
JS

Les structures de données

- Les variables de type Array (tableaux)
- Le principe des tableaux est relativement simple : un indice ou clef va être associé à chaque valeur du tableau. Pour récupérer une valeur dans le tableau, on va utiliser les indices qui sont chacun unique dans un tableau.
- Deux types de tableaux : les tableaux dont les clefs ou indices sont des chiffres et qu'on appelle tableaux numérotés et les tableaux dont les clefs ou indices sont des chaines de caractères définies par le développeur et qu'on appelle tableaux associatifs.
- JavaScript ne gère qu'un type de tableau : les tableaux numérotés.
- Let tableau = ['chaine', 25, 152,35,true];
- Afficher un indice : console.log(tableau[0]);
- Afficher les valeurs du tableau : for(data of tableau){}
- Afficher les indices sous forme de chaine de caractères : for(data in tableau){}
- Notez qu'on va pouvoir utiliser une boucle for ... of pour parcourir les propriétés d'un objet littéral une à une. La boucle for ... in est l'équivalent de la boucle for ... of mais pour les objets

Array () et ses methodes

- Le constructeur Array() ne possède que deux propriétés : la propriété length qui retourne le nombre d'éléments d'un tableau et la propriété protoytpe qui est une propriété que possèdent tous les constructeurs en JavaScript.
- Array() possède également une trentaine de méthodes :
- push(), pop(), shift(), unshift(), splice(), join(), slice(), concat(), includes(), filter(), reduce(), etc...
- Le cas de l'objet Map() : contient des paires de clé-valeur et mémorise l'ordre dans lequel les clés ont été insérées. N'importe quel type de valeur (primitive ou objet) peut être utilisée comme clé ou comme valeur.
- Un objet Map permet de parcourir ses éléments selon leur ordre d'insertion. Par exemple, une boucle for ... of revoie a un tableau [cle - valeur] pour chaque itération
- Le cas de la méthode reduce() :
- En JavaScript, la méthode reduce() est utilisée pour réduire un tableau en une seule valeur en exécutant une fonction donnée. La valeur retournée par la fonction est stockée dans ce qu'on appelle un "accumulateur".

Spread Operator :

- La **syntaxe de décomposition** permet d'étendre un itérable (par exemple une expression de tableau ou une chaîne de caractères) en lieu et place de plusieurs arguments (pour les appels de fonctions) ou de plusieurs éléments (pour les littéraux de tableaux) ou de paires clés-valeurs (pour les littéraux d'objets).
- L'opérateur spread dans JavaScript est une syntaxe introduite dans ECMAScript 6 (ES6) qui vous permet d'étaler les éléments d'un itérable (tels que des tableaux, des chaînes de caractères ou des objets) dans un autre itérable ou dans un appel de fonction.
- Il est désigné par trois points ''...element'' suivis d'une expression ou d'un itérable. L'opérateur spread est un outil puissant qui offre un moyen concis et flexible de travailler avec des données dans JavaScript.
- Il peut être utilisé pour concaténer des tableaux, créer des copies superficielles de tableaux, convertir des chaînes en tableaux de caractères, fusionner ou cloner des objets et passer dynamiquement des valeurs dans des fonctions ou des constructeurs, entre autres cas d'utilisation.
- L'opérateur spread simplifie les opérations complexes et permet d'obtenir un code plus expressif et plus efficace.