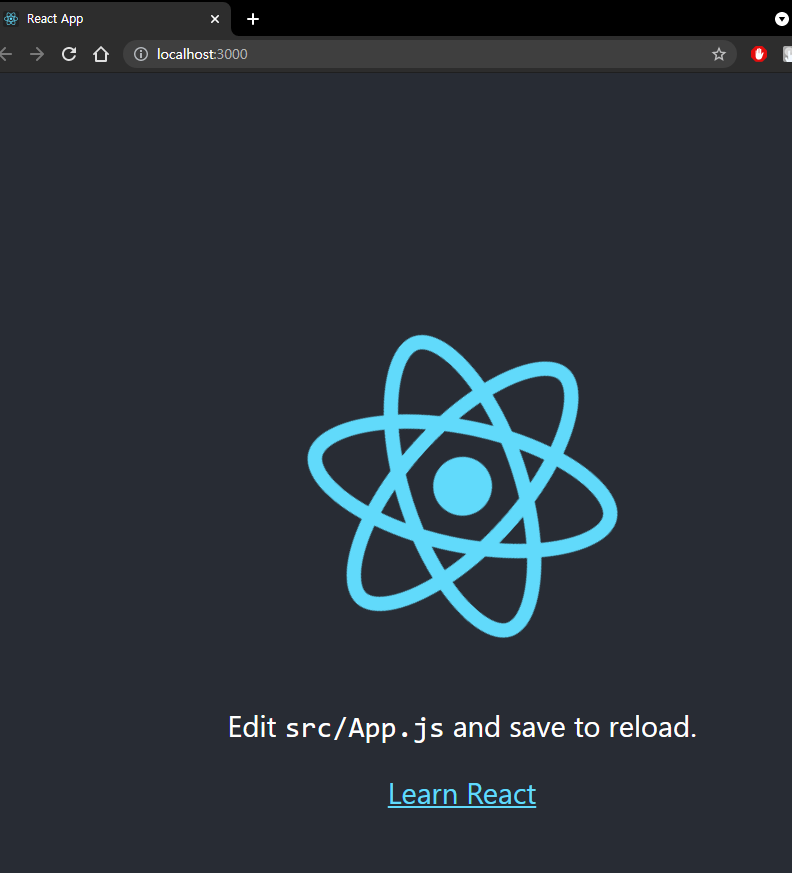
[react-app 실행]

React가 설치된 폴더에서 터미널로 npm run start 치면 좀 걸리다가 react app이 실행된다.

텍스트이(가) 표시된 사진

자동 생성된 설명 터미널에서 이렇게 뜬다.

주소들은 이제 내가 만들고 있는 react 앱을 확인하는 주소로 써먹으면 된다.

브라우저도 자동으로 샘플 페이지가 떴다.

끌려면 ctrl+c 해주면 된다.

[react 코딩하기]

텍스트이(가) 표시된 사진

자동 생성된 설명Create-react-app이 이것저것 폴더에 엄청나게 많이 만들어줬는데, 대체 어떤 부분을 만져야 뭘 할 수 있는것인지 아무것도 알지못한다. 그래서 어디에서 코딩을 해야하는지 알아보도록 하자.

좌측을 보자. 경로들이 나와있다.

텍스트이(가) 표시된 사진

자동 생성된 설명React-app은 크게 public과 src로 구분된다.

그 중에서 public에 있는 곳에 있는 파일들이 아까 봤던 react 샘플 페이지를 구성하는 요소들이다.

그러면 index.html 코드를 보고 어떤 내용들이 있는지 확인해보자.

  <body>

    <noscript>You need to enable JavaScript to run this app.</noscript>

    <div id="root"></div>

    <!--

      This HTML file is a template.

      If you open it directly in the browser, you will see an empty page.

      You can add webfonts, meta tags, or analytics to this file.

      The build step will place the bundled scripts into the <body> tag.

      To begin the development, run `npm start` or `yarn start`.

      To create a production bundle, use `npm run build` or `yarn build`.

    -->

  </body>

이런 식으로 코드가 작성되어 있다. 그런데 여기서 가장 중요한 부분은 id=”root”인 div이다.

리액트에서 각종 html 태그들, 그러니까 <div>와 같은 태그들은 컴포넌트라고 부른다.

리액트를 통해서 컴포넌트들을 만들텐데, 그런 컴포넌트들은 저 root 안에 들어가도록 create-react-app은 약속해뒀다. 수동적으로 바꿀수도 있지만.

어찌됐건, 어떠한 소스들을 짜서 우리는 약속된 저 <div id=”root”></div> 태그 내에 컴포넌트들을 꼽아 넣는 것이다. 그럼 그 소스들은 어디에 짜면 될까?

바로텍스트이(가) 표시된 사진

자동 생성된 설명 이와 같은 src 폴더 내에 짜면 된다.

앞으로 개발을 하면서 필요한 것들은 대부분 src 폴더를 통해서 구현하게 될 것이다.

이중에서 중요한 것은 index.js이다.

텍스트이(가) 표시된 사진

자동 생성된 설명index.js가 이런 모양새로 되어있을 것이다. 세세한 부분은 조금 달라질 수 있을지 몰라도 아무튼

ReactDOM.render(

  <React.StrictMode>

    <App />

  </React.StrictMode>,

  document.getElementById('root')

);

이 코드에서 document.getElementById(‘root’) 코드를 보면 root id를 가진 element를 html 문서로부터 가져오는 것임을 알 수 있다.

즉 여기서 아까 우리가 봤던 index.html 파일의 <div id=”root”></div>를 지정해서 거기에 컴포넌트들을 찍어내는 것이다.

    <App />

그리고 이게 사용자 컴포넌트. 우리가 짜는 코드이다.

그럼 저게 어디있는가?

import App from './App';

여기있지 않은가. App.js라는 말이다.

이 파일 말이다. 들어가보자.

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;

들어가면 위와 같이 코드들이 나올텐데, function 방식일수도 있고, 혹시 class 방식일수도 있다.

지금은 class 방식으로 만지는 것을 배울 것이므로 class 방식으로 바꿔보자.

다음 페이지에서 class 방식으로 바꿔보겠다.

import logo from './logo.svg';

import './App.css';

import { Component } from 'react';

class App extends Component {

  render() {

    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;

자 이렇게 바꿨다. 좀 헷갈릴수도 있지만 render() 함수를 구현한 것이다. 왜 이런것인지는 잘 모르겠다.

App이 constructor function일 경우에도 prototype에 render 함수를 추가한다는 말은 없었는데.. 뭐 아무튼..

class App extends Component {

  render() {

    return (

      <div className="App">

        Hello, React!!

      </div>

    );

  }

}

그럼 이 부분을 이렇게 바꾸면 Hello React가 뜰 것임은 자명하다.

그런데 여기서 주의해야 할 부분이 있다.

코드나, 이미지가 너무 많아서 페이지가 많아지지만 일단 그 점은 감안해주고 다음 페이지로 가보자.

  render() {

    return (

      <div className="App">

        Hello, React!!

      </div>

    );

이 부분 있지 않은가. App 클래스의 render 메소드의 return값 부분.

이 부분에서 그냥 일반 텍스트를 return 할 수는 없다. 최소한 1개 태그 이상을 return을 해줘야 한다.

어찌보면 당연한 것일수도 있다. 태그를 뱉어내는 함수를 정의한 것인데 텍스트를 뱉을 수는 없으니까.

  render() {

    return (

        Hello, React!!

    );

  }

이런 식으로 텍스트만 리턴할 수는 없다. Return에 소괄호 들어간거니까 신경쓰지마라.