Fiche d'investigation de fonctionnalité

Fonctionnalité : `Filtrer les recettes`	Fonctionnalité #1
Problématique : `Accéder rapidement à une recette correspondant à un besoin de l'utilisateur dans les recettes déjà reçues. Le cas d'utilisation commence lorsque l'utilisateur entre au moins 3 caractères dans la barre de recherche principale.`	

Option 1 : `Approche fonctionnelle map() includes()`

Avantages :

- Compatible avec tout les navigateurs
- Plus facile à lire
- Nécessite moins de code
- permet de changer la nature des éléments lors du traitement : map() construit un nouveau tableau.
- Idéal pour utilisez une valeur de retour

Inconvénients :

- N'utilise pas les instructions « break ; » et « continue ; »

Nombre de boucles nécessaires : 3
Nombre de boucles optionnelles : 2

Option 2 : `Loop for…of`

Avantages :

- Utilise les instructions « break ; » et « continue ; »

Inconvénients :

- N'est pas supportée par Internet Explorer
- Moins facile à lire
- Nécessite plus de code

Nombre de boucles nécessaires : 3
Nombre de boucles optionnelles : 2

Solution retenue :

Nous avons donc retenu l'algorithme « Loop for…of » pour ces différentes raisons :

- Parmi les boucles natives et fonctionnelles testées, la boucle `for…of' est apparue comme l'une des meilleures solutions selon le score obtenu au JavaScript Benchmark.

- L'utilisation de la méthode « indexOf() » montre de meilleure performance que les méthodes « include() » et « match() » ;

- Mettre la condition concernant le nom de la recette et la description dans le même « if » permet de gagner en performance ;

- L'utilisation des instructions « break ; » ou « continue ; » permettent un gain de performance puisque dès qu'une condition est remplie, on passe à l'itération suivante.

- La condition « if » placée avant la seconde boucle permet de gagner en performance puisque la boucle parcours un tableau.