

Fichier JS	Lignes de code testées	Fonction testée/objectif	Résultat attendu	Comment vérifier le résultat attendu	Problème possible
script.js	4 à 18	fetch de l'api http://localhost:3000/api/products	1) Accéder à l'api et appel de la fonction lesKanaps() 2) info en H1 et console.log si erreur	1) Mettre le lien dans la barre du navigateur. 2) Réseau(network) -> products (code état)-> Aperçu (ce qui est reçu)	1) serveur inaccessible. 2) changement d'adresse de l'api
script.js	22 à 38	lesKanaps()	1) affichage dynamique des produits de l'api et de certains éléments de produits 2) changement de page au clic sur un produit	visuellement (c'est un affichage)	si les éléments de chaque produit changent ou la valeur d'un élément de produit est erroné/absent

product.js	5 à 7	avoir la valeur de l'id du produit	l'id du produit	1) console.log ligne 8 2) id produit dans l'url	ne pas obtenir de l'api un id produit
product.js	12 à 21	fetch de l'api http://localhost:3000/api/products	1) Accéder à l'api et appel de la fonction lesproduits 2) info en H1 et console.log si erreur	1) Mettre le lien dans la barre du navigateur. 2) Réseau(network) -> products (code état)-> Aperçu (ce qui est reçu) 3) faire un console.log de l'appel	1) serveur inaccessible. 2) changement d'adresse de l'api
product.js	26 à 28	déclaration de articleClient et insertion de l'id produit	créer l'objet javascript du produit et y inclure son id	console.log(articleClient)	pas d'id ou id erroné
product.js	32 à 58	lesProduits()	affichage dynamique du produit et de certains de ses éléments	1) visuellement (c'est un affichage) 2) console.log ligne 57 que l'action a été effectuée	si les éléments de chaque produit changent ou la valeur d'un élément de produit est erroné/absent
product.js	63 à 75	écoute événement type couleur	avoir une valeur d'écoute à l'interaction et reset l'affichage du bouton de choix	1)console.log ligne 74 2) si le bouton change suite au choix d'un produit dès l'action sur l'événement	Si le client, suite à l'action de choix via le bouton, re-clique dessus, l'affichage du bouton n'aura pas été reset dans sa boucle de choix produit
product.js	80 à 92	écoute événement type quantité	avoir une valeur d'écoute à l'interaction et reset l'affichage du bouton de choix	1)console.log ligne 91 2) si le bouton change suite au choix d'un produit dès l'action sur l'événement	Si le client, suite à l'action de choix via le bouton, re-clique dessus, l'affichage du bouton n'aura pas été reset dans sa boucle de choix produit
product.js	97 à 120	conditions de validation du clic via le bouton ajouter au panier	le choix par le clic ne peut être effectué que si les champs événements sont valides	1) une alerte est déclenchée via le clic du bouton si le résultat est non valide 2) le bouton change visuellement si le choix est valide 3) console.log ligne 115 si le clic a été validé	Si l'api a délivré des informations erronées dès le départ sur le produit
product.js	135 à 145	ajoutPremierProduit()	s'il n'y a pas de panier, un panier est créé avec le premier produit	1) console.log ligne 141 de ce qui est envoyé au localStorage 2) regarder le localStorage avoir un nouveau tableau	Si l'api a délivré des informations erronées dès le départ sur le produit
product.js	149 à 171	ajoutAutreProduit()	ajoute l'article dans le tableau non vierge et fait un tri selon l'id et si plusieurs produits ont la même id, un tri sera fait sur la couleur	regarder le résultat dans le localStorage	Si l'api a délivré des informations erronées dès le départ sur le produit
product.js	175 à 199	Panier()	ajuste la quantité si le produit est déjà dans le tableau, fait appel aux fonctions ajoutPremierProduit() et ajoutAutreProduit() pour un résultat combinatoire	regarder le résultat dans le localStorage	Si l'api a délivré des informations erronées dès le départ sur le produit

Fichier JS	Lignes de code testées	Fonction testée/objectif	Résultat attendu	Comment vérifier le résultat attendu	Problème possible
cart.js	2	différencier la page confirmation et panier ayant le même script	ne jamais avoir d'erreur en console log sur un pointage ou un élément non déclaré	la console reste vierge des erreurs stipulées sur la page panier et après le passage sur page confirmation	Y penser si on rajoute du code et des fonctionnalités
cart.js	7 à 21	fetch de l'api http://localhost:3000/api/products	1) Accéder à l'api et appel de la fonction affichagePanier() 2) info en H1 et console.log si erreur	1) Mettre le lien dans la barre du navigateur. 2) Réseau(network) -> products (code état)-> Aperçu (ce qui est reçu) 3) faire un console.log de l'appel	1) serveur inaccessible. 2) changement d'adresse de l'api
cart.js	25 à 55	affichagePanier()	1) détermine les conditions d'affichage des produits du panier combiné aux fonctions affiche(), modifQuantité() et suppression() 2) info en H1 s'il n'y a pas de panier	1) visuellement (c'est un affichage) 2) si il n'y a pas de panier il n'y a pas d'affichage de produit	
cart.js	59 à 88	affiche()	Fonction d'affichage d'un panier (tableau) associée à la fonction totalProduit()	1) visuellement (c'est un affichage) 2) si il n'y a pas de panier il n'y a pas d'affichage	Si l'api a délivré des informations erronées dès le départ sur le produit
cart.js	92 à 114	modifQuantité()	on modifie dynamiquement les quantités du panier via une valeur d'écoute associé à la fonction totalProduit()	regarder le résultat de l'écoute et la modification dans le localStorage	Le cas échéant, la logique serait remise en cause
cart.js	118 à 154	suppression()	suppression on supprime un article dynamiquement du panier et donc de l'affichage via une valeur d'écoute, associé à la fonction totalProduit() (non obligatoire)	1) visuellement (c'est un affichage) 2) regarder le résultat de la modification dans le localStorage	Le cas échéant, la logique serait remise en cause
cart.js	158 à 176	totalProduit()	fonction ajout nombre total produit et coût total	visuellement (c'est un affichage)	Si l'api a délivré des informations erronées dès le départ sur le produit
cart.js	181 à 207	pointage sur la page panier et attribution de classe	préparation des éléments des fonctions	code source sur l'inspecteur	Le cas échéant, le pointage serait remis en cause ou l'attribution des classes
cart.js	212 à 216	définition de regex	délimiter le cadre d'utilisation du champ par l'utilisateur	si le champ est validé par la logique de la regex	Le cas échéant, la logique regex serait remise en cause
cart.js	220 à 247	Ecoute et attribution de point(pour sécurité du clic) si les champs nom prénom et ville sont ok d'après la regex	obtention de point si les 3 champs sont ok, associé aux fonctions couleurRegex() et valideClic()	la valeur du champ s'ajoute dans le local storage tant que le champ est ok et quand les 3 champs sont ok une valeur 3 est attribué dans le local storage	Le cas échéant, la logique regex serait remise en cause

Fichier JS	Lignes de code testées	Fonction testée/objectif	Résultat attendu	Comment vérifier le résultat attendu	Problème possible
cart.js	251 à 253	utilisation de la fonction texteInfo() pour les champs prenom nom et ville	afficher le texte correspondant sous ces champs	visuellement (c'est un affichage)	Le cas échéant, la logique regex serait remise en cause
cart.js	257 à 276	Ecoute et attribution de point(pour sécurité du clic) si le champ adresse est ok d'après la regex	obtention de point si le champ est ok, associé aux fonctions couleurRegex() et valideClic()	la valeur du champ s'ajoute dans le local storage tant que le champ est ok une valeur 1 est attribué dans le local storage	Le cas échéant, la logique regex serait remise en cause
cart.js	280	utilisation de la fonction texteInfo() pour le champ adresse	afficher le texte correspondant sous ce champ	visuellement (c'est un affichage)	Le cas échéant, la logique regex serait remise en cause
cart.js	284 à 303	Ecoute et attribution de point(pour sécurité du clic) si le champ email est ok d'après les regex	obtention de point si le champ est ok, associé aux fonctions couleurRegex() et valideClic()	la valeur du champ s'ajoute dans le local storage tant que le champ est ok une valeur 1 est attribué dans le local storage	Le cas échéant, les logiques regex seraient remises en cause
cart.js	307 à 330	Ecoute et attribution d'une valeur de texte en fonction du champ	afficher le texte correspondant sous ce champ	visuellement (c'est un affichage)	Le cas échéant, les logiques regex seraient remises en cause
carts.js	335 à 351	couleurRegex()	fonction couleurRegex qui modifira la couleur de l'input par remplissage tapé, aide visuelle et accessibilité	visuellement (c'est un affichage)	Le cas échéant, la logique regex serait remise en cause
cart.js	355 à 376	texteInfo()	fonction d'affichage individuel des paragraphes sous input sauf pour l'input email	visuellement (c'est un affichage)	Le cas échéant, la logique regex serait remise en cause
cart.js	380 à 393	valideClic()	Fonction de validation/d'accès au clic du bouton du formulaire ; potentiellement premier et dernier maillon de la chaîne d'envoi	1) au clic, si on est envoyé sur la page confirmation 2) si dans le local storage, la somme des points attribués pour les champs est égale à 5 le bouton sera potentiellement déverrouillé 3) si on essaye de cliquer alors que les conditions ne sont pas remplies le texte du bouton change et dès que les conditions sont ok, le texte du bouton rechange	Le cas échéant, les logiques regex seraient remises en cause
cart.js	397 à 404	Ecoute le clic sur le bouton d'envoi	on ecoute le clic associé aux fonctions valideClic() et envoiPaquet();	on se retrouve sur la page confirmation avec un numéro de commande ou bien on a l'information que quelque chose n'est pas ok pour l'envoi	Le cas échéant, les logiques regex seraient remises en cause

Fichier JS	Lignes de code testées	Fonction testée/objectif	Résultat attendu	Comment vérifier le résultat attendu	Problème possible
cart.js	409 à 422	tableauId()	fonction récupérations des id puis mis dans un tableau	si on clique sur le bouton d'envoi sans avoir rempli le formulaire, un console log de l'objet contact apparaît, et products est le résultat de l'utilisation de ce tableau et de cette fonction	Si l'api a délivré des informations erronées dès le départ sur le produit
cart.js	426 à 441	paquet()	récupération des données client et panier avant transformation en objet d'envoi	si on clique sur le bouton d'envoi sans avoir rempli le formulaire, un console log de l'objet contact apparaît	1) Si l'api a délivré des informations erronées dès le départ sur le produit 2) les logiques regex pourraient être remises en cause
cart.js	445 à 472	envoiPaquet()	1) on prépare l'envoi en appelant les fonctions tableauId() et paquet() pour le contenu et on vérifie que le bouton ai les points pour pouvoir être cliqué puis on envoit à l'api 2) on a l'information que quelque chose n'est pas ok pour l'envoi via une alerte erreur et un console log de l'erreur	on se retrouve sur la page confirmation avec un numéro de commande	1) Si l'api a délivré des informations erronées dès le départ sur le produit et possibilité de se retrouver sur la page confirmation avec un numero commande undefined si les données envoyées ne sont pas conforme à ce qui est attendu 2) les logiques regex pourraient être remises en cause 3) l'api ne répond pas
cart.js	476 à 490	Commande()	Fonction d'affichage auto invoquée du numéro de commande + vide du storage lorsque l'on est sur la page confirmation	on se retrouve sur la page confirmation avec un numéro de commande et un remerciement	Si l'api ne répond pas
FIN					