

Exercice 1 : Génération de listes génériques

Créez une fonction **generateList** qui prend en entrée un tableau de chaînes de caractères et renvoie une liste HTML de ces éléments. Le type de liste sera passé en paramètre (ul, ol). **Utilisez les littéraux de gabarit.**

```
let data = ['Pomme', 'Banane', 'Cerise'];
```

Exercice 2 : Génération de tableaux génériques

Créez une fonction **generateTable** qui prend un tableau d'objets et renvoie une table HTML. Chaque objet représentera une ligne de la table, et les clés de l'objet seront les entêtes. **Utilisez les littéraux de gabarit et les fonctions d'objet.**

```
let data = [  
  {Nom: 'Dupont', Prénom: 'Jean', Age: 25},  
  {Nom: 'Martin', Prénom: 'Marie', Age: 30},  
];
```

Exercice 1 : Conversion d'objet en JSON

Ecrivez une fonction nommée **convertToJson** qui prend un objet JavaScript et retourne sa représentation sous forme de chaîne JSON. Ensuite, testez la fonction en utilisant un objet contenant vos informations, comme nom, âge et pays.

Exemple :

```
let person = {  
  nom: 'Dupont',  
  age: 30,  
  pays: 'France'  
};  
let jsonStr = convertToJson(person);  
console.log(jsonStr);
```

Sortie attendue :

```
"{"nom": "Dupont", "age": 30, "pays": "France"}"
```

Exercice 2 : Lecture d'un JSON

Ecrivez une fonction nommée **readFromJSON** qui prend une chaîne JSON représentant un tableau d'objets et retourne le tableau JavaScript correspondant. Chaque objet représente un livre avec titre, auteur et année.

Puis, testez la fonction avec un exemple de chaîne JSON.

Exemple

```
let jsonStr = '[{"titre":"1984","auteur":"George Orwell","annee":1949},{"titre":"Le  
Meilleur des mondes","auteur":"Aldous Huxley","annee":1932}]';  
let books = readFromJSON(jsonStr);  
console.log(books);
```

Sortie attendue :

```
[  
  { titre: '1984', auteur: 'George Orwell', année: 1949 },  
  { titre: 'Le Meilleur des mondes', auteur: 'Aldous Huxley', année: 1932 }  
]
```

Exercice 3 : Détection d'erreurs JSON

Ecrivez une fonction **safeParse** qui tente de désérialiser une chaîne JSON. Si la chaîne est valide, la fonction doit retourner l'objet ou le tableau JavaScript correspondant. Si la chaîne est invalide, elle doit retourner `null`.

Exemple

```
let validJSON = '{"name":"Alice"}';
let invalidJSON = '{"name":""Alice",}';
console.log(safeParse(validJSON));    // { name: 'Alice' }
console.log(safeParse(invalidJSON)); // null
```

Sortie attendue :

```
{ name: 'Alice' }
null
```

Exercice 4 : Manipulation de format JSON en JavaScript

On vous donne un objet JSON représentant une base de données d'employés. Vos tâches sont :

- Analyser le JSON pour le convertir en objet JavaScript.
- Afficher le nombre total d'employés dans la base de données.
- Afficher le nom de chaque employé qui appartient au département "IT".
- Augmenter le salaire de chaque employé de 10%.
- Convertir l'objet JavaScript mis à jour en une chaîne de caractères JSON.

```
{
  "employees": [
    {"id": 1, "nom": "John Doe", "departement": "IT", "salaire": 50000},
    {"id": 2, "nom": "Jane Smith", "departement": "RH", "salaire": 45000},
    {"id": 3, "nom": "Samantha Right", "departement": "IT", "salaire": 55000},
    {"id": 4, "nom": "Robert Brown", "departement": "Finance", "salaire": 60000}
  ]
}
```