

Local Storage & 비동기 Todo List

# 14번째 세션

NEXT X LIKELION 김지성

# | 목차

**1. Local Storage 개념**

**2. Todo List Local Storage 실습**

**3. Todo List 비동기 실습**

# | 쿠키와 웹 스토리지

지금까지 데이터 저장은? -> 전부 서버에 저장

모든 데이터를 서버에 저장했을 때 문제점

-> 불필요하게 서버와 많이 통신하며 트래픽 부담 상승

-> 사소한 부분에서도 로딩이 생기며 UX 저하

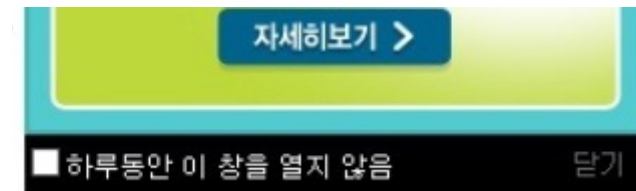
# | 쿠키와 웹 스토리지

웹에서도 로컬에 데이터를 저장할 수 있다!

심지어 우린 이것 많이 써봤다.



The image shows the Naver login interface. At the top is the green 'NAVER' logo. Below it are two input fields: the first for a username (containing a vertical line) and the second for a password (labeled '비밀번호'). A large green button with the text '로그인' (Login) is positioned below the input fields. At the bottom, there are three small links: '로그인 상태 유지' (Keep login state) with a checked checkbox, 'IP보안 ON' (IP Security ON), and '일회용 로그인 ?' (One-time login ?).



The image shows a Naver cookie consent bar. It has a blue header with a button labeled '자세히보기 >' (View details >). Below the header is a black bar with a checkbox labeled '하루동안 이 창을 열지 않음' (Don't open this window for 24 hours) and a button labeled '닫기' (Close).

# | 쿠키와 웹 스토리지

## Web Storage: 쿠키의 라이벌(HTML5에서 등장)

	Cookie	Session Storage (Web Storage)	Local Storage (Web Storage)
개당 사용 가능 용량	최대 4KB	평균 5MB	평균 5MB
유지 기간	설정 가능	한 세션 (탭 닫을 시 내용 제거)	사용자가 삭제하기 전까지
통신 여부	서버와 계속 연결	설정 가능	설정 가능
사용처	기간 설정 필요한 부분 (팝업 창)	보안을 필요로 하는 경우 (은행 로그인 등)	상시 저장이 필요한 경우 (로그인 유지 등)

# 바로 실습...하기 전에 간단히 복습 - ES6 문법

Variable – 변수 선언

## var vs let, const

	재 선언	재 할당
var	O	O
let	X	O
const	X	X

# 바로 실습...하기 전에 간단히 복습 - ES6 문법

## Template String

### Template String (템플릿 리터럴)

이중 따옴표(" ")나 작은 따옴표(' ') 대신 **백틱 (` `)**을 사용하여 변수를 표현하는 문법

``${변수명}``



```
const a = 5;  
const b = 10;  
console.log(`a + b = ${a+b}입니다`);
```

# | 바로 실습!

1. git clone <https://github.com/Kim-Jiseong/2022-Session14-Deploy-Practice.git>
2. pipenv shell
3. pipenv install django
4. cd todo
5. python manage.py migrate
6. python manage.py createsuperuser
7. python manage.py runserver



# | Local Storage 사용법

Local Storage에 아이템 추가: `window.localStorage.setItem(key, value)`

Local Storage 아이템 읽기: `window.localStorage.getItem(key)`

Local Storage 아이템 삭제: `window.localStorage.removeItem(key)`

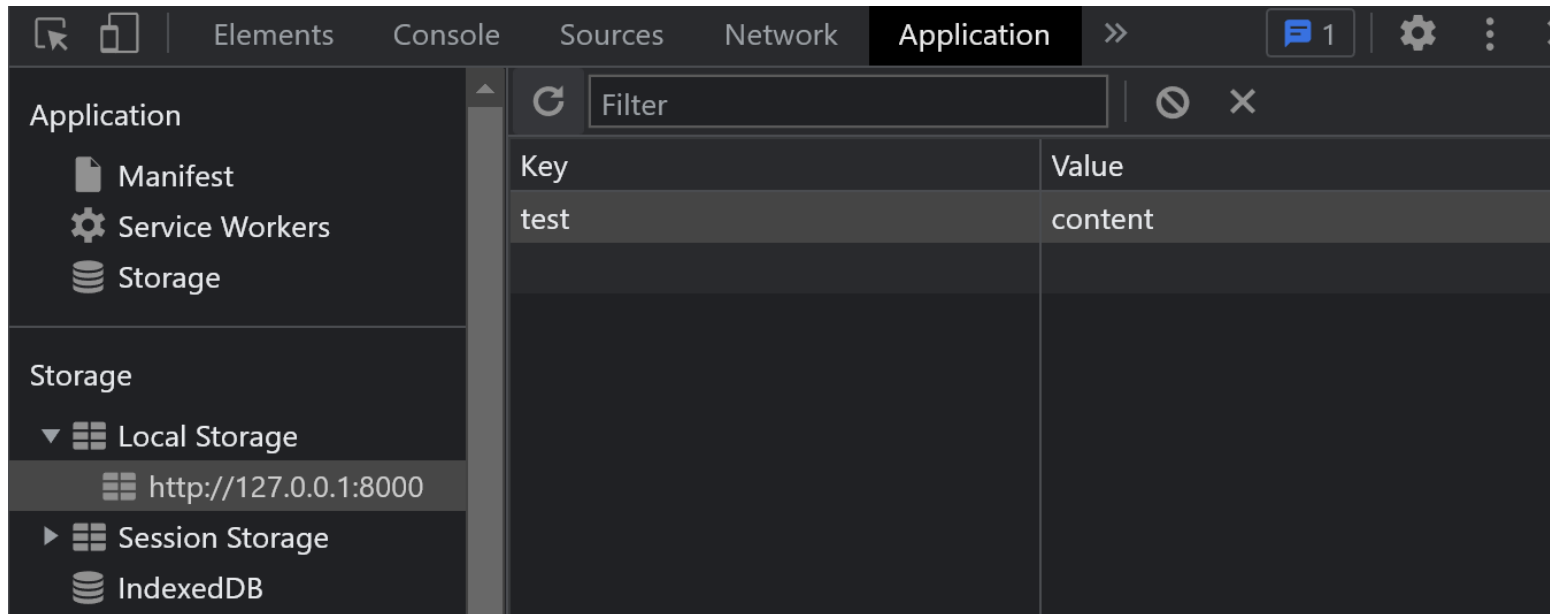
# | Local Storage 예제

base.html 하단 주석 밑에 다음 코드를 복붙해주세요

```
<button type="button" onclick="test()">테스트</button>
<script>const test = () => {
  console.log("localStorage 아이템 생김")
  window.localStorage.setItem('test', 'content')
}
</script>
```

버튼을 누르면??

# | Local Storage 예제



Local Storage 아이템이 만들어졌습니다!!

\*Local Storage 확인법: 크롬 개발자도구 -> Application -> Local Storage

# | Local Storage 실습

main > static > js > localStorage.js 파일에서 비어있는 setUsername() 함수와 resetUsername() 함수를 작성해주세요

setUsername() 함수는 key로 'username', value로 이름을 갖는 아이টে을 생성하며 resetUsername() 함수는 이를 삭제합니다

```
todo > main > static > js > 📄 localStorage.js > ...
1  const username = document.querySelector(".username")
2  const usernameWrapper = document.querySelector(".usernameWrapper")
3  const header = document.querySelector("#header")
4
5  function checkUsername(){
6      const checkName = window.localStorage.getItem('username')
7      if (checkName){
8          usernameWrapper.style.display = 'none';
9          header.innerHTML = `

# ${window.localStorage.getItem('username')} 의 Todo List</h1><button type="button" onclick="resetUsername()">초기화</button>` 10 } else{ 11 usernameWrapper.style.display = 'flex'; 12 header.innerHTML = "" 13 } 14 } 15 16 checkUsername() 17 18 function setUsername() { 19 // 여기에 작성 20 checkUsername() 21 } 22 23 function resetUsername() { 24 // 여기에 작성 25 console.log("username 초기화") 26 checkUsername() 27 }


```

# | Local Storage 실습

제대로 하셨다면 이런 화면이 뜹니다!

## 김지성 의 Todo List

초기화

추가하기

안끝난 일

끝난 일

# | 비동기 실습

이번에는 axios를 이용합니다!

왜? axios인가? == 일반적으로 지금처럼 html만 사용할 경우 fetch나 ajax를 많이 사용하지만, 향후 확장성에서 axios가 뛰어납니다. (그리고 문법이 제일 쉬워요)

# 비동기 실습: 실습코드 해설

```
todo > main > static > js > todos.js > add > data
1  const toDoForm = document.querySelector("#inputForm");
2  const notFinished = document.querySelector("#notFinished");
3  const finished = document.querySelector("#finished");
4  let todolist_notFinished = null
5  let todolist_finished = null
6  let template = null
7
8  function check(id, done) {
9    console.log('id, done', id, done)
10    //여기서부터 직접 작성하세요
11    //item.id와 item.done을 받았습니다. item.done은 item.done의 반대를 나타냅니다. 1를 사용하면 토글 효과를 낼 수 있습니다.
12  }
13
14  function get_finished() {
15    let paintlist = []
16    axios.get("/get_finished")
17      .then((response) => {
18        todolist_finished = response.data.todolist;
19        console.log('finished', todolist_finished);
20        todolist_finished.forEach((item) => {
21          let template = `<div class="todowrapper finished"><div id="inner" onclick="check(${item.id}, ${item.done})">${item.content}</div><div id="delete" onclick="deleteTodo(${item.id})"><i class="fas fa-trash-alt"></i></div></div>`
22          paintlist.push(template)
23        })
24        console.dir(paintlist)
25        finished.innerHTML = paintlist.join('')
26      })
27      .catch(function (error) {
28        console.log(error);
29      });
30  }
31
32  function get_notFinished() {
33    let paintlist = []
34    //여기서부터 직접 작성하세요
35  }
36
37  function deleteTodo(id) {
38    console.log(id)
39    //여기서부터 직접 작성하세요
40  }
41
42  function add() {
43    const input = document.querySelector("#content")
44    const newTodo = input.value;
45    input.value = "";
46    console.log(newTodo)
47    axios('/new',{
48      method: "POST",
49      data: {
50        content: newTodo,
51        done: 'False',
52      }
53    })
54    .then(function (response) {
55      console.log(response);
56      get_notFinished()
57      get_finished();
58    })
59    .catch(function (error) {
60      console.log(error);
61    });
62  }
63
64  function handleToDoSubmit(event) {
65    event.preventDefault(); //새로고침방지
66    add()
67  }
```

# | 비동기 실습

현재 구현된 기능: 할 일을 적고 추가하기를 누르면 안끝난일(class='notFinished')에 할 일이 추가되며 비동기로 안끝난일을 받아와 화면에 출력합니다.

구현해야 하는 기능: 끝난일과 안끝난일을 각각 받아오고, 삭제 버튼을 누르면 할 일이 삭제되며, 할 일의 글 부분(`<div id="inner" onclick="check(${item.id}, ${!item.done})">`)을 클릭하면 끝난일(토글)로 전환되어야 합니다.



# 수고하셨습니다

---

완전한 코드는 세션 종료 후 github에서 받으실 수 있습니다

NEXT X LIKELION 김지성