

# Cours : Shell Scripting et Automatisation

---

## □ 1. Qu'est-ce qu'un Shell Script ?

- Un **Shell script** est un **fichier texte** contenant une série de **commandes Bash**.
  - Il est interprété par le **Shell**, généralement Bash sous Linux.
  - But : **automatiser des tâches répétitives** (sauvegardes, installation, surveillance, etc.).
- 

## ■ 2. Structure d'un Script Bash

*Exemple simple :*

```
#!/bin/bash
echo "Bonjour, $USER !"
date
```

*Explication :*

- `#!/bin/bash` : indique l'interpréteur (appelé *shebang*).
  - `echo` : affiche un message.
  - `$USER` : variable d'environnement (nom de l'utilisateur).
  - `date` : affiche la date actuelle.
- 

## 3. Créer et exécuter un script

*Étapes :*

1. Créer un fichier :
  2. `nano mon_script.sh`
  3. Rendre exécutable :
  4. `chmod +x mon_script.sh`
  5. Lancer le script :
  6. `./mon_script.sh`
- 

## □ 4. Variables dans les scripts

*Déclaration :*

```
nom="Alice"
echo "Bonjour $nom"
```

*Variables système utiles :*

Variable	Description
\$USER	Nom de l'utilisateur
\$HOME	Répertoire personnel
\$PWD	Répertoire courant
\$0, \$1, \$2...	Paramètres de script

---

## 🔗 5. Structures de contrôle

*Condition if:*

```
if [ $1 -gt 10 ]; then
    echo "Supérieur à 10"
else
    echo "Inférieur ou égal à 10"
fi
```

*Boucle for:*

```
for i in 1 2 3; do
    echo "Nombre : $i"
done
```

*Boucle while:*

```
i=1
while [ $i -le 3 ]; do
    echo "Itération $i"
    i=$((i+1))
done
```

---

## 📖 6. Fonctions dans un script

```
bonjour() {
    echo "Bienvenue $1"
}
```

```
bonjour "Alice"
```

---

## 🔧 7. Outils utiles dans les scripts

Commande	Utilité
date	Affiche la date/heure

Commande	Utilité
cut, awk	Traitement de texte
grep	Recherche dans du texte
tar, gzip	Sauvegarde et compression
crontab	Planification de scripts

---

## ❑ 8. Cas pratique : script de sauvegarde

```
#!/bin/bash

src="/home/user/Documents"
dest="/home/user/backup_$(date +%F).tar.gz"

tar -czf $dest $src
echo "Sauvegarde réalisée vers : $dest"
```

---

## ❑ 9. Automatiser avec cron

- cron permet d'exécuter des scripts à intervalle régulier.

```
crontab -e

Exemple de tâche cron (tous les jours à 20h) :
0 20 * * * /home/user/mon_script.sh
```

---

## ✔ 10. Bonnes pratiques

- Toujours ajouter `#!/bin/bash` en haut.
  - Donner des noms clairs aux variables.
  - Ajouter des commentaires avec `#`.
  - Tester vos scripts avec des entrées inattendues.
  - Utiliser `set -e` pour arrêter à la moindre erreur.
- 

## ★ Résumé des commandes-clés

Élément	Commande/Exemple
Créer script	<code>nano script.sh</code>

Élément	Commande/Exemple
Rendre exécutable	<code>chmod +x script.sh</code>
Exécuter	<code>./script.sh</code>
Variable	<code>nom="Alice"</code>
Condition	<code>if [ ... ]; then ... fi</code>
Boucle	<code>for, while</code>
Cron	<code>crontab -e</code>