

foor of loop vs reduce method

v1

- by

-

compares the fastest algorithm with a database of 50 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

49181181.85 ops/s ± 9.59%

Fastest

```
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()))
    ) {
      matchIngredient = true;
    }
  })
}
```

☐ DEFER

for of loop

finished

41067872.35 ops/s ± 10.76%

16.5 % slower

```
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

[Wiki](#) | [Report issue](#) | [Become a sponsor!](#)

Inspired by [Benchmark.js](#), [Jsperf.com](#) and [Jsfiddle.com](#).

v1.1.0-1 prod