2025.01.08.목

생성일	@2026년 1월 8일 오전 8:56
태그	React

목차

▼ 1) DOM

- HTML/CSS 파일의 트리 구조 구현체
- 계층 구조로 각 노드는 속성값을 갖고 있다.
- VDOM (Virtual DOM)
  - 리액트에서 사용하는 가상 돔
  - 엘리먼트로 virtual dom 생성
  - 매번 돔을 만드는 것은 매우 무거운 작업이기에 변경 사항(변경된 노드)만 적용한 가상의 돔을 먼저 만들고 비교 및 업데이트 하는 방법을 사용한다.

▼ 2) JSX (JavaScript XML)

- JS + HTML 처럼 보이지만 컴파일 단계에서 JS로 변환된다. 실제로 브라우저가 HTML로 인식해서 이해할 수는 없다.
- XSS (Cross Site Scripting): 웹 애플리케이션에 악성 스크립트 삽입하여 공격
  - React의 JSX는 동적으로 생성되는 HTML 콘텐츠를 자동으로 escaping 처리
    - JSX를 HTML과 JS로 안전하게 분리
    - 악성 JS가 HTML으로 변환 및 실행되지 않도록 함

▼ 3) Element

- JSX를 통해 생성되며, React 컴포넌트의 인스턴스를 나타냄
- 불변성
- 렌더링: ReactDOM.render() → 초기 렌더링 이후 업데이트 처리
- 업데이트: 불변이기에 변경시 새로운 엘리먼트를 생성하고 ReactDOM이 렌더링
- 변경부분을 VDOM으로 체크하여 업데이트

▼ 4) Props

- Properties
- 외부에서 컴포넌트로 전달하는 데이터
- 컴포넌트 내부에서 Props 값 변경 불가

▼ 5) State

- 컴포넌트의 데이터 또는 정보를 저장하는 객체
- 클래스형 컴포넌트에서 상태 관리: this.state

- 함수형 컴포넌트에서 상태관리: useState() Hook
- 특징
  - 초기값 설정 필요
  - 비동기적 업데이트, 콜백 함수 지원
  - 변경시 컴포넌트 재렌더링
- 스테이트와 인스턴트 필드의 차이

	State	Instant Field
정의	컴포넌트의 상태를 나타내는 데이터	컴포넌트 클래스 내에서 정의된 변수
관리	React의 상태 관리 시스템	시스템 외부에서 관리
용도	사용자 입력, API 응답등 렌더링에 영향을 미치는 데이터	내부 계산 결과, 비동기 작업 결과 저장, 렌더링과 무관한 데이터

- 스테이트와 프롭스의 차이

State	Props
동적 데이터를 저장 컴포넌트 내부에서 관리 컴포넌트 내부에서 변경 가능 주로 동적 데이터와 상태 관리에 사용	외부에서 전달된 데이터 부모 컴포넌트가 자식 컴포넌트에 전달 자식 컴포넌트 내부에서 변경 불가 컴포넌트 간 데이터와 함수를 전달

▼ 6) Component

- 리액트에서 UI를 구성하는 기본 단위
- Class Component: 현재는 자주 사용하지 않음
- Functional Component: 무상태 함수형 컴포넌트

▼ 7) LifeCycle

- 마운트: React 컴포넌트가 처음으로 DOM에 삽입
- 업데이트: props, state의 변경으로 재렌더링
- 언마운트: 컴포넌트를 DOM에서 제거하여 리소스 정리, 메모리 누수 방지
- 오류처리

▼ 8) Hook

- 함수형 컴포넌트에서 라이프사이클을 관리할 수 있게 해주는 기능
- useState: 컴포넌트의 지역 상태를 생성하고 업데이트를 트리거 하는 훅
- useEffect: 렌더링 커밋 후 사이드 이펙트 실행 훅 (DOM 변경, 구독, 타이머...)
- useMemo: 계산 결과를 메모제이션하여 불필요한 재계산 방지
- useCallback: 함수 참조를 메모제이션하여 불필요한 함수 재생성을 방지
- useContext: Context 구독하고 읽어옴
- useReducer: useState의 대안으로 복잡한 상태 로직을 리듀서 패턴으로 관리
- 커스텀 훅