

no title (put title and/or keywords here, which describes your test)

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.)

## 1. For + While

```
1  const recipesDisplayed = [];
2
3  for (let i = 0; i < RECIPES_DATABASE.length; i++) {
4    const recipe = RECIPES_DATABASE[i];
5    let isFound = false;
6
7    // Teste si le titre contient l'input
8    isFound = recipe.name.toLowerCase().includes(input.toLowerCase());
9
10   // Teste si la description contient l'input
11   if (!isFound) {
12     isFound = recipe.description.toLowerCase().includes(input.toLowerCase());
13
14     // Teste si un ingrédient contient l'input
15     if (!isFound) {
16       let j = 0;
17       while (!isFound && j < recipe.ingredients.length) {
18         isFound = recipe.ingredients[j].ingredient
19           .toLowerCase().includes(input.toLowerCase());
20       }
```



## 2. Filter + Find

```
1  const recipesDisplayed = RECIPES_DATABASE.filter((recipe) => {
2    let isFound = false;
3
4    // Teste si le titre contient l'input
5    isFound = recipe.name.toLowerCase().includes(input.toLowerCase());
6
7    // Teste si la description contient l'input
8    if (!isFound) {
9      isFound = recipe.description.toLowerCase().includes(input.toLowerCase());
10
11     // Teste si un ingrédient contient l'input
12     if (!isFound) {
13       isFound = recipe.ingredients.find((element) => {
14         return element.ingredient.toLowerCase().includes(input.toLowerCase());
15       });
16     }
17   })
18
19   // Retourne le test pour la méthode filter et la construction du tableau résultat
```

### 3. For + Find

```
1  const recipesDisplayed = [];  
2  
3  for (let i = 0; i < RECIPES_DATABASE.length; i++) {  
4    const recipe = RECIPES_DATABASE[i];  
5    let isFound = false;  
6  
7    // Teste si le titre contient l'input  
8    isFound = recipe.name.toLowerCase().includes(input.toLowerCase());  
9  
10   // Teste si la description contient l'input  
11   if (!isFound) {  
12     isFound = recipe.description.toLowerCase().includes(input.toLowerCase());  
13  
14     // Teste si un ingrédient contient l'input  
15     if (!isFound) {  
16       ⚠ isFound = recipe.ingredients.find((element) => {  
17         return element.ingredient.toLowerCase().includes(input.toLowerCase());  
18       });  
19     }  
20   }
```

### 4. Filter + While

```
1  const recipesDisplayed = RECIPES_DATABASE.filter((recipe) => {  
2    let isFound = false;  
3  
4    // Teste si le titre contient l'input  
5    isFound = recipe.name.toLowerCase().includes(input.toLowerCase());  
6  
7    // Teste si la description contient l'input  
8    if (!isFound) {  
9      isFound = recipe.description.toLowerCase().includes(input.toLowerCase());  
10  
11     // Teste si un ingrédient contient l'input  
12     if (!isFound) {  
13       let j = 0;  
14       while (!isFound && j < recipe.ingredients.length) {  
15         isFound = recipe.ingredients[j].ingredient  
16           .toLowerCase()  
17           .includes(input.toLowerCase());  
18         j++;  
19       }  
20     }
```

RUN TESTS

GENERATE PAGE URL

NEW BENCHMARK

# result

1. For + While (82158) 🏆

100%

2. Filter + Find (81370)

99.04%

3. For + Find (81284)

98.94%

4. Filter + While (80287)

97.72%