

Fichier JS	Lignes de code testées	Fonction testée	résultat attendu	Comment vérifier le résultat attendu	Problème possible
script.js	7 à 46	listArticles	tous les articles renvoyés par l'API sont ajoutés dans la liste des articles	vérifier que la taille de la liste des articles corresponde au nombre d'articles injectés	la taille de la liste des articles n'est pas égale au nombre d'articles attendus
produit.js	12 à 106	choiceProduct	récupération des données de l'ours choisi par l'utilisateur en page index; choix de la couleur et enregistrement du produit dans le localStorage	vérifier dans le localStorage que l'élément choisi y est bien stocké	l'élément n'est pas stocké ou les caractéristiques choisies sont erronées
produit.js	48 à 52	boucle selection option	liste déroulante pour le choix des options disponibles	vérifier l'insertion dans le Html d'une balise select dans laquelle sont listées les options	pas d'option visible ou pas de liste déroulante
produit.js	75 à 105	newBtnSubmit.addEventListener	ajoute le ou les produits sélectionné par l'utilisateur dans le localStorage	vérifier dans le localStorage que les éléments choisis y sont bien stockés	le contenu du localStorage ne correspond pas aux choix de l'utilisateur
produit.js	96 à 103	instructions pour mettre un ou plusieurs articles dans le localStorage	ajoute le ou les produits sélectionné par l'utilisateur dans le localStorage	tester avec un seul article puis avec plusieurs articles dans le panier et vérifier que le contenu du localStorage correspond au panier	le contenu du localStorage ne correspond pas au contenu du panier
panier.js	20 à 26	insertion message si panier vide	si le panier est vide, disparition du tableau listant les articles du panier et inscription d'un message: "Votre panier est vide"	vérifier dans le html que les éléments qui doivent être cachés sont en "display:none" et que le message apparaît bien	le message n'apparaît pas; les éléments du tableau sont toujours visibles
panier.js	30 à 61	insertElsBasket	insère dans le DOM chaque article acheté par l'utilisateur	introduire des articles dans le localStorage puis vérifier que la fonction met à jour correctement le tbody du tableau	la fonction ne met pas à jour le tbody avec tous les éléments contenus dans le localStorage
panier.js	73 à 94	calcTotal	calcul du total des prix des articles contenus dans le localStorage	vérifier que la variable newPriceTotal est égale au total attendu	le calcul est faux
panier.js	105 à 116	AllBtnSupprProd[b].addEventListener	en cliquant sur un des boutons "supprimer du panier", 'produitpanier' doit être mise à jour et l'article de la ligne concerné doit être retiré de 'produitpanier'.	vérifier que la variable 'produitpanier' a bien été mise à jour	l'article n'a pas été supprimé du panier ou tout le panier a été supprimé
panier.js	121 à 127	btnEmptyBasket.addEventListener	au click, 'productBasket' doit être supprimé du localStorage et la page doit se rafraîchir	vérifier que la variable 'productBasket' n'est plus dans le localStorage	la variable est toujours dans le localStorage; la page n'est pas rafraîchie
panier.js	136 à 224	btnValidBasket.addEventListener	au click, le formulaire d'inscription doit apparaître puis après remplissage par l'utilisateur, les données panier et utilisateur doivent être envoyées au back-end qui doit renvoyer un id unique de commande	vérifier le contenu des variables contact et product; vérifier le retour du back-end doit renvoyer un id de commande	le formulaire n'apparaît pas; le retour du backend est inexistant ou incorrect
panier.js	138 à 167	création et ouverture du formulaire	apparition du formulaire au click de validation	vérifier l'insertion dans le Html de la balise <form> avec tout son contenu	le formulaire n'apparaît pas
panier.js	176 à 223	validation commande pour envoyer au serveur	récupération des données d'identification du formulaire et envoi à l'API avec le contenu du panier ; récupération de l'id de commande	vérification du contenu de l'objet "command" avant envoi au backend et vérification du contenu du retour du backend	les données récupérées dans le formulaire sont manquantes ou inexactes; pas de retour ou retour incorrect du backend
commande.js	17 à 18	récupération de l'id de commande renvoyé par l'API	l'API renvoi un Id pour identifier la commande	présence d'un id dans la réponse de l'API	pas d'id dans le retour ou un id:undefined
commande.js	21	création du message de validation	message de validation de la commande avec id renvoyé par le backend et le prix total du panier	affichage du message avec les 2 variables souhaitées	pas d'affichage de message; manque d'une des 2 variables