



Les petits plats

**Développez un
algorithme de recherche
en JavaScript**

Développez un algorithme de recherche en JavaScript

Soutenance du samedi 22 octobre 2022

Table des matières :

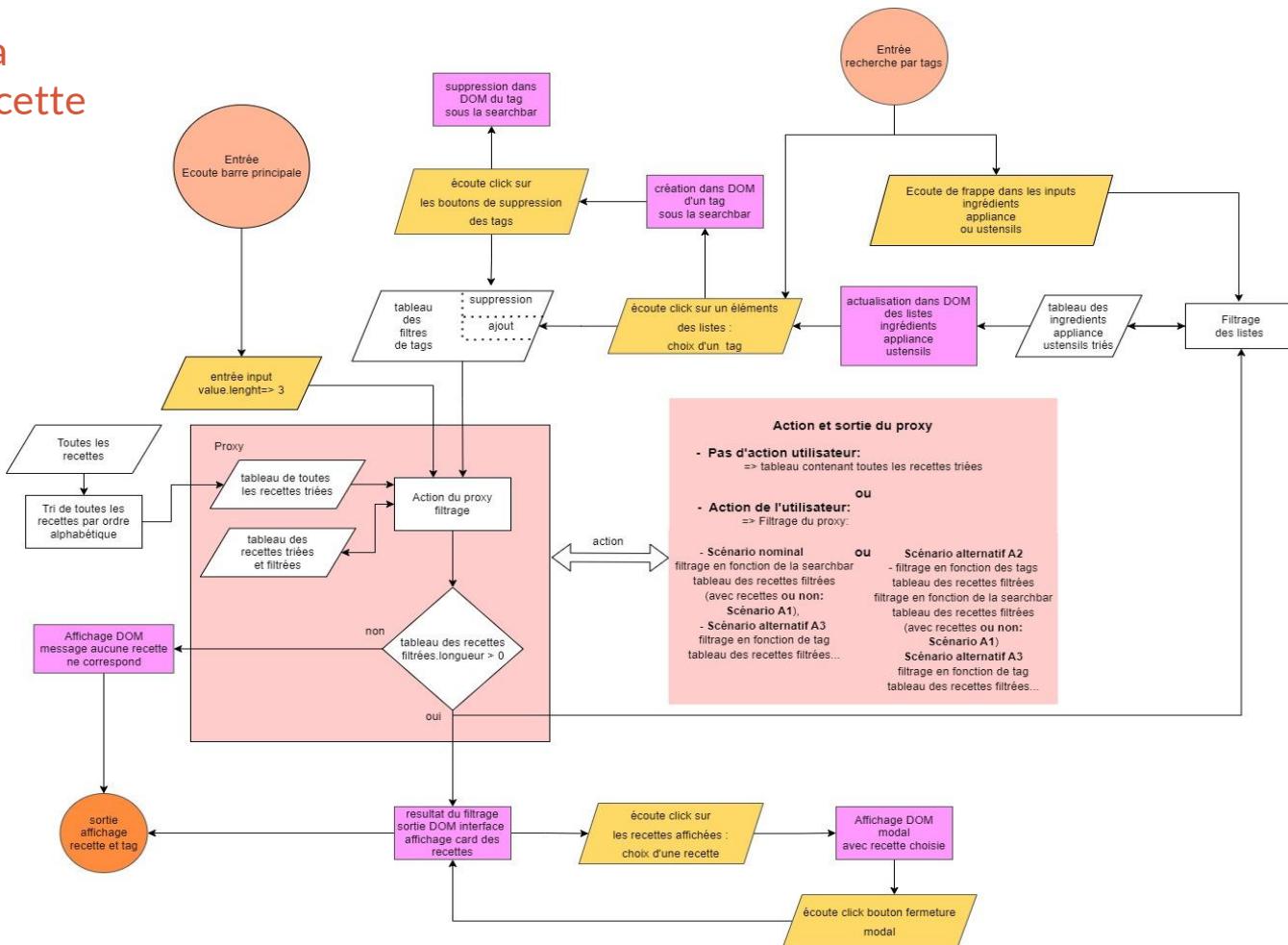
- Objectif du projet
- et Spécifications fonctionnelles et Techniques
- Création du prototype :
 - Mise en œuvre pratique de la recherche de recettes
 - Vue d'ensemble du prototype réalisé avec démonstration
 - rendu visuel du prototype et sa correspondance avec la maquette
 - Notices complémentaires : résultats des tests
 - Validator W3C
 - Conclusion

Objectif du projet et Spécifications fonctionnelles et Techniques :

L'entreprise Les petits plats souhaite réaliser son propre site de recettes de cuisine.
L'objectif est d'optimiser la fonctionnalité de recherche.

- le prototype doit correspondre à la maquette (utilisation possible de bootstrap 4 pour le style)
- deux implémentations différentes vont être créées quant à la recherche principale
 - une version utilisant les boucles natives (while, for...),
 - une version avec les méthodes de l'objet array (foreach, filter, map, reduce). pour pouvoir comparer leurs performances et choisir la meilleure :
- une fiche d'investigation et l'algorithme créés,
- utilisation de jsben.ch pour effectuer les tests et choisir la plus performante.

Algorithme de la recherche de recette



Création d'un objet Proxy

Mise à jour de l'affichage des recettes (tableau recettes filtrées)
si vide => Scénario A1 : message aucune recette,
Mise à jour des listes pour le choix des tags de filtrage par rapport aux nouveaux résultats de recettes filtrées

=> Scénario nominal : Entrée de l'input de la barre principale
puis si approfondissement de la recherche par tag => scénario A3...

Scénario A2 : Entrée du choix d'un tag ou de sa suppression
puis si approfondissement de la recherche par tag => scénario A3...

```
let recipesSort = quickSort(recipes, 0, recipes.length - 1);
let datas = {};
datas.recipes = [...recipesSort]
```

```
datasProxy.filteredRecipes = [...recipesSort];
```

```
let datasProxy = new Proxy(datas, {
  set: function(target, key, value) {
    target[key] = value;
    switch(key) {
      case 'filteredRecipes':
        if ( datasProxy.filteredRecipes.length == 0){
          noRecipeAlert();
        } else {
          if(isAlert){
            removeNoRecipeAlert();
          }
          displayRecipes(value);
          //updating the list of lists of possible tags
          getIngredientsList(value);
          getApplianceList(value);
          getUstensilsList(value);
          getChosenTag();
        }
        break;
      case 'searchString': {
        //filtering based on searchBar
        const result = searchRecipeBySearchBar(value);
        //updating the list of filtered recipes
        if (result) {
          datasProxy.filteredRecipes = [...result];
        }
        //tag recovery
        searchByTag();
        break;
      }
      case 'searchTag' :
        //creation of the chosen tag
        displayTag(datasProxy.searchTag);
        //filtering based on tags
        searchByTag();
        removeTag();
        //searchbar recovery
        if (datas.searchString?.length > 2) {
          const result = searchRecipeBySearchBar(datas.searchString);
          datasProxy.filteredRecipes = [...result];
        }
        break;
    }
    return true;
  }
});
```

Suppression des biais orthographiques et de casse (dus à l'utilisateur et aux propriétés du fichier de recette)

Correction également de faute d'orthographe présente dans le fichier de recette
(Ex : casserolle et poelle)

```
export function getStringForCompare(string) {  
    return string  
        .toLowerCase()  
        .replace(/[\àäâ]/g, "a")  
        .replace(/[\éèêë]/g, "e")  
        .replace(/[']/g, "")  
        .replace(/[.,"]]/g, "")  
        .replace(/[\d]/g, "")  
        .replace(/[\ç]/g, "c")  
        .replace(/[\îï]/g, "i")  
        .replace(/[\ôö]/g, "o")  
        .replace(/[\ùûû]/g, "u");  
}
```

Tri des recettes en fonction de leur nom par ordre alphabétique

Algorithme dit de « tri rapide » (« QuickSort ») : afin d'optimiser le tri « diviser pour régner »

```
function swap(recipesforsort, leftIndex, rightIndex){  
    var temp = recipesforsort[leftIndex];  
    recipesforsort[leftIndex] = recipesforsort[rightIndex];  
    recipesforsort[rightIndex] = temp;  
}  
  
function partition(recipesforsort, left, right) {  
    let pivot = recipesforsort[Math.floor((right + left) / 2)],  
        i = left, //left pointer  
        j = right; //right pointer  
    while (i <= j) {  
        while (getStringForCompare(recipesforsort[i].name) < getStringForCompare(pivot.name)) {  
            i++;  
        }  
        while (getStringForCompare(recipesforsort[j].name) > getStringForCompare(pivot.name)) {  
            j--;  
        }  
        if (i <= j) {  
            swap(recipesforsort, i, j); //swapping two elements  
            i++;  
            j--;  
        }  
    }  
    return i;  
}
```

```
export function quickSort(recipesforsort, left, right) {  
    let index;  
    if (recipesforsort.length > 1) {  
        index = partition(recipesforsort, left, right); //index returned from partition  
        if (left < index - 1) { //more elements on the left side of the pivot  
            quickSort(recipesforsort, left, index - 1);  
        }  
        if (index < right) { //more elements on the right side of the pivot  
            quickSort(recipesforsort, index, right);  
        }  
    }  
    return recipesforsort;  
}
```

```
let recipesSort = quickSort(recipes, 0, recipes.length - 1);
```

Utilisation de jsben.ch afin
d'optimiser chacune des
implémentations différentes

Utilisation d'un fichier de 1000 recettes afin que le test soit représentatif

test sur le passage de toutes les recettes au filtrage (input.length < 3 à l' input.length > 2), puis de 3 à 4 lettres pour revenir ensuite à 3 lettres à nouveau et donc au filtrage sur toutes les recettes.

```
let datas = {};
datas.recipes = [...recipesSort];

let datasProxy = new Proxy(datas, {
  set: function(target, key, value) {
    target[key] = value;
    switch(key) {
      case 'searchString': {
        const result = searchRecipeByearchBar(value);
        if(result){
          datasProxy.filteredRecipes = [...result];
        }
        break;
      }
    }
    return true;
  }
});
```

Limitation du proxy à la recherche principale

```
datasProxy.filteredRecipes = [...recipesSort];
datasProxy.searchLength = 5;
datasProxy.searchString = 'ananas';
```

Exemple de lancement de test

Test de la fonction par boucle for

JSBEN.CH

test function bouclefor (et variable intermédiaire pour recipe, name et description) function1: sans - function2: avec 

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

```
22      },
23      {
24          "ingredient" : "Sucre",
25          "quantite" : 30,
26          "unit" : "grammes"
27      },
28      {
29          "ingredient": "Glaçons"
30      }
31 ],
32 "time": 10,
33 "description": "Mettre les glaçons à votre soupe dans la
```

[ADD LIBRARY](#)

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

[RUN TESTS](#)

[GENERATE PAGE URL](#)

BENCHMARK

BROWSE

DONATE

[NEW BENCHMARK](#)

result

code block 2 (5099) 🏆

100%

code block 1 (5048)

99%



JSBEN.CH

(sur l'utilisation de variables intermédiaires)

(sur l'utilisation d'une fonction intermédiaire)

test function bouclefor (et function intermédiaire) function1: sans - function2: avec 

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

```
43     },
44     {
45         "ingredient" : "Thon Rouge (ou blanc)",
46         "quantity" : 200,
47         "unit" : "grammes"
48     },
49     {
50         "ingredient" : "Concombre",
51         "quantity" : 1
52     },
53     {
54         "ingredient" : "Tomate",
55         "quantity" : 2
56     }
```

[ADD LIBRARY](#)

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

[RUN TESTS](#)

[GENERATE PAGE URL](#)

BENCHMARK

BROWSE

DONATE

[NEW BENCHMARK](#)

result

code block 2 (4989) 🏆

100%

code block 1 (4819)

96.59%



LEARN MORE

Implémentation choisie du filtrage par boucle native for

```
//fonction intermédiaire de recherche searchbar
function ingredientIsHere(recipe, value){
  for (let i = 0; i < recipe.ingredients.length; i++) {
    if(getStringForCompare(recipe.ingredients[i].ingredient).includes(value)){
      return true;
    }
  }
  return false;
}

//fonction intermédiaire de recherche searchbar
function searchStringInAllRecipe(research,arrayOfRecipe,arrayOfResult){
  research = getStringForCompare(research);
  for (let i = 0; i < arrayOfRecipe.length; i++) {
    let recipe = arrayOfRecipe[i];
    let name = getStringForCompare(recipe.name);
    let description = getStringForCompare(recipe.description);
    if (name.includes(research)) {
      arrayOfResult.push(recipe);
    } else if ( ingredientIsHere(recipe, research)){
      arrayOfResult.push(recipe);
    } else if (description.includes(research)) {
      arrayOfResult.push(recipe);
    }
  }
  return arrayOfResult;
}
```

Fonction intermédiaire pour le filtrage en profondeur

Insertion de raccourci dans le filtrage par des if et else if

```
function searchRecipeBySearchBar(research){
  let result = [];
  if(research.length >= datasProxy.searchLength && research.length > 2) {
    return searchStringInAllRecipe(research,datasProxy.filteredRecipes,result);
  }else if (research.length < datasProxy.searchLength && research.length > 2) {
    return searchStringInAllRecipe(research,datasProxy.recipes,result);
  } else if (research.length <= datasProxy.searchLength && research.length == 2){
    const result = [...recipesSort];
    return result;
  } else if(research.length < 2 || (research.length > datasProxy.searchLength && research.length == 2)) {
    return;
  }
}
```

Test de la fonction par la méthode d'array filter

(sur l'utilisation de test successif dans la callback, et d'une function intermédiaire)

JSBEN.CH

BENCHMARK BROWSE DONATE

test filter avec fonction1: opérateur || et fonction2: if, else if [✎](#)

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

```
34      },
35      {
36          "id": 2,
37          "name": "Poisson Cru à la tahitienne",
38          "servings": 2,
39          "ingredients": [
40              {
41                  "ingredient": "Thon Rouge (ou blanc)",
42                  "quantity": 200,
43                  "unit": "grammes"
44              },
45              {
46      }
```

[ADD LIBRARY](#)

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

```
9      .replace(/[\ii]/g, "i")
```

RUN TESTS GENERATE PAGE URL NEW BENCHMARK

result

code block 2 (4992) 🏆

100%

code block 1 (4932)

98.8%



Implémentation choisie du filtrage par la méthode d'array filter

```
//fonction intermédiaire de recherche searchbar
function ingredientIsHere(recipe, research){
    if(recipe.ingredients.filter(item =>
        getStringForCompare(item.ingredient).includes(research)).length > 0){
            return true;
    } else return false;
}

//fonction intermédiaire de recherche searchbar
function searchStringInAllRecipe(recipe,research){
    if (getStringForCompare(recipe.name).includes(research)) {
        return true;
    } else if (ingredientIsHere(recipe, research)){
        return true;
    } else if (getStringForCompare(recipe.description).includes(research)){
        return true;
    }
}
```

Fonction intermédiaire pour le filtrage en profondeur sur les différents ingrédients de chaque recette

Insertion de raccourci dans le filtrage par des if et else if

```
function searchRecipeBySearchBar(research){
    if(research.length >= datasProxy.searchLength && research.length > 2) {
        research = getStringForCompare(research);
        const result = datasProxy.filteredRecipes.filter(recipe => searchStringInAllRecipe(recipe,research));
        return result;
    }else if (research.length < datasProxy.searchLength && research.length > 2) {
        research = getStringForCompare(research);
        const result = datasProxy.recipes.filter(recipe => searchStringInAllRecipe(recipe,research));
        return result;
    } else if (research.length <= datasProxy.searchLength && research.length == 2){
        const result = [...recipesSort];
        return result;
    } else if (research.length < 2 || (research.length > datasProxy.searchLength && research.length == 2)) {
        return;
    }
}
```

Utilisation de jsben.ch afin d'optimiser la fonctionnalité de recherche avec le Choix de la meilleure implémentation : méthode filter

Ex de Test sur 1000 recettes : 'coc' => 'coco' => 'co'

JSBENCH

BENCHMARK BROWSE DONATE

test (1000 recettes) function1:boucle for, function2:methode filter

[RUN TESTS](#) [GENERATE PAGE URL](#) [NEW BENCHMARK](#)

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

```
34445
34446
34447     "id": 50,
34448     "name": "Frangipane",
34449     "servings": 2,
34450     "ingredients": [
34451         {
34452             "ingredient": "Pâte feuilletée",
34453             "quantity": 400,
34454             "unit": "grammes"
34455         },
34456         {
34457             "ingredient": "Oeuf",
34458         }
34459     ]
34460 }
```

[ADD LIBRARY](#)

[boilerplate block](#) (code will be executed before every block and is part of the benchmark. Use it for data initializing.)

result

code block 2 (235) 🏆

100%

code block 1 (229)

97.45%



```
datasProxy.filteredRecipes = [...recipesSort];

datasProxy.searchLength = 2;
datasProxy.searchString = 'coc';
datasProxy.searchLength = 3;
datasProxy.searchString = 'coco';
datasProxy.searchLength = 3;
datasProxy.searchString = 'co';
```

Autre ex de Test sur 1000 recettes : 'far' => 'fari' => 'far'

JSBEN.CH

BENCHMARK BROWSE DONATE

test (1000 recettes - farine3) function1:boucle for,
function2:methode filter 

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

```
34471     "ingredient": "Sucre glace",
34472     "quantity": 500,
34473     "unit": "grammes"
34474   }
34475 ],
34476   "time": 60,
34477   "description": "Préparer la frangipane : Mélanger le suc
34478   "appliance": "Four",
34479   "utensils": ["rouleau à patisserie", "fouet"]
34480 }
34481 ];
34482
```

ADD LIBRARY

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

```
1
2   datasProxy.filteredRecipes = [...recipesSort];
3
4
5   datasProxy.searchLength = 2;
6   datasProxy.searchString = 'far';
7   datasProxy.searchLength = 3;
8   datasProxy.searchString = 'fari';
9   datasProxy.searchLength = 4;
10  datasProxy.searchString = 'far';
```

result

code block 2 (117)  100%

code block 1 (116) 99.15%



5 Families That Rule the World
LEARN MORE

datasProxy.filteredRecipes = [...recipesSort];

datasProxy.searchLength = 2;

datasProxy.searchString = 'far';

datasProxy.searchLength = 3;

datasProxy.searchString = 'fari';

datasProxy.searchLength = 4;

datasProxy.searchString = 'far';

Fiche d'investigation de fonctionnalité

Fiche d'investigation de fonctionnalité

Fonctionnalité : Recherche de recettes par la barre principale	Fonctionnalité #1 (Fiche Cas d'utilisation #3)
--	--

Le filtrage se fait par la recherche d'une correspondance entre la valeur entrée par l'utilisateur dans la barre principale (lorsque supérieure ou égale à 3 lettres) et des mots ou groupes de lettres dans le titre, les ingrédients ou la description des recettes. La recherche s'actualise pour chaque nouveau caractère entré.

Problématique : Nous cherchons à avoir un filtrage des recettes le plus rapide possible. La recherche principale affiche les premiers résultats le plus rapidement possible.

Dans les deux cas, utilisation d'un proxy, le tableau des recettes triées est donc conservé pour effectuer la nouvelle recherche lorsqu'un nouveau caractère est entré. Par contre à la suppression d'un caractère, le filtrage se fait de nouveau sur le tableau de toutes les recettes.

Perte de temps supplémentaire possible en plus de l'optimisation de la fonction elle-même :

- Biais dus à la casse ou aux fautes d'orthographes de l'utilisateur lors de la recherche,

Solution => utilisation d'une fonction de correction d'orthographe et de casse

- Affichage des recettes triées par ordre alphabétique

Solution => tri des recettes directement avant toute action de l'utilisateur

Option 2 : Filtrage par l'utilisation de boucles natives de JavaScript (boucle for)

Dans cette option, on crée dans un premier temps un objet tableau qui va recevoir toutes les recettes filtrées répondant au test. On boucle successivement sur chaque recette en vérifiant sur chacune de leurs propriétés "name", "description" et "ingrédients" les conditions de test. Dès qu'une propriété répond true au test alors la boucle s'arrête et la recette est copiée dans le tableau de résultat.

Avantages

- * Méthode native de base de JavaScript

Inconvénients

- code moins compacte.
- nécessite de créer un nouveau tableau de résultats
- nécessite de mettre la recette passant le test dans le tableau de résultats

- Test jsben.ch

- test 1000 recettes : sur le passage de co à coc puis coco et co => 229 opérations par seconde 97.45%

- test 1000 recettes : sur le passage de far à fari puis far => 112 puis 116 soit 98.25% puis 99.15%

Option 1 : Filtrage par l'utilisation des méthodes de l'objet array de JavaScript.

Dans cette option, nous effectuons le filtrage des recettes par l'utilisation de la méthode filter. La méthode filter() crée et retourne un nouveau tableau contenant tous les éléments du tableau d'origine qui remplissent une condition déterminée par la fonction callback. L'avantage du système est la création d'un nouveau tableau. Et en utilisant une fonction fléchée, nous avons une fonction agréable et propre pour créer le même tableau..

Avantages

- * Création direct d'un nouveau tableau de résultats
- * Sur un tableau d'objet, le processus de filtrage reste relativement inchangé. (S'apparentant à une itération)
- * Code simplifié, méthode ayant fait ses preuves

Inconvénients

- Utilisation d'une function callback

- Test jsben.ch

- test 1000 recettes : sur le passage de 'co' à 'coc' puis 'coco' et 'co' => 235 opérations par seconde 100%

- test 1000 recettes : sur le passage de 'far' à 'fari' puis 'far' => 114 puis 117 soit 100% puis 100%

Solution retenue : Code relativement similaire. Méthodes d'array suffisamment récentes pour être parfaitement supportées. La méthodologie de filtrage retenue est celle qui utilise filter car légèrement plus rapide lorsque la recherche se fait sur une grande quantité de recette (test sur 1000 recettes). Toutefois la rapidité de la recherche passe principalement non par le choix de l'implémentation entre boucle for et méthode d'array mais plus par la préparation en amont (tel que le tri préalable des recettes utilisées par ordre alphabétique), par la correction orthographique pour la réponse la plus adéquate rendue lors du filtrage, par les raccourcis des contrôles effectués (par if et else if), ainsi que par la réutilisation du tableau des recettes déjà triées et donc nettement moins long lors de l'ajout d'une lettre par l'utilisateur, avec l'utilisation d'un proxy pour faciliter la mise à jour du tableau des recettes triées.

Traitement du filtrage par tag

Gestion des listes et du choix du tag de filtrage

```
function getIngredientsList(recipes){  
    let ingredientsArray = [];  
    recipes.forEach((recipe) => {  
        recipe.ingredients.map((element) => ingredientsArray.push  
            (element.ingredient.toLowerCase()));  
    })  
    ingredientsArray = [...new Set(ingredientsArray)].sort();  
    buildULListfilter(ingredientsArray, boxresultsUl[0]);  
}  
  
inputList[0].addEventListener('input', (e) => {  
    let research = e.target.value;  
    let newIngredientsArray = ingredientsArray.filter(item =>  
        getStringForCompare(item).includes(getStringForCompare  
            (research)));  
    buildULListfilter(newIngredientsArray, boxresultsUl[0]);  
    getChosenTag();  
})  
  
boxresults[0].addEventListener('focusin', () => {removeInputList  
    (ingredientsArray, 0)})  
btnList[0].addEventListener('click', () => {removeInputList  
    (ingredientsArray, 0)})  
  
function removeInputList(ArrayList, index){  
    if (inputList[index].value == ''){  
        buildULListfilter(ArrayList, boxresultsUl[index]);  
        getChosenTag();  
    }  
}
```

```
function getChosenTag() {  
    const liSortingItem = document.querySelectorAll('.liSorting-item')  
    liSortingItem.forEach(item => item.addEventListener('click', (e)  
    => {  
        let tag = {};  
        tag.value = e.target.textContent;  
        tag.type = e.target.parentNode.id;  
        if(e.target.parentNode.id == 'ingredientsUL'){  
            inputList[0].value = '';  
        } else if(e.target.parentNode.id == 'applianceUL'){  
            inputList[1].value = '';  
        } else if(e.target.parentNode.id == 'ustensilsUL'){  
            inputList[2].value = '';  
        }  
        datasProxy.searchTag = datasProxy.searchTag?.length > 0 ? [...  
            datasProxy.searchTag, tag] : [tag] ;  
    })  
}  
  
function removeTag(){  
    const btnCloses = document.querySelectorAll('.btnClose');  
    for(let i = 0; i < datasProxy.searchTag?.length; i++){  
        btnCloses[i].addEventListener('click', function()  
        {  
            datasProxy.filteredRecipes = [...recipesSort];  
            datasProxy.searchTag = datasProxy.searchTag.length > 1 ?  
                [...datasProxy.searchTag.slice(0, i), ...datasProxy.  
                    searchTag.slice(i + 1)] : [];  
        })  
    }  
}
```

Function de filtrage

```
//fonction intermédiaire de recherche par tag
function filterRecipeByIngredients(tag){
  const resultTag = datasProxy.filteredRecipes.filter(recipe =>
  recipe.ingredients.filter(item =>
    item.ingredient.toLowerCase().includes(tag.value.toLowerCase()
  )).length > 0)
  datasProxy.filteredRecipes = [...resultTag];
}
```

```
//fonction intermédiaire de recherche par tag
function filterRecipeByAppliance(tag){
  const resultTag = datasProxy.filteredRecipes.filter(recipe =>
  recipe.appliance.toLowerCase().includes(tag.value.toLowerCase()));
  datasProxy.filteredRecipes = [...resultTag];
}
```

```
//fonction intermédiaire de recherche par tag
function filterRecipeByUstensils(tag){
  const resultTag = datasProxy.filteredRecipes.filter(recipe =>
  recipe.ustensils.filter(item =>
    item.toLowerCase().includes(tag.value.toLowerCase())).length > 0)
  datasProxy.filteredRecipes = [...resultTag];
}
```

```
function searchByTag() {
  datasProxy.searchTag?.forEach(tag => {
    if(tag.type == 'ingredientsUl'){
      filterRecipeByIngredients(tag);
    } else if(tag.type == 'applianceUl'){
      filterRecipeByAppliance(tag);
    } else if (tag.type == 'ustensilsUl'){
      filterRecipeByUstensils(tag);
    }
  })
}
```

Rendu du prototype

Test de rendu



Les petits plats

Rechercher une recette



Ingrédients

Appareils

Ustensils



Brownie

Noix: 180 g
Chocolat noir: 150 g
Beurre: 120 g
Oeuf: 2
Sucre en Poudre: 110 g
farine: 90 g

⌚ 60 min

Hachez les noix grossièrement.
Faire fondre le chocolat avec le beurre.
Mélanger l'oeuf et le sucre et mélanger au chocolat.
Ajouter la farine. Mélanger afin d'avoir quelque chose d'homogène puis incorporer le...



Charlotte au poires

Chocolat: 200 g
Oeuf: 3
Poires au jus: 0.5 boîtes
Boudoirs: 15

⌚ 60 min

Commencez par préparer la mousse au chocolat au moins 2 heures avant. Quand la mousse est prête et a reposé. Alors mouiller les boudoirs dans le jus des poires. Disposer. Alterner : mousse au chocolat, boudoirs e...



Compote pomme rhubarbe

Rhubarbe: 160 g
Pommes: 8
Sucre vanillé: 6 sachets
Eau: 0.5 tasses

⌚ 40 min

Éplucher les fruits et les couper en morceaux, les mettre dans une casserole en ajoutant l'eau et le sucre vanillé. Laisser cuire 15 minutes en remuant régulièrement.



Cookies

Sucre: 100 g

⌚ 30 min

Faire fondre le beurre et le



Courgettes farcies au boeuf

Courgette: 2

⌚ 60 min

Couper les courgettes dans le



Crème dessert au chocolat

Lait: 1 litres

⌚ 15 min

Mélanger la farine et le beurre



Les petits plats

Rechercher une recette



Ingrediénts



Appareils



Ustensils



Brownie

⌚ 60 min

Noix: 180 g
Chocolat noir: 150 g
Beurre: 120 g
Oeuf: 2
Sucre en Poudre: 110 g
farine: 90 g

Hachez les noix grossièrement.
Faire fondre le chocolat avec le beurre.
Mélanger les œufs et le sucre et mélanger au chocolat.
Ajouter la farine. Mélanger afin d'avoir quelque chose d'homogène puis incorporer le...



Charlotte au poires

⌚ 60 min

Chocolat: 200 g
Oeuf: 3
Poires au jus: 0.5 boîtes
Boudoirs: 15

Commencez par préparer la mousse au chocolat au moins 2 heures avant. Quand la mousse est prête et a reposé. Alors mouiller les boudoirs dans le jus des poires. Disposer. Alterner : mousse au chocolat, boudoirs e...



Compote pomme rhubarbe

⌚ 40 min

Rhubarbe: 160 g
Pommes: 8
Sucré vanillé: 6 sachets
Eau: 0.5 tasses

Éplucher les fruits et les couper en morceaux, les mettre dans une casserole en ajoutant l'eau et le sucre vanillé. Laisser cuire 15 minutes en remuant régulièrement.



Cookies

⌚ 30 min

Sucre: 100 g

Faire fondre le beurre et le



Courgettes farcies au boeuf

⌚ 60 min

Courgette: 2

Couper les courgettes dans le



Crème dessert au chocolat

⌚ 15 min

Lait: 1 litres

Mélanger la farine et le beurre

Test de rendu

Maquette

 Les petits plats

Rechercher une recette

K P E

Coco (0) Ingredients (0) Appareils (0) Ustensiles (0)

Limonade de Coco (10 min) Poisson Cru à la tahitienne (60 min) Poulet coco réunionnais (80 min)

Mettre les ingrédients à votre goût dans le blender, ajouter le sel, la crème de coco, le jus de 2 citrons et le sucre. Mélanger jusqu'à avoir la consistance désirée.

Thon Rouge (ou blanc): 200g Oeufs: 1 Couveuse: 1 Tomate: 2 Carotte: 1 Oignon: 1 Peperon rouge: 1 Lait de coco: 100ml Huile d'olive: 1

Découper le thon en dés, mettre dans un plat et recouvrir de jus de citron vert (Mélanger tout pour que le jus pénètre bien dans le poisson). Laisser reposer au réfrigérateur au moins 2h.

Salade de riz (50 min) Tarte au thon (45 min) Tarte aux pommes (50 min)

Prototype

 Les petits plats

Rechercher une recette

Farine (0) Ingrédients (0) Appareils (0) Ustensiles (0)

Brownie (60 min) Cookies (30 min) Crème dessert au chocolat (15 min)

Mettre les ingrédients grossièrement. Faire fondre le chocolat avec le beurre. Mélanger les oeuf et le sucre et mélanger au chocolat. Ajouter la farine. Mélanger afin d'avoir quelque chose d'homogène puis incorporer le beurre.

Noix: 150 g Chocolat noir: 150 g Beurre: 120 g Oeuf: 2 Sucre en Poudre: 110 g farine: 90 g

Faire fondre le beurre et le mélanger avec le sucre. Finallement ajouter l'oeuf. Ajouter la farine tout en mélangeant peu à peu pour avoir une masse sans grumeaux. Ajouter les pépites de chocolat.

Lait: 1 litre Chocolat: 200 g Sucre: 100 g Beurre: 50 g farine: 40 g

Mélanger la farine et le beurre fondus en ajoutant le lait peu à peu. Ajouter du sucre après la cuisson. Bien mélanger. Ajouter le chocolat en morceaux et laisser chauffer 8 minutes en mélangeant avec une cuillère.

Test de rendu

Maquette


Les petits plats

Coco

Appareils Ustensiles

Limonade de Coco ⏳ 10 min
Mettez les glaçons à votre goût dans le blender, ajoutez le jus, la crème de coco, le jus de citron et le sucre. Mélangez jusqu'à avoir la consistance désirée.

Poisson Cru à la tahitienne ⏳ 60 min
Filet Rouge (ou blanc): 200g Concombre: 1 Tomate: 2 Cendre: 1 Oignon: 1 Poivre: 1 Huile d'olive: 100ml Découpez le poisson en morceaux, les faire cuire dans un plat et recouvrir de jus de citron vert (mais pas trop pour éviter que le poisson ne soit trop sec). Laisser reposer au réfrigérateur au moins 2...

Poulet coco réunionnais ⏳ 80 min
Poulet: 5 Lait de coco: 400ml Coulis de tomate: 25cl Oignon: 1 Poivre: 1 Huile d'olive: 1 Découper le poulet en morceaux, les faire cuire dans une cocotte avec de l'huile d'olive et du lait de coco. Une fois doré, laisser cuire en ajoutant du feu. Au bout de...

Salade de riz ⏳ 50 min
Lait de coco: 500ml Jus de citron: 2 Crème de coco: 4 cuillères Sucre: 20g Glaçons: 2

Tarte au thon ⏳ 45 min
Thon Rouge (ou blanc): 200g Concombre: 1 Tomate: 2 Cendre: 1 Oignon: 1 Poivre: 1 Huile d'olive: 100ml Découper le poisson en morceaux, les faire cuire dans un plat et recouvrir de jus de citron vert (mais pas trop pour éviter que le poisson ne soit trop sec). Laisser reposer au réfrigérateur au moins 2...

Tarte aux pommes ⏳ 50 min
Sucre: 100 g Farine: 150 g Beurre: 120 g Oeuf: 3 Cuire au four à 180°C pendant 45 minutes. Couper la tarte en quartiers et servir.

Prototype


Les petits plats

Rechercher une recette

coc

Appareils Ustensiles

Crème de coco Lait de coco


Brownie ⏳ 60 min
Noix: 180 g Chocolat noir: 150 g Beurre: 120 g Oeuf: 3 Cuire au four à 180°C pendant 45 minutes. Couper la tarte en quartiers et servir.


Charlotte au poires ⏳ 60 min
Chocolat: 200 g Oeuf: 3 Poires au jus: 0,5 boîtes Boudoirs: 15 Commencez par préparer la mousse au chocolat et laissez-la reposer 2 heures au réfrigérateur. Quand la mousse est tendre et la poisseuse, alors mouiller les boudoirs dans le jus des poires. Disposer. Alterner : mousse au chocolat, boudoirs e...


Compote pomme rhubarbe ⏳ 40 min
Rhubarbe: 160 g Pommes: 8 Sucre vanillé: 6 sachets Eau: 0,5 tasses Eplucher les fruits et les couper en morceaux, les mettre dans une casserole en ajoutant l'eau et le sucre vanillé. Laisser cuire 15 minutes en remuant régulièrement.


Cookies ⏳ 30 min
Sucre: 100 g Faire fondre le beurre et le


Courgettes farcies au boeuf ⏳ 60 min
Courgette: 2 Couper les courgettes dans le


Crème dessert au chocolat ⏳ 15 min
Lait: 1 litres Mélanger la farine et le beurre

Test de rendu

Maquette

 Les petits plats

Coco

Rechercher un ingrédient

Lait de coco	Jus de citron	Crème de coco
Sucre	Glaçons	Thon Rouge
Concombre	Tomate	Carotte
Citron Vert	Poulet	Coulis de tomate
Oignon	Poivron rouge	Huile d'olive
Riz blanc	Thon en miettes	Oeuf dur
Mais	Vinaigrette	Pâte feuilletée
Crème fraîche	Gruyère râpé	Moutarde de Dijon
Pomme	Oeuf	Sucre en Poudre
Chocolat au lait	Crème liquide	Beurre

Appareils  Utensiles 

🕒 60 min Poulet coco réunionnais 🕒 80 min

Mettre les gibbons à votre goût dans le thermomix, ajouter le lait, la crème fraîche et le sucre. Mixer jusqu'à avoir la consistance désirée.

Then Rouge (ou blanc): 200g Concombre: 1 Tomate: 1 Oignon: 1 Poivron rouge: 1 Lait de coco: 100ml

Découper le thon en dés, mettre dans un plat et, recouvrir de jus de citron. Couvrir et laisser au réfrigérateur au moins 2h.

Poulet: 1 Lait de coco: 400ml Coulis de tomate: 25cl Oignon: 1 Huile d'olive

Découper le poulet en morceaux, les faire cuire dans une poêle avec de l'huile d'olive. Saler et poivrer. Une fois cuit, laisser refroidir et ajouter des dés de thon. Au bout de...

🕒 50 min Salade de riz 🕒 45 min Tarte au thon 🕒 50 min Tarte aux pommes 🕒 50 min

Prototype

 Les petits plats

farine

Rechercher un ingrédient

Banane	Beurre	Beurre salé
Chocolat noir en pépites	Chocolat au lait	Chocolat noir
Fromage à raclette	Lait	Noix
Oeuf	Oignon	Pomme
Pruneaux	Saucisse bretonne ou de toulouse	Sucre
Sucre en poudre	Sucre roux	Sucre vanillé

Appareils  Utensils 

🕒 30 min Crème dessert au chocolat 🕒 15 min

Faire fondre le beurre et le mélanger avec le sucre. Finalement ajouter l'oeuf. Ajouter la farine tout en mélangeant peu à peu pour avoir une masse sans grumeaux. Ajouter les pépites de chocolat...

Beurre: 120 g Oeuf: 2 Sucre en Poudre: 100 g farine: 70 g

beurre. Mélanger les oeufs et le sucre et mélanger au chocolat. Ajouter la farine. Mélanger afin d'avoir quelque chose d'homogène puis incorporer le...

Farine: 100 g Chocolat noir en pépites: 100 g Oeuf: 1

🕒 30 min Crème patissière 🕒 60 min Crêpes Chocolat Banane 🕒 40 min Crumble aux pommes

Lait: 50 cl Faire bouillir le lait (on peut y Oeuf: 3 Mélangez dans un saladier, la Pomme: 2 Découper les pommes en...

Affichage du message d'alerte de la barre principale

 Les petits plats

farinn x Q

Aucune recette ne correspond à votre critère... Vous pouvez chercher « tarte aux pommes », « poisson », etc.

Ingrédients Appareils Ustensils



Brownie ⏳ 60 min

Noix: 180 g
Chocolat noir: 150 g
Beurre: 120 g
Oeuf: 2
Sucre en Poudre: 110 g
farine: 90 g

Hachez les noix grossièrement. Faire fondre le chocolat avec le beurre. Mélanger les œufs et le sucre et mélanger au chocolat. Ajouter la farine. Mélanger afin d'avoir quelque chose d'homogène puis incorporer le...



Cookies ⏳ 30 min

Sucre: 100 g
Beurre: 100 g
farine: 100 g
Chocolat noir en pépites: 100 g
Oeuf: 1

Faire fondre le beurre et le miel ensemble avec le sucre. Finalement ajouter l'œuf. Ajouter la farine tout en mélangeant peu à peu pour avoir une masse sans grumeaux. Ajouter les pépites de chocolat...



Crème dessert au chocolat ⏳ 15 min

Lait: 1 litres
Chocolat: 200 g
Sucre: 100 g
Beurre: 50 g
farine: 40 g

Mélanger la farine et le beurre fondre également le lait avec un peu. Ajouter du sucre après la cuisson. Bien mélanger. Ajouter le chocolat en morceaux et laisser chauffer 8 minutes en mélangeant avec une cuillère en...



 Les petits plats

farinn x Q

Aucune recette ne correspond à votre critère... Vous pouvez chercher « tarte aux pommes », « poisson », etc.

Ingrédients Appareils Ustensils



Brownie ⏳ 60 min

Noix: 180 g
Chocolat noir: 150 g
Beurre: 120 g
Oeuf: 2
Sucre en Poudre: 110 g
farine: 90 g

Hachez les noix grossièrement. Faire fondre le chocolat avec le beurre. Mélanger les œufs et le sucre et mélanger au chocolat. Ajouter la farine. Mélanger afin d'avoir...

Affichage de la recette choisie dans une Modal

1440px



Brownie

Noix: 180 g
Chocolat noir: 150 g
Beurre: 120 g
Oeuf: 2
Sucre en Poudre: 110 g
farine: 90 g

Hachez les noix grossièrement. Faire fondre le chocolat avec le beurre. Mélanger les oeuf et le sucre et mélanger au chocolat. Ajouter la farine. Mélanger afin d'avoir quelque chose d'homogène puis incorporer les noix. Verser la préparation dans un moule de préférence rectangulaire. Cuire 20 à 25 minutes à 180°. Sortez du four et attendez quelques minutes pour démouler. Servir avec une boule de glace pour plus de gourmandise.

⌚ 60 min

800px



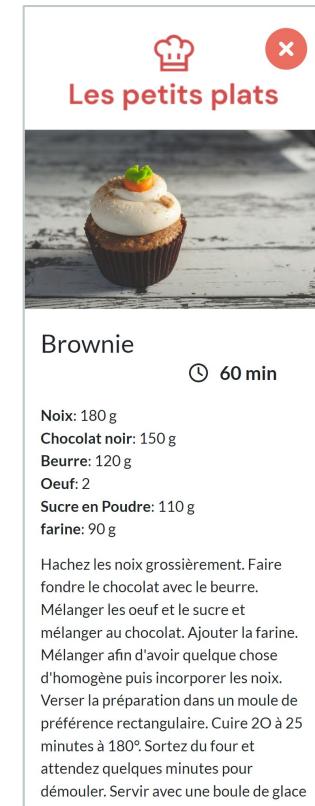
Brownie

Noix: 180 g
Chocolat noir: 150 g
Beurre: 120 g
Oeuf: 2
Sucre en Poudre: 110 g
farine: 90 g

Hachez les noix grossièrement. Faire fondre le chocolat avec le beurre. Mélanger les oeuf et le sucre et mélanger au chocolat. Ajouter la farine. Mélanger afin d'avoir quelque chose d'homogène puis incorporer les noix. Verser la préparation dans un moule de préférence rectangulaire. Cuire 20 à 25 minutes à 180°. Sortez du four et attendez quelques minutes pour démouler. Servir avec une boule de glace pour plus de gourmandise.

⌚ 60 min

380px



Brownie

⌚ 60 min

Noix: 180 g
Chocolat noir: 150 g
Beurre: 120 g
Oeuf: 2
Sucre en Poudre: 110 g
farine: 90 g

Hachez les noix grossièrement. Faire fondre le chocolat avec le beurre. Mélanger les oeuf et le sucre et mélanger au chocolat. Ajouter la farine. Mélanger afin d'avoir quelque chose d'homogène puis incorporer les noix. Verser la préparation dans un moule de préférence rectangulaire. Cuire 20 à 25 minutes à 180°. Sortez du four et attendez quelques minutes pour démouler. Servir avec une boule de glace pour plus de gourmandise.

Notices :

W3C validator résultats identiques pour les pages accueil et menu

- HTML5 : aucune erreur
- CSS3 : aucune erreur

Validator CW3 : index.html

Nu Html Checker

Cet outil est une expérience en cours pour une meilleure vérification HTML, et son comportement reste sujet à changement

Affichage des résultats pour le fichier téléchargé index.html

Entrée de vérificateur

Spectacle la source présenter Options de rapport d'image ...

Vérifier par Aucun fichier choisi

Les fichiers téléchargés avec des extensions .xhtml ou .xht sont analysés à l'aide de l'analyseur XML.

Vérification des documents terminée. Aucune erreur ou avertissement à afficher.

J'ai utilisé l'analyseur HTML.

Temps d'exécution total 10 millisecondes.

Validator CW3 : style.css



Service de validation CSS du W3C

Résultats de la validation W3C CSS de style.css (CSS niveau 3 + SVG)

Aller à: Avertissements (10) CSS valide

Résultats de la validation W3C CSS de style.css (CSS niveau 3 + SVG)

Félicitations ! Aucune erreur trouvée.

Ce document est valide conformément à la recommandation [CSS niveau 3 + SVG](#) !

Pour montrer à vos lecteurs votre souci d'interopérabilité lors de la création de cette page Web, vous pouvez afficher cette icône sur toutes les pages valides. Voici le fragment de XHTML que vous pouvez utiliser pour ajouter cette icône à votre page Web:

```
<p>    <a href="http://jigsaw.w3.org/css-validator/check/referer">
        
    </a>
</p>
```

```
<p>    <a href="http://jigsaw.w3.org/css-validator/check/referer">
        
    </a>
</p>
```

Conclusions :

Pour une recherche rapide et adéquate, il est préférable de préparer en amont le tri des recettes dans l'ordre alphabétique et d'essayer d'anticiper les erreurs d'écriture de l'utilisateur pour pouvoir lui donner la recette correspondante à son envie malgré tout.

Entre méthode filter ou la boucle for, il est surtout essentiel de travailler le corps de la fonction afin de l'optimiser.