

[React 공부 4일차]

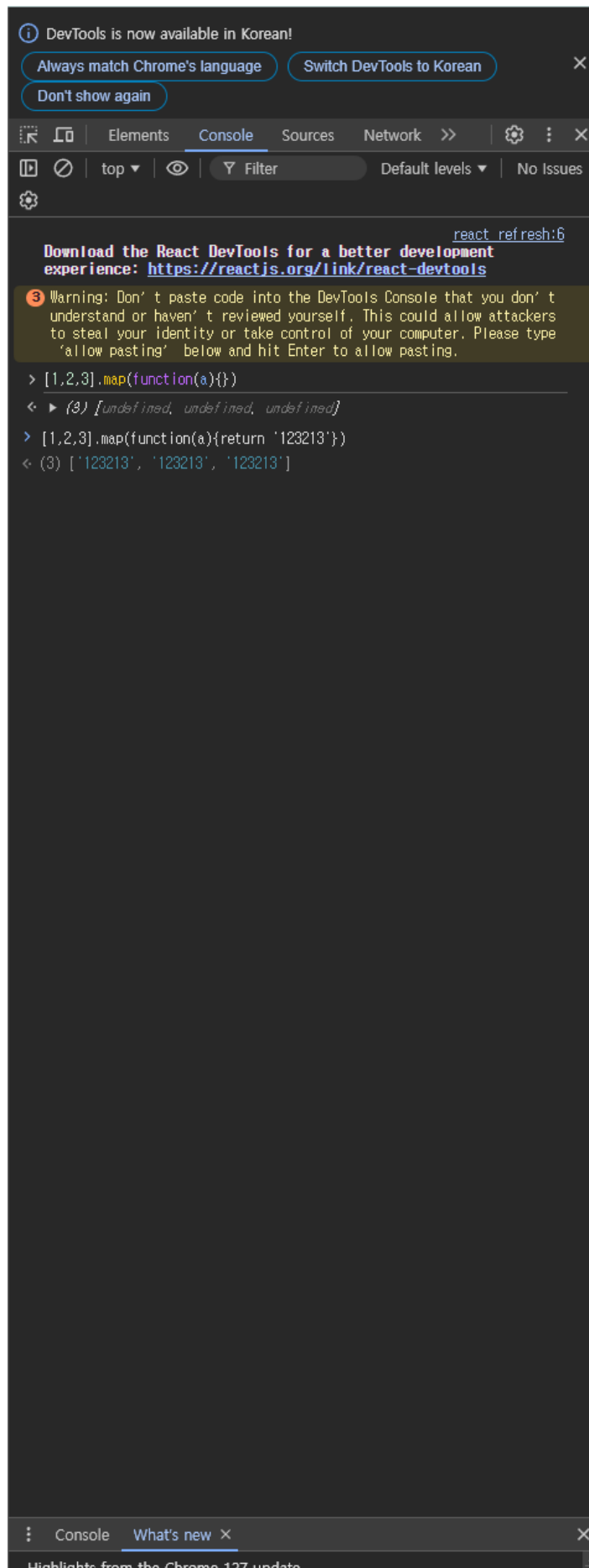
1. Map()을 이용한 반복문 만들기

```
[1,2,3].map(function(a){  
  return '1233211'  
})
```

이번 공부에서는 리액트에서 for 반복문을 만들어보는 공부를 해봤습니다.

react 내에선 for 라는 함수를 사용할 수 없지만 map 이란 함수를 사용 할 수있습니다.

사진과 같이 앞에 아무 배열이나 생성후에 .map(fuction(){})) 을 넣어줍니다. 가로안에 function은 콜백함수로서 기능을 작동합니다.



홈페이지에 콘솔창을 열어서 직접입력해보면 3개의 배열이 잘 들어가는것이 확인됩니다.

```
    { /* {titleName.map((item =>
      <div className='list' key={item.id}>
        <h4 onClick={()=>{
          modal == true ? setModal(false) : setModal(true)
        }}>
          {item.value}
          <span onClick={()=>upLike(like+1)}>❤️</span> {like}
        </h4>
        <p>8월 19일 발행</p>
      </div>
    )} */}

    {
      [1,2,3].map(function(){
        return <div>안녕</div>
      })
    }
  }
```

이제 기존에 쓰던 반복문은 주석처리해주고 새로 배우게된 배열.map 문법을 이용하여 div를 반복해서 만들어줘봅시다.

```
let [titleName, setTitle] = useState([
  {'id':1, 'value':'남성 코트 추천'},
  {'id':2, 'value':'나니노냐노'},
  {'id':3, 'value':'강남 우동 맛집'}]);

let [title, setName] = useState([
  '남성 코트 추천',
  '나니노냐노',
  '강남 우동 맛집']);

[1,2,3].map(function(a){
  return '1233211'
})
```

그러기 위해선 useState 에서 부여했던 id값과 value 값이 필요없어져 새로운 useState 를 생성해줍니다.

```
[1,2,3].map(function(){  
  return (<div className='list'  
    <h4>{ title[1] }</h4>  
    <p>8월 19일 발행</p>  
  </div>  
})
```

먼저 만들기에 앞서 테스트를 해보기위해 이런식으로 새로만든 useState를 넣어서 실행해 줍니다.

ReactBlog
나니노나노 8월 19일 발행
나니노나노 8월 19일 발행
나니노나노 8월 19일 발행

2번째 배열에있는 데이터값이 3번 정상적으로 반복되서 나오는 모습입니다.

```
title.map(function(a, i){  
  return (  
    <div className='list'  
      <h4>{ a }</h4>  
      <p>8월 19일 발행</p>  
    </div>  
  )  
})
```

본격적으로 새로만든 배열에 map을 사용해줍니다. 그리고 같은 데이터값이 나오면안되니 콜백함수에 a 라는 변수를 지정한뒤에 매번 새롭게 갱신시켜 출력시켜줍니다.

남성 코트 추천

8월 19일 발행

나니뇨냐뇨

8월 19일 발행

강남 우동 맛집

8월 19일 발행

이렇게 함으로써 title 이란 배열에있는 데이터값이 모두 정상적으로 출력하게됩니다.

이제 기존에 있던 좋아요 버튼도 추가해줍니다.

```
{
  title.map(function(a, i){
    return (
      <div className='list' key={i}>
        <h4 onClick={()=>{modal == true ? setModal(false) : setModal(true)}}>
          { a }
        </h4>
        <span onClick={()=>{
          let copyLike = [...like];
          copyLike[i] += 1;
          upLike(copyLike);
        }}>
          ♥
        </span> {like[i]}
        <p>8월 19일 발행</p>
      </div>
    )
  })
}
```

```
let [like, upLike] = useState([0,0,0,0]);
```

기존에 있던 좋아요버튼의 저장되는공간은 1개밖에없어서 모두 공통으로 증가했지만 배열로 새로운 공간을 뚫어줌으로써 각각 오르게할 수 있게 바꿔주었습니다.

전에 배웠던 copy 문법을 이용하여 map 함수 안에서 내가 누른 좋아요표시에 배열만 오르게한 뒤 upLike 에 copyLike 를 새로 갱신시켜줍니다.

❤️ 6

8월 19일 발행

나니뇨냐뇨

❤️ 4

8월 19일 발행

강남 우동 맛집

❤️ 5

8월 19일 발행

정상적으로 각각 좋아요 갯수가 오르는것이 확인되었습니다.

2. Props 사용해보기

```
}  
{  
  modal == true ? <Modal detail={title}/> : null  
}
```

이번엔 modal 창에서 다른 function 안에있는 변수들을 받아오는법을 써보려합니다.

이런식으로 사용되고있는 모달안에 이름={받아올변수} 를 적어주면 modal function안에 내가 보내주고싶은 변수를 보내줄 수 가있습니다.

```
function Modal(props){
  return (
    <>
      <div className='modal'>
        <h4>{props.title[0]}</h4>
        <p>날짜</p>
        <p>상세내용</p>
      </div>
    </>
  )
}
```

그런뒤 Modal() 안에 props를 적어서 사용할 수 있게 해줍니다.

그 후 props 안에 저장되어있고 위에서 보내준 데이터인 title 을 불러옵니다.

남성 코트 추천

날짜

상세내용

이렇게하면 정상적으로 title 값을 받아오는것을 확인할 수 있습니다.

```
}
{
  modal == true ? <Modal color={'skyblue'} title={title}/> : null
}
```

```
function Modal(props){
  return (
    <>
      <div className='modal' style={{background : props.color}}>
        <h4>{props.title[0]}</h4>
        <p>날짜</p>
        <p>상세내용</p>
      </div>
    </>
  )
}
```

번외로 배경색을 지정할때에도 이런식으로 색상 데이터를 보내줄 수 있습니다.

```
<button onClick={()=>{
  let copyTitle = [...title];
  copyTitle[0] = '여성 코트 추천';
  setName(copyTitle);
}}> 변경하기 </button>
```

```
function Modal(props){
  return (
    <>
      <div className='modal'>
        <h4>{props.title[0]}</h4>
        <p>날짜</p>
        <p>상세내용</p>
        <button>글수정</button>
      </div>
    </>
  )
}
```

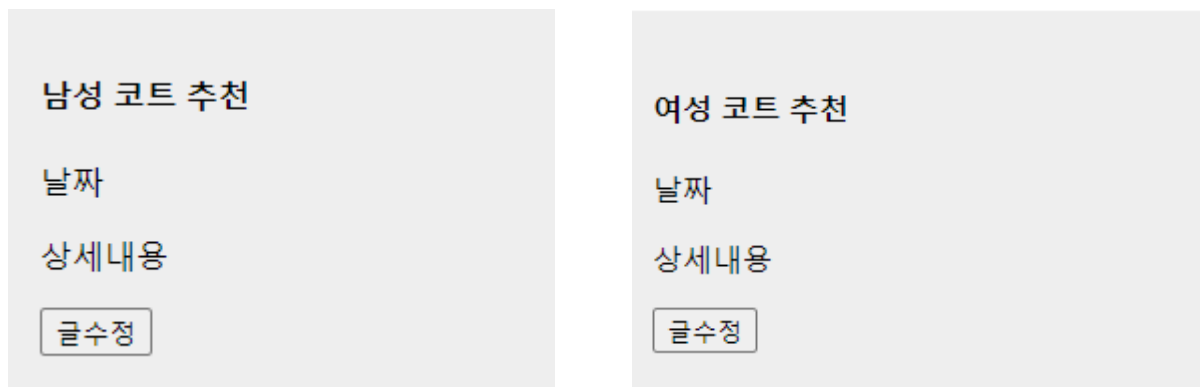
해서 이제 원래 div에 있던 글 수정버튼을 modal 창으로 옮겨봅시다.

```
{
  modal == true ? <Modal title={title} setName={setName}/> : null
}
```

그 후 사용하던 변수와 useState 변경함수를 modal로 전송해줍니다.


```
function Modal(props){
  return (
    <>
      <div className='modal'>
        <h4>{props.title[0]}</h4>
        <p>날짜</p>
        <p>상세내용</p>
        <button onClick={()=>{
          let copyTitle = [...props.title];
          copyTitle[0] == '남성 코트 추천' ? copyTitle[0] = '여성 코트 추천' : copyTitle[0] = '남성 코
          props.setName(copyTitle);
        }}>글수정</button>
      </div>
    </>
  )
}
```

이제 데이터를 받아와 이런식으로 사용이 가능해지는데. 저번에 배웠던 삼항연산자를 사용하여 if문을 주어 적힌 글이 남성 코트 추천이면 여성 코트 추천으로 copy문법을 사용해 적용시키고 그렇지 않으면 남성 코트 추천으로 적용시켜줍니다.



정상적으로 제대로 글이 수정되는것을 확인할 수 있습니다.

이제 글 별로 누르면 제목이 달라지게끔 만들어볼텐데요.

```
let [like, uplike] = useState([0,0,0]),
let [titleIndex, setTitleIndex] = useState(0);
let [page, setPage] = useState('ReactBlog');
```

글의 index값을 받아오기위해 titleIndex 변수를 만들어줍니다.

```

<div className={list} key={i}>
  <h4 onClick={()=>{
    modal == true ? setModal(false) : setModal(true)
    setTitleIndex(i);
  }}>
    { a }
  </h4>

```

이전에 만들어줬던 반복문에서 제목을 누르면 modal 창이 true가 되는동시에 i 값을 titleIndex 에 저장 해줍니다.

```

{
  modal == true ? <Modal title={title} setName={setName} titleIndex={titleIndex}/> : null
}

```

그 후 저장되어있는 값을 modal 창으로 보내줍니다.

```

function Modal(props){
  return (
    <>
      <div className='modal'>
        <h4>{props.title[props.titleIndex]}</h4>
        <p>날짜</p>
        <p>상세내용</p>
        <button onClick={()=>{
          let copyTitle = [...props.title];
          copyTitle[props.titleIndex] = copyTitle
          props.setName(copyTitle);
        }}>글수정</button>
      </div>
    </>
  )
}

```

숫자 0,1,2..... 를받아오니 title[0] 자리에 받아오는 데이터를 props.titleIndex 값으로 대체해줍니다.

그리고 실행시켜보면

남성 코트 추천

날짜

상세내용

❤ 0

글수정

나니노냐노

날짜

상세내용

❤ 0

글수정

강남 우동 맛집

날짜

상세내용

❤ 0

글수정

3개가 정상적으로 잘 나오는 모습을 확인할 수 있습니다.

3. Input 태그 이용해보기

```
<div></div>
<input></input>
<button>입력</button>
```

```
<div></div>
<input onChange={() => { console.log(1) }} />
<button>입력</button>
```

input 를 이용하여 게시글을 추가하는 법을 만드어보려합니다.

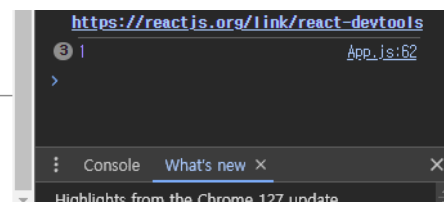
사용하는 이벤트 핸들러는 onChange 를 사용합니다. 이러면 input창에 글을 입력할때마다 발동이됩니다.

확인하기위해 console.log 를 찍어봅니다.

가나다순 정렬 변경하기

모노이

입력



글을 입력하자 console 창에 1이 뜨는것을 확인할 수 있습니다.

```
let [modal, setModal] = useState(false);  
let [inputValue, setInputValue] = useState('');
```

이제 입력한값을 받아오기위해 새로운 useState를 만들어줍니다. 들어가기위해 공백을 뚫어놓습니다.

```
<input onChange={e=>{  
  setInputValue(e.target.value);  
  console.log(inputValue);  
}}/>
```

onChange 안에 e 라는 변수명을 만들고 e.target.value 를 적어줌으로써 input창에 입력한값이 inputValue 로 들어갈 수 있게끔 해줍니다.



이렇게 잘 작동하는것을 확인할 수 있습니다.

```
const removeTitle = (index) => {  
  const updatedTtitle = title.filter((_,i) => i !== index);  
  setName(updatedTtitle);  
}
```

글 삭제 기능을 만들어주기위해 filter 기능을 만들어줍니다.

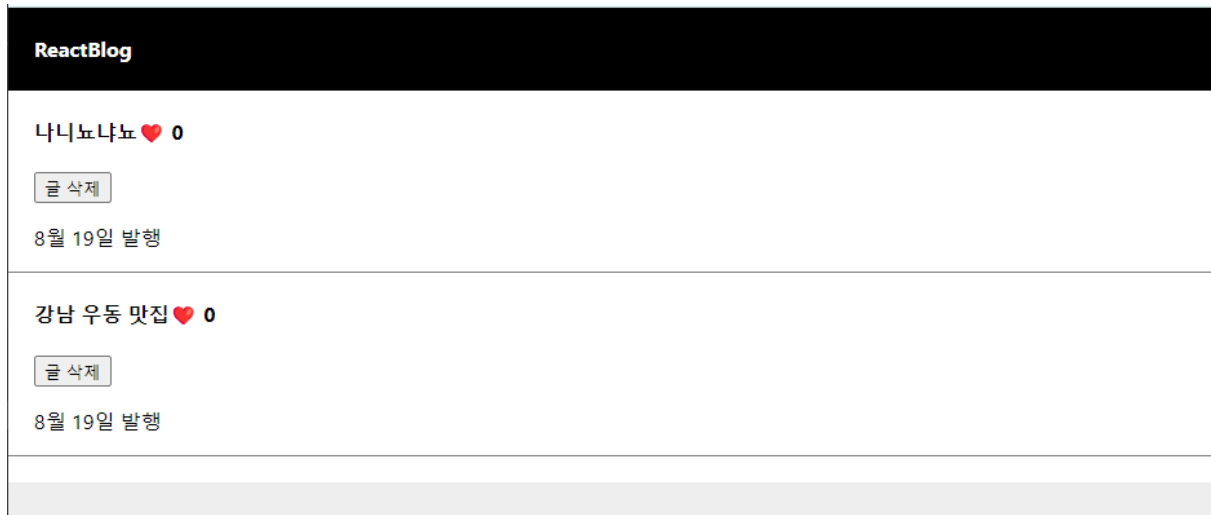
원하는값을 찾아서 제외해준뒤 useState에 새로 업데이트해줍니다.

```

<button onClick={()=>{
  removeTitle(i);
}}>글 삭제
</button>

```

우리는 미리 map을 이용한 for 문을 만들어줬기에 글삭제 버튼을 누르는순간 작동하게끔 만들어줍니다.



제대로 삭제가 된 모습입니다.

```

<input onChange={(e)=>{
  setInputValue(e.target.value);
  console.log(inputValue);
}}></input>
<button onClick={()=>{
  title.push(inputValue);
  console.log(title);
}}>입력</button>

```

```

<input onChange={(e)=>{
  setInputValue(e.target.value);
  console.log(inputValue);
}}></input>
<button onClick={()=>{
  let copyWrite = [...title];
  let copyNewLike = [...like];
  copyWrite.unshift(inputValue);
  copyNewLike.unshift(0);
  setName(copyWrite);
  upLike(copyNewLike);
}}>입력</button>

```

새로운 글을 추가하는법은 2가지의 방법을 사용해보았는데.

push 와 unshift 였습니다. 새로운 글을 맨앞에 추가해주고 새로 업데이트하여 글이 추가된 것처럼 보이게 해줍니다.

ㅋㅋㅋ ❤️ 0

글 삭제

8월 19일 발행

ㄹㅇㅇㄹㅇㅇ ❤️ 0

글 삭제

8월 19일 발행

나니뇨냐뇨 ❤️ 0

글 삭제

8월 19일 발행

강남 우동 맛집 ❤️ 0

글 삭제

8월 19일 발행

가나다순 정렬 변경하기

ㅋㅋㅋ

입력

새로 글이 추가된 모습입니다.

4. Class 를 이용한 Modal창 만들기

```
class Modal2 extends React.Component {  
    
}
```

```
class Modal2 extends React.Component {
  constructor(){
    super()
  }
  render(){
    return(
    )
  }
}
```

class 를 이용한 modal창을 만들기위해 function app 에서 빠져나와 기본 문법을 작성해 줍니다.

위에 사진과 같은 형식이 기본 형태라 보시면됩니다.

```
class Modal2 extends React.Component {
  constructor(){
    super();
    this.state = {
      name : 'kim' ,
      age : 20
    }
  }
  render(){
    return(
      <div>{this.state.name}</div>
    )
  }
}
```

constructor 문법안에 원하는 데이터값을 만들어서 넣어노르면 render return 쪽에서 이용할 수 있습니다.

20 버튼

이런식으로 제대로 정상작동되는것이 확인이 됩니다.

```
constructor(props){
  super(props);
  this.state = {
    name : 'kim' ,
    age : 20
  }
}
```

function 을 이용한 modal창과 동일하게 props 를 넣어주면 데이터값을 받아와서 활용할 수 있습니다.

감사합니다.

연락처

✉ alarm153@Naver.com 📞 +82 10-4750-0732 🏠 서울시 관악구 신림동