

Exercice 1 - JavaScript Module Pattern (sommatif)

Devis (5)

Vous avez - en dur - deux listes de trois produits en vedette, soit des livres jeunesse et des jeux. Au clic d'un produit, affichez ou cachez son détail, ceux-ci listés dans le fichier `produits.js`. Plus précisément :

- Toute la structure *HTML* et les styles sont déjà donnés. D'ailleurs, remarquez la valeur de l'attribut `data-js-produit` qui identifie la clé et l'index de ce produit à l'intérieur de la structure de data `produits`.
- **(1)** Vous devez travailler en orienté objet avec la syntaxe *JavaScript Module Pattern* et le *namespacing*. Notez également que les fonctions fléchées (*arrow function*) ne sont pas autorisées.
- **(.5)** Dans `main.js`, lancez les comportements pour chaque nœud *DOM*. Après avoir récupéré sur tous les éléments `data-js-produit`, bouclez sur chacun afin de vérifier sa catégorie (`livres` ou `jeux`) puis instancier l'objet correspondant en passant en argument l'élément courant.
- **(1)** Définissez le constructeur des objets `Livre` et `Jeu` puis développez leur méthode de création *DOM* respective.
- **(.5)** Le *DOM* à injecter est simple, définissez un élément `<p>` pour chaque propriété à injecter dans l'élément `data-js-produit-detail`.
Exemple : `<p><small>Titre : </small>${this.titre}</p>`
- **(1)** Les objets `Livre` et `Jeu` partagent plusieurs définitions et comportements. Les objets `Livre` et `Jeu` héritent donc de l'objet `Produit` où sont situées les fonctionnalités génériques, celles-ci sont :
 - la définition des propriétés communes à tous les produits ;
 - la fonctionnalité de calculer un potentiel rabais ;
 - le gestionnaire d'événement clic : si le détail est fermé, injectez le *DOM* du détail dans l'élément `data-js-produit-detail` puis affichez-le ; si le détail est ouvert, cachez l'élément précédemment affiché.

Exercice 1 - JavaScript Module Pattern (suite)

- Les classes CSS pour la gestion de l'affichage du détail sont déjà définies (`produit__detail--ferme` et `produit__detail--ouvert`). Sans surprise, l'état initial est fermé.
- **(.5)** Dans l'objet **Produit**, afin d'éviter la redondance, placez à l'intérieur d'une méthode publique la partie commune aux chaînes *DOM* de **Livre** et **Jeu** (propriétés **titre**, **sujet** et **prix**). Celle-ci sera appelée pour construire la partie initiale partagée par les deux détails à injecter.
- **(.5)** Si la clé **rabais** n'est pas **false**, ce produit a un rabais de la valeur indiquée en pourcentage (par exemple **rabais: 15** égale 15% de rabais). Vous devez faire cette gestion dans une méthode privée de **Produit**. Remarquez que, lorsqu'il y a un rabais, le pourcentage de la réduction est également affiché (voir références). Pour vous aider, voici la formule mathématique pour le calcul du rabais :
$$\text{prix initial} - (\text{prix initial} * \text{taux pourcentage} / 100)$$
- Comme toujours, je vous invite à travailler étape par étape. L'étape initiale sera de faire la gestion d'un **Livre** ou d'un **Jeu**, mais pas les deux en même temps. Une fois l'un terminé, il suffira de le dupliquer et de l'adapter pour l'autre. Finalement, factorisez le code en plaçant toutes propriétés et méthodes communes dans l'objet parent **Produit**.