

**DEV**



# **Lignes de commande et scripts**

# Objectif



- Comprendre ce qu'est une ligne de commande (CLI) et son rôle.
- Apprendre les commandes de base pour naviguer, gérer des fichiers et dossiers.
- Comprendre ce qu'est un script et savoir en écrire un simple.
- Être capable d'automatiser une petite tâche avec un script.

# CLI vs GUI



- CLI = taper commandes (texte)
- GUI = clics (visuel)
- Avantages : rapide, automatisable
- Inconvénients : difficile au début, erreurs possibles



# Structure d'une commande

- Commande + option + argument
- Exemple Unix : `ls -l /home/user`
- Exemple Windows : `dir C:\Users`



# Navigation

- Unix : `pwd`, `ls`, `tree`
- Windows : `cd`, `dir`, `tree`
- Répertoires spéciaux :
  - `.` → dossier courant
  - `..` → dossier parent
  - `~` (Unix/Linux) → dossier personnel de l'utilisateur
  - `/` (Unix) ou `C:\` (Windows) → racine du système de fichiers
- Variables d'environnement utiles (Windows : `%USERPROFILE%`, Unix : `$HOME`)



# Manipulation fichiers

- Créer : `mkdir`, `touch` (Unix) / `echo >` (Windows)
- Lire : `cat` (Unix) / `type` (Windows)
- Copier/déplacer : `cp`/`mv` (Unix) / `copy`/`move` (Windows)
- Supprimer : `rm`/`rmdir` (Unix) / `del`/`rmdir` (Windows)



# Redirections & opérateurs

- `>` : sortie vers fichier
- `>>` : ajout
- `<` : entrée
- `|` : pipe
- `&&, |||` : conditions



# Recherche

- Unix : `find`, `grep`
- Windows : `dir /s`, `findstr`



# Scripts Unix

Un script est un fichier texte qui regroupe plusieurs commandes pour les exécuter automatiquement.

En unix, le fichier prend l'extension .sh et la première ligne obligatoire est `#!/bin/bash`

```
#!/bin/bash
echo "Bonjour"
mkdir projet
ls -l
```

1. Donner le droit d'exécution → `chmod +x monscript.sh`
2. Lancer le script → `./monscript.sh`



# Scripts Windows

Un script est un fichier texte qui regroupe plusieurs commandes pour les exécuter automatiquement.

Sous windows, le fichier prend l'extension .bat, commence souvent par @echo off

```
@echo off  
echo Bonjour  
mkdir projet  
dir
```

- Double-clic sur le fichier .bat
- Ou exécuter via l'invite de commandes → monscript.bat

# Structure d'un script



- Commentaires (#, REM)
- Variables
- Conditions (if)
- Boucles (for)

# Structure d'un script: Commentaires



Un commentaires est une ligne dans le script qui ne sera pas exécutée. Elle sert simplement d'information pour l'humain qui lit le script.

- Unix : # Exemple de commentaire
- Windows : REM Exemple de commentaire



# Structure d'un script: Variables

Une variable sert à stocker une valeur dans le script.

- Unix : `NOM="Alice" ; echo "Bonjour $NOM"`
- Windows : `set NOM=Alice & echo Bonjour %NOM%`



# Structure d'un script: Conditions

Permet d'exécuter des commandes seulement si une condition est vraie.

- Unix :

```
if [ -f notes.txt ]; then
    echo "Le fichier existe"
else
    echo "Le fichier n'existe pas"
fi
```

- Windows :

```
if exist notes.txt (
    echo Le fichier existe
) else (
    echo Le fichier n'existe pas
)
```



# Structure d'un script: Boucles

Permet de répéter une action sur plusieurs éléments.

- Unix :

```
for f in *.txt; do  
    echo "Traitement de $f"  
done
```

- Windows :

```
for %%f in (*.txt) do (  
    echo Traitement de %%f  
)
```



# Exercices

Note de cours:

<https://tinyurl.com/2ekzwwnp>

Google form questions:

<https://forms.gle/HgvuSD8fFvnmQLWJg>

Sandbox unix:

<https://sandbox.bio/playgrounds/terminal>