

# Instructions

---

## Les étapes

1. Développer la fonction `displayProducts` qui permet d'afficher les produits de la boutique. L'appeler au chargement de la page.
2. Ajouter les écouteurs d'événement sur les éléments de tri et coder les tri. Tester si les tris fonctionne.
3. Coder l'ajout à la liste d'envie.
4. Coder l'ajout au panier.

## BONUS

- Pouvoir modifier le contenu du panier : modification de la quantité, suppression d'un article
- Pourvoir filtrer les produits par catégorie

## La fonction `displayProducts()`

Cette fonction affiche le contenu de la boutique. Elle est essentiellement constituée d'une boucle sur la table des articles.

- Création des différents éléments à l'aide des instructions `createElement`, `appendChild`, `innerHTML`, `innerText`, `setAttribute`.
- Utilisation de `classList` pour ajouter les classes.
- Ajout d'un écouteur d'événement sur le lien "Ajouter à ma liste d'envie" et sur le bouton "Ajouter au panier".

## La fonction `ascendingSorting()`

Il faut trier les articles par prix croissant, en utilisant la fonction `sort()` + une fonction de comparaison.

## La fonction `descendingSorting()`

Il faut trier les articles par prix décroissant, en utilisant la fonction `sort()` + une fonction de comparaison.

## La fonction `addToWishList()`

- Il faut récupérer le produit qui a été cliqué. Pensez à la notion "this".
- Si le produit n'est pas déjà dans le tableau, on le pousse dedans.
- La liste des envies est stockée dans la `localStorage`. Utilisation de `JSON.stringify()` pour pouvoir le stocker.
- Rafraîchir la liste en appelant la fonction `afficherListeEnvies()`

## La fonction `displayWishList()`

- Récupération des éléments dans la `localStorage`. Utilisation de `JSON.parse()`.
- Générer la liste sous forme de `ol/li`.
- Dans une balise paragraphe : afficher le nombre de produits ou l'indication "La liste est vide".
- Afficher le bouton si la liste n'est pas vide, sinon le cacher.

## La fonction `emptyWishList()`

- Vider le tableau
- Effacer les données du `localStorage`

## La fonction `addToCart()`

- Le traitement est similaire à celui de la liste d'envies, mais les données sont stockées dans la `sessionStorage`.
- Récupérer l'id, le nom et le prix qui se trouvent dans les attributs `data`.
- On ajoute dans le panier uniquement si le produit n'y est pas déjà, sinon on met à jour la quantité.
- Il faut pousser un tableau d'objet.
- Calculer le nouveau total du panier en utilisant la fonction `cartTotals()`.
- Rafraîchir le contenu panier en appelant la fonction `displayCart()`.

## La fonction `displayCart()`

- Récupérer les données qui sont stockées dans la `sessionStorage`.
- Si le panier est vide, afficher le texte "Le panier est vide"

- Sinon
1. Afficher les données dans le header
  2. BONUS : afficher le détail du panier

## BONUS : la fonction updateCart()

Cette fonction permet de mettre à jour le contenu du panier.

- La quantité peut être mise à jour en plus ou en moins (si la quantité tombe à 0, le produit est supprimé du panier).
- Un produit peut être supprimé.

## La fonction emptyCart()

Elle consiste à vider le panier.

## Liens

Tri : [https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Global\\_Objects/Array/sort](https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Global_Objects/Array/sort)

Local Storage : <https://developer.mozilla.org/fr/docs/Web/API/Window/localStorage>

Session Storage : <https://developer.mozilla.org/fr/docs/Web/API/Window/sessionStorage>