# React - 컴포넌트

## 목표

리액트 개발환경을 구축할 수 있다.

리액트의 본질인 컴포넌트를 이해할 수 있다.

## 전체 소스코드

<https://github.com/egoingsb/react-component-tutorial.git>

## React 개발환경

### stackblitz

온라인에서 리액트를 개발할 수 있는 환경을 제공

<https://stackblitz.com/>

### codesandbox

온라인에서 리액트를 개발할 수 있는 환경을 제공

https://codesandbox.io/

### create-react-app

리액트 개발환경을 쉽게 구축하도록 도와주는 툴킷

https://create-react-app.dev/

#### 설치

npx create-react-app my-react-app

cd my-react-app

npm start

#### 실행

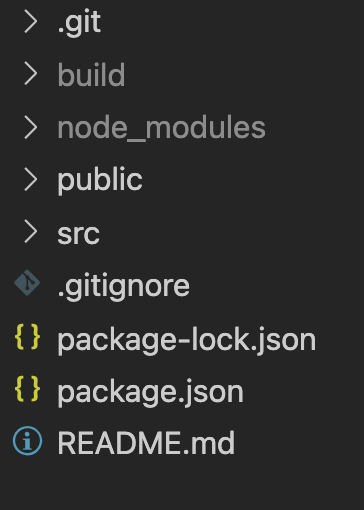
npm run start

#### 배포용 컴파일

npm run build

npx serve -s build

#### 전체 디렉토리 구조



* src : 개발 소스코드가 위치합니다.
* public : react app을 감싸는 html 자원이 위치합니다.
* build : npm run build 의 결과로 만들어지는 배포용 자원이 위치합니다.

#### src 디렉토리 구조

## 

* index.js : 리액트 실행 진입점으로 전역적인 설정 등은 이곳에서 작성하면 됩니다.
* App.js : 리액트 컴포넌트 실행 진입점으로 구체적인 컴포넌트는 이곳에서 작성하기 시작하면 됩니다.

## React의 본질

사용자 정의 태그를 만들기

### 리엑트 이전 코드

| <header>  <h1><a href="/">WEB</a></h1>  </header>  <nav>  <ol>  <li><a href="1.html">html</a></li>  <li><a href="2.html">css</a></li>  <li><a href="3.html">js</a></li>  </ol>  </nav>  <article>  <h2>Welcome</h2>  Hello, WEB  </article> |
| --- |

리액트 적용 코드

<https://github.com/egoingsb/react-component-tutorial/commit/2ccd2a066a3c2e79c64f2a17b850efca3d0d11a6>

| function Header(){  return <header>  <h1><a href="/">WEB</a></h1>  </header>  }  function Nav(){  return <nav>  <ol>  <li><a href="1.html">html</a></li>  <li><a href="2.html">css</a></li>  <li><a href="3.html">js</a></li>  </ol>  </nav>  }  function Article(){  return <article>  <h2>Welcome</h2>  Hello, WEB  </article>  }  function App() {  return (  <>  <Header></Header>  <Nav></Nav>  <Article></Article>  </>  );  } |
| --- |

* Header, Nav, Article 과 같은 사용자 정의 태그를 리액트에서는 **component**라고 부릅니다.
* 컴포넌트를 이용해서 코드를 정리정돈하고, 가독성을 높이고, 재사용성을 증대시킵니다.

## 컴포넌트 만들기

### 문법

컴포넌트를 만드는 방법은 크게 두가지입니다.

#### 함수 방식

| class MyComponent extends React.Component {  render() {  return <h1>Class Style Component</h1>;  }  } |
| --- |

#### 클래스 방식

| function MyComponent{  return <h1>Function Style Component</h1>  } |
| --- |

### 규칙

* 컴포넌트의 이름은 반드시 대문자로 시작해야 합니다.   
  nav => html 네이티브 태그  
  Nav => 리액트 컴포넌트
* 컴포넌트의 최상의 태그는 하나입니다.   
  아래 코드는 최상위 태그가 3개이기 때문에 오류가 발생합니다.

| function App() {  return (  <Header></Header>  <Nav></Nav>  <Article></Article>  );  } |
| --- |

## Props

### 기본형식

컴포넌트의 입력 인터페이스를 제공

<https://github.com/egoingsb/react-component-tutorial/commit/c0923c19be5044eb2176fa1bf07c2eb30cdd8cb9>

| <Article title="Welcome" body="Hello, WEB"></Article> |
| --- |

| function Article(props){  return <article>  <h2>{props.title}</h2>  {props.body}  </article>  } |
| --- |

### 목록형 데이터

<https://github.com/egoingsb/react-component-tutorial/commit/1a90629b2e6faa61ae04d83d74893a9263c3b46a>

| function App() {  let topics = [  {id:1, title:'HTML', body:'HTML is ...'},  {id:2, title:'CSS', body:'CSS is ...'},  {id:3, title:'JavaScript', body:'JavaScript is ...'}  ];  return (  <>  <Header></Header>  <Nav topics={topics}></Nav>  <Article title="Welcome" body="Hello, WEB"></Article>  </>  );  } |
| --- |

| function Nav(props){  let lis = [];  for(let i=0; i<props.topics.length; i++){  let t = props.topics[i];  lis.push(<li key={t.id}><a href={'/read/'+t.id}>{t.title}</a></li>)  }  return <nav>  <ol>  {lis}  </ol>  </nav>  } |
| --- |

* 배열에 컴포넌트를 담아서 리턴하면 태그들로 변환됩니다.
* 목록을 자동으로 생성하면 리액트가 항목을 추적할 수 있도록 key 값을 제공해줘야 합니다. key값은 해당 목록 안에서 고유해야 합니다.

## 이벤트 프로그래밍

props로 함수를 전달하고, 컴포넌트 내에서 함수를 호출한다.

<https://github.com/egoingsb/react-component-tutorial/commit/1caec41d4deeca036f09f41e364d0a451a05923b>

| function App() {  let topics = [  {id:1, title:'HTML', body:'HTML is ...'},  {id:2, title:'CSS', body:'CSS is ...'},  {id:3, title:'JavaScript', body:'JavaScript is ...'}  ];  function onChangeHeaderHandler(){  console.log('Welcome');  }  return (  <>  <Header onChangeMode={onChangeHeaderHandler}></Header>  <Nav topics={topics}></Nav>  <Article title="Welcome" body="Hello, WEB"></Article>  </>  );  } |
| --- |

| function Header(props){  function clickHandler(event){  event.preventDefault();  props.onChangeMode();  }  return <header>  <h1><a href="/" onClick={clickHandler}>WEB</a></h1>  </header>  } |
| --- |

* 네이티브 HTML에서는 onclick이지만 리액트는 onClick 입니다. 리액트의 HTML 코드는 HTML과 유하사지만 동일하지 않습니다.
* 이벤트를 통해서 부모 컴포넌트에 변화를 줄 수 있습니다.

## State

지역변수를 state로 변경하면 state의 값이 바뀌었을 때 컴포넌트가 다시 랜더링됩니다.

<https://github.com/egoingsb/react-component-tutorial/commit/ee186ef1b7aa6613118de7251b08d5d10b3e88a5>

| import {useState} from 'react';  function App() {  let topics = [  {id:1, title:'HTML', body:'HTML is ...'},  {id:2, title:'CSS', body:'CSS is ...'},  {id:3, title:'JavaScript', body:'JavaScript is ...'}  ];  let [mode, setMode] = useState('WELCOME');  let [id, setId] = useState(null);  function onChangeHeaderHandler(){  setMode('WELCOME');  }  function onChangeNavHandler(id){  setMode('READ');  setId(id);  }  let articleTag = '';  if(mode === 'WELCOME'){  articleTag = <Article title="Welcome" body="Hello, WEB"></Article>  } else if(mode === 'READ'){  let title = '';  let body = '';  for(let i=0; i<topics.length; i++){  if(topics[i].id === id){  title = topics[i].title;  body = topics[i].body;  break;  }  }  articleTag = <Article title={title} body={body}></Article>  }  return (  <>  <Header onChangeMode={onChangeHeaderHandler}></Header>  <Nav topics={topics} onChangeMode={onChangeNavHandler}></Nav>  {articleTag}  </>  );  } |
| --- |