

Etape	Action	Ligne de code	Résultats	Problème potentiel
1	ouverture de la page d'accueil	méthode fetch ayant pour argument l'url de l'api renvoyant tous les produits	tous les produits disponibles s'affichent	le serveur n'est pas lancé, et un message d'erreur s'affiche dans la console
2	clic sur un article de la page d'accueil	Chaque carte produit possède un lien vers la page produit. Ce lien a pour paramètre l'id du produit	redirection sur la page produit	le fichier de la page produit est introuvable
3	arrivée sur la page produit	La méthode window.location.search.substring(3) récupère la chaîne de caractère qui nous intéresse soit l'id produit. Celui-ci est stocké dans une variable. On fait un console.log pour vérifier si l'id est bien récupéré	l'id du produit choisi s'affiche dans la console	
4	chargement de la page produit	On utilise la méthode fetch ayant pour argument l'url de l'api et pour paramètre la variable id	le produit sélectionné s'affiche	la console affiche une erreur
5	personnalisation de l'article	Utilisation de structures conditionnelles 'if' qui permettent au menu déroulant de s'adapter au nombre d'options disponibles pour le produit choisi	selon le produit choisi, le menu déroulant de personnalisation affichera 1 option, 2 options ou 3 options.	le mot 'undefined' s'affiche sur un des champs du menu déroulant
6	gestion de la quantité	On emploie un addEventListener qui écoute le click sur les boutons + et - . La quantité par défaut est égale à 1.	Au click de la souris sur les boutons, la quantité augmente ou diminue.	
7	modification dynamique du prix	Dans le addEventListener qui écoute le click sur les boutons + et -, on sélectionne l'id correspondant au prix et on le multiplie par la quantité.	Au click de la souris sur les boutons, le prix se modifie directement.	
8	ajout du produit au panier	Dans le addEventListener qui gère l'ajout au panier, on crée un affichage dynamique d'une carte alertant l'utilisateur que son produit a bien été ajouté.	Une fenêtre s'affiche pour indiquer que le produit a bien été ajouté au panier. L'utilisateur peut revenir à l'accueil pour poursuivre ses achats ou voir son panier pour commander.	

9	stockage des données produits dans le local storage	Le addEventListener qui gère l'ajout au panier comporte des méthodes permettant aux données du produits d'être stockées dans le local storage.	Sur la page panier, le nom du produit, l'option choisie, la quantité choisie ainsi que le prix s'affichent.	L'un des champs de données affiche "undefined" ou "NaN"
10	calcul du montant total à payer	On crée un tableau contenant les prix de chaque ligne du tableau, et on y applique une boucle 'for' pour itérer sur les prix du tableau. On utilise ensuite la méthode reduce pour additionner les prix de chaque ligne	Le prix total est égal à la somme des prix contenus dans le panier	Le champs du prix Total, n'affiche pas un prix correct ou affiche 'NaN' ou 'Undefined'.
11	renseignement du champs 'Nom' du formulaire	Le champs est formaté par une expression régulière de manière à ce que seul des lettres et des tirets soient acceptés	Un message vert apparaît lorsque les conditions sont respectées afin d'informer l'utilisateur que le champs est valide	Un message rouge s'affiche pour informer que le champs n'est pas valide
12	renseignement du champs 'Prénom' du formulaire	Le champs est formaté par une expression régulière de manière à ce que seul des lettres et des tirets soient acceptés	Un message vert apparaît lorsque les conditions sont respectées afin d'informer l'utilisateur que le champs est valide	Un message rouge s'affiche pour informer que le champs n'est pas valide
13	renseignement du champs 'Adresse' du formulaire	Le champs est formaté par une expression régulière de manière à ce que des chiffres, des espaces, des lettres et des tirets soient acceptés	Un message vert apparaît lorsque les conditions sont respectées afin d'informer l'utilisateur que le champs est valide	Un message rouge s'affiche pour informer que le champs n'est pas valide
14	renseignement du champs 'Ville' du formulaire	Le champs est formaté par une expression régulière de manière à ce que seul des lettres et des tirets soient acceptés	Un message vert apparaît lorsque les conditions sont respectées afin d'informer l'utilisateur que le champs est valide	Un message rouge s'affiche pour informer que le champs n'est pas valide
15	renseignement du champs 'Email' du formulaire	Le champs est formaté par une expression régulière de manière à ce que le champs comportent un '@', et un '.' avant la dernière chaîne de caractère (com ou fr)	Un message vert apparaît lorsque les conditions sont respectées afin d'informer l'utilisateur que le champs est valide	Un message rouge s'affiche pour informer que le champs n'est pas valide

16	Clic sur le bouton commander	Un addEventListener, permet à ce qu'au clic du bouton, une requête POST envoie les données du formulaire ainsi qu'un tableau contenant les id des produits choisis au serveur. Le serveur renvoi un numéro de commande. Ce numéro de commande ainsi que le prix total de la commande sont stockés dans le local storage.	L'utilisateur est redirigé vers la page confirmation.	
17	Arrivée sur la page confirmation	On récupère le numéro de commande ainsi que le prix total stockés dans le local storage, pour les afficher sur la page confirmation.	Un message de confirmation contenant le numéro de commande ainsi que le prix total s'affiche sur la page.	