

Plan de test

KANAP

Nom : GRITSIYENKO Yuliya
12/04/2022

Les tests unitaires

Obtenir les données de l'API : tous les produits

description : fonction pour obtenir les données localisées sur l'API, type GET

fonction à tester : `getAllProducts datamanager.js`

déclencheur :

arguments :

affichage attendu :

Extraire l'id du produit de la barre d'adresse

description : fonction pour obtenir l'id du produit

fonction à tester : `getProductId datamanager.js`

déclencheur : chargement de la page du produit

arguments :

affichage attendu :

Obtenir les propriétés du produit par son id

description : fonction pour obtenir les données localisées sur l'API par l'id du produit, fetch type Get

fonction à tester : `getProduct datamanager.js`

déclencheur :

arguments : id

affichage attendu : données pour l'affichage du produit sur la page product.html ou l'affichage des erreurs

Ajouter des produits dans le panier en utilisant localStorage

description : fonction pour ajouter un ou des produits dans localStorage

fonction à tester : addToCart datamanager.js

déclencheur :

arguments : id, couleur, quantité

affichage attendu : utilisée dans l'autre fonction onClickAdd de product.js

Sauvegarder localement les produits

description : fonction pour supprimer des produits dans localStorage

fonction à tester : saveLocal datamanager.js

déclencheur : utilisée dans les autres fonctions, liées aux boutons : addToCart, changeQty, deleteFromCart

arguments : cart, storage

affichage attendu :

Supprimer des produits dans la page panier en utilisant localStorage

description : fonction pour supprimer des produits dans localStorage

fonction à tester : deleteFromCart datamanager.js

déclencheur : utilisée dans l'autre fonction : onClickDelete(id,couleur)

arguments : id, couleur

affichage attendu :



Obtenir le panier en mémoire

description : permet d'avoir le contenu du local storage

fonction à tester : `getStorage` `dataManager.js`

déclencheur : appelée par plusieurs fonctions

arguments : -

retour attendu : un objet contenant le panier

Envoi des données au serveur pour passer la commande

description : Fonction pour envoyer les données du panier avec des articles et formulaire

fonction à tester : `sendOrder` `datamanager.js`

déclencheur : appelée par `sendForm`

arguments : un objet contact et panier

retour attendu : change de page (confirmation) avec le numéro de commande / erreur

Les tests fonctionnels

Mise en forme de texte dans un format Html pour un produit sur la page index.html

fonction à tester : `templateProduit index.js`

déclencheur : appelé par `showPage()`

arguments : (`specsProduit`)

affichage attendu : affichage des produits par l'insertion du code html

Affichage de la page index avec tous les produits disponibles

fonction à tester : `showPage index.js`

arguments : application des résultats/données d'une fonction `getAllProducts` (API) de `datamanager.js` pour chaque produit d'une fonction `templateProduit`

déclencheur : affichage de la page

affichage attendu : affichage de toutes les données des produits sur la page `index.html`

Afficher l'article choisi sur la page produit selon son id : titre, description, prix, couleur

fonction à tester : `showArticle`

déclencheur : au chargement de la page `produit.html`

arguments : -

affichage attendu : affichage du produit sur la page produit en fonction de l'id dans l'url

Ajoute les options des couleurs d'un produit

description : ajoute les options des choix de couleurs dans le menu déroulant

fonction à tester : `showColors(colors)` dans `product.js`

déclencheur : appelé par `showArticle()`

arguments : un tableau de couleurs

affichage attendu : choix des couleurs

Ajoute un produit au panier

description : fonction pour ajouter un produit selon l'id avec les couleurs et quantité dans le panier en cliquant

fonction à tester : `onClickAdd` dans `product.js`

déclencheur : au clic

arguments : l'évènement du clic

affichage attendu : affiche une popup

Mise en forme de page panier (cart.js) dans un format Html (DOM)

description : mise en forme de la structure Html

fonction à tester : `productTemplate` dans `cart.js`

déclencheur : au chargement de la page `cart.html`

arguments : produit, couleur, quantité

affichage attendu : affichage des produits par l'insertion du code html



Affichage du panier

description : fonction pour afficher le contenu du panier enregistré dans local Storage par l'id du produit

fonction à tester : `basketDisplay cart.js`

déclencheur : au chargement de la page ou au clic sur la fenêtre popup

arguments :

affichage attendu : les produits sélectionnés par l'acheteur

Retire un produit du panier

description : fonction pour retirer un produit selon l'id avec les couleurs et quantité

fonction à tester : `onClickDelete cart.js`

déclencheur : au clic

arguments : id et couleur

affichage attendu : modifie l'affichage du panier

Met à jour le panier si on change la quantité

description : fonction pour mettre à jour le panier

fonction à tester : `updateQty cart.js`

déclencheur : de que la quantité, id ou couleur change

arguments : quantité, id, couleur

affichage attendu : affichage du panier avec des changements



Validation d'un formulaire

description : fonction pour valider nom, prénom, l'adresse, mail, ville du client

fonction à tester : `validForm(e)` avec méthode `addEventListener` `cart.js`

déclencheur : en remplissant des lignes du formulaire, inputs

arguments : l'évènement

affichage attendu : affiche des erreurs de remplissage du formulaire ou valide pour l'envoyer

Définit les données de contact/formulaire à envoyer au moment de passer la commande

description : fonction pour définir les données de contact (objet) à envoyer, fait appel à la fonction `sendOrder` (POST) de fichier `datamanger.js` pour passer la commande

fonction à tester : `sendForm` `cart.js`

déclencheur :

arguments : l'évènement du clic

affichage attendu : confirmation avec le N de commande

