

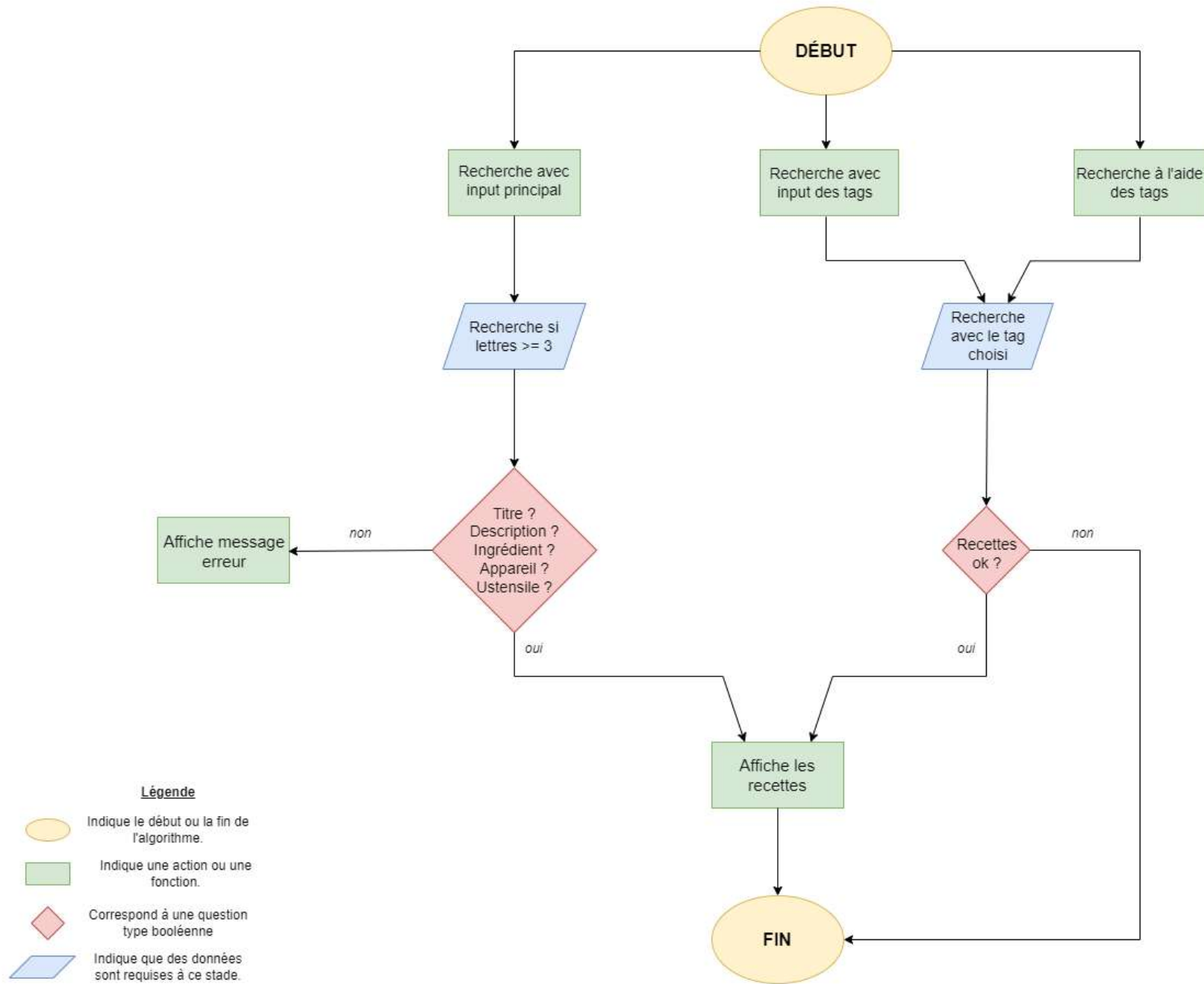
## Fiche d'investigation de fonctionnalité

<b>Fonctionnalité</b> : Recherche input	<b>Fonctionnalité #1</b>
<b>Problématique</b> : Filtrer les recettes selon ce que l'utilisateur a inscrit dans le champ recherche	

<b>Option 1</b> : Native Dans cette version nous avons utilisé les boucles natives (while, for...)	
<b>Avantages</b> <ul style="list-style-type: none"><li>- Plus facile à mettre en place en tant que débutant</li></ul>	<b>Inconvénients</b> <ul style="list-style-type: none"><li>- Réinvention de la roue, en moins bien</li><li>- Code plus long</li></ul>
<b>Nombre de champs minimum à remplir</b> : 1 champs recherche	

<b>Option 2</b> : Fonctionnelle Dans cette version nous avons utilisé les méthodes de l'objet array (foreach, filter, map, reduce).	
<b>Avantages</b> <ul style="list-style-type: none"><li>- Concis, lisible et maintenable</li><li>- Réutilisation du code</li></ul>	<b>Inconvénients</b> <ul style="list-style-type: none"><li>- Plus difficile à comprendre en tant que débutant</li><li>- Nécessite des connaissances sur le fonctionnement des différentes méthodes</li></ul>
<b>Nombre de champs minimum à remplir</b> : 1 champs recherche	

<b>Solution retenue</b> : La version utilisant la programmation fonctionnelle avec les méthodes de l'objet array semble être la meilleure option dans la plupart des cas. Les résultats obtenus par un outil de comparaison de performance (Jsben.ch) pointent également dans la même direction. Nous avons donc retenu la 2e version comme étant la plus performante pour ce projet.
---



JSBEN.CH

BENCHMARKBROWSEDONATE

Comparaison de performance entre les 2 versions

Setup block (useful for function initialization. It will be run before every test, and is not part of the benchmark.)

boilerplate block (code will executed before every block and is part of the benchmark. use it for data initializing.)

Version Native

```
1 // Import 2 function for jsbench's test
2 function displayNoResult(searchTerm) {
3   const parentElt = document.querySelector('.header__searchbar');
4   const noResultsDiv = document.createElement('div');
5   noResultsDiv.classList.add('no-result');
6   const textNoResult = document.createElement('p');
7   textNoResult.innerText = `« Aucune recette ne contient '${searchTerm}', vous pouvez
8   chercher «tarte aux pommes », « poisson », etc...`;
9   noResultsDiv.appendChild(textNoResult);
10  parentElt.appendChild(noResultsDiv);
11 }
12
```

Version Fonctionnelle

```
1 // Responsible for filtering recipes based on search terms and tags.
2 class SearchFilter {
3   constructor(dataRecipes) {
4     this.fullRecipesData = dataRecipes;
5     this.recipesWithTag = [];
6   }
7   filterRecipesByTag(tag) {
8     return this.fullRecipesData.filter(recipe => recipe.tags.includes(tag));
9   }
10  filterRecipesBySearchTerm(searchTerm) {
11    return this.fullRecipesData.filter(recipe => recipe.title.toLowerCase().includes(searchTerm.toLowerCase()));
12  }
13  filterRecipesByTagAndSearchTerm(tag, searchTerm) {
14    return this.filterRecipesByTag(tag).filter(recipe => recipe.title.toLowerCase().includes(searchTerm.toLowerCase()));
15  }
16  getRecipesWithTag(tag) {
17    return this.recipesWithTag;
18  }
19  setRecipesWithTag(recipes) {
20    this.recipesWithTag = recipes;
21  }
22 }
```

result

Version Fonctionnelle (1613093) 🏆

100%

Version Native (1385069)

85.86%

If you like to donate (Thank you!):

Ethereum (ETH)

Chia (XCH)

Cardano (ADA)

Ravencoin (RVN)

Bitcoin (BTC)

Ripple (XRP)

Litecoin (LTC)

Monero (XMR)

Dogecoin (DOGE)