

- [Instructions](#)

# Instructions

1 .

Déclarez et initialisez une variable d'état appelée `cart` qui gardera une trace d'une liste de valeurs de chaîne.

Chacune de ces valeurs de chaîne représente un article d'épicerie que nous avons ajouté à notre panier. Nous ajouterons des écouteurs d'événements et des gestionnaires d'événements pour ajouter et supprimer des articles à notre panier dans les étapes à venir.

Pour l'instant, commençons par initialiser pour `cart` avec la valeur d'un tableau vide pour le premier rendu.

2 .

Jetez un œil au code actuel, nous avons deux fonctions : `addItem()` et `removeItem()`. Ajoutons quelques paramètres à ces fonctions de gestion d'événements, `addItem()` devraient accepter un argument appelé `item` et `removeItem()` devraient accepter un élément appelé `targetIndex`.

Ne vous souciez pas encore d'écrire le corps de la fonction pour ces gestionnaires d'événements, nous le ferons dans les prochaines étapes !

3 .

Il est temps de nous concentrer sur le corps des fonctions du gestionnaire d'événements. Nous commencerons par `addItem()`.

Mettez à jour l'état `cart` avec la fonction de définition d'état à chaque appel `addItem()`. N'oubliez pas d'utiliser une fonction de rappel pour les meilleures pratiques ! À l'intérieur, utilisez la [syntaxe de répartition du tableau](#) pour ajouter un nouvel élément au tableau d'état.

4.

Enfin, `removeItem()` doit être retiré de l'état lorsqu'il est appelé.

Utilisez la méthode array `.filter()` pour supprimer de notre état l'élément situé à l'index de l'élément sur lequel vous avez cliqué dans notre liste.

Pourquoi voudrions-nous utiliser l'index de l'élément cliqué au lieu de l'élément lui-même pour déterminer ce qu'il faut supprimer de notre modèle de données ? Disons que nous avons deux éléments identiques dans un tableau. Utiliser la valeur pour supprimer l'élément supprimerait tous les éléments avec cette valeur, nous utilisons donc l'index comme identifiant unique.