| Fichier JS | Lignes de code testées | Fonction testée | Résultat attendu | Comment vérifier le résultat attendu | Problème possible |
|---|---|---|---|---|---|
| index.js<br>product.js<br>cart.js | 2<br>11<br>15 | fetch(url)<br>fetch(url)<br>fetch(url + id) | L'API Fetch fournit une interface JavaScript pour l'accès et la manipulation des parties de la pipeline HTTP, comme les requêtes et les réponses. Cela fournit aussi une méthode globale fetch() qui procure un moyen facile et logique de récupérer des ressources à travers le réseau de manière asynchrone. | Il faut afficher la page, si fetch ne donne pas de retour de réponse ou de promesse, toutes les infos des produits ne s'afficheront pas. Il faut utiliser des consol. log | les produits ne s'affichent pas car fetch ne renvoie pas de données |
| index.js | 50 à 57 | .catch(function(error)) | Une promesse fetch() va retourner une TypeError quand un problème réseau s'est produit. En cas de rejet de la promesse on affiche une div avec un message d'erreur | ne pas taper la ligne de commande "node app.js" dans le terminal afin de tester l'affichage du message d'erreur de la div | la div contenant le message ne s'affiche pas |
| product.js | 4 à 9 | getId() | cette fonction permet de récupérer l'id et les parametres passés dans l'url du produit | Dans la console on doit avoir un message de navigation, indiquant dans quel URL nous sommes | l'id et les paramétres ne s'affichent pas |
| product.js | 17 à 149 | afficherLeProduit(peluche) | Cette fonction permet de créer la card compléte de l'article choisi sur la page d'accueil, on y retrouve nom, options, description, photo, grâce à la bonne récupération des données | il faut afficher une page d'un article | les élements ne s'affichent pas |
| product.js | 121 | btnAjout.addEventListener("clic", envoiDuProduit) | Au clic sur le btnAjout la fonction envoiDuProduit et l'arte sont appelées | le produit choisi sera envoyé dans le local storage, si pas de quantité séléctionnée il y aura un message d'erreur, si une quantité à été sélectionnée alors le message d'alerte "ajouté au panier sera affiché"<br>il faut placer un console.log dans l'écoute de l'évenementafin de s'assurer que l'on est bien dans le "click" | les fonctions ne sont pas appelées |
| product.js | 122 à 148 | envoiDuProduit(event) | Elle ajoute un tableau stringifié au format JSON dans le local storage. Ce tableau représente les éléments qui sont dans le panier, on y trouve l'id, la couleur et la quantité. Si un produit est déjà dans le panier on augmente sa quantité. | en cliquant sur le bouton "Ajouter" on va appeler la fonction plusieurs fois, pour différentes couleurs de peluches, c'est dans l'onglet application de DevTools que l'on peut vérifier les valeurs.<br>Si un produit est déjà dans le panier et qu'on en rajoute un identique, on pourra voir dans la console "déja présent" et la qutantité sera augmenté | La récupération des données peut être mal faite ou incomplète, par exemple récupérer le mauvais id, la mauvause couleur... |
| cart.js | 14 à 16 | getById(id) | permet de récuperer les donnnées de l'api ainsi que l'id du produits séléctionné au préalable | l'appel de fetch renvoie bien les données de l'API et l'id, la page s'affiche correctement | Fetch ne fonctionne pas, l'id n'est pas récupérer |
| cart.js | 17 à 87 | addTeddie(teddie) | permet la mise en page du panier sous forme d'un tableau. On y retrouve les données principales ( nom, prix, qté, totaux) | ajout de produit au panier afin de voir s'ils sont pris en compte et s'ajoute dans le tableau en mettant à jours les prix | les produits ne s'ajoutent pas dans le tableau, les prix ne se mettent pas à jour |
| cart.js | 70 à 82 | cardBtn.onclick | permet de supprimé 1 élement du panier si celui si ne convient pls | vérifier que seulement 1 article est suprimé et pas tous. Si plus d'article, le panier sera donc égale à 0 on enléve la key panier | le bouton ne fonctionne pas , la clé est vide mais ne se supprime pas |
| cart.js | 92 à 169 | upPanier | permet d'afficher de manière dynamique le contenu du local storage, si le panier est vide " votre panier est vide" si le panier comporte des article alors on appelle la fonction addTeddie(teddies) | dans l'onglet application de devTools il faut vérifier que les valeurs qui sont contenu dans la key panier correspondent aux produits affichés, les élements (id, color, quantité) doivent être présent | mauvais affichage des produits |
| cart.js | 114 à 117 | validateEmail | cela permet de comparer le format de l'email que l'utilisateur rentre et une expréssion régulière définie. Si l'émail est bon : email valide, si l'email n'est pas bon : email invalide | Essayer de rentrer des adresses emails incorrectes | l'email est considéré valide alors qu'il ne remplis pas les critéres, ou inversement. Cela peut être dû à une mauvaise déclaration de l'expression de référence ou un mauvais appel de la fonction |
| Confirm.js | 13 à 19 | endOfOrder.addEventListener("click") | Lors du clic sur le bouton "revenir à l'accueil", le local storage sera vidé entiérement et l'utilisateur redirigé | dans l'onglet application de devTools on vérifie que le local Storage est bien vde. On vérifie également à l'aide d'un console.log() que le bouton est bien écouté et que l'on rentre dans la fonction au moment du clic | le localStorage reste tel quel et ne se vide pas, l'écoute du bouton ne se fait pas |