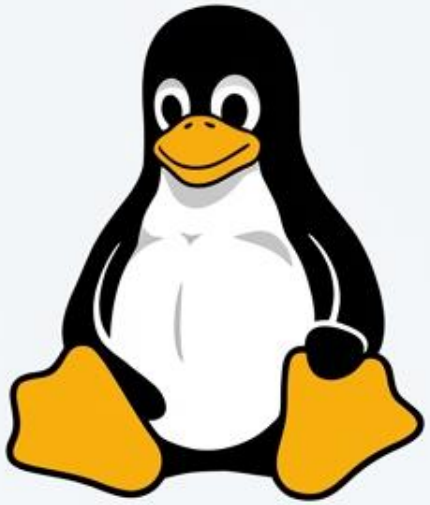


Les Scripts Linux - Partie 5 : Les Tableaux

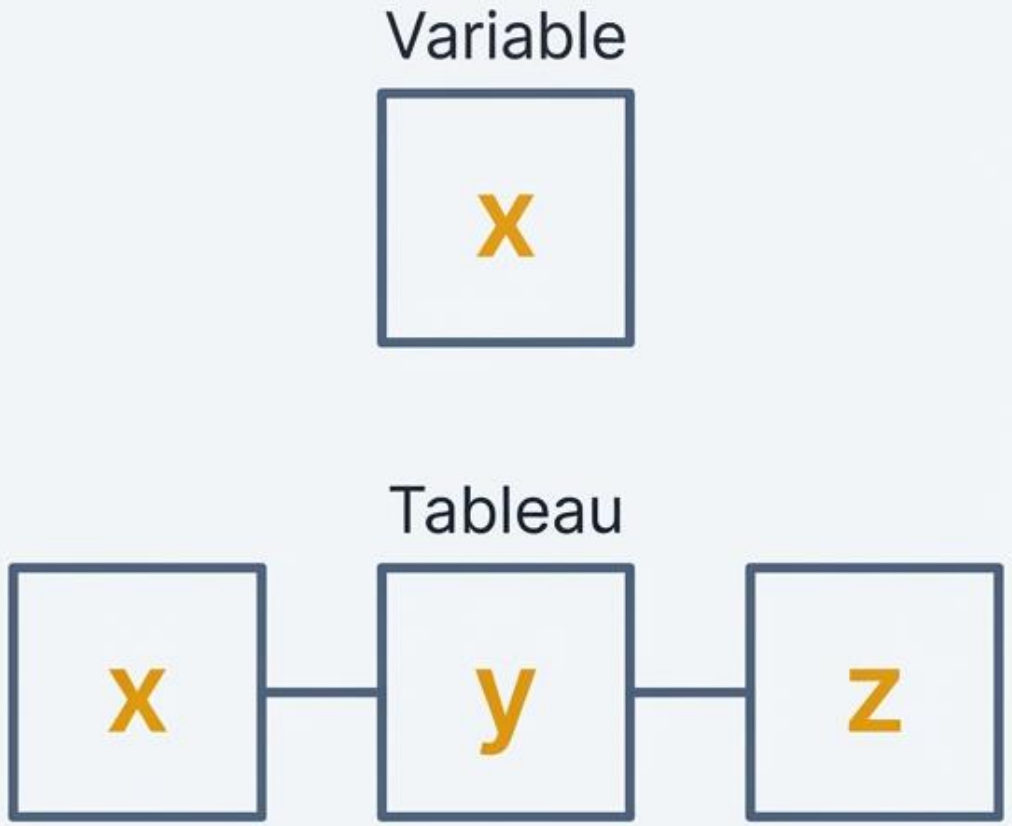
BTS CIEL Informatique et Réseaux

Définition d'un tableau

Un tableau en Bash est une structure de données qui permet de stocker une série de valeurs (éléments).

Les tableaux peuvent être :

- Unidimensionnels (listes simples)
- Multidimensionnels (similaires à des matrices)



Déclaration d'un tableau

Syntaxe :

```
nom_tableau=(élément1 élément2 élément3 ...)
```

Exemple :

```
jours=("Lundi" "Mardi" "Mercredi" "Jeudi" "Vendredi")
```

Accéder aux éléments

Pour accéder à un élément spécifique, utilisez l'indice entre crochets :

```
${nom_tableau[indice]}}
```

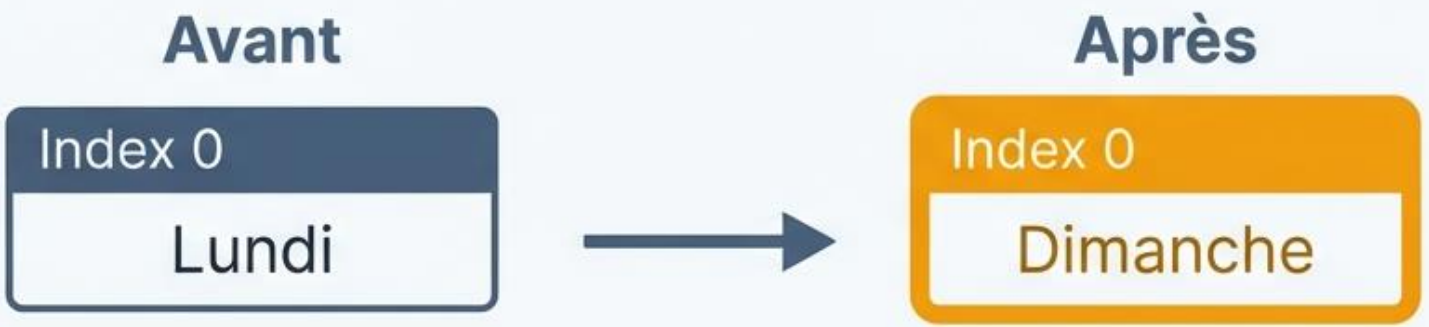
```
echo ${jours[0]}}   # Affiche : Lundi
```

Modifier un élément

Vous pouvez modifier la valeur d'un élément existant en ciblant son indice.

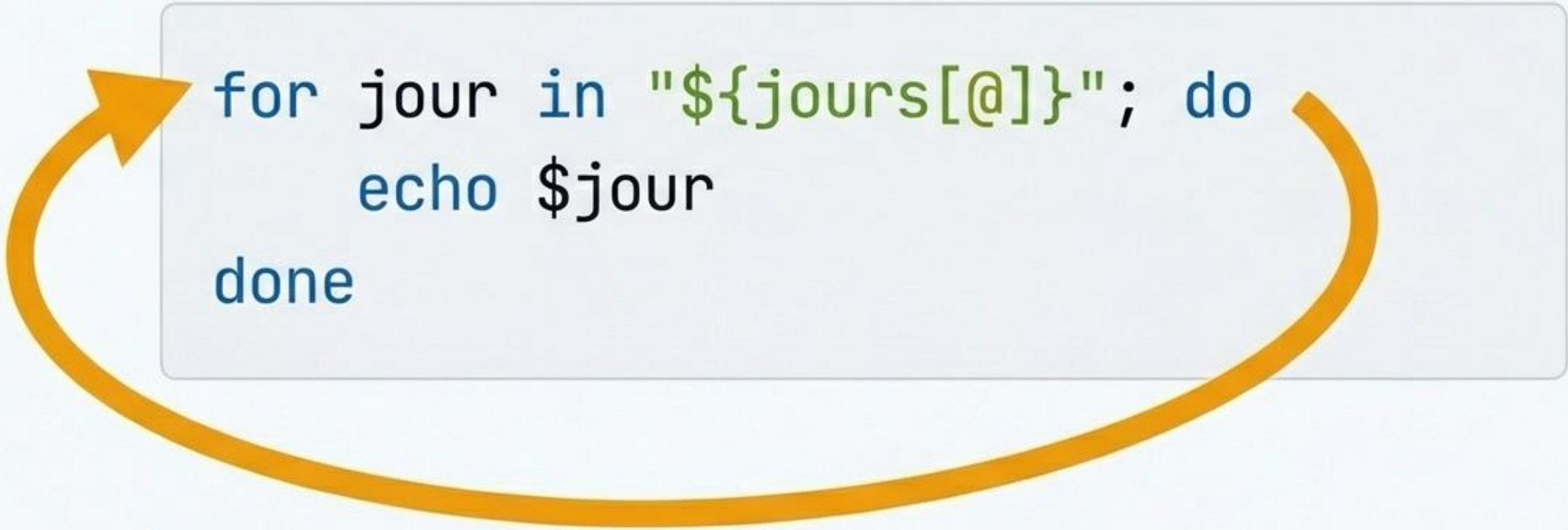
```
jours[0]="Dimanche"

# L'élément à l'indice 0 est désormais "Dimanche"
```



Parcourir un tableau

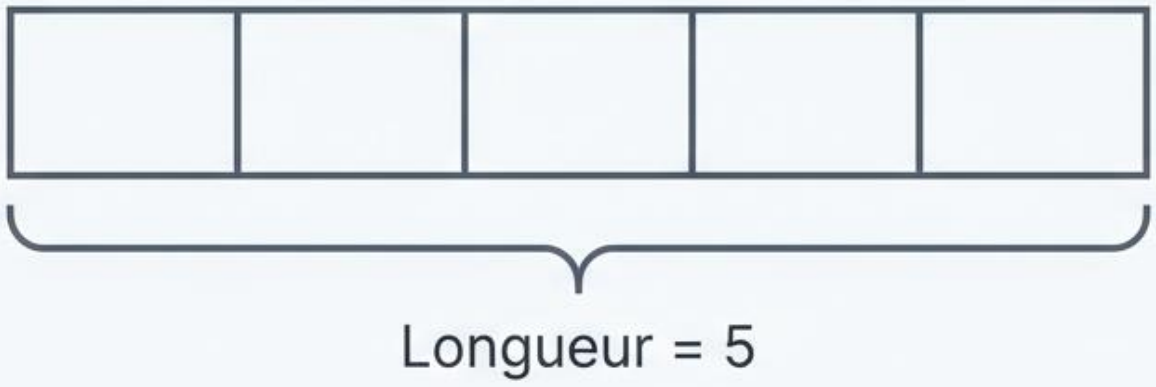
Pour lire tous les éléments séquentiellement, utilisez une boucle for avec la syntaxe [@].

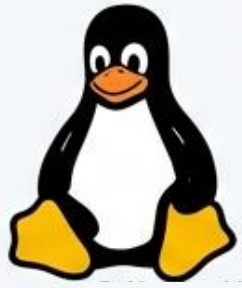


La longueur d'un tableau

Pour obtenir le nombre total d'éléments contenus dans le tableau, utilisez le symbole **#**.

```
echo ${#jours[@]}
# Affiche le nombre total d'éléments (ex: 5)
```





Les Tableaux Associatifs (Introduction)

Bash supporte les tableaux associatifs (Clé → Valeur), similaires aux dictionnaires en Python.

Important : Vous devez obligatoirement les déclarer explicitement avec l'option **-A**.

```
declare -A nom_tableau
```

Les Tableaux Associatifs (Exemple)

```
# Déclaration
declare -A capitales

# Assignation (Clé -> Valeur)
capitales["France"]="Paris"
capitales["Allemagne"]="Berlin"

# Accès par la clé
echo ${capitales["France"]}
# Affiche : Paris
```

Mises en pratique

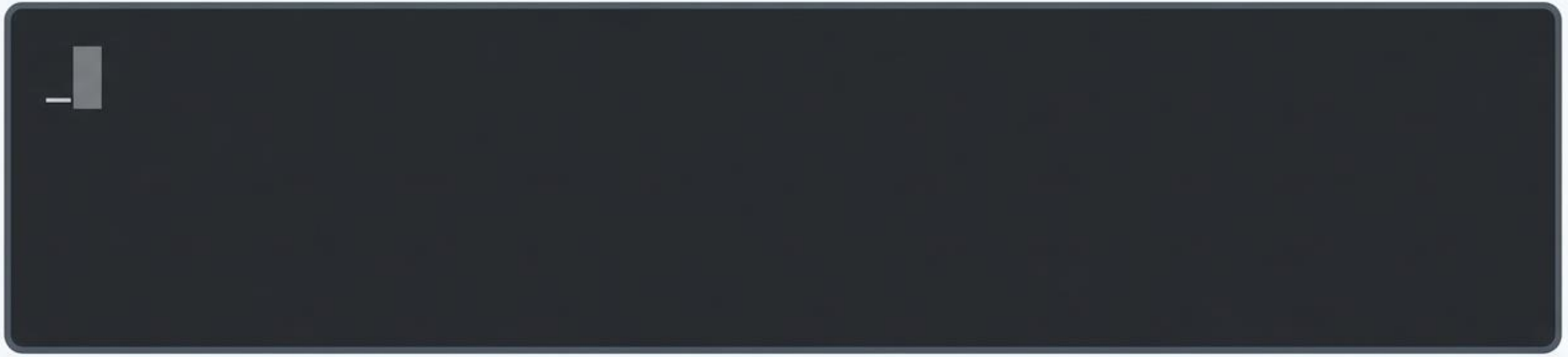
Série de 8 exercices

À vous de coder !

Exercice 1 : Création et Affichage

Objectif :
Créer un tableau contenant les noms de cinq fruits et afficher chaque fruit sur une nouvelle ligne.

💡 *Conseil* : Utilisez une boucle *for* pour parcourir le tableau.



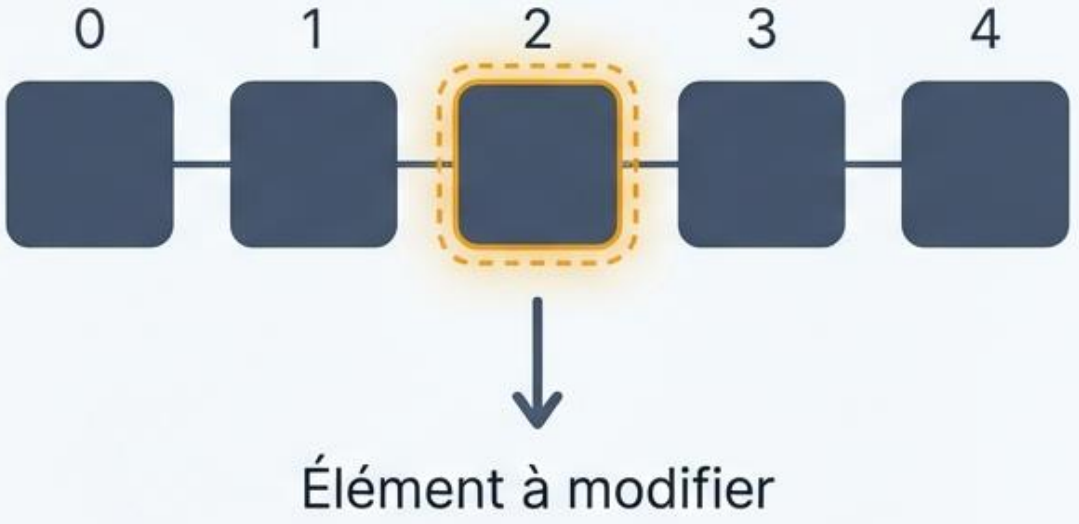
Exercice 2 : Modification d'Éléments

Objectif :

Créer un tableau de cinq nombres. Remplacez ensuite le troisième nombre par un autre, et affichez le tableau modifié.

Conseil : 💡

Accédez à l'élément du tableau par son indice (attention, l'indice commence à 0).



Exercice 3 : Tableau Associatif

Objectif :

Créer un tableau associatif (Pays/Capitales). Ajoutez au moins trois paires clé-valeur, puis affichez la capitale d'un des pays.

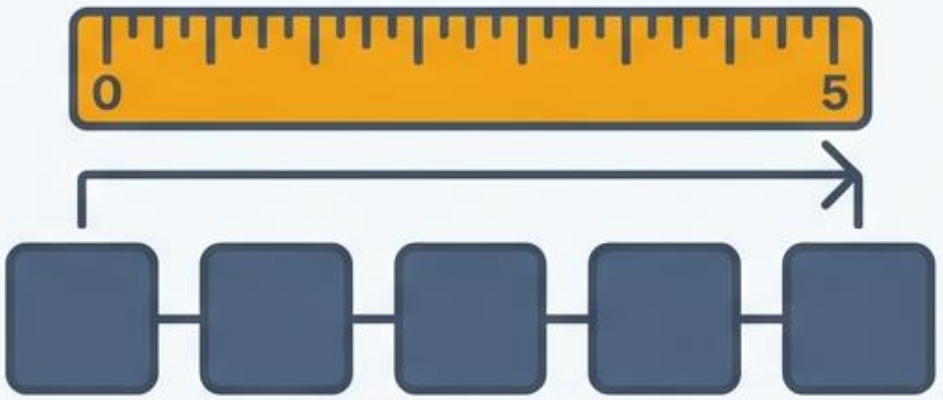
Conseil :

N'oubliez pas d'utiliser `declare -A` au début du script.

Exercice 4 : Longueur du Tableau

Objectif :

Créer un tableau de votre choix, puis écrire un script qui affiche la longueur totale de ce tableau.



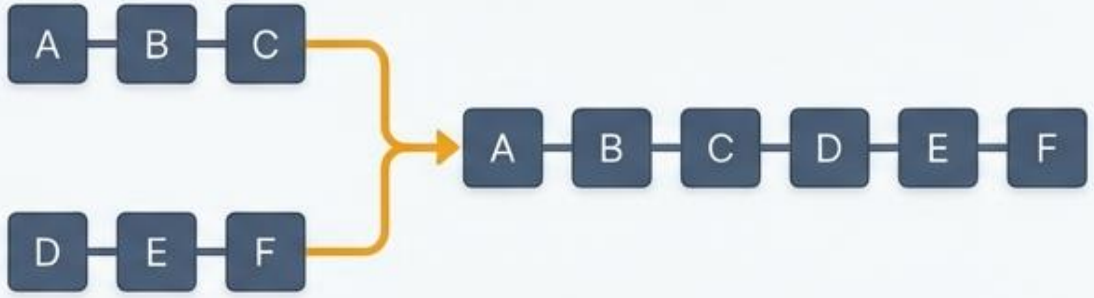
Conseil :

Utilisez la syntaxe `${#nom_tableau[@]}`

Exercice 5 : Concaténation

Objectif :

Créez deux tableaux avec quelques éléments chacun. Créez ensuite un troisième tableau qui est la fusion des deux premiers.



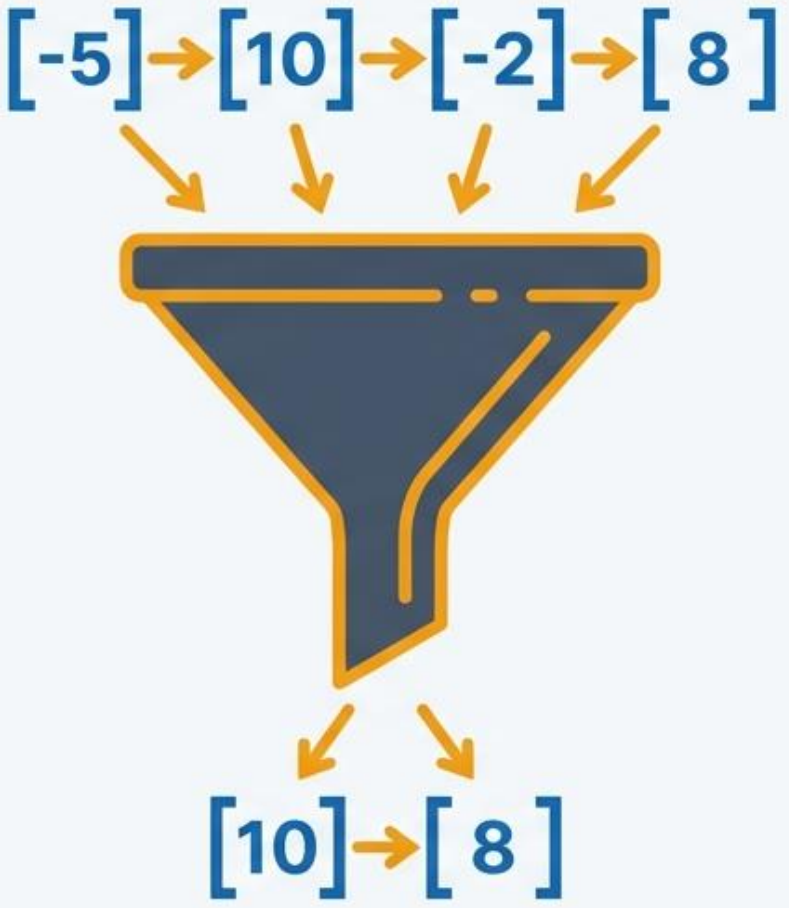
Conseil : Vous pouvez concaténer en utilisant la syntaxe `(${tableau1[@]} ${tableau2[@]})`

Exercice 6 : Filtrage

Objectif :

Créer un tableau de nombres mixtes (positifs et négatifs). Extraire uniquement les nombres positifs vers un nouveau tableau.

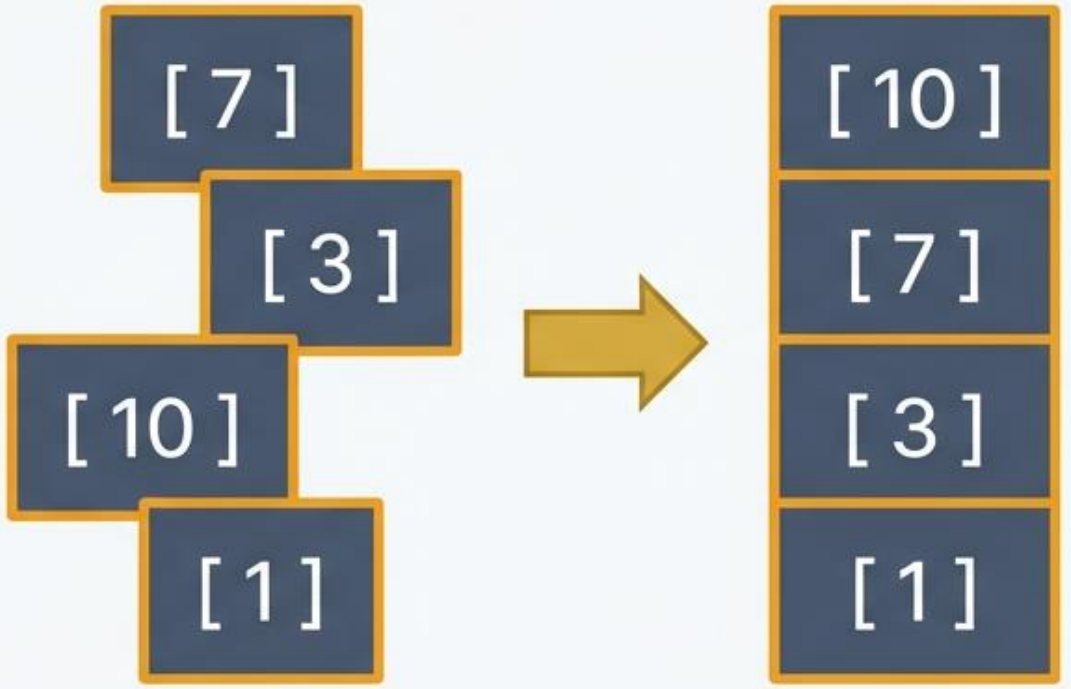
Conseil :
Combinez une boucle `for` pour la lecture et une condition `if` pour le filtrage.



Exercice 7 : Trier un tableau

Objectif :
Créer un tableau de mots ou de chiffres dans un ordre aléatoire. Écrire un script pour trier et afficher le résultat.

Conseil :
Bash a des limites pour le tri interne ; essayez d'utiliser la commande externe sort.



Exercice 8 : Inversion

Objectif : Créer un tableau, puis écrire un script qui inverse l'ordre des éléments (le dernier devient le premier).

Conseil : Pensez à utiliser une boucle qui parcourt le tableau en sens inverse (décrémentation).