

Fiche d'investigation de fonctionnalité

Fonctionnalité : Algorithme 1/ Algorithme 2 de recherche en javascript	Fonctionnalité #1
Problématique : Afin d'obtenir une vitesse de recherche la plus optimale possible, nous mettons en concurrence 2 algorithmes de recherche, l'une avec les boucles for et les conditions if..else if, et, l'autre en utilisant des fonctions natives, forEach, map, filter. Pour se faire, nous les testerons sur jsBench.	

Algorithme 1 : Utilisation des fonctions natives Dans cette option, nous avons une single page dédiée à la recherche d'une recette parmi 50, via un input principal et/ou via des mots clés placés dans trois listes sous les catégories : ingrédients, appareil, et ustensiles. L'avantage du système c'est l'effet d'entonnoir, au fur et à mesure de la manipulation, le choix de l'utilisateur se rétrécit jusqu'à obtention de la réponse recherchée.	
Avantages <ul style="list-style-type: none"> ⊕ Choix multiples pour procéder à la recherche ⊕ Meilleurs résultats sur jsBench (cf Figure) 	Inconvénients <ul style="list-style-type: none"> ⊖ N'accepte pas les caractères accentués ⊖ Nécessite une base de données propres, sans doublons, ni erreurs.
Nombre de champs de recherche : 4 Input principal, un input dans chacun des 3 dropdowns : Ingrédients, Appareil, Ustensiles.	

Algorithme 2 : Utilisation des boucles for et des conditions if... L'objectif est le même que l'Algorithme 1, ainsi que sa fonctionnalité.	
Avantages <ul style="list-style-type: none"> ⊕ Choix multiples pour procéder à la recherche 	Inconvénients <ul style="list-style-type: none"> ⊖ N'accepte pas les caractères accentués ⊖ Nécessite une base de données propres, sans doublons, ni erreurs. ⊖ Résultats moins probants sur jsBench
Nombre de champs de recherche : 4 Input principal, un input dans chacun des 3 dropdowns : Ingrédients, Appareil, Ustensiles	

Solution retenue : Nous avons donc retenu l'algo1, car il présente de meilleurs résultats sous jsBench. Sa vitesse d'exécution est plus rapide, l'algo 1 demandera moins de ressources, ne ralentira pas la page. Cette différence ne se voit pas à l'œil nu avec une base de données de 50 recettes, mais si elle venait à grandir de 500 ou plus, un ralentissement se fera beaucoup plus ressentir avec l'Algo 2. Cet exercice démontre que les fonctions natives doivent être utilisées en priorité.
--