



Objectifs

1. Manipuler le **terminal** avec les commandes de bases
2. Comprendre comment afficher le contenu d'un répertoire
3. Savoir se repérer dans **l'arborescence**
4. Comprendre comment naviguer dans l'arborescence
5. Connaître ce que sont les **arguments** ou les paramètres d'une commande
6. Comprendre la signification de **., ..** et **~**
7. Créer des **répertoires** et des fichiers avec l'interface en lignes de commandes
8. Copier et déplacer des fichiers et répertoires avec le CLI (commande line interface)



Utilisateurs Mac et Windows

Et si nous n'utilisons pas Linux, mais MacOS ou Windows ?

MacOS

MacOS descend aussi d'Unix, et il est livré avec un terminal. Les commandes sont presque les mêmes que Linux.

Windows

Windows est un descendant de **MS-DOS**, qui s'est lui-même inspiré d'Unix. Il fournit un terminal, offrant des *équivalents* aux commandes Unix, mais avec des noms parfois différents !

→ [Cheatsheet des commandes & différences du terminal Linux & Windows](#)



A quoi ça sert ?

Le terminal permet d'**exécuter des commandes en les tapant sur le clavier**.

Quel intérêt à une époque où nos ordinateurs ont des interfaces graphiques conviviales et nos smartphones des écrans tactiles ?

Eh bien, même si ce n'est pas immédiatement évident pour toi, le terminal offre **de nombreuses possibilités**. Par exemple :

- **Manipuler** (copier, déplacer, renommer, supprimer) des fichiers et des répertoires
- **Extraire** des informations spécifiques à partir de fichiers texte volumineux plus rapidement qu'avec un éditeur de texte
- **Rechercher** efficacement des fichiers dans une arborescence de fichiers, selon différents critères (nom, date de modification, etc.)
- **Installer** un logiciel

Avec un peu de pratique, certaines opérations se font **beaucoup plus rapidement** dans le terminal que via l'interface utilisateur graphique !



A quoi ça sert ?

Un peu d'histoire

Le mot terminal désignait à l'origine un "**ordinateur**" **rudimentaire**, équipé d'un simple clavier, et relié par un réseau à un serveur central.

Ci-dessous, un des précurseurs, le Dec VT100.





L'utilisateur se connecte au serveur et peut alors entrer des commandes à l'aide du clavier.

Bien que la souris ait été inventée en même temps que les terminaux, les ordinateurs n'en étaient pas équipés et les écrans étaient très limités et ne pouvaient afficher que du texte, en vert sur un fond noir (#Matrix)

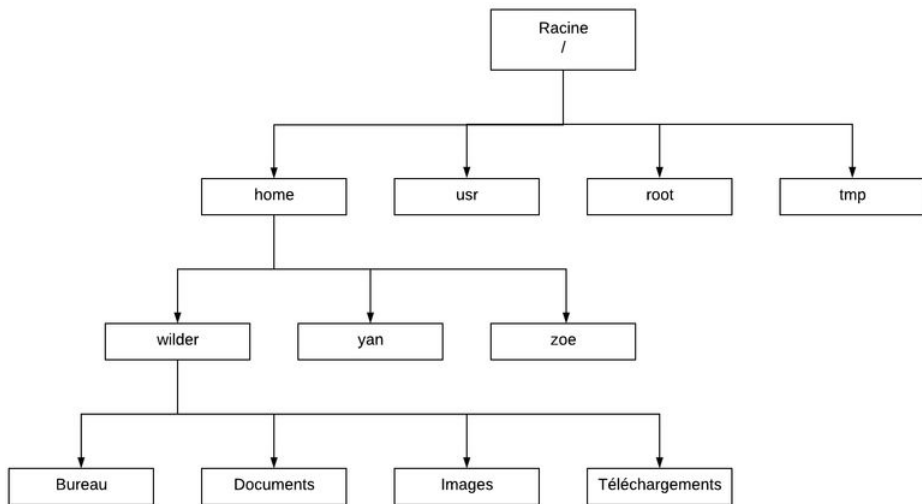




L'arborescence des répertoires sous Linux

Sur tous les systèmes d'exploitation, les fichiers et les répertoires sont organisés en "**arborescence**", c'est-à-dire une **structure hiérarchique**.

Le terme "arborescence" n'est pas une coïncidence, car une arborescence de fichiers ressemble à un bon vieil arbre : la "**racine**" de l'arborescence serait le **tronc**, les **répertoires**, ses **branches**, et les **fichiers**, ses **feuilles**.





👁️ Explorer l'arborescence avec le terminal

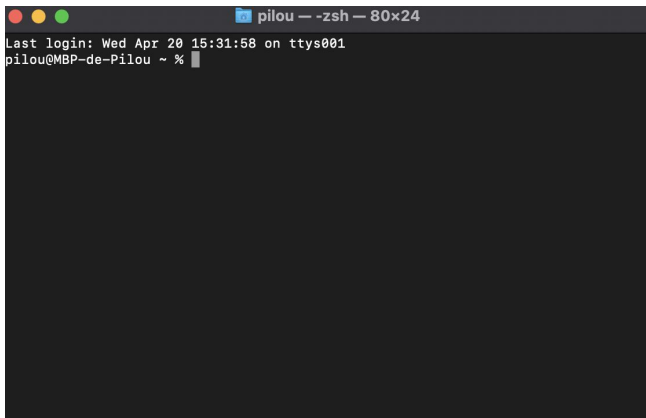
Nous entrons dans le vif du sujet !

Commençons par regarder cette [vidéo](#), qui montre comment utiliser le terminal et quelques commandes de base.

(Ne sois pas passif en regardant la vidéo et tape en même temps les commandes sur ton terminal)



👁️ Explorer l'arborescence avec le terminal



```
pilou - zsh - 80x24
Last login: Wed Apr 20 15:31:58 on ttys001
pilou@MBP-de-Pilou ~ %
```

Un curseur clignotant est affiché, précédé de **l'invite de commande** :

- La première partie **username@Computer-Name** indique **qui** nous sommes et **sur quelle machine nous travaillons**.
- La deuxième partie indique **où l'on se trouve** dans l'arborescence... Nous reviendrons sur la signification de **~** plus tard.
- Le dernier caractère, **%** ou **\$**, **termine l'invite**. C'est après que l'on écrit nos commandes



→ Lister le contenu d'un répertoire avec ls "list"

- **ls** affiche les fichiers et répertoires contenus dans le répertoire courant (celui dans lequel nous sommes actuellement)
- **ls -l** affiche la même chose mais avec plus de détails
- **ls <nom-du-dossier>** affiche le contenu du dossier en question

→ Afficher le répertoire courant avec pwd "print working directory"

- La commande **pwd** permet d'afficher le répertoire courant

→ Changer de répertoire avec cd

- **cd <nom-du-dossier>** se déplacer sous le dossier en question
- **cd ..** : remonter dans le répertoire parent
- **cd ~** accéder au répertoire personnel de l'utilisateur actuel. C'est un moyen rapide d'accéder à l'adresse **/home/username** (remplacer le **username** par celui de l'utilisateur actuel)



→Créer un nouveau répertoire avec mkdir “make directory”

- `mkdir <nom-du-dossier>`

→Supprimer un répertoire VIDE avec rmdir “remove directory”

- `rmdir <nom-du-dossier>`

→Supprimer DÉFINITIVEMENT un répertoire ou un fichier avec rm “remove”

- `rm <nom-du-dossier> ou <nom-du fichier>`

→Créer des fichiers avec touch

- `touch <nom-du-fichier>`

→Copier des fichiers avec cp “copy”

- `cp <nom-du-fichier>`
- La commande est utilisée avec deux paramètres : la **source** et la **destination**. Il peut y avoir plusieurs sources, mais il n'y a qu'une seule destination.

```
cp apple banana
```

```
cp apple fruits/orange
```



→ Déplacer avec mv “move”

- Comme avec cp, on utilise 2 arguments : la source et la destination

```
mv essai.tx test-2/
```

→ Renommer avec mv “move”

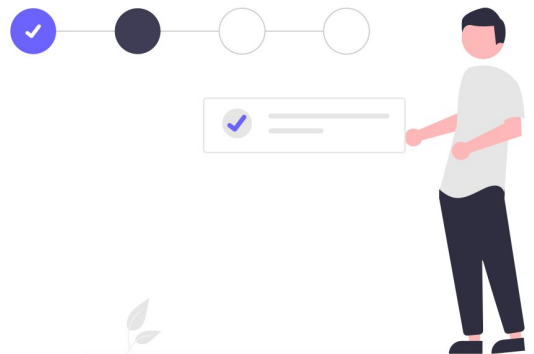
```
mv essai.tx essai.txt
```

→ Déplacer et renommer avec mv “move”

```
mv essai.tx test-2/essai.txt
```








Pratiquons !





[Télécharger ce dossier](#)

-  **Créer trois répertoires sous planets :**
 - real
 - fictional
 - inhabited
-  **Créer trois autres répertoires dans le répertoire planets/real:**
 - terrestrial
 - gas-giants
 - dwarf-planets
-  **Déplacer chacune des neuf planètes réelles** à l'endroit correct parmi les trois sous-dossiers de planets/real,
-  **Déplacer les trois planètes fictives** sous "planets/fictives",
-  **Copier les quatre planètes habitées** (qu'elles soient habitées par des humains ou des robots) dans planets/inhabited
-  Enfin, puisque Pluton n'est plus vraiment considérée comme une planète, **l'effacer, ainsi que son répertoire parent.**