

# Fiche Récap : Méthodes de Tableaux et Objets en JavaScript

## **Array.prototype.map()**

Transformer chaque element -> nouveau tableau

Parametre : (element, index, array)

Retourne : un nouveau tableau de meme taille

Exemple :

`[1, 2, 3].map(x => x * 2) -> [2, 4, 6]`

## **Array.prototype.filter()**

Filtrer les elements selon une condition

Parametre : (element, index, array)

Retourne : un tableau filtre

Exemple :

`[1, 5, 8].filter(x => x > 4) -> [5, 8]`

## **Array.prototype.find()**

Trouve le premier element qui satisfait une condition

Parametre : (element, index, array)

Retourne : l'element ou undefined

Exemple :

`[3, 6, 9].find(x => x % 2 === 0) -> 6`

## **Array.prototype.includes()**

Verifie la presence d'un element

Parametre : valeur, [fromIndex]

Retourne : true / false

Exemple :

`['pomme', 'poire'].includes('pomme') -> true`

## **Array.prototype.some()**

## Fiche Récap : Méthodes de Tableaux et Objets en JavaScript

Verifie si au moins un element satisfait une condition

Parametre : (element, index, array)

Retourne : true / false

Exemple :

```
[1, 3, 5].some(x => x % 2 === 0) -> false
```

### **Array.prototype.every()**

Verifie si tous les elements satisfont une condition

Parametre : (element, index, array)

Retourne : true / false

Exemple :

```
[2, 4, 6].every(x => x % 2 === 0) -> true
```

### **Array.prototype.reduce()**

Reduit un tableau a une seule valeur

Parametre : (acc, val, index, array)

Retourne : une valeur unique

Exemple :

```
[10, 15, 20].reduce((acc, val) => acc + val, 0) -> 45
```

### **Array.prototype.forEach()**

Parcours le tableau sans retour

Parametre : (element, index, array)

Retourne : rien

Exemple :

```
['bleu', 'vert'].forEach((c, i) => console.log(i + ' => ' + c))
```

### **Object.keys/values/entries**

Travailler sur les objets

## Fiche Récap : Méthodes de Tableaux et Objets en JavaScript

`Object.keys(obj)` -> `['nom', 'age']`

`Object.values(obj)` -> `['Ali', 22]`

`Object.entries(obj)` -> `[['nom', 'Ali'], ['age', 22]]`

### Memo Utilisation

Besoin -> Methode

Transformer un tableau -> `map()`

Filtrer des donnees -> `filter()`

Un seul resultat attendu -> `reduce()` ou `find()`

Verification partielle -> `some()`

Verification totale -> `every()`

Juste parcourir -> `forEach()`