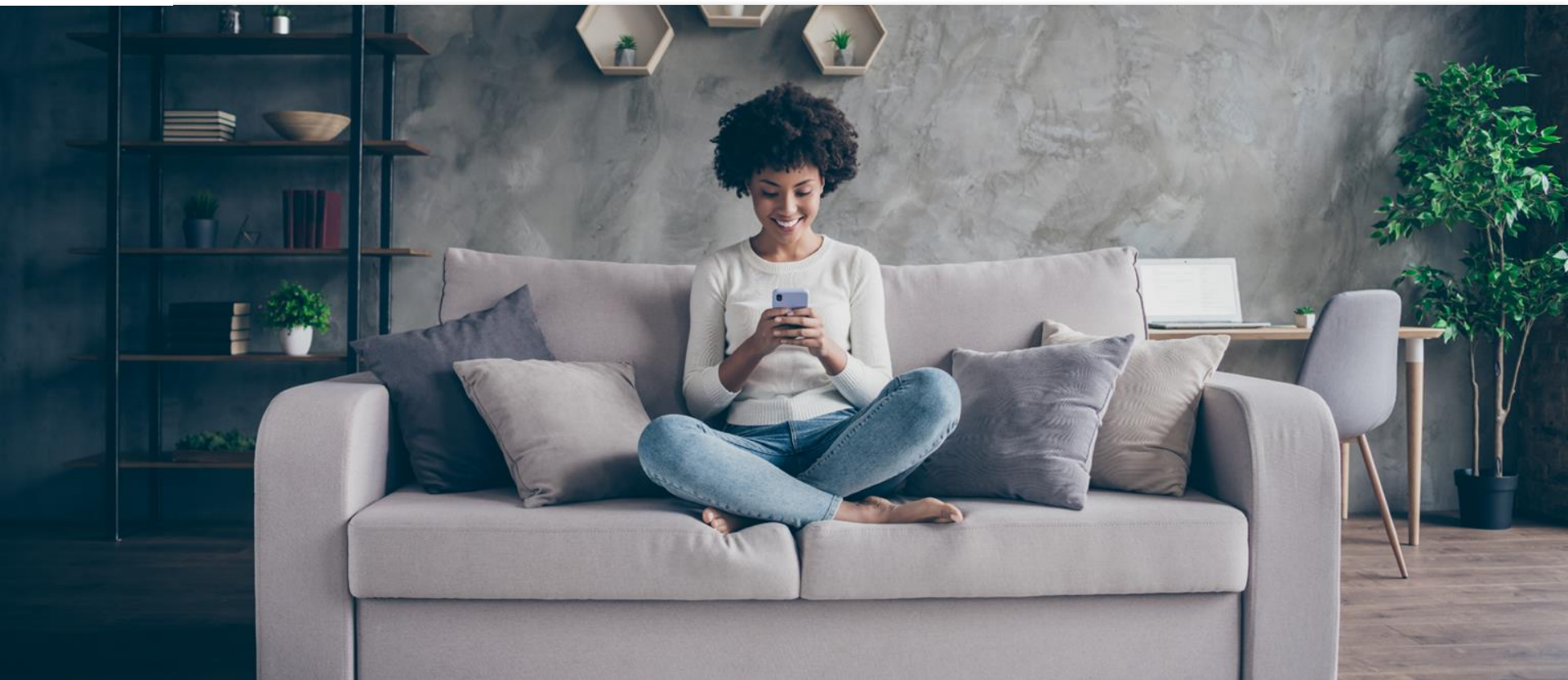




KANAP



Contexte

Mission pour le client **KANAP**
Plateforme e-commerce

Mission :

- Intégrer dynamiquement des éléments via une API
- Mettre en place un plan de test d'acceptation

1 - Présentation du site

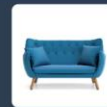
- Visualiser les différentes pages du site

2 - Explication du code des différentes pages JS

- API
- URLSearchParams
- Local Storage
- AddEventListener

3 - Explication du plan de test

- Visualiser les différentes fonctionnalités attendu



KANAP SINOPÉ

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.



KANAP CYLLÈNE

Morbi nec erat aliquam, sagittis urna non, laoreet justo. Etiam sit amet interdum diam, at accumsan lectus.



KANAP CALYCÉ

Pellentesque fermentum arcu venenatis ex sagittis accumsan. Vivamus lacinia fermentum tortor. Mauris imperdiet tellus...



KANAP AUTONOÉ

Donec mattis nisl tortor, nec blandit sapien fermentum at. Proin hendrerit efficitur fringilla. Lorem ipsum dolor sit...



KANAP EURYDOMÉ

Ut laoreet vulputate neque in commodo. Suspendisse maximus quis erat in sagittis. Donec hendrerit purus at congue aliquam.



KANAP HÉLICÉ

Curabitur vel augue sit amet arcu aliquet interdum. Integer vel quam mi. Morbi nec vehicula mi, sit amet vestibulum.



KANAP THYONÉ

EMauris imperdiet tellus ante, sit amet pretium turpis molestie eu. Vestibulum et egestas eros. Vestibulum non lacus orci.



KANAP ORTHOSIE

Mauris molestie laoreet finibus. Aenean scelerisque convallis lacus at dapibus. Morbi imperdiet enim metus rhoncus.

Visualisation du site

Explication du code

Script. JS

```
// On récupère des données de l'API via fetch
fetch("http://localhost:3000/api/products/")
// On obtient la réponse de l'API qu'on converti au format JSON
.then((res) => res.json())
// Les données JSON sont appelées "produits"
.then((produits) => {
  // Informations sur les produits en console sous forme de tableau
  console.table(produits)
  // On retourne l'affichage des produits avec la fonction "afficherProduits"
  afficherProduits(produits)
})
// Si une erreur se produit => On remplace le contenu du titre h1
// avec "erreur 404" et on renvoi une erreur en console
.catch((err) => {
  document.querySelector(".titles").innerHTML = "<h1>erreur 404<h1>"
  console.log("API => erreur 404 : " + err)
})
```

Fetch

(..	colors	_id	name	price	imageUrl	description	altTxt
0	Array(3)	'107fb5b75607497b96722bda5b504926'	'Knap Sinopé'	1849	'http://localhost:3000/images/kanap01.jpeg'	'Excepteur ...	"Photo d'un canapé bleu, deux places"
1	Array(2)	'415b7cacb65d43b2b5c1ff70f3393ad1'	'Knap Cyllène'	4499	'http://localhost:3000/images/kanap02.jpeg'	'Morbi nec ...	"Photo d'un canapé jaune et noir, quatre places"
2	Array(3)	'055743915a544fde83cfdcf904935ee7'	'Knap Calycé'	3199	'http://localhost:3000/images/kanap03.jpeg'	'Pellentesq...	"Photo d'un canapé d'angle, vert, trois places"
3	Array(2)	'a557292fe5814ea2b15c6ef4bd73ed83'	'Knap Autonoé'	1499	'http://localhost:3000/images/kanap04.jpeg'	'Donec matt...	"Photo d'un canapé rose, une à deux place"
4	Array(3)	'8906dfda133f4c20a9d0e34f18adc06'	'Knap Eurydomé'	2249	'http://localhost:3000/images/kanap05.jpeg'	'Ut laoreet...	"Photo d'un canapé gris, trois places"
5	Array(2)	'77711f0e466b4ddf953f677d30b0efc9'	'Knap Hêlicé'	999	'http://localhost:3000/images/kanap06.jpeg'	'Curabitur ...	"Photo d'un canapé gris, deux places"
6	Array(2)	'034707184e8e4ee46400b5a3774b5f'	'Knap Thyoné'	1999	'http://localhost:3000/images/kanap07.jpeg'	'EMauris im...	"Photo d'un canapé rouge, deux places"
7	Array(4)	'a6ec5b49bd164d7f9e10f37b6363f9fb'	'Knap orthosie'	3999	'http://localhost:3000/images/kanap08.jpeg'	'Mauris mol...	"Photo d'un canapé rose, trois places"

Fonction d'affichage

```
function afficherProduits(produits) {  
  // On déclare une variable qui cible l'ID de la section "#items"  
  const tousLesProduits = document.querySelector("#items");  
  // On fait une boucle pour chaque indice "produit" des "produits" de l'API  
  for (const produit of produits) {  
    // Création de : a > article > img + h3 + p avec les valeurs dynamique de l'API  
    tousLesProduits.innerHTML +=  
    `<a href="./product.html?id=${produit._id}">  
      <article>  
          
        <h3 class="productName">${produit.name}</h3>  
        <p class="productDescription">${produit.description}</p>  
      </article>  
    </a>`  
  }  
}
```



KANAP SINOPE

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.



KANAP CYLLÈNE

Morbi nec erat aliquam, sagittis una non, laoreet justo. Etiam sit amet interdum diam, at accumsan lectus.



KANAP CALYCÉ

Pellentesque fermentum arcu venenatis ex sagittis accumsan. Vivamus lacinia fermentum tortor. Mauris imperdiet tellus.



KANAP AUTONOE

Donec mattis nisl tortor, nec blandit sapien fermentum at. Proin hendrerit efficitur fringilla. Lorem ipsum dolor sit...



KANAP EURYDOMÉ

Ut laoreet vulputate neque in commodo. Suspendisse maximus quis erat in sagittis. Donec hendrerit purus at congue aliquam.



KANAP HÉLICÉ

Curabitur vel augue sit amet arcu aliquet interdum. Integer vel quam mi. Morbi nec vehicula mi, sit amet vestibulum.



KANAP THYONÉ

EMauris imperdiet tellus ante, sit amet pretium turpis molestie eu. Vestibulum et egestas eros. Vestibulum non lacus orci.



KANAP ORTHOSIE

Mauris molestie laoreet finibus. Aenean scelerisque convallis lacus at dapibus. Morbi imperdiet enim metus rhoncus.

Product.

JS


```
// On récupère l'URL de la page avec URLSearchParams
const params = new URLSearchParams(window.location.search)
// On y ajoute ID du produit pour récupérer les informations de l'API
const id = params.get("id")
console.log("ID du produit sélectionné :", id)
```

URLSearchParams

ID du produit sélectionné :
107fb5b75607497b96722bda5b504926

/product.html?id=107fb5b75607497b96722bda5b504926

```
// On récupère les données de l'API via l'ID du produit sélectionné
fetch(`http://localhost:3000/api/products/${id}`)
// On obtient la réponse de l'API qu'on converti au format JSON
.then((res) => res.json())
// Les données JSON sont appelées "produit"
.then((produit) => {
  // On affiche les données du produit en console
  console.log("Données du produit sélectionné :", produit)
  // On execute la fonction qui affiche le produit
  afficherProduit(produit)
})
// Si une erreur se produit => remplace le contenu de la section "item"
// par un titre h1 avec erreur 404 et renvoi une erreur en console
.catch((err) => {
  document.querySelector(".item").innerHTML = "<h1>erreur 404<h1>"
  console.log("API => erreur 404 : " + err)
})
```

Fetch

```
Données du produit sélectionné : product.js:20
{colors: Array(3), _id: '107fb5b75607497b96722bda5b504926', nam
▼ e: 'Kanap Sinopé', price: 1849, imageUrl: 'http://localhost:300
0/images/kanap01.jpeg', ...} ⓘ
  altTxt: "Photo d'un canapé bleu, deux places"
  ▶ colors: (3) ['Blue', 'White', 'Black']
  description: "Excepteur sint occaecat cupidatat non proident,
  imageUrl: "http://localhost:3000/images/kanap01.jpeg"
  name: "Kanap Sinopé"
  price: 1849
  _id: "107fb5b75607497b96722bda5b504926"
```

```
// On appelle les fonctions créées avec en paramètre les données API à afficher
function afficherProduit(produit) {
  creationNomOnglet(produit.name)
  creationImage(produit.imageUrl, produit.altTxt)
  creationTitre(produit.name)
  creationPrix(produit.price)
  creationParagraphe(produit.description)
  creationCouleurs(produit.colors)
  creationPanier(produit)
}
```

Fonction d'affichage

🔍 Kanap Sinopé



```
▼ colors: Array(3)
  0: "Blue"
  1: "White"
  2: "Black"
```



Kanap Sinopé

Prix : 1849€

Description :

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Choisir une couleur :

«SVP, choisissez une couleur» ▼

Nombre d'article(s) (1-100) :

0

Ajouter au panier

```
function creationNomOnglet(nom) {
  const nomProduit = document.querySelector('title')
  nomProduit.textContent = nom
}

function creationImage(imageUrl, altTxt) {
  const divImg = document.querySelector(".item__img")
  const img = document.createElement("img")
  img.src = imageUrl
  img.alt = altTxt
  divImg.appendChild(img)
}

function creationTitre(nom) {
  const h1 = document.querySelector("#title")
  h1.textContent = nom
}

function creationPrix(prix) {
  const montant = document.querySelector("#price")
  montant.textContent = prix
}

function creationParagraphe(description) {
  const p = document.querySelector("#description")
  p.textContent = description
}

function creationCouleurs(couleurs) {
  const selectionner = document.querySelector("#colors")
  couleurs.forEach((couleur) => {
    const option = document.createElement("option")
    option.value = couleur
    option.textContent = couleur
    selectionner.appendChild(option)
  });
}
```

Création Panier

```
function creationPanier(produit) {  
  // On écoute la réponse du bouton au clique de l'utilisateur  
  const button = document.querySelector("#addToCart")  
  button.addEventListener("click", () => {  
    // On récupère les valeurs des Input de #colors et #quantity  
    const couleur = document.querySelector("#colors").value  
    const quantite = document.querySelector("#quantity").value  
    // Si l'achat est invalide on retourne l'information  
    if (AchatInvalide(couleur, quantite)) return  
    // Autrement on ajoute les informations sélectionnées par l'utilisateur  
    ajoutPanier(produit, quantite, couleur)  
    // On affiche un message pour rediriger l'utilisateur après avoir validé sa sélection  
    const message = `Le ${produit.name} avec la couleur ${couleur} a bien été ajouté au panier !  
    Consulter le Panier [OK] | Rester sur la page ${produit.name} [Annuler]`  
    confirmationAchat(message)  
  })  
  // On retourne les informations saisie dans le local storage  
  return (panier = JSON.parse(localStorage.getItem("Produit")))  
}
```

```
function AchatInvalide(couleur, quantite) {  
  // Si aucune couleur ET/OU quantité est sélectionné => alerte  
  if (couleur == 0 || quantite == 0) {  
    alert (" Veuillez choisir une couleur et une quantité")  
    return true  
  }  
  // Si une quantité < 0 est écrite OU Si une quantité > 100 est écrite => alerte  
  if (quantite <= 0 || quantite >= 100) {  
    alert ("Veuillez rentrer une valeur entre 1 et 100")  
    return true  
  }  
  return false  
}
```

127.0.0.1:5500 indique

Veuillez choisir une couleur et une quantité

OK

127.0.0.1:5500 indique

Veuillez rentrer une valeur entre 1 et 100

OK

Ajout au Panier

Key	Value
Produit	[{"id": "107fb5b75607497b96722bda5b504926", "color": "White", "quantity": 1}]

```
const confirmationAchat = (message = undefined) => {  
  // Si l'utilisateur clique sur OK => Redirigé sur la page panier  
  if (message && window.confirm(message)) {  
    window.location.href = "cart.html"  
  }  
  // Si l'utilisateur clique sur ANNULER => Reste sur la page du produit  
  else {  
    window.close  
  }  
}
```

127.0.0.1:5500 indique

Le Kanap Sinopé avec la couleur White a bien été ajouté au panier !
Consulter le Panier [OK] | Rester sur la page Kanap Sinopé [Annuler]

OK

Annuler

```
function ajoutPanier(produit, quantite, couleur) {  
  // Les informations sont stockés sous forme d'objet dans le local storage  
  const achat = {  
    id: produit._id,  
    color: couleur,  
    quantity: Number(quantite),  
  }  
  // On créer une variable panier avec comme clé "Produit" dans le local storage  
  let panier = JSON.parse(localStorage.getItem("Produit"))  
  // Si le panier est vide => création d'un tableau + push du 1er produit  
  if (panier == null) {  
    panier = []  
    panier.push(achat)  
    localStorage.setItem("Produit", JSON.stringify(panier))  
    // Confirme l'ajout au panier  
    confirmationAchat()  
  }  
  // Sinon le panier n'est pas vide => ...  
  } else {  
    // On fait une boucle sur les éléments panier  
    for (i = 0; i < panier.length; i++) {  
      // Si le produit est similaire (id/couleur) => Ajout de quantité  
      if (panier[i].id == produit._id && panier[i].color == couleur) {  
        return (  
          // Math.min renvoie la plus petite des 2 valeurs, pas plus de 100  
          panier[i].quantity = Math.min(panier[i].quantity + achat.quantity, 100),  
          localStorage.setItem("Produit", JSON.stringify(panier)),  
          // Confirme l'ajout au panier  
          confirmationAchat()  
        )  
      }  
    }  
    // On fait une boucle sur les éléments panier  
    for (i = 0; i < panier.length; i++) {  
      // Si le produit à (un ID similaire ET une couleur différente) OU un ID différent  
      if (panier[i].id == produit._id && panier[i].color != couleur || panier[i].id != produit._id) {  
        return (  
          // Ajouter un nouveau produit dans le panier  
          panier.push(achat),  
          localStorage.setItem("Produit", JSON.stringify(panier)),  
          // Confirme l'ajout au panier  
          confirmationAchat()  
        )  
      }  
    }  
  }  
}
```

Cart. JS

Fetch

```
fetch("http://localhost:3000/api/products/")
// On obtient la réponse de l'API qu'on converti au format JSON
.then((res) => res.json())
// Les données JSON sont appelées "produits"
.then((produits) => {
  // On exécute la fonction qui fait le liens entre le local storage et l'API
  recupererDonneesAchat(produits)
})
.catch((err) => {
  // Si une erreur se produit => remplace le contenu du titre h1
  // avec "erreur 404" et renvoi une erreur en console
  document.querySelector("#cartAndFormContainer").innerHTML = "<h1>erreur 404<h1>"
  console.log("API => erreur 404 : " + err)
})
```

```

function recupererDonneesAchat(produits) {
  // On récupère les informations (id, nom, couleur) du/des produit(s) ajouté(s) dans notre panier
  const panier = JSON.parse(localStorage.getItem("Produit"))
  // Si le panier est vide => Renvoi le message "Votre panier est vide"
  if (panier === null || panier.length == 0 ) {
    document.querySelector("#cartAndFormContainer").innerHTML = "<h1>Votre panier est vide</h1>"
  } else if (panier != null) {
    // On fait une boucle sur tous les achats de notre panier =>
    for (let achat of panier) {
      // On fait une boucle sur toutes les données d'un produit de l'API =>
      for (let a = 0, b = produits.length; a < b; a++) {
        // Si id de l'achat de notre panier est identique à l'id du produit de l'API
        if (achat.id === produits[a]._id) {
          // On récupère les données de chaque éléments dont on a besoin
          achat.name = produits[a].name
          achat.price = produits[a].price
          achat.imageUrl = produits[a].imageUrl
          achat.altTxt = produits[a].altTxt
        }
      }
    }
    // On affiche les achats du panier en console
    console.log("Panier : ", panier)
    // On appel notre fonction pour afficher les produits du panier
    afficherProduit(panier)
    // On appel nos fonctions pour modifier et supprimer la quantité des articles
    modifierQuantite()
    supprimerProduit()
  }
}

```

Récupération données

```

Panier :  ▼[{}]
```

- ▼ 0:
 - altTxt: "Photo d'un canapé bleu, deux places"
 - color: "Blue"
 - id: "107fb5b75607497b96722bda5b504926"
 - imageUrl: "http://localhost:3000/images/kanap01.jpeg"
 - name: "Kanap Sinopé"
 - price: 1849
 - quantity: 1

```
function afficherProduit(panier) {
  // On déclare et on cible sur quel section du HTML on veut afficher les produits du panier
  const produitPanier = document.querySelector("#cart__items");
  // On créer un HTML dynamique avec la méthode .map() + introduction de data-set
  produitPanier.innerHTML += panier.map((achat) =>
    `<article class="cart__item" data-id="${achat.id}" data-color="${achat.color}" data-quantity="${achat.quantity}" data-price="${achat.price}">
      <div class="cart__item_img">
        
      </div>
      <div class="cart__item_content">
        <div class="cart__item_content_description">
          <h2>${achat.name}</h2>
          <span>Couleur : ${achat.color}</span>
          <p data-price="${achat.price}">Prix : ${achat.price} €</p>
        </div>
        <div class="cart__item_content_settings">
          <div class="cart__item_content_settings_quantity">
            <p>Quantité : </p>
            <input type="number" class="itemQuantity" name="itemQuantity" min="1" max="100" value="${achat.quantity}">
          </div>
          <div class="cart__item_content_settings_delete">
            <p class="deleteItem" data-id="${achat.id}" data-color="${achat.color}">Supprimer</p>
          </div>
        </div>
      </div>
    </article>`
  ).join("")
  // On appel la fonction pour afficher les totaux (Quantité + Prix article)
  totauxPanier()
}
```

```
function totauxPanier() {
  // On déclare des variables de "Total" en tant que nombre
  let totalProduits = 0
  let totalPrix = 0
  // On déclare et on cible toutes les classes ".cart__item"
  const achats = document.querySelectorAll(".cart__item")
  // Pour chaque élément (achat) des achats
  achats.forEach((achat) => {
    // On récupère les quantités des produits via les dataset
    totalProduits += JSON.parse(Math.min(achat.dataset.quantity, 100))
    // On calcul le prix panier total via les dataset
    totalPrix += Math.min(achat.dataset.quantity, 100) * achat.dataset.price
  });
  // On affiche les résultats dans le HTML
  document.getElementById("totalQuantity").textContent = totalProduits
  document.getElementById("totalPrice").textContent = totalPrix
}
```

Afficher produit du panier

Modifier et supprimer quantité

```
function modifierQuantite() {
  // On déclare et on cible tous les éléments avec la classe ".cart_item"
  const produitPanier = document.querySelectorAll(".cart_item");
  // De façon dynamique grâce au dataset =>
  // On écoute les événements (eQuantite) sur chaque input d'article concerné
  produitPanier.forEach((achat) => {
    achat.addEventListener("change", (eQuantite) => {
      // On appelle les ressources de notre "panier" (local storage)
      const panier = JSON.parse(localStorage.getItem("Produit"))
      // On fait une boucle sur les produits du Panier
      for (produit of panier) {
        if (
          // Si ( id + color ) sont similaires
          produit.id === achat.dataset.id &&
          achat.dataset.color === produit.color
        ) {
          // On modifie la quantité qui ne peut pas dépasser 100
          produit.quantity = Math.min(eQuantite.target.value, 100)
          // On envoie les nouvelles données au local storage
          localStorage.Produit = JSON.stringify(panier)
          // On fait une mise à jour du dataset quantity
          achat.dataset.quantity = eQuantite.target.value
          // On appelle la fonction des "totauxPanier" dynamiquement
          totauxPanier();
        }
      }
    })
  })
}
```

```
function supprimerProduit() {
  // On déclare et on cible tous les éléments ".cart_item .deleteItem"
  const supprimerProduit = document.querySelectorAll(".cart_item .deleteItem")
  // Pour chaque produit
  supprimerProduit.forEach((produit) => {
    // On écoute au click (bouton "Supprimer") sur le produit concerné
    produit.addEventListener("click", () => {
      // On appelle les ressources de notre "panier" (local storage)
      const panier = JSON.parse(localStorage.getItem("Produit"))
      // On fait une boucle pour chaque élément du panier
      for (let a = 0, b = panier.length; a < b; a++) {
        if (
          // Si les éléments du panier correspondent aux données dataset ( id + color )
          panier[a].id === produit.dataset.id &&
          panier[a].color === produit.dataset.color
        ) {
          // On déclare une variable utile pour la suppression
          const num = [a]
          // On supprime 1 élément à l'indice num
          panier.splice(num, 1)
          // On renvoie un nouveau panier converti dans le Local Storage
          localStorage.Produit = JSON.stringify(panier)
          // On supprime l'affichage HTML
          const displayToDelete = document.querySelector(
            `article[data-id="${produit.dataset.id}"][data-color="${produit.dataset.color}"]`
          )
          displayToDelete.remove()
          // Si le Panier est vide
          if (panier === null || panier.length == 0) {
            // On vide le local storage
            window.localStorage.clear()
            // On affiche l'information "Votre panier est vide"
            document.querySelector("#cartAndFormContainer").innerHTML = "<h1>Votre panier est vide</h1>"
          }
        }
      }
    })
  })
}
```

Ecoute le formulaire

```
function ecouteFormulaire() {  
  // On cible le bouton "Commander !"   
  const boutonCommander = document.querySelector("#order")  
  // On écoute le bouton et au click => On appelle la fonction "envoiFormulaire"  
  boutonCommander.addEventListener("click", (e) => envoiFormulaire(e))  
  
  // On cible l'input de la saisie  
  let prenomFormulaire = document.getElementById('firstName')  
  // On écoute l'événement à l'input avec la fonction qui vérifie la validité de la saisie  
  prenomFormulaire.addEventListener('input', validationPrenom)  
  
  let nomFormulaire = document.getElementById('lastName')  
  nomFormulaire.addEventListener('input', validationNom)  
  
  let adresseFormulaire = document.getElementById('address')  
  adresseFormulaire.addEventListener('input', validationAdresse)  
  
  let villeFormulaire = document.getElementById('city')  
  villeFormulaire.addEventListener('input', validationVille)  
  
  let emailFormulaire = document.getElementById('email')  
  emailFormulaire.addEventListener('input', validationEmail)  
}
```

Envoi du formulaire

```
function envoiFormulaire(e) {  
  // On récupère les données du client  
  const client = donneeClient(e)  
  // Si les informations sont manquantes, on retourne sans envoyer  
  if (client == null) return  
  // On envoie l'objet "client" qui contient toutes les données à l'API pour obtenir ID de commande  
  fetch("http://localhost:3000/api/products/order", {  
    method: "POST",  
    headers: {  
      'Accept': 'application/json',  
      "Content-Type": "application/json",  
    },  
    body: JSON.stringify(client),  
  })  
  .then(res => res.json())  
  .then(res => {  
    alert("Votre commande a bien été effectuée !")  
    window.location.replace(`./confirmation.html?orderId=${res.orderId}`)  
  })  
  .catch((err) => {  
    alert (err.message)  
    console.log(err)  
  })  
}
```

127.0.0.1:5500 indique

Votre commande a bien été effectuée !

OK

```
function donneeClient(e) {
  // Si tous les éléments du formulaire sont valide, on récupère les données du client
  if (validationPrenom() && validationNom() && validationAdresse() && validationVille() && validationEmail()) {
    const panier = JSON.parse(localStorage.getItem("Produit"));
    const prenom = document.getElementById("firstName").value
    const nom = document.getElementById("lastName").value
    const adresse = document.getElementById("address").value
    const ville = document.getElementById("city").value
    const email = document.getElementById("email").value
    // On récupère les ID des produits du client
    const produitsIds = []
    panier.forEach((produit) => {
      produitsIds.push(produit.id)
    })
    // On créer l'objet pour répondre aux attentes de l'API
    const informationsClient = {
      contact: {
        firstName: prenom,
        lastName: nom,
        address: adresse,
        city: ville,
        email: email
      },
      products: produitsIds
    }
    return informationsClient
  }
  // Sinon on retourne une alerte
} else {
  alert("Le formulaire n'est pas correctement rempli, veuillez réessayer.")
  e.preventDefault()
}
}
```

Données client

127.0.0.1:5500 indique

Le formulaire n'est pas correctement rempli, veuillez réessayer.

OK

Regex Input form

```
function validationPrenom() {  
  let prenomInput = document.getElementById("firstName")  
  let prenomValidation = document.getElementById("firstName").value  
  let prenomMsgErreur = document.getElementById("firstNameErrorMsg")  
  let commande = document.getElementById("order")  
  
  const prenomReGex = /^[a-zA-Zâäåãäçêëëëïïïïñóôöõöüüüýÿæ\ś-]{3,24}$/i  
  let prenomResultat = prenomReGex.test(prenomValidation)  
  
  if (prenomResultat == false ) {  
    prenomInput.style.backgroundColor = "red"  
    prenomInput.style.color = "white"  
    prenomMsgErreur.innerHTML = "Votre prénom doit comporter entre 3 et 24 lettres"  
    prenomMsgErreur.style.display = "inherit"  
    commande.disabled = true  
    return false  
  } else {  
    prenomMsgErreur.style.display = "none"  
    prenomInput.style.backgroundColor = "green"  
    prenomInput.style.color = "white"  
    commande.disabled = false  
    return true  
  }  
}
```

Prénom:

123

Votre prénom doit comporter
entre 3 et 24 lettres

Prénom:

Valentin

Confirmation.

JS

Order ID

```
/*----- Appels les fonctions -----  
orderId()  
cacheLocalStorageVide()  
/*----- Récupérer et afficher orderId -----  
function orderId () {  
    let orderNumber = document.getElementById("orderId")  
    let baseUrl = (window.location).href  
    orderNumber.innerHTML = baseUrl.substring(baseUrl.lastIndexOf('=') + 1)  
    console.log("Numéro de commande :", orderNumber);  
}  
/*----- Supprimer fichier panier -----  
function cacheLocalStorageVide () {  
    const localStorage = window.localStorage  
    localStorage.clear()  
}
```

Commande validée !
Votre numéro de commande est :
ecfe5800-15bc-11ed-bcd3-fb6f56bcaf3f

Numéro de commande : confirmation.
ecfe5800-15bc-11ed-bcd3-fb6f56bcaf3f

Plan de test



Page	Fonctionnalité	Action	Résultat attendu	Résultat observé
Accueil	Une page d'accueil montrant (de manière dynamique) tous les articles disponibles à la vente.	Ouvrir sur la page d'accueil du site web dans un navigateur	Affichage de l'ensemble des produits avec leur image, nom et une partie de description	OK
Accueil	Informations complémentaires d'un produit de la page d'accueil	Cliquer sur l'un des produits proposé sur la page d'accueil	Redirection sur la page du produit sélectionné	OK
Produit	Une page produit montrant toutes les informations du produit sélectionné	Visualiser l'image, le nom, le prix, la description du produit ainsi que les différentes couleurs qu'il propose	Affichage de toutes les informations du produit	OK
Produit	Choix entre plusieurs couleurs pour un produit ainsi qu'une quantité	Choisir une des couleurs proposées et une quantité entre 1 et 100 unités	Si une couleur et une quantité valide ont été renseignés, le produit est ajouté au panier	OK
Produit	Choix entre plusieurs couleurs pour un produit ainsi qu'une quantité	Choisir une quantité négative ou supérieur à 100	Si la quantité n'est pas comprise entre 1 et 100, un message d'erreur apparaît	OK
Produit	Choix entre plusieurs couleurs pour un produit ainsi qu'une quantité	Choisir seulement une couleur ou une quantité ou aucune des deux données	Si aucune ou une seule des deux données a été renseigné, on renvoi une erreur et rien n'est ajouté au panier	OK
Produit	Ajouter une nouvelle quantité d'un produit déjà dans le panier qui a la même couleur	Si on rajoute une quantité d'un produit du panier qui a la même couleur. La valeur ajoutée sera additionnée, mais ne pourra pas dépasser 100 unités	Additionne l'ajout d'une quantité d'un produit avec la quantité du même produit du panier qui a la même couleur	OK
Produit	Possibilité d'ajouter au panier un produit	Cliquer sur "ajouter au panier" pour que le produit se retrouve dans le panier	Avoir le produit dans la page panier	OK
Produit	Redirection après avoir validé un produit	Quand on ajout un article au panier, on est soi dirigé au panier, soit on reste sur la page du produit	Redirection après avoir confirmé l'ajout d'un article au panier	OK
Panier	Une page panier montrant (de manière dynamique) tous les articles ajoutés	Ouvrir sur la page panier du site web dans un navigateur	Affichage de l'ensemble des produits ajoutés au panier	OK

Panier	Produit similaire avec une couleur différente	On peut voir plusieurs produits similaire avec une couleur qui les différencies	Visualiser la différence de couleur pour des produits similaires au panier	OK
Panier	Possibilité de changer la quantité	Utiliser les flèches bas/haut ou entrer une valeur manuellement entre 1 et 100	Nouvelle valeur sauvegardée (Les valeurs ne pourront pas dépasser 100 unités)	OK
Panier	Possibilité de supprimer un produit du panier	Cliquer sur le bouton supprimé	Article supprimé du panier	OK
Panier	Totaux (nombres d'article + prix) affichés de façon dynamique	Changer la valeur de quantité d'un ou des produit(s) du panier	Visualiser le prix et la quantité total des produits du panier qui varie quand on change les valeurs de quantité	OK
Panier	Saisi d'un formulaire correct	Renseigner les données demandées pour valider une commande	Formulaire rempli correctement	OK
Panier	Saisi d'un formulaire incorrect	Si le formulaire n'est pas correctement rempli, il ne pourra pas être envoyé. Les erreurs sont visibles et une solution est écrite pour guider l'utilisateur	Formulaire rempli incorrectement	OK
Panier	Valider la commande	Si le formulaire est correctement rempli, au clique sur le bouton "Commander!" l'utilisateur est dirigé sur la page avec un numéro de commande unique	Redirection sur la page confirmation	OK
Confirmation	Une page confirmation montrant (de manière dynamique) le numéro de commande.	Après avoir correctement rempli le formulaire, l'utilisateur est dirigé sur une page qui indique que la commande a bien été passé avec un numéro de commande attribué	Page confirmation avec un numéro de commande unique	OK
Confirmation	Panier vide	Après avoir validé la commande en ayant eu le numéro de commande. Le panier devient vide	La panier est vide	OK

Questions ?