

Fonctionnalité : Système de recherche	Fonctionnalité #2
<p>Problématique : Afin de rendre l'expérience de recherche la plus fluide possible, nous cherchons l'approche qui nous permettra d'avoir les meilleures performances.</p> <p>Ces deux approches seront testées dans un premier temps sur la barre de recherche.</p> <p>La solution</p>	

Solution 1 : Approche impérative (for, while...)

Dans cette option, nous utiliserons les fonctions natives de JavaScript pour filtrer les recettes en fonction d'une chaîne de caractères saisie dans la barre de recherche.

Une recette sera incluse dans les résultats si la chaîne de recherche est présente dans son titre, sa description ou l'un de ses ingrédients

Avantages :

- Contrôle total sur la logique de recherche
- Meilleure compatibilité avec les anciennes versions de javascript
- Meilleure gestion de la mémoire

Inconvénients :

- Moins lisible et plus verbeux
- Moins performant sur de grands jeux de données

Solution 2 : Approche déclarative (filter(), some(), every(), includes())

Dans cette option, nous utiliserons les méthodes JavaScript telles que **filter()**, **some()** et **every()** pour filtrer les recettes en fonction d'une chaîne de caractères saisie dans la barre de recherche.

Une recette sera incluse dans les résultats si la chaîne de recherche est présente dans son titre, sa description ou l'un de ses ingrédients.

Avantages :

- Plus lisible et concis
- Meilleure maintenabilité

Inconvénients :

- L'ensemble du tableau doit être parcouru

Solution retenue :

En comparant les performances des deux approches sur différentes tailles de jeux de données (50, 100 et 200 recettes), il apparaît que **l'approche déclarative** est plus rapide que l'approche impérative.