Web Studio 2019
9. React.js

김상근 / sisobus1@gmail.com

Contents

1. React
2. Create-React-App
3. Blog (?)

React

A Javascript library for building UI

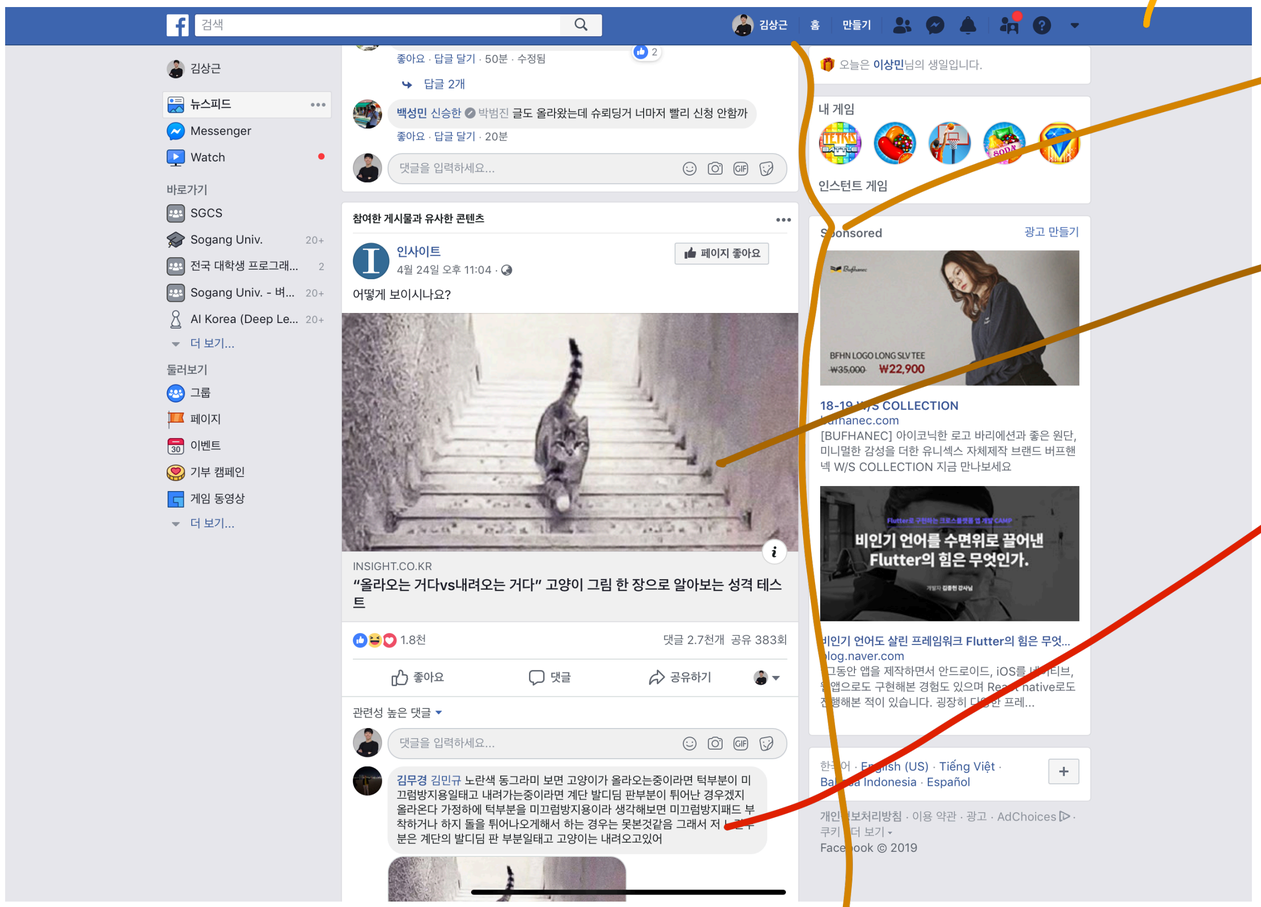
1. Facebook
2. Create-React-App 을 이용한 쉬운 개발환경 setup
3. Component 단위 개발
4. Virtual DOM을 이용한 high performance
   - ~~Memory reconciliation algorithm~~
5. Production level third-party libraries (e.g. ant-design)
6. Unidirectional data flow
7. HTML과 javascript를 합쳐놓은 듯한 jsx 문법을 이용한다.

React

Component 단위 개발

React

Component 단위 개발



<Navigation Component>

<Feeds Component>

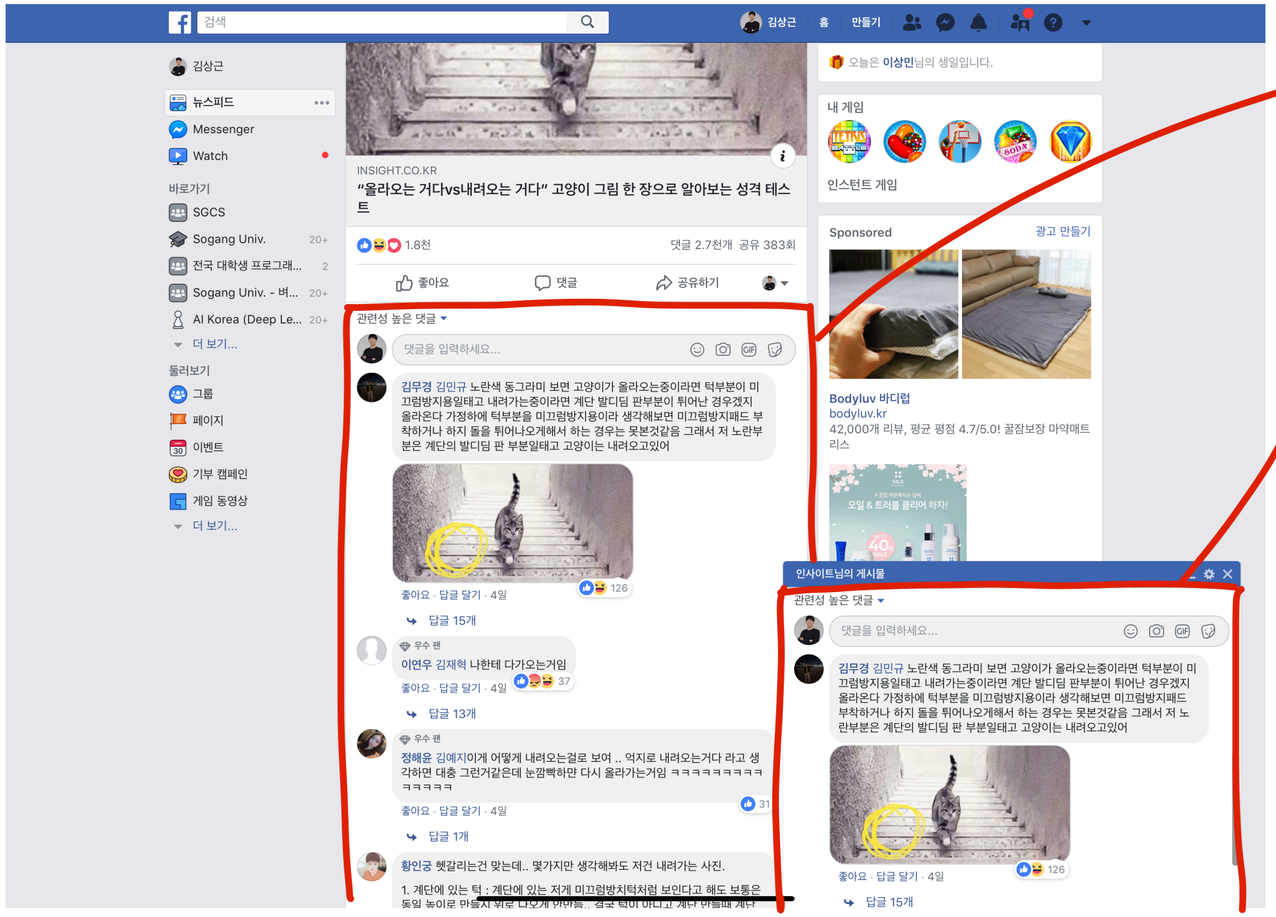
<Feed Component>

<Comments Component>

<Comment Component>

React

Component 단위 개발



<Comments Component>

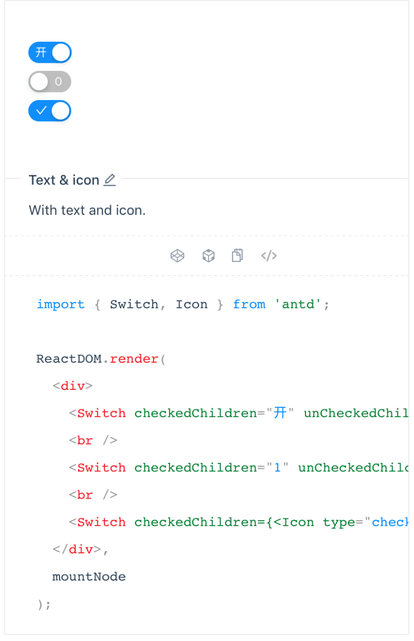
<Comment Component>

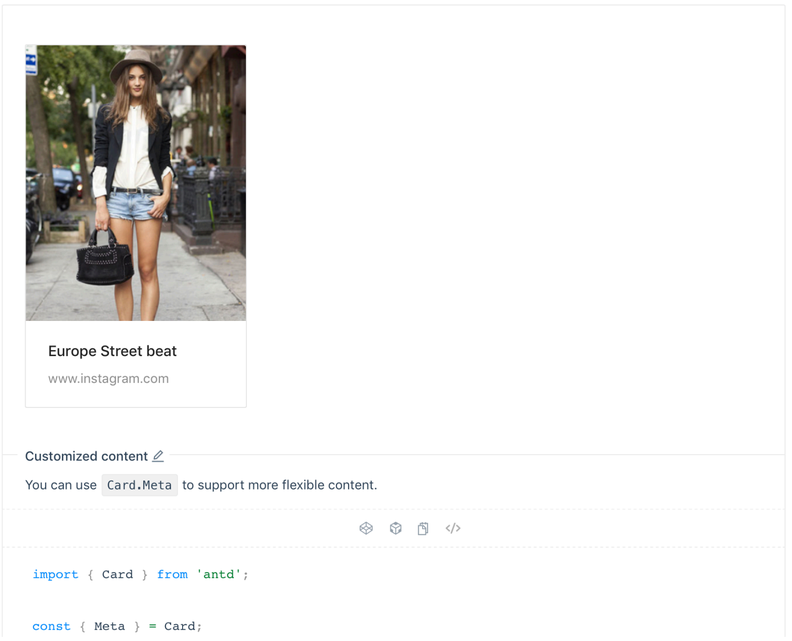
⇒ reusable !!

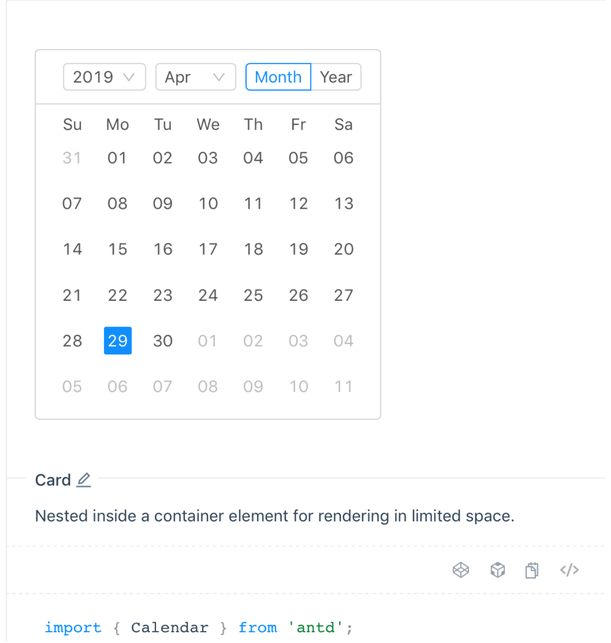
React

Component 단위 개발

1. 결국 여러 Component의 조합 === 하나의 웹사이트
2. Component를 잘 짜면 되겠다
3. React는 Component 단위 개발을 쉽게 해주는 라이브러리구나!
4. 잘 짜여진 Component를 갖다가 조립하면 되겠다 (e.g. ant-design)







React

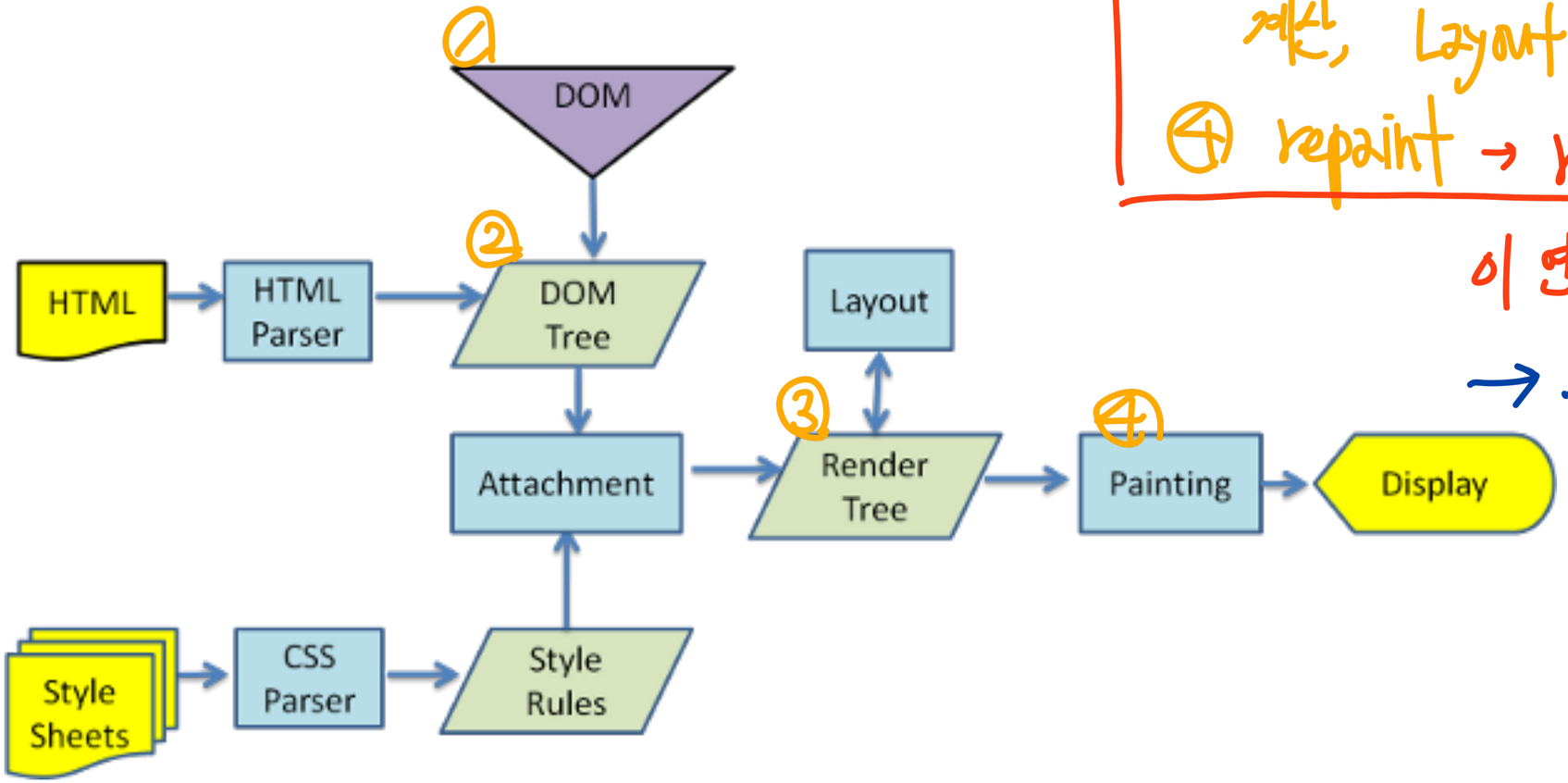
Virtual DOM

1. DOM manipulation은 느리다? ㅇㅇ
2. 왜 느릴까?

```
for (let i = 0; i < response.length; i++) {
  html += `<div class="article" id="article-${response[i].id}">`;
  html += `<h1 class="title">${response[i].title}`;
  html += `<button class="remove-button" id="remove-${response[i].id}">remove</button></h1>`;
  html += `<p class="author">${response[i].author}</p>`;
  html += `<p class="content">${response[i].content}</p>`;
  html += `</div>`;
}
const articleContainer = document.getElementById('article-container');
articleContainer.insertAdjacentHTML('beforeEnd', html);
```

React

브라우저가 화면에 보여주는 과정



① Dom manipulation이 일어남
② Dom tree 변경
③ render tree 확인 후 변경된 node의 계산, Layout 수정 → reflow
④ repaint → repaint

이 연산을 최소화 → 성능↑
→ 가상의 DOM을 만들어 계산하고 실제 DOM연산을 최소화하라!

https://hashnode.com/post/the-one-thing-that-no-one-properly-explains-about-react-why-virtual-dom-cisczhfj41bmssp53mvfwmgrq

React

Virtual DOM

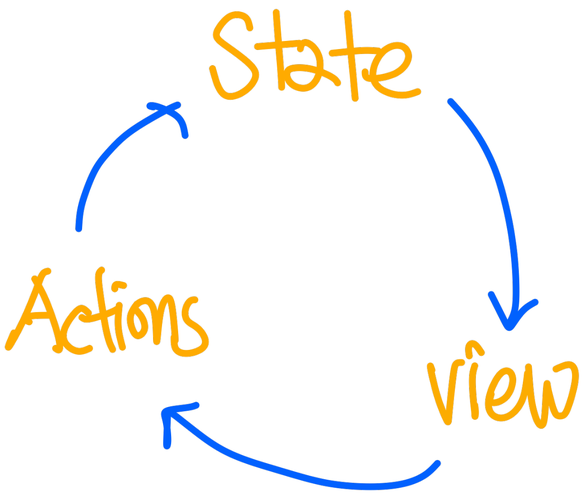
1. React는 Virtual DOM을 이용해 reflow, repaint 연산을 줄임
2. An efficient heuristic algorithm of transform a tree to another for diff
3. 결론: 특정 조건을 지키면 성능도 좋다.

React

Unidirectional Data Flow

반드시 새로운 state 여야함

1. Component에는 state가 존재하고, setState를 통해 변경할 수 있음
2. state는 자식 component의 props로 넘겨줄 수 있음
3. 해당 Component의 state가 변경되면 자식 Component의 props도 변경됨
4. 데이터가 단방향으로 흐른다는 것은 디버깅이 쉽다는 이점이 있음



Create-React-App

One command로 React app 생성해주는 Tool

1. npm 5.2이상 버전이 설치되어 있어야 함
    1. nodejs 최신버전을 설치하면 npm도 설치 됨
2. $ npx create-react-app ⟨Project Name⟩
    1. (e.g. npx create-react-app web-studio)

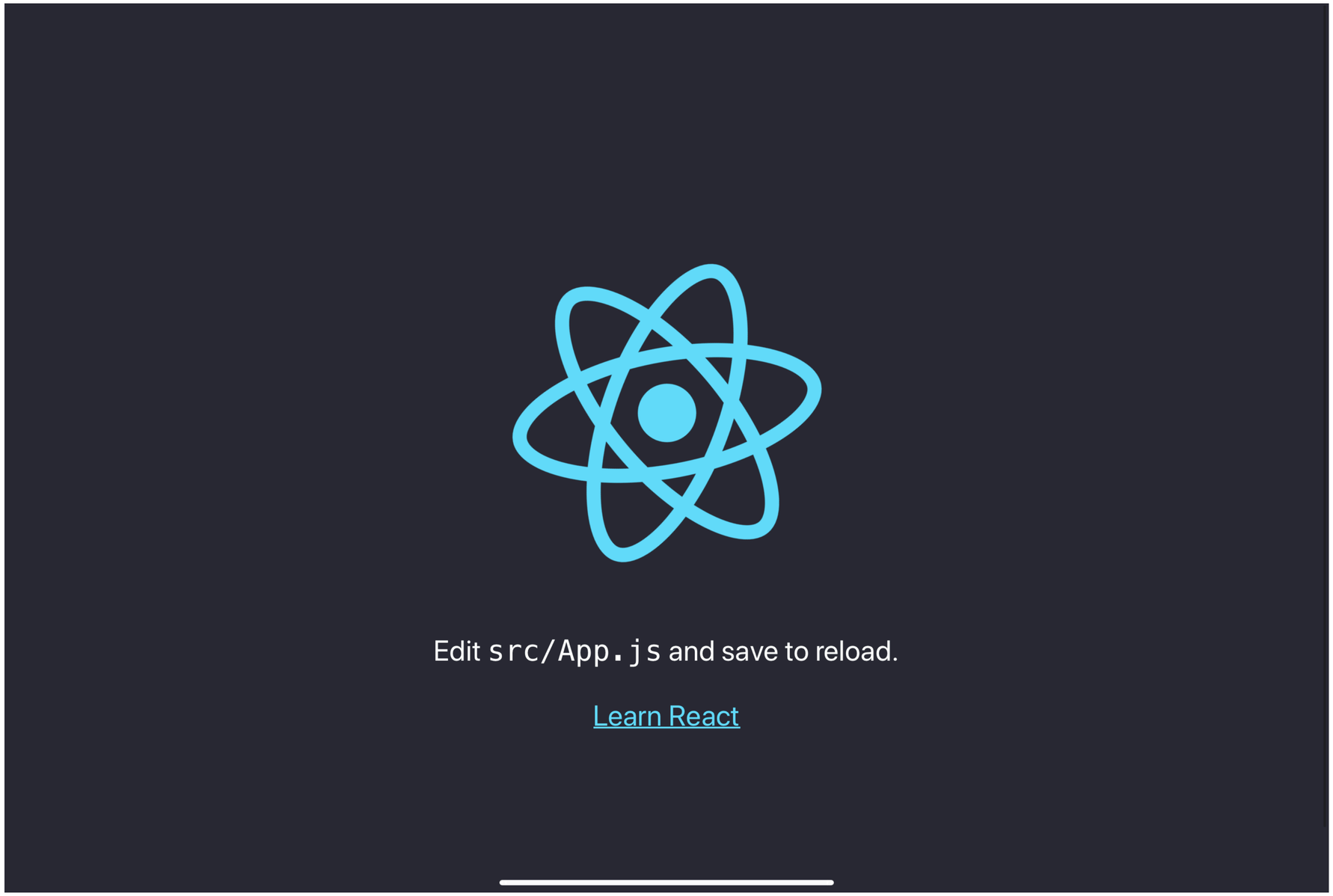
Create-React-App

```
sisobus@ip-172-31-21-135:~$ npx create-react-app web-studio
npx: installed 91 in 4.533s

Creating a new React app in /home/sisobus/web-studio.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts...

+ react-dom@16.8.6
+ react@16.8.6
+ react-scripts@3.0.0
added 1397 packages from 726 contributors in 31.161s

Success! Created web-studio at /home/sisobus/web-studio
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

  cd web-studio
  npm start

Happy hacking!
sisobus@ip-172-31-21-135:~$
```

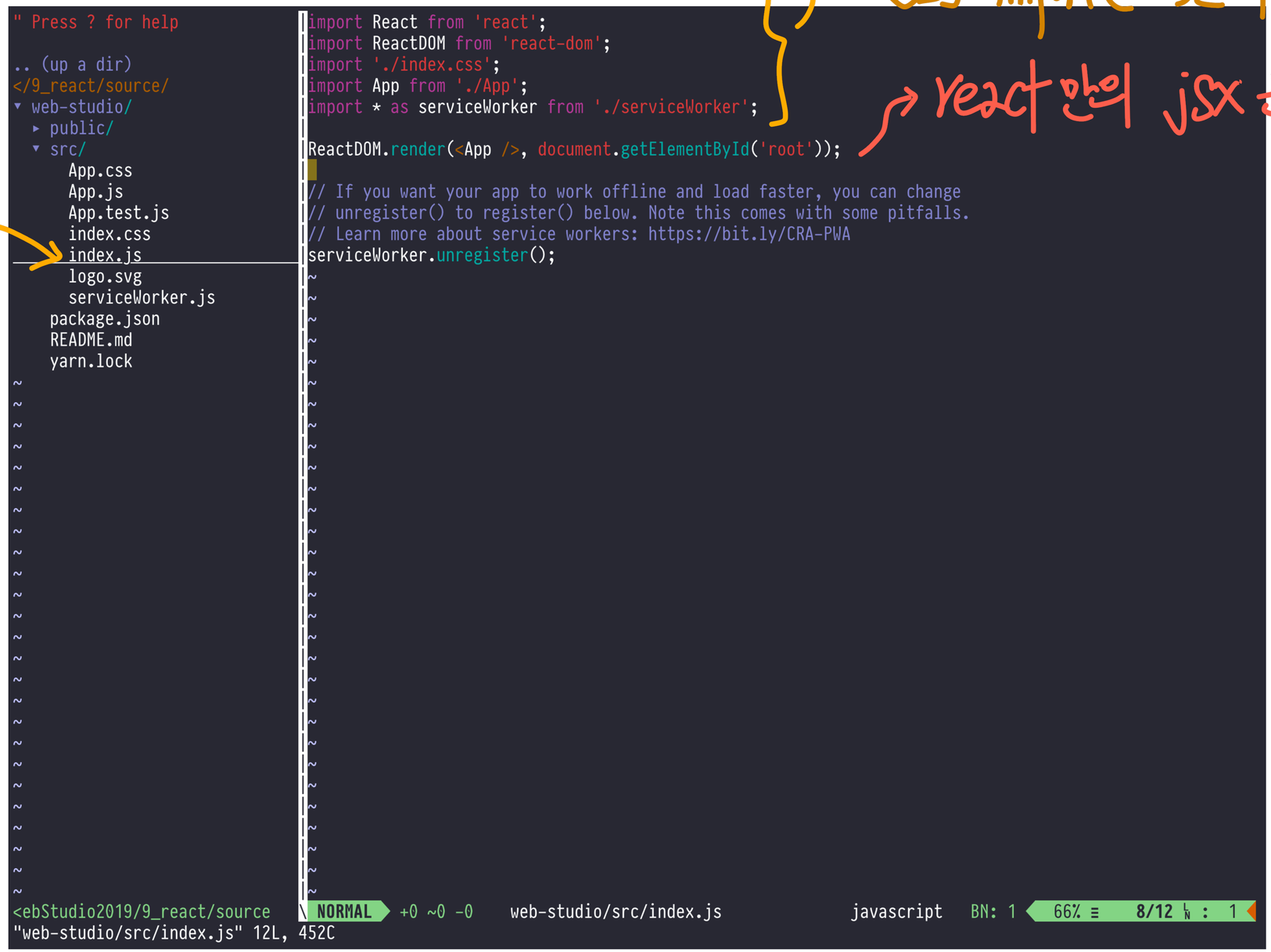
Create-React-App

```
sisobus@ip-172-31-21-135:~$ ls
web-studio
sisobus@ip-172-31-21-135:~$ cd web-studio/
sisobus@ip-172-31-21-135:~/web-studio$ npm run start

Compiled successfully!

You can now view web-studio in the browser.

  Local:            http://localhost:3000/
  On Your Network:  http://172.31.21.135:3000/

Note that the development build is not optimized.
To create a production build, use npm run build.
```

Create-React-App



Create-React-App

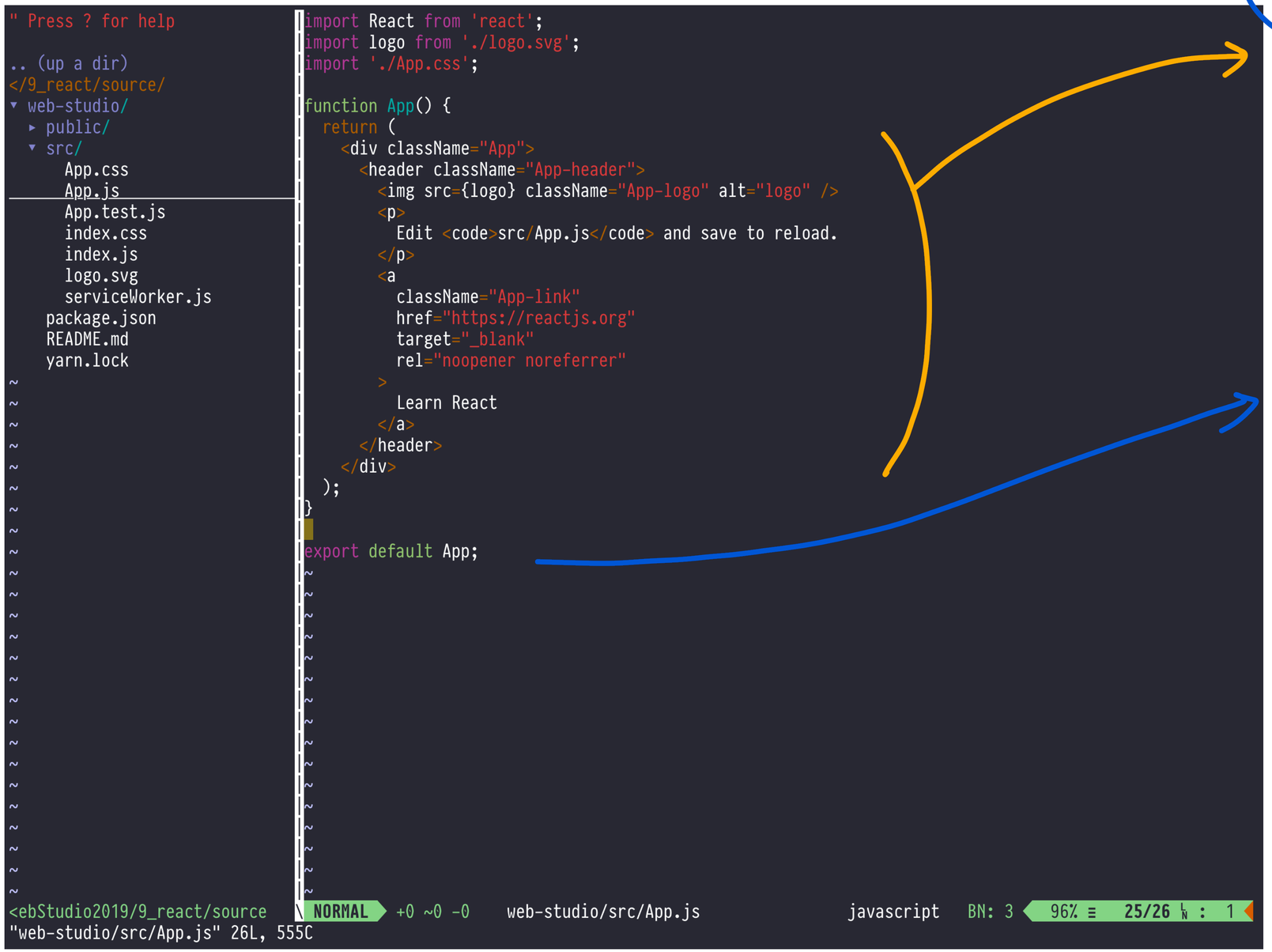


entry point

python의 import와 유사
CSS import는 webpack 플러그인 설치했음.

react만의 JSX 문법

Create-React-App



Attributes 이름이 좀 다름.
(e.g. HTML class → className)

① HTML처럼 사용

② javascript 변수는 중괄호로 표현

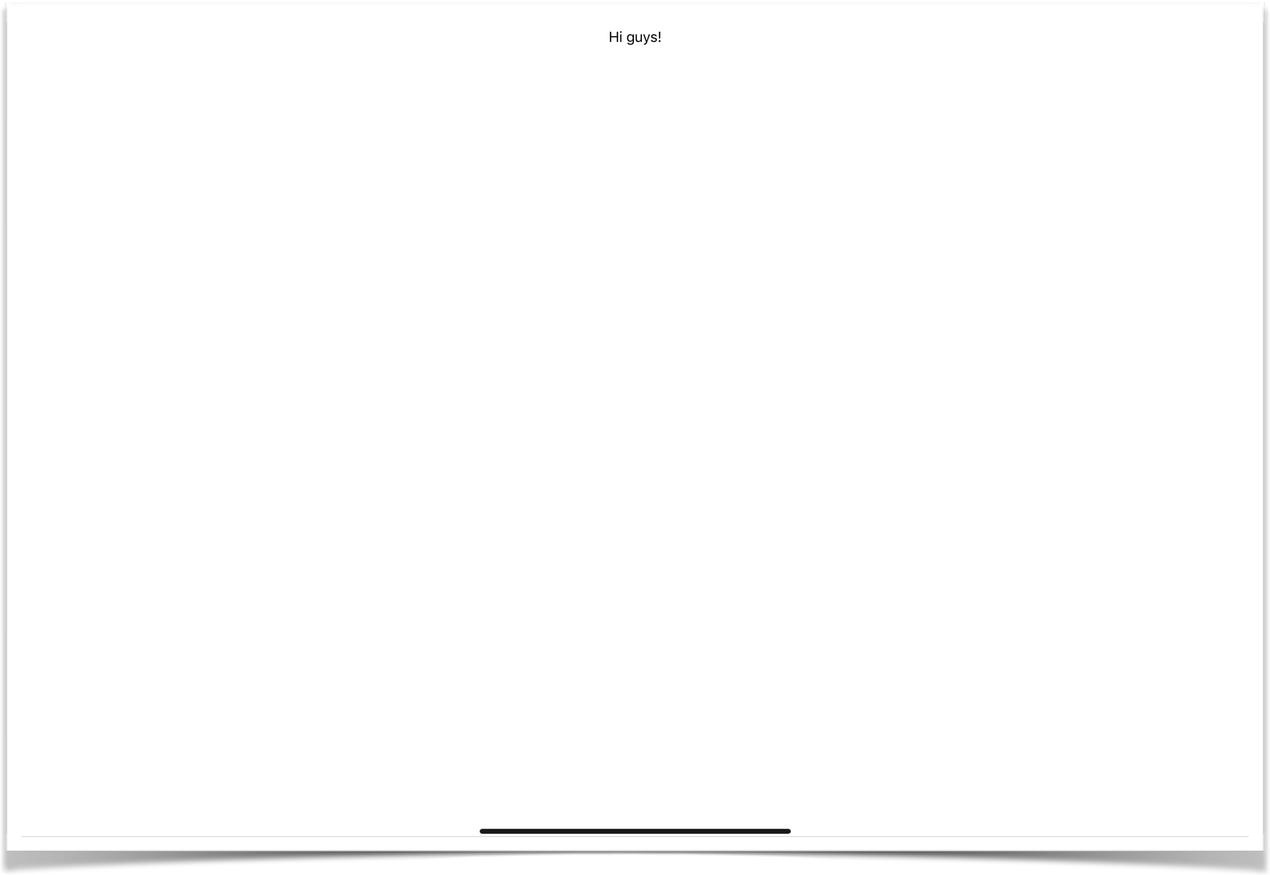
③ 외부코드에서 import해서 쓸 수 있도록

Create-React-App

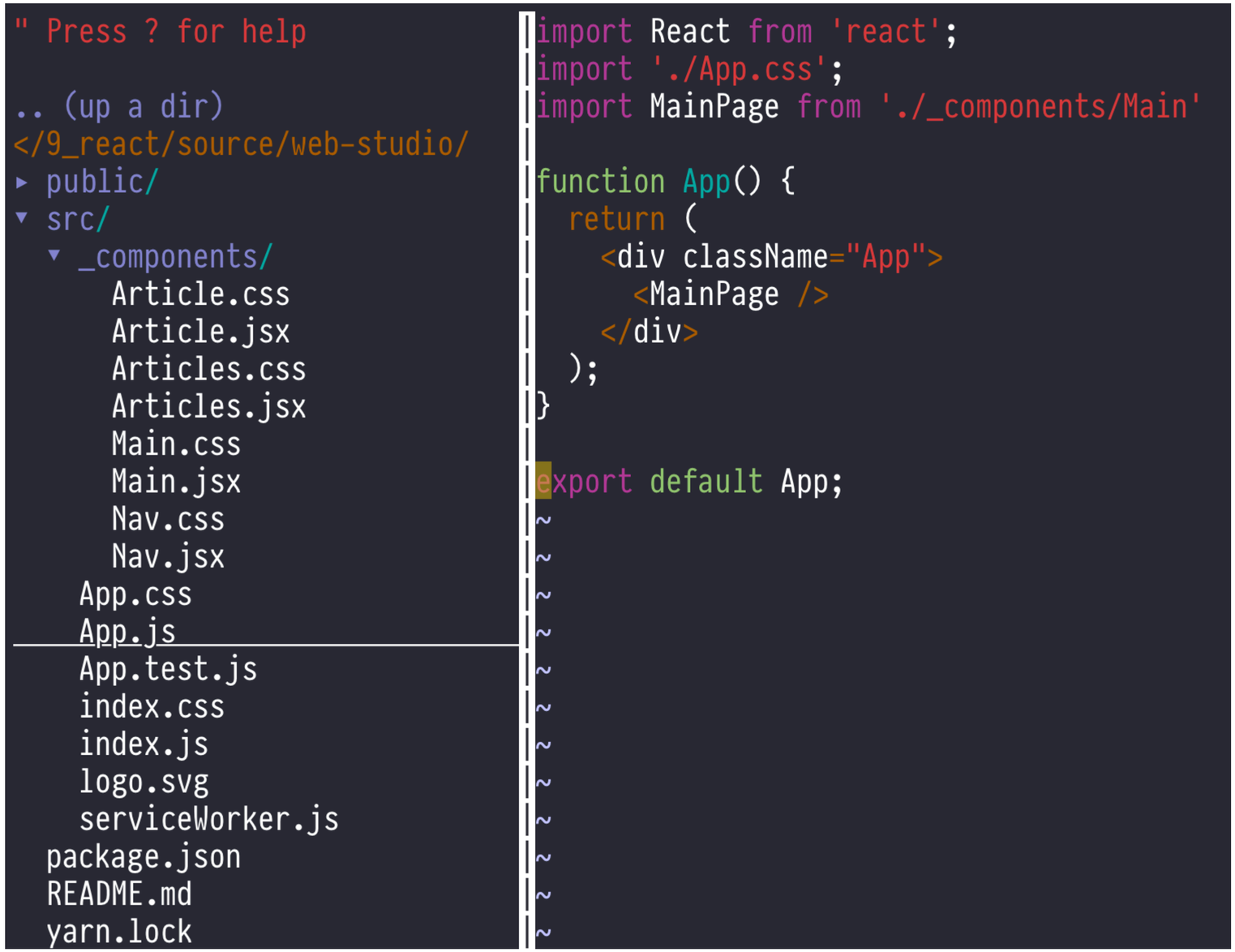
Hello guys!

```
" Press ? for help

.. (up a dir)
/home/sisobus/web-studio/
▸ node_modules/
▸ public/
▾ src/
    App.css
    App.js
    App.test.js
    index.css
    index.js
    logo.svg
    serviceWorker.js
  package-lock.json
  package.json
  README.md
~
~
~
~
~
~
~
~
~
```

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

function App() {
  return (
    <div className="App">
      <p>Hi guys!</p>
    </div>
  );
}

export default App;
~
~
~
~
~
~
~
~
~
~
~
~
```



Blog (?)


" Press ? for help
.. (up a dir)
</9_react/source/web-studio/
▸ public/
▾ src/
▾ _components/
Article.css
Article.jsx
Articles.css
Articles.jsx
Main.css
Main.jsx
Nav.css
Nav.jsx
App.css
App.js
App.test.js
index.css
index.js
logo.svg
serviceWorker.js
package.json
README.md
yarn.lock
import React from 'react';
import './App.css';
import MainPage from './_components/Main'
function App() {
return (

<MainPage />

);
}
export default App;

Blog (?)

```
" Press ? for help

.. (up a dir)
</9_react/source/web-studio/
▸ public/
▾ src/
  ▾ _components/
      Article.css
      Article.jsx
      Articles.css
      Articles.jsx
      Main.css
      Main.jsx
      Nav.css
      Nav.jsx
    App.css
    App.js
    App.test.js
    index.css
    index.js
    logo.svg
    serviceWorker.js
  package.json
  README.md
  yarn.lock
```

```
import React from 'react'
import './Main.css'
import Nav from './Nav'
import Articles from './Articles'

class MainPage extends React.Component {
  render() {
    return (
      <React.Fragment>
        <Nav />
        <div className="main-wrapper">
          <div className="main-container">
            <Articles />
          </div>
        </div>
      </React.Fragment>
    )
  }
}

export default MainPage
```

div 감싼 것

미구현

```
" Press ? for help

.. (up a dir)
</9_react/source/web-studio/
▸ node_modules/
▸ public/
▾ src/
  ▾ _components/
      Article.css
      Article.jsx
      Articles.css
      Articles.jsx
      Main.css
      Main.jsx
      Nav.css
      Nav.jsx
    App.css
    App.js
    App.test.js
    index.css
    index.js
    logo.svg
    serviceWorker.js
  package-lock.json
  package.json
  README.md
  yarn.lock
```

```
.main-wrapper {
  width: 100%;
}
.main-container {
  width: 70%;
  margin: 0 auto;
}
```

Blog (?)

```
" Press ? for help

.. (up a dir)
</9_react/source/web-studio/
▸ node_modules/
▸ public/
▾ src/
  ▾ _components/
      Article.css
      Article.jsx
      Articles.css
      Articles.jsx
      Main.css
      Main.jsx
      Nav.css
      Nav.jsx
    App.css
    App.js
    App.test.js
    index.css
    index.js
    logo.svg
    serviceWorker.js
  package-lock.json
  package.json
  README.md
  yarn.lock
```

```
import React from 'react'
import './Nav.css'

class Nav extends React.Component {
  render() {
    return (
      <div className="navbar-wrapper">
        <div className="navbar-container">
          <p id="logo">LOGO</p>
        </div>
      </div>
    )
  }
}

export default Nav
~
~
~
~
~
~
~
~
~
~
~
~
```

```
" Press ? for help

.. (up a dir)
</9_react/source/web-studio/
▸ node_modules/
▸ public/
▾ src/
  ▾ _components/
      Article.css
      Article.jsx
      Articles.css
      Articles.jsx
      Main.css
      Main.jsx
      Nav.css
      Nav.jsx
    App.css
    App.js
    App.test.js
    index.css
    index.js
    logo.svg
    serviceWorker.js
  package-lock.json
  package.json
  README.md
  yarn.lock
```

```
.navbar-wrapper {
  height: 50px;
  background: #24292e;
  color: hsla(0, 0%, 100%, .7);
}
.navbar-container {
  width: 60%;
  height: 50px;
  margin: 0 auto;
}
#logo {
  font-size: 20pt;
  padding-top: 10px;
}
~
~
~
~
~
~
~
~
~
~
~
~
```

Blog (?)

```
" Press ? for help

.. (up a dir)
</9_react/source/web-studio/
▸ node_modules/
▸ public/
▾ src/
  ▾ _components/
      Article.css
      Article.jsx
      Articles.css
      Articles.jsx
      Main.css
      Main.jsx
      Nav.css
      Nav.jsx
    App.css
    App.js
    App.test.js
    index.css
    index.js
    logo.svg
    serviceWorker.js
  package-lock.json
  package.json
  README.md
  yarn.lock
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
<9/9_react/source/web-studio
```

```jsx
import React from 'react'
import Article from './Article'
import './Articles.css'

class Articles extends React.Component {
  constructor(props) {
    super(props)
    this.state = {
      articles: [
        {
          title: "updated title [1]",
          author: "Sangkeun Kim",
          content: "Lorem ipsum dolor sit amet, consectetur adipiscing elit.",
          like: 0
        },
        {
          title: "updated title [2]",
          author: "Sangkeun Kim",
          content: "Lorem ipsum dolor sit amet, consectetur adipiscing elit.",
          like: 0
        },
      ]
    }
  }
  render() {
    return (
      <div className="article-container">
      {this.state.articles &&
        this.state.articles.map(article => (
          <Article
            title={article.title}
            author={article.author}
            content={article.content}
          />
        ))
      }
      </div>
    )
  }
}

export default Articles
```

```
\ NORMAL  +0 ~0 -0   <ents/Articles.jsx[+]   javascript   BN: 5   100% ≡   42/42 ㏑ :  1   ≡ [44]tra…
```

상위 Component에서 받아온 data
(여기선 아무것도 없음)

Article Component로
data를 넘겨주는 과정
(Article의 props에는
title, auther, content가
있겠구나!!)

* Javascript map : 리스트를 순회하여 각각의 원소를 처리하고 반환하여 새로운 리스트를 만들어 주는 function

```
ex l = [1,2,3].map( x => {
        return  x+5;
    })
// l = [6,7,8]
```

=

```
l = []
t = [1,2,3]
for(let i=0; i<t.length; i++) {
    l.push(t[i]+5);
}

// l = [6,7,8]
```

Blog (?)

```
" Press ? for help

.. (up a dir)
</9_react/source/web-studio/
▸ node_modules/
▸ public/
▾ src/
  ▾ _components/
      Article.css
      Article.jsx
      Articles.css
      Articles.jsx
      Main.css
      Main.jsx
      Nav.css
      Nav.jsx
    App.css
    App.js
    App.test.js
    index.css
    index.js
    logo.svg
    serviceWorker.js
  package-lock.json
  package.json
  README.md
  yarn.lock
```

```jsx
import React from 'react'
import './Article.css'

class Article extends React.Component {
  constructor(props) {
    super(props)
    console.log(props)
  }
  render() {
    return (
      <div className="article">
        <h1 className="title">{this.props.title}</h1>
        <p className="author">{this.props.author}</p>
        <p className="content">{this.props.content}</p>
      </div>
    )
  }
}

export default Article
~
~
~
~
~
~
~
```

Articles Component에서 받은 data

Blog (?)

LOGO

updated title [1]

Sangkeun Kim

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

updated title [2]

Sangkeun Kim

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

updated title [3]

Sangkeun Kim

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Like

* setState는 반드시 새로운 state를 받아와야함.

* [...list]는 원소가 동일한 새로운 list를 반환함

```
l = [1,2,3]    l1 =
l1 = [...l] //[1,2,3]
l2 = []
for(let i=0; i<l.length; i++)
  l2.push(l[i]);
        // l2 = [1,2,3]
```

```
" Press ? for help

.. (up a dir)
</9_react/source/web-studio/
▸ node_modules/
▸ public/
▾ src/
  ▾ _components/
      Article.css
      Article.jsx
      Articles.css
      Articles.jsx
      Main.css
      Main.jsx
      Nav.css
      Nav.jsx
    App.css
    App.js
    App.test.js
    index.css
    index.js
    logo.svg
    serviceWorker.js
  package.json
  README.md
  yarn.lock
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
<9/9_react/source/web-studio
```

```jsx
import React from 'react'
import Article from './Article'
import './Articles.css'

class Articles extends React.Component {
  constructor(props) {
    super(props)
    this.state = {
      articles: [
        ...
      ]
    }
    this.handleLike = this.handleLike.bind(this)
  }
  handleLike(idx) {
    let nextStateArticle = [...this.state.articles]
    nextStateArticle[idx].like = nextStateArticle[idx].like + 1
    this.setState({
      articles: nextStateArticle
    })
  }
  render() {
    return (
      <div className="article-container">
      {this.state.articles &&
        this.state.articles.map((article, idx) => (
          <Article
            title={article.title}
            author={article.author}
            content={article.content}
            like={article.like}
            key={idx}
            idx={idx}
            onClick={this.handleLike}
          />
        ))
      }
      </div>
    )
  }
}
```

생략!

handleLike 함수에서 this를 사용할 수 있도록 binding.

```
NORMAL  +0 ~1 -17   <nts/Articles.jsx[+]   javascript   BN: 1   23% ≡   10/43 ㏑ : 11   ≡ [27]tra…
```

Like

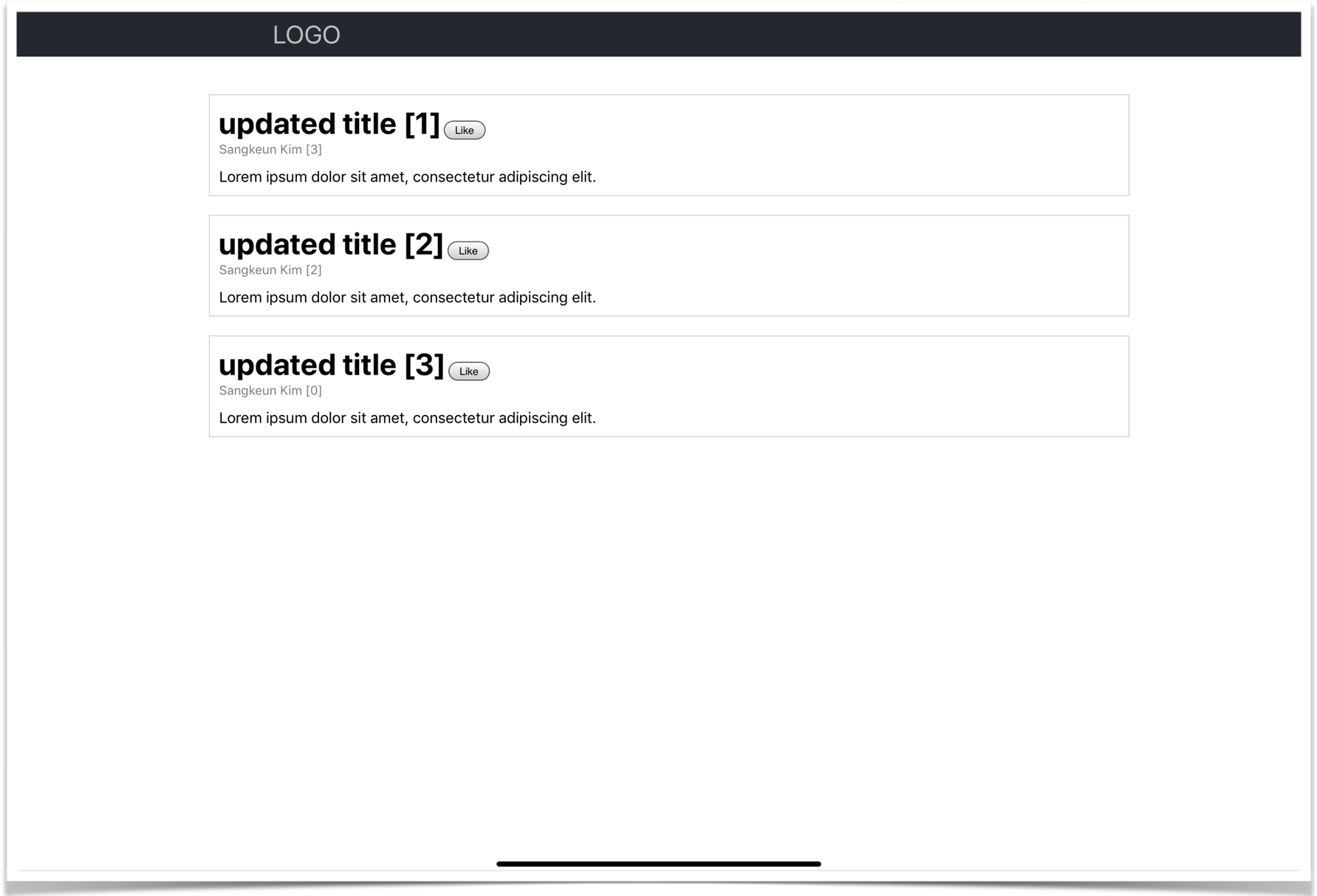
```
" Press ? for help

.. (up a dir)
</9_react/source/web-studio/
▸ node_modules/
▸ public/
▾ src/
  ▾ _components/
      Article.css
      Article.jsx
      Articles.css
      Articles.jsx
      Main.css
      Main.jsx
      Nav.css
      Nav.jsx
    App.css
    App.js
    App.test.js
    index.css
    index.js
    logo.svg
    serviceWorker.js
  package.json
  README.md
  yarn.lock
~
~
~
```

```jsx
import React from 'react'
import './Article.css'

class Article extends React.Component {
  constructor(props) {
    super(props)
    console.log(props)

    this.handleClick = this.handleClick.bind(this)
  }
  handleClick() {
    this.props.onClick(this.props.idx)
  }
  render() {
    return (
      <div className="article">
        <h1 className="title">{this.props.title}
          <button
            onClick={this.handleClick}
          >Like</button>
        </h1>
        <p className="author">{this.props.author} [{this.props.like}]</p>
        <p className="content">{this.props.content}</p>
      </div>
    )
  }
}

export default Article
```

handleClick에서 this를 사용할 수 있도록 binding

상위 Component에서 받아는 data (function도 변수취급)

Like


LOGO
updated title [1]
Like
Sangkeun Kim [3]
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
updated title [2]
Like
Sangkeun Kim [2]
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
updated title [3]
Like
Sangkeun Kim [0]
Lorem ipsum dolor sit amet, consectetur adipiscing elit.

실습

branch: 9-practice-〈UsernameEn〉

1. ~~/9_react/practice/〈UsernameEn〉 에 다음 명령어로 react app 생성
    1. $ npx create-react-app web-studio
2. 아래 명령어로 실행해본다.
    1. $ npm run start
3. 브라우저로 접속해본다.
4. 수업에서 한 component들을 작성(혹은 복붙)하여 똑같이 실행해본다.
5. Article.jsx안의 this.state를 수정하여 본인의 글을 작성해본다.
6. PR을 날린다.

Q & A