



# ADRAR\o/ PÔLE NUMÉRIQUE

- > **INFRASTRUCTURES** SYSTÈMES & RÉSEAUX
- > **CYBERSÉCURITÉ** INFRASTRUCTURES & APPLICATIONS
- > **DEVOPS / SCRIPTING** & AUTOMATISATION
- > **DEVELOPPEMENT** WEB & MOBILE
- > TRANSFORMATION NUMERIQUE DES ENTREPRISES

[www.adrar-numerique.com](http://www.adrar-numerique.com)

## Introduction

### Shell

Dans ce mini cours, nous allons voir les quelques commandes et actions de base sur un shell.

Ce sont les commandes qui vous seront le plus utiles lors de l'utilisation de vos machines. Le terminal permet d'utiliser de nombreux outils sur votre machine, quel que soit le système d'exploitation (OS) que vous utilisez.

```
ls [-a -l|-al]
```

Liste les fichiers et dossiers du répertoire courant

L'option **-a** affiche les fichiers et dossiers cachés

L'option **-l** affiche les informations détaillées pour chaque fichier et dossier

```
cd {repertoire}
```

Permet de se déplacer dans le répertoire désiré

```
mkdir {repertoire}
```

Permet de créer un dossier

```
touch {fichier}
```

Permet de créer un fichier

```
rm [-r -f|-rf] {repertoire}
```

Permet de supprimer un élément

L'option **-r** signifie *recursive*

L'option **-f** signifie *force*

```
cp [-r] {aCopier} {aColler}
```

Permet de copier un dossier/dossier d'une destination à une autre

L'option **-r** permet de copier tous les sous-dossiers également

**NB: Commencer avec "/" pour cibler un élément à la racine du serveur.**

```
pwd
```

Affiche le chemin complet vers le répertoire courant

```
echo {chaîne}
```

Affiche la chaîne de caractère passée en sortie

```
cat {fichier}
```

Affiche le contenu du fichier passé en paramètre

```
grep {recherche} {chemin/fichier}
```

Recherche le texte **recherche** dans le(s) fichier(s) passé(s)

```
clear
```

Permet d'effacer le contenu de la console

```
history [-c]
```

Permet de lister toutes les commandes tapées sur votre instance  
L'option **-c** permet de supprimer l'historique

```
apt update
```

Télécharge les informations sur les paquets à partir des **sources configurées**.

```
apt upgrade
```

Lance la mise à jour des paquets

```
apt install {paquet[,paquet2,...]}
```

Installe le(s) paquet(s) passé(s) en paramètre et gère les dépendances



Il est possible de combiner les commandes en utilisant une double esperluette:

```
mkdir dossier && cd dossier && touch test.txt
```

La commande va créer le répertoire “dossier”, se rendre dedans et créer un fichier “test.txt”

Il est possible d’insérer une sortie dans un fichier:

```
echo “test” > test.txt
```

Viendra ÉCRASER le contenu du fichier test.txt avec le contenu “test”

```
echo “test” >> test.txt
```

Viendra AJOUTER le contenu “test” à la fin du fichier test.txt

Exemple complet:

```
touch index.html && echo “Hello world” >> index.html
```

Il vous est possible de créer des scripts afin d'effectuer des tâches récurrentes par exemple ou des tâches qui suivent un *pattern* répétitif  
Toutes les commandes du SHELL sont utilisables dans un script

```
#!/bin/bash  
  
echo "hello"  
pwd
```

Pour le lancer vous taperez:

```
sh script.sh
```

OU

```
./script.sh
```

Pour utiliser la seconde option, vous devez avoir les droits d'exécution, si ce n'est pas le cas:

```
chmod +x test.sh
```

La première ligne se nomme un **shebang**. Il donne juste le chemin complet vers le shell afin d'être localisable partout où le script pourrait être utilisé.

#!/bin/bash et #!/bin/sh sont les plus communs

Ce script affichera le message "hello" et le répertoire courant.

## Script bash dynamique Shell

Pour utiliser un élément passé en paramètre du script:

```
#!/bin/bash
```

```
mkdir $1
```

```
sh script.sh test
```

Cette commande créera le répertoire test dans le dossier actuel

## Exercice Shell

Construisez un script pour créer l'arborescence suivante dans le dossier  
/var/www/html/:

```
| projet/  
|   index.html  
| css/  
|   style.css  
| images/  
|   .gitKeep
```

Si ce script fonctionne, essayez de rendre le nom du projet dynamique à partir d'une saisie utilisateur.