14. Context

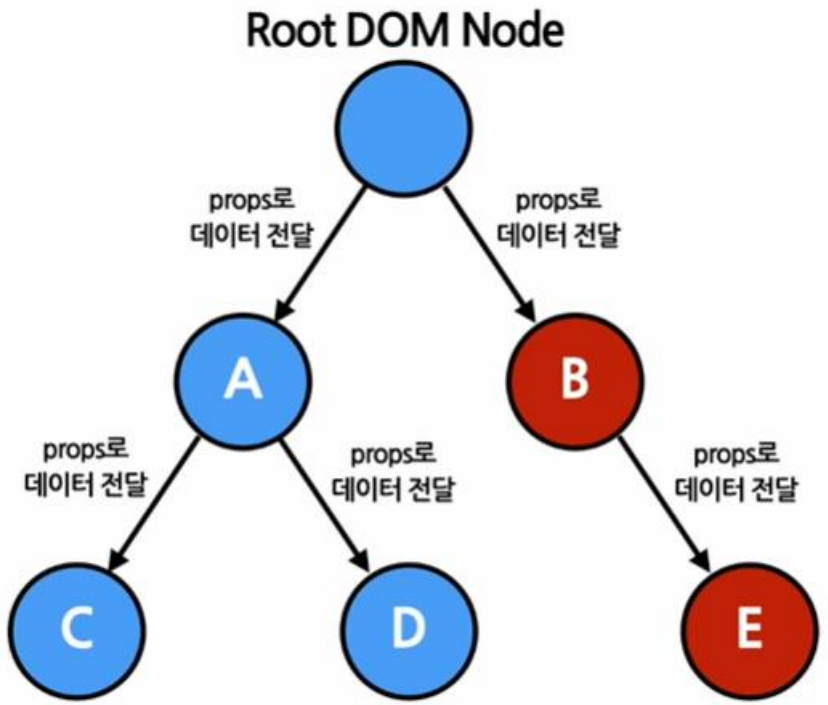
일일이 props를 넘겨주지 않고도
컴포넌트 트리 전체에 데이터를 제공 가능

14.1 컨텍스트(Context)란?

- 컴포넌트들 사이에서 데이터를 props를 통해 전달하는 것이 아닌 컴포넌트 트리를 통해 곧바로 데이터를 전달하는 방식
- 어떤 컴포넌트든지 컨텍스트에 있는 데이터에 쉽게 접근 가능

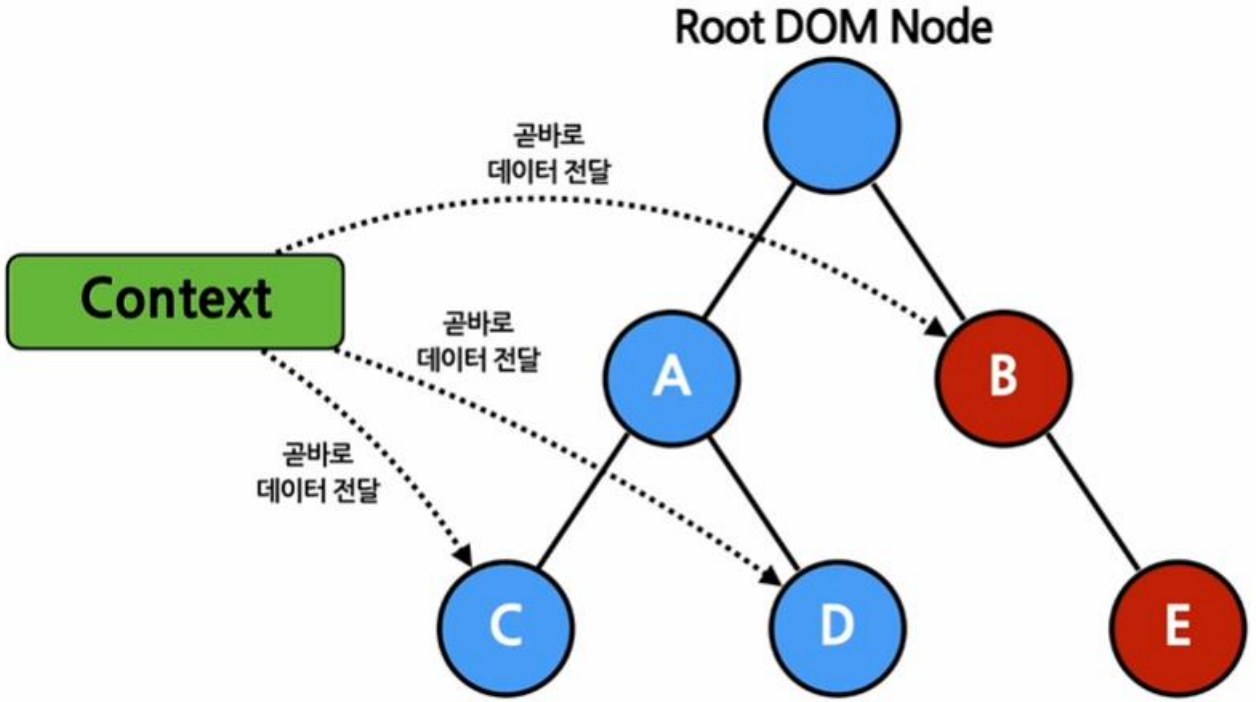
14.1 컨텍스트(Context)란?

- 일반적인 React 앱에서 데이터는 위에서 아래로(부모로부터 자식에게) props를 통해 전달되지만, 앱 안의 여러 컴포넌트들에 전해줘야 하는 props의 경우 이 과정이 번거로울 수 있음
- 예를 들면
  - Locale 정보
  - UI 테마
  - 로그인 유저 정보



14.1 컨텍스트(Context)란?

- context를 이용하면, 트리 단계마다 명시적으로 props를 넘겨주지 않아도 많은 컴포넌트가 이러한 값을 공유하도록 할 수 있음
- How? Broadcast!



14.2 언제 context를 써야 할까?(예제)

- React 컴포넌트 트리 안에서 전역(global) 데이터를 공유할 수 있도록 고안된 방법
  - 전역 데이터로는 현재 로그인한 유저, 테마, 선호하는 언어 등이 있음
- 여러 컴포넌트에서 계속해서 접근이 일어날 수 있는 데이터들이 있는 경우

14.3 Context API

- 1) React.createContext(defaultValue)
  - 컨텍스트를 생성하기 위한 함수
  - 컨텍스트 객체를 리턴함
  - defaultValue: Provider를 **찾지 못했을 때만** 쓰이는 값

```
const MyContext = React.createContext(defaultValue);
```

14.3 Context API

- 2) Context.Provider
  - 모든 컨텍스트 객체는 Provider라는 컴포넌트를 갖고 있음
  - Provider 컴포넌트로 하위 컴포넌트들을 감싸주면 모든 하위 컴포넌트들이 해당 컨텍스트의 데이터에 접근 가능
  - Provider에는 value라는 prop이 있으며, 하위에 있는 컴포넌트에게 전달됨
  - 여러 개의 Provider 중첩 가능

```
<MyContext.Provider value={/* 어떤 값 */}>
```

14.3 Context API

- 3) Context.Consumer
  - 컨텍스트의 데이터 변화를 구독하는 컴포넌트
  - 데이터를 소비한다는 뜻에서 consumer 컴포넌트라고도 부름
  - consumer 컴포넌트는 컨텍스트 값의 변화를 지켜보다가 값이 변경되면 재렌더링됨
  - 하나의 Provider 컴포넌트는 여러 개의 consumer 컴포넌트와 연결될 수 있음
  - 상위 레벨에 매칭되는 Provider가 없을 경우 기본값 사용

```
<MyContext.Consumer>
  {value => /* context 값을 이용한 렌더링 */}
</MyContext.Consumer>
```

14.3 Context API

- 4) Context.displayName
  - 크롬의 리액트 개발자 도구에서 표시되는 컨텍스트 객체의 이름

```
const MyContext = React.createContext(/* some value */);
MyContext.displayName = 'MyDisplayName';

<MyContext.Provider> // "MyDisplayName.Provider" in DevTools
<MyContext.Consumer> // "MyDisplayName.Consumer" in DevTools
```

14.4 여러 개의 context 사용하기(예제)

- Provider 컴포넌트와 Consumer 컴포넌트를 여러 개 중첩해서 사용하면 됨

14.5 useContext

- 함수 컴포넌트에서 컨텍스트를 쉽게 사용할 수 있게 해주는 훅
- 생성된 **컨텍스트 객체를 인자**로 받아서 현재 컨텍스트의 값을 리턴
- 컨텍스트의 값이 변경되면 변경된 값과 함께 useContext()를 사용하는 컴포넌트가 재렌더링됨

```
const value = useContext(MyContext);
```

14.6 테마 변경 기능 만들기(실습)

- my-app/src/chapter14 실습