

Vue Gadget Shop - step8-solution

Création du composant CartDisplay.js

1. Création du fichier components/CartDisplay.js

Ajoute un nouveau fichier dans /components :

 components/CartDisplay.js

```
app.component("cart-display", {
  props: ["cart", "cartTotal", "discountedTotal"], // Le panier et le total sont
  // passés en props
  template: `
    <div class="cart">
      <h2>🛒 Mon Panier ({{ cart.length }})</h2>
      <ul v-if="cart.length > 0">
        <li v-for="(item, index) in cart" :key="index">
          {{ item.name }} - {{ item.price }}€
          <button class="remove-btn" @click="removeItem(index)"> ✕ </button>
        </li>
      </ul>
      <p v-else>Votre panier est vide.</p>

      <!-- Affichage du total -->
      <h3 v-if="cart.length > 0">
        💰 Total :
        <span v-if="cartTotal < 1000">{{ cartTotal }}€</span>
        <span v-else>
          <s>{{ cartTotal }}€</s> →
          <strong>{{ discountedTotal }}€ (-10%)</strong>
        </span>
      </h3>
    </div>
  `,
  methods: {
    removeItem(index) {
      this.$emit("remove-from-cart", index); // Émet un événement pour supprimer
      // du panier
    },
  },
});
```

Explication

- props: ["cart", "cartTotal", "discountedTotal"] → Le panier et les totaux sont passés par le parent (main.js).
- v-for="(item, index) in cart" → Affiche chaque article dans une liste.
- @click="removeItem(index)" → Émet un événement pour supprimer un article du panier.

2. Mise à jour de index.html pour utiliser CartDisplay.js

Remplace la section panier par .

 index.html


```
<cart-display
:cart="cart"
:cart-total="cartTotal"
:discounted-total="discountedTotal"
@remove-from-cart="removeFromCart">
</cart-display>

...

<script src="./main.js"></script>


<script src="./components/GadgetDisplay.js"></script>
<script src="./components/CartDisplay.js"></script>

<!-- Mount App -->
<script>
  const mountedApp = app.mount("#app");
</script>
```

 Pourquoi c'est mieux ?

✅ Le code est plus propre : la logique du panier est séparée dans CartDisplay.js. ✅ Réutilisable : si on veut afficher le panier ailleurs, on n'a plus à dupliquer le code.

✅ Résultat attendu

-  Le panier est maintenant un composant séparé !
- CartDisplay.js affiche et gère le panier.
- GadgetDisplay.js affiche les produits et permet d'ajouter au panier.
- Le parent (main.js) passe les données et écoute les événements.