

Fichier JS	Lignes de code testées	Fonction testée	Résultat attendu	Comment vérifier le résultat attendu	Problème possible
index.js	15 à 46	displayCarrousel	La fonction doit créer en html un conteneur avec dedans le nom, l'image, le lien vers la page et le prix de l'élément passé en paramètre. Ce conteneur devra être ajouté au carrousel.	Appeler la fonction avec l'objet 'testObject' et observer si les éléments se créent bien avec les données de 'testObject'	Les éléments pourraient ne pas se créer dû aux valeurs erronées passées en paramètres.
index.js	50 à 77	displayCatalog	La fonction doit créer en html un conteneur avec dedans le nom, l'image et le lien vers la page de l'élément passé en paramètre. Ce conteneur doit être ajouté au catalogue.	Appeler la fonction avec l'objet 'testObject' et observer si les éléments se créent bien avec les données de 'testObject'	Les éléments pourraient ne pas se créer dû aux valeurs erronées passées en paramètres.
index.js	81 à 95	getProducts	La fonction doit appeler l'API et executer les fonctions 'displayCarrousel' et 'displayCatalog' pour chaque objet récupéré de l'API avec en paramètre les valeurs de ces derniers	faire un console.log dans la fonction pour observer les valeurs retournées. (dans le 'then(data => {})' pour observer ce que renvoie la réponse du fetch, et dans la boucle 'for (let i=0;...)' pour observer ce que renvoient les données bouclées')	Une erreur pourrait survenir à l'appel de l'API (dû à un url invalide dans le fetch ou à un paramètre invalide dans les fonctions appelées)
article.js	17 à 48	displayIdProduct	La fonction doit créer en html les éléments qui vont être affichés sur la page de produit(conteneurs, image, nom, prix, description, couelurs) et leur donner la valeur passée en paramètre	Appeler la fonction avec l'objet 'testObject' et observer si les éléments se créent bien avec les données de 'testObject'	Les éléments pourraient ne pas se créer dû aux valeurs erronées passées en paramètres.
article.js	51 à 62	getIdProduct	La fonction doit appeler l'objet de l'API qui correspond à l'id renvoyé dans l'url et appeler la fonction 'displayIdProduct' avec ses données en paramètres	faire un console.log des données renvoyées dans la fonction pour vérifier ce qu'elle renvoie	L'url du fetch pourrait être erroné dû à l'id récupéré depuis l'url
article.js	65 à 81	addItem	La fonction créé un objet cart (si il n'existe pas déjà) et lui ajoute une donnée correspondant à l'id de l'article avec comme valeur 1 (qui correspond à la quantité) à condition que cette donnée n'existe pas déjà (auquel cas, la valeur de la donnée est incrémentée)	Ouvrir le local storage et verifier si la fonction s'execute bien au déclenchement des événements où elle est appelée	la variable 'id' qui est manipulée dans cette fonction pourrait être erronée dû à sa valeur récupérée dans l'url
cart.js	41 à 77	displayCartItems	La fonction doit créer les éléments html d'un article (conteneurs, image, nom, prix, quantité, prix total) avec les valeurs passées en paramètre et les afficher dans le panier	Appeler la fonction avec l'objet 'testObject' et une valeur numérique, et observer si les éléments se créent bien avec les données de 'testObject' et la valeur numérique	Les éléments pourraient ne pas se créer dû aux valeurs erronées passées en paramètres.
cart.js	82 à 97	getCart	Si l'objet 'cart' existe, la fonction va boucler les données de ce dernier et pour chaque donnée elle va appeler l'objet de l'API sélectionné et lancer la fonction 'displayCartItems' avec en paramètres les données reçues de l'API ainsi que sa valeur (quantité)	faire un console.log des données renvoyées dans la fonction pour vérifier ce qu'elle renvoie	l'objet 'cart' pourrait être vide auquel cas, la fonction renverrait 'error' dans la console
cart.js	100 à 105	CheckInputs	La fonction doit enlever l'attribut 'disabled' au bouton d'envoi de formulaire quand tout les inputs du checkout ont leur valeur 'accept' à 'true'	inclure un console.log de test dans la fonction pour vérifier si elle s'exécute bien quand les conditions sont remplies	La fonction renverra une erreur si une des variables testée n'existe plus (checkoutName, CheckoutMail,...)

Fichier JS	Lignes de code testées	Fonction testée	Résultat attendu	Comment vérifier le résultat attendu	Problème possible
cart.js	108 à 120	regexCheck	La fonction doit vérifier si l'input et le regex entrés en paramètres correspondent.	Lancer la fonction avec en paramètres un input(checkoutName par exemple) et une valeur (string 'test' par exemple) et observer si les instructions s'exécutent	La fonction renvoie une erreur si le premier paramètre entré n'est pas un input
validation.js	7 à 13	getIds	La fonction doit boucler les données de l'objet 'cart' du local storage et les renvoyer dans l'élément html <p> créé à l'appel de la fonction	faire un console.log des données renvoyées dans la fonction pour vérifier ce qu'elle renvoie	La fonction ne renvoie rien si l'objet 'cart' est vide
validation.js	16 à 20	emptyTheLocalStorage	La fonction doit vider le local storage au clic de l'élément passé en paramètres (btn)	Ouvrir le local storage et vérifier si la fonction s'exécute bien au déclenchement des événements où elle est appelée	La fonction renvoie une erreur si la paramètre passé n'est pas un élément html