

# 582-31F-MA - Programmation d'interface Web 2

---

## TP1

---

### Description

Produire une application dynamique à l'aide de JavaScript utilisant la programmation orientée objet modulaire.

### Pondération

25%

### Modalité particulière

Travail individuel

### Sujet

Librairie

### Date de remise

Dimanche 10 septembre 23h59

### Modalités de remise

Le dossier zippé, à votre nom (nomdefamille-prenom) doit être remis sur Léa avant le 10 septembre 23h59.

Le site doit être en ligne avant le 10 septembre 23h59 sur WebDev. Dans ce dossier, vous devez inclure un fichier .txt avec l'URL de votre site.

### Retard

Selon les règles du collège, 5% par jour de retard seront enlevés, jusqu'à 5 jours de retard maximum.

## Énoncé

Vous avez obtenu le contrat pour la refonte du site d'une librairie en ligne. Pour le premier sprint, votre équipe vous a attribué le développement de la page listant les livres en inventaire. Celle-ci affiche les livres sous différentes catégories ainsi que le détail d'un livre suite au clic de sa tuile. Aussi, vous devez développer, côté client, la fonctionnalité d'ajouter au panier.

Comme votre travail doit se greffer à celui de vos collègues, il vous faudra respecter le devis ainsi que les exigences techniques.

Vous devrez mettre une version de test en ligne de votre application sur Webdev.

## Consignes

## Sources de données

L'application backend n'est pas complète, du coup vous devrez travailler avec des données statiques en format JSON (fourni).

Toutes les images sont également données. Je vous invite à les placer suivant l'arborescence `./assets/img/`, autrement il vous faudra changer tous les chemins relatifs du fichier `livres.js`.

## Filtres

Au chargement de la page, seuls les 12 premiers livres sont injectés.

Il y a 8 filtres, ceux-ci sont :

- Tous
- Nouveautés
- Littérature
- Art de vivre
- BD, Jeunesse, Humour
- Culture et société
- Loisirs, Tourisme, Nature
- Savoir et science

Vous aurez compris qu'il faudra afficher les livres correspondants au filtre cliqué.

Remarquez que Nouveauté n'est pas une catégorie mais bien une clé présente pour chaque livre. Bref, au clic de Nouveauté, vous devez injecter tout livre où sa clé `nouveaute` est true.

Tâchez d'éviter la redondance, faire quatre boucles distinctes mais similaires pour les quatre différents cas (chargement de page (12 premiers livres), Tous, Nouveauté et les différentes catégories) serait peu optimal.

## Affichage des tuiles et du modal

Assurez-vous de faire la gestion UX / UI de la catégorie active.

Chaque tuile d'un livre doit afficher son image, son titre, son prix et un bouton pour l'ajouter au panier.

Au clic d'une tuile (toute la tuile doit être cliquable), un modal affiche les informations de ce livre, soit : son image, son titre, son auteur, son éditeur, son nombre de pages et sa description.

Une bonne stratégie serait d'injecter l'index du livre courant en dataset lors de l'injection de sa tuile.

Prévenez le scroll en Y lorsque le modal est ouvert.

Une fois le modal ouvert, l'utilisateur peut le fermer en cliquant n'importe où sur celui-ci. Prenez tout de même la peine de placer un en haut à droite.

Évidemment, si le scroll en Y a été prévenu en état ouvert, il faudra redéfinir ce comportement à la fermeture du modal.

## Gestion du panier d'achat et persistance des données

Au clic d'un bouton Ajouter, vous devez ajouter ce livre au `sessionStorage` assigné au panier. Les informations à conserver à l'intérieur du `sessionStorage` sont le titre et le prix du livre. Pour ce faire, vous devrez placer ces informations dans un objet littéral qu'il faudra push à l'intérieur d'un tableau. C'est ce tableau qui sera gardé en mémoire.

Vous avez peu de gestion à faire sur ce panier, vous devez toutefois vous assurer de ne pas ajouter un livre déjà présent dans le panier.

Au clic de l'icône panier (SVG) à la droite du `<header>`, affichez dans un petit modal le détail du panier ainsi que le prix total avant taxe à l'intérieur d'un élément `<table>`.

S'il n'y a qu'un livre dans le panier, l'intitulé du premier `<th>` sera 'Livre'. S'il y a plus qu'un livre, ce même intitulé sera 'Livres'. Si toutefois le panier est vide, injectez uniquement le paragraphe 'Il n'y aucun livre dans votre panier.'

Au clic suivant sur l'icône panier, si le modal est affiché, fermez-le.

## Stratégies de développement

Ce TP met à l'épreuve votre capacité à utiliser la programmation orientée objet.

Aucun script procédural n'est accepté : le fichier main.js ne sert qu'à lancer les instances de classes nécessaires au chargement de la page et chaque bloc fonctionnel devrait avoir sa classe.

Aussi, dès que possible, encapsulez vos données et utilisez le préfixe # pour déclarer comme private toutes propriétés et méthodes qui n'ont pas à être publiques.

1. Commencez par coder le HTML et le CSS de la page listant les livres en inventaire. Vous pouvez vous inspirer de la maquette fournie.
2. Codez le HTML et le CSS du modal. Mettez le modal en `display: none;` par défaut.
3. Codez le HTML et le CSS du modal du panier d'achat. Mettez le modal en `display: none;` par défaut.
4. Créez les classes nécessaires à votre application.
5. Codez le javascript de la page listant les livres en inventaire. Vous pouvez vous inspirer du fichier livres.js pour injecter les livres.
6. Codez le filtre des livres.
7. Codez l'affichage et la fermeture du modal au clic d'une tuile.
8. Codez l'ajout d'un livre au panier d'achat.
9. Codez l'affichage du panier d'achat.
10. Testez votre application.
11. Mettez en ligne sur Webdev.

## Exigences techniques et critères d'évaluation

- Réussite des différentes fonctionnalités et respect des consignes
- Utilisation efficace de la programmation orientée objet (classes, modules, héritage, encapsulation, polymorphisme, etc.)
- Structure et optimisation du code (indentation, nommage, etc.)
- Qualité des algorithmes et qualité du code source.
- Gestion des erreurs. Le code ne contient aucun avertissement (warning) et aucune erreur (error) au moment de l'exécution

- Commentaires fréquents et pertinents
- Ergonomie générale : ce n'est pas un travail de CSS, mais veillez à intégrer une interface engageante
- Le site est en ligne sur Webdev

## Plagiat

Nous travaillons dans le Web, alors les réponses s'y trouvent et il va de soi que vous allez devoir chercher sur Google. Dans l'esprit open-source du Web, vous avez tout à fait le droit de copier-coller un bout de code que vous avez trouvé. Cela-dit, vous devez citer les extraits de code qui dépassent une ligne ou deux et suivre un tutoriel (en tout ou en partie) sera considéré comme du plagiat.

Il s'agit d'un travail strictement individuel. Vous pouvez poser des questions à vos collègues, mais ce doit être votre logique et vos scripts. L'objectif est de développer votre autonomie. En cas de plagiat, je devrai appliquer les sanctions de la PIEA (Politique institutionnelle d'évaluation des apprentissages).

Si vous êtes vraiment bloqué, contactez-moi.