

Fichier.js	Lignes de code testées	Fonction testée	Résultat attendu	Comment vérifier le résultat attendu	Problème possible
Index.js	1 à 73	Récupération des données server	La fonction doit récupérer les données des différents produits contenus dans l'API.	La requête fetch renvoie une promise qui fournit une fonction then, exécutée quand le résultat est obtenu, et une fonction catch, exécutée s'il y a une erreur.	Réponse d'erreur côté client de type 400 ou réponse d'erreur côté serveur de type 500
Index.js	10 à 68	Création des produits Teddies	La fonction doit créer les éléments du DOM avec une boucle for et faire correspondre le nombre de Teddies de l'API	Appeler la fonction en modifiant les valeurs de la boucle	Les éléments du DOM peuvent être modifiés voir inexistants
Produit.js	1 à 11	Transfert des données des produits de page en page	La fonction permet de récupérer l'id du Teddy sélectionné en page index.html	Appeler la fonction avec différentes valeurs d'id et vérifier la valeur retournée, avec un console.log(idTeddy)	Pas d'affichage du produit sélectionné via index.html
Produit.js	14 à 195	Récupération des données server	La fonction doit récupérer les données du produit sélectionné contenu dans l'API.	La requête fetch renvoie une promise qui fournit une fonction then, exécutée quand le résultat est obtenu, et une fonction catch, exécutée s'il y a une erreur.	Réponse d'erreur côté client de type 400 ou réponse d'erreur côté serveur de type 500
Produit.js	20 à 89	Création du produit Teddy	La fonction doit créer les éléments du DOM et afficher les données du produit sélectionné et contenu dans l'API.	Appeler la fonction avec différentes valeurs d'id et vérifier la valeur retournée, avec un console.log(myTeddy)	Les éléments du DOM peuvent être modifiés voir inexistants
Produit.js	64 à 70	Création des options de couleurs	La fonction doit créer les éléments du DOM avec une boucle for et afficher les options du produit contenues dans l'API.	Appeler la fonction en modifiant les valeurs de la boucle avec différentes valeurs d'id	Pas d'affichage des options de couleurs correspondantes au produit
Produit.js	83 à 89	Création des options de quantité	La fonction doit créer les éléments du DOM avec une boucle for et afficher les quantités	Appeler la fonction en modifiant les valeurs de la boucle	Pas d'affichage des options de quantités correspondantes au produit
Produit.js	105 à 182	Récupération des choix utilisateur dans le localStorage	La fonction écoute le click du bouton «ajouter au panier» et stocke un tableau «productCart» dans le localStorage	Appeler la fonction avec différents produits et vérifier la valeur retournée avec un console.log(infosTeddy) ou vérifier directement dans le localStorage.	Pas de récupération des choix utilisateur dans le localStorage

Fichier.js	Lignes de code testées	Fonction testée	Résultat attendu	Comment vérifier le résultat attendu	Problème possible
Produit.js	141 à 176	Ajout de produit(s) dans le localStorage	La fonction écoute le click du bouton «ajouter au panier» et soit modifie un produit existant soit crée un nouveau produit dans le tableau «productCart» du localStorage	Appeler la fonction avec différents produits et vérifier le localStorage ou vérifier via la page panier.html	Création d'une nouvelle entrée dans le localStorage identique à une entrée déjà existante ou à son écrasement
Panier.js	1 à 3	Stockage des données du localStorage	La fonction doit récupérer les données concernant les choix utilisateur et les stocker dans le localStorage	Appeler la fonction avec différents produits et vérifier la valeur retournée avec un console.log	Le localStorage reste vide
Panier.js	5 à 64	Création des produits sélectionnés	La fonction doit créer les éléments du DOM avec une boucle for et faire correspondre les différents produits choisis par l'utilisateur.	Appeler la fonction en modifiant les valeurs de la boucle	Les éléments du DOM peuvent ne pas correspondre aux choix utilisateurs voir ne pas s'afficher
Panier.js	65 à103	Suppression d'un produit	La fonction écoute le click du bouton «supprimer» et supprime un produit existant en fonction de son id et de son option	Appeler la fonction en présence de 2 produits ayant la même id mais avec une option différente et/ou ayant des id différentes mais avec la même option	Suppression de 2 produits ayant une option différente car ayant la même id et/ou suppression de 2 produits car ayant la même option
Paniers.js	104 à 126	Suppression du panier	La fonction permet de vider le panier , afficher un message 'votre panier est vide '	Tester l'écouteur d'événement	Le panier pourrait ne pas se vider
Paniers.js	128 à 154	Affichage prix total	La fonction affiche le prix total du panier en traitant chaque valeur du panier pour les réduire à une seule en les ajoutant	Ajouter ou supprimer des produits dans le panier et comparer le résultat avec la somme du prix des articles	La valeur retournée pourrait ne pas correspondre au total attendu
Paniers.js	178 à 256	Validation du formulaire	La fonction permet de vérifier que tous les champs du formulaire sont corrects avant envoi dans le localStorage	Compléter le formulaire avec différentes valeurs de tests, valeur vide, valeur erronée	Le formulaire pourrait être validé malgré des champs vide ou erronée
Paniers.js	156 à 256	Validation de la commande	La fonction écoute le click du bouton «valider» et envoie les données du formulaire, l'id des produits du panier et le montant total de la commande dans le localStorage	Vérifier la correspondance des données envoyées dans le localStorage	Le montant et les données pourraient être erronées ou inexistantes
Paniers.js	258 à 324	Dépôt sur serveur	La fonction envoie l'objet «contact» et l'id des produits du panier au serveur via la méthode POST	Vérifier l'état de la promesse avec un console.log ou vérifier le statut code de la requête POST	L'envoi des données vers le serveur peut échouer
Confirm.js	1 à 6	Récupération données du localStorage	La fonction récupère l'orderId généré par le serveur et le montant total de la commande dans le localStorage	Vérifier la valeur retournée avec un console.log	Les valeurs retournées pourraient ne pas correspondre

Fichier.js	Lignes de code testées	Fonction testée	Résultat attendu	Comment vérifier le résultat attendu	Problème possible
Confirm.js	7 à 17	Affichage des données	La fonction doit créer les éléments du DOM en affichant le numéro de la commande et le montant de la commande	Vérifier la correspondance des données	Les éléments du DOM peuvent ne pas correspondre
Confirm.js	18 à 24	Cleaning du localStorage	La fonction vide le localStorage hormis les données du formulaire avec la méthode removeItem	Retour à la page panier et/ou vérifier l'état du localStorage	L'ensemble des informations pourraient être toujours contenues dans le localStorage