

Tests Unitaires Projet 5 Orinoco				
Lignes et fonctions testées, Branche Github : « main » Fichier : Frontend/Orinoco.js	Tests		Résultats	
	LIGNES À ÉCRIRE	EXPLICATIONS	ATTENDUS	INATTENDUS/TESTÉS
2, ceci est une condition	console.log((document.getElementById("mouv")))	on vérifie qu'on détecte bien cette partie du « DOM » et donc qu'on est bien sur la bonne page HTML	la balise contenant l'id « mouv » apparaît dans la console, la suite du code s'exécute	la suite du code ne s'exécute pas, le contenu de n'affiche pas si on se trouve sur une page différente et si l'id mouv n'est pas détecté, la console ne nous affiche pas la balise.Si le serveur ne répond pas la console nous renvoie un code d'erreur
3, fonction « fetch » on fait un appel de type « GET » à l'API pour afficher tous les produits	fetch GET/ then/ catch	on vérifie que la requête pour récupérer les données s'est bien passée, si ce n'est pas le cas un message personnalisé s'affiche dans la console	les données sont traitées les fonctions peuvent s'enchaîner	la fonction catch récupère l'erreur en callback et déclenche un message d'erreur.Une erreur s'affiche dans la console si l'adresse n'est pas la bonne,ainsi que si le serveur ne répond pas
28, la fonction getArticleId retourne l'id placée dans l'url de la page	console.log(getArticleId())	on vérifie que la fonction renvoie bien le résultat attendu (l'id du produit) dans la console	renvoie l'id des produits, qui seront ensuite récupérée depuis l'url	n'affiche rien dans la console
35 ,« fetch », appel à l'API pour afficher un produit précis	voir ligne 2	voir ligne 2(même principe)	voir ligne 2	voir ligne 2
57, ceci est une boucle qui a pour but d'afficher les données de « lenses » dynamiquement pour chaque appareils photos	console.log(lense)	on va vérifie qu'on reçoit bien les données,on attend la une liste avec spécifications des lentilles	la liste apparaît dans la console	on reçoit une erreur dans le bloc, et donc dans la console, la fonction catch prend le relais
74,la fonction isclicked, selectionne un produit l'ajoute au localStorage avec une valeur de 1, si l'item est déjà présent il y a iteration de la valeur présente dans « value »(localStorage)	if (isclicked()){console.log('fonction choisir valide')}else{console.log('fonction choisir non valide')}	on vérifie que tout se passe bien au niveau du choix d'article, il faut cliquer sur « selectionner » de la page produit pour déclencher l'action	« fonction choisir valide » dans la console, si l'utilisateur clique plusieurs fois la fonction se déclenche à nouveau et exécute ses conditions	« fonction choisir » non valide dans la console
95,fonction deleteLastitem	//	cette fonction va supprimer la dernière clé dans le localStorage, si le localStorage est vide alors la clé « prix » contenu dans le session storage est aussi supprimée	si fonctionnelle, il suffit de vérifier le localStorage avec l'inspecteur dans le navigateur web, si le localStorage est vide, il va supprimer la clé « prix » dans le session storage	la dernière clé n'est pas effacée , si la clé prix n'est pas présente à ce moment la , l'action se déclenche tout de même
113,la fonction allstorage	console.log(Object.keys(localStorage)[i])	cette fonction contient une boucle qui va parcourir le localStorage et faire un fetch à l'api pour chacune des « id » placées dans ce même localStorage	les éléments sélectionnés par le client doivent apparaître ainsi que leur quantités	le serveur pourrait ne pas répondre, .catch devrait prendre le relais et envoyer un message d'erreur,si la clé renvoyé est incorrecte les produits n'apparaîtront pas , si la réponse envoyé est incorrecte, les produits n'apparaîtront pas non plus
145, fonction isValidated	test visuel	Regex de validation à terme générique avec espace, sans caractère spéciaux, valable pour le champs prénom, nom et ville,controle l'entrée de l'utilisateur	si fonctionnelle on une message de validation s'affiche en vert	un message de non validation s'affiche en rouge
156,168	test visuel	Regex de validation spécifique au adresses et aux mails	si fonctionnelle on obtient un message de validation qui s'affiche en vert	un message de non validation s'affiche en rouge, et l'entrée utilisateur est effacée dans le champs, si la fonction ne fonctionne pas une erreur s'affichera en console puisque le reste du code ne pourra s'exécuter
185,195,206,218,231, conditions	console.log(****.value) *=constante correspondante, exemple: « console.log(Prenom.value) »	ces fonctions vont écouter les champs à remplir par l'utilisateur et les soumettre a un test de validation	les entrées utilisateurs sont affichées dans la console	les entrées utilisateurs ne s'affichent pas dans la console
251, la constante « isEmpty » qui est en fait une fonction	if (isEmpty){console.log(' l'objet contact est vide')}	cette fonction verifie que chaque attribut de l'objet contact est égal à 0 ou vide	« l'objet contact est vide » s'affiche dans la console	rien ne s'affiche en console
257, fonction « Forms »	fetch POST/then/catch	on envoie des données à l'API	on reçoit une réponse qui contient plusieurs éléments dont l'id de commande	la fonction catch renvoie : « erreur lors de l'envoi de la commande » dans la console
291,301, fonctions Displayorder, Displayprice	test visuel	on envoie des informations dans le session storage et on affiche des éléments sur la page	on peut vérifier que les données sont bien dans le sessionstorage et que les éléments s'affichent correctement	rien ne s'affiche, si les données envoyés sont incorrectes, la fonction ,catch prend le relais et affiche une erreur en console(erreur lors de l'envoi de la commande.L'erreur BAD request survient également de la part du serveur.