

Plan de tests unitaires - Orinoco

Fichier JS	Lignes de code testées	Element testé	Résultat attendu	Comment vérifier le résultat attendu
Page index.html				
index.html	10	title	Le titre de page apparaît dans l'onglet du navigateur	Ouvrir la page
index.html	16 à 24	header	L'en-tête de la page présente le logo d'Orinoco et un menu de navigation avec les liens cliquables qui renvoie aux pages Accueil (page actuelle) et Panier	Ouvrir la page. Cliquer sur les liens Accueil et Panier
index.html	28 à 31	section intro	Affichage du titre et d'une phrase s'accroche	Ouvrir la page
index.js	5 à 11	fetch	Connexion à l'API	Vérifier le résultat du console.log de "res.ok" : Connexion API, est true
index.js	1	const url	Redirection vers page 404.html si l'API ne répond pas, est erroné, ou inexistante	Supprimer l'url ou renseigner une url erronée
index.js	16 à 17	for of	Récupérations des informations spécifiques à chaque produit	Vérifier le résultat du console.log de "produit" : Informations de chaque produit
index.js	23 à 32	inner.HTML	Affichage de chaque produit avec sa photo, son nom et son prix	Vérifier sur la page que chaque produit présenté a ses informations spécifiques
index.js	2	const htmlCatalogue	Redirection vers page 404.html si l'id de destination est erroné, ou inexistant	Supprimer l'id ou renseigner un id erroné
index.js	25 et 31	a href	Redirection vers la page Produit	Cliquer n'importe où sur une carte produit
index.html	38 à 50	footer	Le pied de page présente les liens cliquables (non fonctionnels pour le moment) : contact, cgv, mentions légales	Ouvrir la page
card.scss	21 à 22	media queries	Le site est responsive sur mobile et tablette : - même informations que sur desktop - menu de navigation se place sous le logo - information textuelle du produit se place sous la photo	Ouvrir le site sur mobile ou tablette.
header.scss	7 à 8			
Page produit.html				
produit.html	11	title	Le titre de page apparaît dans l'onglet du navigateur	Ouvrir la page
produit.html	18 à 24	header	L'en-tête de la page présente le logo d'Orinoco et un menu de navigation avec les liens cliquables qui renvoie aux pages Accueil et Panier	Ouvrir la page. Cliquer sur les liens Accueil et Panier
produit.html	28 à 32	section intro	Affichage du titre	Ouvrir la page
produit.js	3	const produitId	Récupération de l'id du produit sélectionné par l'utilisateur, contenu dans l'URL actuelle de la page	Vérifier le résultat du console.log de "produitId" : Id du produit
produit.js	11 à 17	fetch	Connexion à l'API	Vérifier le résultat du console.log de "res.ok" : Connexion API, est true
produit.js	6	const url	Redirection vers page 404.html si l'API ne répond pas, est erroné, ou inexistante	Supprimer l'url ou renseigner une url erronée
produit.js	20 à 21	fetch	Récupération de l'article sélectionné par l'utilisateur	Vérifier le résultat du console.log de "article" : Infos du produit
produit.js	28 à 41	inner.HTML	Affichage du produit avec sa photo, son nom, sa description, et son prix. (Options voir plus loin dans le plan de tests)	Vérifier si chaque produit présenté sur la page a ses informations spécifiques
produit.js	8	const htmlProduit	Redirection vers page 404.html si l'id de destination est erroné, ou inexistant	Supprimer l'id ou renseigner un id erroné
produit.js	48 à 49	foreach	Récupérations des options spécifiques du produit	Vérifier le résultat du console.log de "option" : Option
produit.js	50 à 51	inner.HTML	Affichage des options spécifiques au produit	Vérifier sur la page que l'article a des options différentes
produit.js	46	const htmlOptions	Redirection vers page 404.html si l'id de destination est erroné, ou inexistant	Supprimer l'id ou renseigner un id erroné
produit.js	58 à 72	addEventListener	A la soumission par le bouton "Ajouter au panier" : récupération des informations du produit personnalisé	Mettre 2 breakpoints lignes 65 et 78 et vérifier si les résultats des console.log de : - "panierObjet" : <i>Produit ajouté au panier</i> , correspond au choix de l'utilisateur - "opionLentille" : <i>Option sélectionnée</i> , correspond au choix sélectionné par l'utilisateur
produit.js	82 à 90	const popupConfirmation	A la soumission par le bouton "Ajouter au panier" : une popup s'affiche confirmant que l'article est ajouté au panier et proposant à l'utilisateur de retourner sur la page Accueil ou d'aller à la page Panier	Cliquer sur "Ajouter au panier", puis sur Annuler pour revenir à l'accueil. Choisir un produit, le personnalisé, puis cliquer à nouveau sur "Ajouter au panier", puis OK pour aller sur la page Panier.

Plan de tests unitaires - Orinoco

Fichier JS	Lignes de code testées	Element testé	Résultat attendu	Comment vérifier le résultat attendu
produit.js	101 à 113	if esle	A la soumission par le bouton "Ajouter au panier" : - Si le panier est vide : un array est créé, le produit y est ajouté et envoyé au localStorage - Si le panier n'est pas vide, le produit est ajouté à l'array créé à la première condition et envoyé au localStorage	Vérifier contenu du localStorage avant d'ajouter le premier produit, puis après l'ajout du premier produit.
produit.html	39 à 51	footer	Le pied de page présente les liens cliquables (non fonctionnels pour le moment) : contact, cgv, mentions légales	Ouvrir la page
card.scss	11 à 12	media queries	Le site est responsive sur mobile et tablette : - même informations que sur desktop - menu de navigation se place sous le logo - information textuelle du produit se place sous la photo	Ouvrir le site sur mobile ou tablette.
header.scss	7 à 8			
Page panier.html				
panier.html	11	title	Le titre de page apparaît dans l'onglet du navigateur	Ouvrir la page
panier.html	18 à 24	header	L'en-tête de la page présente le logo d'Orinoco et un menu de navigation avec les liens cliquables qui renvoie aux pages Accueil et Panier	Ouvrir la page. Cliquer sur les liens Accueil et Panier
panier.html	28 à 32	section intro	Affichage du titre	Ouvrir la page
panier.js	6	const panier	Récupération des données (produit(s) du panier) contenues dans le localStorage	Vérifier le console.log de "panier" : Le localStorage contient les éléments
panier.js	14 à 17	inner.html	Affichage du texte "Le panier est vide", si aucun article n'a été ajouté au panier	Ne rien ajouté au panier ou vider le localStorage, puis aller sur la page
panier.js	24 à 33	for of	Récupérations des informations spécifiques à chaque produit	Vérifier le résultat du console.log de "article" : Informations de chaque produit
panier.js	27 à 32	inner.html	Affichage du produit ajouté au panier : nom, personnalisation, prix	Vérifier sur la page que le produit a ses propres informations
panier.js	44 à 47	for of	Ajout des prix de chaque produit dans un array	Vérifier le console.log de totalPanier : <i>Push prix au total</i>
panier.js	50 à 53	const prixTotal	Addition des prix contenus dans l'array	Vérifier dans le console.log de prixTotal : TOTAL du panier
panier.js	56	localStorage	Envoi du prix total au localStorage	Vérifier que le localStorage contient la même valeur que la const prixTotal
panier.js	59 à 62	inner.html	Affichage du prix total du panier	Vérifier sur la page que le produit a ses propres informations
panier.js	68 à 98	inner.html	Affichage du formulaire de validation de commande, avec : saisie du nom, prénom, email, adresse, ville	Vérifier sur la page que tous les champs apparaissent
panier.js	195 à 200	if else	Redirection vers la page confirmation.html, si la saisie du formulaire est conforme aux formats attendus	Remplir le formulaire en respectant les informations demandées
panier.js	139 à 192	fonctions de validation de champs	Un texte d'erreur ou une bulle s'affiche à côté du champs erroné/inexistant et la soumission ne se déclenche pas	Remplir le formulaire avec des champs vides ou erronés
panier.js	219 à 225	fetch	Une requête est envoyée au serveur avec l'objet "contact" et l'array "products"	Mettre un breakpoint ligne 236 et vérifier console.log "Données à envoyer"
panier.js	231 à 236	fetch	Récupération de la réponse du serveur qu'on envoie au localStorage	Mettre un breakpoint ligne 252 et vérifier key "order" du localStorage
panier.html	49 à 62	footer	Le pied de page présente les liens cliquables (non fonctionnels pour le moment) : contact, cgv, mentions légales	Ouvrir la page
header.scss	7 à 8	media queries	Le site est responsive sur mobile et tablette : - même informations que sur desktop - menu de navigation se place sous le logo	Ouvrir le site sur mobile ou tablette.
Page confirmation.html				
confirmation.html	10	title	Le titre de page apparaît dans l'onglet du navigateur	Ouvrir la page
confirmation.html	17 à 23	header	L'en-tête de la page présente le logo d'Orinoco et un menu de navigation avec les liens cliquables qui renvoie aux pages Accueil et Panier	Ouvrir la page. Cliquer sur les liens Accueil et Panier
confirmation.html	28	section intro	Affichage du titre	Ouvrir la page
confirmation.js	2	const order	Récupération de la réponse du serveur qu'on convertit au format JS	Vérifier console.log de "order" : Réponse serveur. Doit contenir les données de l'objet contact, de l'array products et l'orderId
confirmation.js	16 à 20	innerHTML	Affichage du message de remerciement, du prix total et du numéro de commande	Vérifier sur la page que toutes les données apparaissent

Plan de tests unitaires - Orinoco

Fichier JS	Lignes de code testées	Element testé	Résultat attendu	Comment vérifier le résultat attendu
confirmation.js	23	localStorage	Suppression des données du localStroage	Vérifier contenu du localStorage
header.scss	7 à 8	media queries	Le site est responsive sur mobile et tablette : - même informations que sur desktop - menu de navigation se place sous le logo	Ouvrir le site sur mobile ou tablette.