

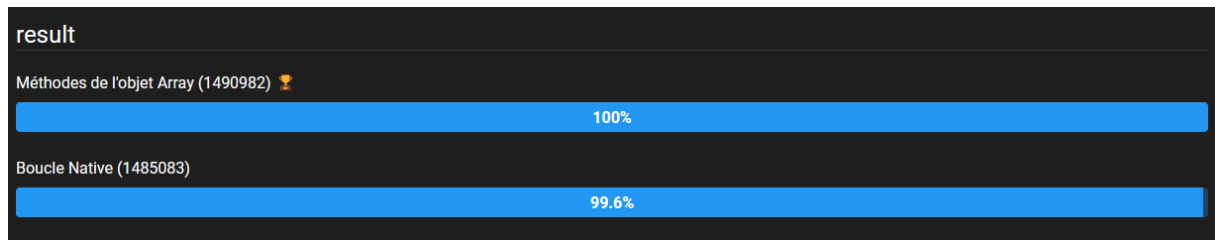
## Fiche d'investigation de fonctionnalité

<b>Fonctionnalité :</b> Rechercher et filtrer les recettes affichées sur le site	<b>Fonctionnalité #1</b>
<b>Problématique :</b> Nous devons afficher les résultats de la recherche de l'utilisateur sur l'input principal le plus rapidement possible, pour cela l'utilisateur peut saisir sur le champ principal des lettres/mots c'est alors que la page s'actualise en fonction des correspondances avec les titres / descriptions / ingrédients / appareils et ustensiles de chaque recette. L'utilisateur peut également utiliser des filtres (ingrédients / appareils et ustensiles) pour affiner sa recherche, il peut également commencer par rechercher des recettes par des filtres et dispose d'un input dans chaque un d'eux pour trouver un élément.	

<b>Option 1 :</b> Utilisation d'une boucle (while, for...)	
Nous allons coder et utiliser une boucle native pour traiter nos tableaux et afficher les recettes à l'utilisateur	
<b>Avantages :</b> <ul style="list-style-type: none"><li>- Boucle qui peut être interrompue et contrôlée</li><li>- Coder un algorithme sans avoir nécessairement besoin de connaissances approfondies en Javascript</li></ul>	<b>Inconvénients :</b> <ul style="list-style-type: none"><li>- Le code peut être moins lisible</li></ul>
<b>Nombre de champs à remplir :</b> 1	

<b>Option 2 :</b> Utilisation des méthodes de l'objet array (foreach, filter, map, includes ...)	
Nous allons coder et utiliser les méthodes de l'objet Array pour traiter nos tableaux et afficher les recettes à l'utilisateur	
<b>Avantages :</b> <ul style="list-style-type: none"><li>- Ne modifie pas le tableau sur lequel la méthode est appelée</li><li>- Créer un algorithme lisible et simple dans son utilisation</li></ul>	<b>Inconvénients :</b> <ul style="list-style-type: none"><li>- Nécessite une certaine compréhension de l'utilisation de ces méthodes d'objet</li><li>- Boucle qui ne peut être interrompue</li></ul>
<b>Nombre de champs à remplir :</b> 1	

Les résultats du test des algorithmes sur JSBENCH.CH sont les suivants :



**Solution retenue :**

Nous pouvons voir à la suite de ce test que le second algorithme est le plus rapide dans son exécution. La différence entre nos deux options est très faible mais avec mes connaissances actuel de JavaScript je privilégie l'utilisation des méthodes de l'objet Array dans la construction de mes algorithmes pour une meilleure lisibilité et maintenabilité du code.

## Annexe

