

The background features abstract green geometric shapes. On the left, a solid green triangle points downwards. On the right, a complex composition of overlapping, semi-transparent green triangles and polygons in various shades of green (from light lime to dark forest green) creates a layered effect. A thin, light gray line extends from the bottom left towards the right, passing through the green shapes.

Exercise 1

Objectifs

- ▶ Exercice pour utiliser tout ce qu'on a vu
- ▶ [Lien vers Exercice](#)

Indication 1

- Utiliser de chatgpt pour générer les données

Indication 2

- ▶ Créer d'abord la version html du composant avant de mettre les codes JS
 - ▶ Tableau en html
 - ▶ CSS si besoin

Indication 3

- Calcul du montant

```
{{ product.unit_price * product.quantity }} €
```

- En utilisant une méthode

```
{{ getTotalProduct(product) }} €
```

```
methods: {  
  ⚡ getTotalProduct(product) {  
    return product.unit_price * product.quantity;  
  }  
}
```

Indication 4

- ▶ On a besoin de faire le total du montant pour plus tard
- ▶ En utilisant une computed, on peut faire

```
computed: {  
  productsTotal() {  
    let data = {};  
    this.cart_products.forEach((product) => {  
      data[product.id] = product.unit_price * product.quantity;  
    })  
    return data;  
  },  
}
```

- ▶ Différence
 - ▶ Let : portée locale
 - ▶ Var : portée globale
 - ▶ Const : constante, ne peut plus être modifié

```
{{ productsTotal[product.id] }} €
```

Indication 5

- Créer une computed pour le total

```
totalExclTaxes() {  
  let total = 0;  
  Object.values(this.productsTotal).forEach((product_total) => {  
    total += parseFloat(product_total);  
  })  
  return total;  
}
```

```
<tfoot>  
  <tr>  
    <td colspan="3"></td>  
    <td>Total HT</td>  
    <td>{{ totalExclTaxes }} €</td>  
  </tr>  
</tfoot>
```

Indication 6

► Calcul TVA

```
totalTaxes() {  
    return this.totalExclTaxes*20/100;  
}
```

```
<tfoot>  
  <tr>  
    <td colspan="3"></td>  
    <td>Total HT</td>  
    <td>{{ totalExclTaxes }} €</td>  
  </tr>  
  <tr>  
    <td colspan="3"></td>  
    <td>Total TVA</td>  
    <td>{{ totalTaxes }} €</td>  
  </tr>  
</tfoot>
```

Indication 7

- ▶ Total TTC
 - ▶ Total hors taxe + Tva

Indication 8

- Créer une computed qui return true si le total TTC est supérieur à 100

```
isCartSup100() {  
  return this.totalInclTaxes >= 100;  
}
```

- Mettre le total TTC en vert si le total est supérieur à 100

```
:class="{ 'bg-green-500': isCartSup100 }"
```

- Tester avec Vue Tools dans le navigateur

Indication 9

- Créer une computed qui retourne la différence par rapport à 100, ou null sinon

```
lackForShippingCosts() {  
  return !this.isCartSup100 ? 100 - this.totalInclTaxes : null  
}
```

- Afficher le message

```
<div v-if="!isCartSup100">  
  <p class="text-sm bg-indigo-400">Plus que {{ lackForShippingCosts }} € pour avoir les frais de port offert</p>  
</div>
```

Indication 10

- Arrondir à 2 chiffres

```
methods: {  
  parsePrice(price_to_parse) {  
    return parseFloat(price_to_parse).toFixed(fractionDigits: 2);  
  }  
},
```