

Fichier JS	Lignes de code testées	Fonction testée	Résultat attendu	Comment vérifier le résultat attendu	Problème possible
main.js	45 à 49	numberItemInCartNav	Affiche le nombre d'articles différents dans le panier dans la barre de navigation	Ajouter ou retirer des articles du panier, le nombre doit afficher le nombre d'articles différents correctement	N'affiche pas le nombre correct d'articles
main.js	52 à 66	newElement	Fonction permettant la création d'un nouvel élément avec son type, son parent, ses attributs et son contenu	L'élément est correctement généré	Ne crée pas l'élément demandé ou le crée de façon incorrect
main.js	69 à 76	newValue	Ajoute d'une nouvelle valeur à un élément via son ID	La valeur est correctement ajoutée	N'ajoute pas la valeur demandée
main.js	79 à 82	removeItem	Retirer un article du panier via son ID	L'article est correctement retiré du panier	Ne retire pas l'article du panier ou retire le mauvais article
main.js	85 à 113	sendFormData	Envoyer la commande au serveur avec un fetch, method POST au format JSON Récupération des coordonnées du client pour les stocker dans le sessionStorage Ouverture de la page "order-confirmed.html"	La commande est correctement envoyée au serveur avec toutes les informations demandées et la page "order-confirmed.html" s'ouvre	La commande n'est pas envoyée ou est envoyée avec des informations incorrectes ou manquantes ou la page "order-confirmed.html" ne s'ouvre pas
main.js	116 à 130	auto_format	Fonction auto-formatant un input	Ajouter des caractères au champ "Numéro de votre carte bancaire" ou "Date d'expiration"	Les champs faisant appels à la fonction ne sont pas correctement formatés
main.js	133 à 136	clearLocalStorage	Suppression du LocalStorage et du SessionStorage (Vide le panier une fois la commande passée)	Le panier est correctement vidé de son contenu	Le LocalStorage et le SessionStorage ne sont pas vidés (Le panier contient alors encore des articles)
index.js	3 à 34	fetch API Teddies List	Récupération de la liste des ours en peluche au format JSON :  - Si la liste est correctement récupérée : Affichage des articles sous forme de cartes, contenant la photo, le nom, le prix et le lien vers l'article  - Si il y a une erreur : Affichage d'un message d'erreur	- Si la liste est correctement récupérée, tous les articles sont affichés correctement avec les informations  - Si une erreur est survenue dans la récupération de la liste des articles, le message d'erreur est correctement affiché	La liste récupérée n'est pas la bonne, est incomplète ou le message d'erreur ne s'affiche pas correctement
index.js	11 à 25	fetch API Teddies List, newElement	Si le Fetch ne retourne pas d'erreur : Boucle générant les cartes des articles en HTML	Les cartes sont correctement générées	Les cartes ne sont pas correctement générées ou pas du tout
index.js	28 à 34	fetch API Teddies List, newElement	Si le Fetch retourne une erreur : Génération d'un message d'erreur en HTML	Le message d'erreur est correctement généré	Le message d'erreur n'est pas correctement généré ou pas du tout
item.js	3 à 68	fetch API Teddies Item Info	Récupération des informations de l'article (nom, photo, prix, couleurs, description) au format JSON :  - Si les informations sont correctement récupérées : Affichage de ces informations  - S'il y a une erreur : Affichage d'un message d'erreur	- Si les informations sont correctement récupérées : toutes les informations sont affichées correctement  - Si une erreur est survenue dans la récupération des informations de l'article, le message d'erreur est correctement affiché	Les informations sont incomplètes, concernent un autre article ou le message d'erreur ne s'affiche pas correctement
item.js	11 à 52	fetch API Teddies Item Info, newValue	Si le Fetch ne retourne pas d'erreur : Affichage des informations de l'article et ajout du nom de l'article dans le titre de la page	Les informations récupérées sont affichées correctement et le nom de la page contient le nom de l'article	- Les informations récupérées ne sont pas affichées correctement ou de façon incomplète - Le nom de la page ne contient pas le nom de l'article ou contient le nom d'un autre article
item.js	55 à 68	fetch API Teddies Item Info, newElement	Si le Fetch retourne une erreur : - Suppression du bloc #item-container - Génération d'un message d'erreur en HTML	Le message d'erreur est correctement généré et le bloc #item-container a été supprimé	Le message d'erreur n'est pas correctement généré ou pas du tout ou le bloc #item-container n'a pas été supprimé
item.js	17 à 34	addToLocalStorage	Ajoute l'article demandé (article et couleur) et sa quantité au panier via le localStorage Si l'article est déjà présent dans le panier, alors est ajoutée la quantité demandée	L'article ajouté au panier est bien le bon article, avec la bonne couleur et la quantité demandée Si l'article était déjà dans le panier, la quantité demandée a correctement été ajoutée au panier	- Aucun article n'a été ajouté au panier - L'article ajouté au panier n'est pas le bon - La quantité de l'article est incorrecte
item.js	47 à 52	addEventListener, addToLocalStorage, numberItemInCartNav	Utilisation de la fonction addToLocalStorage et numberItemInCartNav au clic sur le bouton "Ajouter au panier"	Le bouton fonctionne correctement dans ce cas : - L'article ajouté au panier est bien le bon article, avec la bonne couleur et la quantité demandée - Si l'article était déjà dans le panier, la quantité demandée a correctement été ajoutée au panier - Le nombre d'articles différents dans le panier est correctement mis à jour dans la barre de navigation	Le bouton ne fonctionne pas correctement ou pas du tout : - Aucun article n'a été ajouté au panier - L'article ajouté au panier n'est pas le bon - La quantité de l'article est incorrecte - Il ne se passe rien au clic sur le bouton - Le nombre d'articles différents dans le panier n'est pas correctement mis à jour dans la barre de navigation
cart.js	3 à 11	newElement	Si le panier est vide : - Supprime le bloc #cart - Affiche un message indiquant que le panier est vide	Le bloc #cart est correctement supprimé et le message d'erreur s'affiche correctement alors que le panier est bien vide	- Le bloc #cart n'est pas supprimé, ou est supprimé alors que le panier n'est pas vide  - Le message d'erreur ne s'affiche pas correctement ou est affiché alors que le panier n'est pas vide
cart.js	12 à 72	newElement	Si le panier n'est pas vide : - Affichage de la liste des articles dans le panier et du formulaire de commande	La liste des articles et le formulaire de commande sont correctement affichés	La liste des articles n'est pas correctement affichée ou est incomplète
cart.js	14 à 32	newElement, newValue	Boucle générant la liste des articles et les prix dans le panier en HTML et push les ID des articles du panier dans le tableau "products"	La liste des articles est correctement affichée, complète et les prix sont corrects Le contenu du tableau "products" est correctement mis à jour	La liste des articles n'est pas correctement affichée, incomplète ou les prix sont incorrects Le contenu du tableau "products" n'est pas correctement mis à jour ou est vide
cart.js	35 à 37 et 50 à 52	addEventListener	Bouton ajoutant la classe "was-validated" à son formulaire au clic	La classe "was-validated" est correctement ajoutée au formulaire au clic sur le bouton	La classe "was-validated" n'est pas correctement ajoutée au formulaire au clic sur le bouton
cart.js	40 à 47	auto_format	Appel de la fonction auto_format : Les fonctions sont correctement exécutées	La fonction auto_format est correctement exécutée au chargement de la page	La fonction auto_format n'est pas correctement exécutée au chargement de la page  La commande est envoyée alors que le formulaire n'est pas validé
cart.js	55 à 111	addEventListener, sendFormData	Au clic sur "Confirmer la commande", envoie le tableau "products" contenant la liste des articles du panier et les coordonnées du client au serveur grâce à la fonction sendFormData si les données du formulaire sont validées	Le tableau est correctement envoyé au serveur (uniquement si le formulaire est bien validé, la page "order-confirmed.html" s'ouvre et affiche les coordonnées du client et le numéro de commande	Le tableau n'est pas correctement envoyé au serveur : - Les données sont incomplètes ou incorrectes - Le serveur ne retourne pas de numéro de commande - La page "order-confirmed.html" ne s'ouvre pas ou n'affiche pas les données demandées
order-confirmed.js	4 à 10	newValue	Si order_id n'est pas null : Récupère les informations de la commande (numéro de commande, prix total, nom du client et adresse de livraison) via la sessionStorage	Si order_id n'est pas null : Les informations de la commande sont correctement affichées et complètes	Les informations de la commande ne sont pas correctement affichées, sont incomplètes ou sont affichées alors que order_id n'est pas null
order-confirmed.js	13	clearLocalStorage	Appel de la fonction clearLocalStorage : La fonction est correctement exécutée	Le panier est correctement vidé de son contenu sur la page "order-confirmed.html"	Le LocalStorage et le SessionStorage ne sont pas vidés (Le panier contient alors encore des articles)
order-confirmed.js	16 à 22	newElement	Si order_id est null : - Supprime le bloc #order - Affiche un message indiquant que la commande n'est pas validée	Le bloc #order est correctement supprimé et le message d'erreur s'affiche correctement alors que order_id est bien null	- Le bloc #order n'est pas supprimé, ou est supprimé alors que order_id n'est pas null  - Le message d'erreur ne s'affiche pas correctement ou est affiché alors que order_id n'est pas null