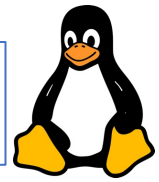


# LES SCRIPTS LINUX



**PARTIE 2**

# VARIABLES ET TYPES DE DONNÉES



## Déclaration et Utilisation des Variables :

- **Déclaration:** En Bash, une variable est créée dès que vous lui attribuez une valeur. Exemple : `nom="Alice"`.
- **Utilisation:** Pour utiliser la valeur d'une variable, ajoutez `$` avant son nom. Exemple : `echo $nom`.

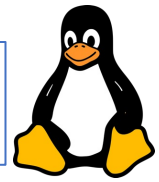
## Types de Données :

- **Chaînes de caractères:** Toutes les données sont traitées comme des chaînes de caractères par défaut.
- **Nombres:** Bien que Bash traite les nombres comme des chaînes, il permet des opérations arithmétiques de base.
- **Tableaux:** Bash supporte les tableaux indexés et associatifs.

## Citations : Simple vs Double

- **Simple citation ('):** Tout ce qui est entre simples citations est traité littéralement.
- **Double citation ("):** Les variables et certains caractères spéciaux sont interprétés.

# VARIABLES ET TYPES DE DONNÉES



## Exercices Pratiques

### Exercice 1: Manipulation de Variables

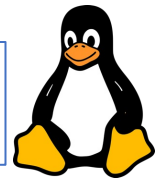
- Créer un script variables.sh :

```
#!/bin/bash
prenom="Alice"
age=30
ville='Paris'
```

```
echo "Je suis $prenom, j'ai $age ans et je viens de $ville."
```

- Exécutez ./variables.sh.
- Notez comment les variables sont remplacées par leurs valeurs.

# CONCEPTS DE BASE DES SCRIPTS SHELL



## Exercices Pratiques

### Exercice 2: Utilisation de Citations

- Modifier variables.sh :

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

```
item='pomme'
```

```
echo 'J'\''aime les $item.'
```

```
citation
```

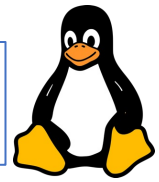
```
echo "J'aime les $item."
```

```
# Utilisation de l'échappement et simple
```

```
# Utilisation de double citation
```

- Exécutez ./variables.sh.
- Comparez les différences dans le traitement des variables entre simples et doubles citations.

# CONCEPTS DE BASE DES SCRIPTS SHELL



## Exercices Pratiques

### Exercice 3: Opérations Basiques sur les Nombres

- Créer un script math.sh :

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

```
a=10
```

```
b=3
```

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

```
difference=$((a - b))
```

```
produit=$((a * b))
```

```
quotient=$((a / b))
```

```
reste=$((a % b))
```

```
echo "Somme: $somme, Différence: $difference, Produit: $produit, Quotient:  
$quotient, Reste: $reste"
```

- Exécutez ./math.sh.
- Observez comment Bash gère les opérations arithmétiques.