**Setup data**

The “recipes” array was also included.

const normalize = (text) => { return text.normalize('NFD').replace(/[\u0300-\u036f]/g, '').toLowerCase(); };

let recipesLeftArray =[] ;

let NormalizedInput = normalize('***testNameHere***');

**Code Blocks:**

**ForEach:**

recipes.forEach((recipe) => {

let NormalizedRecipeName = normalize(recipe.name);

let NormalizedDescription = normalize(recipe.description);

if (NormalizedRecipeName.includes(NormalizedInput) || NormalizedDescription.includes(NormalizedInput)) {

recipesLeftArray.push(recipe);

} else {

recipe.ingredients.forEach((item) => {

let NormalizedIngredient = normalize(item.ingredient);

if (NormalizedIngredient.includes(NormalizedInput)) {

recipesLeftArray.push(recipe);

}

});

}

});

**Filter:**

let sortedrecipesLeftArray = recipes.filter(recipe => normalize(recipe.name).includes(NormalizedInput) ||

normalize(recipe.description).includes(NormalizedInput) ||

recipe.ingredients.some(i => normalize(i.ingredient).includes(NormalizedInput)));

**For Loop:**

for (let i = 0; i < recipes.length; i++) {

/\* clean recipe data for search \*/

let NonNormalizedName = recipes[i].name;

let NormalizedName = normalize(NonNormalizedName);

let NonNormalizedDescription = recipes[i].description;

let NormalizedDescription = normalize(NonNormalizedDescription);

if (NormalizedName.includes(NormalizedInput) || NormalizedDescription.includes(NormalizedInput)) {

recipesLeftArray.push(recipes[i]);

} else {

for (let j = 0; j < recipes[i].ingredients.length; j++) {

let NonNormalizedIngredient = recipes[i].ingredients[j].ingredient;

let NormalizedIngredient = normalize(NonNormalizedIngredient);

if (NormalizedIngredient.includes(NormalizedInput)) {

recipesLeftArray.push(recipes[i]);

}

}

}

}