

foor of loop vs reduce method

v1

- by

-

compares the fastest algorithm with a database of 200 recipes

+ Setup HTML - click to add setup HTML

Setup JavaScript

```
const data = [
  {
    "id": 1,
    "name" : "Limonade de Coco",
    "servings" : 1,
    "ingredients": [
      {
        "ingredient" : "Lait de coco",
        "quantity" : 400,
        "unit" : "ml"
      }
    ]
  }
]
```

reduce method

finished

37436251.35 ops/s ± 8.72%
15.41 % slower

```
checkValues_FilterRecipes = () => {
  const searchbar = document.querySelector('form .search-bar');
  let searchbarValue = '';
  if(searchbar.value.length >= 3) {
    searchbarValue = searchbar.value.toLowerCase().trim();
  }

  this.filteredResult = data.reduce((filteredRecipes, currentRecipe) => {
    let matchIngredient, matchAppliance, matchUstensil;

    if(this.currentTags.ingredients.every(tagIngredient =>
      currentRecipe.ingredients.some(ingredient => ingredient.toLowerCase().includes(tagIngredient.toLowerCase())))
```

DEFER

for of loop

finished

44255877.92 ops/s ± 7.11%
Fastest

```
checkValues_FilterRecipes = () => {
  this.filteredResult = new Set();
  const searchbar = document.querySelector('form .search-bar');
  let searchbarValue = '';
  if(searchbar.value.length >= 3) {
    searchbarValue = searchbar.value.toLowerCase().trim();
  }

  for(let currentRecipe of data) {
    let matchIngredient, matchAppliance, matchUstensil;

    // si une recette contient tous les tag ET le mot clé dans la barre de recherche => matchIngredient = true
```

DEFER

+ Test Case - click to add another test case

+ Teardown JS - click to add teardown JavaScript

+ Output (DOM) - click to monitor output (DOM) while test is running

▶ RUN again