



React 공부

설치하기

- node.js 설치
- 폴더 생성
- vscode 터미널 에서 `npx create-react-app .`
 - `.` 은 현재디렉토리를 의미하는 것. 만약 앱의 이름을 정하고싶다면 `.` 대신에 이름을 적어주면 됨
- `npm start` —→ 리액트 개발을 할 수 있는 환경이 실행됨.
- `npm`

JSX?

- React에서는 본질적으로 렌더링 로직이 UI 로직(이벤트가 처리되는 방식, 시간에 따라 state가 변하는 방식, 화면에 표시하기 위해 데이터가 준비되는 방식 등)과 연결된다는 사실을 받아들입니다.

엘리먼트 렌더링

- 엘리먼트는 리액트앱의 가장 작은 단위
- 엘리먼트는 화면에 표시할 내용을 기술

- 브라우저 DOM 엘리먼트와 달리 React 엘리먼트는 일반 객체이며(plain object) 쉽게 생성할 수 있다. React DOM은 React 엘리먼트와 일치하도록 DOM을 업데이트.

- EX) `const element = <h1>Hello, world</h1>;`

- DOM에 엘리먼트 렌더링 하기

- `<div id="root"></div>`

-

```
const root = ReactDOM.createRoot(
  document.getElementById('root')
);
const element = <h1>Hello, world</h1>;
root.render(element);
```

렌더링 된 엘리먼트 업데이트 하기

- React 엘리먼트는 불변객체
- 엘리먼트를 생성한 이후에는 해당 엘리먼트의 자식이나 속성을 변경할 수 없다.
- 엘리먼트는 영화에서 하나의 프레임과 같이 특정 시점의 UI를 보여줌.
- EX) 똑딱거리는 시계
-

```
const root = ReactDOM.createRoot(
  document.getElementById('root')
);

function tick() {
  const element = (
    <div>      <h1>Hello, world!</h1>      <h2>It is {new Date().toLocaleTimeString()}</h2>    </div>);
  root.render(element);}

setInterval(tick, 1000);
```

변경된 부분만 업데이트 하기

- React DOM은 해당 엘리먼트와 그 자식 엘리먼트를 이전의 엘리먼트와 비교하고 DOM을 원하는 상태로 만드는데 필요한 경우에만 DOM을 업데이트

컴포넌트와 Props

- 컴포넌트를 정의하는 가장 간단한 방법은 JavaScript 함수를 작성하는 것
- ex)

```
function Welcome(props) {  
  return <h1>Hello, {props.name}</h1>;  
}
```

- 이 함수는 데이터를 가진 하나의 “props” (props는 속성을 나타내는 데이터) 객체 인자를 받은 후 React 엘리먼트를 반환하므로 유효한 React 컴포넌트

컴포넌트 렌더링

- react 엘리먼트는 사용자 정의 컴포넌트로도 나타낼 수 있다.
- `const element = <Welcome name="Sara" />;`
- React가 사용자 정의 컴포넌트로 작성한 엘리먼트를 발견하면 JSX 어트리뷰트와 자식을 해당 컴포넌트에 단일 객체로 전달. 이 객체를 “props”라 부름.
- 페이지에 “Hello, Sara”를 렌더링하는 예시
-

```
function Welcome(props) { return <h1>Hello, {props.name}</h1>;  
}  
  
const root = ReactDOM.createRoot(document.getElementById('root'));  
const element = <Welcome name="Sara" />;root.render(element);
```

- 어떻게 수행?
1. `<Welcome name="Sara" />` 엘리먼트로 `root.render()` 를 호출합니다.

2. React는 `{name: 'Sara'}` 를 props로 하여 `Welcome` 컴포넌트를 호출합니다.
3. `Welcome` 컴포넌트는 결과적으로 `<h1>Hello, Sara</h1>` 엘리먼트를 반환합니다.
4. React DOM은 `<h1>Hello, Sara</h1>` 엘리먼트와 일치하도록 DOM을 효율적으로 업데이트합니다.

cf)

주의: 컴포넌트의 이름은 항상 대문자로 시작합니다.

React는 소문자로 시작하는 컴포넌트를 DOM 태그로 처리합니다. 예를 들어 `<div />` 는 HTML div 태그를 나타내지만, `<Welcome />` 은 컴포넌트를 나타내며 범위 안에 `Welcome` 이 있어야 합니다.

컴포넌트 합성

- 컴포넌트는 자신의 출력에 다른 컴포넌트를 참조할 수 있음.
- 이는 모든 세부 단계에서 동일한 추상 컴포넌트를 사용할 수 있음을 의미.
- React 앱에서는 버튼, 폼, 다이얼로그, 화면 등의 모든 것들이 흔히 컴포넌트로 표현.
- ex) welcome을 여러번 렌더링하는 app컴포넌트
-

```
function Welcome(props) {
  return <h1>Hello, {props.name}</h1>;
}

function App() {
  return (
    <div>
      <Welcome name="Sara" />
      <Welcome name="Cahal" />
      <Welcome name="Edite" />
    </div>
  );
}
```

컴포넌트 추출

- ex)

-

```
function Comment(props) {
  return (
    <div className="Comment">
      <div className="UserInfo">
        <img className="Avatar" src={props.author.avatarUrl} alt={props.author.name}/>
        <div className="UserInfo-name">
          {props.author.name}
        </div>
      </div>
      <div className="Comment-text">
        {props.text}
      </div>
      <div className="Comment-date">
        {formatDate(props.date)}
      </div>
    </div>);
}
```

- avatar 추출

-

```
function Avatar(props) {
  return (
    <img className="Avatar" src={props.user.avatarUrl} alt={props.user.name}
  /> );
}
```

-

```
function Comment(props) {
  return (
    <div className="Comment">
      <div className="UserInfo">
        <Avatar user={props.author} />
        <div className="UserInfo-name">
          {props.author.name}
        </div>
      </div>
      <div className="Comment-text">
        {props.text}
      </div>
      <div className="Comment-date">
        {formatDate(props.date)}
      </div>
    </div>);
}
```

props 는 읽기 전용?

- 함수 컴포넌트나 클래스 컴포넌트 모두 컴포넌트의 자체 props를 수정해서는 안된다.
- 모든 React 컴포넌트는 자신의 props를 다룰 때 반드시 순수 함수처럼 동작해야한다.