

9월 15일 (수)

1. 코딩애플 리액트 강의 Part 3.

- DB없이 데이터 저장하고싶으면 localStorage

The screenshot displays a React application interface and its associated code editor. On the left, a browser window titled 'React App' shows a product page for three types of shoes: 'White and Black', 'Red Knit', and 'Grey Yordan'. Each shoe has a small image, a name, and a price. Below the names, there are two '10' buttons. The code editor on the right shows a file named 'Detail.js' with the following content:

```
useState(true);
변경] = useState('');

useState(0);
useState(false);

context);

() => { alert변경(false), 2000};

out(타이머) }
```

Annotations on the right side of the code editor provide explanatory text:

- state 데이터를 기억하게하려면
- 서버로보내서 DB에 저장
- 아니면 브라우저 저장공간에 저장

White and Black
Born in France & 120000

Red Knit
Born in Seoul & 110000

Grey Yordan
Born in the States & 130000

useState(true);
변경] = useState('');
useState(0);
useState(false);
context);
(() => { alert변경(false) }, 2000);
out(타이머)

Key	Value
여기	여기가 그 비밀공간

Select a value to preview

어플리케이션 칸에 있다.

The screenshot shows the Chrome DevTools Application tab. On the left, the storage hierarchy is visible: Application, Storage, Cache, and Network. Under Storage, Local Storage is expanded, showing a single entry for the URL `http://localhost:3000`. The table view shows one key-value pair:

Key	Value
자료이름	자료값

A message at the bottom center says "Select a value to preview".

키와 값 쌍으로 이루어진다.

The screenshot shows the Chrome DevTools Application tab. The storage hierarchy is identical to the previous screenshot. However, under Storage, Session Storage is expanded, showing a single entry for the URL `http://localhost:3000`. The table view is empty, indicating no data is currently stored.

session storage 도 데이터를 저장하는 공간이지만, 이는 브라우저를 끄면 날라간다.

tekst 5MB 정도

Select a value to preview

텍스트만 5MB 정도 저장이 가능하다.

최근본상품 [상품1, 상품3, 상품6]

Select a value to preview

쇼핑몰의 최근 본 상품의 경우 로컬스토리지를 이용하여 구현한다.

```
> localStorage.setItem('name', 'Kim')
< undefined
```

첫번째 인자는 key, 두 번째 인자는 value이다.

Key	Value
name	Kim

Line 1, Column 1
1 Kim

The screenshot shows a React application displaying three shoe models: White and Black, Red Knit, and Grey Yordan. The Red Knit shoe is highlighted. The browser's developer tools (Chrome DevTools) are open, showing the following interactions with localStorage:

```
> localStorage.setItem('name', 'Kim')
< undefined
> localStorage.getItem('name')
< "Kim"
> |
```

A separate dark panel on the right contains Korean text explaining localStorage methods:

- localStorage 문법 3개
- 자료저장은 localStorage.setItem
- 자료출력은 localStorage.getItem

At the bottom of the dark panel, there is some code: `2000);`

The screenshot shows the same React application, but the Red Knit shoe is no longer visible. The browser's developer tools (Chrome DevTools) show the following interactions with localStorage:

```
> localStorage.setItem('name', 'Kim')
< undefined
> localStorage.getItem('name')
< "Kim"
> localStorage.removeItem('name')
< undefined
> |
```

A separate dark panel on the right contains Korean text explaining localStorage methods:

- localStorage 문법 3개
- 자료저장은 localStorage.setItem
- 자료출력은 localStorage.getItem
- 자료삭제는 localStorage.removeItem

At the bottom of the dark panel, there is some code: `2000);`

```

localStorage.setItem('name', 'Kim');
< undefined
> localStorage.getItem('name')
< "Kim"
> localStorage.removeItem('name')
< undefined
>

```

sessionStorage.어쩌구

localStorage 문법 3개

자료저장은
localStorage.setItem

자료출력은
localStorage.getItem

자료삭제는
localStorage.removeItem

sessionStorage 도 같은 문법을 사용하면 된다.

```

localStorage.setItem('obj', {name:'kim'})
< undefined
>

```

localStorage에 object자료를 저장하려면

localStorage에
object자료를 저장하려면

White and Black
Born in France & 120000

Red Knit
Born in Seoul & 110000

Grey Yordan
Born in the States & 130000

Select a value to preview

localStorage에
object자료를 저장하려면

- 그냥 넣으면 깨짐

White and Black
Born in France & 120000

Red Knit
Born in Seoul & 110000

Grey Yordan
Born in the States & 130000

object 자료를 강제로 글로 바꾸면
이렇게 나옴

localStorage에
object자료를 저장하려면

White and Black
Born in France & 120000

Red Knit
Born in Seoul & 110000

Grey Yordan
Born in the States & 130000

```
> localStorage.setItem('arr', [1,2,3])
< undefined
> localStorage
localStorage
location
locationbar
Location
Lock
LockManager
Geolocation
GeolocationCoordinates
GeolocationPosition
GeolocationPositionError
```

localStorage에
object자료를 저장하려면

- 그냥 넣으면 깨짐

2000);

배열도 넣어보면

Key	Value
arr	1,2,3
obj	[object Object]

Manifest
Service Workers
Clear storage

Storage

- Local Storage
 - http://localhost:3000
- Session Storage
- IndexedDB
- Web SQL
- Cookies

1 [object Object]

Line 1, Column 1

localStorage에
object자료를 저장하려면

- 그냥 넣으면 깨짐

2000);

대괄호가 없어진다.

localStorage에
object자료를 저장하려면
- 그냥 넣으면 깨짐

White and Black
Born in France & 120000

Red Knit
Born in Seoul & 110000

Grey Yordan
Born in the States & 130000

```
> localStorage.setItem('arr', [1,2,3])
< undefined
> localStorage.getItem('arr')
< "1,2,3"
>
```

localStorage에
object자료를 저장하려면
- 그냥 넣으면 깨짐

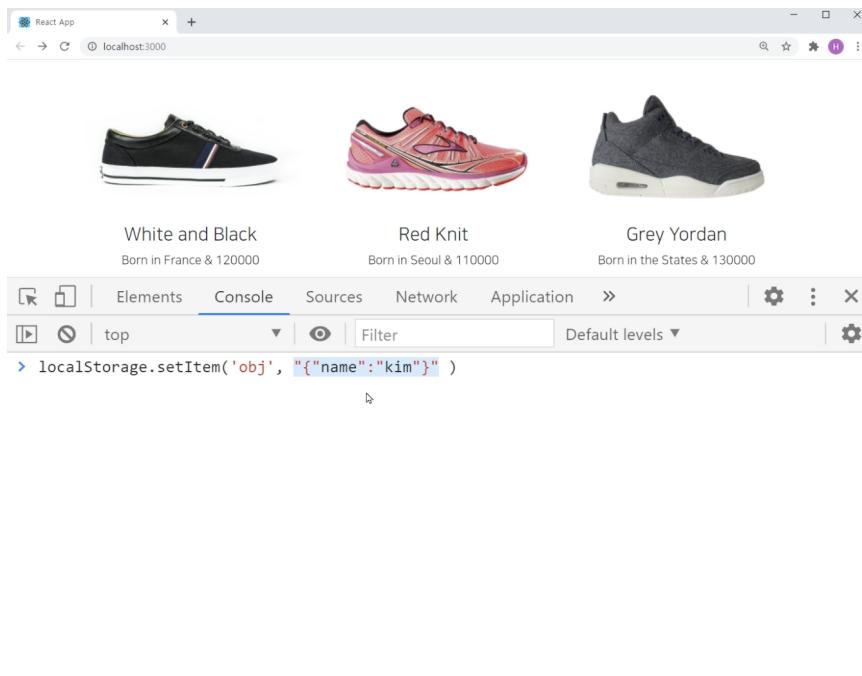
White and Black
Born in France & 120000

Red Knit
Born in Seoul & 110000

Grey Yordan
Born in the States & 130000

Key	Value
arr	1,2,3
obj	[object Object]

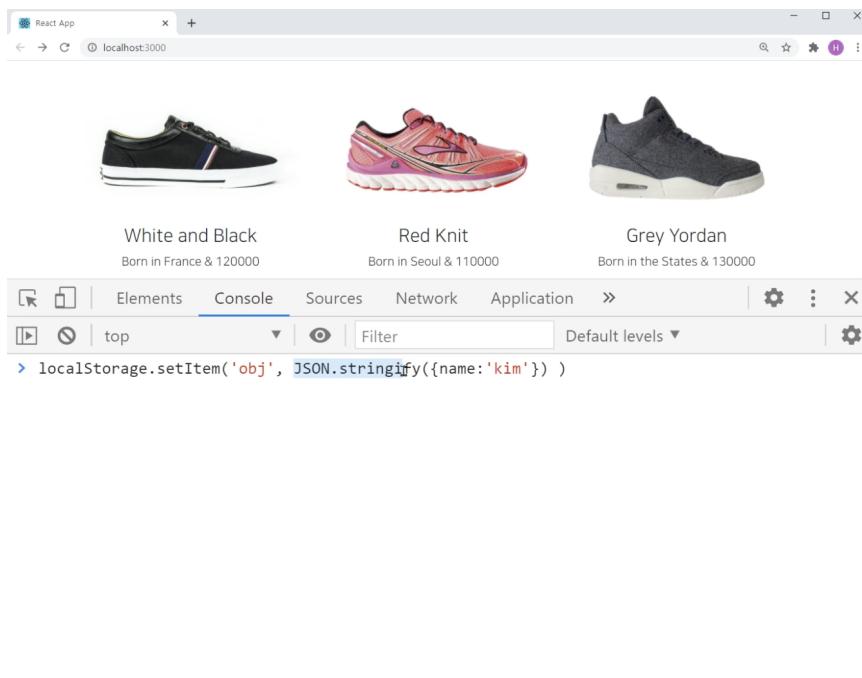
삭제는 우클릭으로 가능하다.



localStorage에
object자료를 저장하려면
- 그냥 넣으면 깨짐
- 따옴표쳐서 JSON 만드셈

```
localStorage.setItem('obj', '{"name":"kim"}')
```

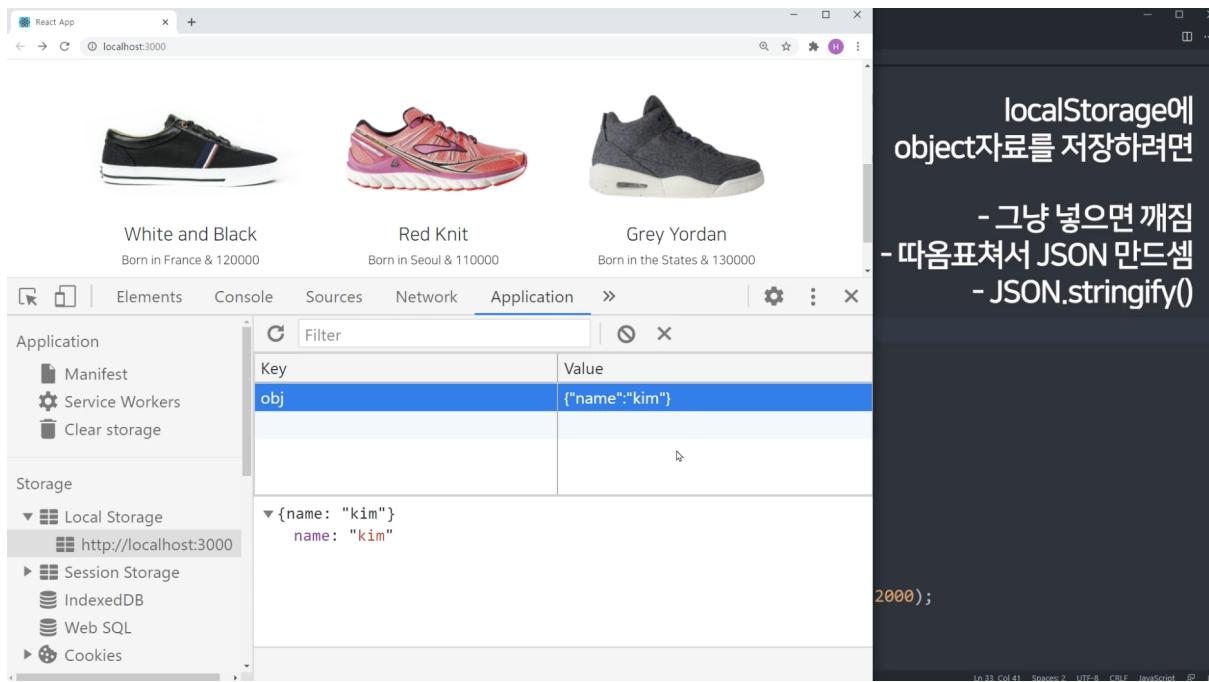
객체를 문자열처럼 만들어주면 된다.



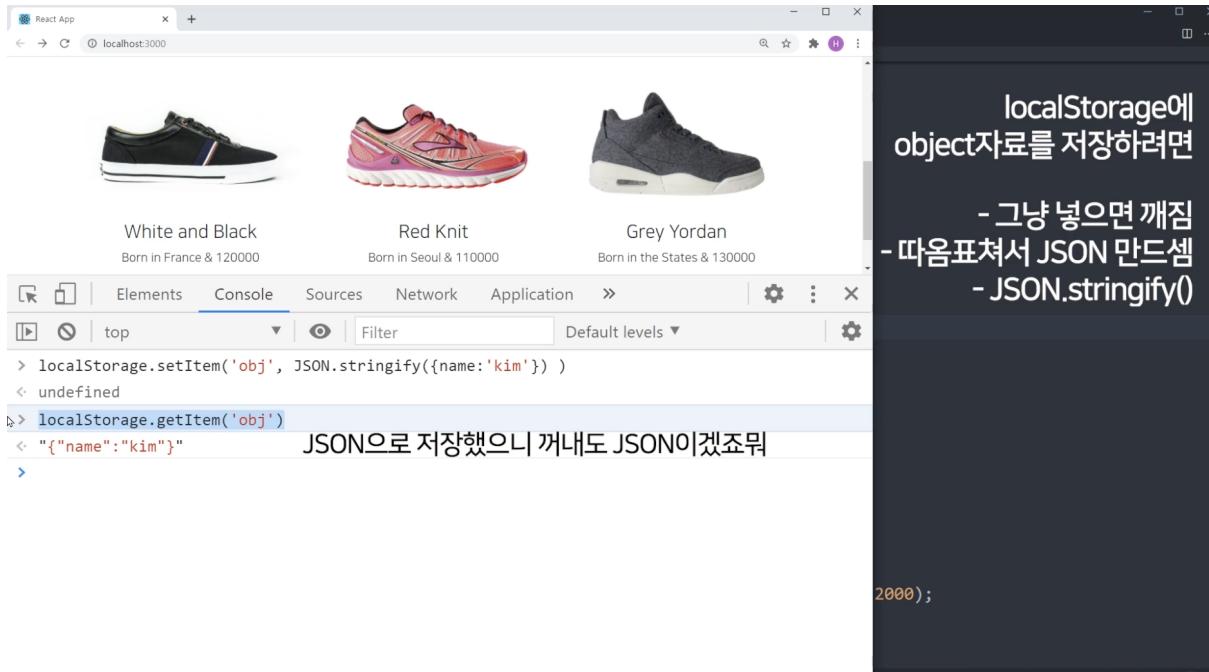
localStorage에
object자료를 저장하려면
- 그냥 넣으면 깨짐
- 따옴표쳐서 JSON 만드셈
- JSON.stringify()

```
localStorage.setItem('obj', JSON.stringify({name:'kim'}))
```

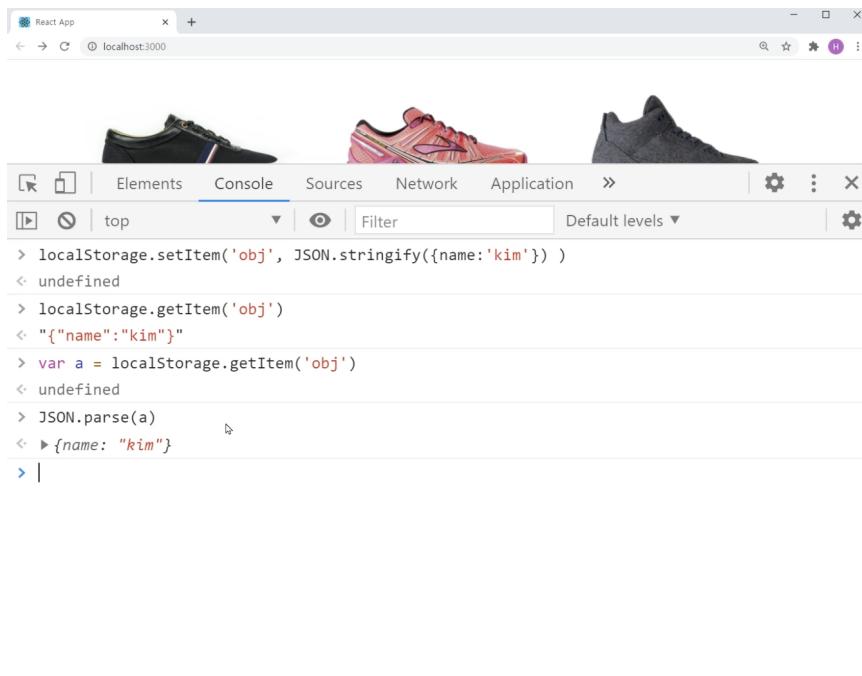
손 쉽게 JSON.stringify 함수를 사용하면 된다.



객체가 깨지지 않고 유지된다.



하지만 값이 json 형식이라 바로 가져다 못 쓴다.



```

React App
localhost:3000

Elements Console Sources Network Application
top Filter Default levels

localStorage.setItem('obj', JSON.stringify({name:'kim'}))
undefined
localStorage.getItem('obj')
{"name":"kim"}
var a = localStorage.getItem('obj')
undefined
JSON.parse(a)
{name: "kim"} ↴
|

```

localStorage에 object자료를 저장하려면

- 그냥 넣으면 깨짐
- 따옴표쳐서 JSON 만드셈
- JSON.stringify()

- 따옴표 제거하려면

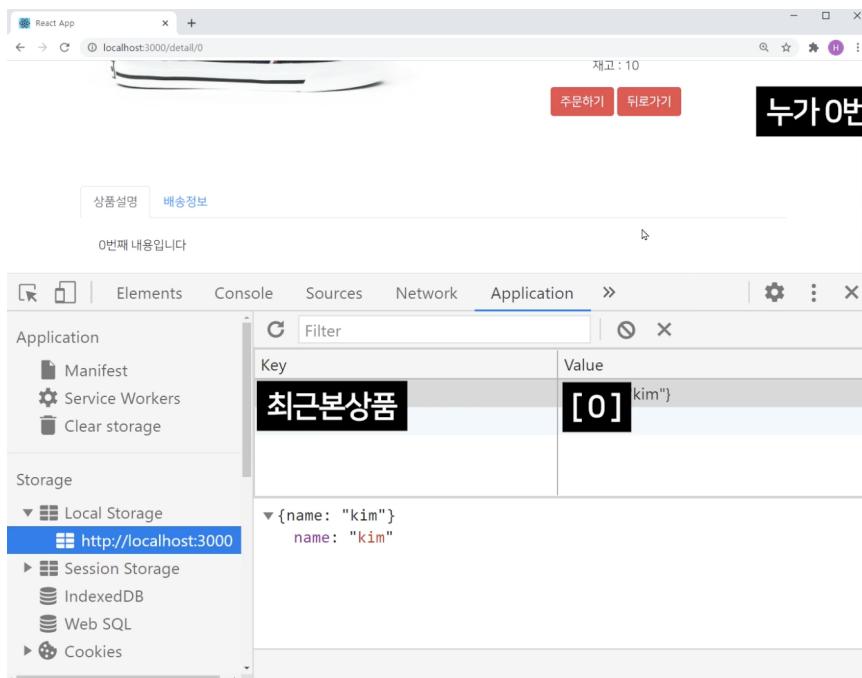
- JSON.parse()

```

2000);

```

JSON.parse() 함수를 사용하여 "" 를 제거해준다.



상품설명 배송정보

0번째 내용입니다

주문하기 뒤로가기

최근본상품 [0] kim

{name: "kim"}
name: "kim"

누가 0번째 상품 Detail페이지 들가면

```

2000);

```

A screenshot of a browser window titled "React App" showing a product detail page with a "상품설명" tab selected. Below it, the browser's developer tools show the "Console" tab with the output of `localStorage.getItem('ddddd')` being `null`. To the right is a Visual Studio Code editor with the file "Detail.js" open. The code contains a function "Detail" that uses useState hooks for "alert" and "inputData". A list of 6 tips is overlaid on the code:

- 누가 Detail페이지 들어가면
- localStorage에 있는 항목을 깨냄
- 경우가 두가지가 있겠네 null이 나오거나 [] 가 나오거나
- []가 나오면 거기에 URL파라미터의 id부분을 push()함 (추가함)
- 중복 처리하기
- 그러면 | **팁 : 중복을 허용하지 않는 Set 자료형**

```
function Detail(props) {  
  let [alert, alert변경] = useState(true);  
  let [inputData, inputData변경] = useState('');  
};
```

A screenshot of a browser window titled "React App" showing a product detail page with a "상품설명" tab selected. Below it, the browser's developer tools show the "Console" tab with the output of `localStorage.getItem('ddddd')` being `null`. To the right is a Visual Studio Code editor with the file "Detail.js" open. The code contains a function "Detail" that uses useState hooks for "alert" and "inputData". A list of 6 tips is overlaid on the code:

- 누가 Detail페이지 들어가면
- localStorage에 있는 항목을 깨냄
- 경우가 두가지가 있겠네 null이 나오거나 [] 가 나오거나
- []가 나오면 거기에 URL파라미터의 id부분을 push()함 (추가함)
- 중복 처리하기
- 그러면 [] 를 다시 localStorage에 저장함 (따옴표쳐서)

```
function Detail(props) {  
  let [alert, alert변경] = useState(true);  
  let [inputData, inputData변경] = useState('');  
};
```

2. 리액트 관련 내용

- 리액트 이미지 경로

보통 이미지는 public 폴더에 img 폴더를 만들어서 모아 쓴다.

```
import React from 'react';
import './Card1.css';

function Card1() {
    return (
        <div className="Card1">
            <div className="c1image">
                
            </div>
        </div>
    )
}

export default Card1;
```