

Projetez des données avec la fonction `map`

5–7 minutes

En manipulant les listes, vous avez rendu votre page interactive. Cette dernière contient désormais plusieurs fiches produits. Dans ce chapitre, nous mettrons de côté ces éléments pour nous intéresser à une autre fonction. Imaginez que votre page contienne beaucoup (trop) d'informations. Comment pouvez-vous faciliter la lecture de cette page ? En affichant un résumé des produits grâce à la fonction `map`.

Mappez une liste

Une première étape consiste à extraire le nom de toutes nos pièces à partir du fichier JSON. Problème : les noms des pièces ne sont pas facilement accessibles. Ils se situent dans les objets des pièces, qui eux-mêmes se situent dans la liste de toutes les pièces.

Oups ! Comment on fait alors ?

La fonction **`map`** va nous aider à **récupérer** le nom des pièces automobiles. Grâce à elle, nous allons transformer notre liste de pièces d'origine en une liste contenant uniquement le nom des pièces. Magique ! 🤩

```
[
  {
    "nom": "Ampoule LED",
    "prix": 60,
    "disponibilite": true
  },
  {
    "nom": "Plaquettes de frein (x4)",
    "prix": 40,
    "disponibilite": true
  },
  {
    "nom": "Ampoule boîte à gants",
    "prix": 5.49,
    "disponibilite": false
  }
]
```



```
[
  "Ampoule LED",
  "Plaquettes de frein (x4)",
  "Ampoule boîte à gants",
]
```

```

},
{
  "nom": "Liquide de frein",
  "prix": 9.60,
  "disponibilite": true
},
{
  "nom": "Balai d'essuie-glace",
  "prix": 29.1,
  "disponibilite": true
}
]

```

map

```

"Liquide de frein",

"Balai d'essuie-glace"
]

```

La fonction map permet de récupérer le nom des pièces automobiles.

Voici le code qui nous permet de récupérer le nom de toutes les pièces :

```
const noms = pieces.map(piece => piece.nom);
```

Cela nous donne la liste suivante :

```

[
  "Ampoule LED",
  "Plaquettes de frein (x4)",
  "Ampoule boîte à gants",
  "Liquide de frein",
  "Balai d'essuie-glace"
]

```

On constate dans ce résultat que les objets du JSON ont disparu, et qu'il ne reste que le nom des pièces automobiles.

Peut-être avez-vous également remarqué la syntaxe "piece => piece.nom" ? Laissez-moi vous dire quelques mots à son sujet. Il s'agit d'une fonction JavaScript à l'écriture grandement simplifiée. On l'appelle **fonction lambda**. Elle correspond à l'écriture suivante :

```

// Fonction lambda
piece => piece.nom

// Fonction normale

```

```
function (piece) {  
    return piece.nom;  
}
```

Au final, `map` se sert de cette fonction lambda pour générer une nouvelle liste à partir de la liste d'origine. La fonction lambda est appelée pour chaque élément de la liste, et sa valeur de retour est la chaîne de caractères du nom de la pièce. Cette chaîne de caractères est utilisée pour remplir la nouvelle liste que nous stockons dans la constante "noms". La nouvelle liste ne contient donc que les noms des pièces auto.

```
const prix_doubles = pieces.map(piece => piece.prix *  
2);
```

Ajoutez et supprimez les éléments d'une liste

Nous avons réussi à générer une liste avec tous les noms des pièces. Mais certaines ne sont pas abordables. Nous voulons donc supprimer le nom de toutes les pièces non abordables. Pour cela, nous procéderons en deux étapes :

1. Parcourir la liste des pièces avec une boucle `for`.
2. Supprimer les noms avec la fonction **`splice`**.

Lorsque l'on souhaite chercher des éléments dans une liste pour les supprimer, on doit parcourir la liste **de la fin vers le début**. Sans cela, certains éléments ne seraient pas testés à cause du décalage que provoquerait la suppression des éléments indésirables.

Pour supprimer les éléments indésirables, nous allons utiliser la fonction **`splice`**. Elle prend en argument deux valeurs :

- l'indice à partir duquel supprimer des éléments ;
- la quantité d'éléments à supprimer.

Ici, nous nous servons de l'indice `i` comme premier argument, et

nous supprimerons toujours un seul élément à la fois.

```
const noms = pieces.map(piece => piece.nom);
for(let i = pieces.length -1 ; i >= 0; i--){
    if(pieces[i].prix > 35){
        noms.splice(i,1)
    }
}
console.log(noms)
```

Une fois ce code exécuté, la liste noms sera vidée de tous les noms des pièces qui n'étaient pas abordables.

Comme vous avez pu le constater, la syntaxe filter() simplifie grandement l'écriture du code. 😊

Affichez la liste des pièces abordables

Grâce à cette liste de noms, nous pouvons générer les éléments du DOM pour afficher une liste à puces sur notre site. Pour cela, nous reprenons les fonctions createElement et appendChild que nous avons rencontrées dans les chapitres précédents.

```
//Création de la liste

const abordablesElements =
document.createElement('ul');

//Ajout de chaque nom à la liste
for(let i=0; i < noms.length ; i++){

    const nomElement = document.createElement('li');
    nomElement.innerText = noms[i];
    abordablesElements.appendChild(nomElement)
}


// Ajout de l'en-tête puis de la liste au bloc
```

résultats filtres

```
document.querySelector('.abordables')
```

```
.appendChild(abordablesElements)
```

Et voici le résultat :



Les Bonnes Pièces

Filtres

Trier par prix croissant


Filtrer les pièces non abordables

Trier par prix décroissant

Filtrer les pièces sans description


Pièces abordables :

- Ampoule boîte à gants
- Liquide de frein
- Balai d'essuie-glace




Ampoule LED

Prix : 60 € (€€€)
Optiques
Distance d'éclairage : 100 mètres !
En stock




Plaquettes de frein (x4)

Prix : 40 € (€€€)
Freinage
Une qualité de freinage optimale, par tous les temps
En stock




Ampoule boîte à gants

Prix : 5.49 € (€)
Optiques
Pour y voir clair dans l'habitacle
Rupture de stock



Liquide de frein

Prix : 9.6 € (€)
Freinage
Pas de description pour le moment.
En stock



Balai d'essuie-glace

Prix : 29.1 € (€)
(aucune catégorie)
Performances d'essuyage au Top !
Longueur : 550 mm
En stock

Le site des Bonnes Pièces contient désormais un résumé de toutes les pièces abordables

La liste des pièces disponibles affichée à l'écran ne contient que trois éléments, alors qu'on est partis de la liste pieces qui en contenait cinq au départ.

Récapitulons en vidéo

Vous pouvez revoir les différentes étapes de ce chapitre dans la vidéo ci-dessous :

À vous de jouer !

Depuis la branche [P2C2-Exercice](#), affichez une description des pièces disponibles à côté de la description des pièces abordables.

L'intitulé de la pièce devra aussi contenir son prix. Par exemple :

Pièces disponibles :

- Ampoule LED – 60 €.
- Plaquette de frein (x4) – 40 €.
- Liquide de frein – 9,6 €.

Corrigé

Vous pouvez vérifier votre travail en consultant [la branche P2C2-Solution](#) et en regardant la vidéo ci-dessous :

En résumé

- Utilisez la syntaxe lambda pour écrire des fonctions JavaScript simplifiées et fluidifier la lecture du code.
- Mappez une liste avec la fonction map pour transformer une information, par exemple :
- l'objet d'une pièce automobile vers le nom de la pièce.
- Supprimez des éléments dans une liste avec la fonction splice.

Les produits abordables sont affichés en haut de la page. Bien joué ! Il ne nous reste plus qu'à mettre à jour l'affichage de la page web lorsque l'utilisateur souhaite filtrer les fiches produits. Allons-y !

