

Fichier JS	Lignes de code testées	Fonction testée	Résultat attendu	Comment vérifier le résultat attendu	Problème possible
getData.js	1 à 9	getData	La fonction doit retourner une response de l'url si le status est ok	Utiliser différentes url d'api avec un <code>console.log(response.status)</code> pour vérifier la si la response n'est bien retourné que si elle obtient un statut "200".	Erreur dans le retour du status et donc dans l'affichage html de la page
productsList.js	1 à 26	bindProductsToView	La fonction affiche les produits de l'api dans la page html	Appeller la fonction avec en argument un array comprenant un ou plusieurs object avec les clés (id, name, price, imageUrl, description) et valeurs correspondantes	Les produits ne s'affichent pas ou partiellement dans la page html
productsList.js	28 à 41	creationProducts	La fonction insère les produits dans une balise html a	Créer une balise et la nommer, appeler ensuite la fonction avec 4 arguments : un nom, un nom de balise et la valeur de cette balise (p, span, div...) puis le nom de la balise créée initialement. On contrôle avec <code>console.log</code> que la balise contienne bien les arguments envoyés	Balise vide ou erreur dans la console
productsList.js	43 à 47	separateNumber	La fonction sépare les chiffres supérieur à 100 par une virgule avant les deux derniers chiffres	Appeller la fonction avec différentes valeurs de tests (valeurs inférieures et supérieures, vide), et on observe la valeur retournée, avec <code>console.log</code>	Retourne une valeur avec la virgule à la mauvaise position ou sans virgule
product.js	57 à 84	creationColorMenu	La fonction crée un menu déroulant avec les couleurs présente dans un array	Appeller la fonction avec un tableau de nom de couleur et utiliser un <code>console.log(colorSelect)</code> ; pour contrôler l'affichage de toutes les couleurs et leur valeurs	Un tableau avec un trop grand nombre de couleur pourrait ne pas être entièrement affiché dans le sélecteur
product.js	92 à 96	setAttributes	La fonction ajoute un attribut avec un nom et une valeur sur l'élément spécifié	Appeller la fonction avec la syntaxe : (element, {"test": "test"}) et contrôler avec un <code>console.log(element)</code> si l'attribut a bien été ajouté	L'attribut peut ne pas être ajouté si l'élément spécifié n'existe pas ou alors l'attribut peut ne pas fonctionner s'il n'existe pas dans la liste des attributs html
product.js	98 à 124	addToCart	La fonction ajoute le produit avec la couleur sélectionné dans le panier	Ajouter plusieurs produits en html dans le panier et contrôler dans le localStorage si l'on retrouve bien la clé "order" avec en object les produits transmis	Il pourrait y avoir un problème d'ajout de produit dans le localStorage ou une erreur avec le choix de la couleur qui ne fonctionnerait pas
cart.js	12 à 31	productAdditionInCart	La fonction ajoute en ligne dans un tableau html les produits présents (nom, image, couleur, prix et supprimer) dans le localStorage	Ajouter plusieurs produits dans la page html dans le panier avec toutes les couleurs existantes et contrôler le résultat dans le localStorage (objects dans un array)	Un trop grand nombre d'article dans le panier pourrait créer une erreur d'affichage

cart.js	59 à 69	totalPrice	La fonction retourne le total de l'addition du prix de chacun des produits contenu dans le localStorage	Ajouter des produits dans le panier et contrôler que le total corresponde bien à l'ensemble des prix additionné	Pourrait ne pas retourner l'intégralité du total ou un total erroné
cart.js	71 à 80	emptyCart	La fonction affiche une ligne "votre panier est vide" dans le tableau du panier et supprime le prix total dans le tableau	Ajouter un ou plusieurs produits dans le panier, ensuite appeler la fonction et contrôler que la ligne "votre panier est vide" s'affiche bien et que le prix total est supprimé	Pourrait ne pas afficher la ligne ou supprimer le prix total
cart.js	90 à 106	removeItem	La fonction supprime du panier et du localStorage la ligne dont le numéro est passé en paramètre et retourne un nouveau tableau dans le storage puis dans le panier	Appeler la fonction avec le numéro correspondant à la ligne que l'on souhaite supprimer	Pourrait ne pas supprimer la bonne ligne (en html ou storage) ou renvoyer un tableau incorrect dans le storage
cart.js	114 à 148	control	La fonction contrôle que les inputs du formulaire correspondent aux formats requis avec les regex	Insérer différentes valeur de test dans chacun des inputs (chiffres, lettres, chiffres et lettres, email incomplet, etc...) puis contrôler que le message Alert soit bien affiché si l'input ne correspond pas à ce que l'on attend comme résultat	Il pourrait être possible de valider le formulaire avec des inputs qui ne correspondent aux résultats attendus
cart.js	150 à 188	createData	La fonction crée un objet contact avec un tableau products ainsi que la date de la commande et ajoute ces 2 informations dans le localStorage	Ajouter des produits dans le panier, valider le formulaire puis contrôler le localStorage pour vérifier que les clés "order " et "dateOrder" sont bien ajoutés	Il pourrait y avoir un problème dans la création des clés dans le storage
cart.js	186 à 202	postData	La fonction envoie à l'api une requête post comprenant la commande "order" puis supprime la clé "addToCart" du storage et ensuite ouvre la page html "confirmation" avec le numéro order.id généré par l'api	Valider une commande avec un ou plusieurs produits et contrôler que la page confirmation.html s'ouvre bien avec les informations attendus (id généré par l'api + récapitulatif de la commande avec la date de commande) et que "addToCart" a été supprimé du storage	Erreur dans la récupération des données, la suppression de la clé du storage ou dans l'ouverture de confirmation.html