

Fonctionnalité : Recherche de recette	Fonctionnalité 1
Problématique: Faciliter la recherche de recette selon des critères d'ingrédients, d'appareils ou d'ustensiles	

Option 1 : Boucle for Ici nous utiliserons uniquement des boucles for ou while, son avantage est que son processus de construction est contrôlé par le développeur du début à la fin.	
Avantages : <ul style="list-style-type: none"> • Contrôle du code de bout en bout 	Inconvénient : <ul style="list-style-type: none"> • Perte de lisibilité dans le code • Dépendant de la complexité de la recherche
Nom de champ minimum à remplir : 1 "Rechercher une recette"	

Option 2 : Boucle native javascript Ici, nous utiliserons des boucles natives au code javascript (.forEach) avec des conditions .includes, facilitant l'écriture, la compréhension et la maintenance du code dans le temps	
Avantages : <ul style="list-style-type: none"> • Facilité de mise en place • Maintenabilité 	Inconvénient : <ul style="list-style-type: none"> • Structure de data à formater en amont
Nom de champ minimum à remplir : 1 "Rechercher une recette"	

Solution retenue : Option 2 nous avons retenu, l'utilisation des boucle native javascript pour des raisons de maintenabilité du code dans le temps avec des performances équivalentes aux boucles for l'outil de benchmark Perflink nous donne le nombre d'opération par seconde suivant : <ul style="list-style-type: none"> • Option 1 : 702 ops/s • Option 2 : 736 ops/s
--