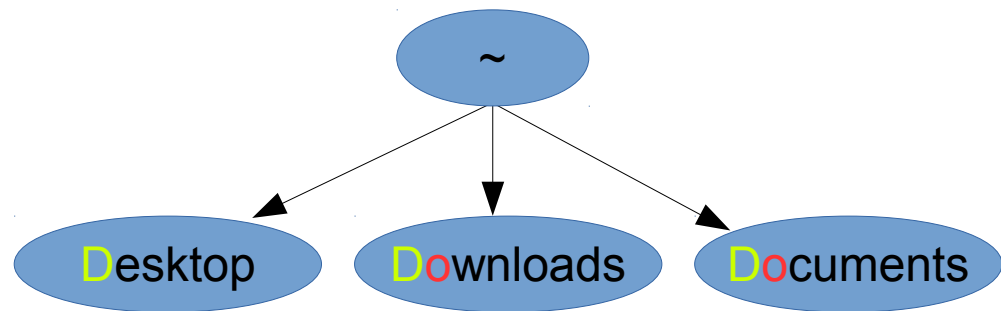


```
sandro@linux-gttg:~> cd D<tab!><tab!>  
Desktop/  Documents/  Downloads/  
sandro@linux-gttg:~> cd Do<tab!><tab!>  
Documents/  Downloads/  
sandro@linux-gttg:~> cd Dow<tab!>
```



# Be efficient!

## Tab completion



```
sandro@linux-gttg:~> cd D
Desktop/  Documents/ Downloads/
sandro@linux-gttg:~> cd Do
Documents/ Downloads/
sandro@linux-gttg:~> cd Downloads/
```

# Be efficient!

**Do a command again: Arrow keys**

```
sandro@linux-gttg:~> <Arrow key up!>
```

# Be efficient!

**Do a command again: Arrow keys**

```
sandro@linux-gttg:~> cd Downloads/
```

# Be efficient!

---

## Use your navigation keys

Your cursor behaves like in a text document:

- Use Home and End to jump to the beginning / end of your line
- Often, Ctrl+Arrows will jump one word

# Be efficient!


## Search command history: Ctrl+R

```
sandro@linux-gttg:~> ^R
```

# Be efficient!

## Search command history: Ctrl+R

```
(reverse-i-search) `cd': cd Dokumente/website/
```



Autocompleted:  
This is what we did  
with “cd[...]” recently

# Be efficient!

## Forgot sudo? sudo !!

```
sandro@linux-gttg:~> cat miao.txt  
cat: miao.txt: Permission denied  
sandro@linux-gttg:~> sudo !!  
sudo cat miao.txt  
meeow :-)  
sandro@linux-gttg:~>
```



Rerun last command  
with sudo

# Be efficient!

**Using wildcards: \* will be replaced by anything**

```
sandro@linux-gttg:~/stuff> ls
analysis-slides.txt  docu-physics.txt
docu-analysis.txt   my analysis summary.odt
docu-bio.txt        slides of analysis.pdf
linalg.txt
sandro@linux-gttg:~/stuff>
```

**We want to delete all files starting with “docu-”.**



# Be efficient!

**Using wildcards: \* will be replaced by anything  
We want to delete all files starting with “docu-”.**

option “v” for “verbose”, meaning: tell me more

```
sandro@linux-gttg:~/stuff> rm -v docu-*  
removed `docu-analysis.txt`  
removed `docu-bio.txt`  
removed `docu-physics.txt`  
sandro@linux-gttg:~/stuff>
```

# Be efficient!

**Using wildcards: \* will be replaced by anything**

```
sandro@linux-gttg:~/stuff> ls
analysis-slides.txt  my analysis summary.odt
linalg.txt          slides of analysis.pdf
sandro@linux-gttg:~/stuff>
```

**Now, we want to delete any file containing “analysis” in its name**

# Be efficient!

**Using wildcards: \* will be replaced by anything**  
**Now, we want to delete any file containing “analysis” in its name**

```
sandro@linux-gttg:~/stuff> rm -v *analysis*
removed `analysis-slides.txt'
removed `my analysis summary.odt'
removed `slides of analysis.pdf'
sandro@linux-gttg:~/stuff> ls
linalg.txt
sandro@linux-gttg:~/stuff>
```

# Get me outta here!

**Ctrl+C: Kill a running command,  
Clear unconfirmed command**

**yes :** Repeat the given argument over and over

```
...  
Hello world!  
Hello world!  
Hello world!  
Hello world!  
Hello world!  
Hello world!  
Hello world!  
Hello world!  
sandro@linux-gttg:~> yes Hello world! Let me^C  
sandro@linux-gttg:~>
```

# Editing text files

## nano

If the given file does not exist, it will be created.  
No need for .txt under Linux.

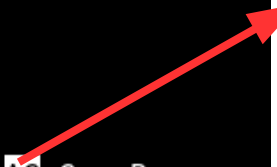


```
sandro@linux-gttg:~> nano letter
```

# Editing text files

```
GNU nano 2.4.2      File letter
[ New File ]
^G Get Help      ^O Write Out    ^W Where Is    ^K Cut Text    ^J Justify    ^C Cur Pos
^X Exit          ^R Read File   ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

^ stands for Ctrl



# Editing text files

```
GNU nano 2.4.2           File: letter           Modified
Dear reader,
this are my words.
Kind regards.

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

Ctrl+O; Enter to save

Ctrl+X to quit

No spell or grammar check under nano

vim is a much more advanced text editor, but it's harder to learn

# Looking at text files

**less**

```
sandro@linux-gttg:~> less letter
```



# Looking at text files

```
Dear reader,  
this are my words.  
Kind regards.  
letter lines 1-3/3 (END)
```

less looks like man  
because man uses  
less to display the  
manual!

Therefore, the  
keyboard shortcuts  
are the same:

- / to search
- q to quit
- h for help
- etc.

# Looking at text files

## cat

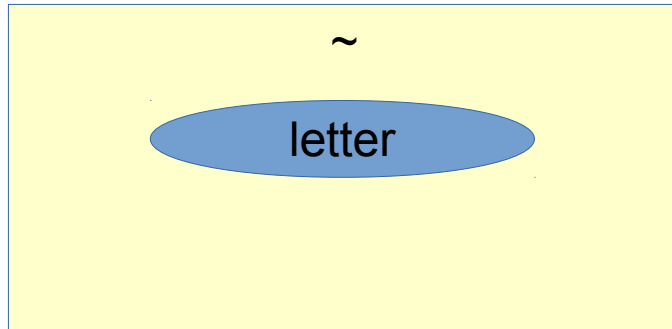
```
sandro@linux-gttg:~> cat letter  
Dear reader,  
this are my words.  
Kind regards.  
sandro@linux-gttg:~>
```

# Cats

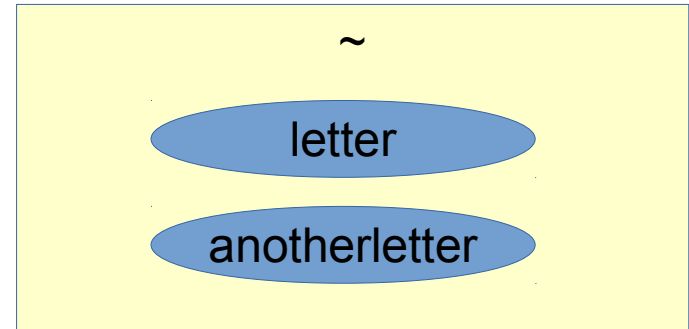


# Copying files

**cp [source] [destination]**



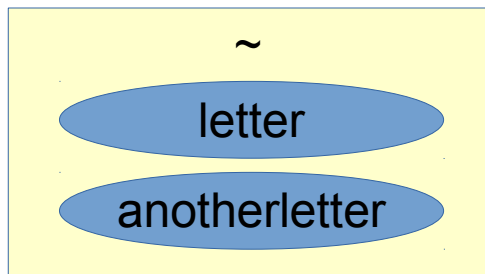
==>



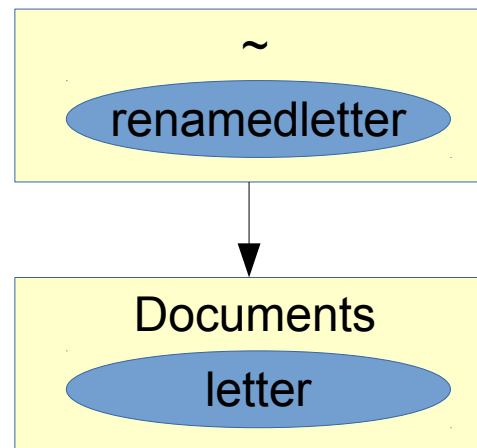
```
sandro@linux-gttg:~> cp letter anotherletter  
sandro@linux-gttg:~>
```

# Renaming / moving files

**mv [source] [destination]**



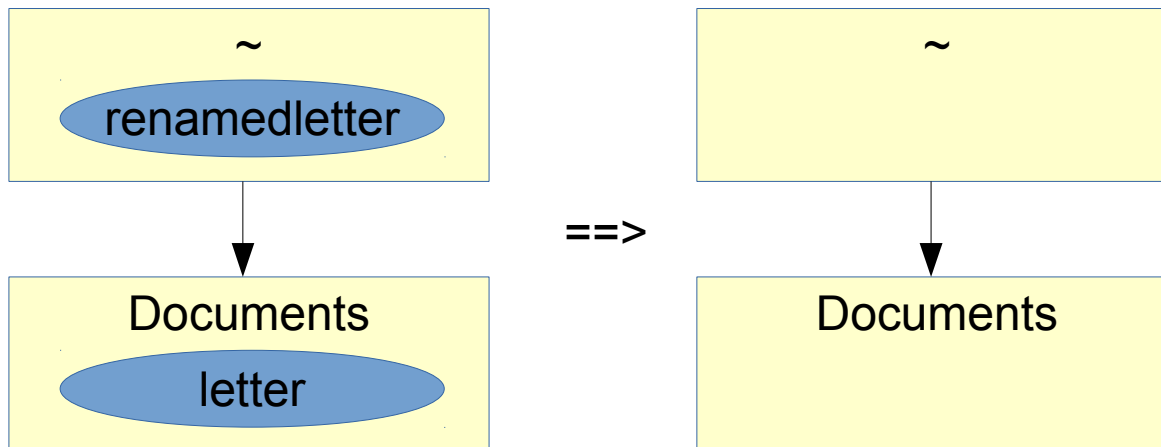
==>



```
sandro@linux-gttg:~> mv anotherletter renamedletter
sandro@linux-gttg:~> mv letter Documents/
sandro@linux-gttg:~> ls Documents
letter  miau.txt
sandro@linux-gttg:~>
```

# Deleting files

rm [file]



```
sandro@linux-gttg:~> rm renamedletter Documents/letter  
sandro@linux-gttg:~>
```



rm is irreversible. One typo and the wrong file is gone forever!

# Creating and deleting folders

## mkdir, rmdir

```
sandro@linux-gttg:~/Downloads> mkdir aFolder
sandro@linux-gttg:~/Downloads> ls
aFolder  Video.ogv
sandro@linux-gttg:~/Downloads> rmdir aFolder/
sandro@linux-gttg:~/Downloads> ls
Video.ogv
sandro@linux-gttg:~/Downloads>
```

# Delete recursively

**rmdir is safe because it only deletes empty folders.**

**To delete a folder with all its contents, use `rm -r` (r for recursive):**

```
sandro@linux-gttg:~> rmdir Downloads/  
rmdir: failed to remove `Downloads/': Directory not empty  
sandro@linux-gttg:~> rm -r Downloads/  
sandro@linux-gttg:~>
```



**`rm -r` = The Ultimate Killer Command  
`rm -r` on your home directory deletes...  
... everything you have!  
... irreversibly!  
... forever!**



# Running a program in the current folder

**Simply typing a command looks for the executable program in a few predefined directories.**

On OpenSUSE:

/home/sandro/bin

/usr/local/bin

/usr/bin

/bin

/usr/bin/X11

/usr/games

# Running a program in the current folder

Just like “..” means “parent directory”, “.” means “current dir”.

in OpenSUSE, executable files are green in ls

```
sandro@linux-gttg:~> ls
bin          helloWorld  public_html
Desktop      letter      Templates
document 1.odt  Music       Videos
Documents   Pictures
Downloads   Public
sandro@linux-gttg:~> cd Downloads/
sandro@linux-gttg:~/Downloads> mv ../helloWorld ./
sandro@linux-gttg:~/Downloads> ls
helloWorld
sandro@linux-gttg:~/Downloads>
```

moving helloWorld  
from the parent to  
this folder

# Running a program in the current folder

To run an executable in the current folder, you must add “./” before the file name of the desired executable file.

```
sandro@linux-gttg:~/Downloads> ./helloWorld
Hi there :-)
I'm done now.
sandro@linux-gttg:~/Downloads> cd ..
sandro@linux-gttg:~> Downloads/helloWorld
Hi there :-)
I'm done now.
sandro@linux-gttg:~>
```

run a program in the current directory

“.” not necessary for subfolders or absolute paths

# Where is the configuration saved?

## Per-user settings: hidden directories in the home folder

```
sandro@linux-gttg:~> ls -a
.          .emacs    .vboxclient-clipboard.pid
..         .fonts    .vboxclient-display.pid
.bash_history .gstreamer-0.10 .vboxclient-draganddrop.pid
.bashrc    .i18n     .vboxclient-hostversion.pid
bin        .ICEauthority .vboxclient-seamless.pid
.cache     .inputrc  Videos
.config    .local    .xfce4-session.verbose-log
.dbus      Music     .xfce4-session.verbose-log.last
Desktop    Pictures  .xim.template
.dmrc      .profile  .xinitrc.template
document 1.odt    Public    .xsession-errors
Documents public_html .xsession-errors-:0
Downloads Templates .xsession-errors.old
sandro@linux-gttg:~>
```

# Where is the configuration saved?

Often stored in `~/.config/`

```
sandro@linux-gttg:~> ls .config/  
goa-1.0  gtk-3.0  Thunar  user-dirs.dirs  
user-dirs.locale  xfce4  
sandro@linux-gttg:~>
```

Configuration for the XFCE  
Desktop Environment

File Manager configuration

# Where is the configuration saved?

## System wide configuration: typically in /etc/ (coloring omitted)

```
sandro@linux-gttg:~> ls /etc/
adjtime                cron.monthly          gemrc                  issue                  modules-load.d        permissions.secure     samba                  systemd
aliases                crontab               gimp                  issue.net              motd                   pkcs11                sane.d                 termcap
aliases.d              cron.weekly           gnome-chess           java                   mtab                   pki                   sas12                  tmpdirs.d
aliases.db             csh.cshrc            gnome_defaults.conf  joe                    mtools.conf           plymouth              screenrc              tmpfiles.d
alternatives           csh.login            gnupg                 jvm                    named.d                polkit-1              security               ttytype
apparmor               cups                  group                 jvm-commmon           netconfig              polkit-default-privs.local security               tuned
apparmor.d            cupshelpers          grub.d                kde4                   netconfig.d           polkit-default-privs.restrictive selinux                udev
at.deny                dbus-1               grub.d                krb5.conf             netgroup               polkit-default-privs.standard sensors3.conf          uefi
at-spi2                dconf                gtk-2.0               ld.so.cache           NetworkManager         postfix                sensors.d              UPower
audisp                 default              gtk-3.0               ld.so.conf            networks                postfix                services               usb_modeswitch.conf
audit                 defaultdomain        gtk-3.0               ld.so.conf            news                    ppp                   shadow                 uuwp
autofs_ldap_auth.conf depmod.d             host.conf             ld.so.conf.d          nfsmount.conf         pptp.d                shadow                 vconsole.conf
auto.master            dhclient6.conf       hostname              lesskey                nscd.conf             printcap               shadow.YaST2save      vdpau_wrapper.cfg
auto.master.d          dhclient.conf        HOSTNAME              lesskey.bin           nsswitch.conf         products.d             shells                 vimrc
auto.misc              dialogrc             hosts                 libao.conf            nsswitch.confbak     profile.d              silc                   vnc
auto.net               DIR_COLORS           hosts.allow           libaudit.conf         ntp.conf              protocols              skel                   wgetrc
auto.smb               dnsmasq.conf        hosts.deny            libnl                  ntp.keys              pulse                  slp.conf               wicked
avahi                  dnsmasq.d           hosts.equiv           lightdm                omc                    python3start          slp.reg.d              wpa_supplicant
bash_bashrc           dracut.conf          hosts.lpd              lirc                   openldap              rc.d                   slp.spi                X11
bash_command_not_found dracut.conf.d        hp                     localtime              opt                     raw                    smartd.conf            xattr.conf
bash_completion.d     drirc               hushlogins            login.defs             os-release            rc.status              smart_drivedb.h        xdg
bind.keys             environment          icewm                 logrotate.conf        PackageKit            rc.d                   smartd_warning.sh     xfce_defaults.conf
bindresvport.blacklist esd.conf             idmapd.conf           logrotate.d           pam.d                  rc.splash              snapper                xinetd.conf
binfmt.d              ethers              idnalias.conf         lvm                     machine-id            rc.status              ssh                     xinetd.d
blkid.conf            exports              idn.conf              magic                   mail.rc                reader.conf.d          sudoers                 xscreensaver
bootsplash            filesystems          ifplugd               manpath.config        mime.types             request-key.conf      sudoers.d              YaST2
ca-certificates       fonts               ImageMagick-6_Q16-1  inetc                  manpath.config        request-key.d          sudoers.d              yp.conf
cifs-utils            fstab               init.d                inputrc                manpath.config        request-key.d          SUSE-brand             zprofile
ConsoleKit            ftpusers            insserv.conf          iproute2               mime.types             rpm                    SUSE-release           zsh_command_not_found
cron.d                gconf               iscsi                 iscsi                  mke2fs.conf           rmt                    sysconfig              zsh_completion.d
cron.daily             gconf              gdbinit               iscsid                 modprobe.d             rpc                     sysctl.conf            zshenv
cron.deny             gdbinit.d           iscsid.conf          modprobe.d             permissions.local      rpm                     rsyncd.conf            zshrc
cron.hourly           gdbinit.d           iscsid.conf          modprobe.d             permissions.local      rpm                     rsyncd.conf            zshrc
sandro@linux-gttg:~>
```

# Where is the configuration saved?

## System wide configuration: typically in /etc/

cron.*:	Auto-execution schedules (“tasks”)
cups:	Printing
fonts:	Installed text fonts
fstab:	Autoloading drives at boot
grub:	Bootloader
hostname:	Name of this computer
NetworkManager:	Wireless networks
ntp:	Setting the clock from the internet
pulse:	Audio system
samba:	Windows shares
systemd:	Initialization stuff

...and many more

# The superuser



sandro

- ~ is /home/sandro
- has full access to that folder
- can change own settings (~/.config, etc.)
- may not access home folder of other users
- may not change system-wide settings
- may start and stop own processes



root

- ~ is /root
- has full access to any file and folder on the machine
- can change anything
- may access all home folders
- is the only one to be able to:
  - change system-wide config
  - install / remove software
  - starting and stopping any process

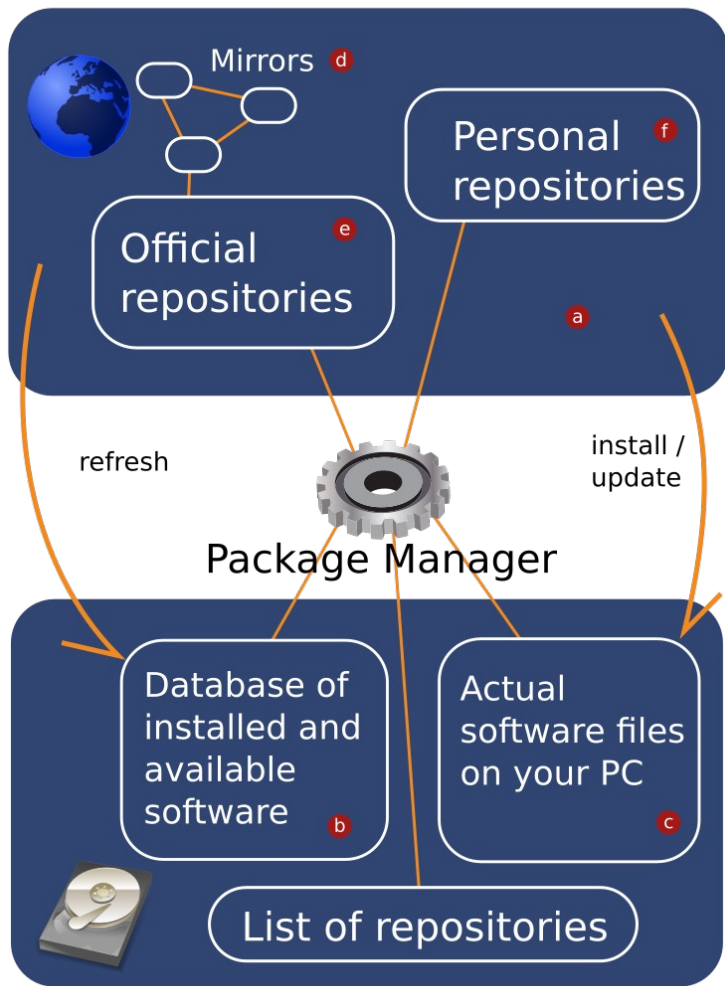


# Become a different user

- sudo [cmd]:** Execute cmd as the user root
- sudo su:** Permanently become root
- sudo su [user]:** Become a different user
- exit (or Ctrl+D):** Log out of current session

```
sandro@linux-gttg:~> cp /etc/fstab /etc/anotherfstab
cp: cannot create regular file '/etc/anotherfstab': Permission denied
sandro@linux-gttg:~> sudo cp /etc/fstab /etc/anotherfstab
root's password:
sandro@linux-gttg:~> rm /etc/anotherfstab
rm: remove write-protected regular file '/etc/anotherfstab'? y
rm: cannot remove '/etc/anotherfstab': Permission denied
sandro@linux-gttg:~> sudo su
linux-gttg:/home/sandro# rm /etc/anotherfstab
linux-gttg:/home/sandro# exit
exit
sandro@linux-gttg:~>
```

# Install and remove software packages



**Package:** Group of files, forming pieces of software, often providing one or multiple programs

**Repositories:** Online resources providing software

**Refresh** (Ubuntu: “update”): Package Manager goes through the list of repositories you are subscribed to and downloads information about available packages (e.g. version, size, dependencies, ...)

**Update** (Ubuntu: “upgrade”): PM downloads and installs the packages that are newer in the repos than on your system.

When installing a program, the PM checks whether additional programs are needed to run it (**dependencies**)

Note: Different distros, different PMs!  
e.g. Ubuntu has apt (aptitude / apt-get)

# Zypper: OpenSUSE's package manager

**Check for updates:** `zypper refresh` (short: `zypper ref`)  
**Install updates:** `zypper update` (short: `zypper up`)

**Remember: Need to be root!**

Note that if the repos are out of date, zypper will refresh automatically.

```
sandro@linux-gttg:~> sudo zypper ref
root's password:
Repository 'Main Repository (NON-OSS)' is up to date.
Repository 'Update Repository (Non-Oss)' is up to date.
Repository 'Main Repository (OSS)' is up to date.
Retrieving repository 'Main Update Repository' meta[done]
Building repository 'Main Update Repository' cache [done]
Repository 'openSUSE-42.1-0' is up to date.
Retrieving repository 'openSUSE-Leap-42.1-Update' m[done]
Building repository 'openSUSE-Leap-42.1-Update' cac[done]
All repositories have been refreshed.
sandro@linux-gttg:~>
```

# Zypper: OpenSUSE's package manager

**Check for updates:** zypper refresh (short: zypper ref)  
**Install updates:** zypper update (short: zypper up)

**Remember: Need to be root!**

no updates available

```
sandro@linux-gttg:~> sudo zypper up
Loading repository data...
Reading installed packages...

Nothing to do.
sandro@linux-gttg:~>
```

# Zypper: OpenSUSE's package manager

**Search a package:** `zypper search` (short: `zypper se`)

**Install a package:** `zypper install` (short: `zypper in`)

**Remove a package:** `zypper remove` (short: `zypper rm`)

```
sandro@linux-gttg:~> zypper se chromium
```

```
Loading repository data...
```

```
Reading installed packages...
```

S	Name	Summary	Type
	Chromium Web Browser	Browse the World->	application
	chromium	Google's opens s->	package
	chromium	Google's opens s->	srcpackage
	chromium-desktop-gnome	Update to chromi->	package
	chromium-desktop-kde	Update to chromi->	package
	chromium-ffmpegsumo	Library to provi->	package

```
sandro@linux-gttg:~>
```

# Zypper: OpenSUSE's package manager

**Search a package:** `zypper search` (short: `zypper se`)

**Install a package:** `zypper install` (short: `zypper in`)

**Remove a package:** `zypper remove` (short: `zypper rm`)

```
sandro@linux-gttg:~> sudo zypper in chromium
```

```
Loading repository data...
```

```
Reading installed packages...
```

```
Resolving package dependencies...
```

```
The following 3 NEW packages are going to be installed:  
chromium chromium-ffmpegsumo libjpeg62
```

```
3 new packages to install.
```

```
Overall download size: 53.2 MiB. Already cached: 0 B.
```

```
After the operation, additional 213.3 MiB will be used.
```

```
Continue? [y/n/? shows all options] (y):
```

chromium  
needs those  
in order to run  
=>  
automatically  
installed

“(y)”:  
just hitting  
Enter = yes

# Zypper: OpenSUSE's package manager

**Search a package:** `zypper search` (short: `zypper se`)

**Install a package:** `zypper install` (short: `zypper in`)

**Remove a package:** `zypper remove` (short: `zypper rm`)

```
...
Retrieving: chromium-47.0.2526.106-10.1.x86[done (3.1 MiB/s)]
Checking for file conflicts: .....[done]
(1/3) Installing: libjpeg62-62.1.0-31.1 .....[done]
(2/3) Installing: chromium-ffmpegsumo-47.0.2526.106-10.[done]
(3/3) Installing: chromium-47.0.2526.106-10.1 .....[done]
Additional rpm output:
update-alternatives: using /usr/lib64/chromium/chromium-generic
to provide /usr/bin/chromium (chromium) in auto mode

sandro@linux-gttg:~>
```

ready for your input ==> installation completed


# Zypper: OpenSUSE's package manager

**Search a package:** zypper search (short: zypper se)

**Install a package:** zypper install (short: zypper in)

**Remove a package:** zypper remove (short: zypper rm)

forgot sudo!



```
sandro@linux-gttg:~> zypper rm chromium  
Root privileges are required for installing  
or uninstalling packages.  
sandro@linux-gttg:~>
```



# Zypper: OpenSUSE's package manager

**Search a package:** `zypper search` (short: `zypper se`)

**Install a package:** `zypper install` (short: `zypper in`)

**Remove a package:** `zypper remove` (short: `zypper rm`)

```
sandro@linux-gttg:~> sudo zypper rm chromium
Loading repository data...
Reading installed packages...
Resolving package dependencies...

The following 2 packages are going to be REMOVED:
  chromium chromium-ffmpegsumo

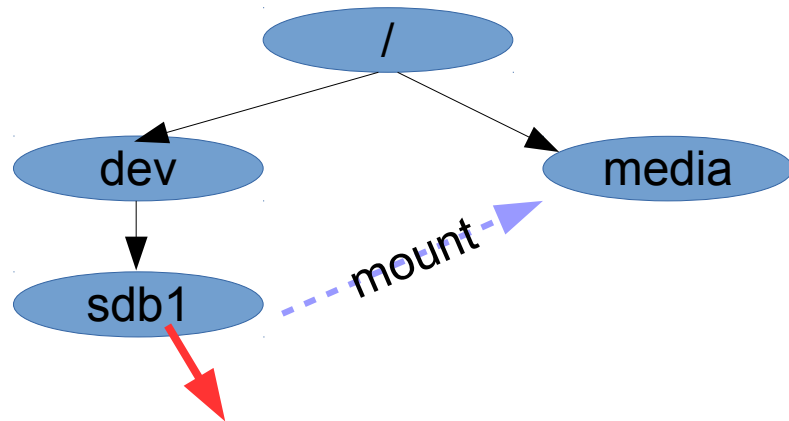
2 packages to remove.
After the operation, 213.1 MiB will be freed.
Continue? [y/n/? shows all options] (y):
```



newly unneeded dependencies are removed as well

# Dealing with storage devices

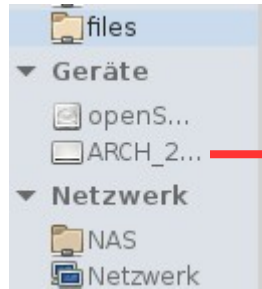
- Under Linux, devices show up as files in `/dev/`
- In order to access a device (i.e. the directories and files that it contains), the device must be mounted.
- You may mount a device into any folder you wish.



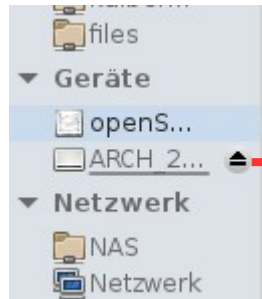
1<sup>st</sup> partition on your 2<sup>nd</sup> device (e.g. your USB disk)

# Dealing with storage devices

## Mounting and unmounting in the file manager:



Clicking the disk will mount and open it.



Clicking the eject button will unmount the disk after writing the changes to it ("flush" the caches)

# Dealing with storage devices

**mount [device] [mount point]**

folder must exist!

```
sandro@linux-gttg:~> sudo mount /dev/sdb1 /media/myUSB  
sandro@linux-gttg:~>
```

mount returns silently ==> success!

**You will now find the contents of your  
USB disk under /media/myUSB/**

# Dealing with storage devices

**sync: just flush cache to disk**

```
sandro@linux-gttg:~> sudo mount /dev/sdb1 /media/myUSB  
sandro@linux-gttg:~> sync
```



this may take a while,  
wait for the command to return

# Dealing with storage devices

**umount [device or mount point]**

**This includes sync.**

```
sandro@linux-gttg:~> sudo mount /dev/sdb1 /media/myUSB  
sandro@linux-gttg:~> sudo umount /media/myUSB  
sandro@linux-gttg:~>
```

equivalent to `sudo umount /dev/sdb1`  
command has returned ==> safe to remove disk now




**Always unmount devices before pulling the cable.**

# Managing free space on your disk

**df -h : Show disk usage (human readable format)**

```
sandro@linux-gttg:~> df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        1.2G   0    1.2G   0% /dev
tmpfs           1.2G   0    1.2G   0% /dev/shm
tmpfs           1.2G  1.9M  1.2G   1% /run
tmpfs           1.2G   0    1.2G   0% /sys/fs/cgroup
/dev/sda2       6.6G  4.2G  2.1G  68% /
sandro@linux-gttg:~>
```



This is the root folder.  
We have 2.1 GB free.

# Clearing the package cache to gain space

- **Remember: We installed chromium and threw it away again.**
- **The package files are still cached so that it's not necessary to download them again for a reinstall.**
- **sudo zypper clean: Delete cached packages**

```
sandro@linux-gttg:~> df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda2       6.6G  4.6G  1.7G  74% /
sandro@linux-gttg:~> sudo zypper clean
root's password:
All repositories have been cleaned up.
sandro@linux-gttg:~> df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda2       6.6G  4.2G  2.1G  68% /
sandro@linux-gttg:~>
```



# Scripting: automatize your workflow

- There are many scripting languages out there: bash, python, ...
- Scripts are run by so-called interpreters.
- All the commands you have used so far are bash commands.
- How can the computer tell what interpreter to use?

- either run a command specifying the interpreter to run:

```
sandro@linux-gttg:~> bash myscript.bash  
sandro@linux-gttg:~> sh myscript.sh  
sandro@linux-gttg:~> python myscript.py
```

Unlike under Windows, the file extension doesn't matter.

bash and sh ("shell") scripts are essentially the same.

- or begin the script with `#!/path/to/interpreter`

# Scripting: very basic shell scripting

```
miau.sh
```

```
#!/bin/sh
```

```
echo hello world!
```

```
cp miau.sh copyOfMyself.sh
```

```
ls
```

put every command on a single line

```
sandro@linux-gttg:~> sh miau.sh
```

```
hello world!
```

```
copyOfMyself.sh miau.sh
```

```
sandro@linux-gttg:~> ./miau.sh
```

```
bash: ./miau.sh: Permission denied
```

The file has not been marked as executable yet!

# Scripting: very basic shell scripting

## chmod a+x [file]: Making a file executable

chmod is out of the scope of this course.

To learn about chmod, check the man page or read our course script.

```
sandro@linux-gttg:~/scripting> chmod a+x miau.sh
sandro@linux-gttg:~/scripting> ./miau.sh
hello world!
copyOfMyself.sh  miau.sh
sandro@linux-gttg:~/scripting>
```

We can not execute the script as if it was an actual program, thanks to the #!... line

# Scripting: very basic shell scripting

## Example: conditional execution

true if a regular file `/home/sandro/Thesis.tex` exists.

```
isWorkDone.sh  
  
#!/bin/sh  
if [ -f /home/sandro/Thesis.tex ]  
then  
    echo "Sandro has started his thesis."  
else  
    echo "No thesis detected."  
    yes "Start your thesis!"  
fi
```

Don't forget to terminate the conditional block!

# Scripting: very basic shell scripting

## Example: loops

equivalent to: `for i in 1 2 3 4 5`

```
countToSeven.sh
```

```
#!/bin/sh
```

```
for i in {1..7}
```

```
do
```

```
    echo "Counting sheep no. $i"
```

```
done
```

Don't forget to terminate the loop!

To retrieve variables, start them with \$

# Scripting: automatize your workflow

- **Bash is very powerful and way beyond this course**
- **You can type loops etc. directly in the console (the interpreter treats your commands like a script, hitting Enter corresponds to a line break)**
- **Can run scripts at startup, upon login, before shutdown, etc. etc.**
- **Often, pre-installed commands are actually bash scripts (e.g. `/usr/bin/xflock4` which locks the screen under XFCE) => you can modify them at will for arbitrary behavior**
- **In scripts you can even read hardware sensors (e.g. on HP laptops, the file `/sys/devices/platform/lis3lv02d/position` contains data from a tilting sensor, can read out temperatures etc.)**
- **Many tutorials and examples available online**

# Commands are universal

- **Under Linux, every installed “program” is run as a command**
- **Shutdown, reboot, screen brightness, volume, WiFi and pretty much any system functionality can be controlled from the console and therefore be included into scripts**
- **If your computer has a sensor for environmental brightness, you could easily write your own script to set your screen brightness according to the light around you and the time etc.**
- **Scripts can themselves be reused as commands**
- **Pipe commands together (live demo using fortune and cowsay)**
- **Any command sequence can be bound to a keyboard shortcut... See the power of commands?**

# The course script

---

- **More complete than this course**
- **Written last semester for OpenSUSE or one year ago for Ubuntu**
- **Available under [www.thealternative.ch](http://www.thealternative.ch) → Know-How → HS'15**
- **Starts from zero, goes to more advanced topics**



# Coming up

- **Course: “Unleash the power of Linux”**
  - **23.03.16 | 17:15-19:00 | ETH HG D 1.2 (tomorrow)**
  - **Real-life applications that rock!**
- **Course: “Linux for experts”**
  - **30.03.16 | 17:15-19:00 | ETH HG D 3.2**
  - **Tiling Window Managers, a new way of controlling your PC**
  - **Name is misleading, you are already “expert” enough**
- **Stammtische: Community meetings in a casual setting (i.e. beer!)**