

Fichier JS	Lignes de code testées	Fonction testée	Résultat attendu	Comment vérifier le résultat attendu	Problème possible
main.js	4 à 10	cutString	La fonction doit retourner une chaîne de caractère d'une certaine longueur passée en paramètre.	Appeler la fonction avec différents paramètres (chaîne de caractères et longueur max) et on observe la chaîne retournée avec un console.log par exemple	Je ne lui ai pas envoyé une chaîne de caractères. Si j'appelle la fonction en envoyant une mauvaise valeur en paramètre.
main.js	15 à 17	priceFormat	La fonction doit retourner un prix (passé en paramètre en centimes) en euro.	Appeler la fonction avec différentes valeurs de prix et on observe la valeur retournée	Si j'appelle la fonction en envoyant une mauvaise valeur en paramètre. Si le paramètre passé n'est pas en centimes.
Cameras.js	4 à 9	getAllCameras	La fonction doit récupérer les caméras depuis l'API.		La connexion à l'API peut échouer.
Cameras.js	14 à 44	setLayoutCameras	La fonction doit, pour chaque caméras reçues en paramètre, ajouter au DOM des éléments.	Appeler la fonction puis vérifier que les éléments existent bien dans le DOM (assertExists).	
Cameras.js	49 à 54	getDetailsOfCamera	La fonction doit récupérer la caméra depuis l'API grâce à son ID.		La connexion à l'API peut échouer.
Cameras.js	59 à 115	setLayoutDetails	La fonction doit ajouter des éléments au DOM si l'objet passé en paramètre n'est pas vide.	Appeler la fonction puis vérifier que les éléments existent bien dans le DOM (assertExists).	Changements du format de la réponse API.
Basket.js	18 à 34	addItemToCart	La fonction doit ajouter la caméra au panier.	Vérifier que l'ID de la caméra a bien été ajouté au panier en faisant un console.log avant et après.	La création du panier n'est pas prévue dans cette fonction, le panier peut, donc, ne pas exister.
Basket.js	39 à 41	getNumberProductInCart	La fonction doit retourner le nombre d'éléments présent dans le panier	Faire un console.log du panier et compter les éléments.	Le panier peut ne pas être initialisé.
Basket.js	46 à 51	setNumberProductInCart	La fonction doit mettre à jour l'affichage du nombre de d'éléments présent dans le panier.	Faire un console.log du panier et compter les éléments.	Le panier peut ne pas être initialisé.
Basket.js	61 à 66	getProductInCart	La fonction doit récupérer la caméra dont l'ID passé en paramètre correspond.	Faire un console.log de la caméra retournée.	La caméra correspondante peut ne pas exister.
Basket.js	71 à 85	setLayoutBasket	La fonction doit appeler la fonction getProductInCart pour chaque éléments présents dans le panier, si le panier n'est pas vide.		
Basket.js	95 à 132	setListOfProduct	La fonction doit ajouter des éléments au DOM selon l'objet passé en paramètre.		L'objet peut être vide
Basket.js	137 à 150	removeItemtoCart	La fonction doit supprimer un produit du panier	Faire un console.log du panier et compter les éléments.	Le panier peut ne pas être initialisé.
Basket.js	155 à 158	cleanBasket	La fonction doit vider le panier et rediriger vers la page d'accueil.	Faire un console.log du panier et vérifier qu'il est vide	Le panier peut ne pas être initialisé.

Order.js	4 à 62	validForm	La fonction doit valider les champs du formulaire de commande.		
Order.js	67 à 86	sendForm	La fonction doit envoyer les données de la commande à l'API puis enregistrer la réponse de l'API et vider le panier.	Faire un console.log du résultat et du panier.	La connexion à l'API peut échouer.
Order.js	98 à 123	setLayoutConfirmation	La fonction doit mettre en page la confirmation de commande si une commande vient d'être passée.	Passer une commande et vérifier les informations reçu sur la page de	