

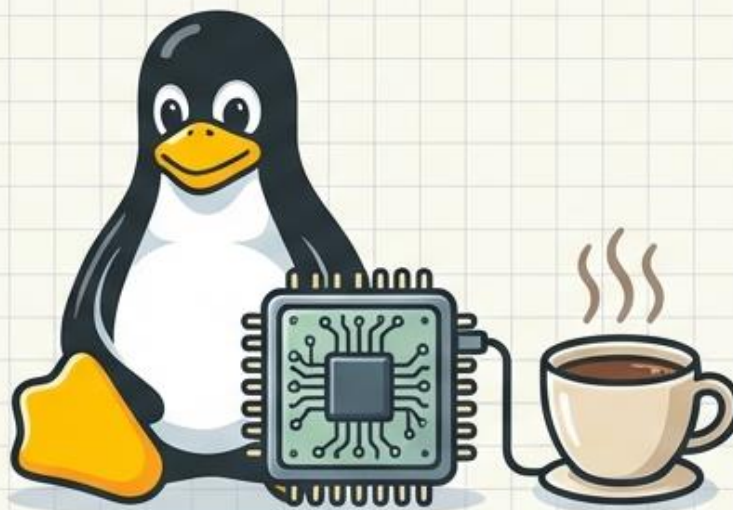
# Prêts à automatiser ? Le pouvoir du Shell Scripting

## Manuel de Bord de l'Administrateur Système - Niveau 1



### Ce que vous allez maîtriser :

- L'automatisation : Gain de temps et puissance
- La création de scripts .sh
- La gestion des variables, des maths et des conditions



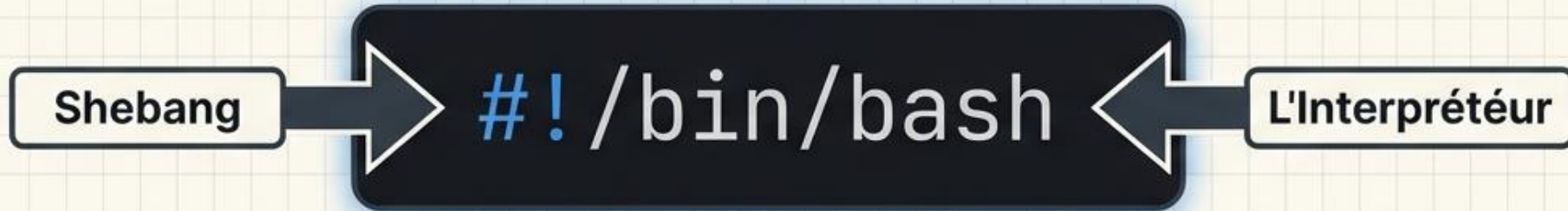
Un script shell est simplement une série de commandes exécutées par l'interpréteur. Si vous savez taper une commande, vous savez scripter !



Bienvenue dans ce parcours pratique. L'objectif est de manipuler le Shell pour transformer votre terminal en centre de commande.

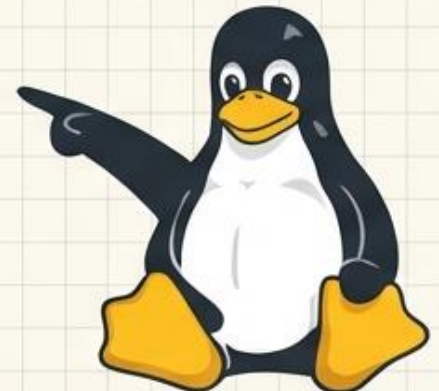
# L'Anatomie d'un Script & Le Rituel de Lancement

## The Brain



Indique au système d'utiliser Bash pour exécuter le fichier.

## The Workflow





# MISSION 1 : HELLO WORLD (PREMIER CONTACT)

Niveau : Débutant

## Briefing

### Objectif

Créer votre tout premier script pour afficher un message simple dans la console.

### Instructions

1. Ouvrez votre éditeur de texte (ex: nano).
2. Créez un fichier nommé 'hello\_world.sh'.
3. Écrivez le code ci-contre.
4. Rendez le fichier exécutable (chmod +x).
5. Lancez le script (./hello\_world.sh).

```
#!/bin/bash

echo "Bonjour, monde !"
```

### Console Output

Bonjour, monde !

Hello!



# MISSION 2 : INTERACTION UTILISATEUR

Niveau : Débutant

## Briefing

### Objectif

Modifier le script pour le rendre interactif. Le script doit poser une question et attendre une réponse.

### Indice Clé

Utilisez la commande 'read' pour capturer la saisie clavier.

```
Terminal Window

echo "Quel est votre nom ?"
read nom
echo "Bonjour, $nom !"
```

## The Flow

### Script

Quel est votre nom ?

### Utilisateur

Thomas

### Script

Bonjour, Thomas !





# MISSION 3 : IDENTITÉ & VARIABLES

Niveau : Intermédiaire

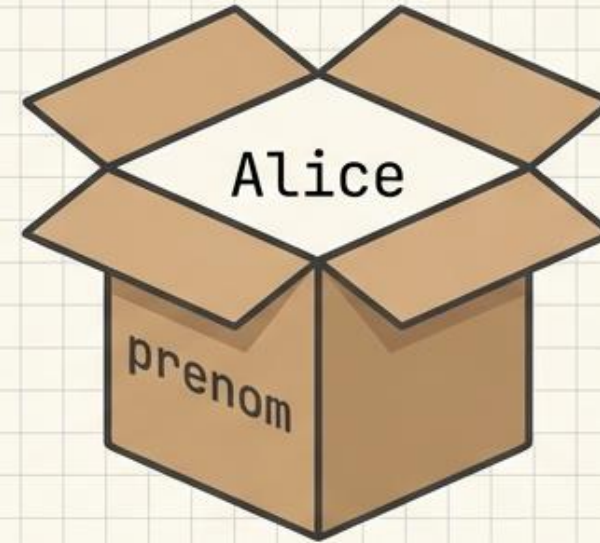
## Briefing

### Objectif

Créer un script "variables.sh" qui stocke des données en mémoire.

### Instructions

1. Définir : "prenom" ("Alice"), "age" (30), "ville" ("Paris").
2. Afficher une phrase complète avec "echo" et "\$".



### ATTENTION !

Pas d'espaces autour du signe "=".

**Good:** "var=valeur" ✓

**Bad:** "var = valeur" ✗

Console Output

```
Je suis Alice, j'ai 30 ans et je viens de Paris.
```



# MISSION 4 : LE PIÈGE DES CITATIONS (QUOTES)

Niveau : Intermédiaire

**Objectif :** Comprendre la différence cruciale entre ' ' et " ".

## Simple Citation ( ' ' )



**Littéral.**  
Rien n'est interprété.

```
Terminal
> echo 'Hello $nom'
> outputs $nom
```

## Double Citation ( " " )



**Interprété.**  
Les variables (\$) sont  
remplacées par leur  
valeur.

```
Terminal
> echo "Hello $nom"
> outputs Alice
```

**Règle d'Or :** Utilisez ' ' pour protéger le texte brut,  
et " " pour laisser passer les variables.





# MISSION 5 : CALCULATRICE SHELL

Niveau : Intermédiaire

Définir les variables :



```
a=10
```



```
b=3
```

```
resultat=$(( a + b ))
```



Somme

13



Différence

7



Produit

30



Quotient

3



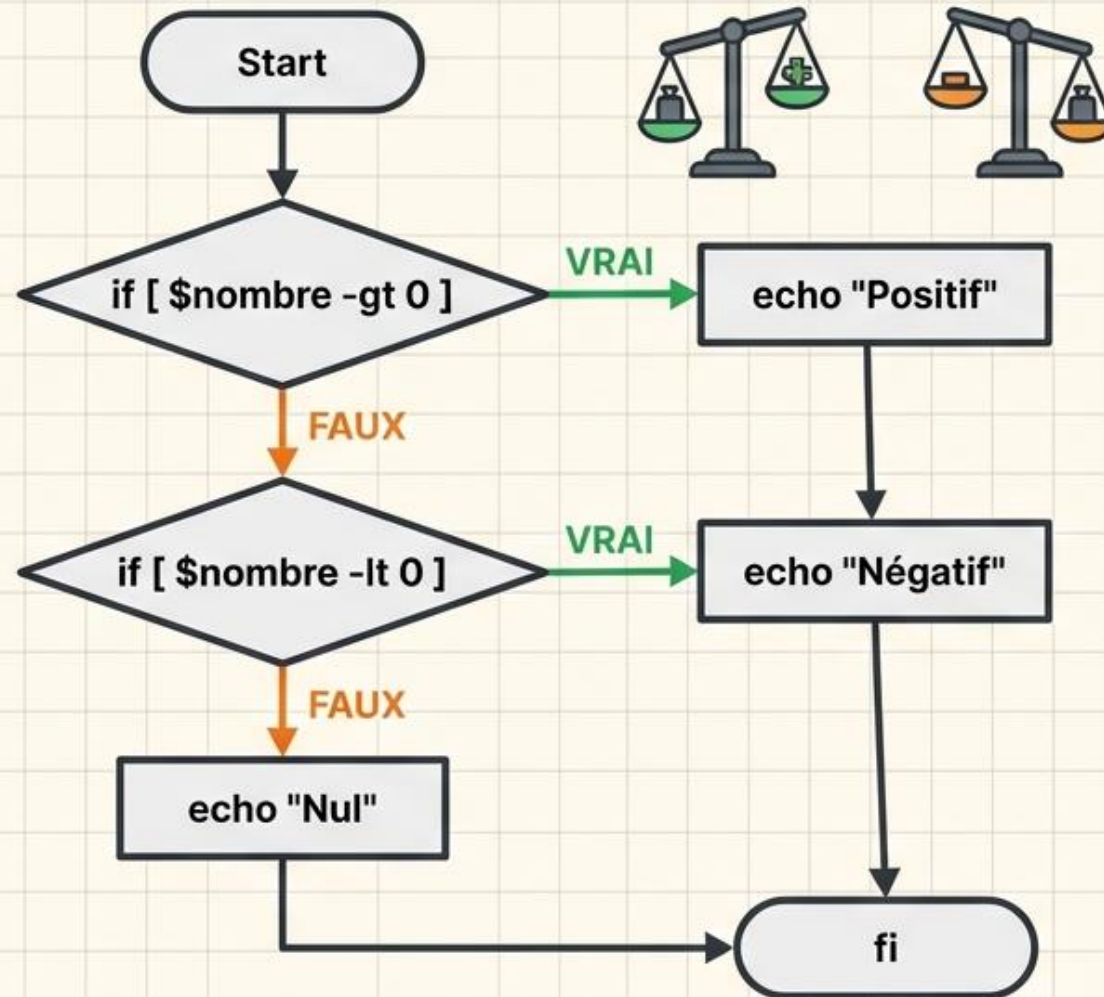
Reste/Modulo

1



# MISSION 6 : LE TRIEUR (CONDITIONS IF/ELSE)

Niveau : Avancé



```
if [ $nombre -gt 0 ]; then
    echo "Positif"
elif [ $nombre -lt 0 ]; then
    echo "Négatif"
else
    echo "Nul"
fi
```





# MISSION 7 : LE DUEL (COMPARAISON)

Niveau : **Avancé**



**Objectif :** Comparer deux nombres (n1 et n2) et afficher le vainqueur.

## Mémento des Opérateurs

Opérateur	Signification
-eq	Equal (Égal)
-ne	Not Equal (Différent)
-gt	Greater Than (Plus grand)
-lt	Less Than (Plus petit)

```
if [ $n1 -gt $n2 ]; then
    echo "$n1 est plus grand"
fi
```

Ces codes sont spécifiques  
aux nombres !



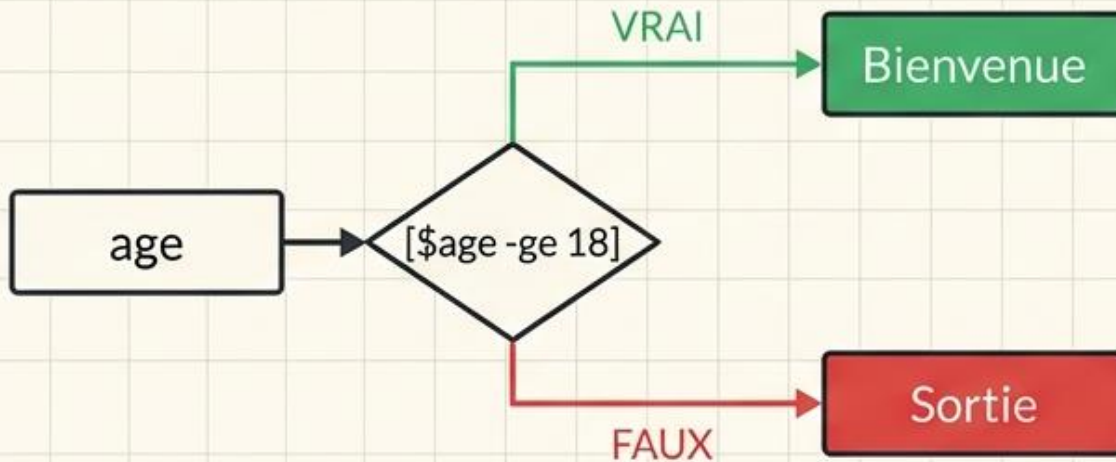
# MISSION 8 : LE VIDEUR (SÉCURITÉ)

Niveau : Avancé



## Scénario :

1. Le script demande : "Quel est votre âge ?"
2. Si l'âge est  $\geq 18$ , afficher "Accès Autorisé".
3. Sinon, afficher "Accès Refusé".



```
if [ $age -ge 18 ]; then
    echo "Accès Autorisé"
else
    echo "Accès Refusé"
fi
```



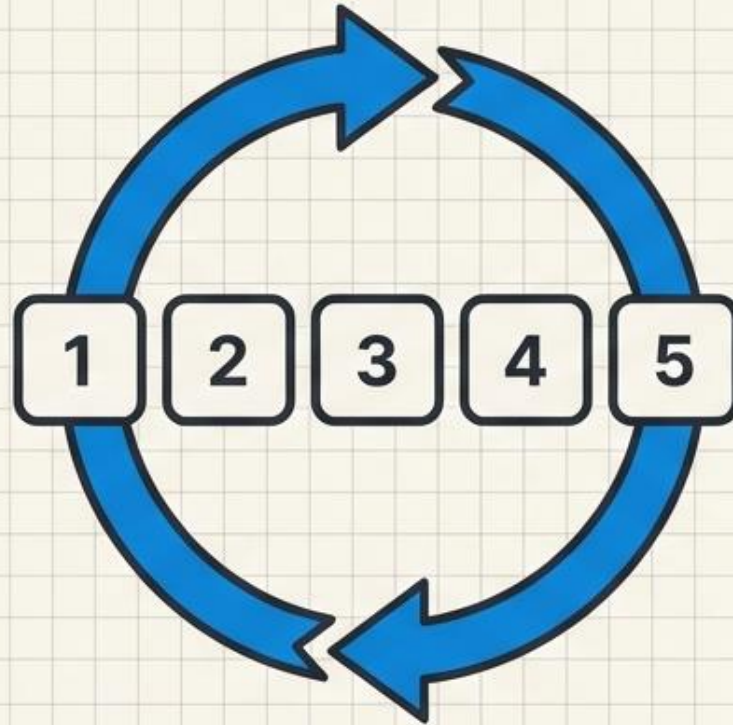


## MISSION 9 : L'ÉCHAUFFEMENT (BOUCLES FOR)

Niveau : Maîtrise



Écrivez un script qui  
affiche les nombres de  
1 à 5  
séquentiellement.



Pour chaque élément de  
la liste...

Terminal Window

```
for i in 1 2 3 4 5
do
    echo "Numéro : $i"
done
```

Output Visual

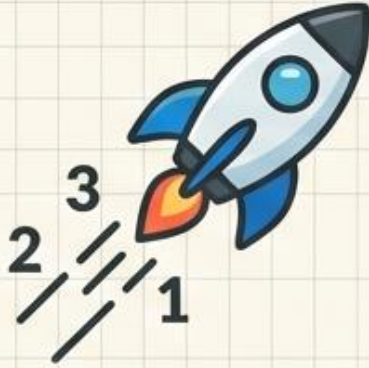
```
Numéro : 1
Numéro : 2
Numéro : 3
Numéro : 4
Numéro : 5
```



# MISSION 10 : COMPTE À REBOURS (BOUCLES WHILE)

Niveau : Maîtrise

## Concept



Tant que la condition est vraie... répéter.

## Scénario

Partir d'un nombre (ex: 10) et descendre jusqu'à 0.

```
Terminal Window

compteur=10

while [ $compteur -ge 0 ]; do
    echo $compteur
    compteur=$((compteur - 1))
done

echo "Décollage !"
```



**Attention aux boucles infinies !**  
La condition doit devenir fausse.



# MÉMENTO DU COMMANDANT



**Vous avez maintenant les clés du système.  
À vous d'automatiser le monde.**