

react

1강. 리액트 소개

리액트는 무엇인가?

사용자 인터페이스를 만들기 위한 자바스크립트 라이브러리

라이브러리 : 자주 사용되는 기능을 만들어 모아놓은 것

(문자열 관련, 숫자 관련, 날짜 관련...)

UI : 사용자 인터페이스, 사용자와 컴퓨터 간의 상호작용

UI 라이브러리 : react, vue.js

웹 개발의 트렌드 알기

웹사이트의 작동 원리와 흐름을 함께 이해하는 것이 중요!

리액트의 장점과 단점

장점

1. 빠른 업데이트와 렌더링 속도

virtual DOM, 최소한의 부분만 업데이트

2. component-based

수많은 component로 구성

3. 재사용성 (reusability)

소프트웨어를 개발할 때 중요, 개발 기간이 단축됨, 유지 보수가 용이, component가 재사용

4. meta 지원군

meta가 만든 오픈소스 프로그램, 꾸준한 버전 업데이트

5. 활발한 지식공유 & 커뮤니티

단점

1. 방대한 학습량

기존과 다른 방식의 UI

2. 높은 상태 관리 복잡도

상태 관리의 기본 개념을 제대로 이해하자

jsx

정의와 역할

자바스크립트의 확장 문법

js + html/xml

장점

1. 간결한 코드
2. 가독성 향상
3. 버그를 발견하기 쉬움!
4. injection attacks 방어

사용법

xml/html

{javascript}

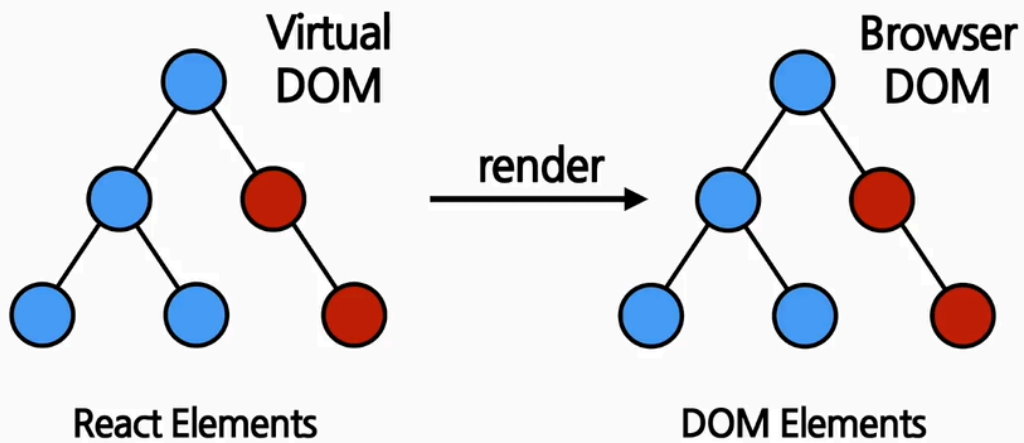
xml/html

element

리액트 앱을 구성하는 가장 작은 블록들

화면에서 보이는 것들을 기술

자바스크립트 객체 형태로 존재



elements의 특징

불변성(immutable)

elements 생성 후에는 children이나 attribute를 바꿀 수 없다!

```
const element = <h1>안녕, 리액트!</h1>;
ReactDOM.render(element, document.getElementById('root'));
/*렌더링하기*/
```

```
//렌더링 된 elements를 업데이트 하기
function tick() {
  const element = (
    <div>
      <h1>안녕, 리액트!</h1>
      <h2>현재 시간: {new Date().toLocaleTimeString()}</h2>
    </div>
  );

  ReactDOM.render(element, document.getElementById('root'));
}
```

```
setInterval(tick, 1000);
```