A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is a light green. They are positioned diagonally, with the blue one partially covering the green one.

Statenanagement t, Context API 2



Agenda

- Repetition
 - Children
 - Prop drilling
 - Context API
- Statemanagement med Context API
- Övningar



Children

Children prop i komponenter

```
// our button
const Button = ({ color }) => (
  <button className={color}>
    I am a button
  </button>
)

// somewhere else
<Button color="red" />
```



```
// our button
const Button = ({ color, children }) => (
  <button className={color}>
    {children}
  </button>
)

// somewhere else
<Button color="red">
  I am still a button
</Button>
```



Children

Övning:

- Skapa en komponent Container som renderar en `<div>`
 - Inuti `<div>{ children }</div>`
- Använd Container
 - För att rendera inuti div en ``
 - För att rendera inuti div en `<p>`
 - För att rendera inuti div en text "Hello World"

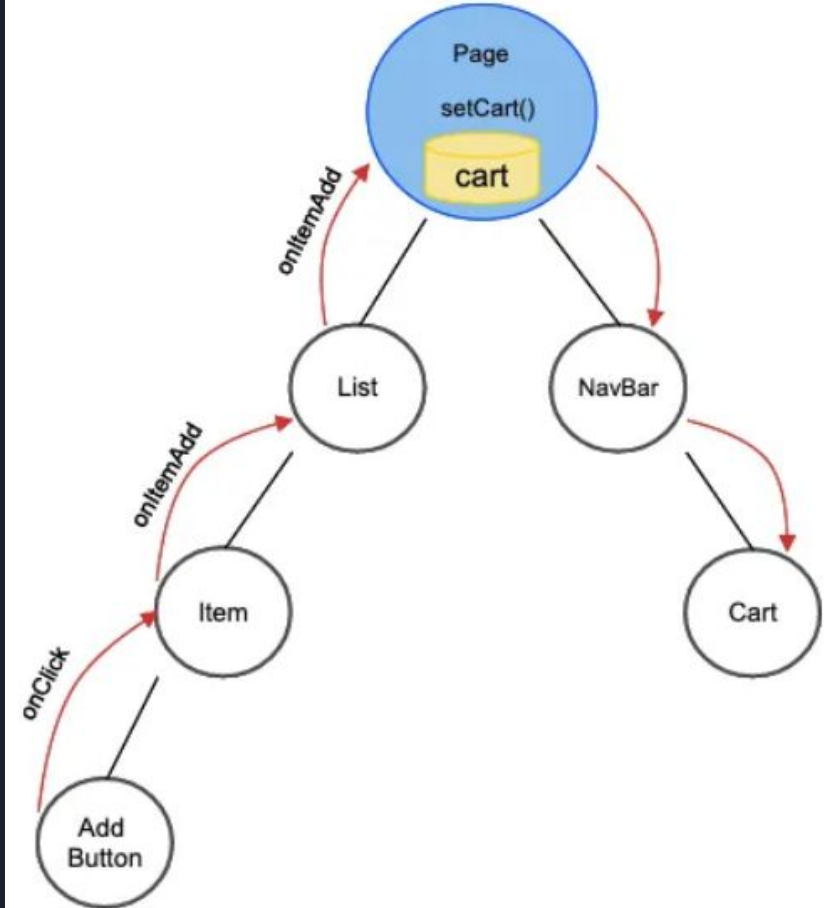
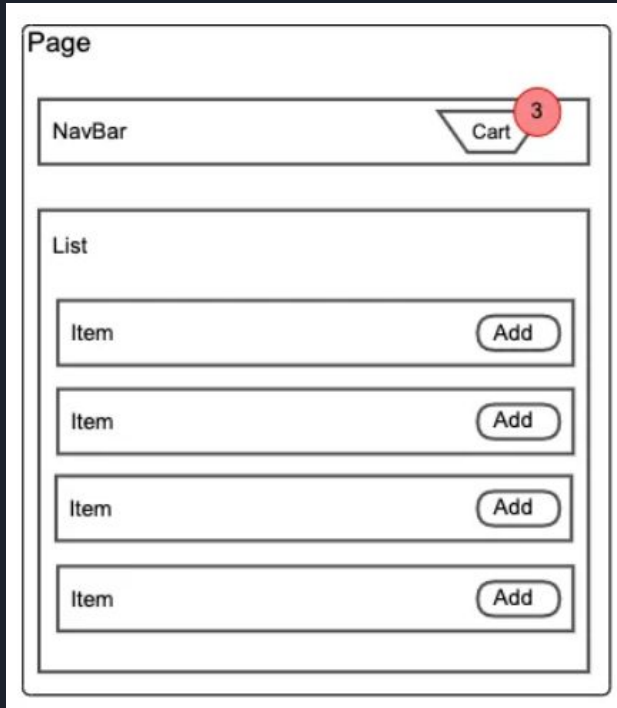


Prop drilling

Demo på prop drilling

<https://codesandbox.io/p/sandbox/compassionate-waterfall-4nn9fz?file=%2Fsrc%2FApp.js%3A51%2C1>

Prop drilling





Prop drilling

Demo av app med prop drilling

<https://codesandbox.io/p/sandbox/react-statefull-stateless-forked-yd2nr5?file=%2Fsrc%2Findex.js>



Context API

Steg för att skapa och använda Context API

1. Skapa ett “Context”
2. Skapa en “Provider” med hjälp av ditt context
3. Konsumera ditt context varifrån som helst i appen



Context API

1. Skapa ett "Context"

```
import { createContext } from "react";  
  
const MyContext = createContext("defaultValue");
```



Context API

2. Skapa en "Provider" med hjälp av ditt context

```
const MyContext = createContext("defaultValue");  
  
<MyContext.Provider value="Hello, World!">  
  | <YourApp />  
</MyContext.Provider>;
```



Context API

3. Konsumera ditt context varifrån som helst i appen

```
import { useContext } from "react";  
import { MyContext } from "../App"  
  
const value = useContext(MyContext);
```



Context API

Tillsammans använd Context API

- Skapa en `MessageContext` för att skicka en text
- Skapa en `FunctionContext` för att skicka en funktion
- Skapa en `StateContext` för att skicka state, och `setState`



Context API

Övning

- Skapa en fil (`TextContext.js`)
 - `const TextContext = createContext()`
 - exportera `TextContext` och `Provider`
 - `export default TextContext`
- Tillför en `Provider` runt `App`
 - `import TextContext from "../TextContext.js"`
 - `<TextContext.Provider value="test"> ... </TextContext.Provider>`
- I någon barnkomponent konsumera värde
 - `import TextContext from "../TextContext.js"`
 - `const text = useContext(TextContext)`



Context API

Demo av app med Contex API

<https://codesandbox.io/p/sandbox/compassionate-waterfall-4nn9fz?file=%2Fsrc%2FApp.js%3A51%2C1>



Statenmanagement

Övning

- Öppna <https://codesandbox.io/p/sandbox/compassionate-waterfall-forked-4pigkl?file=%2Fsrc%2FApp.js%3A30%2C1>
- I Profile.js ändra app så att vi använder 1 Context istället
 - Lägg in både profile och setProfile i objekt och skicka in i ProfileContext value
- I App.js dekonstruera objekt och plocka ut respektive del
 - `const { profile } = useContext(ProfileContext)`
 - `const { setProfile } = useContext(ProfileContext)`



Statenmanagement

Övning

- Skapa en TodosProvider.js på samma sätt som föregående övning
 - Importera createContext
 - Skapa context (TodosContext)
 - Skapa en komponent TodosProvider som tar in children prop
 - Skapa ett state ([todos, setTodos]) = []
 - Runt children lägg in <TodosContext.Provider> med både todos och setTodos i objekt
 - Exportera TodosProvider och TodosContext
- I main.js
 - Importera TodosProvider och lägg App runt TodosProvider
 - <TodosProvider><App/></TodosProvider>
- I App.js skapa 2 barn komponenter
 - AddTodo (formulär med knapp och textfält för att skapa todo)
 - importera TodosContext för att hämta setTodos
 - TodosList (ul lista som visar alla todos)
 - importera TodosContext för att hämta todos