

# Développement de Script Shell

**ANDRIANARISON Andry Nirina**

- Script Shell : Programmes exécutables écrits en shell (`bash`, `sh`, `zsh`, ...).
- But : Automatisation des tâches, déploiement, gestion de configuration.

# Syntaxe de base

- Shebang : `#!/bin/bash` pour spécifier l'interpréteur.
- Commandes : `echo` , `cd` , `ls` , `cp` , `mv` , `rm` , ... .
- Variables : `VAR=value` , `$VAR` , `${VAR}` .

## Variables d'environnement

- Définition : `export VAR=value` .
- Accès : `$VAR` OU `${VAR}` .
- Portée : Locale, globale, d'exportation.  
→ Exemple : `hostname` (Variable globale), `VAR="Bonjour"` (Variable locale)

# Conditions

- If :

```
if [ condition ];  
then  
    command;  
fi
```

- Exemple :

```
# Verification si on est en root  
idNumber=$(id -u) # prendre la valeur de l'id  
if [[ $idNumber -ne 0 ]]  
then  
    echo "→ Executer en tant que root"  
    exit 1  
fi
```

# Boucles

- For :

```
for i in {1..5};  
do  
    echo $i;  
done
```

- While :

```
while [ condition ];  
do  
    command;  
done
```

# Fonctions

- Définition :

```
function_name()  
{  
    commands;  
}
```

- Appel : `function_name` .
- Retour de valeur : `return value` .

## Gestion des erreurs

- Exit status : `$?` .
- Trap : `trap 'commands' ERR` .
- Set : `set -e` , `set -u` , `set -o pipefail` .

- Exemple :

```
#!/bin/bash
read -p "What's your name? " NAME
echo "Hello, $NAME"

if [ -e ~/.bashrc ];
then
    echo "Le fichier de configuration de bash existe"
fi

for i in {1..5};
do
    echo "→ $i"; # Sortie : 1, 2, 3, 4, 5
done

function pingGoogle(){
    ping -c 4 8.8.8.8
}

pingGoogle
```

# Conclusion

- Meilleures pratiques : Lecture, débogage, portabilité.
- Ressources :
  - Man pages ( `man bash` )
  - [Letsdefend](#)
  - Jordan Assoulin : Livre `Scripting avancé` , `Shell scripting`