

Fichier JS	Lignes de code	Fonction testée	Résultat attendu
common.js	18 à 31	updateCartCounter( )	La fonction calcule le nombre de produits enregistré dans le panier (à partir d'entrée correspondant au panier dans <i>localStorage</i> ) et met à jour la valeur affichée dans l'entête du fichier html
common.js	35 à 50	requestItem(type,_id)	Requête à l'API pour obtenir les informations de l'objet de la catégorie de produits <b>type</b> ayant pour id pour <b>_id</b> . Si la requête a aboutie, la fonction renvoie les informations concernant le produit sous la forme d'une chaîne de caractères JSON
common.js	54 à 73	createCurrencies( )	La fonction crée les options de l'élément select du bas de page des pages html (concernant la devise utilisée sur la page), à partir du tableau <b>currencies</b> (s'il existe et n'est pas vide), sélectionne la devise actuellement utilisée par l'utilisateur si elle est enregistrée dans <i>localStorage</i> et dans le cas contraire, choisit par défaut l'euro et l'enregistre dans <i>localStorage</i> .
common.js	75 à 80	updatePrice(price)	La fonction récupère le taux de conversion en fonction de la devise enregistrée dans <i>localStorage</i> et renvoie la conversion de <b>price</b> (en centimes d'euros) dans cette devise et en unités
common.js	91 à 114	hideTags()	La fonction détermine le nombre d'onglets dans le menu de navigation qui seront visibles en fonction de la largeur de la page, en conservant systématiquement l'onglet actif (avec la classe "active") et le dernier onglet : les onglets sont rendus "invisibles" via une classe "hidden".
common.js	116 à 131	fadeOutEffect(item)	La fonction fait disparaître progressivement le premier élément dans le document correspondant au sélecteur <b>item</b> et lui donne la classe "hidden"
common.js	134 à 139	loadImage(img)	La fonction renvoie l'état final d'une promesse, à savoir, en cas de complétion : la valeur <b>true</b> si l'image <b>img</b> est chargée et <b>false</b> sinon.
index.js	2 à 17	requestList(type)	Requête à l'API pour obtenir l'ensemble des produits et leurs informations pour la catégorie de produits <b>type</b> . Si la requête a aboutie, la fonction renvoie ces données sous la forme d'une chaîne de caractères JSON
index.js	21 à 67	creatingListContent(data, ul)	A partir de la chaîne de caractères JSON <b>data</b> qui contient les produits d'une catégorie donnée, la fonction crée pour chaque produit un élément enfant <b>li</b> de la liste <b>ul</b> qui contient les infos du produit (nom, id, image) et un bouton pour rediriger vers la page <i>products.html</i> . Quand tous les éléments sont créés et toutes les images chargées, la promesse est considérée comme aboutie et la fonction renvoie une valeur true.
index.js	69 à 141	createList(arr)	Le tableau <b>arr</b> contient une catégorie de produits (le "nom" utilisé pour appeler l'API spécifique à ce type de produits) et le titre associé (en français). Si la requête pour obtenir les données de cette catégorie aboutie, la fonction : 1) crée une section spécifique de <i>main</i> dans le fichier html 2) crée les éléments de cette section contenant les informations relatives à ce type de produits (nom, nombre) ainsi que la liste des produits. 3) Une fois cette liste est entièrement créée, le <i>loading spinner</i> disparaît et une fois qu'il a disparu, la liste des produits est affichée. La fonction renvoie alors un état final de complétion de la promesse.
index.js	143 à 151	doNextList(d,obj)	<b>Obj</b> est un objet contenant les tableaux de la forme <b>arr = [nom, titre]</b> correspondant aux catégories de produits. La fonction fait appel à la fonction <i>createList(arr)</i> pour les entrées de <b>obj</b> à partir de l'indice <b>d</b> jusqu'à la fin, les unes après les autres.  La fonction crée donc successivement les sections (et les listes de produits associés) pour les différentes catégories d'objets listées dans <b>obj</b> de l'indice d jusqu'aux derniers.

index.js	156 à 163	sameRow( )	La fonction attribue la classe "sameRow" à tout élément correspondant au sélecteur li.item (les produits du catalogue) qui sont à la gauche d'un autre élément correspondant au sélecteur <i>li.item</i>
product.js	2 à 13	checkCart(type)	La fonction vérifie si un panier correspondant à la catégorie type de produits existe déjà (dans localStorage) et renvoie : - En cas d'échec (si un panier d'une autre catégorie existe déjà), le nom de cette autre catégorie - En cas de réussite, la valeur <i>true</i> si un panier de la catégorie <b>type</b> existe déjà, <i>false</i> si aucun panier n'existe
cart.js	2 à 12	isValid(input,regex)	La fonction compare l'expression régulière <b>regex</b> avec la valeur de <b>input</b> : la fonction renvoie <i>true</i> si une correspondance entre les deux est trouvée, et <i>false</i> sinon.
cart.js	101 à 117	sendOrder(type, order)	Requête à l'API correspondant à la catégorie <b>type</b> pour passer commande et envoyer le JSON <b>order</b> contenant les informations de l'utilisateur et le tableau des id des produits achetés. Si la requête a aboutie, la fonction renvoie une chaîne de caractères JSON contenant la référence de la commande.