

ES6 & Todo List 만들기

JavaScript

NEXT X LIKELION 김동현

추천 유튜브

알박한 코딩사전
구독자 4.91만명

홈 동영상 재생목록 커뮤니티 채널 정보

업로드한 동영상 모두 재생

정렬 기준

Virtual DOM
알코*라디오 15:49

DOM은 뭐고 가상 DOM은 뭔가요? (+ Svelte와 React의 ...)

조회수 4.2천회 · 6일 전

APACHE NGINX
알코*라디오 13:46

아파치, NginX, 톨캣이 뭔가요? (+ 웹서버, WAS, 로드밸...

조회수 7.3천회 · 1주 전

CI/CD
알코*라디오 12:48

CI/CD가 뭔가요? (+ 젠킨스, Travis CI, Github Actions)

조회수 6.4천회 · 2주 전

Angular, React, Vue
알코*라디오 8:12

[코드수정본] Angular, React, Vue가 뭔가요?

조회수 2.5천회 · 2주 전

JVM JRE JDK
알코*라디오 11:50

JVM, JRE, JDK가 뭔가요?

조회수 4.5천회 · 3주 전

+ ReactiveX 강좌
반응형 프로그래밍이 뭔가요? (+ ReactiveX 강좌)

조회수 6.6천회 · 4주 전

보안기초
보안기초 - 웹 공격과 방어와 미친토끼 (파라미터 변조,...)

조회수 1.5만회 · 3개월 전

XML, JSON, YAML
알코*라디오 7:44

XML, JSON, YAML이 뭔가요?

조회수 1.9만회 · 3개월 전

<https://www.youtube.com/channel/UC2nkWbaJt1KQDi2r2XclzTQ/videos>

John Ahn
구독자 1.11만명

홈 동영상 재생목록 커뮤니티 채널 정보

구독

Movie App Clone #5
11:02

Movie App Clone #5 Movie Detail Page (React Project...

조회수 1천회 · 1년 전

Movie App Clone #4
21:20

Movie App Clone #4 Landing Page (React Project , MER...

조회수 1.3천회 · 1년 전

HOOKS
9:30

How to Use Redux with React Hooks (REDUX...

조회수 6.1천회 · 1년 전

Movie App Clone #3
7:11

Movie App Clone #3 The MovieDB API Explained (...)

조회수 1.9천회 · 1년 전

Movie App Clone #2
9:57

Movie App Clone #2 Layout and Mongo DB Connection...

조회수 3천회 · 1년 전

Movie App Clone #1
2:19

Movie App Clone #1 Introduction (React Projec...

조회수 6.3천회 · 1년 전

Node JS Upload File W/ CLOUDINARY
14:50

Node JS Upload File W/ CLOUDINARY

조회수 4.8천회 · 1년 전

YouTube Clone
4:38:13

MERN STACK PROJECT - YOUTUBE WEBSITE CLONE...

조회수 2.3만회 · 1년 전

YouTube Clone #19
23:11

YOUTUBE Clone #19 Complete Project (React...

조회수 3.3천회 · 1년 전

YouTube Clone #18
18:01

YOUTUBE Clone #18 Like DisLike Feature(2)...

조회수 1.6천회 · 1년 전

YouTube Clone #17
5:08

YOUTUBE Clone #17 Like DisLike Feature Explained (...)

조회수 2.9천회 · 1년 전

YouTube Clone #16
14:33

YOUTUBE Clone #16 Reply Comment Component (...)

조회수 2.1천회 · 1년 전

<https://www.youtube.com/channel/UCFyXA9x8lpL3EYWeYhj4C4Q/videos>

| 목차

1. ES6 문법 익히기

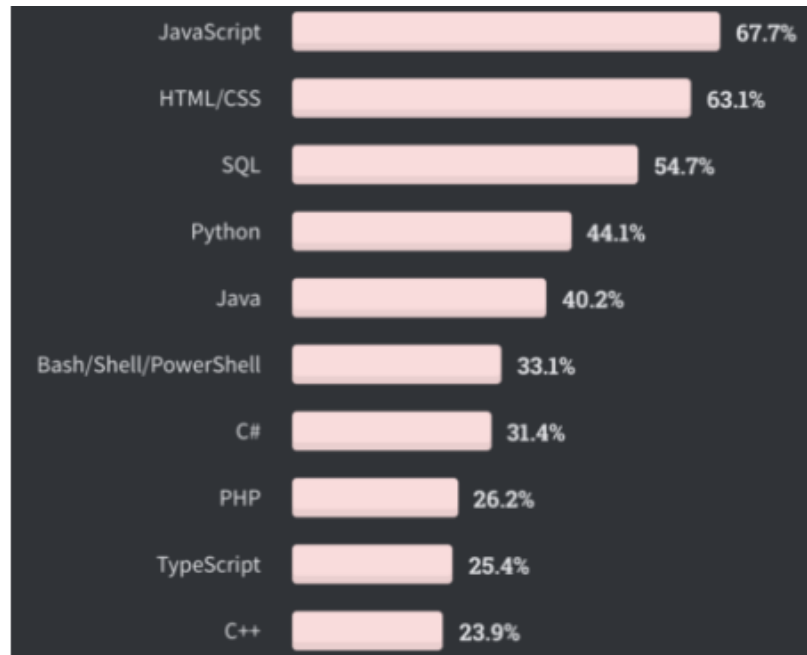
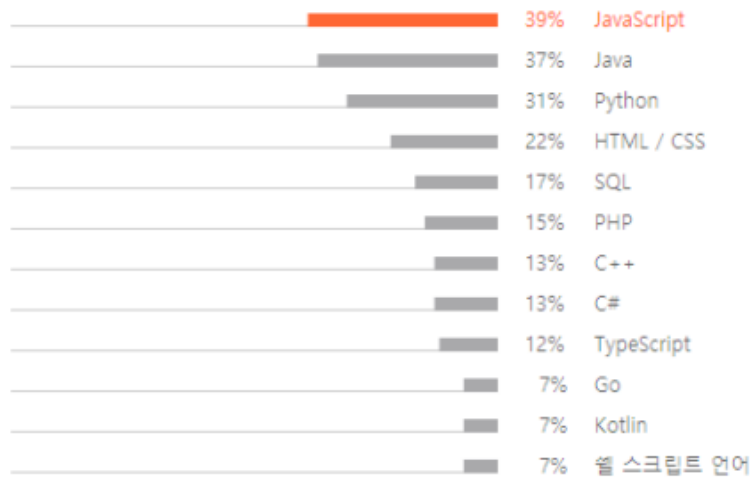
2. Todo List 실습

3. Local Stroage 적용

JavaScript 인기

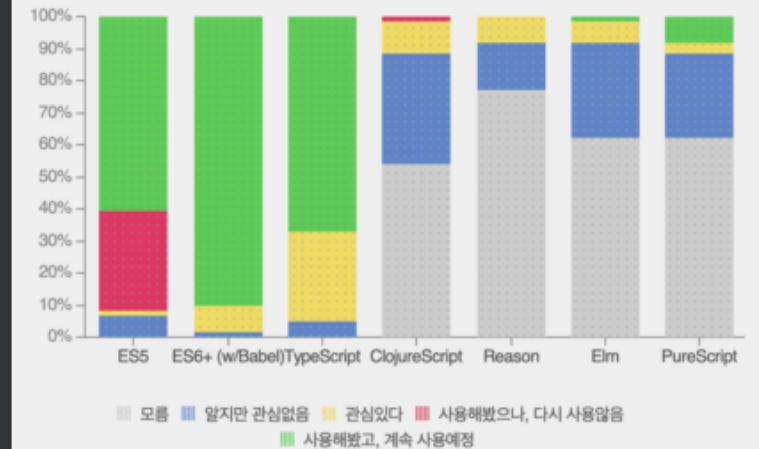
사용하는 주요 프로그래밍 언어는 무엇입니까?

언어 3개까지 선택



어떤 JavaScript를 사용하고 계신가요?

선호도: ES6+ > TypeScript > ES5 > PureScript > Elm > Reason > ClojureScript



| JavaScript의 동향

① 2008년 구글 Chrome 출시 -> ES5, ES6 출시



구글에서 JIT(just-in-time compilation)이라는 강력한 엔진 출시

node.js, NPM, Transpiler, 빌드/번들러 개념 등장

② SPA(Single Page Application)



현재 프론트엔드의 3대장 - ★React, Angular, Vue★

Javascript의 단점을 보완해줄 **Typescript**의 등장

ECMAScript

- ECMAScript는 **스크립트 언어**를 어떻게 만들어야 하는지 알려주는 설명서
- JavaScript는 ECMAScript의 사양을 바탕으로 만들어지는 언어
- ES6는 **ES2015**으로도 불림 (현재 제일 최신 버전은 **ES11**)

ECMAScript

자격 조건

- 사용성을 고려한 UI 설계에 대한 이해도가 높은 분
- HTML5, CSS3, JavaScript (ES6), TypeScript
- 모던 프론트엔드 프레임워크 (React, Angular, Vue) 운영 및 개발 경험
- 상태관리 라이브러리 경험(Redux, MobX 등)

ES6 변경 사항

- Variable
- Destructuring
- Template String
- Arrow Function
- Function Parameter
- Class
- for of
- Object 할당 문법 등등..

ES6 문법

Variable - 변수 선언 방식 변화

var → **let, const**

- 재선언, 재할당 여부
- scope level
- hoisting

ES6 문법

Variable - 변수 선언 방식 변화

var vs let, const

```
var name = '김동현';  
console.log(name);    // 김동현
```

```
var name = '김동현 재할당';  
console.log(name);    // 김동현 재할당
```



ES6 문법

Variable - 변수 선언 방식 변화

var vs let, const

```
var name = '김동현';  
console.log(name); // 김동현
```

```
var name = '김동현 재할당';  
console.log(name); // 김동현 재할당
```

변수 재선언

변수 재할당

변수 재선언 및 재할당 가능

필요할 때 마다 변수를 사용할 수 있다는 장점이 있지만,
같은 변수명을 남용하는 문제를 야기할 수 있음

ES6 문법

Variable - 변수 선언 방식 변화

var vs let, const

```
let name = '김동현';  
console.log(name);      // 김동현
```

```
name = '김동현 재할당';  
console.log(name);      // 김동현 재할당
```

```
let name = '다시 김동현 선언함';  
console.log(name);      // 다시 김동현 선언함
```

ES6 문법

Variable - 변수 선언 방식 변화

var vs let, const

```
let name = '김동현';  
console.log(name); // 김동현
```

```
name = '김동현 재할당';  
console.log(name); // 김동현 재할당
```

```
let name = '다시 김동현 선언함';  
console.log(name); // 다시 김동현 선언함
```

변수 재선언 X
변수 재할당 O
일반적인 변수 개념으로 사용

✖ Uncaught SyntaxError: Identifier 'name' has already been declared VM85428:2

ES6 문법

Variable - 변수 선언 방식 변화

var vs let, const

```
const name = '김동현';  
console.log(name);           // 김동현
```

```
name = '김동현 재할당';  
console.log(name);           // 김동현 재할당
```

```
const name = '다시 김동현 선언함';  
console.log(name);           // 다시 김동현 선언함
```

ES6 문법

Variable - 변수 선언 방식 변화

var vs let, const

```
const name = '김동현';  
console.log(name); // 김동현
```

```
name = '김동현 재할당';  
console.log(name); // 김동현 재할당
```

```
const name = '다시 김동현 선언함';  
console.log(name); // 다시 김동현 선언함
```

변수 재선언 X

변수 재할당 X

일반적인 상수 개념으로 사용

ES6 문법

Variable - 변수 선언 방식 변화

var vs let, const

	재 선언	재 할당
var	O	O
let	X	O
const	X	X

Destructuring (구조 분해 할당)

배열이나 객체 내 변수들을 편하고 쉽게 만들어서 사용할 수 있게 해주는 문법

ES6 문법

Destructuring - 구조 분해 할당

// EX1 - 배열 할당

```
const languages = ['python', 'java', 'javascript']  
let [a, b, c, d='go'] = languages  
console.log(a)    // python  
console.log(d)    // go
```

기존에 정의한 배열을 새로운 변수에 쉽게 할당할 수 있음!

변수 'd' 처럼 기본값을 설정할 수도 있음!!

ES6 문법

Destructuring - 구조 분해 할당

// EX2 - 객체 할당

```
let student = {  
  name: "donghyun",  
  age: 25  
}  
let { name, age } = student  
console.log(name) // donghyun
```

객체의 **key**값을 변수로 할당하여 구조분해할당이 가능

let name = student.**name** 에서 뒤에 .name이 생략된 것

let age = student.**age** 에서 뒤에 .age가 생략된 것

ES6 문법

Destructuring - 구조 분해 할당

// EX3 - 객체 할당2

```
let name = "donghyun", age = 25  
let student = { name: name, age: age }  
  
let student = { name, age }
```

key, value가 같을 때 생략 가능 !

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`);
```

Arrow Function (화살표 함수)

function 키워드보다 간단하게 함수를 표현할 수 있는 문법

(차이점 : 함수 선언식과 표현식의 차이, this 바인딩, 호이스팅..)

```
const double = function(x) {  
  return x*2;  
}
```



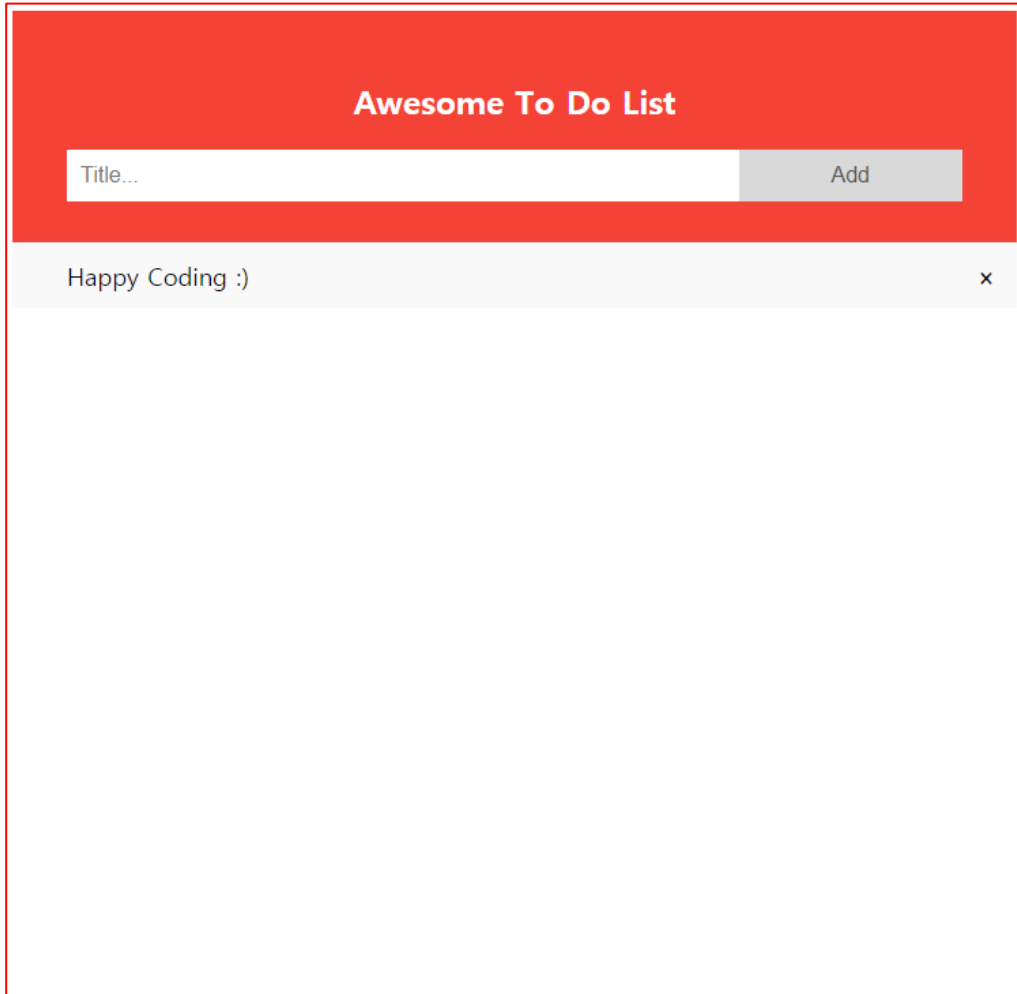
```
const double = (x) => {  
  return x*2;  
}
```

| 쉬는 시간

1. 지금까지 했던 실습 복습

2. 구글에 ES6 문법 검색해보기

| To Do List

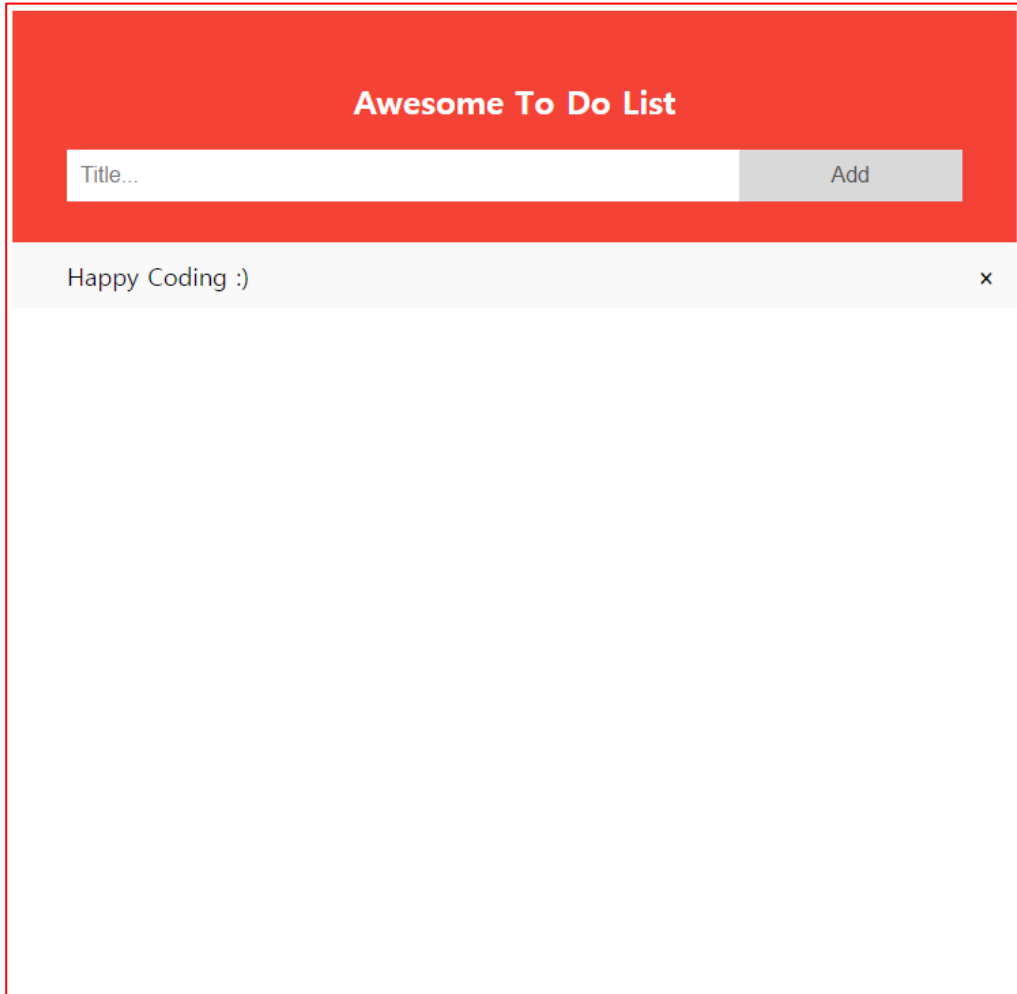


The screenshot shows a web application with a red header bar containing the title "Awesome To Do List". Below the header is a form with a text input field labeled "Title..." and a grey "Add" button. Below the form is a light grey box containing the text "Happy Coding :)" and a small "x" icon in the top right corner. The main content area below is empty.

1. 실습 파일 Live Server로 실행 하기

2. HTML, CSS, JS 구조 파악 하기

| To Do List



Awesome To Do List

Title... Add

Happy Coding :) x

<실습>

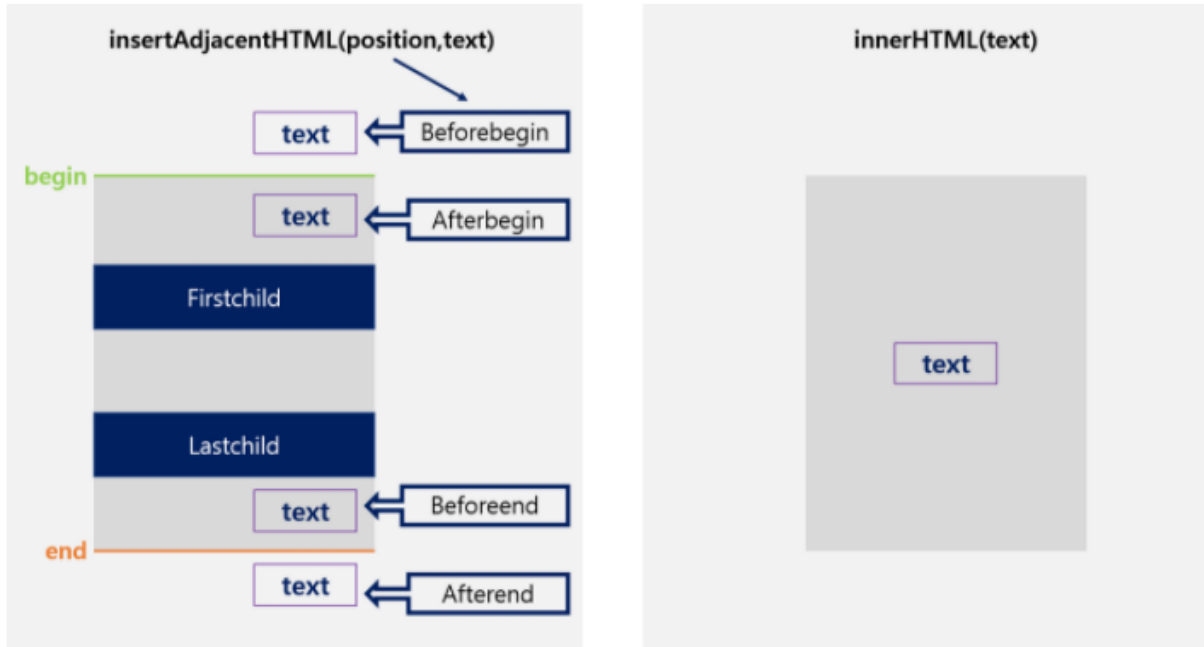
1. var 변수를 let 또는 const로 바꾸기
2. addEventListener 안에 있는 function을 화살표 함수로 바꾸기
3. newElement 함수에서 createElement를 쓰지 않고 Todo Element(li Tag) 만들기

(Hint : insertAdjacentHTML, template literal)

To Do List

insertAdjacentHTML

`element.insertAdjacentHTML(position, text)`



innerHTML처럼 HTML 코드를 text 형식으로
element에 삽입하는 코드지만,
해당 element의 앞에 넣을지, 뒤에 넣을지
순서를 결정할 수 있음!

<예시>

```
$ulElement.insertAdjacentHTML('Afterbegin', '<span>Example</span>')
```

To Do List

정답 코드

```
const $ulElement = document.querySelector('ul');

$ulElement.addEventListener("click", (e) => {
  const $target = e.target;
  if ($target.classList.contains('close')) {
    const $parentTarget = $target.parentElement;
    $parentTarget.style.display = "none";
  }
  $target.classList.toggle('checked');
})

function newElement() {
  const inputValue = document.getElementById("myInput").value;
  const $liElement = `
    <li>
      ${inputValue}
      <span class="close">✕</span>
    </li>
  `

  if (inputValue === "") {
    alert("You must write something!");
  } else {
    $ulElement.insertAdjacentHTML('beforeend', $liElement);
    localStorage.setItem("")
  }
  document.getElementById("myInput").value = "";
}
```

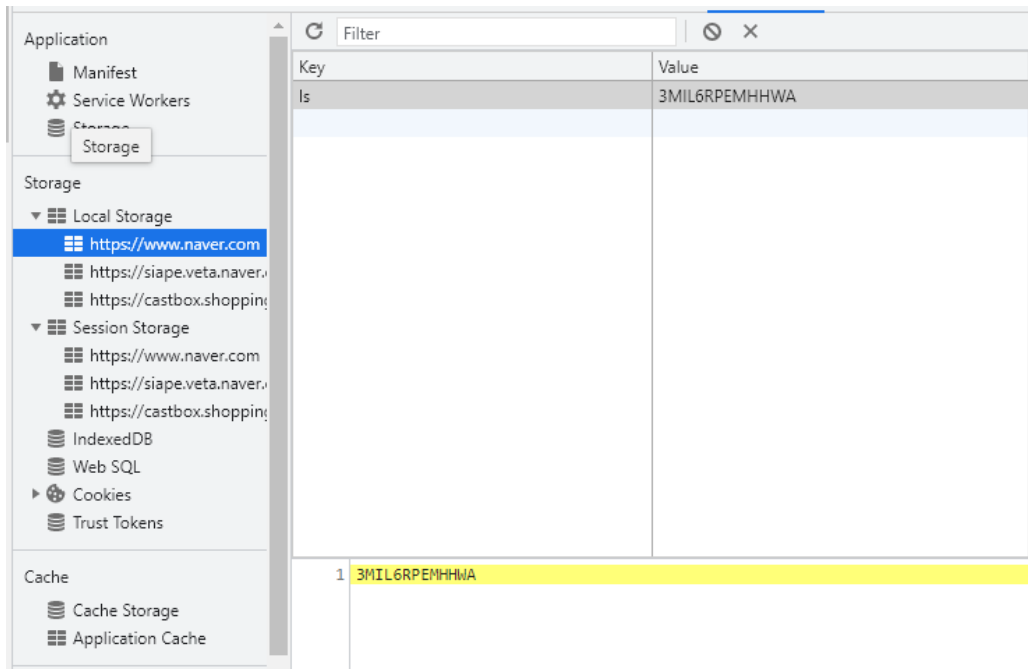
| To Do List

새로고침 하면
다시 처음으로 돌아가지요!?



Local Storage

로컬 스토리지



브라우저에서는 **localStorage**와 **sessionStorage**가 존재

key-value 형태의 데이터를 사용자 로컬에 저장

데이터를 **Obj** 형태로 저장, 덮어쓰기, 삭제 등 조작 가능

장바구니, 팝업창, 서버리스 프로젝트 등에 적용 가능

개발자 도구 > Application

Local Storage

문법 익히기

// 키에 데이터 쓰기

localStorage.setItem("key", value);

// 키로 부터 데이터 읽기

localStorage.getItem("key");

// 키의 데이터 삭제

localStorage.removeItem("key");

// 모든 키의 데이터 삭제

localStorage.clear();

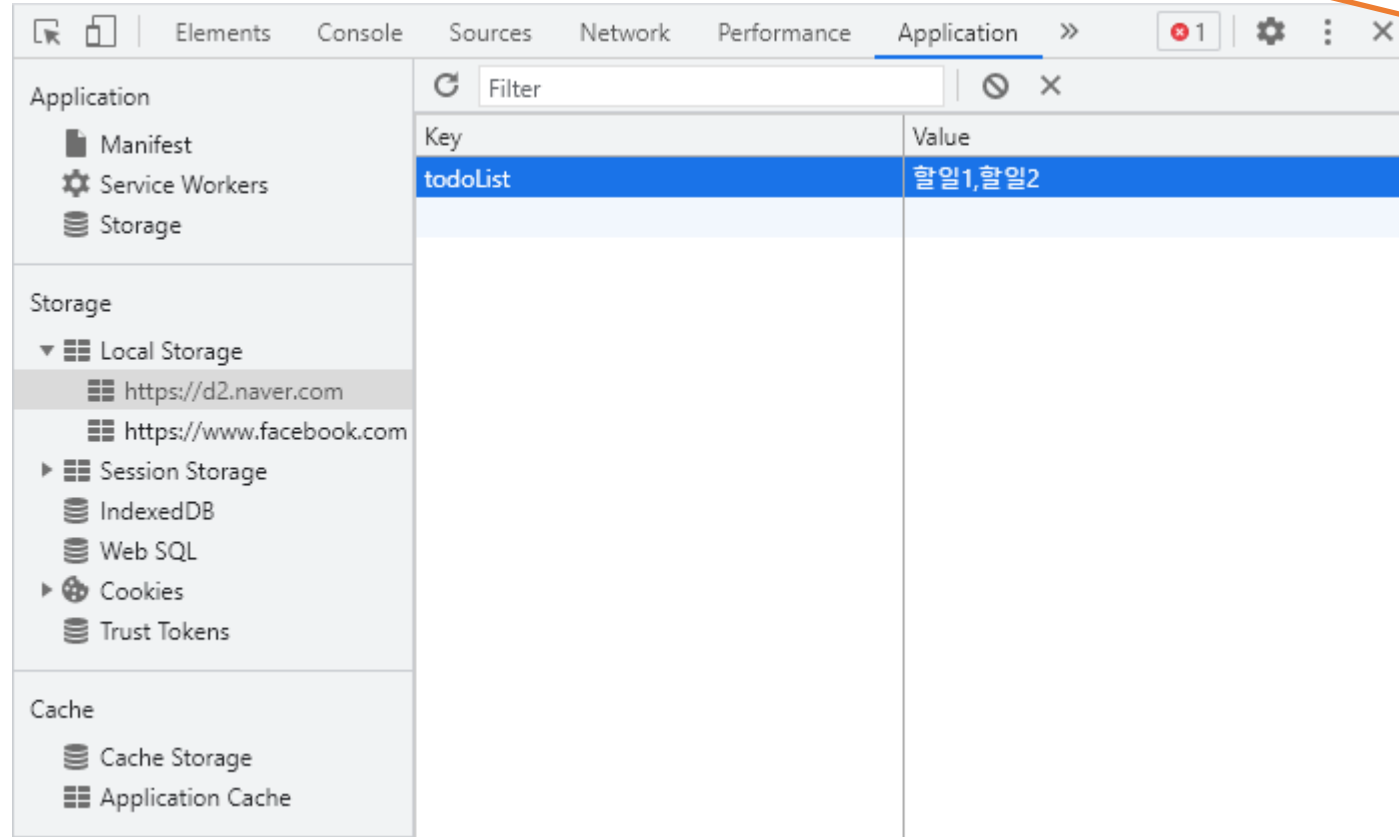
// 저장된 키/값 쌍의 개수

localStorage.length;

Local Storage

배열을 value로 저장할 때

로컬 스토리지는
value를 string으로 저장함



```
localStorage.setItem('todoList', ['할일1', '할일2'])
```

```
localStorage.getItem('todoList')
```

| To Do List 실습

ex1.js 아래에 있는 주석을 풀고 순서에 따라 4개의 함수를 작성해주세요!

① getTodoList

- key를 매개변수를 받았을 때, localStorage에서 key에 해당 되는 value를 반환하는 함수
- 실습에 사용되는 To Do List는 value가 최종적으로 배열 형태가 되어야 하므로 **split**을 이용해서 string -> array로 반환해주기 (value가 undefined인지 확인하는 로직 필요)

② init

- 페이지를 로드했을 때 실행되며, localStorage에 저장된 데이터를 바탕으로 To Do List 구성하는 함수
- ① 함수를 통해 배열 변수를 얻고, for문을 돌면서 tag를 string 형태로 만들어주며, **insertAdjacentHTML** 함수를 통해 ulElement에 삽입해주는 로직 필요

| To Do List 실습

ex1.js 아래에 있는 주석을 풀고 순서에 따라 4개의 함수를 작성해주세요!

③ addTodoList

- key에 해당 되는 localStorage에서 새롭게 입력된 value를 추가하는 함수
- ① 함수를 통해 배열 변수를 얻고, 해당 배열에 value를 추가한 새로운 배열을 만들기
- 최종적으로 localStorage.setItem 문법을 이용하여 업데이트 하기!

④ deleteTodoList

- key에 해당 되는 localStorage에서 특정 value를 삭제하는 함수
- ① 함수를 통해 배열 변수를 얻고, 해당 배열에서 특정 value를 삭제 시켜 새로운 배열을 만들기
- 최종적으로 localStorage.setItem 문법을 이용하여 업데이트 하기!

| To Do List 실습

ex1.js 아래에 있는 주석을 풀고 순서에 따라 4개의 함수를 작성해주세요!

⑤ 원래 있던 코드에 지금까지 만든 함수들 적용하기

- Add 버튼을 눌렀을 때, localStorage에 추가하기
- X 버튼을 눌렀을 때, localStorage에서 해당 value을 삭제하기

(Hint 1 : console.dir(\$parentTarget) 코드를 이용해서 찾고자 하는 value 찾기)

(Hint 2 : childNodes 부분을 찾아 보기)

과제

https://www.w3schools.com/howto/howto_js_todolist.asp

1. 위 사이트에 접속해서, 아래에 있는 기능 중 1가지 이상을 To Do List에 추가하기 (~7/8)

Modal Boxes

Progress Bars

Tooltips

Popups

Collapsible

Loaders

Callout

Toggle Hide/Show

Toggle Dark Mode

| Appendix

<https://blog.banksalad.com/tech/typescript-local-storage/>

LocalStorage는 value에 Object를 저장하면 string 형식으로 저장됨

JSON stringify, JSON parse를 통해 변환해야 하는데 (Deep Copy)

이 과정에서 안정성을 높이기 위해 고민한 과정을 가볍게 한번 읽어보시는걸 추천드립니다~!