

Systeme d'exploitation

Interpréteurs de commandes



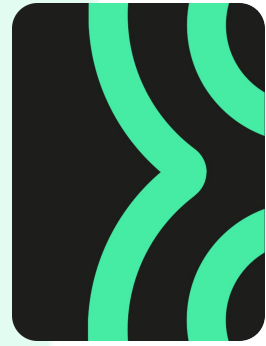
Qu'est-ce qu'un interpréteur de commandes ?



Sommaire

De quoi s'agit-il ?

- 01** Interagir avec l'ordinateur
- 02** GNU bash
- 03** PowerShell



Interagir avec l'ordinateur





Les principales IHM

Les périphériques
d'entrées/sorties

Les **IHM** (Interface Homme Machine) :

Ecran :

- Séquence de texte
- Interface graphique

Clavier :

- Touche => code => traduit (en général) en caractères par l'OS

Souris

- Déplacement et clics



CLI

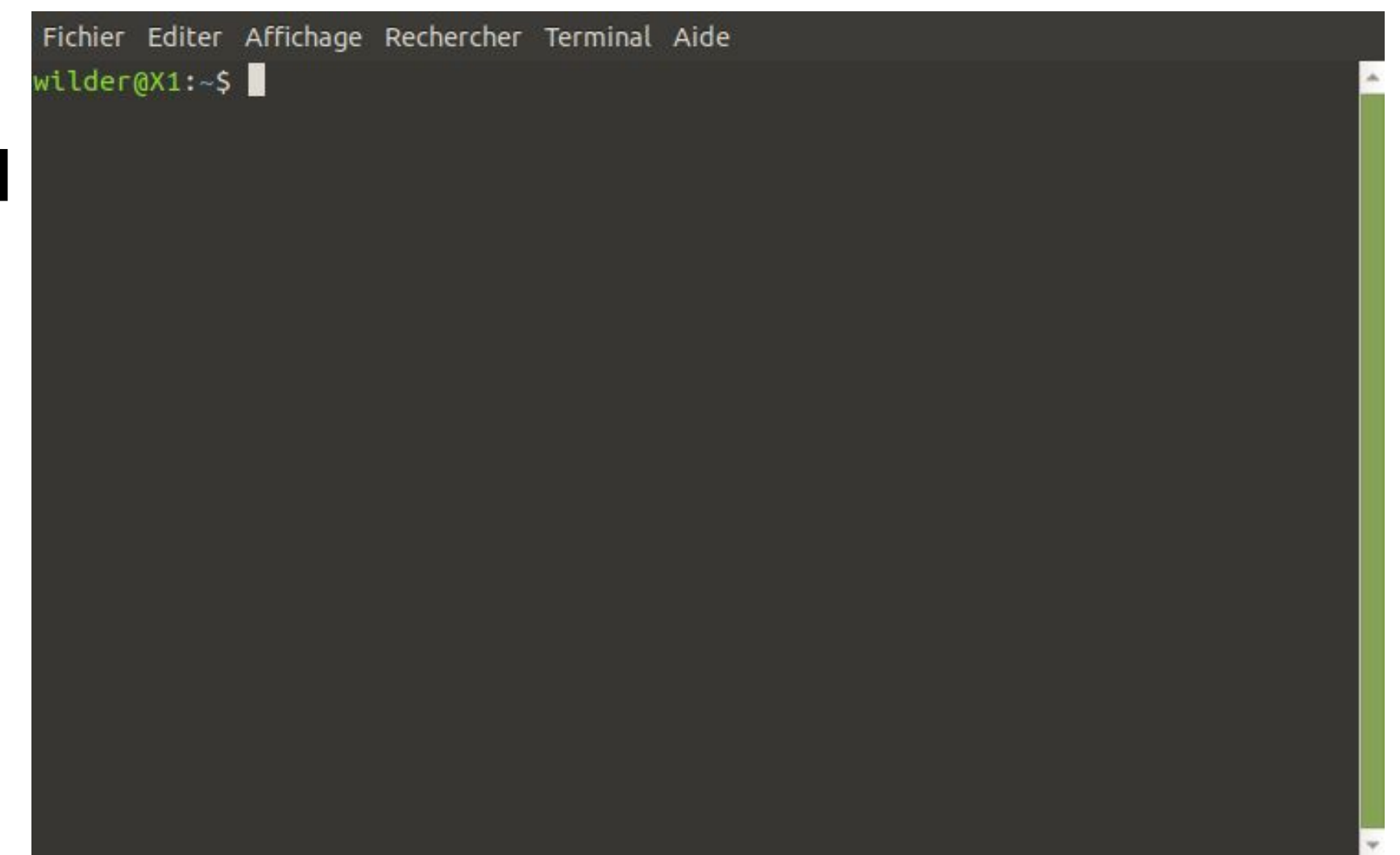


Interface textuelle

Fonctionnement général des interfaces en ligne de commande, ou

CLI (*Command Line Interface*) :

- Invite de commande (**prompt**)
- Taper une commande (clavier) + entrée (↵)
- Affichage du résultat (texte)
- Retour au prompt





Shell : CLI de l'OS

Des shells

MS Windows :

- Invite de commandes **cmd.exe** ou **Powershell** (Windows 7)

Unix :

- Le shell historique : **sh**
- Le classique : **bash**
- Nombreux autres (csh, ksh, tcsh, zsh, dash, ash...)



GUI



Interface
graphique

Pour l'Interface Graphique Utilisateur, ou **GUI** (*Graphical User Interface*) MS Windows :

- Basée sur clavier/souris
- Métaphore du bureau
- Icônes => Fichiers :
 - Programmes
 - Documents (association avec programme par défaut)
- Barres menus (déroulants)



Les plus courantes

Du choix ou pas

MS Windows

- Une interface graphique de MS-DOS => système complet

MacOS

- [Aqua](#)

GNU/Linux - Nombreux composants

- [Système de fenêtrage](#) (X11 ou Wayland)
- [Compositeur](#) (Compiz, Metacity, Mutter)
- [Environnement de bureau](#) (Gnome, KDE, mate, cinnamon, Xfce...)



GNU bash





bash



The GNU shell

Shell du projet GNU et le plus courant
Version 1 (1988) - [Brian Fox](#)
Licence GPLv3





Principales fonctionnalités

Simplifier
l'utilisation du
système

- Historique (rappel des commandes précédentes)
- Édition en ligne (modification d'une commande)
- Auto-complétion
- Variables d'environnements
- Alias
- Gestion des tâches en arrière plan "&"

```
Fichier Editer Affichage Rechercher Terminal Aide
wilder@X1:/home/romain$ bash --version
GNU bash, version 5.0.17(1)-release (x86_64-pc-linux-gnu)
Copyright (C) 2019 Free Software Foundation, Inc.
Licence GPLv3+ : GNU GPL version 3 ou ultérieure <http://gnu.org/licenses/gpl.ht
ml>

Ceci est un logiciel libre ; vous être libre de le modifier et de le redistribue
r.
AUCUNE GARANTIE n'est fournie, dans les limites permises par la loi.
wilder@X1:/home/romain$
```



L'invite de commande ou prompt

Mode interactif

Chaîne de caractères

Par défaut :

- **<nom de login>@<nom d'hôte>:<répertoire courant>**
- suivi de **\$** pour un utilisateur normal et **#** pour root

Configurable : variable **PS1**

```
user@host:~$  
root@host:/#
```



Exécuter une commande



Syntaxe

Syntaxe : **\$ [chemin/]nom_commande [option...] [argument...]**

Séparateur : espace(s)

Sensible à la casse

Méta-caractères (|, >, <, ...)

Mots clés réservés (if, else, function, do...)

En général :

- Options courtes précédées de -
- Options longues précédées de -- (syntaxe GNU)



RTFM

Read The F***
Manual

RTFM pour *Read The Fine Manual*.

La première commande à connaître : **man**

- Affiche la page de manuel
- Disponible pour chaque commande/programme



Quelques exemples de commandes internes

Builtin commands

cd : changer de répertoire

type : afficher le type d'une commande

echo : afficher une valeur

Plus d'exemples : [doc bash](#)

```
wilder@X1:~$ type echo
echo est une primitive du shell
wilder@X1:~$ type cd
cd est une primitive du shell
wilder@X1:~$ type type
type est une primitive du shell
wilder@X1:~$ type mkdir
mkdir est /usr/bin/mkdir
```




Exécuter des suites de commandes

Une à la fois

- Exécution puis retour au prompt : **commande <entrée>**
 - Séquence de commandes : **commande1;commande2 <entrée>**
 - Exécution conditionnelle :
 - **commande1 && commande2 <entrée>**
 - **commande1 || commande2 <entrée> (échec)**
- <ctrl> + <c> : arrêt de la commande en cours (SIGTERM)



Exemples d'exécution de suites de commandes

Une à la fois

Créer un dossier, s'y rendre, créer un fichier vide :

Avec “;” :

```
mkdir ~/dossier1; cd dossier1 ; touch file1
```

Avec “&&” :

```
mkdir ~/dossier2 && cd dossier2 && touch file2
```

Avec “||” :

```
cd dossier3 || mkdir ~/dossier3 && cd dossier3 && touch file3
```



Commande asynchrone

Exécution en
arrière plan

- Exécution dans un sous shell en arrière plan :
 - **\$ commande & <entrée>**
- Retour au prompt immédiat

```
wilder@X1:~$ pluma &
[1] 217299
wilder@X1:~$ ps
  PID TTY          TIME CMD
 209697 pts/0    00:00:00 bash
 217299 pts/0    00:00:00 pluma
 217306 pts/0    00:00:00 ps
wilder@X1:~$
[1]+  Fini              pluma
wilder@X1:~$
```



Sortie standard et sortie d'erreur

Deux pour le prix
d'une

La **sortie standard (stdout)**, représentée par **1** est généralement affichée à l'écran.

La **sortie d'erreur standard (stderr)**, représentée par **2** est également affichée à l'écran, mais elle est traitée séparément de la sortie standard.



Redirections standards

Entrée et sortie
standard

- Redirection sortie standard : **commande > fichier**
- Redirection sortie standard (ajout) : **commande >> fichier**
- Redirection entrée standard : **commande < fichier**
- Redirection sortie erreur (ajout) : **commande 2> (ou 2>>) fichier**



Exemples de redirections standards

Entrée et sortie
standard

Redirige la sortie standard vers le fichier files.txt :

```
find /home/wilder/Documents/ -name "*.txt" > files.txt
```

Redirige contenu de la commande (+ erreur) dans 2 fichiers :

```
find /home/wilder/Documents/ -name "*.txt" > files.txt 2> erreurs.txt
```

```
find /home/wilder/Document/ -name "*.txt" > files.txt 2> erreurs.txt
```

Redirige la sortie standard et la sortie d'erreur vers 1 fichier :

```
find /home/wilder/Document/ -name "*.txt" > output.txt 2>&1
```



Interconnecter les commandes



Une histoire de
tuyau

Le symbole “|” est appelé un “pipeline” ou “**pipe**”.

- Connexion de la sortie d'une commande sur l'entrée d'une autre :
commande1 | commande2
- Ajout de la sortie d'erreur : **commande1 |& commande2**



Exemples d'interconnexion de commandes

Une histoire de
tuyau

Filtre le contenu d'un repertoire :

```
ls -l | grep ".txt"
```

```
ls toto.txt |& grep "Aucun fichier"
```

Filtre la liste des processus ;

```
ps aux | grep "chrome"
```

Trie un fichier :

```
cat fichier.txt | sort
```




Commande interne alias

Renommer des
commandes

- Créer un alias
\$ alias nomAlias='cibleAlias'
- Lister les alias
\$ alias
- Supprimer un alias
\$ unalias nomAlias



Les wildcards

Plusieurs d'un
coup

N'importe quel caractère (y compris aucun) : *

Un (seul) caractère quelconque : ?

Un caractère parmi : []

Début d'une chaîne de caractères : ^

liste : **[abc]**

suite : **[a-z]**

Combinaisons possible : **a[bc]***



Les wildcards (suite)

Plusieurs d'un
coup

```
wilder@X1:~$ for i in {1..5};do mktemp aaa_XXXXXX;mktemp bbb_XXXXXX;done
wilder@X1:~$ ls
wilder@X1:~$ ls aaa_*
wilder@X1:~$ ls a*
wilder@X1:~$ mkdir rep_OLD && cp b* rep_OLD
wilder@X1:~$ rename 's/^/_OK_/' *
wilder@X1:~$ rename 's/^(_OK_bbb)(.*)$/SAVE_$1$2/' _OK_bbb*
```



Commande interne echo

Affichage et arguments

echo argument [argument...]

```
wilder@X1:~$ echo bonjour
bonjour
wilder@X1:~$ echo salut tout le monde
salut tout le monde
wilder@X1:~$ echo salut      tout      le      monde
salut tout le monde
wilder@X1:~$ echo 'salut    à    tous'
salut    à    tous
wilder@X1:~$ echo "salut    à    tous"
salut    à    tous
wilder@X1:~$
```



Références :

[La doc officielle](#)

Trouver de l'aide :

Sur StackExchange

- [Server Fault](#) : Q&A administration systèmes et réseaux
- [Unix & Linux](#) : Q&A sur les Unix-like
- [Ask Ubuntu](#) : Q&A centré sur Ubuntu et dérivées
- [Super User](#) : Q&A pour power users

Conseils :

- Cherchez d'abord
- Rappelez-vous : personne ne vous **doit** de l'aide



PowerShell





Windows PowerShell

The new Windows
scripting shell

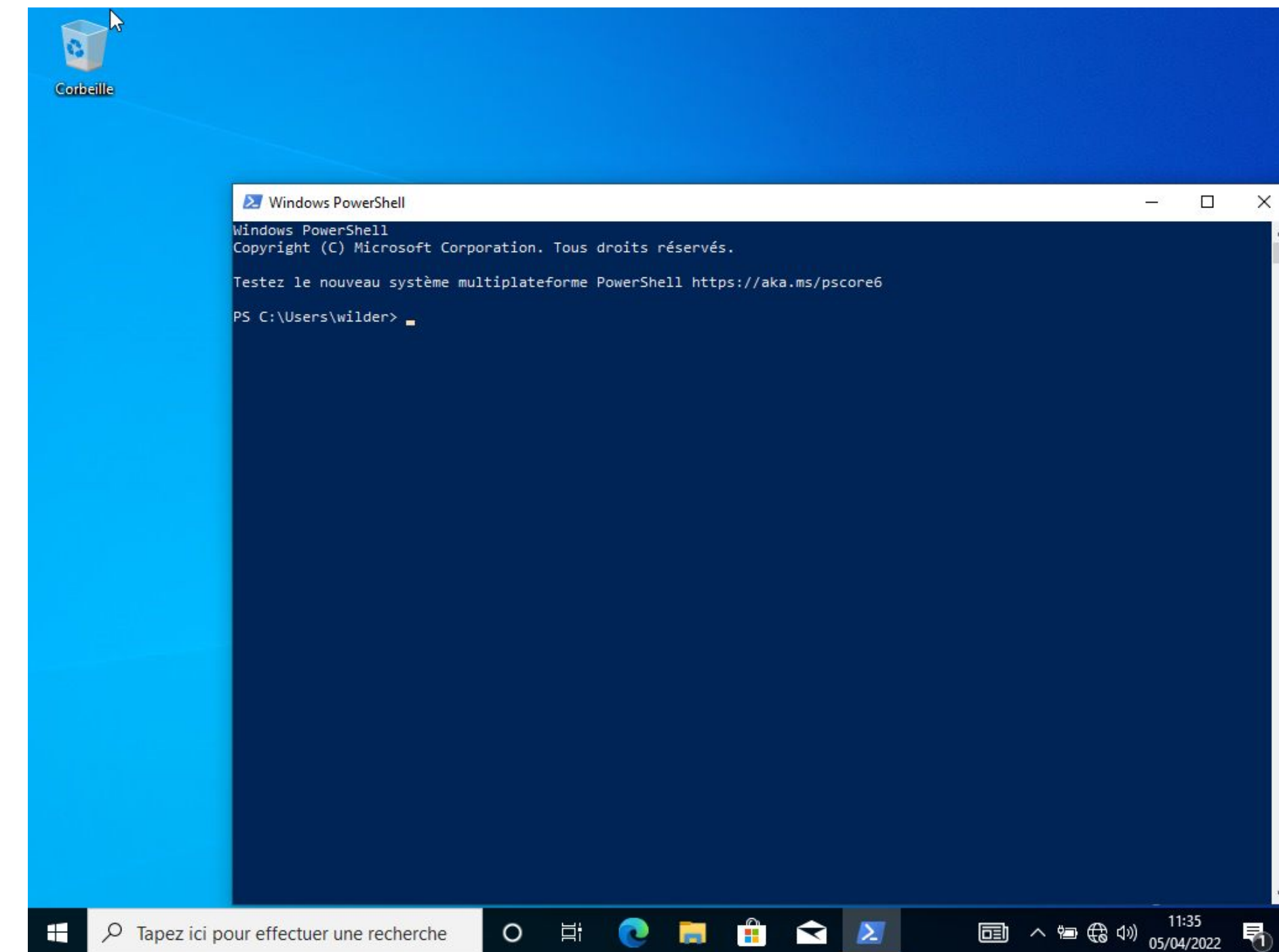
Nouveau shell depuis Windows 7
Successeur de command.com et cmd.exe
Possibilité de scripts avancés



Principales fonctionnalités

Simplifier
l'utilisation du
système

- Historique (rappel des commandes précédentes)
- Édition en ligne (modification d'une commande)
- Auto-complétion
- Variables d'environnements
- Alias
- Similarité avec shell unix





Le prompt

Mode interactif

Chaîne de caractères

Par défaut :

- **PS <répertoire courant> >**
- Pas de distinction administrateur

Configurable : [fonction prompt](#)

```
PS C:\Users\wilder>
```



Différent type de commandes



Mode interactif

- Programmes (application)
- Applets de commandes (**Cmdlet**)
- Fonction (function)
- Scripts - selon la politique d'exécution (**Set-ExecutionPolicy**)



Connecter les entrées/sorties



Mode interactif

Mécanisme de [pipe](#) similaire
commande1 | commande2

Mécanisme de [redirection](#) similaire
commande1 > commande2



Les wildcards



Substitutions
automatiques

Similaires

Pour plus de détail : voir [la doc](#)



Références :
[La doc officielle](#)



En résumé

A retenir

Notion d'interpréteur de commandes

Découverte de bash

Découverte de PowerShell



MERCI

pour votre participation.

C'est à vous maintenant.

Des questions ?

Des remarques ?



Atelier : Modifie ton shell

