

Fichier JS	lles lignes de code	Fonction testée	Résultats attendus	Comment vérifier le résultat attendu	Problème probable
script.js	1 à 32	callProducts()	Récupération des données de l'API et leur affichage dans le DOM	Si les données s'affichent Si en appuyant sur la photo, un lien s'active	absence des données appel à l'API qui échoue
product.js	5 à 17	productComponents()	Affichage des données de chaque produit	Si les données s'affichent Si on peut sélectionner couleur et quantité	absence des données appel à l'API qui échoue
product.js	18 à 55	addProduct()	Garder dans le localStorage la quantité choisie	L'utilisateur est redirigé vers la page du panier La quantité est enregistrée dans le localStorage	alerte si l'utilisateur ne choisit pas quantité et couleur
product.js	58 à 70	addSameProduct()	Incrémenter la quantité d'un même article dans le localStorage	Si la quantité augmente dans le localStorage Si l'ajout n'écrase pas les données de l'article	Les données du produit soient écrasées dans le localStorage
product.js	72 à 77	setLocalStorage()	pousser les données de l'article dans localStorage	Si on observe que le localStorage possède une paire clé/valeur avec des données dedans	Si on envoie des données complexes qu'on a pas mis sous forme de string
cart.js	6 à 12	emptyCart()	affiche message 'votre panier est vide' disparition du formulaire	Si nous voyons le message s'afficher sinon notre panier contient des choses donc le formulaire et les produits s'affichent	
cart.js	18 à 96	une boucle for	Boucler à travers tous les articles du LS Afficher les articles dans le DOM	Si l'article ou les articles sont visibles	Si l'appel à l'API échoue
cart.js	108 à 120	setInputValue()	effacer les valeurs dans les inputs de quantité si ce sont autre chose que des chiffres	tenter d'entrer des lettres ou des caractères speciaux, ils vont être effacés	
cart.js	123 à 134	cartCount()	Addition des quantités et des prix des articles envoyés depuis product.js	regarder dans le localStorage si les données correspondent aux données du DOM	
cart.js	137 à 154	deleteOneArticle()	Suppression de l'article sélectionné	Bien regarder le DOM et le LocalStorage	
cart.js	157 à 171	changeQuantity()	Mise à jour des quantités d'articles au click	Bien regarder le DOM et le LocalStorage	
cart.js	174 à 210	calculateTotals()	calcul des totaux après avoir cliqué sur les inputs Gestion du cas des inputs qui sont vides accord du pluriel et du singulier d'articles	Bien regarder le DOM et le LocalStorage	
cart.js	230 à 298	validationFormulaire()	Test des Regex pour les champs du formulaire	Message d'erreur affiché en cas d'inadéquation avec ce qui est attendu par les Regex	Si l'utilisateur persiste à entrer des données non-valides
cart.js	301 à 342	sendFormulaire()	Envoi du formulaire valide au backend Envoi d'une commande d'articles sélectionnés	Un numéro de commande nous est envoyé	S'il y a un problème le numéro de commande sera Undefined
confirmation.js	6 à 12	searchId()	Affichage du numéro de commande Confirmer que notre commande est validée Vider le panier après la commande	Si le numéro de commande s'affiche Si le localStorage est vide	