

Fichier js	Lignes de code	Fonctionnalité	Action	Résultat attendu	Résultat observé
<b>1</b> 1_helpers.js 2_index.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 - Affichage des produits de manière dynamique Err - L'API pourrait ne pas répondre et dans ce cas affichage d'un message d'erreur sur la page d'accueil "Désolés, les produits demandés sont actuellement introuvables !"
2_index.js	5 à 17	Fetch => GET	Requêter l'API pour demander l'ensemble des produits	réponse au format JSON regroupant tous les produits de l'API	OK - on récupère une réponse au format JSON. Err - Affichage d'un message d'erreur sur la page d'accueil "Désolés, les produits demandés sont actuellement introuvables !"
1_helpers.js	4 à 44	class Products	Récupérer et parcourir la réponse émise pour insérer chaque éléments dans le DOM	Pour chaque produit, l'image de celui-ci, ainsi que son nom et le début de sa description doit être affiché	OK - affichage des produits de manière dynamique Err - les éléments de chaque produit pourraient être mal configurés (ex. oubli de la balise altTxt)
<b>2</b> 1_helpers.js		En cliquant sur le produit, l'utilisateur sera redirigé sur la page du produit pour consulter celui-ci plus en détails	Clic sur un des produits => Faire le lien entre un produit de la page d'accueil et la page Produit	Possibilité d'ouvrir une page Produit en sachant quel produit afficher	OK - ouverture de la page produit du produit sélectionné Err - l'ID du produit pourrait être mal configurée et la page produit ne s'affichera pas
1_helpers.js	11	class Products	Paramétrer chaque produits de la page d'accueil	balise <a href> paramétrée avec l'id du produit sélectionné	
<b>3</b> 1_helpers.js 2_product.js		Une page produit qui représente un seul produit	Affichage des détails du produit sélectionné	Une page Produit complétée à partir des données de l'API	OK - Affichage des produits de manière dynamique Err - L'API pourrait ne pas répondre et dans ce cas affichage d'un message d'erreur sur la page produit
3_products.js	11 à 14	getIdParams()	-Utiliser URLSearchParams pour récupérer l'id du produit à afficher	récupérer le paramètre "_Id" de l'URL pour requêter le produit sélectionné dans l'API	
3_products.js	21 à 215	Fetch => GET	-Interroger l'API pour récupérer les détails du produit et les insérer au DOM	Les détails du produits sélectionné s'affichent ainsi que des options de personnalisation	
<b>4</b> 3_products.js		un menu déroulant permet à l'utilisateur de choisir une option de personnalisation	sélectionner la couleur du produit souhaité	retourne la valeur de la couleur choisie	OK - on obtient la couleur choisie Err - l'utilisateur pourrait ne pas sélectionner une couleur et dans ce cas l'ajout au panier ne peut s'effectuer => voir fonctionnalité "Ajouter au panier".
3_products.js	190 à 193 46 à 48	addEventListener de type change selectColor()	Ecouter le changement de couleur grâce à "addEventListener" et récupérer la valeur choisie	la fonction doit retourner un string contenant la couleur choisie	
<b>5</b> 3_products.js		un input permet de saisir la quantité voulue de 1 à 100	entrer la quantité de produits souhaités	retourne la valeur de la quantité choisie	OK - on obtient la quantité choisie Err - l'utilisateur pourrait saisir une quantité inférieure à 1 ou supérieure à 100 et dans ce cas l'ajout au panier ne peut s'effectuer => voir fonctionnalité "Ajouter au panier".

	Fichier js	Lignes de code	Fonctionnalité	Action	Résultat attendu	Résultat observé
	3_products.js	197 à 200 54 à 56	addEventListener de type input getQuantity()	Ecouter la saisie de la quantité grâce à "addEventListener" et récupérer la valeur choisie	la fonction doit retourner un string contenant la quantité choisie entre 1 et 100	
6	1_helpers.js 2_product.js		possibilité d'ajouter le produit avec les éléments sélectionnés au panier	au clic sur "Ajouter au panier" ajout des éléments sélectionnés au localStorage	Ajout d'un tableau contenant les données sélectionnées sur la page produit (couleur, quantité) sans créer de doublons	OK - les produits s'ajoutent sans créer de doublons Err - un produit existant pourrait s'ajouter en plus dans le panier et créer un doublon
	3_products.js	209 à 211 141 à 188	addEventListener de type click addToCart()	Ecouter le click sur le bouton "Ajouter au panier" et envoyer le/les articles sélectionnés dans le localStorage	ajout du tableau cart au localStorage	
	3_products.js	144 à 146	if( color undefined)	l'utilisateur n'a pas sélectionné de couleur	affichage d'un message d'alerte et annulation de l'envoi du produit au panier	
	3_products.js	149 à 151	if(quantity < 1    quantity > 100)	l'utilisateur à saisie une quantité inférieure à 1 ou supérieure à 100	affichage d'un message d'alerte et annulation de l'envoi du produit au panier	
	3_products.js	63 à 64	calculateTotalPrice()	calculer le prix total de la sélection (quantité x prix unitaire)	montant du prix total	Err - la quantité pourraient ne pas avoir été modifiée en type {number}
	3_products.js 1_helpers.js	158 à 165 50 à 53	for(items in Cart) class Selection	créer l'objet selection contenant le produit sélectionné ainsi que les options de personnalisation ( couleur, quantité, prix total)	regroupement des détails du produits et des options choisies dans un objet selection	
	3_products.js	73 à 76	getCart()	On extrait le tableau cart du localStorage s'il existe	récupération des valeurs du tableau cart	
	3_products.js	83 à 91	isSelectionInCart()	Si le tableau cart existe, on vérifie si le produit sélectionné existe déjà dans le tableau, en le comparant avec les éléments "id" et "color" du tableau	retourne "true" si produit existe	
	3_products.js	98 à 106	getObjInCart()	Si le tableau cart existe, et le produit sélectionné existe déjà dans le panier, récupérer l'index de l'objet à modifier	retourne l'index de l'objet à modifier	
	3_products.js	113 à 115	modifyObjInCart()	modifier l'objet déjà existant en ajoutant la nouvelle quantité de produit et le nouveau prix total	modification du tableau cart	Err - l'objet selection pourrait être ajouté à une nouvelle ligne au lieu de prendre la valeur ajoutée
	3_products.js	122 à 124	addSelectionToLocalStorage()	ajouter le tableau cart au localStorage	localStorage contient le tableau cart avec une clé nommée "sélection"	
	3_products.js	130 à 136	confirmPopUp	afficher une fenêtre confirmant l'ajout du/des articles au panier	affiche la quantité d'articles ajoutée, au clic sur OK, l'utilisateur est redirigé vers la page cart.html, sinon au clic sur Annuler, il est redirigé vers la page d'accueil	
7	3_cart.js 1_helpers.js		Une page panier regroupant un récapitulatifs des produits choisis et un formulaire de contact	affichage de la page panier avec tous les articles sélectionnés classé par nom de produit	Affichage de tous les produits sélectionnés incluant la quantité sélectionnée, le prix total pour chaque article et une quantité et un prix total de la commande au dessus du formulaire utilisateur	OK - Affichage des produits de manière dynamique et rangée par nom de produits Err - les produits pourraient ne pas s'afficher correctement et ne pas être trier
	3_cart.js	48 à 51	getCart()	Extraire le tableau cart du localStorage	récupération des valeurs du tableau cart	
	3_cart.js	59 à 67	compare()	classer les articles par nom de produit	tableau cart réorganisé par nom de produit	
	3_cart.js 1_helpers.js	18 à 36 50 à 79 85 à 89	class Selection instance de la classe Selection - displayProduct displayTotal()	affichage des produits du panier classé par nom de produit	Affichage de tous les produits sélectionnés incluant la quantité sélectionnée, le prix total pour chaque article et une quantité et un prix total de la commande.	

	Fichier js	Lignes de code	Fonctionnalité	Action	Résultat attendu	Résultat observé
8	4_cart.js		Possibilité de modifier la quantité d'un produit	entrer la quantité de produits souhaités	Mis à jour du prix total du produit ainsi que du prix total de la commande et la quantité total de produits	OK - on obtient la quantité choisie Err - l'utilisateur pourrait saisir une quantité inférieure à 1 ou supérieure à 100 et dans ce cas une alerte s'affiche pour indiquer que la quantité saisie est invalide et la page revient à la valeur initialement choisie
	4_cart.js	197 à 200 54 à 56	addEventListener de type change getNewQty()	Ecouter la saisie de la quantité grâce à "addEventListener" et récupérer la valeur choisie	la fonction doit retourner la quantité saisie entre 1 et 100	
	4_cart.js	131 à 132	inputBtn.closest(id)	sélectionner l'élément à modifier	on écoute le changement de quantité seulement de l'élément sélectionné	
	4_cart.js	134 à 137	if(quantity < 1    quantity > 100)	l'utilisateur à saisie une quantité inférieure à 1 ou supérieure à 100	affichage d'un message d'alerte et annulation de l'envoi du produit au panier	
	4_cart.js	90 à 101	changeQty()	modifier la quantité d'un produit dans cart	récupérer la valeur sélectionnée et renvoyer la nouvelle valeur dans localStorage	
	4_cart.js	74 à 76	modifyCart()	ajouter la nouvelle valeur de la quantité dans cart	le produit sélectionné est modifié dans cart	
	4_cart.js	83 à 85	addSelectionToLocalStorage()	renvoyer cart modifié sur le localStorage	cart est envoyé au localStorage	
9	4_cart.js		Possibilité de supprimer un produit du panier	au clic sur "supprimer" le produit doit disparaître de la page panier	Mise à jour du prix total du produit ainsi que du prix total de la commande et la quantité total de produits	OK - une fenêtre d'alerte apparaît pour confirmer la suppression et le produit disparaît du panier et le panier est mis à jour Err - le panier pourrait ne pas se mettre à jour
	4_cart.js	146 à 155	addEventListener de type click removeFromStorage	cliquer sur le bouton "Supprimer"	Afficher une fenêtre de confirmation avant suppression du produit dans le localStorage et supprimer le produit après confirmation	
	4_cart.js	150 à 151	deleteBtn.closest(id)	sélectionner l'élément à modifier	o récupère l'id de l'élément le plus proche du bouton cliqué	
10	5_sendOrder.js		le formulaire de contact doit être validé pour permettre l'envoi de la commande	saisir les données utilisateur dans les champs correspondant	validation des données et envoi de la commande	OK - Les données utilisateur validées, le bouton d'envoi est activé Err - Un message d'erreur apparaît sous le champ à corriger et le bouton d'envoi est inactif ou le formulaire pourrait être envoyé sans être validé
	5_sendOrder.js	127 à 143	addEventListener() de type click sendOrder()	écouter le clic sur le bouton "commander"	envoi des données utilisateurs et de l'ID des produits dans le panier	
	5_sendOrder.js	14 à 16	lettersOnlyValid()	Saisir un nom, un prénom ou une ville dans le formulaire les données saisies ne doivent contenir que des lettres	Si données invalides alors le bouton "Commander" est désactivé et un message d'erreur s'affiche sous le champs invalide	
	5_sendOrder.js	18 à 20	addressValid()	Saisir une adresse dans le formulaire les données saisies peuvent contenir des lettres chiffres et tirets	Si données invalides alors le bouton "Commander" est désactivé	
	5_sendOrder.js	22 à 24	emailValid()	Saisir un email dans le formulaire les données doivent être au format email	Si données invalides alors le bouton "Commander" est désactivé	

Fichier js	Lignes de code	Fonctionnalité	Action	Résultat attendu	Résultat observé
5_sendOrder.js	33 à 42	validFormInput()	valider la saisie des données du formulaire utilisateur	retourne "true" si valeurs saisies sont valides sinon retourne "false".	OK - Les données sont validées et peuvent être envoyées Err - affichage d'une alerte "Données saisies invalides" et message d'erreur sous les champs à corriger
<b>11 5_sendOrder.js 1_helpers.js</b>		<b>Possibilité d'envoyer la commande quand tous les éléments requis sont validés</b>	<b>Au clic sur "commander" la commande est envoyée</b>	<b>ouverture de la page confirmation</b>	<b>OK - Les données utilisateur sont validées, le panier n'est pas vide et la commande est envoyée Err - Le panier pourrait être vide, dans ce cas affichage d'une alerte "Veuillez ajouter un article au panier" et envoi annulé; - Le formulaire pourrait être envoyé sans être valide dans ce cas affichage d'une alerte "Données saisies invalides" et envoi annulé; -L'API pourrait ne pas répondre et ne pas envoyer la commande dans ce cas affichage d'une alerte "Une erreur est survenue pendant l'envoi de la commande, veuillez réessayer plus tard"/Envoi annulé.</b>
1_helpers.js	94 à 98	class Contact	créer un objet "contact" contenant les données saisies dans le formulaire	regroupement des données utilisateur saisies dans le formulaire	
5_sendOrder.js	60 à 65	addProductIdToOrder()	ajouter les Id des produits du panier dans un tableau "products"	un tableau d'ID produits	
5_sendOrder.js	92 à 96	if(products.length == 0)	verifier si le panier contient des articles	Au clic sur commander, si le panier est vide un message d'alerte s'affiche pour ajouter des articles au panier et l'envoi de la commande est annulé	
5_sendOrder.js	98 à 120	Fetch -> POST	Envoyer l'objet "contact" et le tableau "products" à l'API	réponse au format JSON regroupant l'objet contact, le tableau produits et orderId (string) qu'on ajoute en paramètre à l'url de la page confirmation. html	OK - Affichage des produits de manière dynamique Err - L'API pourrait ne pas répondre et dans ce cas affichage d'une alerte d'erreur et réactualisation de la page
<b>12 confirmation.js</b>		<b>Confirmation de la commande et affichage de l'ID de la commande</b>	<b>recupérer l'ID de commande dans les paramètres de l'url</b>	<b>affiche ID de commande</b>	<b>OK - Affichage de la page confirmation.html Err - L'ID de confirmation pourrait ne pas être affichée</b>
confirmation.js	5 à 10	getIdParams()	-Utiliser URLsearchParams pour récupérer l'id du produit à afficher	recupérer le paramètre "orderId" de l'URL pour afficher l'id de confirmation de la commande	