useContext Exercises

setup

challenge

https://codesandbox.io/s/cart-product-questions-zzi2ln

Let's create a React app with Two components: Cart and ProductListing.

1. Pages are already created. Add the routes first, like we practiced yesterday.
    1. Routes
    2. Header
    3. NavLink in the Header

```
<h1>meKart</h1>
      <h3> items in cart: 2 </h3>
      <nav>
        <NavLink to="/"> home || </NavLink>
        <NavLink to="/cart"> cart </NavLink>
      </nav>
      <Routes>
        <Route path="/" element={<ProductList products={products} />} />
        <Route path="/cart" element={<Cart />} />
      </Routes>
```

COPY

1. Add a router to `index.js`

solution

https://codesandbox.io/s/cart-product-solution-bcliio

create and provide context

Let’s learn by creating a cart context:

In the React app we created above, create an `index.js` file and add context using the `createContext()` from react.

```
// index.js
import { createContext } from 'react' // importing createContext from react
```

```
export const CartContext = createContext() // Creating context
```

COPY

Now in your root file (in our case `index.js`), provide the context to the

`<App />` component. Provide `{cartItems: 4}` as value with your provider.

```
// index.js
<CartContext.Provider value={{ items: 4 }}>
  <App />
</CartContext.Provider>
```

COPY

solution

https://codesandbox.io/s/create-context-tqwvog

consume context

Now that we have context ready and has been provided to `<App />` component, let's consume it in the Cart Component.

We don't need to pass things from App to Cart anymore. You can directly use the ‘cartItems’ value in the Cart component without passing it as props to `<Cart />`.

You can also do this for Header component

```
// syntax for consuming context
const { cartItems } = useContext(CartContext)
```

COPY

solution

https://codesandbox.io/s/consume-context-solution-krxjp2

passing a function

Create a simple function logger which does `console.log()` and nothing else. Can we pass this function in the cart? If yes, do that and consume it in Cart.

```
const cartLogger = () => console.log(`carting`)
```

COPY

solution

https://codesandbox.io/s/logger-in-context-solution-ufh9di

moving the provider to a separate file

- You can move your cartProvider values into `cartProvider.js` for a clean code in your root file.
- Now, we get to learn `{children}` syntax
- There’s another syntax for exporting from `index.js` as well.

solution

https://codesandbox.io/s/moving-context-solution-ktbloj

useState for the item

Right now we are passing cartItem value directly as 4. Let's change that value from a direct value to be something coming from state.

Now, instead of logger, you can pass setItem. You are basically passing `{ cartItems, setCartItems }` pair.

solution

https://codesandbox.io/s/usestate-in-usecontext-s32egl