

Fiche d'investigation de fonctionnalité

Fonctionnalité : Recherche

Fonctionnalité #1

Problématique : L'utilisateur doit pouvoir rechercher une recette en entrant un mot-clé qui sera recherché dans le titre, description et ingrédients des recettes pour ressortir la ou les recettes correspondantes.

Option 1 : Boucle « for »

Dans cette option, nous utilisons la boucle « for » pour parcourir les recettes avec un index *i* qui représente l'index de la recette dans un Array, pour comparer les éléments de la recettes *i* à un objet qui contient l'ensemble des conditions de recherches.

Avantages :

- *Le navigateur connaît déjà le nombre de boucle à itérer.*
- *S'adapte très bien à l'augmentation des recettes grâce au condition pour stopper les boucles.*

Inconvénients :

- *N'est pas dynamique, on doit connaître le nombre de boucle à faire.*

Nombres d'input minimum à remplir : 0 ou 3 minimum

- 0 : Réinitialise totalement la recherche du mot-clé à l'état de départ
- 3 : Active algorithme pour rechercher la ou les recettes contenant le mot-clé.

Option 2 : Boucle « for each »

Dans cette option, nous utilisons la boucle « for each » pour parcourir les recettes dans un Array, pour comparer les éléments de la recette parcouru à un objet qui contient l'ensemble des conditions de recherches.

Avantages :

- *Pas besoin de déclarer d'incrément.*
- *On peut parcourir des tableaux dont on ne connaît pas d'avance la longueur facilement.*

Inconvénients :

- *Le navigateur travaille beaucoup plus.*
- *Travaille directement sur le tableau (ce qui peut mener à des erreurs si le tableau est modifié).*

Nombres d'input minimum à remplir : 0 ou 3 minimum

- 0 : Réinitialise totalement la recherche du mot-clé à l'état de départ
- 3 : Active algorithme pour rechercher la ou les recettes contenant le mot-clé.

Solution retenue :

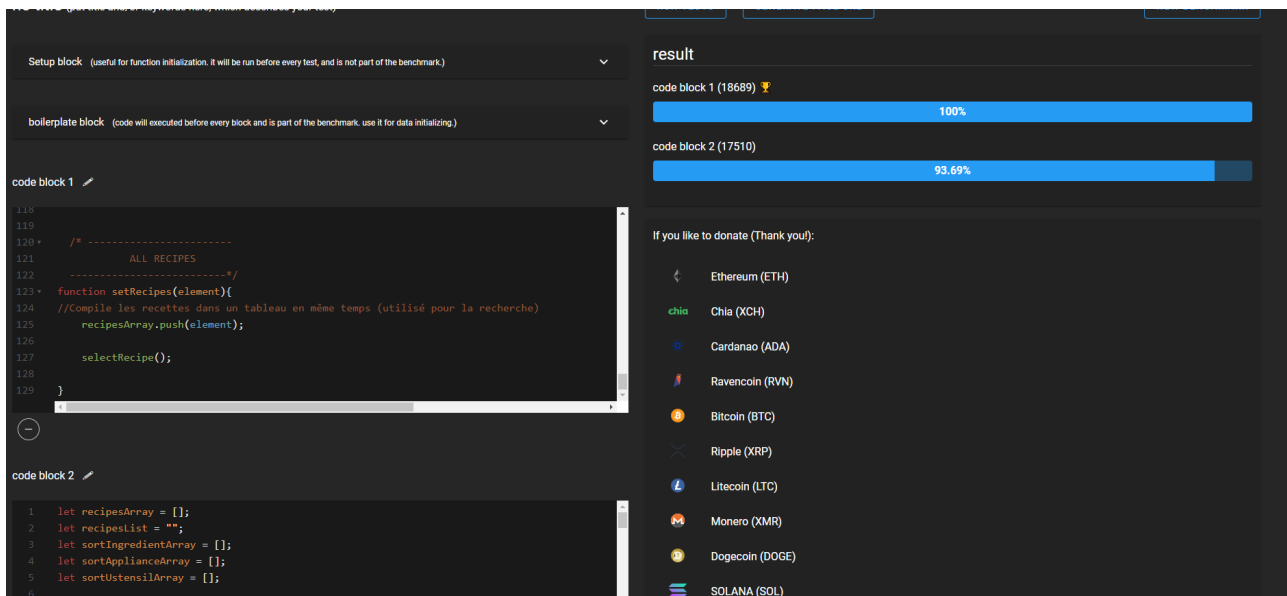
L'option retenu pour ce projet sera la boucle « for », car nous parcourons un tableau de recette simple dans cette fonctionnalité, dont on peut connaître la taille finale assez simplement, suivant le nombre d'objet récupérer par le Fetch à l'aide d'un simple `array.length`.

De plus suite aux différents tests (voir en-dessous), même si la différence est minime, la boucle « for » est systématiquement plus rapide que la boucle « for each » bien que les écarts soient variables d'un test à l'autre.

Pour cette raison la boucle « for » est privilégié.

ANNEXES

1] Résultat des tests entre les deux boucles :



Légende :

code block 1 = boucle « for »

code block 2 = boucle « for each »

II] Algorithme pour les deux boucles :

