Les tableaux en JS



⚠️ en JS : pas de différence entre tableau et liste. Les tableaux en JS n’ont pas de taille fixe, si on rajoute des éléments dedans, il grandit et rétrécit si on en enlève.

⚠️ On peut mettre n’importe quoi dans un tableau, cela peut causer beaucoup de soucis par la suite, donc il faut absolument faire des tableaux avec le même type de données à l’intérieur !

const user =



User1 User2

On crée un tableau qui s’appelle hobbies (nom de variable à utiliser) dans lequel on a mis :   
- dans la 1ère case : Pétanque  
- dans la 2ème case : Ricard  
- dans la 3ème case : Poulet-compote



Les tableaux ont un index, aussi appelé indice, pour pouvoir consulter une page en particulier. Il commence toujours à 0

L’index maximum vaudra toujours la taille de votre tableau -1

A screenshot of a computer

AI-generated content may be incorrect.

Si je veux afficher la valeur de la 2ème case, elle porte la valeur 1



**Consulter la taille d’un tableau**

****

****

**Ajouter un nouvel élément au tableau**

On utiliser une méthode qui permet d'ajouter un élément à la fin d'un tableau existant

A black background with yellow and blue text

AI-generated content may be incorrect.

Moins utilisé mais existe, une méthode pour ajouter au début du tableau :



**Supprimer un élément au tableau**

Pop()

Supprime toujours le dernier élément 🡪 Peu utilisée



Splice()

S’écrit : tableau.splice(1) / (indexDeDepart, nbElementASupp)

nbElementASupp est optionnel mais si vous ne le mettez pas, supprime tout jusqu'à la fin



## Quelques méthodes utiles

**Convertir en chaîne de caractères**

.toString()

Crée une chaîne de caractères en concatétant tous les élements du tableau et en les séparant par une virgule



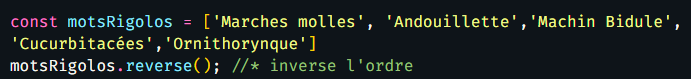
.join([séparateur])

si on laisse vide --> , comme le comme toString()   
si je mets une valeur, intègre un séparateur



**Inverser l’ordre des éléments**

reverse()





**Trier un tableau**

sort([methodeComparaison])

Permet de trier les éléments d’un tableau sur base d’une méthode qui doit retourner un nombre positif, négatif ou zéro. Le tri ne se fait pas sur les nombres mais sur le poids des caractères ascii (askip), donc pas besoin de fonction de comparaison

* **Par ordre croissant**

**A black background with numbers and letters

AI-generated content may be incorrect.**

* 8, 15, 23, 42
* **Par ordre décroissant**



* 42, 23, 15, 8
* **Pour un tableau de chaine**



* "George", "John", "Paul", "Ringo"