

1. Map, Filter & Reduce

02 May 2024 15:36

How does Map function work in react for rendering List?

1. It allows us to create a new array of react elements based on the original array

```
const products = [
  {id: 1, name: "Product A", price: 20, category: "Electronics"},
  {id: 2, name: "Product B", price: 30, category: "Clothing"},
  {id: 3, name: "Product C", price: 15, category: "Electronics"},
  {id: 4, name: "Product D", price: 25, category: "Clothing"},
  {id: 5, name: "Product E", price: 50, category: "Electronics"},
  {id: 6, name: "Product F", price: 40, category: "Electronics"},
];
```

Example this JSON data

```
<ul>
  {products.map((product) => {
    return (
      <li key={product.id}>
        <strong>{product.name}</strong> - ${product.price} - Category:{ " "}
        {product.category}
      </li>
    );
  })}
</ul>
```

I you use () brackets instead of {} u don't need to use return

Product A - 20 - Category: Electronics
Product B - 30 - Category: Clothing
Product C - 15 - Category: Electronics
Product D - 25 - Category: Clothing
Product E - 50 - Category: Electronics
Product F - 40 - Category: Electronics

Op

2. Follow Up question :
 > Filter based on Electronics Category

```
<h5>Question 2: How can you filter products with a specific category?</h5>
<ul>
  {products.filter((product) => product.category === "Electronics").map((product) => {
    return (
      <li key={product.id}>
        <strong>{product.name}</strong> - ${product.price} - Category:{ " "}
        {product.category}
      </li>
    );
  })}
</ul>
```

Question 2: How can you filter products with a spec

Product A - \$20 - Category: Elect
Product C - \$15 - Category: Elect
Product E - \$50 - Category: Elect
Product F - \$40 - Category: Elect

3. Render a summary of total prices for products

```
<div>
  <p>
    Total Price Summary: ${ " "}
    {products.reduce((acc, product) => {
      return acc + product.price;
    }, 0)}
  </p>
</div>
```

Question 4: Add **discountedPrice** key with 10% discount to all the products with price more than \$20 and render it.

```
</h5>
<ul>
  {products
    .filter((product) => product.price > 20)
```

```
.map((product) => {  
  return {  
    ...product,  
    discountedPrice: product.price - product.price * (10 / 100),  
  };  
})  
.map((product) => {  
  return (  
    <li key={product.id}>  
      <strong>{product.name}</strong> - ${product.discountedPrice} -  
      Discounted from: {product.price}  
    </li>  
  );  
});  
}}}  
</ul>
```

4.