



Les petits plats

PRESENTATION DU PROJET n°7

MISE EN SITUATION

Freelance je viens d'être missionné par l'entreprise "Les petits plats" en temps que Développeuse Front-end pour une mission de 3 mois.

Après avoir édité des livres de cuisine pendant plusieurs années, vous avez décidé de vouloir lancer dans un nouveau projet : réaliser son propre site de recettes de cuisine

faire deux implémentations différentes pour la recherche de recettes afin de pouvoir comparer leurs performances et choisir la meilleure.

LISTE DEROULANTE

```
219 selectIngredients = new FilterTags({
220   selectElement: "#selectIngredients",
221   defaultSelectLabel: "Ingrédients",
222   initialListItem: loadIngredients(recipes.data),
223   searchEventCallback: handleSelectIngredientOnSearchEvent,
224   deleteTagEventCallback: (tags) => {
225     searchResult = recipes.searchRecipeExt(searchTerms, tags, selectAppareils.listItem, selectUstensiles.listItem)
226     displaySearchResult({
227       searchTerms: "",
228       result: searchResult
229     });
230
231     resetSelects();
232   },
233   resetEventCallback: handleSelectOnResetEvent
234 });
```

```
33 // Function to load ingredients from recipes, removing duplicates
34 // Fonction pour charger les ingrédients à partir des recettes, en supprimant les doublons
35 const loadIngredients = (fromRecipes) => {
36   const ingredients = [];
37   fromRecipes.forEach((recipe) => {
38     recipe.ingredients.forEach((ingredient) => {
39       ingredients.push({
40         name: ingredient.ingredient,
41         isSelected: false
42       });
43     });
44   });
45   return removeDuplicateObjects(ingredients, "name");
46 };
47
```

```
201
202 // Creates a list item in the options container
203 // Crée un élément de liste dans le conteneur des options
204 createItemElement({ name, isSelected }) {
205   let li = document.createElement("li");
206   li.textContent = name;
207   li.className = isSelected ? "selected" : "";
208
209   // Ajoutez l'image seulement si l'élément est sélectionné
210   if (isSelected) {
211     const image = document.createElement("img");
212     image.className = "img my-custom-class";
213     image.src = "../assets/xmark-circle.svg";
214     image.alt = "xmark-circle";
215
216     // Ajoutez un gestionnaire d'événements pour le clic sur l'image
217     image.addEventListener("click", (event) => {
218       event.stopPropagation(); // Empêche la propagation de l'événement de clic au conteneur li
219       this.removeListItem(li);
220     });
221
222     li.appendChild(image);
223   }
224
225   li.addEventListener("click", () => {
226     this.handleItemClick(li);
227   });
228
229   this.options.appendChild(li);
230 }
231
```

```

1 //DISPLAY RECIPES
2 //AFFICHAGE DES RECETTES
3 import { recipes } from "../data/recipes.js";
4 const recipeContainer = document.querySelector("#recipes");
5 let totalRecipes = document.querySelector("#totalRecipes");
6
7 // Generates HTML for a recipe card
8 // Génère du HTML pour une carte de recette
9 function generateHTMLCard(recipe, ingredientInsertHtml) {
10   return `
11     <article class="recipe">
12       <img src=../assets/images/${recipe.image} alt=${recipe.name} loading="lazy" />
13       <span class="timeCooking">${recipe.time}min </span>
14       <h2 class="recipe-h2">${recipe.name}</h2>
15       <section class="recipe-header">
16         <span class="recipe-small-title">recette</span>
17         <p class="recipe-content">
18           ${recipe.description}
19         </p>
20       </section>
21       <section class="recipe-ingredient">
22         <span class="recipe-small-title">Ingrédients</span>
23         <ul class="recipe-container-ingredient">
24           ${ingredientInsertHtml}
25         </ul>
26       </section>
27     </article>
28   `;
29 }

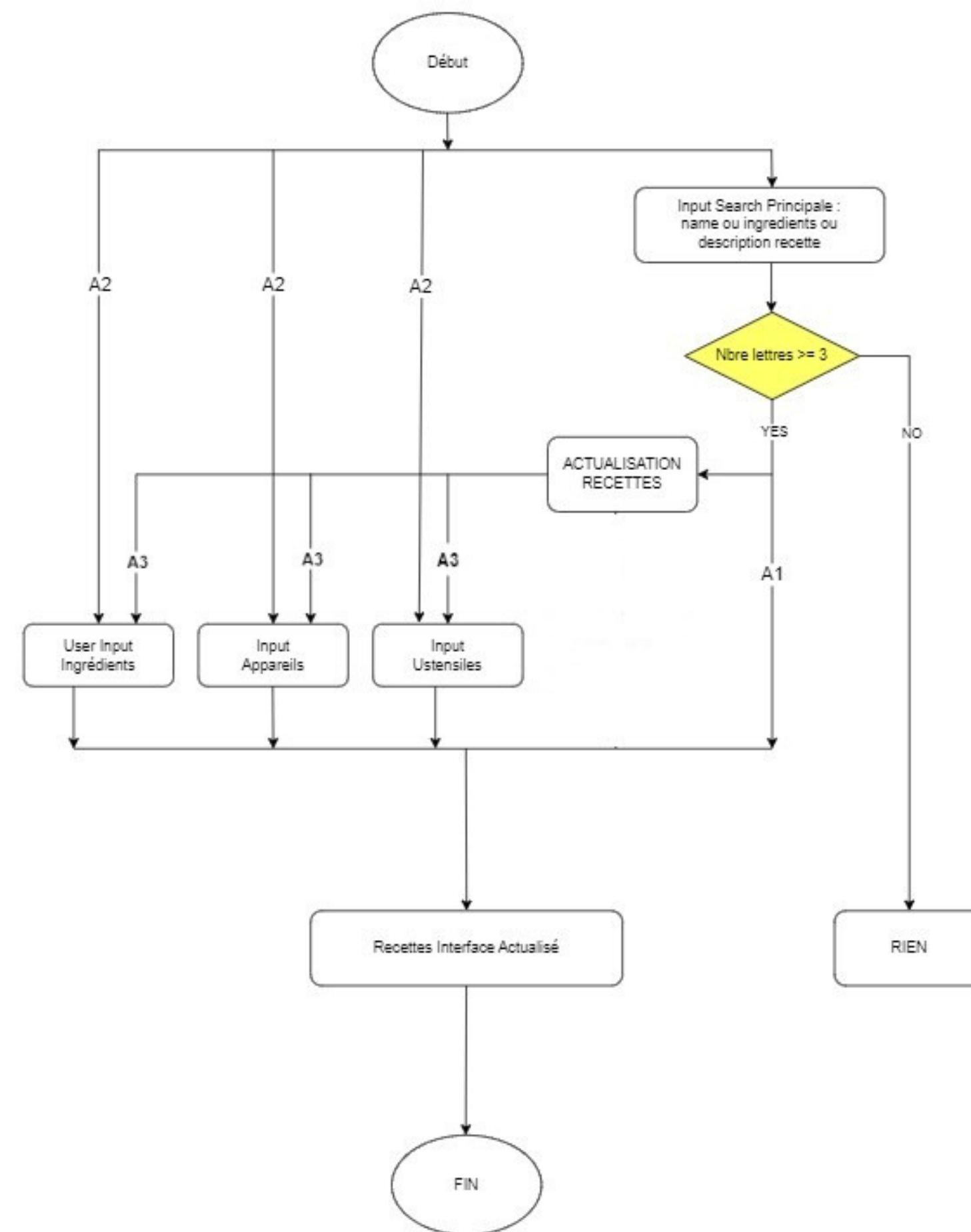
```

Affichage des recettes

```

60 export function displayRecipes(data) {
61   if (!data) {
62     data = recipes;
63   }
64
65   if (totalRecipes) {
66     totalRecipes.innerHTML = `${data.length} recettes`;
67   }
68
69   recipeContainer.innerHTML = "";
70   const recipeCards = data.map((recipe) => {
71     let ingredientInsertHtml = recipe.ingredients
72       .map((ingredient) => generateHTMLIngredient(ingredient))
73       .join("");
74
75     return generateHTMLCard(recipe, ingredientInsertHtml);
76   });
77
78   recipeContainer.innerHTML += recipeCards.join("");
79 }
80

```



Recherche principale

-A1-

Tests

```
22 // BARRE DE RECHERCHE (SEARCH BAR)
23
24 // Checks if there is a match for a given ingredient in a list
25 // Vérifie s'il y a une correspondance pour un ingrédient donné dans une liste
26 IngredientMatch(ingredients, searchTerm) {
27   const matchingIngredient = ingredients.find((anIngredient) => {
28     const { ingredient } = anIngredient;
29     return ingredient.toLowerCase().includes(searchTerm.toLowerCase());
30   });
31   return matchingIngredient != null;
32 }
33
34 // Checks if there is a match for a given appareil (appliance)
35 // Vérifie s'il y a une correspondance pour un appareil donné
36 AppareilMatch(anAppareil, searchTerm) {
37   return anAppareil.toLowerCase().includes(searchTerm);
38 }
39
40 // Checks if there is a match for a given ustensile (ustensil)
41 // Vérifie s'il y a une correspondance pour un ustensile donné
42 UstensilesMatch(ustensiles, searchTerm) {
43   const matchingUstensiles = ustensiles.filter((anUstensile) => {
44     return anUstensile.toLowerCase() === searchTerm.toLowerCase();
45   });
46   return matchingUstensiles.length > 0;
47 }
48
```

```
49 // Searches for recipes based on a search term
50 // Recherche des recettes en fonction d'un terme de recherche (recherche principale)
51 searchRecipes(searchTerms) {
52   searchTerms = searchTerms.toLowerCase();
53
54   return this.data.filter((recipe) => {
55     const { name, description, ingredients } = recipe;
56     const lowerCaseName = name.toLowerCase();
57     const lowerCaseDescription = description.toLowerCase();
58
59     if (lowerCaseName.includes(searchTerms) || lowerCaseDescription.includes(searchTerms)) {
60       return true; // Correspondance trouvée dans le nom ou la description
61     } else {
62       // Aucune correspondance dans le nom ou la description, vérifiez les ingrédients
63       const ingredientMatch = this.IngredientMatch(ingredients, searchTerms);
64       return ingredientMatch;
65     }
66   });
67 }
```

```
185 const initializeEvents = () => {
186   // Récupère l'élément du champ de recherche par son identifia
187   const searchValue = document.getElementById("search-bar");
188   // Ajoute un gestionnaire d'événements pour l'événement "keyup"
189   searchValue.addEventListener("keyup", () => {
190     // Met à jour la variable 'searchTerms' avec la valeur du c
191     searchTerms = searchValue.value;
192     // Utilise la fonction de délai (debounce) pour éviter une
193     const callSearch = debounce(() => {
194       searchRecipes(searchTerms);
195     }, 300);
196     // Vérifie si la longueur des termes de recherche est supér
197     if (searchTerms.length >= 3) {
198       // Si oui, appelle la fonction de recherche avec un déla
199       callSearch();
200     } else {
201       // Si la longueur des termes de recherche est inférieure
202       // Réinitialise l'affichage des recettes, ainsi que les s
203       displayRecipes();
204       selectIngredients.reset();
205       selectAppareils.reset();
206       selectUstensiles.reset();
207       searchResult = [...recipes.data];
208     }
209   });
210 };
211
```

Recherche uniquement par tag

-A2-

```
219 selectIngredients = new FilterTags({           You, il y a 4 semaines • refactor
220   selectElement: "#selectIngredients",
221   defaultSelectLabel: "Ingrédients",
222   initialListItem: loadIngredients(recipes.data),
223   searchEventCallback: handleSelectIngredientOnSearchEvent,
224   deleteTagEventCallback: (tags) => {
225     searchResult = recipes.searchRecipeExt(searchTerms, tags, selectAppareils.listItem, selectUstensiles.listItem)
226     displaySearchResult({
227       searchTerms: "",
228       result: searchResult
229     });
230
231     resetSelects();
232   },
233   resetEventCallback: handleSelectOnResetEvent
234 });
235
```

```
35 const loadIngredients = (fromRecipes) => {           You, le mois d
36   const ingredients = [];
37   fromRecipes.forEach((recipe) => {
38     recipe.ingredients.forEach((ingredient) => {
39       ingredients.push({
40         name: ingredient.ingredient,
41         isSelected: false
42       });
43     });
44   });
45   return removeDuplicateObjects(ingredients, "name");
46 };
47
```

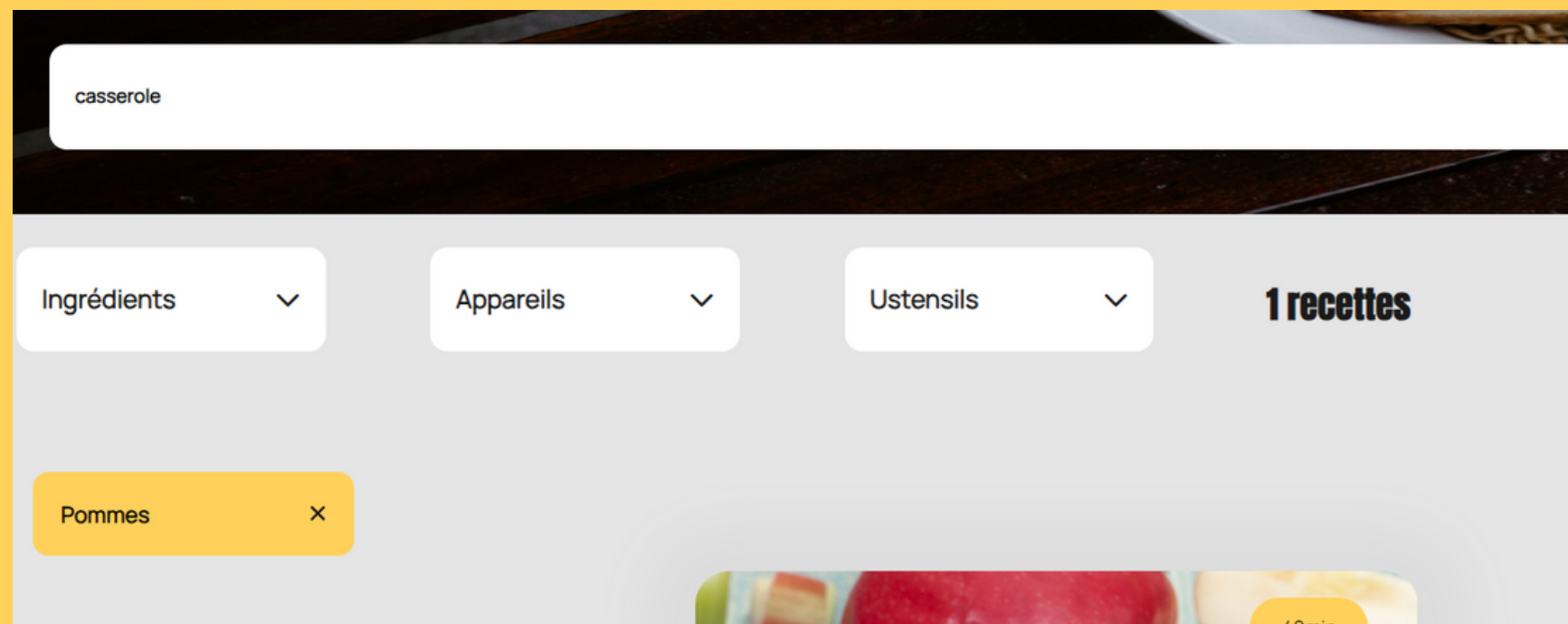
```
106 const handleSelectIngredientOnSearchEvent = (searchTerms) => {
107   const filteredResults = searchResult.filter((element) => {
108     const temp = element.ingredients.filter((ingredient) => {
109       return ingredient.ingredient.toLowerCase() === searchTerms.toLowerCase();
110     });
111     return temp.length > 0;
112   });
113
114   searchResult = [...filteredResults];
115
116   selectAppareils. updateSelectedItems(loadAppareils(filteredResults));
117   selectUstensiles. updateSelectedItems(loadUstensils(filteredResults));
118
119   displaySearchResult({
120     searchTerms: searchTerms,
121     result: filteredResults
122   });
123
124   selectIngredients. updateSelectedItems(loadIngredients(filteredResults));
125 };
126
```

```
90 // Function to search recipes based on input search terms
91 // Fonction pour rechercher des recettes en fonction des termes de recherche
92 const searchRecipes = (searchTerms) => {
93   searchResult = recipes.searchRecipes(searchTerms);
94   displaySearchResult({
95     searchTerms: searchTerms,
96     result: searchResult
97   });
98
99   selectIngredients. updateSelectedItems(loadIngredients(searchResult));
100   selectAppareils. updateSelectedItems(loadAppareils(searchResult));
101   selectUstensiles. updateSelectedItems(loadUstensils(searchResult));
102 }
```

TESTS

Recherche principale plus tags

-A3-



```
216 //PARTIE SELECT
217 // Create Select instances
218 // Créer des instances de sélection
219 selectIngredients = new FilterTags({
220   selectElement: "#selectIngredients",
221   defaultSelectLabel: "Ingrédients",
222   initialListItem: loadIngredients(recipes.data),
223   searchEventCallback: handleSelectIngredientOnSearchEvent,
224   deleteTagEventCallback: (tags) => {
225     searchResult = recipes.searchRecipeExt(searchTerms, tags, selectAppareils.listItem, selectUstensiles.listItem)
226     displaySearchResult({
227       searchTerms: "",
228       result: searchResult
229     });
230     You, le mois dernier * recipes
231     resetSelects();
232   },
233   resetEventCallback: handleSelectOnResetEvent
234 });
235
```

TESTS


```
223 searchEventCallback: handleSelectIngredientOnSearchEvent,
224 deleteTagEventCallback: (tags) => {
225   searchResult = recipes.searchRecipeExt(searchTerms, tags, selectAppareils.listItem, selectUstensiles.listItem)
226   displaySearchResult({
227     searchTerms: "",
228     result: searchResult
229   });
230
```


JSBEN.CH

Tests de performance

JSBEN.CH

BENCHMARKBROWSEDONATE


datas 

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


```
1817         quantity: 500,
1818         unit: "grammes",
1819       },
1820     ],
1821     time: 60,
1822     description:
1823       "Préparer la frangipane : Mélanger le sucre la poudre d'amandes, le beurre et les oeufs. Étaler la moitié de la pate feuilleté et mettre dans
1824     appliance: "Four",
1825     ustensils: ["rouleau à pâtisserie", "fouet"],
1826   },
1827 ];
```


ADD LIBRARY

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

array 

```
35 •   if (filteredRecipes.length === 0) {
36     filteredRecipes = [...this.data];
37   }
38
39 •   const filterRecipesByTag = (tags, property, matchFunction) => {
40     const selectedTags = tags.filter(tag => tag.isSelected);
41     if (selectedTags.length > 0) {
42       const filteredRecipesByTag = selectedTags.flatMap(selectedTag =>
43         filteredRecipes.filter(recipe => matchFunction(recipe[property], selectedTag.name))
44       );
45       filteredRecipes = [...filteredRecipesByTag];
46     }
47   }
```



boucles-natives 


```
111     };
112   }
113 }
114 •   if (filteredRecipesByUstensils.length > 0) {
115     filteredRecipes = [...filteredRecipesByUstensils];
116   }
117
118   return filteredRecipes;
```

RUN TESTS

GENERATE PAGE URL

NEW BENCHMARK

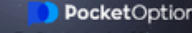
result

array (119039) 


100%


boucles-natives (112185)


94.24%



Profitez toutes les 60 secondes
CRYPTO
COMMENCER


If you like to donate (Thank you!):


 Ethereum (ETH)


 Chia (XCH)


 Cardano (ADA)


 Ravencoin (RVN)


 Bitcoin (BTC)

 Ripple (XRP)

 Litecoin (LTC)

 Monero (XMR)

 Dogecoin (DOGE)

 SOLANA (SOL)

W3C

Nu Html Checker

Google Translate

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for index.html

Checker Input

Show ☐ source ☐ outline ☐ image report

Check by Aucun fichier choisi

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

10 messages hidden by filtering

Document checking completed. No errors or warnings to show.

Used the HTML parser.

Total execution time 12 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 23.11.28