

| Fichier JS | Lignes de code testées | Fonction testée | Résultat attendu | Comment vérifier le résultat attendu | Problème possible |
|------------------------|------------------------|------------------------|---|--|---|
| addToCart.js | 8-71 | addToCart() | ajoute au localStorage le contenu du panier. Si le panier est vide : ajout du produit sélectionné. Si le panier contient un ou plusieurs produits : utilise la fonction IIFE pour modifier la liste des produits et ajouter cette nouvelle liste | appeler la fonction au clic sur le bouton, et vérifier dans le localStorage que la clé "products" se crée bien lorsque le localStorage était préalablement vide (voir ensuite fonction IIFE) | - la clé ne se crée pas et le localStorage reste vide - la clé se crée au premier ajout, mais l'appel de la fonction IIFE ne se fait pas |
| addToCart.js | 41-62 | IIFE | vérifie la présence du produit sélectionné dans le panier. Si le panier ne contient pas le produit sélectionné : ajoute le produit sélectionné à la liste de produits et envoie la liste au localStorage. Si le panier contient le produit sélectionné : modifie la quantité de ce produit dans la liste de produits et envoie la liste au localStorage | appeler la fonction au clic sur le bouton après avoir vérifié la présence de la clé "products" au localStorage, et vérifier : - que la clé est bien modifiée pour constituer un array des produits préalablement existants + nouveau produit sélectionné si la clé était déjà présente et qu'on sélectionne un produit différent - que la quantité d'un produit déjà présent dans la clé est bien modifiée au lieu d'un ajout d'un élément supplémentaire à l'array si on sélectionne un produit identique | - l'ajout d'un produit supplémentaire modifie les informations de l'objet initial au lieu de le modifier en array d'objets pour chaque produit - à l'ajout d'un produit supplémentaire identique à un élément de la clé existante, un élément supplémentaire est ajouté à l'array au lieu de modifier la quantité de l'élément identique |
| formatCart.js | 5-15 | formatCart() | formate le contenu du panier récupéré dans le localStorage pour l'uniformiser sous forme d'array pour les appels ultérieurs | appeler la variable "arrayOfProducts" dans le console.log et vérifier qu'il s'agit bien d'un array | le console.log affiche "null" ou un objet qui n'est pas un array si la fonction ne fonctionne pas |
| getFormattedPrice | 3-9 | getFormattedPrice() | formate le prix affiché sous la forme 00,00 | vérifier dans console.log l'affichage de getFormattedPrice(0000) | le console.log affiche 0000 si la fonction ne fonctionne pas |
| getFormattedPrice | 11-16 | calculateTotal() | calcule le total du prix du panier et en formate l'affichage en utilisant getFormattedPrice() | faire un console.log de la fonction et vérifier que le nombre affiché correspond bien à la somme attendue en fonction des objets présents dans le localStorage | la somme attendue est incorrecte ou l'appel à la fonction getFormattedPrice ne se fait pas et le format d'affichage n'est pas 00,00 |
| getNumberOfProducts.js | 3-14 | getTotalQuantity() | comptabilise le nombre de produits présents dans le panier et renvoie ce nombre | faire un console.log de la fonction et vérifier que le nombre affiché correspond bien à la somme attendue en fonction des objets présents dans le localStorage | la somme attendue est incorrecte |
| getNumberOfProducts.js | 20-24 | displayTotalQuantity() | affiche le nombre de produits comptabilisé avec getTotalQuantity() | vérifier dans le DOM la création de la portant le numéro comptabilisé uniquement si celui-ci est différent de 0 | la n'est pas intégrée au DOM ou elle est intégrée même si getTotalQuantity() renvoie 0 ou l'appel à la fonction getTotalQuantity() ne se fait pas |

| | | | | | |
|----------------|-------|----------------------|--|---|--|
| getTeddies.js | 5-16 | getTeddies() | appelle l'API pour récupérer la liste des teddies sur la page d'accueil et les retourne au format .json | faire un console.log de la fonction asynchrone, en passant en paramètre l'url de l'API, vérifier le retour de la liste des teddies en résultat sous forme d'array | - pas d'accès au serveur de l'API ou accès au serveur mais pas à l'url correcte : affichage du message d'erreur 'Failed to fetch' et la fonction formatDataTeddies() ne peut pas afficher la liste des produits - l'accès à l'url de l'API se fait, mais le retour n'est pas au format .json et l'affichage de la liste des produits restera impossible : affichage du message d'erreur 'Invalid data format' |
| getTeddies.js | 18-43 | formatDataTeddies() | formate la liste des teddies récupérée de l'API par la fonction getTeddies() pour en permettre l'affichage sur la page d'accueil grâce à la fonction displayTeddies() | faire un console.log de l'objet teddy créé et vérifier qu'il présente bien les clefs implémentées | - l'accès à l'API ne s'est pas fait correctement, et l'objet .json n'est pas créé, la fonction ne peut donc pas s'exécuter - la fonction s'exécute, mais l'objet récupéré in fine ne comporte pas les clefs demandées |
| getTeddies.js | 47-48 | getOneTeddy() | appelle l'API pour récupérer le détail du teddy sélectionné par son ID sur la page produit, retourné au format .json, en utilisant la fonction getTeddies() | faire un console.log de la fonction asynchrone, en passant en paramètre "await {l'url de l'API}/{un id}" et vérifier le retour de l'objet teddy en résultat | - pas d'accès au serveur de l'API ou accès au serveur mais pas à l'url correcte : affichage du message d'erreur 'Failed to fetch' et la fonction formatDataOneTeddy() n'affiche pas le détail du teddy - l'accès à l'url de l'API se fait, mais le retour n'est pas au format .json : le contenu de la page se charge mais les informations restent en erreur + affichage du message d'erreur 'Invalid data format' |
| getTeddies.js | 50-69 | formatDataOneTeddy() | formate l'objet teddy à partir des informations récupérées par la fonction getOneTeddy() pour en permettre l'affichage sur la page produit par la fonction displayOneTeddy() | faire un console.log de l'objet teddy et vérifier qu'il s'affiche et qu'il présente bien les clefs nécessaires et que leur valeur est bien renseigné | - si l'accès au serveur a échoué, la fonction ne pourra pas déstructurer l'objet .json, et l'objet teddy ne sera pas créé - si l'objet est créé mais que la valeur des clefs est "undefined" : affichage du message d'erreur 'Invalid data format' |
| htmlContent.js | 7-19 | displayTeddies() | gère l'affichage des cartes des teddies sur la page d'accueil en fonction des informations récoltées par formatDataTeddies(getTeddies()) | vérifier dans le code source de la page la création des éléments html correspondant aux cartes des teddy dans le DOM | - erreur d'appel à l'élément du DOM dans lequel intégrer les cartes - erreur de syntaxe html dans les éléments créés |
| htmlContent.js | 24-50 | displayOneTeddy() | gère l'affichage du détail d'un teddy sur la page produit en fonction des informations récoltées par formatDataOneTeddy(getOneTeddy()) | vérifier dans le code source de la page la création des éléments html correspondant au détail d'un teddy dans le DOM | - erreur d'appel à l'élément du DOM dans lequel intégrer le détail du teddy - erreur de syntaxe html dans les éléments créés (détail du produit, bouton quantité, bouton Ajouter au panier) |

| | | | | | |
|----------------|---------|----------------------|---|--|---|
| htmlContent.js | 56-85 | selectColor() | affiche les coloris disponibles et sélectionne la couleur du teddy choisie par l'utilisateur sur la page produit | <ul style="list-style-type: none"> - vérifier dans le code source de la page la création des éléments html correspondant aux pastilles de couleur dans le DOM - vérifier la sélection de la couleur au clic sur la pastille (changement de class) et console.log de l'élément choiceColor créé | <ul style="list-style-type: none"> - erreur d'appel à l'élément du DOM dans lequel intégrer les pastilles - erreur de syntaxe html dans les éléments créés - erreur de transposition du nom de couleur au format bas de casse sans espace : erreur dans le format de choiceColor - problème de sélection au check de l'input (non suppression de la class "active" des éléments non sélectionnés, non ajout de la class "active" à l'élément sélectionné) |
| htmlContent.js | 90-123 | cartContent() | affiche le contenu du panier sur la page panier | <ul style="list-style-type: none"> - vérifier dans le code source de la page la création des éléments html correspondant au contenu du panier dans le DOM | <ul style="list-style-type: none"> - erreur d'appel à l'élément du DOM dans lequel intégrer les éléments du panier - erreur de syntaxe html dans les éléments créés |
| htmlContent.js | 127-170 | formContent() | affiche le formulaire de contact sur la page panier | <ul style="list-style-type: none"> - vérifier dans le code source de la page la création des éléments html correspondant au formulaire dans le DOM et la suppression de la class "visually-hidden" qui dissimule le formulaire avant le clic sur le bouton "valider le panier" | <ul style="list-style-type: none"> - erreur d'appel à l'élément du DOM dans lequel intégrer le formulaire - erreur de syntaxe html dans les éléments créés, notamment dans les pattern requis pour les input - non suppression de la class "visually-hidden" |
| htmlContent.js | 174-186 | checkoutContent() | affiche le contenu de la page checkout | <ul style="list-style-type: none"> - vérifier dans le code source de la page la création des éléments html correspondant à la confirmation de commande dans le DOM | <ul style="list-style-type: none"> - erreur d'appel à l'élément du DOM dans lequel intégrer la confirmation de commande - erreur de syntaxe html dans les éléments créés - erreur d'appel à la fonction getStorageItem() |
| popup.js | 9-15 | createPopup() | ouvre un popup d'alerte, en précise le style et l'ajoute au DOM | <ul style="list-style-type: none"> - vérifier dans le code source de la page la création des éléments html correspondant au popup dans le DOM | <ul style="list-style-type: none"> - le bloc ne se crée pas ou ses attributs ne sont pas correctement renseignés, l'affichage sera alors inexistant ou incorrect sur la page |
| popup.js | 19-24 | closePopup() | ferme le popup d'alerte en supprimant ses éléments du DOM | <ul style="list-style-type: none"> - vérifier au clic sur le bouton que tous les éléments du popup sont supprimés du DOM | <ul style="list-style-type: none"> - les éléments ne sont pas supprimés et le popup reste apparent |
| popup.js | 30-39 | createErrorMessage() | crée le popup de message d'erreur d'accès à l'API sur la page d'accueil en utilisant createPopup(). En gère la fermeture en appelant closePopup() | <ul style="list-style-type: none"> - appeler la fonction en passant un string test en argument et vérifier dans le code source de la page la création des éléments html correspondant au popup dans le DOM - vérifier la fermeture du popup au clic sur le bouton | <ul style="list-style-type: none"> - l'appel à la fonction createPopup() ne se fait pas, le popup n'est pas créé - erreur de syntaxe html dans l'élément créé pour le message d'erreur - erreur d'appel à la fonction closePopup(), au clic sur le bouton, le popup ne se ferme pas |

| | | | | | |
|----------|---------|------------------------------|--|--|---|
| popup.js | 43-56 | createContentAddToCart() | crée le contenu du popup d'ajout au panier. Au clic sur le bouton, renvoie sur la page en cours ou sur le panier selon le bouton sélectionné, et ferme le popup en appelant closePopup() | - appeler la fonction en passant en argument un objet {name: test1, color: test2} et vérifier dans le code source de la page la création des éléments html correspondant au popup dans le DOM - vérifier la fermeture du popup au clic sur le bouton | - erreur de syntaxe html dans l'élément créé pour le message d'erreur - erreur d'appel à la fonction closePopup(), au clic sur un bouton, le popup ne se ferme pas - au clic sur le bouton continuer votre shopping, la page ne rafraîchit pas |
| popup.js | 60-67 | createContentOrderInvalid() | crée le contenu du popup alertant d'un formulaire de commande non conforme. Revient sur la page en cours à la fermeture au clic sur le bouton en appelant closePopup() | - appeler la fonction et vérifier dans le code source de la page la création des éléments html correspondant au popup dans le DOM - vérifier la fermeture du popup au clic sur le bouton | - erreur de syntaxe html dans l'élément créé pour le message d'erreur - erreur d'appel à la fonction closePopup(), au clic sur un bouton, le popup ne se ferme pas |
| popup.js | 71-99 | createContentValidateOrder() | crée le contenu du popup pour confirmation de la commande. Au clic sur le bouton, récupère les champs du formulaire en utilisant getInputValue() renvoyés dans l'objet contact, et crée l'objet productsID. Appelle la fonction sendingToBack() pour envoi au back et retour du numéro de commande ou message d'erreur | - appeler la fonction et vérifier dans le code source de la page la création des éléments html correspondant au popup dans le DOM - désactiver la fonction sendingToBack() et ajouter un console.log(contact, productsID); pour vérifier la création des éléments à envoyer au back au clic sur le bouton | - erreur de syntaxe html dans l'élément créé pour le message d'erreur - au clic sur un bouton, les objets à envoyer au back ne sont pas créés ou sont non conformes au format attendu - l'appel à la fonction sendingToBack() ne se fait pas |
| popup.js | 78-81 | getInputValue() | récupère les valeurs des champs du formulaire | faire un console.log de la fonction en passant en paramètre l'id de n'importe lequel des inputs du formulaire | - erreur d'appel à l'élément du DOM correspondant à l'id renseigné - la valeur de l'input n'est pas renvoyée |
| popup.js | 103-112 | addErrorMessage | modifie le contenu du popup pour alerter l'utilisateur d'une erreur lors de la confirmation de la commande, grise le bouton de confirmation de commande (inactif) et génère un nouveau bouton de fermeture du popup pour retour à la page en cours | - appeler la fonction en passant une string test en paramètre et vérifier dans le code source de la page la création des éléments html correspondant à la modification du message d'erreur dans le DOM - vérifier l'ajout de la class "button-inactive" dans le code source - vérifier la fermeture du popup au clic sur le bouton | - l'ajout de la class "button-inactive" ne se fait pas, le bouton de confirmation de commande reste coloré - erreur de syntaxe html dans l'élément créé pour le message d'erreur - erreur d'appel à l'élément du DOM correspondant au nouveau bouton créé - erreur d'appel à la fonction closePopup(), au clic sur le nouveau bouton, le popup ne se ferme pas |

| | | | | | |
|------------------|-------|---|---|---|---|
| sendingToBack.js | 4-36 | sendingToBack() | <p>envoie les paramètres renseignés au back-end pour retour du numéro de commande. Si tout est conforme : récupération de l'orderId qui est renvoyé au localStorage par la fonction setStorageItem(), vide le panier par la fonction removeStorageItems("products"), ferme le popup par la fonction closePopup() et renvoie sur la page checkout. Si une erreur est constatée : appel de la fonction addErrorMessage() pour alerter l'utilisateur</p> | <p>- vérifier dans le localStorage l'ajout de la clé orderId et la suppression de la clé produits, le popup doit se fermer et la page checkout doit s'être affichée au clic sur le bouton</p> <p>- générer une erreur en modifiant l'adresse de fetch et vérifier la création du message d'erreur et que rien n'est créé ni supprimé dans le localStorage</p> | <p>- l'envoi au back ne se fait pas car l'accès au serveur ne fonctionne pas : le message d'erreur s'affiche 'Failed to fetch'</p> <p>- l'envoi se fait mais comporte des erreurs, le retour du serveur ne contient pas l'orderId conforme : le message d'erreur s'affiche 'Erreur 400' (Bad request)</p> <p>- l'orderId n'est pas enregistré dans le localStorage et/ou la clé produit n'est pas supprimée à la validation de la commande</p> <p>- l'orderId est enregistré dans le localStorage et/ou la clé produit est supprimée alors qu'il y a une erreur</p> <p>- erreur d'appel à la fonction closePopup(), au clic sur le nouveau bouton, le popup ne se ferme pas</p> <p>- la redirection vers la page checkout ne s'effectue pas</p> |
| storage.js | 1-3 | getStorageItem() | accède au localStorage et en récupère le contenu par la clé renseignée en paramètre | faire un console.log de la fonction en passant en paramètre n'importe quelle clé visualisée dans la console > application > local storage | le log n'affiche pas le contenu de la clé, les informations ne sont pas récupérées |
| storage.js | 5-7 | setStorageItem() | envoi une clé et son contenu passés en paramètres au localStorage | appeler la fonction en passant une clé et un contenu tests en paramètres, et vérifier dans la console > application > local storage, la création de cette clé et de son contenu | rien ne s'affiche dans le localStorage |
| storage.js | 9-11 | removeStorageItem() | supprime la clé renseignée en paramètre du localStorage | appeler la fonction en passant en paramètre n'importe quelle clé visualisée dans la console > application > local storage et vérifier qu'elle en est bien supprimée | la clé subsiste dans la console > application > local storage |
| storage.js | 13-16 | emptyLocalStorage(orderId, total) | supprime les deux clés restantes en page checkout (orderId, total) du localStorage | appeler la fonction et vérifier dans la console la suppression des deux clés appelées en paramètres | les clés subsistent dans la console > application > local storage |
| form.js | 19-30 | fonction à l'écoute de la modification des inputs | vérifie la validité d'un champ du formulaire en fonction du pattern requis et si une erreur est constatée, modifie le style de l'input pour visualisation des champs erronés (bordure rouge, croix rouge). La bordure des champs correctement renseignés disparaît et une coche verte apparaît à la place de la croix | vérifier à la frappe dans les champs de formulaire que la modification du style se fait bien en fonction des patterns requis | le style n'est pas modifié, la croix ou la coche n'apparaissent pas |

| | | | | | |
|---------|--------|--------------------|--|---|--|
| form.js | 35-59 | checkValidity() | <p>au clic sur le bouton "valider la commande", vérifie la validité de l'ensemble des champs du formulaire requis et si au moins une erreur ou omission est constatée, renvoie le message d'erreur passé en second argument pour l'input passé en premier argument et modifie le style de l'input concerné</p> | <ul style="list-style-type: none"> - renseigner des erreurs dans les champs du formulaire, et vérifier au clic sur le bouton que les inputs en erreur présentent bien la class invalide, et que le message d'erreur adéquat s'affiche - modifier les champs erronés pour en retirer les erreurs et vérifier qu'au clic, la class invalide et le message d'erreur sont retirés | <ul style="list-style-type: none"> - les messages personnalisés ne sont pas pris en compte - la class invalide ne s'ajoute pas - lors de la correction, la suppression de la class invalide et des messages d'erreur ne se fait pas |
| form.js | 92-100 | checkAllValidity() | <p>vérifie que toutes les vérifications de checkValidity obtiennent le résultat 'true'</p> | <p>faire un console.log() de la fonction et vérifier qu'elle est bien false lorsque le formulaire présente une erreur et true lorsqu'il est conforme</p> | <p>la fonction ne renvoie pas la bonne valeur</p> |