

# TP Boutique

**Documentation :**

<http://champin.net/enseignement/intro-js>

<https://devdocs.io/>

**Outil de vérification de code :**

<https://jshint.com/>

**Objectif :**

L'objectif est de réaliser une application type "boutique en ligne". Vous n'aurez pas à tout programmer dans cet exercice car vous partirez d'un code existant. Vous serez guidé(e)s pas à pas dans les différentes étapes qui devraient vous amener à un site fonctionnel, après l'introduction qui explique le fonctionnement général attendu.

**Remerciements :**

Cet exercice est une adaptation d'un projet de Jean-Christophe Routier (Professeur à l'université de Lille).

**Introduction :**

Comme l'illustre la figure ci-dessous, la boutique en ligne présente deux zones principales. Celle de gauche (#boutique) contient les différents produits mis en vente par la boutique, celle de droite (#panier) correspond au panier construit par l'utilisateur.

## Boutique Top Chaises

rechercher des produits...

<b>chaise bleue</b>  une chaise bleue avec 4 pieds 0	<b>chaise blanche</b>  une chaise blanche matelassée avec 4 pieds noirs 0	<b>chaise en bois</b>  une chaise en bois avec 4 pieds et un dossier rouge 0	<b>chaise en cuir</b>  une chaise en cuir vieilli avec un dossier confortable 0	<b>Total du panier : 406 €</b> <div> chaise en cuir 2x 125 €</div> <div> chaise rouge 3x 52 €</div>
<b>chaise design</b>  une chaise design mais peut-être pas confortable 0	<b>chaise noire</b>  une chaise noire avec un pied métallique 1	<b>chaise plastique</b>  une chaise basique avec une coque plastique 0	<b>chaise rouge</b>  une chaise tout en plastique transparent rouge avec 4 pieds rouges aussi 0	

Le contenu de la boutique est construit automatiquement au chargement de la page par rapport à des données stockées dans une variable globale catalogue qui contient les produits de la boutique dans des objets javascript.

Dans la boutique, l'utilisateur peut, pour un article, choisir la quantité qu'il souhaite acheter à l'aide du champ de saisie correspondant. Cette quantité doit toujours être

comprise entre 0 et 9. Cette contrainte de validité est garantie lorsque l'on utilise les "flèches" mais devra être gérée si l'utilisateur saisit directement une valeur. Il doit ensuite cliquer sur le bouton représentant un chariot de course pour mettre dans le panier cet article avec la quantité désirée. Le bouton de mise en panier devra être inactif tant que la quantité est 0, actif sinon. Après la mise en panier le compteur de quantité sera remis à 0.

Lorsqu'il est mis en panier, l'article apparaît dans la zone du panier. Un même article n'apparaîtra toujours qu'une seule fois dans le panier, avec sa quantité totale commandée. Donc lorsqu'un article déjà dans le panier sera à nouveau commandé, sa quantité sera mise à jour dans le panier. Il n'est jamais autorisé de commander plus de 9 fois un même article, même en plusieurs fois. Donc, la quantité d'un même article dans le panier ne devra jamais dépasser 9. En cliquant sur le bouton représentant une poubelle, on supprimera complètement un article du panier (quelle que soit la quantité). Le montant total des articles dans le panier apparaîtra dans la zone #total. Ce montant sera mis à jour à chaque modification du panier (ajout ou suppression d'articles).

Au-dessus du total, se trouve une zone de texte qui permettra de filtrer les produits de la boutique. Seuls les produits dont la description contiendra la chaîne de caractères présente dans le filtre seront affichés. L'affichage s'adaptera à chaque caractère tapé par l'utilisateur.

## **Etape 0**

Récupérez le zip du TD5 qui contient le projet à compléter pour obtenir les fonctionnalités décrites dans l'introduction et détaillées ci-dessous. Repérez les fichiers. Lisez le code déjà présent dans index.html, data/catalogue.js et scripts/script.js.

## **Etape 1**

Modifiez script.js pour qu'il permette l'affichage des photos de produit dans la boutique.

## **Etape 2**

Un événement input est émis à chaque fois que la valeur d'un <input> change en lien avec une action de l'utilisateur. Modifiez script.js pour mettre en place la gestion de la zone de saisie des quantités, avec le contrôle des valeurs : si l'utilisateur saisit "à la main" une valeur interdite, il faudra annuler cette saisie, en la remplaçant par 0. Il vous faut aussi gérer l'activation/désactivation du bouton de mise en panier selon que la quantité vaut 0 ou non. D'autres événements peuvent être écoutés sur un <input> : blur, change...

## **Etape 3**

Gérez l'action déclenchée par le bouton de mise en panier (lorsque celui est actif).

### **Etape 3.1**

Commencez par modifier le contenu du tableau quantities pour ajouter à la quantité du produit concerné la nouvelle quantité commandée. Ne pas oublier que la quantité totale commandée pour un produit ne peut pas excéder 9. Utilisez console.log pour vérifier le contenu du tableau.

## Etape 3.2

Créez une fonction `creerDivAchat` (en vous inspirant de la fonction `creerDivProduit`) qui prendra l'index d'un produit en paramètre. Elle construit et retourne une div de code similaire au suivant (dans l'exemple, c'est le produit d'index 3 du catalogue) :

```
<div class="achat" id="3-achat">
  <figure>
    
  </figure>
  <h2>chaise en cuir</h2>
  <div class="quantite">2</div>
  <div class="prix">125</div>
  <div class="controle">
    <button class="supprimer" id="3-supprimer"></button>
  </div>
</div>
```

## Etape 3.3

Lorsqu'on clique sur un bouton pour commander un produit, il faut regarder si le produit est déjà dans le panier. Si c'est le cas, on gère juste la modification de la quantité dans sa div de classe `quantite`. Sinon, on utilisera `creerDivAchat` pour créer une div.achat le concernant qu'on ajoutera au panier.

## Etape 3.4

Lorsque le produit est ajouté au panier, il faut remettre sa quantité à commander à 0 dans la boutique et désactiver son bouton de commande.

## Etape 3.5

Vous devez également garantir que la valeur de `#montant` soit mise à jour.

## Etape 4

Gérez l'action des boutons de suppression des produits du panier. Vous devez toujours maintenir la cohérence de `#montant`.

## Etape 5

Faites le nécessaire pour que seuls les produits dont la description contient le texte présent dans `#filter` soient affichés dans la boutique. Le filtrage doit être modifié au fur et à mesure de la saisie des caractères dans `#filter`.

## **Etape 5.1**

Modifiez `afficherBoutique` pour qu'elle commence par vider la div de la boutique avant d'ajouter les produits uniquement si la valeur contenue dans le tableau `tabProduitsFiltres` à l'index correspondant au produit est à `true`.

## **Etape 5.2**

Abonnez une fonction à l'événement `input` sur l'input du filtre qui met à jour le contenu du tableau `tabProduitsFiltres` en fonction de la valeur du filtre. Vous pouvez pour cela utiliser la fonction `indexOf` par exemple.

## **Question**

Une mauvaise pratique s'est glissée dans ce TP, pour le simplifier compte tenu du temps de développement dont vous disposez et compte tenu du fait que nous ne mettrons pas ce projet en production. Quelle est-elle ? Comment pourriez-vous remplacer le code concerné ? Pourquoi est-ce important ?