**ch 05. function**

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';

function App() {

  return (

    <div>

      <Header></Header>

      <Nav></Nav>

      <Article></Article>

    </div>

  );

}

function Header(){

  return (

    <header>

      <h1><a href="/">WEB</a></h1>

      <hr/>

    </header>

  );

}

function Nav(){

  return (

    <nav>

      <ul>

        <li><a href="/read/1">HTML</a></li>

        <li><a href="/read/2">CSS</a></li>

        <li><a href="/read/3">js</a></li>

      </ul>

      <hr/>

    </nav>

  );

}

function Article(){

  return (

    <article>

      <h2>Welcome</h2>

      <p>Hello, WEB</p>

    </article>

  );

}

export default App;

1. props 이용

<img src="xxx.jpg" width="100" height="100">처럼 우리가 만든 컴포넌트도 속성을 갖도록

import './App.css';

function App() {

  const topics = [

    {id:1, title:"HTML", desc:"html is hyper Text Markup Language"},

    {id:2, title:"CSS", desc:"CSS is ..."},

    {id:3, title:"JavaScript", desc:"JavaScript is ..."},

  ];

  return (

    <div>

      <Header title="React"></Header>

      <Nav topics={topics}></Nav>

      <Article title="Welcome" body="Hello, Web"></Article>

    </div>

  );

}

function Header(props){

  //console.log('props :', props, props.title);

  return (

    <header>

      <h1><a href="/">{props.title}</a></h1>

      <hr/>

    </header>

    );

}

function Nav(props){

  //var lists = [

  //  <li><a href="/read/1">HTML</a></li>,

  //  <li><a href="/read/2">CSS</a></li>,

  //  <li><a href="/read/3">JavaScript</a></li>

  //] 이거 자체를 쓸 일은 없으므로 props에 따라 다르게 표현

  var lists = [];

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

    let t = props.topics[i];

    lists.push(

      <li key={t.id}> // 동적으로 만들어지는 태그는 key값이 unique하게

        <a href={"/read/"+t.id}>

          {t.title}

        </a>

      </li>);

  }

  return (

    <nav>

      <ol>

        {lists}

      </ol>

      <hr/>

    </nav>

  );

}

function Article(props){

  return (

    <article>

      <h2>{props.title}</h2>

      <p>{props.body}</p>

    </article>

  );

}

export default App;

1. 이벤트

<input type="button" onclick="alert('hi')"/>처럼 이벤트를 추가

import './App.css';

function App() {

  const topics = [

    {id:1, title:"HTML", desc:"html is hyper Text Markup Language"},

    {id:2, title:"CSS", desc:"CSS is ..."},

    {id:3, title:"JavaScript", desc:"JavaScript is ..."},

  ];

  return (

    <div>

      <Header title="React" onChangeMode={()=>{ // arrow function으로 간략히 바꿈

        alert('header');

      }}></Header>

      <Nav topics={topics} onChangeMode={(id)=>{

        alert(id);

      }}></Nav>

      <Article title="Welcome" body="Hello, Web"></Article>

    </div>

  );

}

function Header(props){

  //console.log('props :', props, props.title);

  return (

    <header>

      <h1><a href="/" onClick={(event)=>{

        event.preventDefault();

        props.onChangeMode('welcome');

      }}>{props.title}</a></h1>

      <hr/>

    </header>

    );

}

function Nav(props){

  //var lists = [

  //  <li><a href="/read/1">HTML</a></li>,

  //  <li><a href="/read/2">CSS</a></li>,

  //  <li><a href="/read/3">JavaScript</a></li>

  //]

  var lists = [];

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

    let t = props.topics[i];

    lists.push(

      <li key={t.id}>

        <a id={t.id}

          href={"/read/"+t.id} onClick={event=>{ // 파라미터가 하나면 괄호 생략가능

          event.preventDefault();

          props.onChangeMode(event.target.id); // event.target : 이벤트를 유발시킨 태그 : a태그

        }}>

          {t.title}

        </a>

      </li>);

  }

  return (

    <nav>

      <ol>

        {lists}

      </ol>

      <hr/>

    </nav>

  );

}

function Article(props){

  return (

    <article>

      <h2>{props.title}</h2>

      <p>{props.body}</p>

    </article>

  );

}

export default App;

1. State

Props(컴포넌트를 사용하는 외부자를 위한 속성)

State

**↓**

: 컴포넌트를 만드는 내부자를 위한 속성

return

import './App.css';

function App() {

  let mode = 'WELCOME';

  let topics = [

    {id:1, title:"HTML", desc:"html is hyper Text Markup Language"},

    {id:2, title:"CSS", desc:"CSS is ..."},

    {id:3, title:"JavaScript", desc:"JavaScript is ..."},

  ];

  let content = null;

  if(mode === 'WELCOME'){

    content = <Article title="Welcome" body="Hello, Web"></Article>

  }else if(mode === 'READ'){

    content = <Article title="READ" body="Hello, READ"></Article>

  }

  return (

    <div>

      <Header title="React" onChangeMode={()=>{ // arrow function으로 간략히 바꿈

        //alert('header');

        mode = 'WELCOME';

      }}></Header>

      <Nav topics={topics} onChangeMode={(id)=>{

        //alert(id);

        mode = 'READ';

      }}></Nav>

      {content}

    </div>

  );

}

function Header(props){

  //console.log('props :', props, props.title);

  return (

    <header>

      <h1><a href="/" onClick={(event)=>{

        event.preventDefault();

        props.onChangeMode('welcome');

      }}>{props.title}</a></h1>

      <hr/>

    </header>

    );

}

function Nav(props){

  var lists = [];

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

    let t = props.topics[i];

    lists.push(

      <li key={t.id}>

        <a id={t.id}

          href={"/read/"+t.id} onClick={event=>{ // 파라미터가 하나면 괄호 생략가능

          event.preventDefault();

          props.onChangeMode(event.target.id); // event.target : 이벤트를 유발시킨 태그 : a태그

        }}>

          {t.title}

        </a>

      </li>);

  }

  return (

    <nav>

      <ol>

        {lists}

      </ol>

      <hr/>

    </nav>

  );

}

function Article(props){

  return (

    <article>

      <h2>{props.title}</h2>

      <p>{props.body}</p>

    </article>

  );

}

export default App;

* 위와 같이 하여도 UI가 바뀌지 않아 state를 사용한다

import './App.css';

import {useState} from 'react'; //useState HOOK : react에서 제공하는 기본적인 함수

function App() {

  //let \_mode = useState('WELCOME'); // 상태를 만듦

  //console.log('\_mode : ', \_mode);

  //let mode = \_mode[0];

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

  // 위의 3줄을 아래의 한줄로 축약

  const [mode, setMode] = useState('WELCOME'); // setMode를 통해 mode를 변경

  const [id, setId] = useState(null);

  let topics = [

    {id:1, title:"HTML", desc:"html is hyper Text Markup Language"},

    {id:2, title:"CSS", desc:"CSS is ..."},

    {id:3, title:"JavaScript", desc:"JavaScript is ..."},

  ];

  let content = null;

  if(mode === 'WELCOME'){

    content = <Article title="Welcome" body="Hello, Web"></Article>

  }else if(mode === 'READ'){

    let title, desc = null;

    for(let i = 0 ; i<topics.length ; i++){

      if(topics[i].id === id){

        title = topics[i].title;

        desc = topics[i].desc;

        console.log(title, desc);

        break;

      }

    }

    content = <Article title={title} body={desc}></Article>

  }

  return (

    <div>

      <Header title="React" onChangeMode={()=>{ // arrow function으로 간략히 바꿈

        //alert('header');

        //mode = 'WELCOME';

        setMode('WELCOME');

        setId(null);

      }}></Header>

      <Nav topics={topics} onChangeMode={(\_id)=>{

        //alert(id);

        //mode = 'READ';

        setMode('READ'); // setMode에 의해 mode값이 바뀌면 App()함수가 다시 실행하며 화면이 rendering 됨

        setId(\_id);

      }}></Nav>

      {content}

    </div>

  );

}

function Header(props){

  //console.log('props :', props, props.title);

  return (

    <header>

      <h1><a href="/" onClick={(event)=>{

        event.preventDefault();

        props.onChangeMode('welcome');

      }}>{props.title}</a></h1>

      <hr/>

    </header>

    );

}

function Nav(props){

  var lists = [];

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

    let t = props.topics[i];

    lists.push(

      <li key={t.id}>

        <a id={t.id}

          href={"/read/"+t.id} onClick={event=>{ // 파라미터가 하나면 괄호 생략가능

          event.preventDefault();

          // event.target.id : 태그의 속성은 문자

          props.onChangeMode(Number(event.target.id)); // event.target : 이벤트를 유발시킨 태그 : a태그

          //props.onChangeMode(t.id);

        }}>

          {t.title}

        </a>

      </li>);

  }

  return (

    <nav>

      <ol>

        {lists}

      </ol>

      <hr/>

    </nav>

  );

}

function Article(props){

  return (

    <article>

      <h2>{props.title}</h2>

      <p>{props.body}</p>

    </article>

  );

}

export default App;

1. Create

import './App.css';

import {useState} from 'react'; //useState HOOK : react에서 제공하는 기본적인 함수

function App() {

  //let \_mode = useState('WELCOME'); // 상태를 만듦

  //console.log('\_mode : ', \_mode);

  //let mode = \_mode[0];

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

  // 위의 3줄을 아래의 한줄로 축약

  const [mode, setMode] = useState('WELCOME'); // setMode를 통해 mode를 변경

  const [id, setId] = useState(null);

  const [nextId, setNextId] = useState(4);

  const [topics, setTopics] = useState([

    {id:1, title:"HTML", desc:"html is hyper Text Markup Language"},

    {id:2, title:"CSS", desc:"CSS is ..."},

    {id:3, title:"JavaScript", desc:"JavaScript is ..."},

  ]);

  let content = null;

  if(mode === 'WELCOME'){

    content = <Article title="Welcome" body="Hello, Web"></Article>

  }else if(mode === 'READ'){

    let title, desc = null;

    for(let i = 0 ; i<topics.length ; i++){

      if(topics[i].id === id){

        title = topics[i].title;

        desc = topics[i].desc;

        console.log(title, desc);

        break;

      }

    }

    content = <Article title={title} body={desc}></Article>

  }else if(mode === 'CREATE'){

    content = <Create onCreate={(\_title, \_desc)=>{

      const newTopic = {id: nextId, title:\_title, desc:\_desc}

      const newTopics = [...topics]; // 복제본. 복제본을 만들지 않고 push하고 setTopics하면 주소가 바뀌지 않아 renderring 안됨

      newTopics.push(newTopic);

      setTopics(newTopics);

      //topics.push(newTopic); 이 두줄은 topics변수의 주소가 바뀌지 않아 rendering되지 않음

      //setTopics(topics);

      setMode('READ'); // 글을 추가하고 추가한 글의 상세보기 페이지로 감

      setId(nextId);

      setNextId(nextId+1);

    }}></Create>

  }

  return (

    <div>

      <Header title="React" onChangeMode={()=>{ // arrow function으로 간략히 바꿈

        //alert('header');

        //mode = 'WELCOME';

        setMode('WELCOME');

        setId(null);

      }}></Header>

      <Nav topics={topics} onChangeMode={(\_id)=>{

        //alert(id);

        //mode = 'READ';

        setMode('READ'); // setMode에 의해 mode값이 바뀌면 App()함수가 다시 실행하며 화면이 rendering 됨

        setId(\_id);

      }}></Nav>

      {content}

      <a href="/create" onClick={event => {

        event.preventDefault();

        setMode('CREATE');

      }}>Create하기</a>

    </div>

  );

}

function Create(props){

  return(

    <article>

      <h2>Create</h2>

      <form onSubmit={event=>{

        event.preventDefault();

        const title = event.target.title.value;

        const desc = event.target.desc.value;

        props.onCreate(title, desc);

      }}>

        <p>

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

        </p>

        <p>

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

        </p>

        <input type="submit" value="CREATE"/>

      </form>

      <hr/>

    </article>

  );

}

function Header(props){

  //console.log('props :', props, props.title);

  return (

    <header>

      <h1><a href="/" onClick={(event)=>{

        event.preventDefault();

        props.onChangeMode('welcome');

      }}>{props.title}</a></h1>

      <hr/>

    </header>

    );

}

function Nav(props){

  var lists = [];

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

    let t = props.topics[i];

    lists.push(

      <li key={t.id}>

        <a id={t.id}

          href={"/read/"+t.id} onClick={event=>{ // 파라미터가 하나면 괄호 생략가능

          event.preventDefault();

          // event.target.id : 태그의 속성은 문자

          props.onChangeMode(Number(event.target.id)); // event.target : 이벤트를 유발시킨 태그 : a태그

          //props.onChangeMode(t.id);

        }}>

          {t.title}

        </a>

      </li>);

  }

  return (

    <nav>

      <ol>

        {lists}

      </ol>

      <hr/>

    </nav>

  );

}

function Article(props){

  return (

    <article>

      <h2>{props.title}</h2>

      <p>{props.body}</p>

      <hr/>

    </article>

  );

}

export default App;

1. Update

주의점 : props의 값을 state로 갈아타고 바뀐 값을 state로

import './App.css';

import {useState} from 'react'; //useState HOOK : react에서 제공하는 기본적인 함수

function App() {

  //let \_mode = useState('WELCOME'); // 상태를 만듦

  //console.log('\_mode : ', \_mode);

  //let mode = \_mode[0];

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

  // 위의 3줄을 아래의 한줄로 축약

  const [mode, setMode] = useState('WELCOME'); // setMode를 통해 mode를 변경

  const [id, setId] = useState(null);

  const [nextId, setNextId] = useState(4);

  const [topics, setTopics] = useState([

    {id:1, title:"HTML", desc:"html is hyper Text Markup Language"},

    {id:2, title:"CSS", desc:"CSS is ..."},

    {id:3, title:"JavaScript", desc:"JavaScript is ..."},

  ]);

  let content = null;

  let contextControl = null;

  if(mode === 'WELCOME'){

    content = <Article title="Welcome" body="Hello, Web"></Article>

  }else if(mode === 'READ'){

    let title, desc = null;

    for(let i = 0 ; i<topics.length ; i++){

      if(topics[i].id === id){

        title = topics[i].title;

        desc = topics[i].desc;

        console.log(title, desc);

        break;

      }

    }

    content = <Article title={title} body={desc}></Article>

    contextControl = <li><a href={'/update/'+id} onClick={event=>{

      event.preventDefault();

      setMode('UPDATE');

    }}>Update하기</a></li>

  }else if(mode === 'CREATE'){

    content = <Create onCreate={(\_title, \_desc)=>{

      const newTopic = {id: nextId, title:\_title, desc:\_desc}

      const newTopics = [...topics]; // 복제본. 복제본을 만들지 않고 push하고 setTopics하면 주소가 바뀌지 않아 renderring 안됨

      newTopics.push(newTopic);

      setTopics(newTopics);

      //topics.push(newTopic); 이 두줄은 topics변수의 주소가 바뀌지 않아 rendering되지 않음

      //setTopics(topics);

      setMode('READ'); // 글을 추가하고 추가한 글의 상세보기 페이지로 감

      setId(nextId);

      setNextId(nextId+1);

    }}></Create>

  }else if(mode === 'UPDATE'){

    let title, desc = null;

    for(let i = 0 ; i<topics.length ; i++){

      if(topics[i].id === id){

        title = topics[i].title;

        desc = topics[i].desc;

        console.log(title, desc);

        break;

      }

    }

    content = <Update title={title} desc={desc} onUpdate={(\_title, \_desc)=>{

      const newTopics = [...topics]; // 복제

      const updateTopic = {id:id, title:\_title, desc:\_desc};

      for(var idx=0 ; idx < newTopics.length ; idx++){

        if(newTopics[idx].id === id){

          newTopics[idx] = updateTopic;

          break;

        }

      }

      setTopics(newTopics);

      setMode('READ');

    }}></Update>

  }

  return (

    <div>

      <Header title="React" onChangeMode={()=>{ // arrow function으로 간략히 바꿈

        //alert('header');

        //mode = 'WELCOME';

        setMode('WELCOME');

        setId(null);

      }}></Header>

      <Nav topics={topics} onChangeMode={(\_id)=>{

        //alert(id);

        //mode = 'READ';

        setMode('READ'); // setMode에 의해 mode값이 바뀌면 App()함수가 다시 실행하며 화면이 rendering 됨

        setId(\_id);

      }}></Nav>

      {content}

      <ul>

        <li>

          <a href="/create" onClick={event => {

            event.preventDefault();

            setMode('CREATE');

          }}>Create하기</a>

        </li>

        {contextControl}

      </ul>

    </div>

  );

}

function Update(props){

  const [title, setTitle] = useState(props.title);

  const [desc, setDesc] = useState(props.desc);

  return(

    <article>

      <h2>Update</h2>

      <form onSubmit={event=>{

        event.preventDefault();

        const title = event.target.title.value;

        const desc = event.target.desc.value;

        props.onUpdate(title, desc);

      }}>

        <p>

          <input type="text" name="title" placeholder="title" value={title}

          onChange={event=>{

            //console.log(event.target.value);

            setTitle(event.target.value);

          }}/>

        </p>

        <p>

          <textarea name="desc" placeholder="description" value={desc}

          onChange={event=>{

            setDesc(event.target.value);

          }}></textarea>

        </p>

        <input type="submit" value="UPDATE"/>

      </form>

      <hr/>

    </article>

  );

}

function Create(props){

  return(

    <article>

      <h2>Create</h2>

      <form onSubmit={event=>{

        event.preventDefault();

        const title = event.target.title.value;

        const desc = event.target.desc.value;

        props.onCreate(title, desc);

      }}>

        <p>

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

        </p>

        <p>

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

        </p>

        <input type="submit" value="CREATE"/>

      </form>

      <hr/>

    </article>

  );

}

function Header(props){

  //console.log('props :', props, props.title);

  return (

    <header>

      <h1><a href="/" onClick={(event)=>{

        event.preventDefault();

        props.onChangeMode('welcome');

      }}>{props.title}</a></h1>

      <hr/>

    </header>

    );

}

function Nav(props){

  var lists = [];

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

    let t = props.topics[i];

    lists.push(

      <li key={t.id}>

        <a id={t.id}

          href={"/read/"+t.id} onClick={event=>{ // 파라미터가 하나면 괄호 생략가능

          event.preventDefault();

          // event.target.id : 태그의 속성은 문자

          props.onChangeMode(Number(event.target.id)); // event.target : 이벤트를 유발시킨 태그 : a태그

          //props.onChangeMode(t.id);

        }}>

          {t.title}

        </a>

      </li>);

  }

  return (

    <nav>

      <ol>

        {lists}

      </ol>

      <hr/>

    </nav>

  );

}

function Article(props){

  return (

    <article>

      <h2>{props.title}</h2>

      <p>{props.body}</p>

      <hr/>

    </article>

  );

}

export default App;

1. Delete

import './App.css';

import {useState} from 'react'; //useState HOOK : react에서 제공하는 기본적인 함수

function App() {

  //let \_mode = useState('WELCOME'); // 상태를 만듦

  //console.log('\_mode : ', \_mode);

  //let mode = \_mode[0];

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

  // 위의 3줄을 아래의 한줄로 축약

  const [mode, setMode] = useState('WELCOME'); // setMode를 통해 mode를 변경

  const [id, setId] = useState(null);

  const [nextId, setNextId] = useState(4);

  const [topics, setTopics] = useState([

    {id:1, title:"HTML", desc:"html is hyper Text Markup Language"},

    {id:2, title:"CSS", desc:"CSS is ..."},

    {id:3, title:"JavaScript", desc:"JavaScript is ..."},

  ]);

  let content = null;

  let contextControl = null;

  if(mode === 'WELCOME'){

    content = <Article title="Welcome" body="Hello, Web"></Article>

  }else if(mode === 'READ'){

    let title, desc = null;

    for(let i = 0 ; i<topics.length ; i++){

      if(topics[i].id === id){

        title = topics[i].title;

        desc = topics[i].desc;

        console.log(title, desc);

        break;

      }

    }

    content = <Article title={title} body={desc}></Article>

    contextControl = <><li><a href={'/update/'+id} onClick={event=>{

      event.preventDefault();

      setMode('UPDATE');

    }}>Update하기</a></li>

    <li><button value="Delete하기" onClick={event=>{

      const newTopics = []

      for(let i=0 ; i<topics.length ; i++){

        if(topics[i].id !== id){

          newTopics.push(topics[i]);

        }

      }

      setTopics(newTopics);

      setMode('WELCOME');

      setId(null);

    }}>Delete하기</button></li></>

  }else if(mode === 'CREATE'){

    content = <Create onCreate={(\_title, \_desc)=>{

      const newTopic = {id: nextId, title:\_title, desc:\_desc}

      const newTopics = [...topics]; // 복제본. 복제본을 만들지 않고 push하고 setTopics하면 주소가 바뀌지 않아 renderring 안됨

      newTopics.push(newTopic);

      setTopics(newTopics);

      //topics.push(newTopic); 이 두줄은 topics변수의 주소가 바뀌지 않아 rendering되지 않음

      //setTopics(topics);

      setMode('READ'); // 글을 추가하고 추가한 글의 상세보기 페이지로 감

      setId(nextId);

      setNextId(nextId+1);

    }}></Create>

  }else if(mode === 'UPDATE'){

    let title, desc = null;

    for(let i = 0 ; i<topics.length ; i++){

      if(topics[i].id === id){

        title = topics[i].title;

        desc = topics[i].desc;

        console.log(title, desc);

        break;

      }

    }

    content = <Update title={title} desc={desc} onUpdate={(\_title, \_desc)=>{

      const newTopics = [...topics]; // 복제

      const updateTopic = {id:id, title:\_title, desc:\_desc};

      for(var idx=0 ; idx < newTopics.length ; idx++){

        if(newTopics[idx].id === id){

          newTopics[idx] = updateTopic;

          break;

        }

      }

      setTopics(newTopics);

      setMode('READ');

    }}></Update>

  }

  return (

    <div>

      <Header title="React" onChangeMode={()=>{ // arrow function으로 간략히 바꿈

        //alert('header');

        //mode = 'WELCOME';

        setMode('WELCOME');

        setId(null);

      }}></Header>

      <Nav topics={topics} onChangeMode={(\_id)=>{

        //alert(id);

        //mode = 'READ';

        setMode('READ'); // setMode에 의해 mode값이 바뀌면 App()함수가 다시 실행하며 화면이 rendering 됨

        setId(\_id);

      }}></Nav>

      {content}

      <ul>

        <li>

          <a href="/create" onClick={event => {

            event.preventDefault();

            setMode('CREATE');

          }}>Create하기</a>

        </li>

        {contextControl}

      </ul>

    </div>

  );

}

function Update(props){

  const [title, setTitle] = useState(props.title);

  const [desc, setDesc] = useState(props.desc);

  return(

    <article>

      <h2>Update</h2>

      <form onSubmit={event=>{

        event.preventDefault();

        const title = event.target.title.value;

        const desc = event.target.desc.value;

        props.onUpdate(title, desc);

      }}>

        <p>

          <input type="text" name="title" placeholder="title" value={title}

          onChange={event=>{

            //console.log(event.target.value);

            setTitle(event.target.value);

          }}/>

        </p>

        <p>

          <textarea name="desc" placeholder="description" value={desc}

          onChange={event=>{

            setDesc(event.target.value);

          }}></textarea>

        </p>

        <input type="submit" value="UPDATE"/>

      </form>

      <hr/>

    </article>

  );

}

function Create(props){

  return(

    <article>

      <h2>Create</h2>

      <form onSubmit={event=>{

        event.preventDefault();

        const title = event.target.title.value;

        const desc = event.target.desc.value;

        props.onCreate(title, desc);

      }}>

        <p>

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

        </p>

        <p>

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

        </p>

        <input type="submit" value="CREATE"/>

      </form>

      <hr/>

    </article>

  );

}

function Header(props){

  //console.log('props :', props, props.title);

  return (

    <header>

      <h1><a href="/" onClick={(event)=>{

        event.preventDefault();

        props.onChangeMode('welcome');

      }}>{props.title}</a></h1>

      <hr/>

    </header>

    );

}

function Nav(props){

  var lists = [];

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

    let t = props.topics[i];

    lists.push(

      <li key={t.id}>

        <a id={t.id}

          href={"/read/"+t.id} onClick={event=>{ // 파라미터가 하나면 괄호 생략가능

          event.preventDefault();

          // event.target.id : 태그의 속성은 문자

          props.onChangeMode(Number(event.target.id)); // event.target : 이벤트를 유발시킨 태그 : a태그

          //props.onChangeMode(t.id);

        }}>

          {t.title}

        </a>

      </li>);

  }

  return (

    <nav>

      <ol>

        {lists}

      </ol>

      <hr/>

    </nav>

  );

}

function Article(props){

  return (

    <article>

      <h2>{props.title}</h2>

      <p>{props.body}</p>

      <hr/>

    </article>

  );

}

export default App;