

PLAN DE
TEST
ORINOCO

Fichier JS	Ligne de code testées	Fonction testée	Résultat attendu	Comment vérifier le résultat attendu	Problème possible
product.js	16 -> 61	showTeddies	La fonction doit récupérer les données présente dans l'API, créer de nouveaux éléments html, les ajouter à la DOM et afficher les données le tout autant de fois que d'objet présent dans l'api, et lors du clique sur un des produits ouvrir la page web product.html avec l'id du produit dans l'url	Lancer le serveur, et ouvrir le fichier : index.html situé dans le dossier html et vérifier que tout les produits soient bien affichés et que tout les éléments html créé avec le JS apparaissent bien sur la page	Aucun produit n'est affiché si un problème survient au niveau de l'api (API non lancée ou autre)
product.js	66 -> 114	selectProduct	la fonction doit récupérer l'id qui est dans l'url de la page product.html, et suivant celui-ci affiche toutes les données du produit lié à cet ID	Vérifier que l'url correspond bien à ../html/product.html?id=\${[article.dataset.id]}, et que le produit, affiché correspond bien à l'ID de l'url	Aucun produit n'est affiché si un problème survient au niveau de l'api, ou si l'id dans l'URL ne correspond à aucun des ID des produits dans l'API
cart.js	4 -> 157	addToCart	La fonction doit créer un évènement lors du clique sur le bouton "ajouter au panier" qui permet d'exécuter les fonctions cartNumbers et totalCost. Ajoute également une variable "quantity = 0" à l'objet récupéré de l'API.	La variable "quantity" n'apparaît pas dans le console.log(data), et lors du clique sur le bouton "ajouter au panier" rien ne se passe.	Le produit n'est pas ajouté au panier, donc impossible de valider la commande du client.
cart.js	23 -> 28	onLoadCartNumbers	La fonction doit modifier l'affiche de la balise span qui permet de voir la quantité du panier par rapport à la valeur de la KEY "cartNumbers" dans le localStorage	Vérifier que l'affichage au niveau de la balise header à bien changé et qu'il correspond bien à la valeur de "cartNumber"	L'affiche de la quantité de produit dans le panier n'est pas modifiée. Elle reste donc à son nombre initial au lieu de faire un +1.
cart.js	30 -> 45	cartNumbers	La fonction doit récupérer la clé "cartNumbers" du localStorage sous le nom de variable "productNumbers", puis elle vérifie si elle existe ou non, si c'est le cas elle ajoute 1 à la valeur de cartNumbers, sinon elle crée la clé "cartNumbers" et met la valeur à "1" dans le localStorage. Et affiche un message de confirmation qui confirme l'ajout du produit dans le panier.	Vérifier dans le dev tools -> application, que la clé/valeur cartNumbers à bien été créée, et le message de confirmation apparaît bien	Si la clé cartNumbers n'existe pas, celle-ci n'est pas créée, ou n'est pas modifiée si elle existe déjà.
cart.js	47 -> 66	setItems	la fonction doit ajouter les données du produit afficher au localStorage et modifier la valeur quantité dans l'objet.	Vérifier dans le dev tools -> application, que la clé productInCarts contient bien les données du produit, et que la variable quantity à bien été modifiée	Aucun produit n'est ajouté dans le localStorage.
cart.js	68 -> 81	totalCost	la fonction doit ajouter le prix des produits dans le localStorage dans la clé "totalCost" et de l'additionner afin d'avoir le prix total.	Vérifier dans le dev tools -> application, que le clé totalCost contient bien le prix global (exemple : le client choisit d'acheter 2 Teddies (Norbert) à 29€, la valeur de totalCost sera donc de 58.	Soit la clé totalCost n'existe pas, soit sa valeur est NULL.
cart.js	83 -> 153	displayCart	la fonction doit récupérer les données présente dans le localStorage et les afficher sur la page cart.html.	Vérifier sur la page cart.html que les produits ajoutés dans le panier s'affichent bien avec le prix unitaire, la quantité, et le prix total.	Aucun produit n'est affiché sur la page cart.html