

# Fiche d'investigation de fonctionnalité

<b>Fonctionnalité</b> : Rechercher dans l'input principal	<b>Fonctionnalité #2</b>
<b>Problématique</b> : Accéder <b>rapidement</b> à une recette suite au choix de l'utilisateur (depuis le fichier de recettes mise à disposition).	

## Option 1 : Approche code fonctionnel

Dans cette option, nous avons un filtre de recettes qui utilise le code fonctionnel et renvoie un tableau avec les recettes recherchées.

La fonction analyse le tableau avec map, elle filtre (.filter) le tableau de recettes sous condition de la présence (.includes) de la valeur de l'input et renvoie un tableau de recettes filtrées.

On utilise les méthodes .map, .filter, .includes, .some pour filtrer un tableau en fonction d'une chaîne de minimum 3 caractères dans les propriétés description, name et ingredients.

Avantages	Inconvénients
Lisibilité et maintenance Légèrement plus rapide	Les méthodes peuvent être difficile à appréhender pour les débutants

## Option 2 : Approche boucles natives

Dans cette option, on utilise des boucles "for" pour l'itération des datas recette, elles retournent un tableau avec les recettes recherchées.

La fonction loopByMainInput, va analyser chaque itération du tableau de recettes, elle vérifie la présence de la valeur saisie. Si la valeur est présente elle rajoute la recette à un tableau qu'elle renvoie.

On utilise les boucles for et indexOf pour retrouver la présence d'une chaîne de 3 caractères minimum dans une chaîne plus longue

Avantages	Inconvénients
Facilité d'écriture d'une boucle dans une boucle Utilisé dans de nombreux langages de programmation	Plus verbeux Lisibilité moindre du code

### Solution retenue :

Nous avons retenu l'approche "code fonctionnel". La raison est qu'elle offre une meilleure lisibilité et maintenance. Les performances bien d'infimes nous ont également convaincu.

# Fiche d'investigation

