

# TP : Maîtrise des Fonctions, Objets et Callbacks en JavaScript

## Consignes :

1. Tous les résultats doivent être affichés uniquement dans la **console**.
  2. Pour les exercices 1 à 6, écrivez d'abord la fonction en syntaxe `function name()`, puis réécrivez-la en **expression fléchée (lambda)**.
  3. Utilisez les structures de boucles demandées.
- 

## Partie 1 : Fonctions, Objets et Logique (6 Exercices)

### Exercice 1 : Gestion de stock (Boucle for...of)

Contexte : Un magasin de sport.

- **Objet** : Un tableau d'objets produits (ex: `{ nom: "Ballon", prix: 20, stock: 5 }`).
- **Mission** : Créer une fonction `calculerValeurStock` qui retourne la valeur totale du stock (`prix * stock`).
- **Structure** : Utilisez `for...of`.

### Exercice 2 : Filtrage d'utilisateurs (Boucle while)

Contexte : Une application de rencontre ou de réseau social.

- **Objet** : Un tableau d'utilisateurs `{ nom: "Alice", age: 25, actif: true }`.
- **Mission** : Créer une fonction `trouverMajeurs` qui parcourt le tableau avec une boucle `while` et retourne un nouveau tableau contenant uniquement les noms des personnes de 18 ans et plus.
- **Structure** : `while` avec un compteur.

### Exercice 3 : Analyse de propriétés (Boucle for...in)

Contexte : Fiche technique d'un véhicule.

- **Objet** : Un seul objet voiture avec diverses propriétés (marque, modèle, année, puissance, couleur).
- **Mission** : Créer une fonction `afficherInfos` qui prend l'objet en paramètre et retourne une chaîne de caractères listant toutes les clés et valeurs sous la forme "Clé : Valeur".
- **Structure** : `for...in`.

### Exercice 4 : Calcul de moyenne de classe (Boucle for classique)

Contexte : Gestion scolaire.

- **Objet** : Un objet `classe` contenant un tableau de notes [12, 15, 8, 19].
- **Mission** : Créer une fonction `evaluerClasse` qui calcule la moyenne. Si la moyenne est  $> 10$$ , elle retourne "Passable", sinon "Insuffisant".
- **Structure** : `for` (indexé) et `if/else`.

### Exercice 5 : Recherche d'ID (Instructions break/continue)

Contexte : Base de données de sécurité.

- **Objet** : Un tableau d'objets badges `{ id: "A12", acces: "OK" }`.
- **Mission** : Créer une fonction `verifierAcces` qui cherche un ID spécifique. Si elle trouve l'ID, elle s'arrête immédiatement et retourne l'objet. Si l'accès est "Interdit", elle ignore l'élément et passe au suivant.
- **Structure** : `for...of` avec `break` et `continue`.

### Exercice 6 : Transformation de prix (Map manuel)

Contexte : Soldes d'hiver.

- **Objet** : Un tableau d'objets articles { nom: "Veste", prix: 100 }.
  - **Mission** : Créer une fonction appliquerRemise qui prend un pourcentage en paramètre et retourne un nouveau tableau d'objets avec les prix modifiés.
  - **Structure** : Syntaxe classique vs Lambda.
- 

## Partie 2 : Fonctions de Callback et Tri (4 Exercices)

**Le concept** : On crée une fonction de traitement (ex: tri) qui reçoit en paramètre une autre fonction (le callback) pour afficher ou formater le résultat final.

### Exercice 7 : Le Podium des Gamers

**Contexte** : Score d'un tournoi d'E-sport.

- **Données** : Tableau { pseudo: "Player1", score: 4500 }.
- **Traitement** : Une fonction trierScores qui trie les joueurs du plus grand au plus petit score.
- **Callback** : Une fonction afficherPodium qui reçoit le tableau trié et affiche en console uniquement les 3 premiers.

### Exercice 8 : Répertoire Téléphonique

**Contexte** : Liste de contacts.

- **Données** : Tableau { nom: "Zola", tel: "06..." }.
- **Traitement** : Une fonction organiserContacts qui trie par ordre alphabétique des noms.
- **Callback** : Une fonction formatSMS qui prend le résultat et affiche pour chaque contact : "Envoi du message à [Nom]...".

### Exercice 9 : Gestion de File d'Attente (Hôpital)

**Contexte** : Urgences médicales.

- **Données** : Tableau { patient: "Jean", gravite: 3 } (1 = urgent, 5 = bénin).
- **Traitement** : Une fonction ordonnerUrgences qui trie par niveau de gravité croissant.
- **Callback** : Une fonction appelerPatient qui affiche "Le patient [Nom] est attendu en salle de soin".

### Exercice 10 : Catalogue de Bibliothèque

**Contexte** : Archivage de livres.

- **Données** : Tableau { titre: "Le Cid", pages: 150 }.
  - **Traitement** : Une fonction trierParEpaisseur qui trie les livres par nombre de pages.
  - **Callback** : Une fonction genererEtiquette qui affiche "Livre : [TITRE] ([PAGES] pages)".
- 

## Exemple de corrigé pour l'exercice 1 (pour vous aider)

JavaScript

```
const produits = [
  { nom: "Ballon", prix: 20, stock: 5 },
  { nom: "Maillot", prix: 40, stock: 10 }
];
```

// Syntaxe Classique

```
function calculerValeurStock(liste) {
    let total = 0;
    for (const p of liste) {
        total += p.prix * p.stock;
    }
    return total;
}

// Syntaxe Lambda (Expression fléchée)
const calculerValeurStockLambda = (liste) => {
    let total = 0;
    for (const p of liste) {
        total += p.prix * p.stock;
    }
    return total;
};

console.log("Valeur totale : " + calculerValeurStock(produits));
```