

# KANAP

site e-commerce



Plan de test - Akrouit Julian

---

Fichier JS	Lignes de code	Fonction	Résultat attendu	Vérifier le résultat attendu	Erreur
script.js	4 à 15	getProducts	Récupération des données via l'API. La méthode fetch renvoie les promesses. Fonction asynchrone et résultat en objet JSON.	Console.log(data) permettra de savoir si la console comprend les données via l'API.	ok
script.js	19 à 30	addProducts	Affichage des éléments via une boucle	Affichage des éléments sur la page (titre,image,description,...)	ok
script.js	58 à 63	makeImage	Récupération des images produits.	La fonction appelée dans addProducts affichera l'image du produit dans sa carte.	ok
script.js	67 à 71	makeTitle	Récupération des titres produits.	La fonction appelée dans addProducts affichera le titre du produit dans sa carte.	ok
script.js	75 à 79	makeParagraph	Récupération des descriptions produits.	La fonction appelée dans addProducts affichera la description du produit dans sa carte.	ok
script.js	34 à 38	makeLink	Création du lien vers la page produit via l'url de la page produit et de l'ID du produit.	Redirection vers la page du produit sélectionné.	ok
product.js	19 à 30	addProductsChoices	Fonction main. Appel des fonctions pour la création de la carte produit (image, titre, couleur et quantité).	Affichage des éléments du produit sur la page (titre,image,description,...).	ok

FICHIER JS	LIGNES DE CODE	FONCTION	RÉSULTAT ATTENDU	VERIFIER LE RESULTAT ATTENDU	ERREUR
product.js	65 à 73	makeColors	Création des options de couleur du produit dans un menu déroulant	Un menu déroulant opérationnel afin de sélectionner une option de couleur.	« Veuillez choisir une quantité et/ou une couleur »
product.js	77 à 83	button.addEventListener	Récupération des valeurs quantité et couleur du produit via le bouton #addToCart ('click'). Appel de la fonction de sauvegarde.	La page affiche la couleur du produit ainsi que la quantité choisie.	ok
product.js	86 à 105	saveCartLocalStorage	Sauvegarde de l'objet (data) du produit et création d'un tableau. la méthode GET est utilisé ainsi que de la syntaxe JSON.stringify.	Sauvegarde du produit dans le local storage via son ID & sa couleur dans le tableau [cart].	ok
product.js	109 à 116	ifSelectIsEmpty	Vérification des champs quantité et couleur demandés.	Redirection vers la page panier.	« Veuillez choisir une quantité et/ou une couleur »
cart.js	155 à 163	deleteItemOnClick	Suppression au 'click' du produit sélectionné (findIndex) dans le tableau via 'splice'. La suppression du produit est acceptée si son ID et sa couleur sont logiques.	Suppression du produit de la page et nouveau calcul de la quantité et du prix total.	ok
cart.js	121 à 127	updatePriceAndQuantity	Modifie le prix et la quantité totale si modification de la quantité d'un ou de plusieurs produits.	Modification du prix et du nombre total des articles sur la page.	ok
cart.js	131 à 136	saveNewValueQuantity	Sauvegarde des nouvelles valeurs sélectionnées via 'filter' dans le localStorage.	Modification des quantités dans le local storage.	ok

FICHIER JS	LIGNES DE CODE	FONCTION	RÉSULTAT ATTENDU	VERIFIER LE RESULTAT ATTENDU	ERREUR
cart.js	270 à 274	displayTotalQuantity	Affichage du nombre total des articles sélectionnés.	Affichage du nombre total des articles sélectionnés sur la page et modification de la quantité dans le localStorage.	ok
cart.js	278 à 282	displayTotalPrice	Affichage du prix total des articles sélectionnés.	Affichage des prix des articles sélectionnés sur la page.	ok
cart.js	209 à 217	isEmailInvalid	Vérification de l'input « email » via un regex.	Pas de message d'erreur lors du remplissage du formulaire d'envoi.	« Entrez une adresse mail valide »
cart.js	221 à 229	isFirstNamesInvalid	Vérification de l'input « firstName » via un regex.	Pas de message d'erreur lors du remplissage du formulaire d'envoi.	« Entrez un prénom valide »
cart.js	233 à 241	isLastNamesInvalid	Vérification de l'input « name » via un regex.	Pas de message d'erreur lors du remplissage du formulaire d'envoi.	« Entrez un nom valide »
cart.js	178 à 204	submitForm(e)	Envoi du formulaire et création du bon de commande des articles sélectionnés.	Redirection vers la page confirmation.	« Veuillez renseigner ce champs »
confirm.js	2 à 16	displayOrderId	Récupération de l'orderId via la méthode GET dans le localStorage.	Affichage du numéro de commande sur la page.	ok

# MERCI

POUR VOTRE CONFIANCE



Plan de tests - Akrouit Julian

---