

Bölüm
2

React Temelleri

Kurulumuz React Projesi Geliştirme

React Nedir?

React bir Javascript kütüphanesidir.

1

React CDN

```
<script src="https://unpkg.com/react@18/umd/react.development.js"></script>  
<script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>
```

2

React Projesi

```
create-react-app
```

```
vite
```

React Nedir?

React bir Javascript kütüphanesidir.

React ile kolaylıkla bir dinamik görüntüler (view) oluşturabiliriz.

Pure Javascript

```
function App() {  
    return "<h1>BTK Akademi</h1>";  
}
```

React

```
function App() {  
    return (<h1>BTK Akademi</h1>);  
}
```

JSX - Javascript XML

Pure Javascript

```
function App() {  
    const kurum = "BTK Akademi";  
    return `

# ${kurum}</h1>`; }


```

React

```
function App() {  
    const kurum = "BTK Akademi";  
    return (<h1>{kurum}</h1>);  
}
```

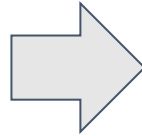
```
function List() {
```

```
  const people = [ 'Ali', 'Ayşe', 'Ahmet' ];
```

```
  const listItems = people.map(person =>  
    <li>{person}</li>  
  );
```

```
  return <ul>{listItems}</ul>;
```

```
}
```



```
<ul>
```

```
  <li>Ali</li>
```

```
  <li>Ayşe</li>
```

```
  <li>Ahmet</li>
```

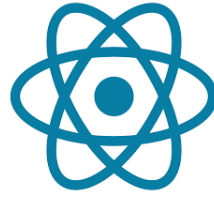
```
</ul>
```

React Nedir?

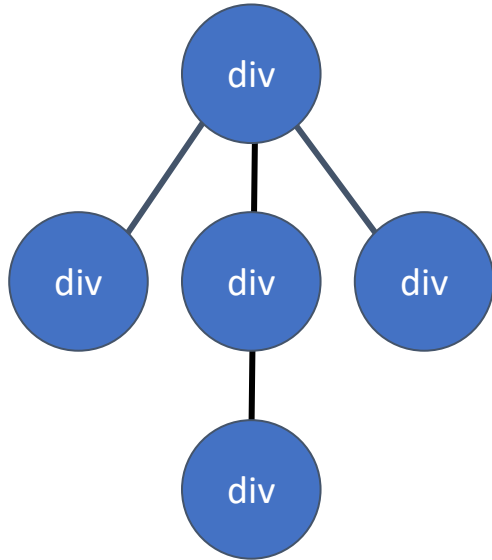
React bir Javascript kütüphanesidir.

React ile kolaylıkla bir dinamik görüntüler (view) oluşturabiliriz.

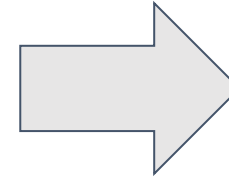
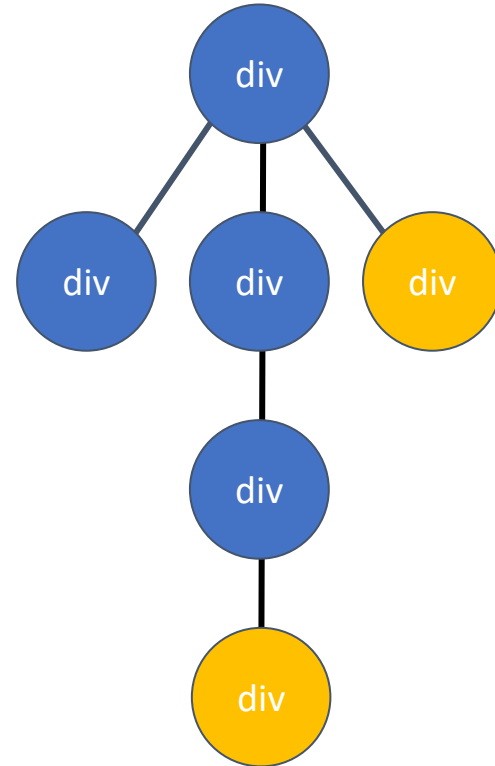
Virtual Dom kullanır.



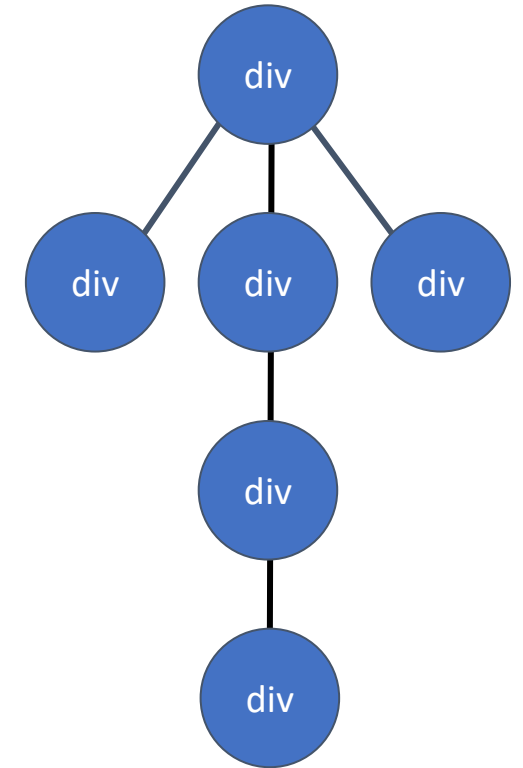
Virtual Dom



Update



Render to DOM



React Kullanmanın Avantajları

- 1 **Düzenli bir proje yapısı**
- 2 **Kolay kullanım**
- 3 **Performans**

Bölüm
2

React Temelleri

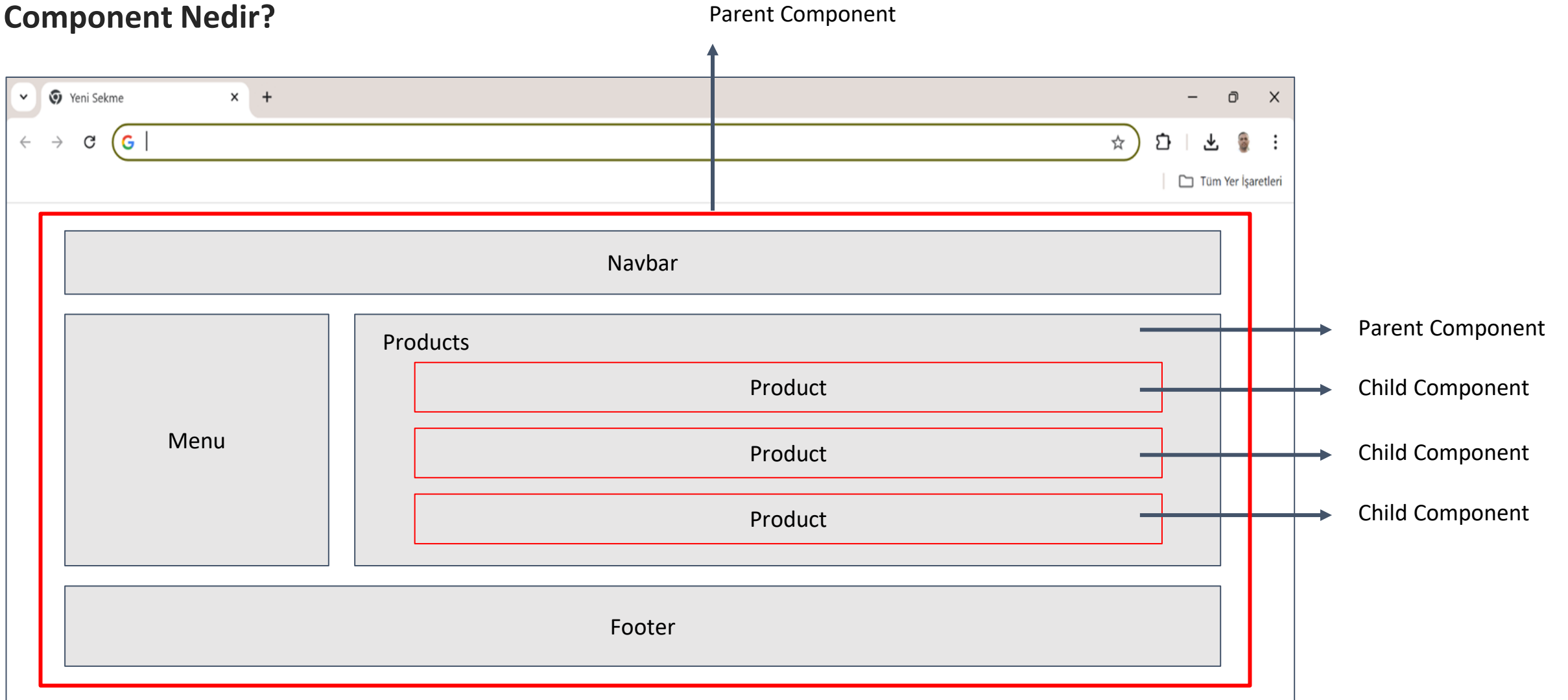
Projemizi İnceleyelim

Bölüm
3

React Components

Component Nedir?

Component Nedir?





BTK
AKADEMİ

Bölüm 4

React State & Events

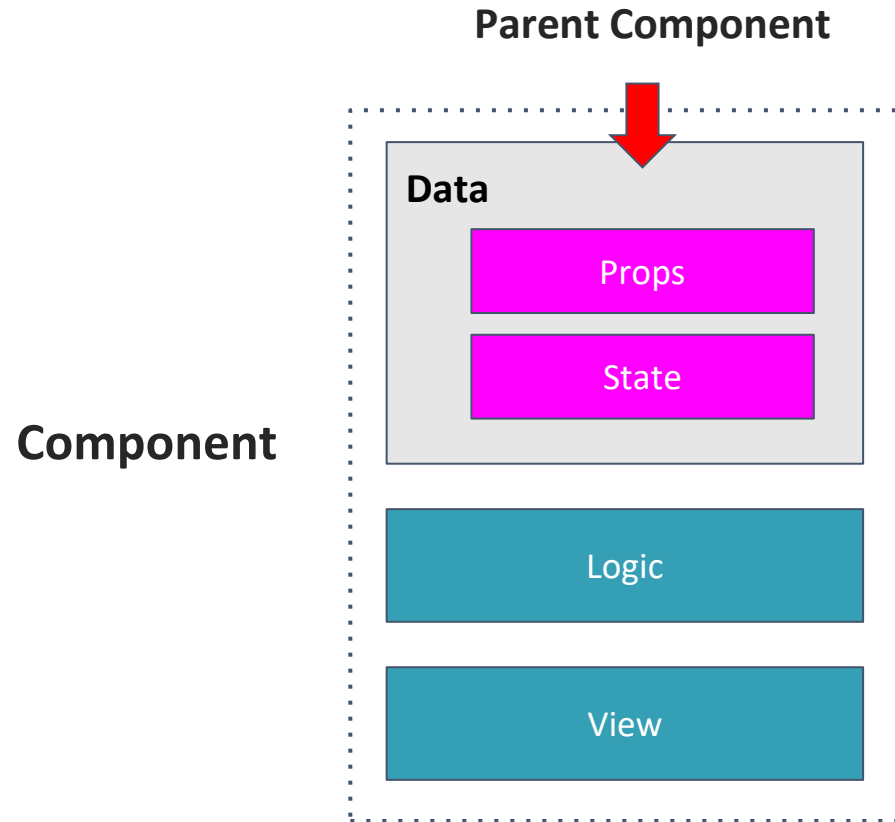
React Events

Bölüm
4

React State & Events

State Nedir?

State Nedir?



- 1 State component' in hafızasıdır.
- 2 State güncellediğinde component tekrar render edilir.
- 3 Local değişkenler korunur.

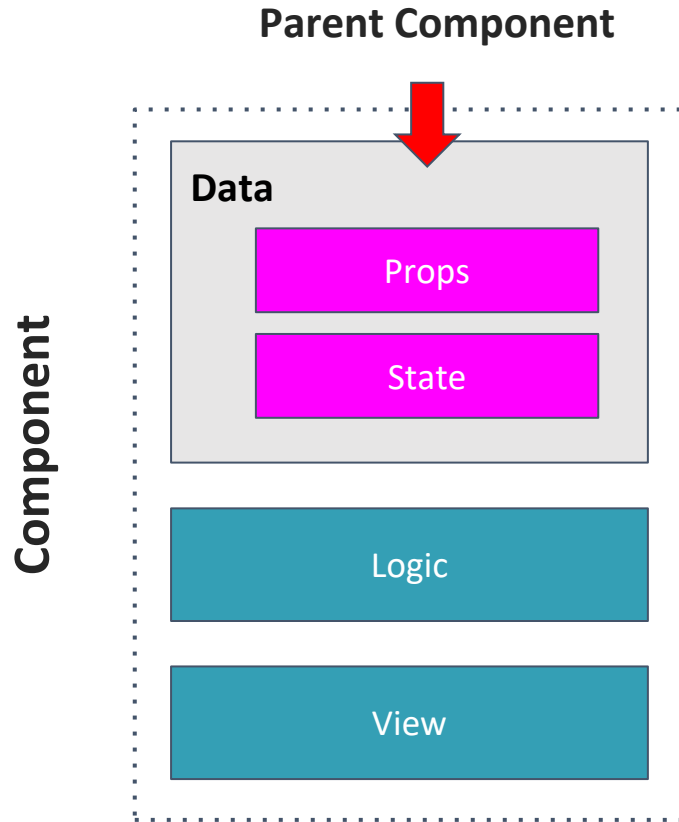
State Yöntemleri

- 1 **useState()**
- 2 **useReducer()**
- 3 **Context API**
- 4 **React Redux**

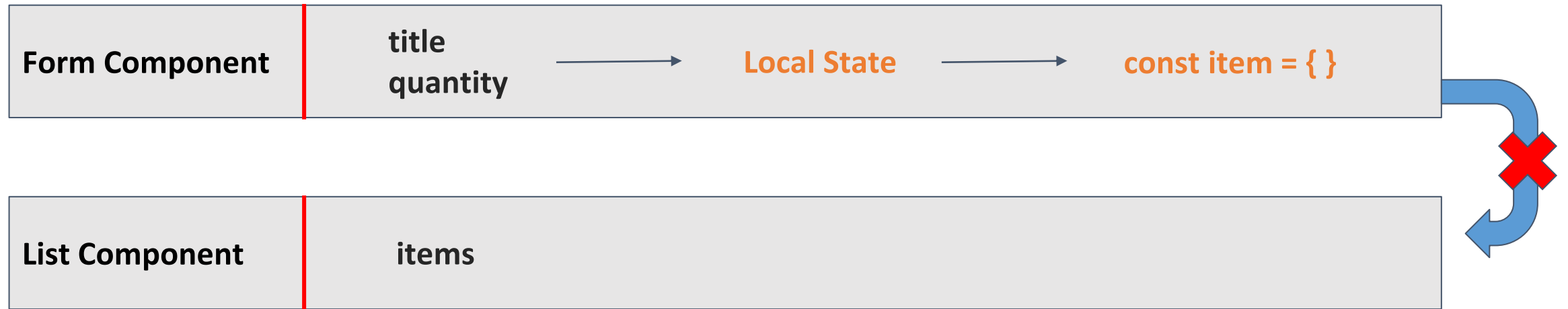
Bölüm
4

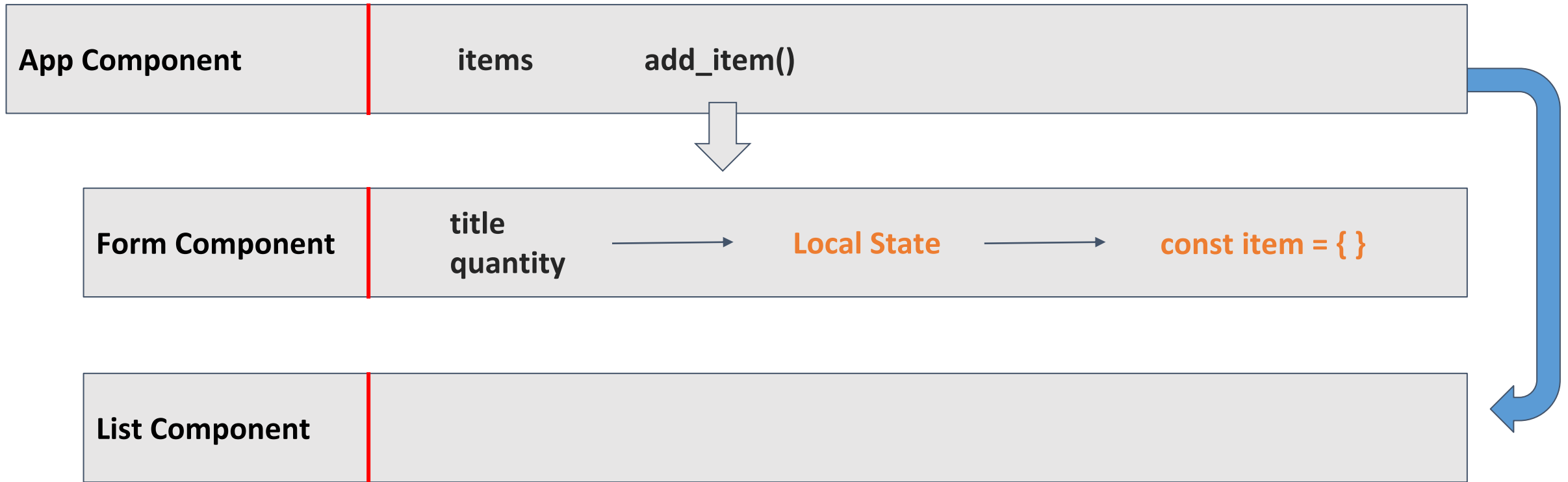
React State & Events

State Management



- ✓ Değişecek bilgiler State ile tanımlanır.
- ✓ State güncellendiğinde component tekrar render edilir.
- ✓ Değişmeyecek sabit bilgiler const ile tanımlanır.
- ✓ Parent component child component'e props ile veri gönderir.
- ✓ State kullanıldığı component içinde tanımlanırsa Local State olur.
- ✓ Local state eğer child component' de kullanılacaksa props ile gönderilir.
- ✓ State eğer kardeş (sibling) component' de kullanılacaksa parent component' e taşınması gerekir.







BTK
AKADEMİ

Bölüm 4

React State & Events

Global State

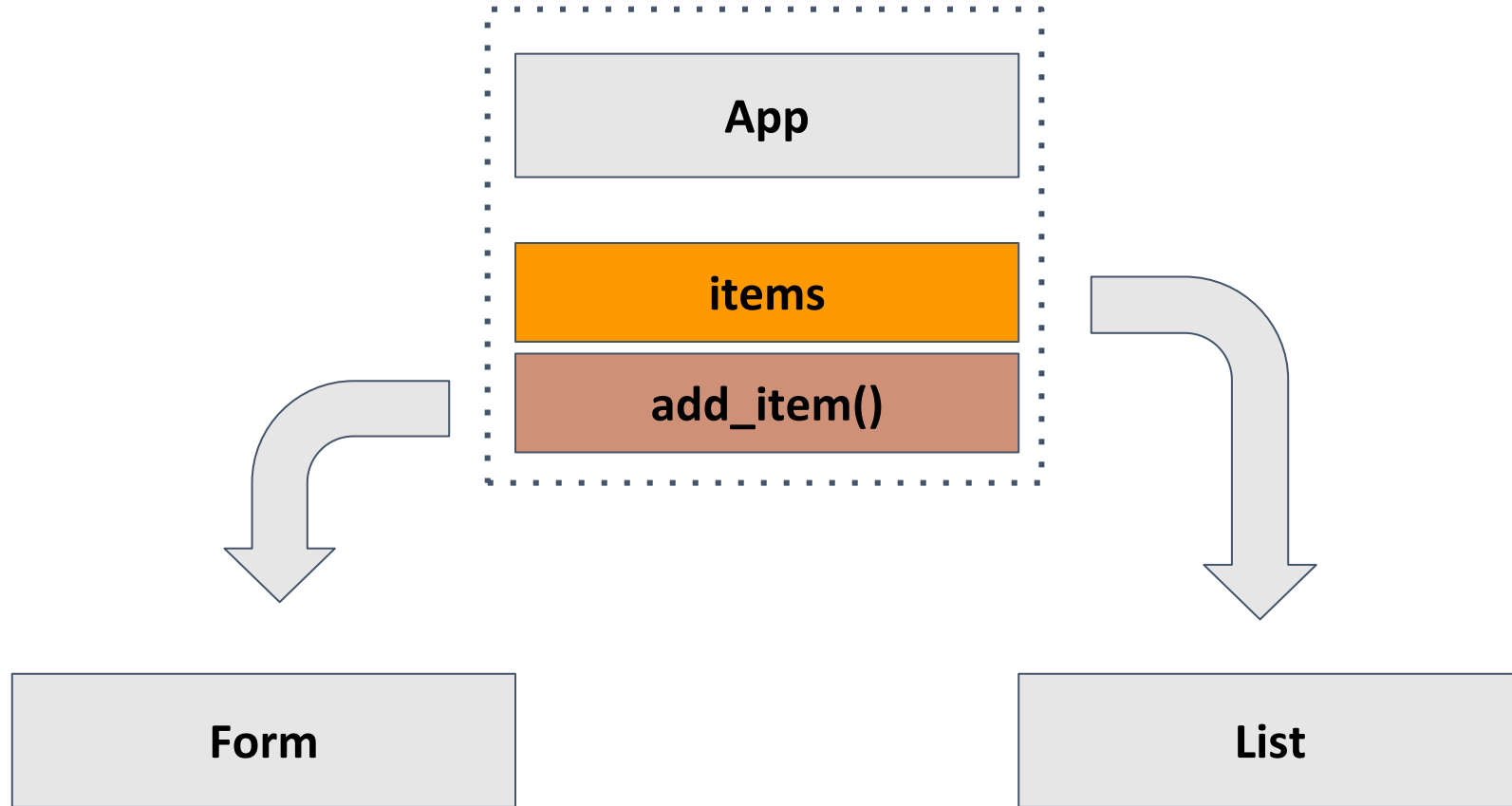
App

Form

items

add_item()

List



Bölüm
5

Component Compositions

PropTypes

Bölüm
5

UseEffect & APIs

UseEffect

Bölüm 6

UseEffect & APIs

UseEffect Cleanup Function

Component Lifecycle

Mount
(First Render)

Component' in ilk render edilme anı.

Re-Render

Component' in tekrar render edilmesi. (state, props güncellemesi)

Unmount
(Destroy)

Component' in silinme anı.

```
useEffect(() => {
```

```
  return () => {  
    // Cleanup  
  };
```

```
}, [ ]);
```

```
useEffect( () => {
```

```
  let timerId = setTimeout( () => {  
  
    }, 1000);
```

```
  // clear the timer when component unmouts  
  return () => clearTimeout(timerId);
```

```
}, [ ]);
```

Bölüm
6

UseEffect & APIs

Component Files

Bölüm 8

Form Validations

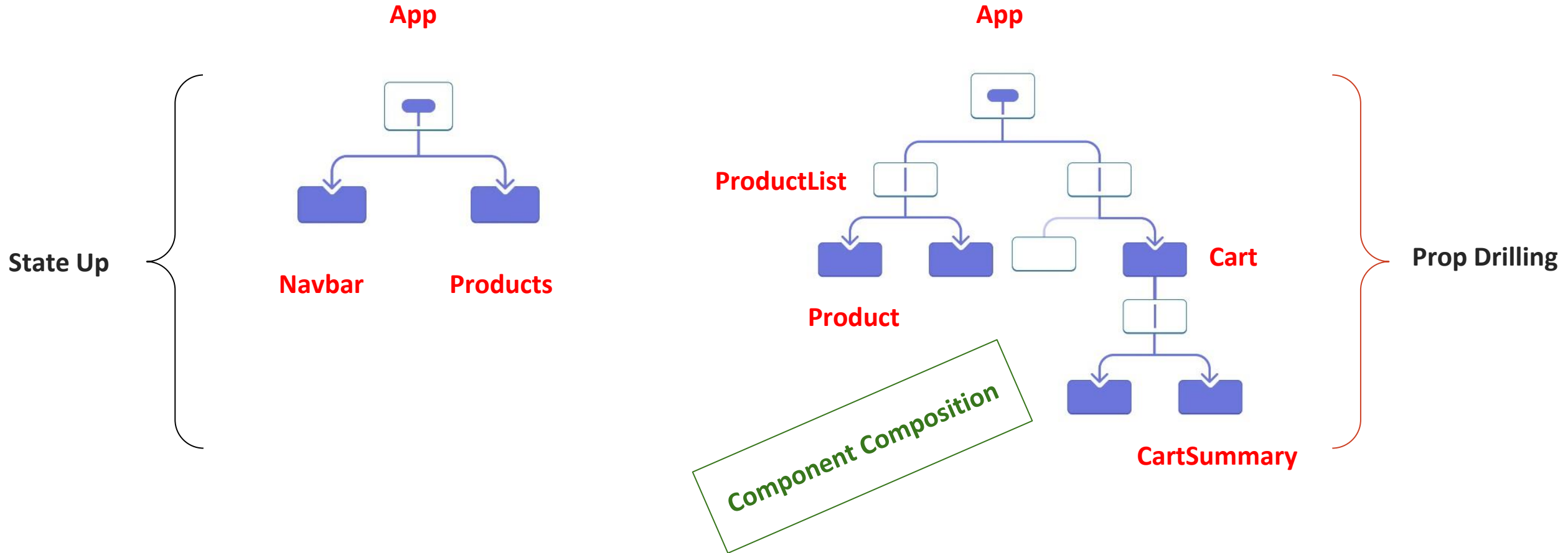
Validations in `useInput()`

Bölüm
9

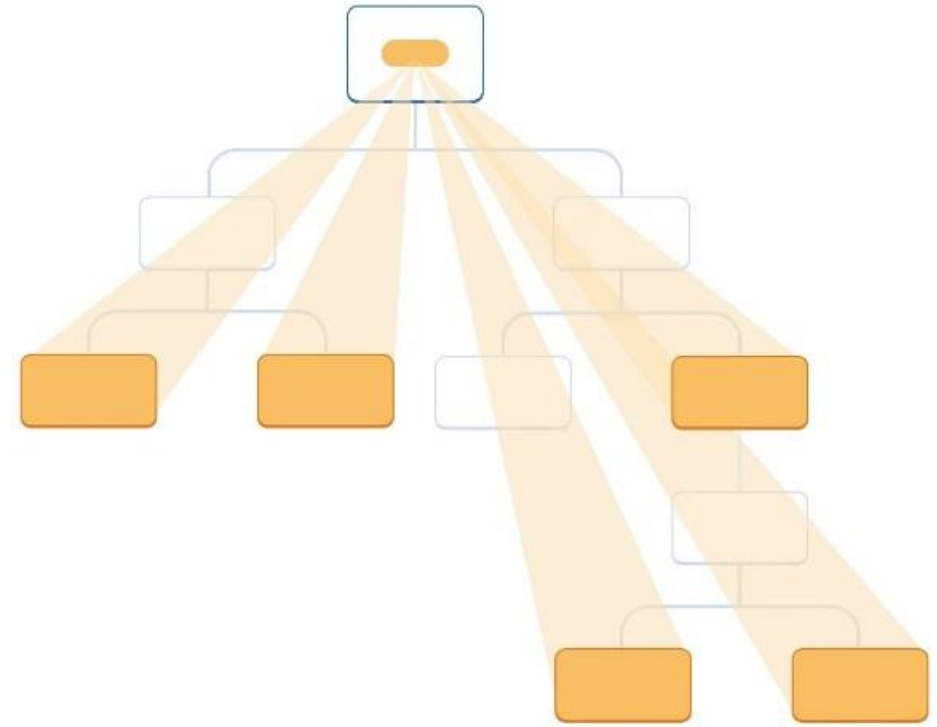
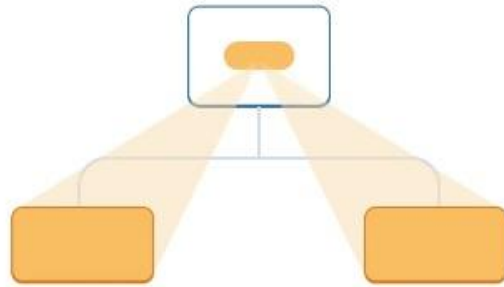
Context Api & Reducer

Context Api

useState()



useContext()



Bölüm
10

Material UI

Card



BTK
AKADEMİ

Bölüm
11

İleri Seviye Konulara Giriş

İleri Seviye Konular

- **React Redux**
- **React Router**
- **Authentication**
- **React Query**
- **Next.js**

Bölüm
12

Eğitim Sonu - Kapanış