



# Les Scripts Linux - Partie 6 : Les Fonctions

Syntaxe, Paramètres, Valeurs de retour et  
Portée des variables.

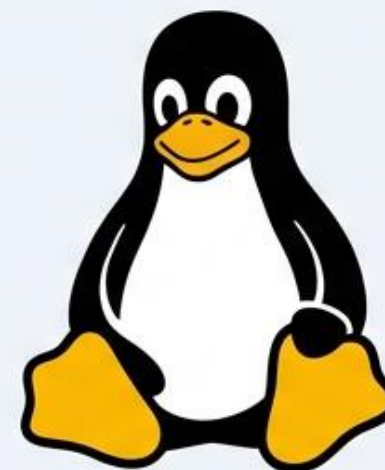
# Déclaration et Syntaxe de base

Une fonction permet de regrouper un bloc de code pour l'exécuter plusieurs fois.

```
nom_de_fonction() {  
    # Code à exécuter  
}
```

Déclaration

Bloc de code



Exemple :

```
dire_bonjour() {  
    echo 'Bonjour tout le monde!'  
}
```

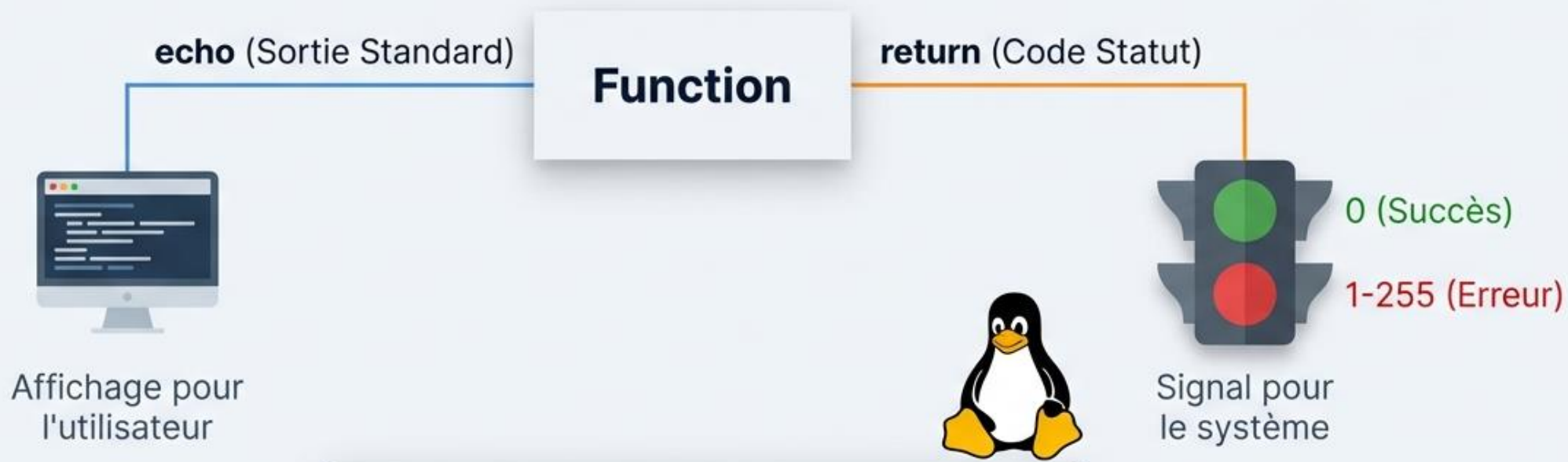
# Appel de Fonctions et Paramètres

1. **Appel** : Pour exécuter, écrire simplement le nom (ex: 'dire\_bonjour').
2. **Paramètres** : Passés à la suite du nom. Accessibles via \$1, \$2, etc.



# Comprendre la Valeur de Retour (Return)

Distinction cruciale entre Sortie Standard et Statut.



Affichage pour l'utilisateur

Signal pour le système

```
Terminal Window
verifier_nombre() {
  if [ $1 -gt 10 ]; then
    return 1 # Erreur
  else
    return 0 # Succès
  fi
}
```

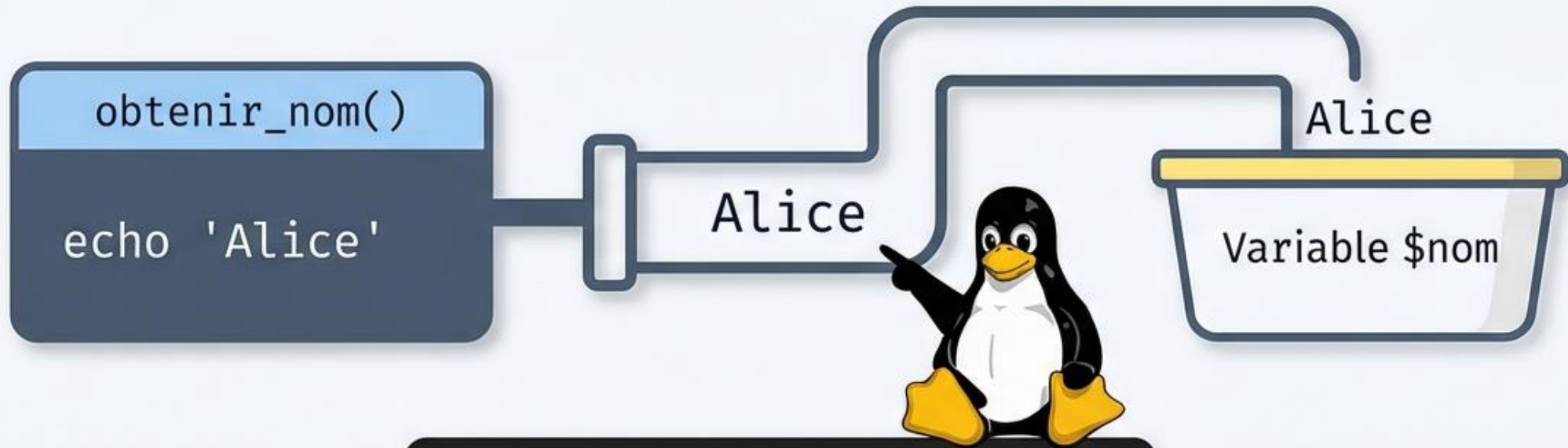
# Capturer le code de retour



- La variable spéciale `$?` stocke le dernier statut.
- Elle doit être lue immédiatement après l'exécution.
- Dans cet exemple :  
Affiche 1 (car  $15 > 10$ ).

# Retourner des Valeurs Complexes

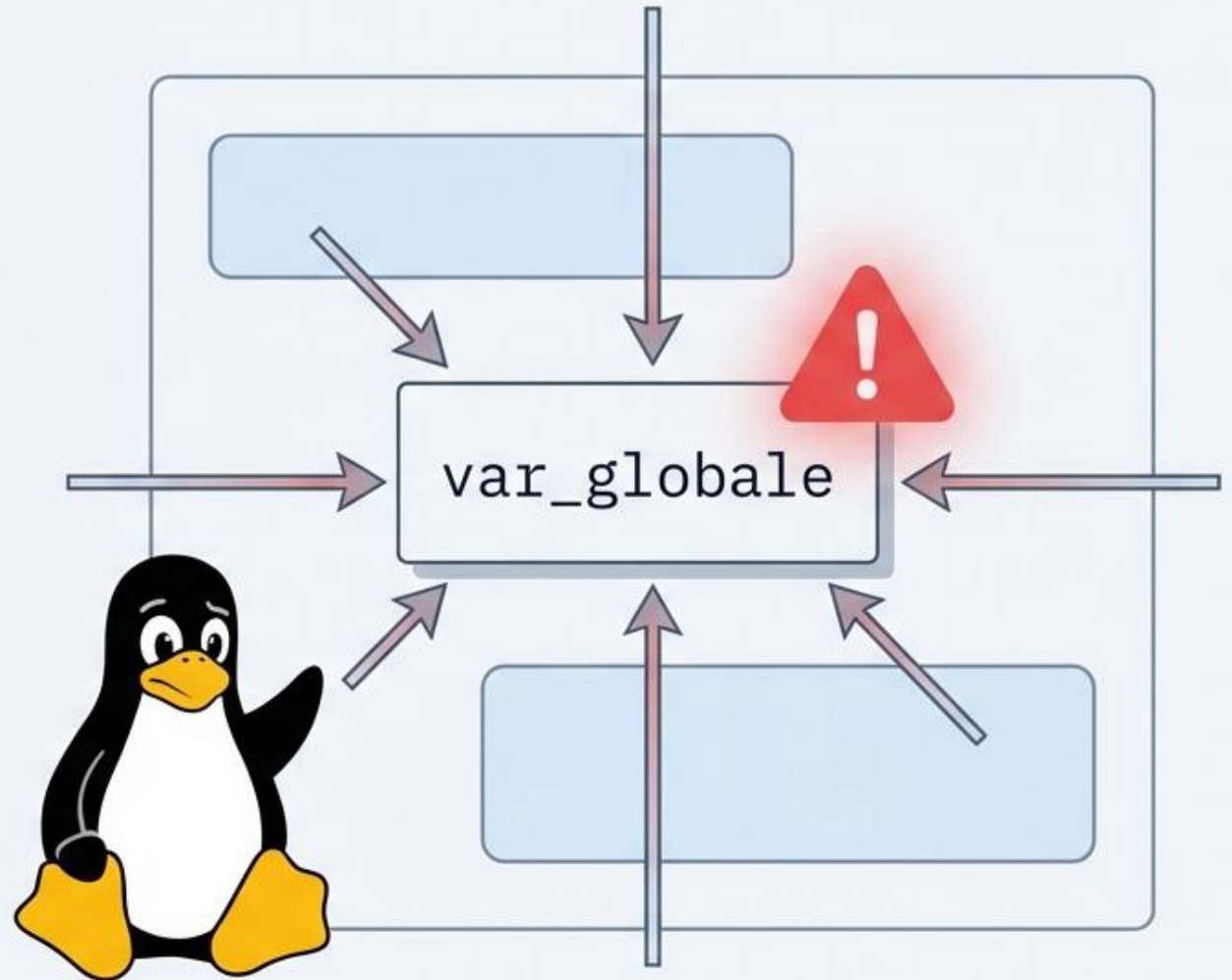
Contourner la limitation des entiers avec echo.



```
nom=$(obtenir_nom)
echo "Le nom est $nom"
```

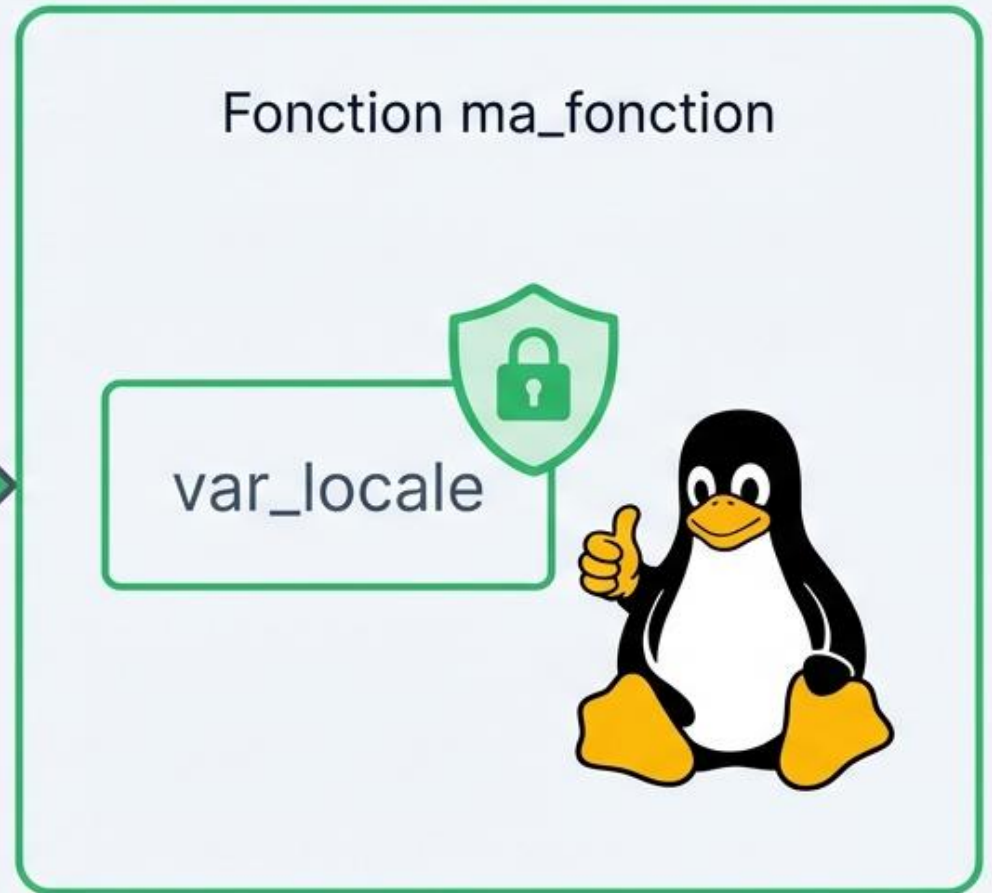
# La Portée Globale des Variables

- DANGER : Accessible partout.
- Par défaut, toute variable est globale.
- Risque de modification accidentelle.



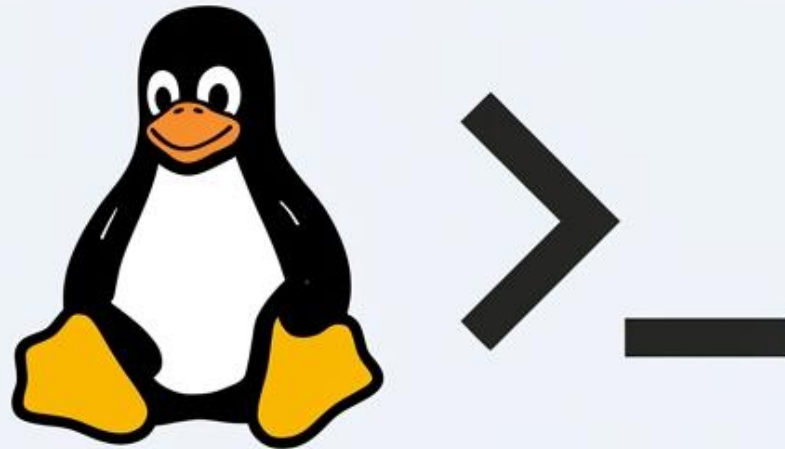
# La Portée Locale des Variables

- SÉCURITÉ : Accessible uniquement dans la fonction.
- Mot-clé : local
- Détruite à la fin de la fonction.



# TRAVAUX PRATIQUES

Mise en application des concepts



Ouvrez votre terminal. Pour chaque exercice, créez un script ou tapez la fonction directement.

# Exercice 1 - Fonction Basique

## The Brief

### Objectif

Écrire une fonction nommée 'saluer'.

### Consigne

La fonction doit afficher 'Bonjour, [nom]!'.

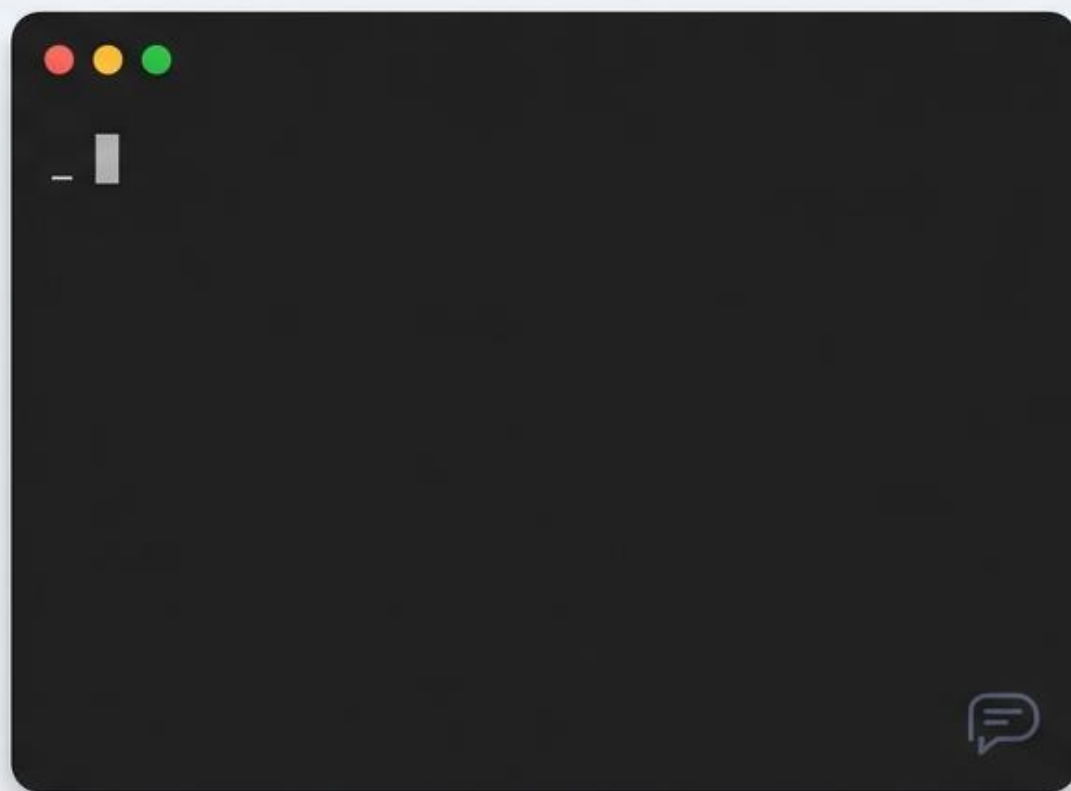
### Entrée

Un nom passé en paramètre.

### Sortie Attendue

Bonjour, Thomas!

## The Canvas



# Exercice 2 - Calcul Simple

## The Brief

### Objectif

Créer une fonction nommée 'ajouter'.

### Consigne

La fonction accepte deux arguments numériques et calcule leur somme.

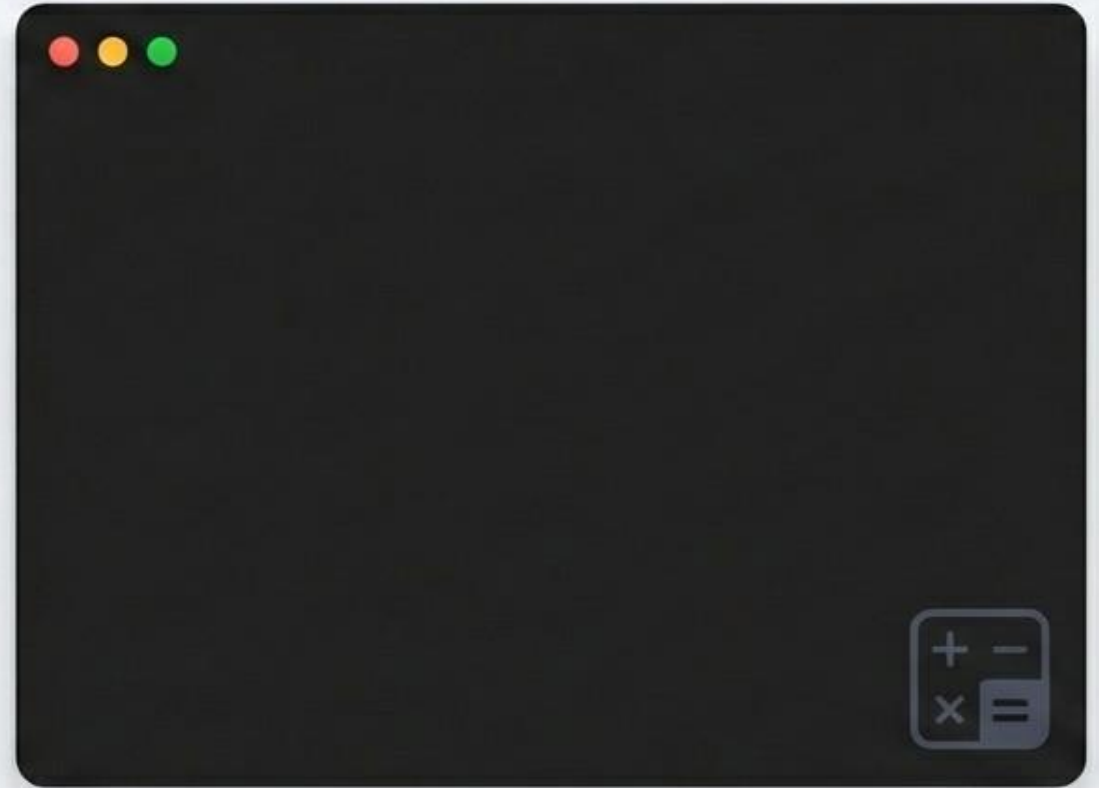
### Entrées

Deux nombres (ex: 10 et 5).

### Sortie Attendue

15

## The Canvas



# Exercice 3 - Contrôle de Flux

## The Brief

### Objectif

Écrire une fonction 'est\_pair'.

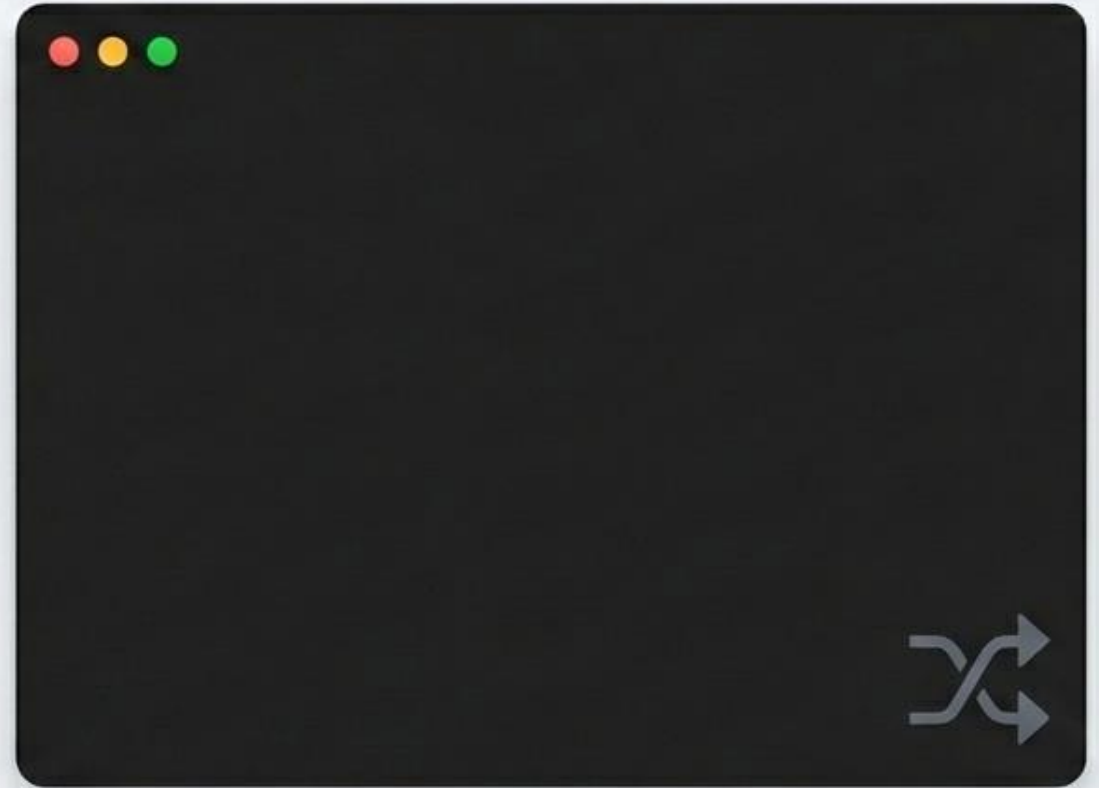
### Consigne

Vérifier la parité d'un nombre entier.

### Sortie Attendue (Return Code)

- return 0 (Si pair)
- return 1 (Si impair)

## The Canvas



# Exercice 4 - Utilisation de return

## The Brief

**Objectif:** Écrire une fonction "verifier\_fichier".

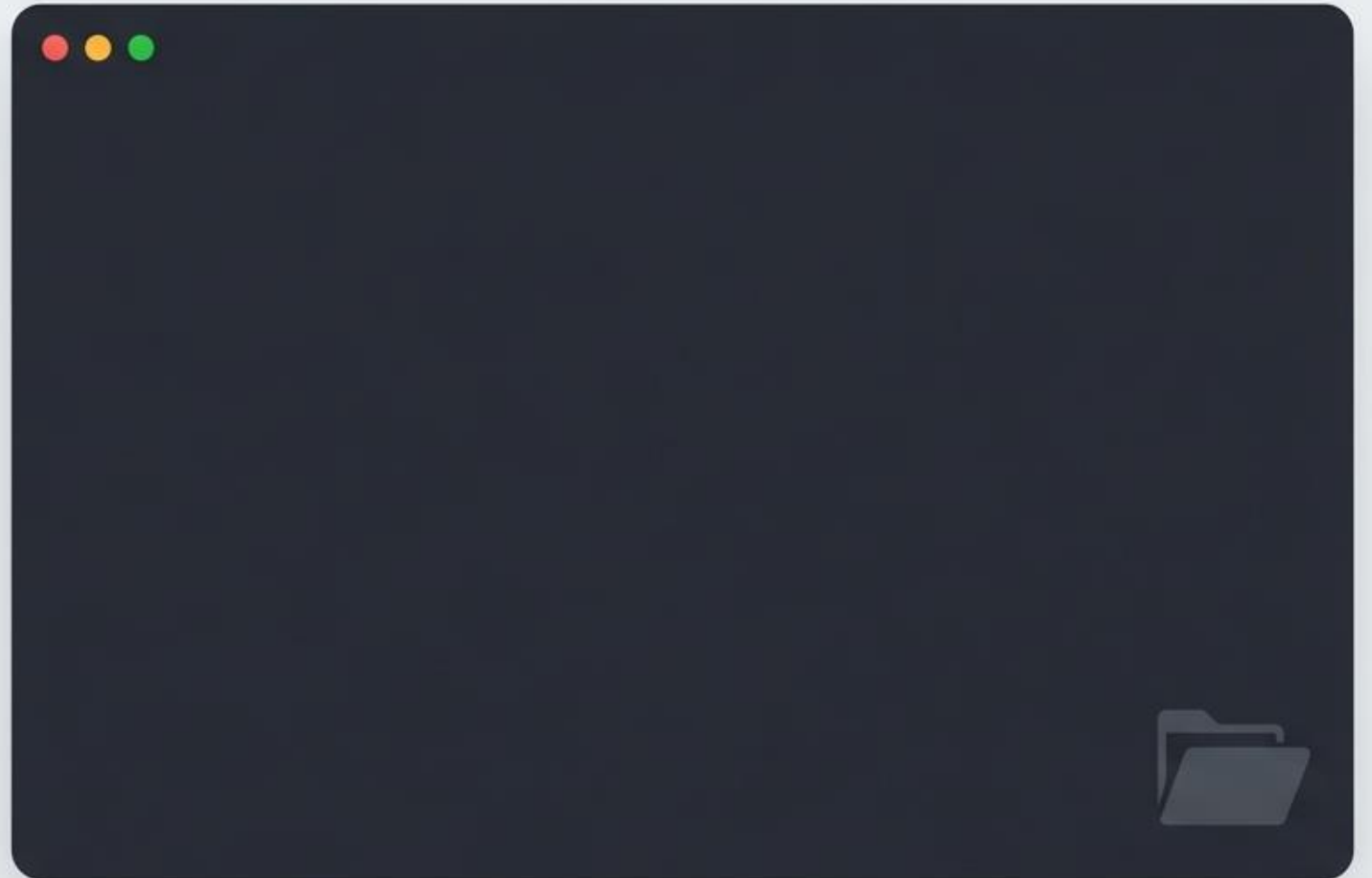
**Consigne:** Vérifier l'existence d'un fichier spécifique sur le système.

**Entrée:** Le chemin/nom du fichier.

**Sortie Attendue (Return Code):**

- 0 si le fichier existe.
- 1 si le fichier n'existe pas.

## The Canvas



# Exercice 5 - Traitement de Tableaux

## The Brief

### Objectif

Écrire une fonction  
"trouver\_maximum".

### Consigne

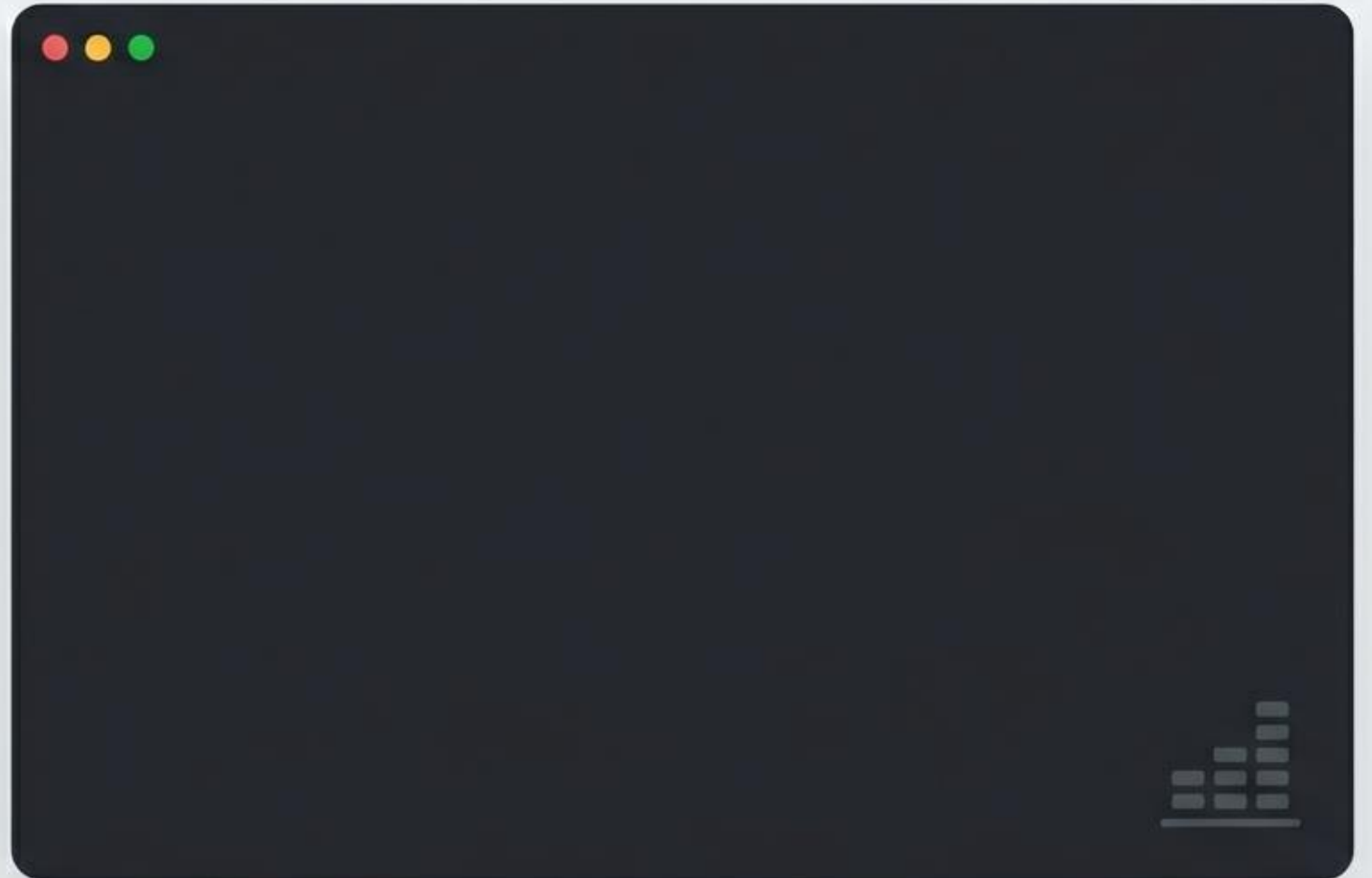
Parcourir un tableau pour  
identifier la valeur la plus élevée.

**Entrée:** Un tableau de nombres.

### Sortie Attendue

Affichage du plus grand nombre.

## The Canvas



# Exercice 6 - Fonction avec Sortie Multiple

## The Brief

### Objectif

Créer une fonction 'stats\_fichier'.

### Consigne

Analyser un fichier texte pour extraire des statistiques.

### Sortie Attendue

1. Nombre de lignes
2. Nombre de mots
3. Nombre de caractères

## The Canvas

