**ch 05. 함수컴포넌트**

1. 실습 준비

* 온라인 개발환경 : <https://ko.reactjs.org>의 시작하기에서 Stackblitz :

<https://stackblitz.com/edit/react-mg6fq6?file=src%2FApp.js> 바로 사용 가능

* 추천 툴 체인 중 create-react-app 이용 : <https://create-react-app.dev>(사용방법)
* npx를 쓰기 위해서는 node.js를 설치 후 npx create-react-app .

1. 소스 수정 방법

src/index.js을 찾아 동작

* App.js

import './App.css';

function App() {

  return (

    <div className="App">

      Hello, React!!!

    </div>

  );

}

export default App;

1. 배포

npm run build

npx serve –s build

1. function을 이용한 컴포넌트 생성

[App.js]

import './App.css';

import Subject from './components/Subject';

import TOC from './components/TOC';

import ReadContent from './components/ReadContent';

import Control from './components/Control';

function App() {

  return (

    <>

      <Subject></Subject>

      <TOC></TOC>

      <Control></Control>

      <ReadContent></ReadContent>

    </>

  );

}

export default App;

[Subject.js]

import './Subject.css';

const Subject = ()=>{

  return (

    <header className="subject">

      <h1>WEB</h1>

    </header>

  );

};

export default Subject;

[Subject.css]

.subject h1 { color:red; cursor: pointer;}

[TOC.js]

const TOC = ()=>{

  return(

    <nav>

      <ol>

        <li><a href="1.html">HTML</a></li>

        <li><a href="2.html">CSS</a></li>

        <li><a href="3.html">JavaScript</a></li>

      </ol>

    </nav>

  );

};

export default TOC;

[ReacContent.js]

const ReadContent = ()=>{

  return(

    <article>

      <h2>Welcome</h2>

      hello, WEB

    </article>

  );

};

export default ReadContent;

[Control.js]

import './Control.css';

const Control = ()=>{

  return (

    <ul className="control">

      <li><button>CREATE</button></li>

      <li><button>UPDATE</button></li>

      <li><button>DELETE</button></li>

    </ul>

  );

};

export default Control;

[Control.css]

ul.control{

  padding: 0;

  overflow: hidden;

}

ul.control li{

  list-style: none;

  padding-left: 5px;

  float: left;

}

[ReadConten.js]

const ReadContent = ()=>{

  return(

    <article>

      <h2>Welcome</h2>

      hello, WEB

    </article>

  );

};

export default ReadContent;

1. props 추가

[App.js]

import './App.css';

import Subject from './components/Subject';

import TOC from './components/TOC';

import ReadContent from './components/ReadContent';

import Control from './components/Control';

function App() {

  let mode = 'welcome';

  const subject = 'WEB';

  const welcome = {title:'Welcome', desc:'Hello, React'};

  const contents = [

    {id:1, title:'HTML', desc:'HTML is HyperText Markup Langauge.'},

    {id:2, title:'CSS', desc:'CSS is for design'},

    {id:3, title:'JavaScript', desc:'JavaScript is for interacitive'},

  ];

  let \_article = null;

  if(mode==='welcome'){

    \_article = <ReadContent title={welcome.title} desc={welcome.desc}></ReadContent>

  }

  return (

    <>

      <Subject title={subject}

              onChangePage = {function(){

                alert('head click');

              }}

      ></Subject>

      <TOC data={contents}

          onChangePage = {function(\_id){

            alert(\_id);

          }}

      ></TOC>

      <Control onChangeMode= {function(\_mode){

        alert(\_mode);

        if(\_mode === 'delete'){

          // delete 로직

        }else{

          // 그외 로직

        }

      }}></Control>

      {\_article}

    </>

  );

}

export default App;

[Subject.js]

import './Subject.css';

const Subject = props=>{

  //console.log(props, props.title);

  return (

    <header className="subject"

            onClick={function(){

              props.onChangePage();

            }}>

      <h1>{props.title}</h1>

    </header>

  );

};

export default Subject;

[TOC.js]

const TOC = props => {

  console.log('TOC 실행');

  var lists = [];

  var data = props.data;

  for(var i=0 ; i<data.length ; i++){

    lists.push(<li key={data[i].id}>

                <a href={data[i].id+".html"}

                  id={data[i].id}

                  data-id={data[i].id}

                  onClick={event => {

                    event.preventDefault();

                    //props.onChangePage(event.target.dataset.id);

                    props.onChangePage(event.target.id);

                  }}>

                  {data[i].title}

                </a>

              </li>);

  }

  return(

    <nav>

      <ol>

        {lists}

      </ol>

    </nav>

  );

};

export default TOC;

[Control.js]

import './Control.css';

const Control = props =>{

  return (

    <ul className="control">

      <li>

        <button onClick={function(){

          props.onChangeMode('create');

        }}>

          CREATE

        </button>

      </li>

      <li>

        <button onClick={function(){

          props.onChangeMode('update');

        }}>

          UPDATE

        </button>

      </li>

      <li>

        <button onClick={function(){

          props.onChangeMode('delete');

        }}>

          DELETE

        </button>

      </li>

    </ul>

  );

};

export default Control;

[ReadContent.js]

const ReadContent = props =>{

  return(

    <article>

      <h2>{props.title}</h2>

      {props.desc}

    </article>

  );

};

export default ReadContent;

1. state개념

입력된 Props(외부에서 사용) => [ state(내부에서 사용) ]=> return (유사 HTML);

[App.js]

import './App.css';

import Subject from './components/Subject';

import TOC from './components/TOC';

import ReadContent from './components/ReadContent';

import Control from './components/Control';

import { useState } from 'react';

function App() {

  // let \_mode = useState('welcome'); // 지역변수를 상태로 업그레이드

  // let mode = \_mode[0];// 상태

  // let setMode = \_mode[1]; // 상태의 값을 변경할 때 사용하는 함수

  //console.log(\_mode);

  const [mode, setMode] = useState('welcome');

  const [selectedId, setSelectedId ] = useState(0);

  const subject = 'WEB';

  const welcome = {title:'Welcome', desc:'Hello, React'};

  const contents = [

    {id:1, title:'HTML', desc:'HTML is HyperText Markup Langauge.'},

    {id:2, title:'CSS', desc:'CSS is for design'},

    {id:3, title:'JavaScript', desc:'JavaScript is for interacitive'},

  ];

  let \_article, \_content = null;

  const getContent = function(){

    //console.log('길이 : ',contents.length, ',선택:' , typeof(selectedId));

    for(var i=0 ; i<contents.length ; i++){

      //console.log(contents[i].id, typeof(contents[i].id));

      if(contents[i].id === selectedId){

        return contents[i];

      }//if

    }//for

  }; // getContent()

  if(mode==='welcome'){

    \_article = <ReadContent title={welcome.title} desc={welcome.desc}></ReadContent>

  }else if(mode === 'read'){

    \_content = getContent();

    console.log(\_content);

    \_article = <ReadContent title={\_content.title} desc={\_content.desc}></ReadContent>

  }else if(mode === 'create'){

    \_article = <p>CREATE</p>

  }else if(mode === 'update'){

    \_article = <p>UPDATE</p>

  }

  return (

    <>

      <Subject title={subject}

              onChangePage = {function(){

                setMode('welcome');

                setSelectedId(0);

              }}

      ></Subject>

      <TOC data={contents}

          onChangePage = {function(\_id){

            setMode('read');

            setSelectedId(Number(\_id));

          }}

      ></TOC>

      <Control onChangeMode= {function(\_mode){

        //alert(\_mode);

        if(\_mode === 'delete'){

          // delete 로직

        }else{

          setMode(\_mode);

        }

      }}></Control>

      {\_article}

    </>

  );

}

export default App;

1. CREATE

[CreateContent.js]

const CreateContent = props => {

  return (

    <article>

      <h2>CREATE</h2>

      <form onSubmit={event=>{

              event.preventDefault();

              props.onCreate(event.target.title.value, event.target.desc.value);

            }}>

        <p><input type="text" name="title" placeholder="title" /></p>

        <p><textarea name="desc" placeholder="desc"></textarea></p>

        <p><input type="submit" value="추가"/></p>

      </form>

    </article>

  );

};

export default CreateContent;

[App.js]

import './App.css';

import Subject from './components/Subject';

import TOC from './components/TOC';

import ReadContent from './components/ReadContent';

import Control from './components/Control';

import { useState } from 'react';

import CreateContent from './components/CreateContent';

let maxId = 3;

function App() {

  // let \_mode = useState('welcome'); // 지역변수를 상태로 업그레이드

  // let mode = \_mode[0];// 상태

  // let setMode = \_mode[1]; // 상태의 값을 변경할 때 사용하는 함수

  //console.log(\_mode);

  const [mode, setMode] = useState('welcome');

  const [selectedId, setSelectedId ] = useState(0);

  const [contents, setContents] = useState([

    {id:1, title:'HTML', desc:'HTML is HyperText Markup Langauge.'},

    {id:2, title:'CSS', desc:'CSS is for design'},

    {id:3, title:'JavaScript', desc:'JavaScript is for interacitive'},

  ]);

  const subject = 'WEB';

  const welcome = {title:'Welcome', desc:'Hello, React'};

  let \_article, \_content = null;

  const getContent = function(){

    //console.log('길이 : ',contents.length, ',선택:' , typeof(selectedId));

    for(var i=0 ; i<contents.length ; i++){

      //console.log(contents[i].id, typeof(contents[i].id));

      if(contents[i].id === selectedId){

        return contents[i];

      }//if

    }//for

  }; // getContent()

  if(mode==='welcome'){

    \_article = <ReadContent title={welcome.title} desc={welcome.desc}></ReadContent>

  }else if(mode === 'read'){

    \_content = getContent();

    // console.log(\_content);

    \_article = <ReadContent title={\_content.title} desc={\_content.desc}></ReadContent>

  }else if(mode === 'create'){

    \_article = <CreateContent

                  onCreate = {(\_title, \_desc) => {

                    //alert(\_title + '/' + \_desc);

                    \_content = {id:++maxId, title:\_title, desc:\_desc};

                    //console.log('추가할 데이터 :', \_content);

                    let \_contents = Array.from(contents); // 복제

                    \_contents.push(\_content);

                    setContents(\_contents);

                    setMode('read'); // read 모드로

                    setSelectedId(maxId);

                  }}

              ></CreateContent>

  }else if(mode === 'update'){

    \_article = <p>UPDATE</p>

  }

  return (

    <>

      <Subject title={subject}

              onChangePage = {function(){

                setMode('welcome');

                setSelectedId(0);

              }}

      ></Subject>

      <TOC data={contents}

          onChangePage = {function(\_id){

            setMode('read');

            setSelectedId(Number(\_id));

          }}

      ></TOC>

      <Control onChangeMode= {function(\_mode){

        //alert(\_mode);

        if(\_mode === 'delete'){

          // delete 로직

        }else{

          setMode(\_mode);

        }

      }}></Control>

      {\_article}

    </>

  );

}

export default App;

1. UPDATE & Delete

[UpdateContent.js]

const UpdateContent = props => {

  var data = props.data;

  return (

    <article>

      <h2>UPDATE</h2>

      <form onSubmit={event=>{

              event.preventDefault();

              props.onUpdate(

                Number(event.target.id.value),

                event.target.title.value,

                event.target.desc.value

              );

            }}>

        <p><input type="hidden" name="id" defaultValue={data.id}/></p>

        <p><input type="text" name="title" placeholder="title" defaultValue={data.title}/></p>

        <p><textarea name="desc" defaultValue={data.desc}></textarea></p>

        <p><input type="submit" value="수정"/></p>

      </form>

    </article>

  );

};

export default UpdateContent;

[App.js]

import './App.css';

import Subject from './components/Subject';

import TOC from './components/TOC';

import ReadContent from './components/ReadContent';

import Control from './components/Control';

import { useState } from 'react';

import CreateContent from './components/CreateContent';

import UpdateContent from './components/UpdateContent';

let maxId = 3;

function App() {

  // let \_mode = useState('welcome'); // 지역변수를 상태로 업그레이드

  // let mode = \_mode[0];// 상태

  // let setMode = \_mode[1]; // 상태의 값을 변경할 때 사용하는 함수

  //console.log(\_mode);

  const [mode, setMode] = useState('welcome');

  const [selectedId, setSelectedId ] = useState(0);

  const [contents, setContents] = useState([

    {id:1, title:'HTML', desc:'HTML is HyperText Markup Langauge.'},

    {id:2, title:'CSS', desc:'CSS is for design'},

    {id:3, title:'JavaScript', desc:'JavaScript is for interacitive'},

  ]);

  const subject = 'WEB';

  const welcome = {title:'Welcome', desc:'Hello, React'};

  let \_article, \_content = null;

  const getContent = function(){

    //console.log('길이 : ',contents.length, ',선택:' , typeof(selectedId));

    for(var i=0 ; i<contents.length ; i++){

      //console.log(contents[i].id, typeof(contents[i].id));

      if(contents[i].id === selectedId){

        return contents[i];

      }//if

    }//for

  }; // getContent()

  if(mode==='welcome'){

    \_article = <ReadContent title={welcome.title} desc={welcome.desc}></ReadContent>

  }else if(mode === 'read'){

    \_content = getContent();

    // console.log(\_content);

    \_article = <ReadContent title={\_content.title} desc={\_content.desc}></ReadContent>

  }else if(mode === 'create'){

    \_article = <CreateContent

                  onCreate = {(\_title, \_desc) => {

                    //alert(\_title + '/' + \_desc);

                    \_content = {id:++maxId, title:\_title, desc:\_desc};

                    //console.log('추가할 데이터 :', \_content);

                    let \_contents = Array.from(contents); // 복제

                    \_contents.push(\_content);

                    setContents(\_contents);

                    setMode('read');

                    setSelectedId(maxId);

                  }}

              ></CreateContent>

  }else if(mode === 'update'){

    if(selectedId === 0){

      \_article = <h3>수정할 데이터를 선택 후 update 기능을 이용하세요</h3>;

    }else {

      \_content = getContent();

      \_article = <UpdateContent

                  data= {\_content}

                  onUpdate = {(\_id, \_title, \_desc)=>{

                    console.log('수정할 데이터 :',{id:\_id, title:\_title, desc:\_desc});

                    let \_contents = [ ...contents];

                    for(var i=0 ; i<\_contents.length ; i++){

                      if(\_contents[i].id === \_id){

                        //\_contents[i] = {id:\_id, title:\_title, desc:\_desc};

                        \_contents[i].title = \_title;

                        \_contents[i].desc = \_desc;

                        break;

                      }//if

                    }//for - \_contents 교체 완료

                    setContents(\_contents);

                    setMode('read');

                  }}

                ></UpdateContent>

    }

  }

  return (

    <>

      <Subject title={subject}

              onChangePage = {function(){

                setMode('welcome');

                setSelectedId(0);

              }}

      ></Subject>

      <TOC data={contents}

          onChangePage = {function(\_id){

            setMode('read');

            setSelectedId(Number(\_id));

          }}

      ></TOC>

      <Control onChangeMode= {function(\_mode){

        //alert(\_mode);

        if(\_mode === 'delete'){

          // delete 로직

          if(selectedId!==0 && window.confirm('삭제시 복구 불가합니다. 삭제하시겠습니까?')){

            let \_contents = Array.from(contents); // 복제

            for(let idx=0 ; idx<\_contents.length ; idx++){

              if(\_contents[idx].id === selectedId){

                // 지울 데이터

                \_contents.splice(idx, 1); // idx번째 1개 제거

                setContents(\_contents); // hooks 변경

                var \_selectedId;

                if(\_contents.length > 0) { // contents.length가 1이상이면 read모드

                  \_selectedId = \_contents[\_contents.length-1].id;

                  //\_selectedId = \_contents[0].id;

                  setMode('read'); //  hooks변경

                  setSelectedId(\_selectedId); // hooks변경

                }else{ // contents.length가 0이면 welcome모드로

                  \_selectedId = 0;

                  setMode('welcome');

                  setSelectedId(\_selectedId);

                }

                break;

              }//if

            }//for

          }else if(selectedId===0){

            alert('삭제할 데이터를 선택 후 delete 기능을 이용하세요');

          }//if - confirm

        }else{

          setMode(\_mode);

        }

      }}></Control>

      {\_article}

    </>

  );

}

export default App;

<!DOCTYPE html>

<html lang="ko">

<head>

<meta charset="UTF-8">

<title></title>

<style>

h2, button {display:inline-block; margin:10px;}

</style>

</head>

<body>

<div>다른 부분</div>

<hr>

<div id="react"></div>

<hr>

<div>또 다른 부분</div>

<!-- 개발용 -->

<script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>

<script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>

<!-- 배포용 -->

<!--<script src="https://unpkg.com/react@18/umd/react.production.min.js" crossorigin></script>-->

<!--<script src="https://unpkg.com/react-dom@18/umd/react-dom.production.min.js" crossorigin></script>-->

<!-- 실무에서는 배포용보다 웹팩이나 바벨로 사용한다 -->

<script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

<script type="text/babel">

ReactDOM.createRoot(document.querySelector('#react')).render(

<div>

</div>

);

</script>

</body>

</html>