

Fonctionnalité : Algorithme de recherche

Problématique : Accéder rapidement à une recette correspondant à un besoin de l'utilisateur dans les recettes déjà reçues.

Objectif : Améliorer le filtrage des recettes en comparant les méthodes native (boucles for) et fonctionnelle (forEach) pour évaluer performances et lisibilité.

La **barre de recherche** filtre les recettes en fonction des termes saisis dans les titres, les ingrédients et les descriptions. Elle prend en charge plusieurs valeurs séparées par des espaces pour retourner les recettes correspondant à toutes les spécifications.

Les **sélecteurs** (Ingrédients, Appareils, Ustensiles) influent sur la recherche principale en filtrant les recettes selon des mots-clés spécifiques à chaque catégorie. Les résultats comprennent uniquement les recettes qui répondent aux critères de la recherche globale et des sélecteurs.

Fonction 1 : utilisation des méthodes fonctionnelles

Les fonctions telles que `forEach()`, `map()` ou `reduce()` en JavaScript sont intégrées au langage. Elles simplifient la description des actions pour chaque élément d'un tableau, sans avoir à spécifier comment l'itération doit être effectuée.

Avantages

- Syntaxe plus lisible et déclarative.
- Facilité d'utilisation.

Inconvénients

- Moins de contrôle direct sur l'itération.
- Peut être moins performant sur des grandes quantités de données.

Fonction 2 : utilisation des méthodes natives

Les boucles for offrent un contrôle précis et une performance potentielle, elles peuvent être moins élégantes et nécessiter une gestion plus détaillée que les approches fonctionnelles.

Avantages

- Contrôle précis.
- Plus performant sur des grandes quantités de données.

Inconvénients

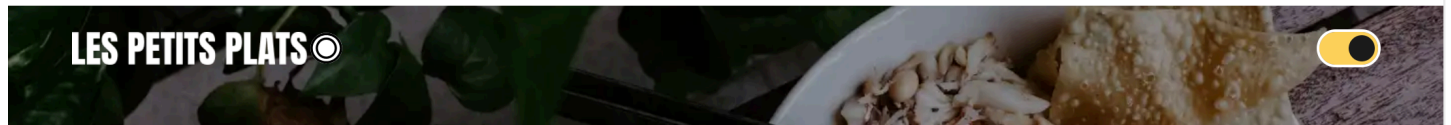
- Syntaxe plus verbeuse
- Moins déclarative, nécessite une gestion plus détaillée.
- Risque d'erreurs.

Capture des performances :

Performances (average)

Native = 1.4 ms

Functionnal = 0.8 ms



Solution retenue :

Une fonction de mesure des performances a été implémentée pour calculer le temps en millisecondes nécessaire à l'exécution de la fonction de filtrage. Il est observé que la fonction utilisant des méthodes fonctionnelles est plus rapide que celle utilisant des méthodes natives. Par conséquent, la fonction basée sur les méthodes fonctionnelles a été choisie.

Annexe : Algorithme

