

	Functionalité	Action	Résultat attendu	Résultat observé
script.js	Une page d'accueil montrant (de manière dynamique) tous les articles disponibles à la vente.	Ouvrir sur la page d'accueil du site web dans un navigateur	Affichage de l'ensemble des produits	OK / Description erreur
	Fonction Get avec Fetch	- Cette fonction permet de récupérer les objets de l'api de façon async (en utilisant une promesse) - On traite la réponse avec .json	Si la promesse retourne (réponse positive) on fait une boucle for et l'on récupère le tableau / Si non on récupère une erreur.	Array(8) s'affiche ds la console / si non Error
	appel de la fonction d'affichage des produits	on utilise la propriété (display)	affichage des produits sur la page d'accueil	Les produits sont affichés
product.js	Une page produit permettant de récupérer un produit de l'api et de l'afficher.	Afficher le produit avec ses données, ajouter des fonctions qui vont permettre d'ajouter à l'objet l'id, les quantités et la couleur.	récupération et affichage d'un seul objet avec ses données. Possibilité de choisir une couleur / Possibilité de définir une quantité et de valider	Après avoir choisi l'objet sur la page d'accueil, l'objet apparaît dans la page produit, avec ses données. 3 actions: choix de la couleur / sélection de la quantité / Validation
	Fonction Get avec la méthode Fetch, utilisation de (window, location, href) et de searchParams.	Récupération de l'ID dans l'URL. Faire des requêtes vers l'api pour récupérer les données image, couleur, quantités. Créer des messages d'instruction pour prévenir l'utilisateur.	On doit récupérer l'id, la couleur la quantité / on doit pouvoir sélectionner une quantité et choisir une couleur. Des messages doivent apparaître si les actions ne sont pas exécutées correctement.	Récupération de l'id dans la console "le id". Les données s'affichent et l'on peut sélectionner la quantité et choisir la couleur. Le message "choisissez une couleur" apparaît en cas d'oubli idem pour la qt.
	Fonction saveCart avec la méthode setItem	Stocker les données converties en chaîne de caractères dans localStorage. Le localStorage n'accepte que les chaînes de caractères	Cette fonction va permettre de sauvegarder les données dans le local Storage	les données sont stockées sous forme de chaînes de caractères.
	Fonction getCart avec la méthode getItem	Création du panier ds le local Storage.	La Key doit être égale à Cart.	Key = Cart
	Fonction add Cart	Envoyer sur le localStorage, les éléments sélectionnés ds une variable	On doit récupérer l'id et la couleur mais également la quantité sous forme de nombre.	{id: "107fb5b75607497b96722bda5b504926", quantity: 20, color: "Blue"}
cart.js	Création de la page Panier	Ajouter des produits ds le panier / Gérer la modification et la suppression de produits. Valider le formulaire pour passer la commande.	Afficher des produits dans le panier, sélectionnés sur la page produit. L'utilisateur doit pouvoir modifier les quantités, remplir le formulaire pour pouvoir valider la commande. Vérifier si le prix et la qt total. Des alertes signalent les erreurs de saisies.	Produits affichés / Modifications possibles / Formulaire adapté / Calcul (prix, qt) réalisés / Alertes mises en place.
	Fonction getFromBack avec la méthode Fetch	Retourner les données depuis le BackEnd	La fonction nous permet de récupérer l'ensemble des objets	Récupération de l'ensemble des objets sur la console.
	Fonction showProducts avec une boucle for	Afficher l'ensemble des objets, (image, price)(id, color, qt). Les positionner ds l'interface DOM.	L'ensemble des objets doit s'afficher et se positionner.	Les produits sont affichés et placés correctement.
	Function eraseQt	Supprimer un produit / Sélection de l'ID qui sera supprimé en cliquant sur supprimer / Récupérer l'id et la couleur / Filtrer le produit à supprimer ds le localStorage: Effacer l'article ds le DOM: Signaler la suppression de l'objet / Récupérer la zone de saisie	Produit supprimé	Produit supprimé
	Fonction ChangeQuantity précédé d'un input.quantity	Récupérer les valeurs ds l'inputItemQuantity, pour pouvoir changer la qt.	Qt récupérée de la page produit.	Qt récupérée de la page produit.
	Fonction ChangeQuantity suivie d'une alerte	Changement de quantité création d'une alerte, Qt modifiée.	quantité modifiée dans le panier	Quantité modifiée ds le panier
	Fonction getTotalProduct	Après avoir changé la Qt d'un produit la qt total doit se calculer ainsi que le px total.	Cette fonction après modification de la Qt doit recalculer le px total et la qt total	Changement de px réalisés: Calcul du pxTotal et de la qtTotal réalisée.
	Fonction addSurfaceControle	Ajout d'un objet global "formValidation" qui contiendra la validité ou non des zones de surface. L'objet est parcouru au click sur commander. Envoyer la commande vers le serveur avec la méthode Post de l'objet newOrder	L'utilisateur doit pouvoir remplir son formulaire et passer la commande	L'utilisateur remplit le formulaire et click sur commander.
	Fonction Main	Centraliser les fonctions dans une fonction principale	Le code est mieux structuré	code bien structuré ce qui donne de la visibilité.
confirmation.js	confirmation de la commande	On récupère l'id passé en paramètre,	L'utilisateur doit recevoir un message, "commande validée avec son numéro de commande"	commande validée 57a8fc00-d44f-11ec-9301-cf355fea608d