

220629

React 기초

HACK YOUR LIFE!

{

JSX

컴포넌트

map() 함수

}

React란?

리액트 | 유저인터페이스를 만들기 위한 자바스크립트 **라이브러리**

ex) 자바스크립트가 기본 캐릭이라면 리액트는 도구 짱짱한 업글 캐릭

virtual DOM | 실제 DOM에 접근하여 조작하는 대신,

이를 추상화한 자바스크립트 객체를 구성하여 사용하여

실제 DOM의 가벼운 사본 DOM의 상태를 메모리에 저장하고,

변경 전과 변경 후의 상태를 비교한 뒤 최소한의 내용만 반영 하는 기능

* 프론트엔드 끝판왕~

리액트 앱 생성하기

```
yarn create react-app 프로젝트명
```

```
cd 프로젝트명
```

```
yarn start
```

```
import logo from './logo.svg';  
import './App.css';
```

```
function App() {  
  return (  
    <div className="App">  
      <header className="App-header">  
        <img src={logo} className="App-logo" alt="logo" />  
        <p>  
          Edit <code>src/App.js</code> and save to reload.  
        </p>  
        <a  
          className="App-link"  
          href="https://reactjs.org"  
          target="_blank"  
          rel="noopener noreferrer"  
        >  
          Learn React  
        </a>  
      </header>  
    </div>  
  );  
}
```

```
export default App;
```

import | 불러오기

→ 다른 파일들을 불러와 사용

function | 함수 컴포넌트

→ 함수에서 반환하고 있는 내용 보여줌

export | 내보내기

→ 다른 파일들에 불러질 때 쓰일 이름

```
▼ node_modules
  > .bin
  > @ampproject
  > @babel
  > @bcoe
  > @csstools
  > @eslint
  > @humanwhocodes
  > @istanbuljs
  > @jest
  > @jridgewell
  > @leichtgewicht
  > @nodelib
  > @pmmmwh
  > @rollup
  > @rushstack
  > @sinclair
  > @sinonjs
  > @surma
  > @svgr
  > @testing-library
```

```
▼ public
  ★ favicon.ico
  <> index.html
  🖼 logo192.png
  🖼 logo512.png
  {} manifest.json
  ≡ robots.txt
```

```
▼ src
  # App.css
  JS App.js
  JS App.test.js
  # index.css
  JS index.js
  🖼 logo.svg
  JS reportWebVitals.js
  JS setupTests.js
```

```
📄 .gitignore
{} package.json
📖 README.md
👤 yarn.lock
```

node_modules | 모듈 등이 저장된 폴더

public | 정적 파일들이 담겨있음

→ 정적 파일이란? 미리 저장된 파일

src | 썩네기

→ 우리가 채워넣어야 함!

.gitignore | 민감한 정보가 들어있는 파일이나 형상
관리할 필요가 없는 파일들이 git에 업로드되지 않도록
무시할 목록들

package.json | 어떤 모듈들이 설치되어있는지에
대한 기본정보들

yarn.lock | 패키지 잠금 파일. 프로젝트에 사용되는
패키지의 모든 버전을 고정시킴

▼ public

★ favicon.ico

<> index.html

🖼️ logo192.png

🖼️ logo512.png

{ } manifest.json

≡ robots.txt

▼ src

App.css

JS App.js

JS App.test.js

🌀 Hello.jsx

index.css

JS index.js

🖼️ logo.svg

JS reportWebVitals.js

JS setupTests.js

index.html

- 메인 프로그램인 index.js에 대응되는 것으로, HTML 템플릿 파일
- 직접 표시 x
- index.js에 의해 렌더링된 결과가 표시됨 (index.html 이름을 바꿀시 오류발생)

App.js

- 컴포넌트를 정의하는 프로그램
- 실제로 화면에 보이는 내용 등이 정의 되는 곳

index.js

- 메인 프로그램
- HTML 템플릿, JS 컴포넌트를 조합하여 렌더링
→ index.html과 jsx 연결고리!
- 실제로 표시를 하는 곳
- 건드릴 일은 없다!

이름.jsx

- 이름.js와 혼용해서 사용해도 가능.
ex) Hello.jsx = Hello.js

React가 뭐가 좋아요?

자동 업데이트! → F5가 뭐죠?

코드 재활용을 많이 한다! → 코딩을 덜 한다!

html + js 라면 혹시 이런건가요..?

```
<html>
<head>
  <title>Document</title>
  <script>
    document.write("hello world!"); // script 태그 안에 작성한 코드가 실행된다.
  </script>
</head>
<body>

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

JSX

html이랑 js를 합쳤다고요?

JSX 란?

JSX = JS + XML

자바스크립트의 확장문법이면서
XML과 매우 비슷하게 생긴 것

아직 뭔지 감이 잘 안와요!

HTML

```
<div>
  <h1>멋쟁이사자처럼</h1>
  <p>중앙대학교</p>
  <span>최고</span>
</div>
```

XML

```
<puang>
  <club>멋쟁이사자처럼</club>
  <univ>중앙대학교</univ>
  <ddabong>최고</ddabong>
</puang>
```

JS

```
function session(){
  let club="멋쟁이사자처럼";
  const univ="중앙대학교";
  let ddabong="최고";
  document.write(club);
  document.write(univ);
  document.write(ddabong);
}
```

JSX = XML + JS

JSX

```
function session(){  
  return (  
    <main>  
      <h1>멋쟁이사자처럼</h1>  
      <p>중앙대학교</p>  
      <span>최고</span>  
    </main>  
  );  
}
```

JSX 란?

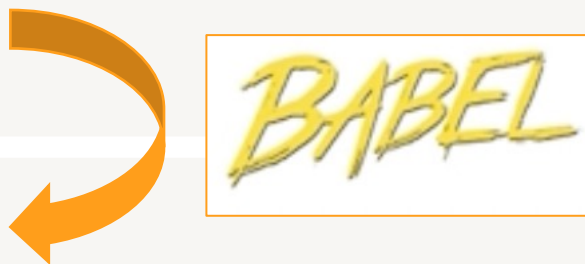
JSX = JS + XML

자바스크립트의 확장문법이면서
XML과 매우 비슷하게 생긴 것

바벨을 사용하여 일반 자바스크립트 형태의 코드로 변환

```
//jsx  
function App(){  
  return(<h1>Hi</h1>);  
}
```

선언형



```
//js  
function App(){  
  return React.createElement("h1", null, "Hi");  
}
```

명령형

JSX 왜 쓰나요?

1. 보기 쉽고 익숙하다
2. 더욱 높은 활용도

JSX 문법을 배워보자!

1. 하나의 요소로 감싼다.
2. JS 표현식 사용
3. 조건부 렌더링

하나의 요소로 감싼다

리액트에서는 요소 여러개를 꼭 하나의 요소로 감싸주어야합니다!

why?

virtual DOM에서 컴포넌트 변화를 감지해낼 때 효율적으로 비교할 수 있도록
컴포넌트 내부는 하나의 구조로 이루어져야한다는 규칙이 있기 때문

JS 표현식 사용

자바스크립트 표현식을 쓸 수 있다!

→ {} 사용하기

조건부 렌더링

return 문 안에서 if else 선언 불가능!!

(1) 조건부 연산자

```
{ 조건 ? (맞다면 출력할 내용) : (틀렸다면 출력할 내용) }
```

(2) &&와 || 이용

```
{ 조건 && 맞다면 출력할 내용 }  
{ 조건 || 틀렸다면 출력할 내용 }
```

컴포넌트

그래서 어떻게 써먹나요..?!

컴포넌트란?

Component

리액트로 만들어진 앱을 이루는
‘최소단위’

props나 state와 같은 데이터를 입력받아 DOM 노드를 생성한다.
재사용성을 높이고 유지보수를 용이하게 하기 위해 사용한다.

컴포넌트 선언 방식

class

vs

function

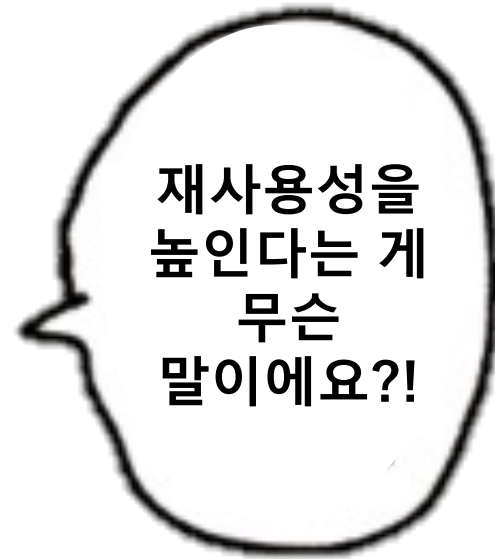
```
class App extends Component {  
  render(){  
    return <h1>리액트 재밋당</h1>  
  }  
}
```

```
function App(){  
  return <h1>리액트 재밋당</h1>  
}
```

```
const App = () => {  
  return <h1>리액트 재밋당</h1>  
};
```

→ 리액트 매뉴얼에서는 함수형을 권고하고 있으므로 함수형을 사용해봅시다!

```
function session(){  
  return (  
    <main>  
      <h1>멋쟁이사자처럼</h1>  
      <p>중앙대학교</p>  
      <span>최고</span>  
    </main>  
  );  
}
```

```
<div class="item">
  
  <p class = "title">참쌀 3kg</p>
  <p class = "address">서울 동작구 대방동</p>
  <div class="info">
    <p class = "pr">7,500원</p>
    <div class="price"><img class="heart" src = "../static/images/heart.png">1</div>
  </div>
</div>
```

```
</div>
<div class="item">
  
  <p class = "title">성인여성자전거</p>
  <p class = "address">서울 동작구 대방동</p>
  <div class="info">
    <p class = "pr">30,000원</p>
    <div class = "price">
      <img class="heart" src = "../static/images/heart.png">13
    </div>
  </div>
</div>
```

```
</div>
<div class="item">
  
  <p class = "title">현대상품권 5만원짜리 팔아요</p>
  <p class = "address">서울 동작구 대방동</p>
  <div class="info">
    <p class = "pr">40,000원</p>
  </div>
</div>
```

똑같은 코드의 반복?
시러시러~

컴포넌트 생성하기

1. 파일 만들기
2. 코드 작성하기
3. 모듈 내보내기 및 불러오기

컴포넌트 스타일링을 배워보자!


1. 파일 안에서 스타일링하기
2. css 이용하기
3. styled-components

<https://github.com/MicheleBertoli/css-in-js>

styled-components

Css -in - JS 라이브러리 중 개발자들이 가장 선호하는 라이브러리

```
yarn add styled-components
```



설치를
해보자

map() 함수

엥?! 근데 계속 똑같은 박스만 만드나요???

map..? 지도...?

map() 함수

반복적인 내용을 효율적으로 보여주고 관리하는 방법 중 하나!

자바스크립트 배열 객체의 내장 함수

배열 내 각 요소를 원하는 규칙에 따라 변환한 후

그 결과로 새로운 배열을 생성한다.

쉬는시간

실습!

페어코딩 당근마켓 페이지를 리액트로 만들어봅시다!

인기 중고



신세계 상품권 10만원권

서울 동작구 대방동

90,000원



프리즘 32인치 tv 팝니다

서울 동작구 대방동

20,000원

♡ 1

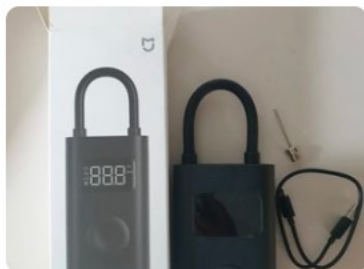


옐로우콘 20형 7단 접이식
오즈 성인자전거

자전거 가로사이즈(cm) 140cm

30,000원

♡ 13



샤오미 에어퍼프 팝니다

서울 동작구 대방동

30,000원



컬처랜드문화상품권 1만원권...

서울 동작구 대방동

17,000원



롯데상품권 5만원권 4장 할...

서울 동작구 대방동

188,000원

인기 중고



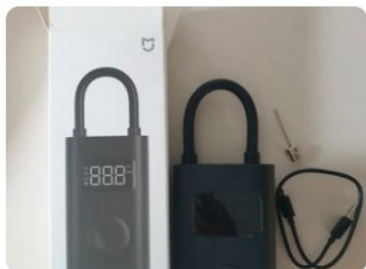
신세계 상품권 10만원권
서울 동작구 대방동
90,000원



프리즘 32인치 tv 팝니다
서울 동작구 대방동
20,000원 1



오토 20형 7단 접이식
오즈 성인자전거
자전거 가로사이즈(cm) 140cm
색상
블랙
성인여성자전거
서울 동작구 대방동
30,000원 13



샤오미 에어퍼프 팝니다
서울 동작구 대방동
30,000원

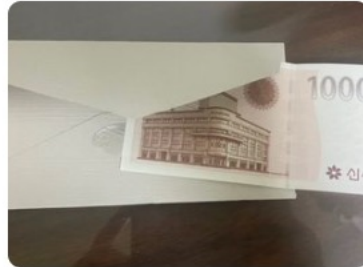


컬처랜드문화상품권 1만원권...
서울 동작구 대방동
17,000원



롯데상품권 5만원권 4장 할...
서울 동작구 대방동
188,000원

인기 중고



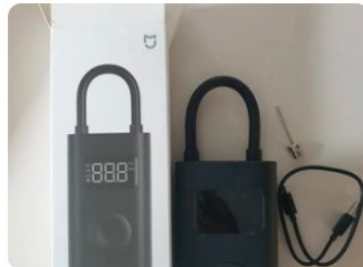
신세계 상품권 10만원권
서울 동작구 대방동
90,000원



프리즘 32인치 tv 팝니다
서울 동작구 대방동
20,000원 1



옐로우콘 20형 7단 접이식
오즈 성인자전거
자전거 가로사이즈(cm) 140cm
색상
블랙
성인여성자전거
서울 동작구 대방동
30,000원 13



샤오미 에어퍼프 팝니다
서울 동작구 대방동
30,000원



컬처랜드문화상품권 1만원권...
서울 동작구 대방동
17,000원



롯데상품권 5만원권 4장 할...
서울 동작구 대방동
188,000원



홈 > 확장 프로그램 > React Developer Tools



React Developer Tools

 추천

★★★★★ 1,393 ⓘ | [개발자 도구](#) | 사용자 3,000,000+명

Chrome에서 삭제

렌더링이 언제 어디서 되는지 확인하는 용도로 사용하는 툴

(1) components → 리액트 구조 확인 가능

(2) profiler → 렌더링에 시간이 얼마나 걸리는지 확인 가능

필수 과제

기한 | ~7/5 (화) 자정까지

내용 | 본인 레포 > assignment > react01이라는 폴더 생성 후 진행

리액트로 아래의 내용 구현

사진, 멘트, 하트 변경 가능



10분 컷 과카몰리 만드
는 법 공개... ❤️



여름엔 입맛 살리는 오이
냉국 어때... ❤️



비 오는 장마철엔 부추전
만들어 먹... ❤️

조건이 있따!



10분 컷 과카몰리 만드
는 법 공개... ❤️



여름엔 입맛 살리는 오이
냉국 어때... ❤️



비 오는 장마철엔 부추전
만들어 먹... ❤️

1. 이미지 배열, 텍스트 배열 따로 두 개의 배열을 만들어주세요.
(이 때, 텍스트 배열에 하트는 포함되지 않습니다.)
2. 꼭 map 함수를 이용해야합니다.
3. 겉모양 컴포넌트, 회색박스 컴포넌트 따로
두 개의 컴포넌트를 만들어주세요.
4. 포기하지않아요~ 할 수 있어요!



선택(이라쓰고 필수라 읽는다) 과제

CODELION 강의 수강

CH 1

CH 2-3



복습하장
어어?
예습도
한다구?
아주 좋아~

Thank you