

Fichier JS	Is de code testé	Fonction testée	Résultat attendu	Comment vérifier le résultat attendu	Problème possible
script.js	15	Changement de page	Au clic, arriver sur la page produit.html du teddie correspondant	Création d'un noeud <a> et création de l'attribut qui permettra d'ouvrir la page html selon l'id du produit. S'il y a une erreur de récupération de l'id la console va afficher le message : "Affichage de l'id du produit : null"	Erreur d'écriture du lien et/ou de cible de l'id
script.js	43 à 49	fetch('http://localhost:3000/api/teddies')	En utilisant fetch, je récupère les données de l'API (data). L'utilisation de ces données va me permettre d'afficher le nom, l'image, la description, ...	Le contrôle s'effectue en regardant le message dans la console qui nous retourne le message de succès de la requête ainsi que les données récupérées (console.log : Chargement des données + affichage de l'objet)	Avec ".catch", s'il y a une erreur dans l'url de l'api, la réponse en json ou dans la fonction data. La console va renvoyer un message d'erreur console.log('Erreur, vérifier le fetch, les promesses')
produit.js	4 à 5	Récupération des paramètres de l'url	Récupération et affichage du produit via les paramètres url	Le contrôle s'effectue en regardant le message dans la console. Console.log pour afficher l'url qui est récupérée. Utilisation de GET pour obtenir l'objet par son id et console.log pour vérifier que l'article est bien récupéré par son ID.	Mauvaise syntaxe ou mauvaise requête pour récupérer l'id (cela peut-être en lien avec la page produits.js si le lien est mal codé)
produit.js	18 à 25	Mise en place du session storage et création du panier	A l'ajout d'un article, celui-ci sera conservé temporairement dans le session storage [panier]	Le contrôle s'effectue en regardant le message dans la console. Console.log pour vérifier que le panier existe et si l'on peut y placer les articles (session storage)	Pas de mise en place du panier dans le session storage. Blocage de l'envoi des articles dans le panier (monPanier)
produit.js	71 à 72	Récupération des couleurs, fonction : couleur	Les couleurs des articles doivent être apparentes (liste déroulante)	Le contrôle s'effectue en regardant le message dans la console. Console.log indiquera si les couleurs sont récupérées, sinon en réponse on signalera une erreur et qu'il faut vérifier le fetch et/ou les promesses	Pas de récupération des couleurs depuis l'api. Il faut vérifier la syntaxe, la requête
produit.js	83	Ajout des produits dans le panier, fonction écoute du bouton	Au clic sur le bouton l'article sera pushé dans le session storage [panier]. Une alerte apparaîtra en indiquant que l'action a bien fonctionné	Visualisation d'une alerte	La fonction n'ajoute pas au panier, cela peut provenir de la syntaxe
produit.js	90 à 91	Article dans le session storage	Dans la console, on va contrôler sessionStorage.getItem("panier") pour voir son contenu	La vérification se fera dans la console, l'article présent sera ajouter dans le [panier] sinon cela rendra [] (panier initialisé mais vide)	Vérifier que le panier est existant et que l'envoi est codé correctement (ex : push, getItem au lieu de setItem)
produit.js	94 à 98	fetch('http://localhost:3000/api/teddies/' + teddielid)	En utilisant fetch, je récupère les données de l'API (data). L'utilisation partielle de ces données va me permettre d'afficher le nom, l'image, la description, le prix et la couleur	Le contrôle s'effectue en regardant le message dans la console qui nous retourne le message de succès de la requête ainsi que les données (console.log)	Avec ".catch", s'il y a une erreur dans l'url de l'api, la réponse en json ou dans la fonction data. La console va renvoyer un message d'erreur console.log('Erreur, vérifier le fetch, les promesses')
panier.js	4 à 9	Récupération des produits dans la page panier	Visualisation des articles ou d'une alerte	Si le panier est vide : affichage d'une alerte et le montant total du panier sera de 0€. La console affichera le panier si des élément y sont stockés avec "console.log"	Mauvaise récupération du session storage qui peut provenir par exemple d'un setItem au lieu d'un getItem par exemple
panier.js	37	Fonction totalPrice	A l'ajout d'un article, le prix va être conservé dans le local storage. A l'ajout de plusieurs articles les montants seront additionnés automatiquement avec le '+'	La console doit afficher le résultat et 'price' doit apparaître dans le local storage. Ex: ajout dant le panier 2 Arnold à 39€ pièce = montant total de 78€	Pas d'addition ni de récupération du prix
panier.js	47 à 49	REGEX	Contrôle des champs input, des données que peut saisir l'utilisateur	S'il y a une erreur de remplissage, l'utilisateur aura un message d'erreur lui indiquant de vérifier le champ	Erreur d'écriture des regex ou écriture trop complexe qui peut bloquer l'utilisateur dans son remplissage de formulaire
panier.js	61 à 154	formControle	Le formulaire sera envoyé que si les données entrées sont valides.	Dans la console, le contrôle des champs valides se fera par la lecture ex : "Prénom ok, Nom ok". Si des champs sont vides ou invalides, il sera indiqué : "nom input erreur".	L'utilisateur a oublié de remplir des champs, ou ils contiennent des erreurs (regex)
panier.js	149 à 151	fetch('http://localhost:3000/api/teddies/orderId')	Les données collectées pour le serveur doivent être envoyées dans le session storage pour récupérer l'orderId	S'il y a une erreur, .catch va indiquer dans la console que la requête a échouée et la nature de l'erreur	Envoyer les mauvaises données ex sessionStorage.setItem('orderId', orderId) au lieu de sessionStorage.setItem('orderId', orderId), va retourner : La requête a échouée ReferenceError: orderId is not defined. Faire une requête GET au lieu de POST

Fichier JS	Is de code testé	Fonction testée	Résultat attendu	Comment vérifier le résultat attendu	Problème possible
commande.js	12 à 16	retour serveur	Affichage de l'orderId	Le console.log va permettre de vérifier que l'orderId est bien réceptionné en s'affichant dans la console	Problème lors de l'envoi des éléments dans le session storage