

# CLI Productivity for Mac Admins

(or anyone else, really)

# Better...stronger...faster

- Tab completion (readline support)
- Wildcard (\*)
- Option-arrow navigation for jumping between words
- ^A, ^E, ^C, and ^D
- !! (Repeat the last command, e.g., `sudo !!`)
- ^W (Delete word behind cursor)

# More !

- `!:2` (or 1, or 3, etc.)  
`ping google.com`  
`nslookup !:1 (=nslookup google.com)`
- `!foo`  
Run the most recent command that starts with 'foo', (e.g., `!ls`)
- `!$`  
Run the last word of the previous command  
`ping google.com`  
`nslookup !$`
- `!*`  
Run the previous command except for the last word

# Even more !

- `!*:p`  
Print out the previous command except for the last word
- `^foo^bar`  
Run the previous command replacing foo with bar
- `^r` (reverse-intelligent-search)
- `ESC .`  
Insert the arguments from your last command

# and so on...

- tab-tab, (i.e., where are my pythons?)

python	python2-config	python2.7	pythonw2.6
python-build	python2.5	python2.7-config	pythonw2.7
python-config	python2.5-config	pythonw	
python-local-exec	python2.6	pythonw2	
python2	python2.6-config	pythonw2.5	

- source ~/.bash\_profile (or . ~/.bash\_profile)

- ~/.ssh/config

Host someserver

User you

IdentityFile "~/.ssh/id\_rsa"

HostName someserver.example.com

Port 1337

- `which` (use it instead of browsing `/usr/bin` in the Finder)
- `vi` keybindings for `man` (`/`, `^D`, `^U`, `q`, etc.)
- `xattr -d com.egnyte.syncstatus id_rsa.pub`  
Modify extended attributes
- `sudo=echo "$password" | sudo -S`  
Josh's `sudo` hack
- `alias ls='ls -leAh@'`  
Set your favorite options for `ls` in `~/ .bash_profile`
- `cd ~/src && for file in *.m; do cp "$file" "${file%.m}.m-old"; done`  
Rename lots-o-files

# awkward

- `awk ' {print $2} '`  
Print the second word
- `awk '/foo/ { print $2 } '`  
It's like awk has grep built in!
- `awk ' {if (NR!=1) {print}} '`  
Print all lines except the first
- `awk ' { NR==2 } '`  
Print the second line
- `awk ' { NR==2 { print $3 } } '`  
Print the third occurrence on the second line

# Pick your editor

- Just don't use ed
- `:%s/\~\~/Documents/\~\~/Sites/gc`  
Search and replace
- `:retab`
- `:set sw=num first`
- `:set paste, set nopaste`
- `^d` (Half page down)
- `^u` (Half page up)
- `' '` (Go to last position)



# Pick your editor, cont.

- `let@/= " "`
- `gU` (uppercase)
- `u` (undo)
- `^r` (redo)
- `>> tab (set sw=4 first)`

# find

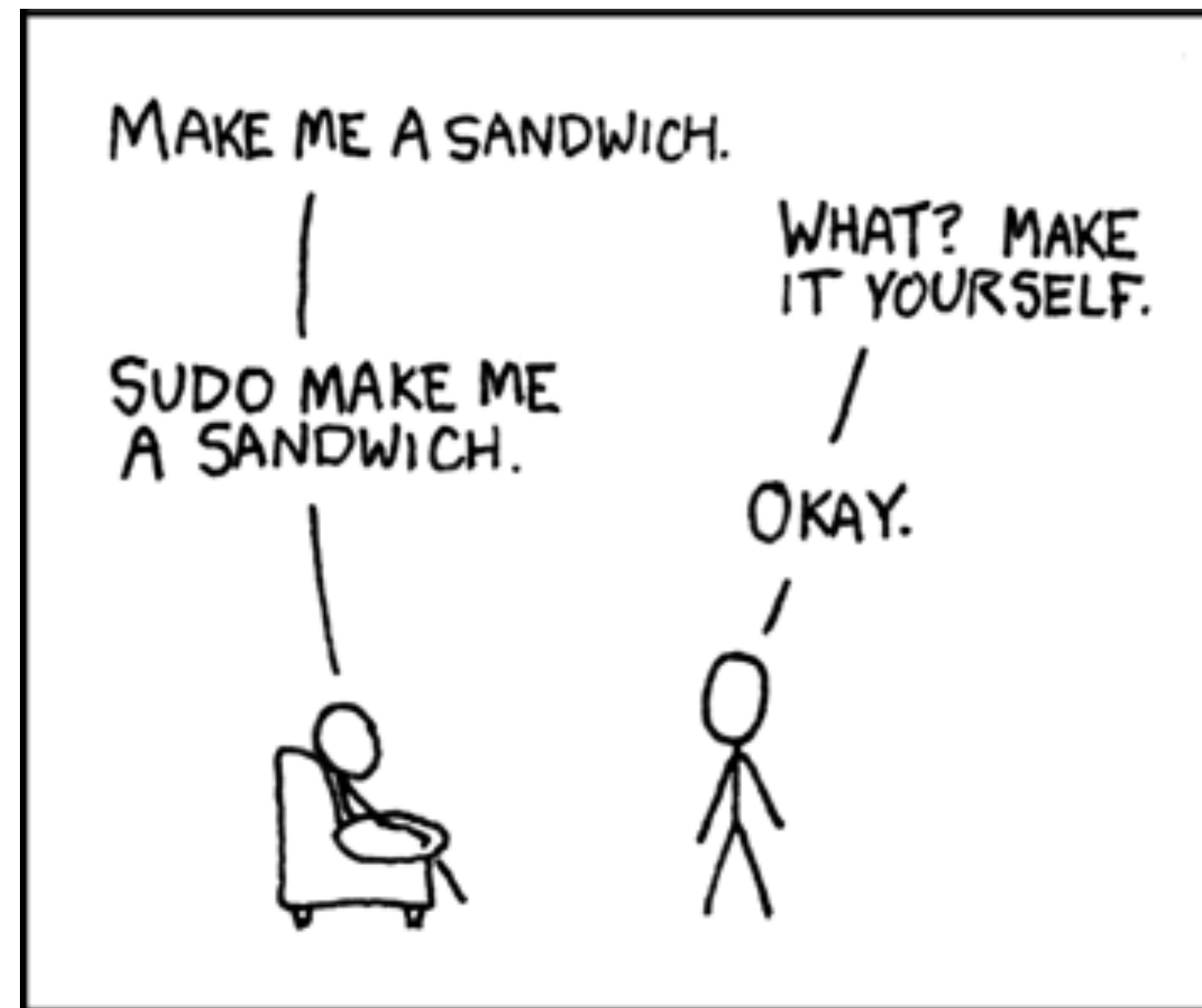
- `find . -iname filename`  
Case insensitive
- `cd ~ && find . -name "*~" -exec ls -l {} \;`
- `cd ~ && find . -name "*~" -exec mv {} .vim/tmp/ \;`
- `cd ~ && test -d .vim/tmp && find . -name "*~" -exec mv {} .vim/tmp/ \;`
- `some_var=`cd "$some_dir" && find . -name *.txt | wc -l`  
echo "There are $some_var items in $some_dir."`

# Misc.

- `tr -d '"'`  
Remove " from output
- `grep \.htm$`  
Match exactly with `grep`, (e.g., `.htm` but not `.html`)
- `echo "export PATH=\" /Users/$userName/.example:\$PATH\" " >> ~/.bash_profile`  
Substitute `$userName` but not `$PATH`
- use `test -f` (or `-d`) when testing scripts, then `echo $?`
- `curl --data "some-field=foo" example.com/form`  
POST with `cURL`
- `ln`, `ldapsearch` (useful)

# Useful shell scripting stuff

- for loops, counters, recursion
- `$?`
- `${#x}`
- `&&`
- `||` (or)
- `[ ]` (syntactic sugar for `test`)
- `[ [ ] ]` (shell builtin)



# Thanks!

<https://github.com/futureimperfect>