



***RAPPELS***  
***ligne de commande***

# *Linux on windows*

- <https://learn.microsoft.com/fr-fr/windows/wsl/install>

# ***Why master the linux command line?***

- Better control over the machine
- Faster (once you get the hang of it)
- You can automate a lot of things (create 1000 files in one go)
- Available everywhere (linux, mac, and windows with a bit of work)
- => Basically a requirement if you do computer science

# *Operating Systems*

- Basically 2 groups:
  - Microsoft descendants
  - Unix descendants
- Family Tree:
  - [https://eylenburg.github.io/os\\_familytree.html](https://eylenburg.github.io/os_familytree.html)

# ***What is linux ?***

- Linux is a kernel (noyau)
- Kernel connects ths software to the hardware (HD, RAM, CPU...), you can see it as a piece of the operating system
- Different ‘flavors’ of Linux (linux distributions)
  - Ubuntu
  - Debian
  - Kali
- Open source => main reason why it’s so popular

# ***The Shell***

- Shell is the interface we use to interact with the operating system
- It takes our commands and gives them to the OS to perform
  - It's named 'shell' because it forms an outer layer around the OS
- Different shells:
  - bash => Bourne-Again Shell
  - Zsh => Z shell

# *The terminal*

- A terminal is a program that runs the shell
- Originally those were physical devices, but now we work with software terminals

VT100, 1978

- Terminal Emulator is what we use to interact with the shell
- Terminal (physical device) → Shell (user interface)



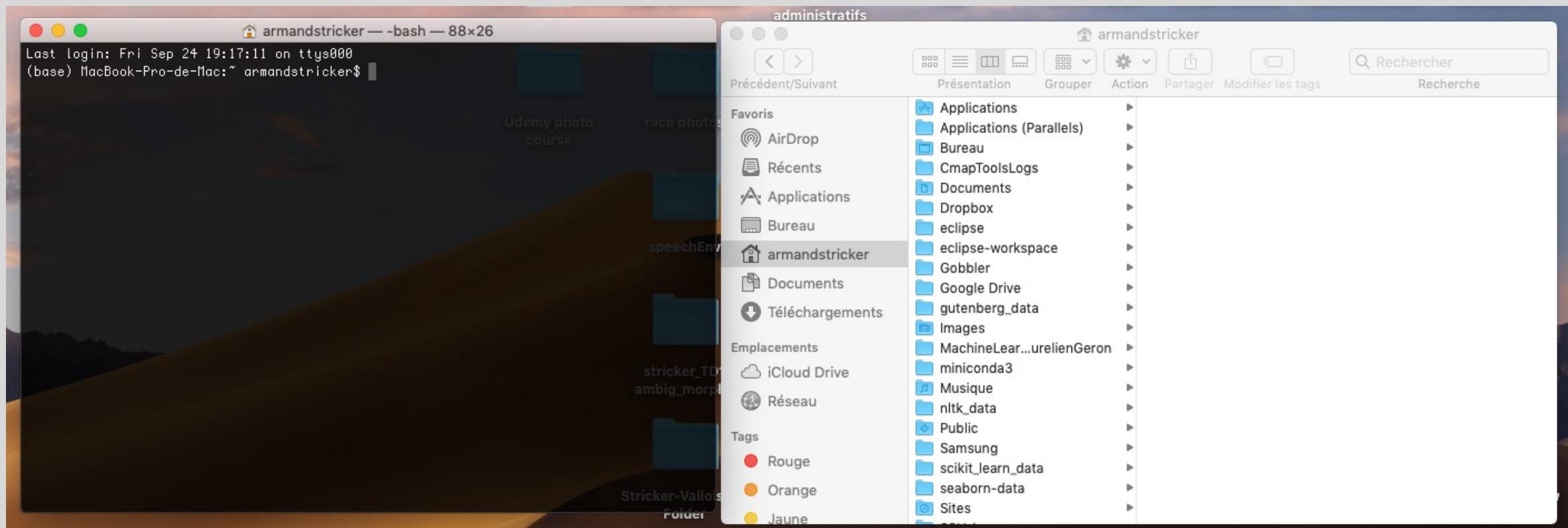
# ***How to open the terminal***

- Easy on mac and linux
- Cf. first slide for Windows

# ***In case you need help***

- Can't remember what a command is for ?
  - Consult the man
    - man (manual) try 'man man'
    - Or **-h**
    - Or **--help**
  - Search for something you would like to do:
    - Apropos => does a keyword search through the manual
    - Apropos apropos

# *Command line/terminal vs GUI (Graphical User Interface)*



# ***Command line/terminal vs GUI (Graphical User Interface)***

- Moving around your directories:
  - GUI advantage => always know where we are
  - How do we see where our current location using the terminal ?
  - How do we see what's in a folder ?
  - How do we change folders ? (and go back to the previous directory ?/go back to home ?)  
Absolute vs. Relative paths

# ***Basic/ Command line/terminal vs GUI (Graphical User Interface)***

- pwd => print working directory
- cd => change directory (cd .. To go back)
- ls => list contents in directory (-a and -l)
- cat => concatenate : print file content in terminal

# ***Folder Structure***

- Keep in mind : everything is a file inside a directory ! (even the commands) cf. bin file

# ***Options syntax***

- Examples:
  - Clear -x (see man)
  - Ls -al

# ***Create a file or directory***

- Mkdir => creat a direcory (-p option)
- Touch => create any file (.txt, .pdf, .pptx etc...)
- Display a line of text: echo

# ***Redirecting Standard output***

- Dump content into a file: >
- Write something into a file: echo ‘my favorite band is’ > file1.txt
- What happens when you try to add the band name ?
- To concatenate : >> (-n option with echo, use quotation marks)
- Other examples
  - date > today.txt
  - pwd > directory.txt

# ***See contents of a file***

- Cat
- Less
- Head

# ***Command line/terminal vs GUI (Graphical User Interface)***

- Manipulating files/directories:
  - Copy a file ?
  - Remove a file ?
  - Rename or move a file into a directory?

- Cp (-r)
- Rm (-r)
- Mv (-r)

# **Piping**

- Pass the output of 1 command to another command
- Cat hello.txt | wc -w (wc : word count, returns # lines, # words, #bytes)

# ***Useful Keyboard shortcuts***

- Ctrl L to clear (or type clear)
- Ctrl A to get to the start of a line
- Ctrl E to get to the end of a line
- Tab for autocomplete
- Ctrl ➔ to skip a word | alt ➔ on mac

# ***Quick Recap***

- Everythong is a file !
- Important to know where things are and moving them to where you want them to go
- Practice being comfortable switching directories with the terminal => better grasp on your machine + what is actually stored on your computer

# ***Done***

- Resources:
  - <https://www.freecodecamp.org/news/the-linux-commands-handbook/>
  - <https://www.youtube.com/watch?v=ZtqBQ68cfJc>

# *Installing Packages*

- A package can contain anything someone wants to install => firefox, minecraft etc.
- To install them we need package managers :
  - Dpkg
  - Apt (advanced package tool)
  - Brew for mac
  - All rely on repositories : a server/storage location that contains a collection of the software we might want to use

# *Installing Packages*

- Install python if you haven't already => sudo apt-get install python3.9 or go to python website

# ***Create a virtual environment***

- Sudo apt install python3.9-venv

# ***Exercise – use the command line to set up a project + get/analyze texts from project gutenberg***

- Using the command line:
  - Create a directory
  - Create and open a python file
- Using python :
  - Visit project gutenberg
  - In a python script, download at least 2 of the most popular texts from project gutenberg and save them into a directory (cf urllib documentation)
- Analyze first paragraph of one of the texts using spacy
  - Cut the text into paragraphs and identify the first one.
  - Create a virtual environment and install the spacy package
  - Go to the spacy documentation to see how you can do POS-tagging for the first sentence of the first paragraph.