

웹 게임을 만들며 배우는 React(Chap1~Chap4)

0 - 페이지

웹 게임을 만들며 배우는 React, <https://www.inflearn.com/course/web-game-react/dashboard>

<https://github.com/ZeroCho/react-webgame>

https://www.youtube.com/results?search_query=인간+자바스크립트+엔진

1 - 리액트를 왜 쓰는가

SPA(Single Page Application)

- 웹페이지와 비교되는 웹 애플리케이션
- 프론트엔트 개발자가 나오게 된 계기이기도 함
- 데이터가 바뀌면 화면이 바뀌도록 하는데 사용하기 좋은게 앵귤러, 리액트(m.페이스북), 뷰, 스벨트 등이고, 한국에서 리액트가 대세가 됨
- 복잡한 웹앱에서 데이터와 화면 일치 문제 해결하기 위함

2 - 강의 수강 시 주의할 점

기본기

- HTML, CSS, JavaScript 공부가 잘 되어있어야 리액트를 잘 다룰 수 있음
- 위 세가지 공부를 꾸준히 해야함

깃

<https://github.com/ZeroCho/react-webgame>

3 - 첫 리액트 컴포넌트(아직은 Class)

컴포넌트

- 데이터와 화면을 하나로 묶어놓은 덩어리
- 데이터 : state 부분
- 화면: render 에 return 부분
- 화면에 바뀔 부분을 state 로 만든다는 것을 머리에 넣기

HTML 실행방법

1. 확장 프로그램 설치 (Open in browser) Extension -> Open in browser 검색/선택 -> install.
2. HTML을 브라우저에서 실행하기 (단축키 : Alt + B)

JS - 'use strict' 란?

- [참고] <https://m.blog.naver.com/on21life/221654555798>
- 암묵적인 느슨한 모드(sloppy mode) 를 해제하고 명시적인 엄격 모드(strict mode) 를 사용하는 방법
- 일반적인 코딩 실수나 안전하지 않은 동작을 포착하도록 하므로 아래 장점을 갖는다.
 - 디버깅이 쉬워진다.
 - 비 Strict 코드보다 빠르게 수행할 수 있다.
 - 발생가능한 에러를 예방한다.

실습코드

- 1. 구구단 > like-button.html

```
<html>
<head>
  <meta charset="utf-8" />
  <title>웹게임</title>
</head>
<body>
  <div id="root"></div>
  <script
    crossorigin
    src="https://unpkg.com/react@18/umd/react.development.js"
  ></script>
  <script
    crossorigin
    src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"
  ></script>
  <!--<script src="https://unpkg.com/react@16/umd/react.production.min.js" crossorigin></script>-->
  <!--<script src="https://unpkg.com/react-dom@16/umd/react-dom.production.min.js" crossorigin></script>-->
  <script>
    "use strict";

    const e = React.createElement;

    class LikeButton extends React.Component {
      constructor(props) {
        super(props);
        this.state = { liked: false };
      }
      render() {
        if (this.state.liked) {
          return "You liked this.";
        }

        return e(
          "button",
          { onClick: () => this.setState({ liked: true }) },
          "Like"
        );
      }
    }
  </script>
  <script>
    ReactDOM.render(e(LikeButton), document.querySelector("#root"));
  </script>
```

```
</body>
</html>
```

4 - 가독성을 위한 JSX(XML임!)

버전 차이 확인하는 방법

- render(React17) 를 쓰는지 createRoot(React18) 를 쓰는지 확인
- 대부분의 회사에서 17을 쓰는 곳이 많기 때문에 중요한 부분.

```
<script>
  ReactDOM.render(<LikeButton />, document.querySelector("#root"));
  ReactDOM.createRoot(document.querySelector('#root')).render(<LikeButton />);
</script>
```

기타 유의할 점

- for 문(배열)과 if 문(삼항연산자) 사용방법이 다르고, map 자주 사용됨
- 자바스크립트 문법은 중괄호로 묶어줘야 함
- return 안에 형제 관계 태그가 있으면 부모 태그 하나로 묶어줘야 함, return 은 하나의 태그만 가능
- 가짜 태그 <></> 사용 가능 : Fragment

```
return (
  <div>
    <button></button>
    <input></input>
  </div>
);
```

- 기타 자주 나오는 코드 형태 눈에 익혀둘 것, 삼항연산자 등

```
return (
  <button
    onClick={() => {
      this.setState({ liked: true });
    }}
  >
    {this.state.liked ? "Liked" : "Like!"}
    {[1, 2, 3].map((i) => {
      return <div>i</div>;
    })}
  </button>
);
```

삼항연산자

- 앞에서부터 조건문, 물음표(?), 조건문이 참(truthy)일 경우 실행할 표현식, 콜론(:), 조건문이 거짓(falsy)일 경우 실행할 표현식이 배치됩니다. 해당 연산자는 `if...else` 문의 대체재로 빈번히 사용됩니다.
- https://developer.mozilla.org/ko/docs/Web/JavaScript/Reference/Operators/Conditional_Operator