Webpack

코드들을 의존하는 순서대로 합쳐서 하나의 또는 여러 개의 결과물을 만들어낸다.

일일이 HTML을 불러오는게 아니라 자바스크립트 파일에서 Import를 통해서 번들링 작업을 통해 불러와준다.

-원한다면 규

ES6 모던 자바스크립트

구형브라우저에서 일부가 지원이 안된다.

바벨이라는 것을 사용.하여 구형 자바스크립트도 지원하게된다.

Sass 사용 할 수 도 있다.

바벨-자바스크립트 변환도구 이다.

-https://babeljs.io/

[https://codesandbox](https://codesandbox.io/s/4r6lqrlvj9)

<https://bit.ly/beginreact>

import React, { Component } from 'react';

class App extends Component {

render() {

return (

<div>

<h1>안녕하세요 리액트</h1>

</div>

);

}

}

export default App;

사용 할 때 import React를 꼭 해줘야한다.

컴포넌트를 만드는 두가지 방법

1. 클래스를 통해서 만드는 방법이 가장 일반적인 방법

import React, { Component } from 'react';

class App extends Component {

render() {

return (

<div>

<h1>안녕하세요 리액트</h1>

</div>

);

}

}

JSX 기반의 형태로 함수를 리턴 해줘야한다.

1. 함수를 통해서 만드는 것.

JSX는

태그가 꼭 닫혀서 사용되어야 한다.

<input type=”button”/>닫아줘야 오류가 안난다.

위에처럼 닫아주는 마지막 표시 기호를 셀프클로징태그라고한다.

두개 이상의 엘리먼트는 무조건 하나의 엘리먼트로 감싸져있어야한다.

이렇게 두개가 쓰는게 귀찮거나 불편하게 된다면 Fragment를 사용해서 간단하게 할 수 도 있다.

16.2v에서 사용가능

<div>H</div>

<dive>S</div>

위의 코드처럼 작성하면 에러가 뜬다 하지만. 밑의 코드처럼 하면 에러가 안뜬다.

<div>

<div>H</div>

<dive>S</div>

</div>

FRAGMENT 사용

import React, { Component,Fragment } from 'react';

class App extends Component {

render() {

return (

<Fragment>

<h1>안녕하세요 리액트</h1>

<input type="text"/>

</Fragment>

);

}

}

export default App;

JSX 안에 자바스크립트 값 사용하기

import React, { Component,Fragment } from 'react';

class App extends Component {

render() {

const name='react';

return (

<div>

hello{name}

</div>

);

}

}

export default App;

const

let

스코프가 블록 단위이다. 둘다.

Var vs const vs let

Var는 잘 사용하지 않는다.

Const는 한번 선언 후 고정적인 값에 사용

Let은 유동적인 값을 사용할 때 선언

조건부 렌더링

JSX내부에서 조건부 렌더링을 할 때는 보통 삼항 연산자를 사용하거나, AND 연산자를 사용합니다. 반면에 if문을 사용 할 수 는 없습니다. 사용하려면 IIFE(즉시실행함수 표현)을 사용해야합니다.

import React, { Component,Fragment } from 'react';

class App extends Component {

render() {

const name='veloopoart';

return (

<div>{

1+1===3

?'맞다'

:'틀리다!'

}

</div>

);

}

}

export default App;

조건부 if를 엔드를 사용해서 나타낸 경우 조건부 렌더링

import React, { Component,Fragment } from 'react';

class App extends Component {

render() {

const name ='veloport!';

return (

<div>{

name==='veloport!'&& <div>벨로퍼티</div>

}

</div>

);

}

}

export default App;

일치 하지 않으면 화면에 표시 되지 않는다.

조건이 여러 개인 경우에는 JSX 밖에 부분에서 하는 것이 일반적이다.

JSX에서 내부에서 여러가지 조건을 적용하려면 IIFE를 함수 선언하고 바로 호출해서 사용 할 수 있다.

import React, { Component,Fragment } from 'react';

class App extends Component {

render() {

const value =1;

return (

<div>{

function(){

if(value===1)return <div>1</div>

if (value ===2) return <div>2</div>

if (value ===3) return <div>3</div>

return <div>없다.</div>

}

}

</div>

);

}

}

export default App;

함수를 호출하기 전의 코드 함수 호출후의 코드는 아래와 같다.

import React, { Component,Fragment } from 'react';

class App extends Component {

render() {

const value =1;

return (

<div>{

(function(){

if(value===1)return <div>1</div>

if (value ===2) return <div>2</div>

if (value ===3) return <div>3</div>

return <div>없다.</div>

})()

}

</div>

);

}

}

리액트에서 CSS를 삽입 할 때는 객체 상태로 입력해준다.

Const style=등등

import React, { Component } from 'react';

class App extends Component {

render() {

const style={

backgroundColor:'black',

padding:'16px',

color:'white',

fontSize:'36px'

};

return (

<div style={style}>

<h1>안녕하세요 리액트</h1>

</div>

);

}

}

export default App;

위의 형식처럼 스타일을 적용해서 사용 할 수 있다.

JSX에서 클래스를 사용 하는 방법.

1. import를 해서 적용시키기
2. div className=”App(파일명)”를 통해서 적용이가능하다.

import React, { Component } from 'react';

import './App.css';

class App extends Component {

render() {

return (

<div className="App">

<h1>안녕하세요 리액트</h1>

</div>

);

}

}

export default App;

-------App.css파일을 만들어서 위의 소스에 적용

.App{

background:black;

color: aqua;

font-size: 36px;

padding: 1rem;

font-weight:600;

}

--리액트 주석처리방법

import React, { Component } from 'react';

import './App.css';

class App extends Component {

render() {

return(

<div>

{/\*주석처리 방법이다.\*/}

<h1

//내가여기에 주석을 달거야

>리액트</h1>

</div>

);

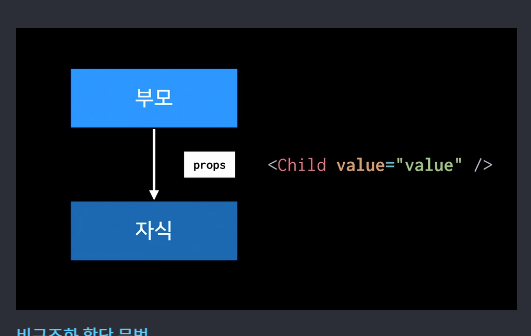
}

}

export default App;

Props와 State

리액트에서 데이터를 다룰 때 사용하는 것이다.



부모 컴포넌트가 자식 컴포넌트한테 데이터를 전달할 때 props를 사용한다.

사용예시

import React, { Component } from 'react';

class MyName extends Component {

static defaultProps = {

name: '기본이름'

};

render() {

return (

<div>

안녕하세요!제이름은<b>{this.props.name}</b>에요.

</div>

);

}

}

export default MyName;

-------------파일을 두개 생성해서 상속 받을 수 있도록 하는 예제

import React, { Component } from 'react';

import MyName from './MyName';

class App extends Component {

render() {

return<MyName name="리액트"/>;

}

}

export default App;

별다른 기능없이 값을 받아와서 보여주는 경우에는 보통 함수형 자바스립트를 사용한다.

import React from 'react';

const MyName = ({ name }) => {

return (

<div>

안녕하세요! 제 이름은 {name} 입니다.

</div>

);

};

export default MyName;