

Fichier JS	Lignes de code	Fonction testée	Résultat attendu	Comment vérifier le résultat attendu	Problème possible
script.js	5 à 10	userInputChecker	La fonction doit retourner 1 si le mot passé en paramètre est un mot (pas de chiffre, caractères accentués, symboles ...), et retourner 0 si ce n'en est pas un	Appeller la fonction avec différentes valeurs de tests (valeurs limites, valeur vide...), et on observe la valeur retournée, avec un console.log par exemple	La valeur retournée pourrait être erronée (0 au lieu de 1; et 1 au lieu de 0)
Api.js	7 à 12	getAllProducts	La fonction doit retourner une promesse contenant un tableau d'objets produits	Appeller la fonction et observée la valeur retournée	La promesse retournée pourrait ne pas être résolue
Api.js	19 à 25	getProduct	La fonction doit retourner une promesse contenant un objet produit	Appeller la fonction et observée la valeur retournée	La promesse retournée pourrait ne pas être résolue
Api.js	32 à 44	postOrder	La fonction doit retourner un objet contenant un objet contenant un objet contact, le tableau de produits commandés et le numéro de commande	Appeller la fonction et observée la valeur retournée	La promesse retournée pourrait ne pas être résolue
App.js	3 à 5	router.dispatch	La fonction doit exécuter les méthodes appropriées selon l'url de la page en cours	Appeler la fonction et vérifier si les fonctions correspondantes ont été exécutés selon l'url	L'url pourrai ne pas être reconnue ou les méthodes correspondantes ne pas être appelé
Router.js	12	constructor	Le constructeur doit récupérer le pathname de l'url	Appeler le constructeur et vérifier que le pathname a été récupéré	Le pathname pourrait ne pas être récupéré
Router.js	19 à 23	getRoute	La méthode renvoi la string du type de route selon le pathname	Appeler la méthode et vérifier le type de route retournée	La route pourrait ne pas être retournée

Router.js	29 à 51	dispatch	La méthode doit exécuter les fonctions correspondantes au type de route	Appeler la fonction et vérifier si les fonctions correspondantes ont été exécutés selon l'url	Les méthodes pourraient ne pas être exécutées
Storage.js	7 à 9	addItem	La méthode doit ajouter un item au local storage	Appeler la méthode et vérifier que l'item a bien été ajouté au local storage	L'item pourrait ne pas avoir été ajouté
Storage.js	11 à 13	removeItem	La méthode doit supprimer un item au local storage	Appeler la méthode et vérifier que l'item a bien été supprimé du local storage	L'item pourrait ne pas avoir été supprimé
Storage.js	15 à 17	syncStore	La méthode doit supprimer l'item actuel du storage et ajouter le nouveau	Appeler la méthode et vérifier que l'ancien item a été retiré et le nouveau ajouté	L'item pourrait ne pas avoir été supprimé ou ajouter
Cart.js	5 à 13	constructor	Le constructeur doit générer un tableau de produits depuis l'item du local storage ou créer un tableau vide	Appeler le constructeur et vérifier l'état du cart	Le cart pourrait ne pas avoir été créé
Cart.js	20 à 29	setQuantity	La méthode doit attribuer la quantité d'un produit au panier selon la quantité du produit passé en argument	Appeler la méthode et vérifier que la quantité de produit a été ajusté	La quantité pourrait ne pas avoir été ajustée correctement
Cart.js	36 à 39	remove	La méthode doit supprimer du panier le produit passé en argument	Appeler la méthode et vérifier que le produit a bien été supprimé	Le produit correspondant pourrait ne pas avoir été supprimé

Cart.js	42 à 44	clearCart	La méthode doit vider le tableau de produits du panier et synchroniser le store	Appeler la méthode et vérifier l'état du store et du panier	Le panier ou le store pourraient ne pas être synchronisés
Cart.js	52 à 55	getIndex	La méthode doit retourner l'index du produit passé en argument	Appeler la méthode et vérifier la valeur de l'index retourné	La valeur de l'index pourrait être erronée
Cart.js	65 à 67	isInCart	La méthode doit renvoyer true si le produit passé en argument est dans le panier, ou renvoyer false sinon	Appeler la méthode et vérifier la valeur retournée	La valeur retournée pourrait ne pas être la bonne
Cart.js	73 à 78	getTotalPrice	La méthode doit retourner le montant total de tous les produits actuellement dans le panier	Appeler la méthode et vérifier la valeur retournée	La valeur retournée pourrait avoir été mal calculée
Cart.js	84 à 88	getAllId	La méthode doit renvoyer la liste de tous les id présents actuellement au panier	Appeler la méthode et vérifier la liste des Id	La liste Id pourrait être erronée
Checkout	9 à 11	renderCheckout	La fonction doit appeler les fonctions handleUserInput et hendleOrder	Appeler la fonction et vérifier si les fonctions correspondantes ont été exécutées	Les fonctions pourraient ne pas être exécutées
Checkout	18 à 19	getFirstName	La fonction doit retourner le first name de l'utilisateur	Appeler la fonction et vérifier la valeur retournée	La valeur retournée pourrait être incorrecte

Checkout	26 à 27	getLastName	La fonction doit retourner le last name de l'utilisateur	Appeler la fonction et vérifier la valeur retournée	La valeur retournée pourrait être incorrecte
Checkout	34 à 35	getAddress	La fonction doit retourner l'adresse de l'utilisateur	Appeler la fonction et vérifier la valeur retournée	La valeur retournée pourrait être incorrecte
Checkout	42 à 43	getCity	La fonction doit retourner la ville de l'utilisateur	Appeler la fonction et vérifier la valeur retournée	La valeur retournée pourrait être incorrecte
Checkout	50 à 51	getEmail	La fonction doit retourner l'email de l'utilisateur	Appeler la fonction et vérifier la valeur retournée	La valeur retournée pourrait être incorrecte
Checkout	59 à 60	displayMessageError	La fonction doit faire apparaître le message d'erreur	Appeler la fonction et vérifier l'insertion du message	Le message pourrait ne pas être inséré
Checkout	63 à 64	removeErrorMessage	La fonction doit retirer le message d'erreur	Appeler la fonction et vérifier que le message a bien été supprimé	Le message pourrait ne pas être supprimé
Checkout	71 à 88	isFirstNameValid	La fonction doit vérifier si la saisie du first name est valide	Appeler la fonction et vérifier si elle renvoi bien true	La fonction pourrait ne pas renvoyer true

Checkout	95 à 113	isLastNameValid	La fonction doit vérifier si la saisie du last name est valide	Appeler la fonction et vérifier si elle renvoi bien true	La fonction pourrait ne pas renvoyer true
Checkout	110 à 132	isAddressValid	La fonction doit vérifier si la saisie de l'adresse est valide	Appeler la fonction et vérifier si elle renvoi bien true	La fonction pourrait ne pas renvoyer true
Checkout	139 à 152	isCityValid	La fonction doit vérifier si la saisie de la ville est valide	Appeler la fonction et vérifier si elle renvoi bien true	La fonction pourrait ne pas renvoyer true
Checkout	163 à 175	isEmailValid	La fonction doit vérifier si la saisie de l'email est valide	Appeler la fonction et vérifier si elle renvoi bien true	La fonction pourrait ne pas renvoyer true
Checkout	182 à 194	isFormValid	La fonction doit vérifier si la saisie du formulaire est valide	Appeler la fonction et vérifier si elle renvoi bien true	La fonction pourrait ne pas renvoyer true
Confirmation	5 à 13	renderConfirmation	La fonction doit afficher le numéro de la commande sur la page	Appeler la fonction et vérifier si le numéro de la commande est bien affichée	Le numéro de la commande pourrait ne pas s'afficher
Product	10 à 15	renderProduct	La fonction doit exécuter les fonctions renderProductUI et handleAddToCart	Appeler la fonction et vérifier si les fonctions correspondantes ont été exécutées	Les fonctions pourraient ne pas être exécutées

Product	21 à 32	handleAddToCart	La fonction doit ajouter un produit au panier lorsque l'on appuie sur le bouton correspondant	Appeler la fonction et vérifier si le produit a bien été ajouté au panier	Le bouton pourrait ne pas avoir été ajouté au panier
Product	39 à 49	createProduct	La fonction doit créer un produit	Appeler la fonction et vérifier si le produit a été créé	Le produit pourrait ne pas être créé
Product	56 à 61	getColorOption	La fonction doit récupérer l'URL de l'image du produit	Appeler la fonction et vérifier que l'image a bien été récupérée	L'URL pourrait ne pas être récupéré
Product	69 à 71	getImage	La fonction doit récupérer l'image du produit	Appeler la fonction et vérifier que l'élément a bien été récupérée	L'élément pourrait ne pas être récupéré
Product	78 à 80	getName	La fonction doit récupérer le nom du produit	Appeler la fonction et vérifier que l'élément a bien été récupérée	L'élément pourrait ne pas être récupéré
Product	87 à 89	getQuantity	La fonction doit récupérer la quantité du produit	Appeler la fonction et vérifier que l'élément a bien été récupérée	L'élément pourrait ne pas être récupéré
Product	97 à 99	getPrice	La fonction doit récupérer le prix du produit	Appeler la fonction et vérifier que l'élément a bien été récupérée	L'élément pourrait ne pas être récupéré

Product	106 à 114	getId	La fonction doit récupérer l'id du produit	Appeler la fonction et vérifier que l'élément a bien été récupérée	L'élément pourrait ne pas être récupéré
Cart-Item	7 à 38	createCartItemUI	La fonction doit retourner un élément HTML de cart item	Appeler la fonction et vérifier que l'élément HTML a été retourné	L'élément pourrait ne pas être créé
Product-Card	7 à 19	createProductCardUI	La fonction doit retourner un élément HTML de product item	Appeler la fonction et vérifier que l'élément HTML a été retourné	L'élément pourrait ne pas être créé
Cart	12 à 24	renderCart	La fonction doit retourner un élément HTML de cart item	Appeler la fonction et vérifier que l'élément HTML a été retourné	L'élément pourrait ne pas être créé
Cart	32 à 45	handleQuantityChange	La fonction doit modifier la quantité du produit lorsque l'on clique sur le bouton correspondant	Appeler la fonction et vérifier que la quantité du produit a bien été modifié	La quantité pourrait ne pas avoir été ajustée correctement
Cart	50 à 70	handleDelete	La fonction doit supprimer le produit du panier	Appeler la fonction et vérifier que le produit a bien été supprimé	Le produit pourrait ne pas être supprimé
Cart	77 à 85	getProductFeatures	La fonction retourner les informations du produit en cours	Appeler la fonction et vérifier si les informations ont bien été retournées	Les informations pourraient ne pas être retournées

Product	6 à 26	renderProductUI	La fonction doit retourner un élément HTML de product UI	Appeler la fonction et vérifier que l'élément a bien été récupérée	L'élément pourrait ne pas être créé
Product	34 à 38	createProductImage	La fonction doit retourner un élément HTML de img	Appeler la fonction et vérifier que l'élément HTML a été retourné	L'élément pourrait ne pas être créé
Product	46 à 51	createProductOption	La fonction doit retourner un élément HTML de option	Appeler la fonction et vérifier que l'élément HTML a été retourné	L'élément pourrait ne pas être créé
Product	7 à 15	renderProductsGallery	La fonction doit retourner un élément HTML de gallery items	Appeler la fonction et vérifier que l'élément HTML a été retourné	L'élément pourrait ne pas être créé