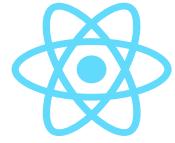
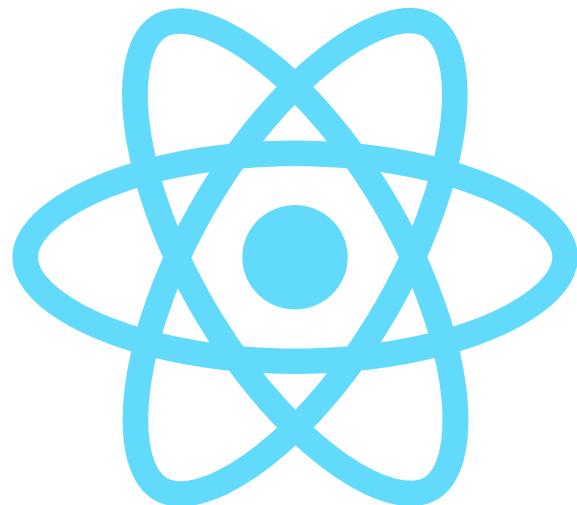


React Reference

React



- React는 웹 프레임워크로, 자바스크립트 라이브러리의 하나로서 사용자 인터페이스를 만들기 위해 사용된다.
- facebook에서 제공해주는 프론트엔드 라이브러리라고 볼 수 있습니다.



React

Data Flow

→ React는 데이터의 흐름이 한 방향으로만 흐르는 단방향 데이터 흐름을 가집니다.

Component 기반 구조

→ 독립적인 단위의 소프트웨어 모듈을 말합니다.

Virtual Dom

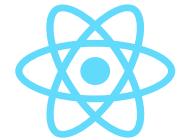
→ 가상의 Document Object Model을 말합니다.

Props and State

→ Props : 부모 컴포넌트에서 자식 컴포넌트로 전달해 주는 데이터

State : 컴포넌트 내부에서 선언하며 내부에서 값을 변경할 수 있습니다.

SPA



- SPA = Single Page Application의 약자입니다. 말 그대로 페이지가 하나인 어플리케이션입니다.

SPA의 장점

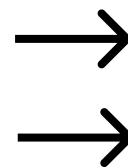
- 자연스러운 사용자 경험 (UX)
- 필요한 리소스만 부분적으로 로딩 (성능)
- 서버의 템플릿 연산을 클라이언트로 분산 (성능)
- 컴포넌트별 개발 용이 (생산성)
- 모바일 앱 개발을 염두에 둔다면 동일한 API를 사용하도록 설계 가능 (생산성)

SPA의 단점

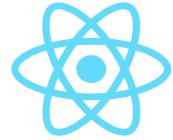
- 앱의 규모가 커지면 자바스크립트 파일이 너무 커짐
- 웹 브라우저에서 페이지 로딩 시킬 때 사용자가 발문하지 않을 수도 있는 페이지의 스크립트들도 불러옵니다.
- 검색 엔진의 검색 결과에 페이지가 잘 나타나지 않을 수 있음
- 흰 페이지가 나타날 수 있음

단점 개선 방법

- 코드 스파리팅을 사용하여 라우트 별로 파일들을 나누어서 트래픽과 로딩 속도를 개선 가능함
- 서버 사이드 렌더링을 통해 모두 해결할수 있음



렌더링



- html 요소(element), 또는 React 요소 등의 코드가 눈으로 볼 수 있도록 그려지는 것을 렌더링(rendering) 이라고 말합니다.

```
import React from 'react';
import ReactDOM from 'react-dom';

// let clock = document.getElementById("clock");

// setInterval(function(){
//   clock.innerHTML = new Date().toLocaleTimeString();
// });

  기존의 자바스크립트에서 사용했던 방식입니다.
```

```
function clock(){
  const element =
    <div>
      리액트에서 사용하는 방식입니다.(JSX)
      <h1>Hello, webstoryboy</h1>
      <h2>지금은 {new Date().toLocaleTimeString()} 입니다.</h2>
    </div>
  )
  (1)           (2)
  ReactDOM.render(element, document.getElementById("root"));
}

setInterval(clock, 1000);
```

← React 요소가 DOM node에 추가되어 화면에 렌더 되려면 ReactDOM.render 함수를 사용합니다.

출력 결과

Hello, KIMJUNGHO

지금은 오후 4:33:27 입니다.

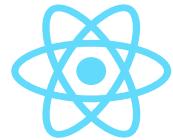
```
<h2> == $0
"지금은 "
"오후 4:47:04"
" 입니다."
</h2>
```

- 변화하는 부분의 엘리먼트만 바뀌는 것을 알 수 있습니다.

핵심

- (1) : 첫 번째 인자에는 JSX로 React 요소를 인자로 넘깁니다.
- (2) : 두 번째 인자는 해당 요소를 렌더하고 싶은 container(부모요소)를 전달합니다.

JSX



- HTML 문법을 JavaScript 코드 내부에 쓴 것
- JavaScript eXtension.. JavaScript의 확장 버전이고 결론은 자바스크립트이다.

```
import React from 'react';
import ReactDOM from 'react-dom';

const name = "KIMJUNGHO";
const hello = <h1>Hello {name}</h1>;
(1)
ReactDOM.render(hello, document.getElementById("root"));
```

HTML를 변수 안에 사용가능하고 함수의 인자로도 넘길 수 있습니다.
태그를 넣을 경우 “ `` `` ` 따옴표를 생략하고 변수를 설정합니다.

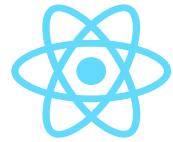
::출력 결과

Hello, KIMJUNGHO

(1) : 변수를 넣어줄 때 자바스크립트와 달리
중괄호 {} 안에 변수를 넣어줍니다.

자바스크립트 변수 전달 : “ + name + ”

Component



- CS에서 다루기로는 하드웨어처럼 “독립적인 기능을 수행하는 단위 모듈”을 의미합니다.
- 소프트웨어의 재사용성을 높이고 유지보수를 용이하게 하기 위해 나온 기술이 바로 **컴포넌트(Component)** 입니다.

```
import React from 'react';
import ReactDOM from 'react-dom';

function Hello(){
  return <h1>Hello, Kimjungho</h1>
}

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

return의 경우 두 줄 이상일 때 **소괄호()**를 사용해서 줄바꿈을 해줍니다.
ex) return (
 <StyledButton color={color} background={background} †>
 {children}
 </StyledButton>

props나 state와 같은 데이터를 입력받아 DOM 노드를 생성하는게 일반적입니다.
(props는 다음 예제에서 다루겠습니다.)



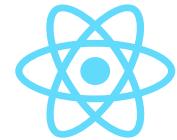
위의 예제의 경우 Component는 Hello입니다.

import Hello from "./xxx/Hello" 를 통해 다른 페이지에 import가 가능합니다.
component는 다른 곳에서 연속적으로 사용이 가능합니다.

:: 출력 결과

Hello, KIMJUNGHO

Props



- React.js에서 props는 Immutable Data 즉, 변하지 않는 데이터입니다.
- 상위 컴포넌트에서 하위 컴포넌트로 데이터를 넘겨줄 때 사용합니다.

```
import React from 'react';
import ReactDOM from 'react-dom';

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

const element = <Hello name="Kimjungho" />;
  ↗
ReactDOM.render(element, document.getElementById('root'));
```

::props 사용방법

<컴포넌트 이름 props이름 = "값">



Immutable Data

불변하는 데이터라는 뜻입니다.

변경될 필요가 없는 부분을 확실히 냉동시켜두는 것이 좋습니다.

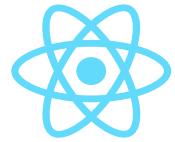
::출력 결과

Hello, KIMJUNGHO

위의 예제의 경우 Component는 Hello입니다.

import Hello from "./xxx/Hello" 를 통해 다른 페이지에 import가 가능합니다.

Router



- URL에 따라서 그에 상응하는 화면을 전송해주는 것입니다.
- 앱에서 발생하는 라우팅이 location과 history 같은 web API 와 연동된다.

:: Router 설치하기(web용)

```
npm install react-router-dom
```

```
<Router>  ←  
    <Link/>    [ ]  
    <Link/>  
    (a)  
    <Router />  
    <Router /> [ ]  
  </Router>
```

Route와 Link 컴포넌트는 DOM트리 상에서 항상 Router를
공통 상위 컴포넌트로 가져야 합니다.

(a) : 실제 프로젝트에서 위의 컴포넌트들은
여러 파일에 걸쳐서 흩어져있을 확률이 높습니다.

★ Link : html의 <a>태그와 유사합니다.

```
<Link to="/about">About</Link>
```

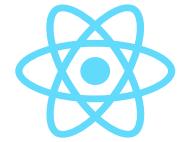
to prop을 통해서 이동할 경로를 지정합니다.

★ Route : 현재 주소창의 경로와 매치될 경우
보여줄 컴포넌트를 지정하는데 사용

```
<Route path="/about" component={About}>
```

path prop을 통해서 매치시킬 경로를 지정합니다.
현재 경로가 /about이기 때문에 About이라는 컴포넌트를 보여줍니다.

MAIN



Portfolio KIMJUNGHO About Reference Youtube Script Contact

WE PROVIDE VISUAL CODING SOLUTIONS FOR YOU WEBS

email dkdwo@naver.com Kakao @dkdwo

social YouTube Instagram GitHub Cafe Reference Tutorials CSS

import 이름 from 위치

```
import React from 'react';
import Header from '../layouts/Header';
import Footer from '../layouts/Footer';
import Contents from '../layouts/Contents';
```

```
function Info({text}){
  return <div>{text}</div>
}

const textInfo = [
  {text : "We Provide"},
  {text : "visual coding"},
  {text : "solution"},
  {text : "for you webs"},
]

function Main() {
  return (
    <div>
      <Header />
      <Contents>
        <section id="mainCont">
          <h2 className="sr-only">메인 컨텐츠 영역입니다.</h2>
          <div className="main__cont">
            {textInfo.map((info) => (
              <Info text={info.text} key={info.text} />
            ))}
          </div>
        </section>
        </Contents>
        <Footer />
      </div>
    )
}

export default Main;
```

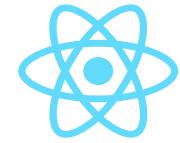
textInfo라는 변수 안에 text라는 key 값 안에
밸류를 각각 지정합니다.

Main이라는 함수 안에 여러 개의
컴포넌트를 가져옵니다.

Main이라는 함수를 이름을 유지한 채 내보냅니다.

- 파일, 설정, 컴포넌트를 외부 파일이나 모듈에서 가져올 때 사용합니다.

About



INTRODUCE ME



```
function AboutText({title, desc, image}){
    return (
        <div class="about">
            <div class="about__img">
                <img src={image} alt={title} />
            </div>
            <div class="about__txt">
                <h3>{title}</h3>
                <p>{desc}</p>
            </div>
        </div>
    )
}
```



```
const aboutInfo = [
  {
    id: '1',
    title: '꿈을 설계하고 디자인하다.',
    desc: '나는 공간을 만드는 것을 좋아한다. 어렸을 때부터 나만의 공간을 만드는 ....',
    image: 'img/aboutImg01.jpg'
  },
  {
    id: '2',
    title: '열심히 할수록 기회는 따른다.',
    desc: '운이 좋은 사람은 없다. 단지 운을 만들 뿐이다. 운을 만들기 위해서는 내가 ....',
    image: 'img/aboutImg02.jpg'
  },
  {
    id: '3',
    title: '나에게 정직하다.',
    desc: '정직은 다른 사람보다 나에게 큰 의미를 부여해야 한다. 자신이 정직하지 ....',
    image: 'img/aboutImg03.jpg'
  }
];

function About() {
  return (
    <div>
      <Header />
      <Contents>
        <section id="aboutCont">
          <div class="container">
            <WrapTitle text={['introduce', 'me']} />
            <div class="about__cont">
              {aboutInfo.map((info) => (
                <AboutText key={info.id} title={info.title} desc={info.desc} image={info.image} />
              ))}
            </div>
          </div>
        </section>
        <ContInfo />
      </Contents>
      <Footer />
    </div>
  )
}
```