```
--## Enjoy my ascii slide ##
--## To show the slide install TPP (Text PowerPoint)
--## see u
--title Minimalism: CLI tools , vim & co
--date 2013-05-21
--author s@mba <samba@netstudent.polito.it>
--footer | ASCII SLIDE | Use TPP client to view this slide! |
--withborder
--newpage
--withborder
--heading INTRO
--huge hello

Questo corso si pone l'obiettivo di stimolare l'ingegno ad ogni persona che voglia automatizzare il proprio pc.

E' pensato per chi ha la curiosita' di voler esplorare qualcosa di diverso dalla solita ubuntu.

Impara a conoscere il tuo pc, programmarlo per fare un sacco di cose Ci sono un sacco di modi e linguaggi ed ognuno di essi e'
validissimo.

Vi mostrero' alcuni strumenti essenziali per fare delle operazioni che solitamente facciamo graficamente oppure in maniera ripetitiva a
mano (es:rm,mv,edit files).

--newpage
--withborder
--heading OK, COSA SERVE ?

Per fare questo abbiamo bisogno di :
 * saper leggere         i manuali sono importanti, e poi c'Ã¨ internet
 * saper pensare     sembra assurdo ma spesso si preme invio senza ragionare
 * saper chiedere    quando si chiede aiuto, non basta dire "non va"
 * saper resistere   perche' quando dura di piÃ¹ Ã¨ piÃ¹ gustoso..risolverlo
 * saper documentare perchÃ¨ quando hai risolto, non devi solo dire:a me funziona


Tutto questo con semplici comandi da tastiera e un po' di ingegno.


..e magari anche qualche link per la conf:
http://github.com/opensamba/dotfiles/
--newpage
--withborder
--heading XTERM
--center il terminale, strumento utile per fare ogni cosa

Il terminale Ã¨ uno strumento fondamentale per tutto quello che fate.
Ecco perchÃ¨ Ã¨ importante usare uno strumento flessibile che possa far tutto.

Xterm Ã¨ uno dei primi terminali che esistono e se configurato bene, funziona alla grande.

La configurazione di xterm si trova nel file ~/.Xdefaults e si ricarica con xrdb
Puo' essere ridotta a poche righe in un file ad esempio io uso:
--beginshelloutput
XTerm*background: black
XTerm*foreground: lightgrey
XTerm*cursorColor: darkorange1
XTerm*faceName: xft:Bitstream Dejavu Sans Mono:pixelsize=13
XTerm.VT100.geometry: 80x26
XTerm*VT100.translations: #override <Btn1Up>: select-end(PRIMARY, CLIPBOARD, CUT_BUFFER0)
--endshelloutput

--newpage
--withborder
--heading XTERM + screen
screen Ã¨ un programma che si puÃ² lanciare all'interno di xterm e permette
di avere piÃ¹ terminali virtuali all'interno dello stesso xterm

screen Ã¨ pensato per chi ha dei programmi che girano in remoto

screen Ã¨ molto utile anche gestire il proprio lavoro in terminali diversi
in base a quello che si fa: programmazione, chat, musica, email..

Ecco come si usa screen:

 C-a ?     | mostra un help con le varie funzioni di screen
 C-a c     | crea un nuovo terminale
 C-a space | va al prossimo terminale
 C-a p     | va al terminale precedente
 C-a ESC   | entra in copy mode (INVIO: copy, C-a ]: paste)
 C-a d     | detach: screen scompare ma tutto rimane attivo
                                (tutto senza usare il mouse!)

Per recuperare una sessione di screen in detach mode basta dare il comando:
$ screen -r
--newpage
--withborder
--heading SCRIPT (1)
Gli script sono sequenze di comandi messe in un file.
Il file a sua volta viene reso eseguibile e chiamato ad ogni necessita'.
---
```

```
--beginshelloutput
Prendo un file con dei comandi scritti dentro e vedo che tipo di file Ã¨:
$ file halt.sh
halt.sh: Bourne-Again shell script, ASCII text executable

---
E' uno script! Lo rendo eseguibile con il comando:
$ chmod +x halt.sh
---

Infine lo eseguo con il comando:
$ ./halt.sh
$ ATTENZIONE: il tuo computer si sta per spegnere, sei sicuro? [y/N]: n
---
$ Ah, ecco mi sembrava strano...
--endshelloutput
Esempi:
Scaricando il file zip[1] nella cartella example[2] ci sono una serie di script
utili da vedere e commentare per fare esperimenti e prove.
[1] http://is.gd/T49AFJ
[2] https://github.com/opensamba/minimalism
--newpage
--withborder
--heading SCRIPT (2) : halt.sh
--beginshelloutput
#!/bin/bash
#Author: s@mba
# Description: a script for shutup your pc

echo "ATTENZIONE: il tuo computer si sta per spegnere, sei sicuro? [y/N]:"
read risposta          # attendo un Input dall'utente
case $risposta in
    y|Y|yes|Yes)
        echo "Il computer verra' spento entro 30 secondi"
        sleep 30
        halt           # qui si spegno il pc
    ;;
    *)
        echo "Ah, ecco mi sembrava strano..."
    ;;
esac
exit 0
--endshelloutput
--newpage
--withborder
--heading SCRIPT (3): reboot.sh
--beginshelloutput
#!/bin/sh
#Author: samba
#Description: a script for reboot confirm
echo "ATTENZIONE: vuoi veramente riavviare il sistema? [yes/NO]: "
read resp

if [ $resp = "Yes" ] || [ $resp = "yes" ] || [ $resp = "y" ];then
    echo "Inserire l'hostname per confermare il reboot: "
    read myhost
    REALHOST=$(hostname)            # il nome della mia macchina
    if [ $myhost = $REALHOST ];then # controllo se ho inserito il nome corretto
        echo "OK, faccio il reboot"
        reboot                      # QUI ESEGUO IL RIAVVIO
    fi
else
    echo "Ah, ecco mi sembrava strano..."
fi
exit 0
--endshelloutput
--newpage
--withborder
--heading VIM
E' sempre utile conoscere un editor da terminale, perche' puo' sempre capitare di trovarsi senza grafica e dover fare delle operazioni
su file.

VI e' installato di default anche nei sistemi minimali.
VIM e' il suo successore (la M sta per VI-iMproved )
VIM e' presente su tutti i sistemi Linux, basta solo installarlo

Per utilizzarlo basta dare il comando :
$ vim nuovofile.txt
---

i     | entro in INSERT mode e posso scrivere nel file
ESC   | entro in SET mode e posso fare un sacco di cose
:w    | come ad esempio: salvare il file
:wq   | salvare e uscire
:q!   | uscire senza salvare
yy    | (yank)  copiare tutta la riga dove sono
p     | (paste) incollare quello che ho copiato
dd    | (del)   tagliare una riga
50dd  | cancellare 50 righe
dt"   | cancellare tutta la riga fino al carattere " (escluso)
100p  | incollare per 100 volte quello che ho copiato
```

```
--newpage
--withborder
--heading VIM (2): tips & tricks

[modifica caratteri]
gUw            | rende maiuscola una parola
gu$            | rende minuscola tutta la riga
/ciao          | per cercare la parola "ciao" nel file
:s/ciao/hello/ | per cercare la parola "ciao" e sostituirla con "hello"
:s/ciao/hello/g| sostituire "ciao" con "hello" in tutto il file (global)

[autocompletamento]
C-P  | completa la parola cercando all'indietro parole simili nel file
C-N  | completa ciÃ² che stiamo scrivendo cercando parole simili nel file
C-D  | quando digito :<cmd name> mostra i possibili completamenti

[commands]
:e file.c | Apro file.c e se non esiste lo creo
:sp       | (split) crea una nuova finestra orizzontale
:vsp      | (split) crea una nuova finestra verticalmente
C-w C-w   | Cambio di finestra
C-w C-c   | Chiudo la finestra

--newpage
--withborder
--heading VIM (3): commands and conf

I comandi in vim permettono di modificare il suo comportamento:

:number on   | mostra i numeri di riga
:syntax on   | colora la sintassi
:autoindent  | indenta automaticamente il codice
:bg=dark     | colora la sintassi sapendo che ho un terminale nero
:history=500 | ricorda gli ultimi 500 comandi

---
il file ~/.vimrc puo' essere configurato a nostro piacere per rendere queste configurazioni persistenti.
Inoltre si possono mappare anche tasti per fare qualcosa di utile come:

- commentare il codice automaticamente con i tasti "cc":
noremap   <silent> cc      :s,^\(\s*\)[^# \t]\@=,\1# ,e<CR>:nohls<CR>zvj
---
- salvare il file ed eseguire python del file stesso con F5:
map <f5> :w<CR>:!python %<CR>
---
- salvare ed eseguire perl del file con F6:
map <f6> :w<CR>:!perl %<CR>

e cosÃ¬ via...  WIKI: http://vim.wikia.com
               conf: https://github.com/opensamba/dotfiles/blob/master/.vimrc
--newpage
--withborder
--heading AWK(1)

E' uno strumento fondamentale per il parsing di file
Parsing significa che prendi un file, lo filtro e ti cambio l'output mandandoti solo la parte interessante.
---
--beginshelloutput
$ cat file.txt
sopra la panca
la capra
campa
---
$ awk '{print $1}' file.txt
sopra
la
campa
---
$ awk '{print $NF}' file.txt
panca
capra
campa
--endshelloutput
--newpage
--withborder
--heading AWK (2)
AWK usa delle macro (delle funzioni) del linguaggio c, infatti Ã¨ molto utile da usare per chi conosce un po' di programmazione c.

Pero' e' molto piÃ¹ intuitivo del C!

AWK e' presente su tutti in sistemi operativi unix based con tanto di manuale e nelle distribuzioni piu' nuove anche esempi:

$ man awk
$ ls /usr/share/doc/mawk/

print $1          | stampo la prima colonna
print $2          | stampo la seconda colonna
print $NF         | stampo l'ultima colonna (Number Field)
print $(NF-1)     | stampo la penultima colonna
printf()          | stampa caratteri in mille modi, la stessa printf del c
a=1               | assegno un valore a una variabile
```

for(i=0;i<NF;i++) | eseguo un ciclo che parte da i=0 e finisce con i=NF

--newpage
--withborder
--heading AWK (3): esempi

Vediamo AWK all'opera con alcuni esempi:

--beginshelloutput
$ cat /etc/passwd | awk -F: '{print $1}'
...
sshd
postfix
samba
...

$ cat FILE
1. ciao
2. io mi
3. chiamo pippo e
4. sono solamente un semplice
5. file di prova ...

$ awk '/[0-9]/{for(i=2;i<=NF;i++){printf("%s ",$i)}}END{printf("\n")}' FILE
ciao io mi chiamo pippo e sono solamente un semplice file di prova ...
--endshelloutput
--newpage
--withborder
--heading SED

SED e' un editor non interattivo.
SED non apre il file completamente per modificarlo, ma esegue le operazioni nelle linee specifiche.

E' come pensare a VIM pero' senza vedere cosa state andando a modificare.

PerchÃ¨ bisognerebbe pensare ad una follia del genere ?
PerchÃ¨ spesso gli script hanno bisogno anche di modificare i files e quindi Ã¨ meglio lanciare un comando :)

Se ad esempio abbiamo un file di dimenstione 4GB, non possiamo aprirlo immediatamente con VIM ma nemmeno con un qualsiasi editor perchÃ¨ ci metterebbe troppo tempo per caricare in memoria tutto il contenuto del file. SED serve per eseguire delle operazione sulle righe specifiche di un file.

Generalmente si usa SED per sostituzioni/inserimenti/cancellazioni la sua codifica e' molto intuitiva e per aiutare nella ricerca di pattern (schemi di ricerca) si usano delle espressioni regolari.

--newpage
--withborder
--heading SED (2): esempi

--beginshelloutput
$cat FILE
1. ciao
2. io mi
3. chiamo pippo e
4. sono solamente un semplice
5. file di prova ...

- tolgo i numeri:
$ cat FILE | sed 's/[0-9]\. //'
ciao
io mi
chiamo pippo e
sono solamente un semplice
file di prova ...
--endshelloutput
--newpage
--withborder
--heading SED (3): esempi
--beginshelloutput

- metto la il punto a fine riga
$ cat FILE | sed 's/$/./'
1. ciao .
2. io mi .
3. chiamo pippo e .
4. sono solamente un semplice.
5. file di prova ... .

- sostituisco pippo con samba
cat FILE | sed 's/pippo/samba/'
1. ciao
2. io mi
3. chiamo samba e
4. sono solamente un semplice
5. file di prova ...

--endshelloutput
more info:

```
http://www.grymoire.com/Unix/Sed.html

--newpage
--withborder
--heading Regular Expression (Regexp)
sono dei modi per richiamare un gruppo di parole/stringhe, usano dei caratteri particolari per comprendere quello che si desidera ed e'
piÃ¹ facile vedere alcuni esempi per capire come funzionano

[0-9]       : cerco un numero compreso tra 0 e 9
[0-9][0-9]  : cerco due numeri consecutivi compresi tra 0 e 9 , ad esempio 83 o 38 o 13 ecc...
[A-Z]       : cerco una lettera maiuscola
^[A-Z]      : cerco una lettera maiuscola all'inizio di una riga
ros.*       : cerca una parola che inizi con 'ros' e poi abbia qualisasi carattere ripetuto
              il "." sta per QUALSIASI CARATTERE e l'* sta per RIPETUTO
            es (rose sul tuo corpo come luna sul mio viso
                rosso fuoco arde il demonio in autunno
                rosina e' un giocatore che fa prrrr... )
ros?        : cerco una parola che inizi con 'ros' e abbia UN SOLO carattere carattere dopo
            es: rosa Ã¨ proprio una gran bella figliola
                  rose rosse per te ho comprato stasera
              no: rosina/rospo/rosicare, etc.. )
^[0-9]\+    : cerco una serie di numeri ripetuti  all'inizio di una riga
            es: (0000 ciao , 1234 pippo )

--newpage
--withborder
--heading Regexp (2)

caratteri importanti:

^       : inizio riga
$       : fine riga
.       : qualsiasi carattere
+       : carattere ripetuto (quale carattere? quello specificato  precedentemente)
*       : qualsiasi carattere ripetuto (quale carattere, quello specificato precedentemente)
[abc]   : un insieme di caratteri (in questo caso: a,b oppure c)
\t      : tabulatore
\n      : a capo

consiglio:  per imparare a conoscere le regulare expression usate l'opzione cerca
            di VIM con l'opzione ":set hlsearch"

link A Tao of Regular Expression
http://www.let.uu.nl/~Michael.Moortgat/personal/Courses/TT02/tao_regexps.html
--newpage
--withborder
--heading ALTERNATIVES:  Minimal way to do the same

- Grafica:      Fluxbox invece di GNOME, KDE, XFCE
- Musica:       MPD & ncmpcpp invece di rhythmbox, banshee, vlc
- Mail:                 Mutt invece di thunderbird o Evolution
- System check: Conky invece di gnome-applet, top, ps, df, free, ...
- Screen lock:  xscreensaver invece di gnome-screensaver
- File Manager: vimfm invece di nautilus, konqueror
- Volume:               alsamixer, amixer invece di varie applet
- Chat:                 weechat o irssi invece di pidgin, kopete, empathy, ...

--newpage
--withborder
--heading FLUXBOX

Fluxbox e' un Window Manager, come GNOME, KDE, XFCE o altro
La differenza e' che fluxbox e' leggero e permette di fare tutto quello che si vuole.

Appena installato fluxbox crea una cartella .fluxbox che va permette di essere modificata.
i file piÃ¹ importanti da ricordare sono:

.fluxbox/keys     modifico le scorciatoie da tastiera
.fluxbox/init     modifico come il mio sistema deve iniziare e dove mettere le cose
.fluxbox/startup  specifico quali applicazioni far partire quando lancio fluxbox
.fluxbox/style    modifico l'aspetto e sceglo i colori delle finestre

more info:
http://fluxbox-wiki.org

--newpage
--withborder
--heading FLUXBOX(2): conf

.fluxbox/keys
Control q       :Close
Mod1 F9         :Minimize
None F10        :Maximize
Mod4 t          :Exec xterm -e "screen"
Mod4 v          :Exec xterm -e "vim"
...

more at:
https://github.com/opensamba/dotfiles/tree/master/.fluxbox

--newpage
```

```
--heading Philosophy

--center Minimalism Philosophy

https://en.wikipedia.org/wiki/Minimalism_%28computing%29

KISS: Keep It Simple, Stupid!
- essere capace a risolvere i problemi piu' in fretta
- poter scrivere codice per problemi complessi in poche righe
- produrre codice di qualitÃ
- poter costruire grandi sistemi, facili da mantenere
- codice flessibile, facile da estendere/modificare all'occorrenza
- archiviare piÃ¹ di quanto ti immagini
- lavorare a livello professionale in grandi gruppi con poco sforzo

Se segui piccoli concetti di minimalismo scoprai che spesso le cose piÃ¹
semplici sono anche quelle piÃ¹ veloci, di qualitÃ  e personalizzabili!

more info:
http://people.apache.org/~fhanik/kiss.html



--newpage
--heading More Links

LEARN:
http://www.pluto.it/files/ildp/guide/abs/index.html
http://www.shell-fu.org/
http://vim.wikia.com
http://www.grymoire.com/Unix/Sed.html
http://cb.vu/unixtoolbox.xhtml#shells

THINK:
https://en.wikipedia.org/wiki/Minimalism_%28computing%29
http://people.apache.org/~fhanik/kiss.html

FUN:
http://www.shell-fu.org/
http://commandlinefu.com/

MY NERD PLACE:
http://www.autistici.org/underscore - Underscore_TO (Hacking Lab Torinese)
http://link.autistici.org/          - Link dell'hacking italiano
http://sambismo.wordpress.com       - my Blog

--newpage
--heading Thanks!



--center <samba> samba@netstudent.polito.it


--footer |askmeanything|buymeabeer|lookmeintheeyes|kissmyshell|smilemewhenyouseeme|
```