

Fichier JS	Lignes de code testées	Fonction testée	Résultat attendu	Comment vérifier le résultat attendu?	Problème possible
items.js	lignes 3 à 9	main()	Contient le code de base et s'exécute dès le chargement de la page		
items.js	lignes 11 à 19	getArticles()	Récupère les articles via une URL		
items.js	lignes 22 à 33	displayArticle(article)	Affiche tous les articles en créant une carte par produit en reproduisant le même modèle html (clone + template) pour chaque peluche avec des valeurs différentes (image, name, price, description, colors et id)	Les 5 produits s'affichent correctement sur la page.	Rien ne s'affiche
product.js	lignes 4 à 10	getData()	Va chercher les données dans l'URL		
product.js	lignes 15 à 77	productPage()	Réparti les valeurs de chaque élément au bon endroit.	Une page produit bien affiché avec toutes les informations et la possibilité de changer la couleur.	Erreurs d'affichage.
cart.js	lignes 122 à 124	regexString(value)	Met en place les caractères alphabétiques autorisés pour les noms propres (noms et villes) ainsi que le nombre de caractères minimum et maximum.	Entrer des caractères non autorisés et voir si le message d'erreur s'affiche.	La mise en place du RegExp ne fonctionne pas et les données erronées sont acceptées et validées.
cart.js	lignes 156 à 158	regexCp(value)	Rejète les caractères alphabétiques et spéciaux dans les codes postaux	Entrer des caractères non autorisés et voir si le message d'erreur s'affiche.	La mise en place du RegExp ne fonctionne pas et les données erronées sont acceptées et validées.
cart.js	lignes 170 à 172	regexEmail(value)	Donne un format de caractères alphanumériques séparés par un @	Entrer des caractères non autorisés et voir si le message d'erreur s'affiche.	La mise en place du RegExp ne fonctionne pas et les données erronées sont acceptées et validées.
cart.js	lignes 133 à 185	checkLastName(), checkFirstName(), checkCity(), checkCp(), checkEmail()	Vérifient que les données entrées par le client sont correctes. Pas de chiffres dans les noms et les villes, pas de lettres dans les CP, un @ dans les adresses email ...	Un formulaire correctement rempli avec des données non erronées.	Le formulaire sera validé même avec des erreurs.
cart.js	lignes 206 à 238	sendDataOff(sendOrder)	Envoie les données avec une requête POST	L'utilisateur passe sa commande et envoie ses coordonnées.	La commande c'est pas prise en compte et un numéro de commande (id) n'est pas généré.
cart.js	lignes 246 à 248	addData(input)	Met les données du localStorage dans le formulaire	Les informations personnelles de l'utilisateur sont sauvegardées.	Les données s'effacent si on change de page et revient sur le formulaire.
conf.js	lignes 15 à 17	emptyConfPage(key)	Supprime les données du localStorage	Une fois la commande validée, les articles et l'id sont supprimées dans le localStorage.	Les données sont toujours sur le localStorage et s'accumuleront.