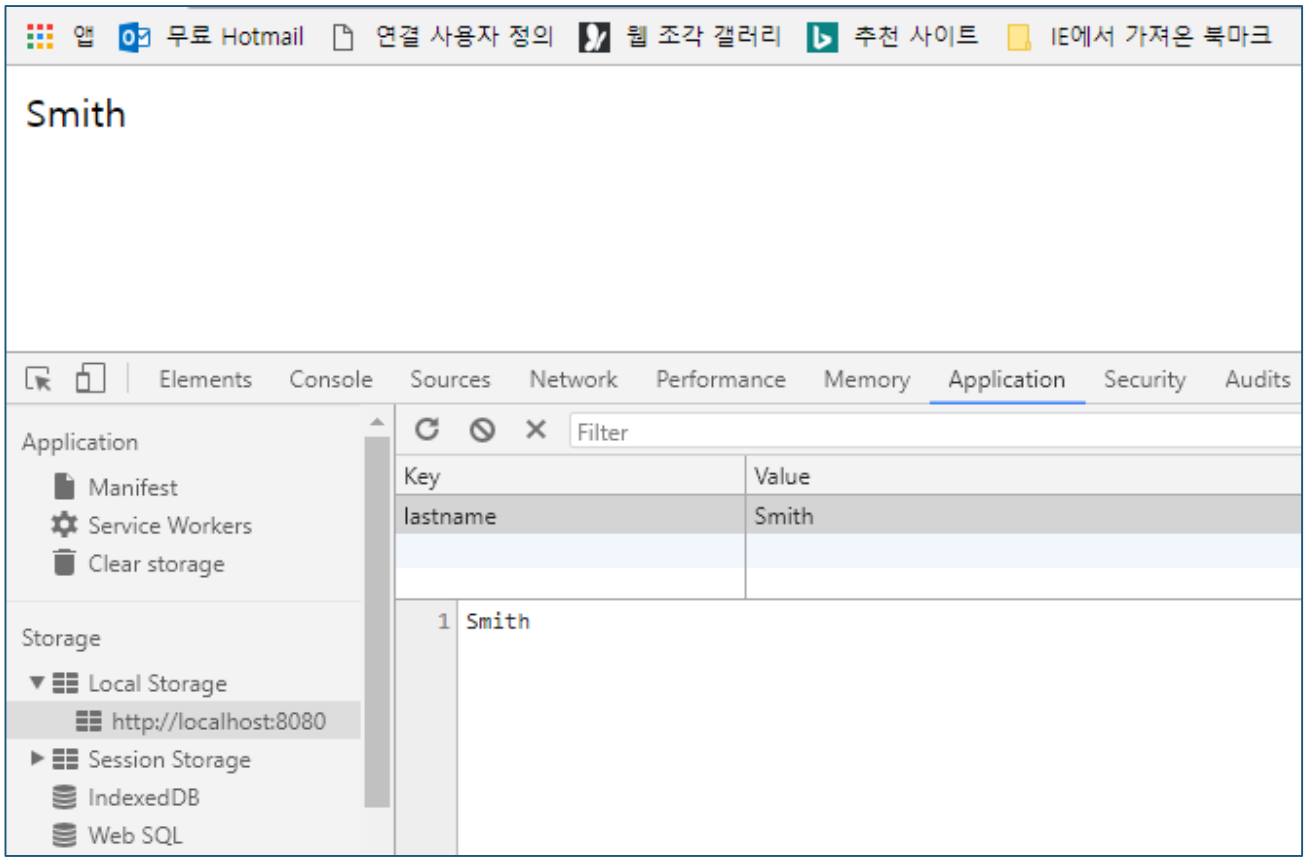
이벤트, 핸들러 , 기타등등:
작지만 알찬 상호 작용

<실습1> 이벤트발생 시 localstorage 저장 – localStorage.setItem(), localStorage.getItem()

- local1.html
- local2.html



- https://www.w3schools.com 방문하여 예제 실습
- 검색어 - HTML Local Storage - Try it Yourself - 실행

<실습2> local3.html – 삭제 – localStorage.clear()

- local2.html에서 clickcount가 10보다 크면 값을 삭제하는 local3.html 을 만들어 보자

local3.html

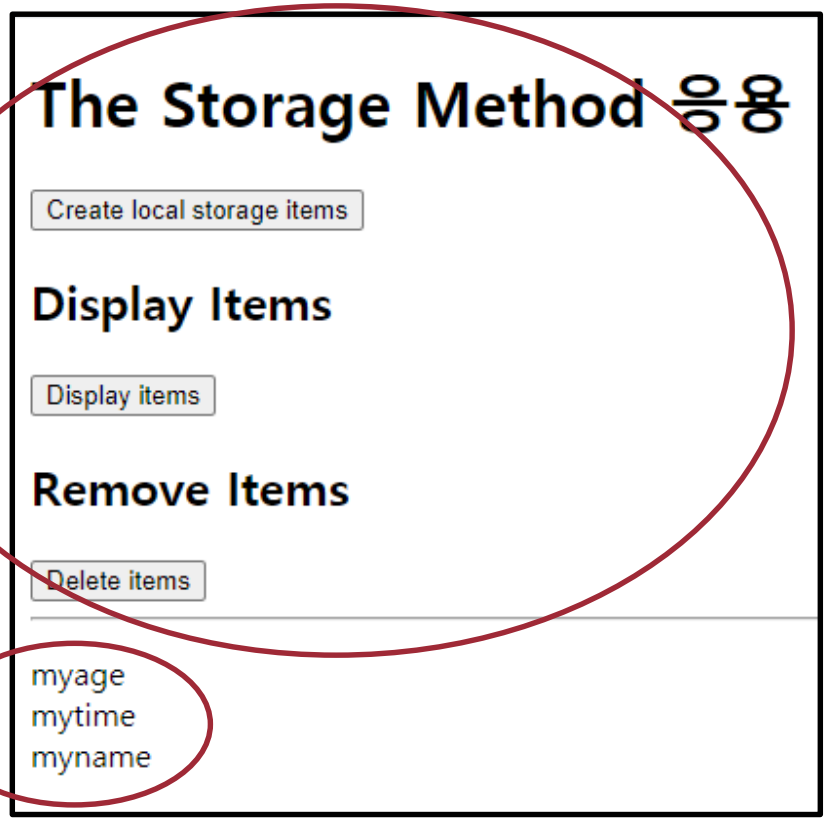
```
if(localStorage.clickcount > 10){
      if(confirm('모두 지울까요?')){
         localStorage.clear()
      }
}
```

<실습3> local4.html – localStorage 응용

- create, display, delete – 함수처리를 하는 button만들고 disaplay 영역 만들기

local4.html

```
<script>
function createItems() {
  localStorage.setItem("mytime", Date.now());
  localStorage.setItem("myname", “Yoon");
  localStorage.setItem("myage", 20);
}

function displayItems() {
  var l, i;
  document.getElementById("demo").innerHTML = "";
  for (i = 0; i < localStorage.length; i++) {
  x = localStorage.key(i);
  document.getElementById("demo").innerHTML += x + "<br>";
  }
}

function deleteItems() {
  localStorage.clear();
}
</script>
```



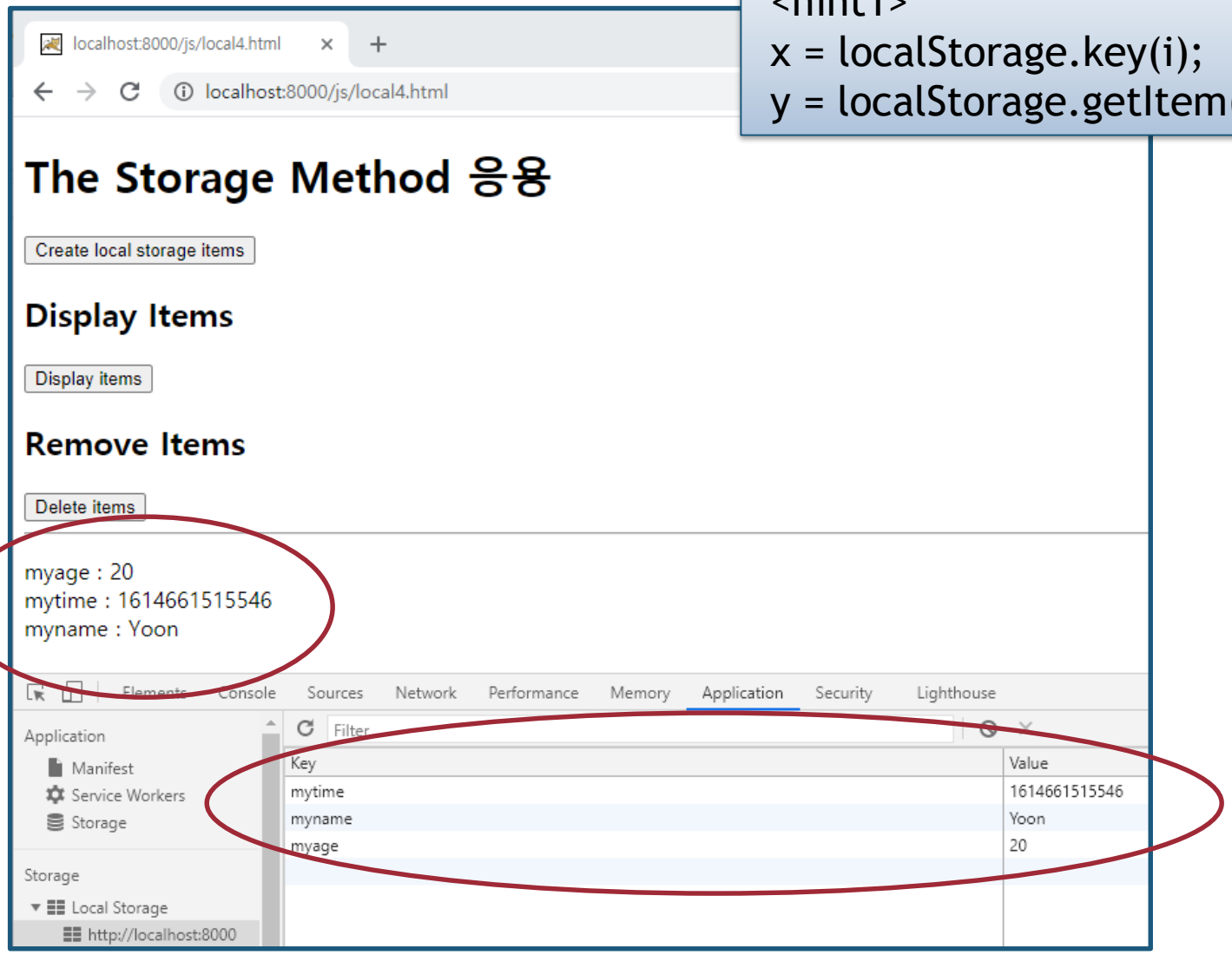
https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_storage_clear

<실습4> local5.html – localStorage 응용

- key와 value를 보여주기

local5.html

<hint1>
x = localStorage.key(i);
y = localStorage.getItem(x);



localstorage의 메소드 5가지

Local Storage 메소드

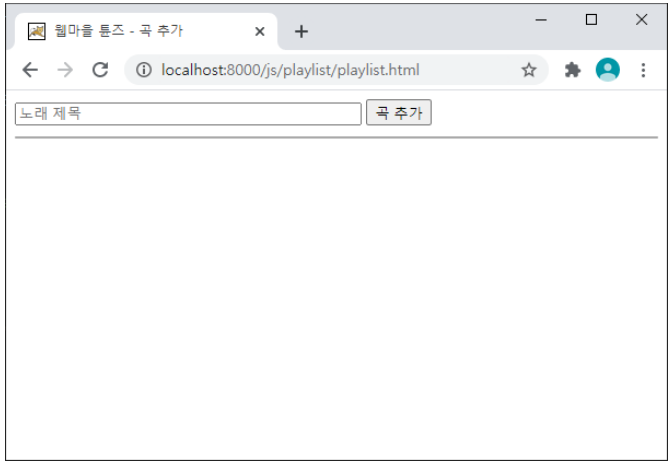
메소드	설명
length	저장된 변수의 개수에 대한 메소드
setItem(key, value)	키, 값를 로컬 스토리지에 저장하는 메소드
getItem(key)	키와 연관된 값을 반환하는 메소드
removeItem(key)	키, 값를 로컬 스토리지에서 삭제하는 메소드
clear()	모든 키와 값를 삭제하는 메소드

Local Storage 메소드 5가지

length : 저장된 변수의 개수에 대한 메소드
setItem(key, value) : 키, 값을 로컬스토리지에 저장하는 메소드
getItem(key) :키와 연관된 값을 반환하는 메소드
removeItem(key) :키, 값을 로컬 스토리지에서 삭제하는 메소드
clear() : 모든 키와 값을 삭제하는 메소드

- **간단히 페이지를 만들어 보자(add song=곡 추가)**

playlist.html



추가된 노래는 localstorage 저장

```
<!doctype html>
<head>
<title>웹마을 튠즈 - 곡 추가</title>
<meta charset="utf-8">
<script src="playlist.js"></script>
<script src="playlist_store.js"></script>
<link rel="stylesheet" href="playlist.css">
</head>
<body>
<form>
  <!-- 추가1 -->
<input type="text" id="songTextInput" size="40" placeholder="노래 제목">
<input type="button" id="addButton" value="곡 추가">

</form>

<ul id="playlist">

</ul>
</body>
</html>
```

버튼 클릭 할 때 JavaScript code가 어떻게 호출 할 것인가?

We need two things: 핸들러 설정

1. 버튼이 클릭되면 '곡추가' 코드를 실행하라고 자바스크립트에 알리는 코드를 연결하는 방법이 필요합니다.

```
button.onclick = handleButtonClick;
```

2. 사용자가 '곡추가' 버튼을 클릭하면 이를 감지하는 자바스크립트 코드가 필요합니다. – 핸들러 작성

```
function handleButtonClick(e) {
    ……
}
```

구체적 계획 – playlist에 노래 추가

목적 : “곡 추가” 을 클릭하면 playlist에 노래 추가

1. 사용자가 “곡 추가” 버튼을 클릭하는 것을 처리하는 핸들러 설정
2. 사용자가 입력한 노래 제목을 얻어보기 위한 핸들러 작성
3. 새로운 노래를 담아 둘 새로운 요소 생성
4. 생성한 요소를 페이지의 DOM에 추가

〈참고〉

브라우저에 html interpreter, xml interpreter등이 존재하여 로딩된 문서를 DOM으로 만들고 브라우저가 닫히면 사라진다.

1. 사용자가 '곡추가' 버튼 클릭하는 것을 처리하는 핸들러 설정

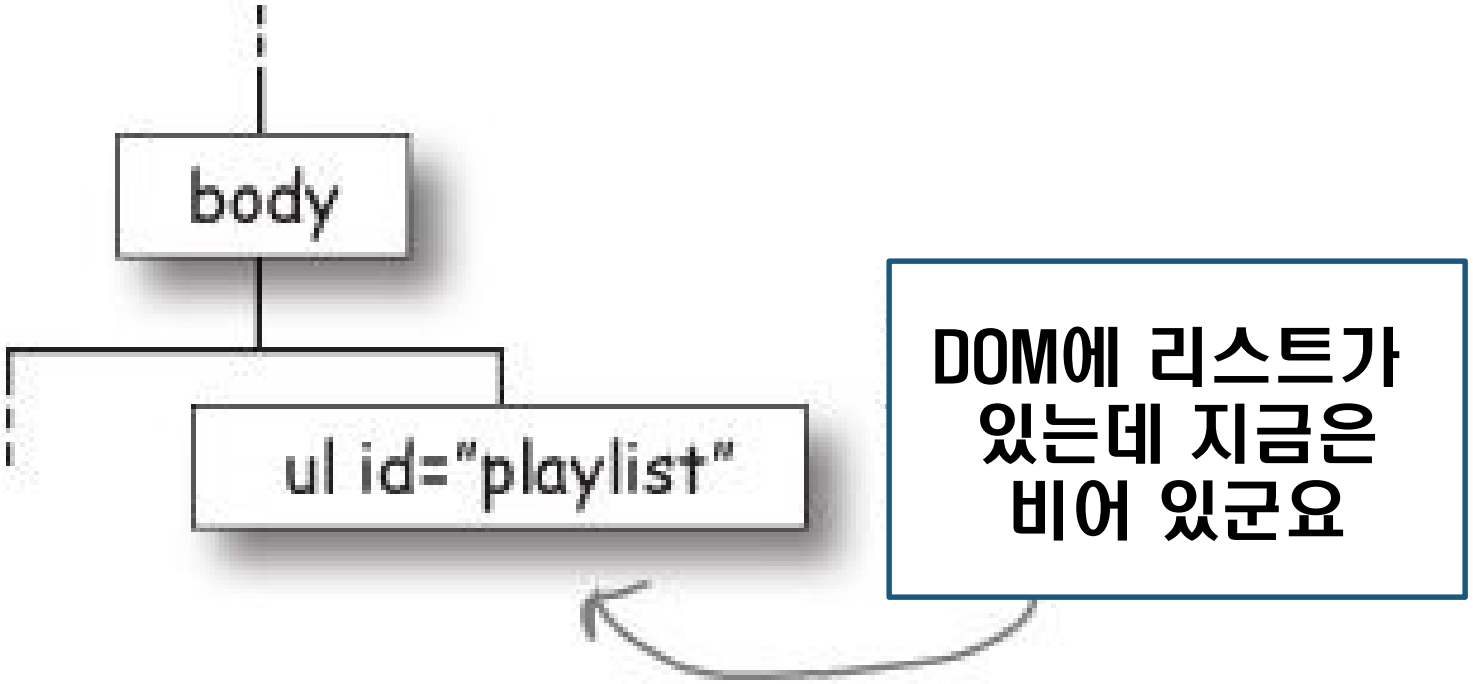
```
var button = document.getElementById("addButton");
button.onclick = handleButtonClick;
```

2. 사용자가 입력한 곡제목을 얻어보기 위한 핸들러 작성

```
window.onload = init;

function init() {
    var button = document.getElementById("addButton");
    button.onclick = handleButtonClick;
}

function handleButtonClick(e) {
    alert(“Button was clicked!”);
}
```

3. 새로운 노래를 담아 둘 새로운 요소 생성 – (1/3)

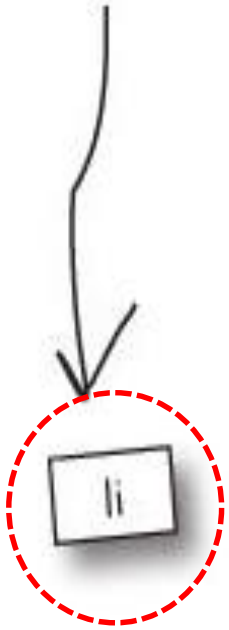
 이 만들어져 있다.



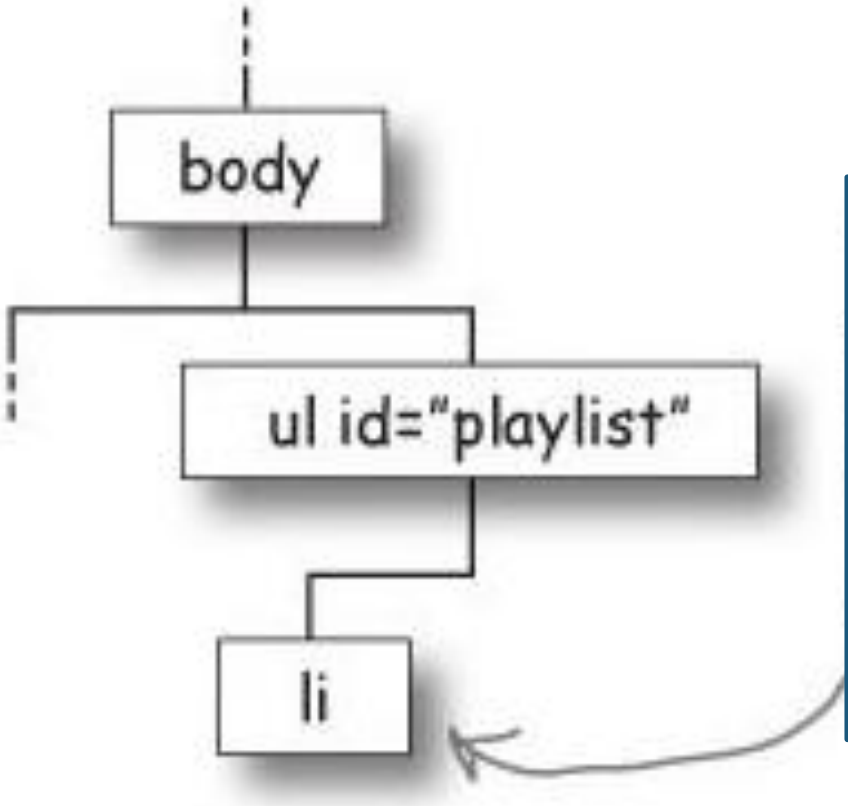
3. 새로운 노래를 담아 둘 새로운 요소 생성 – (2/3)

노래제목을 담을 새로운 요소를 만든다.
Create a <li> element

```
var li = document.createElement("li");
```



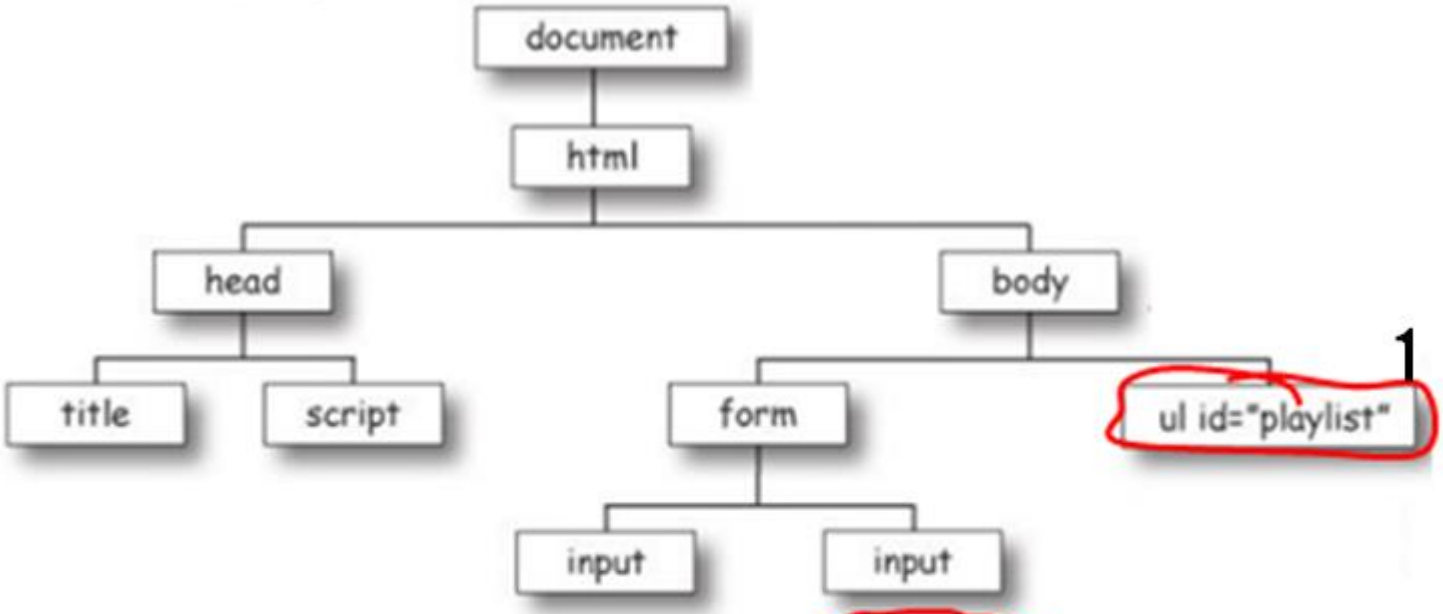
3. 새로운 노래를 담아 둘 새로운 요소 생성 – (3/3)



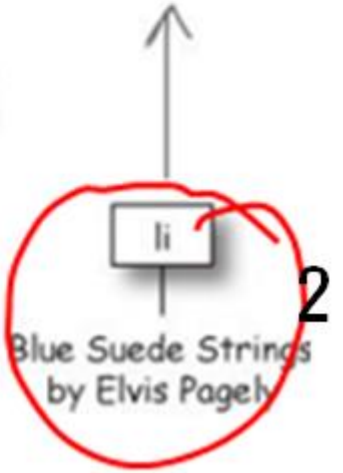
노래를 입력하면 새로운 리스트 항목 을 만들어 <ul> 리스트에 추가할 겁니다.

4. 생성한 요소를 페이지의 DOM에 추가

the top of the tree

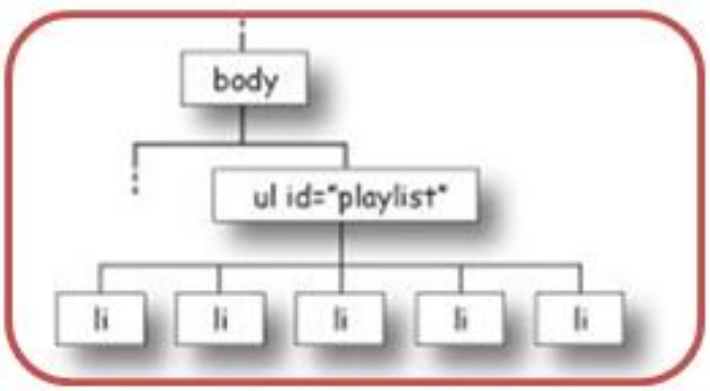


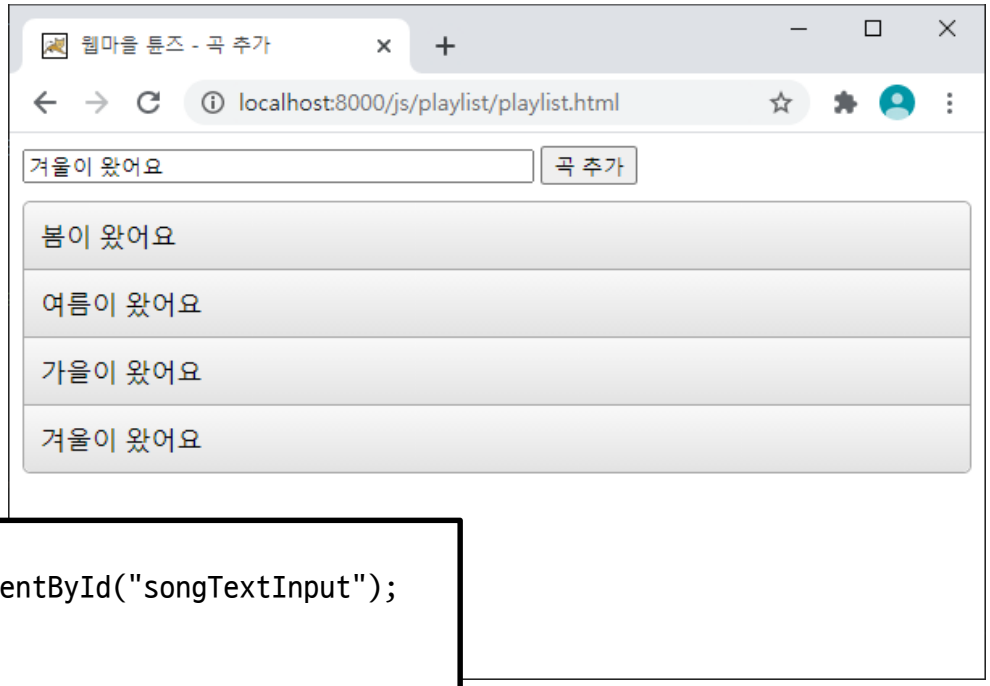

```
1 var ul = document.getElementById("playlist");
2 ul.appendChild(li);
```



<실습5> playlist> playlist.js – 곡추가 핸들러 함수 만들기 – 교재 참조(2/2)

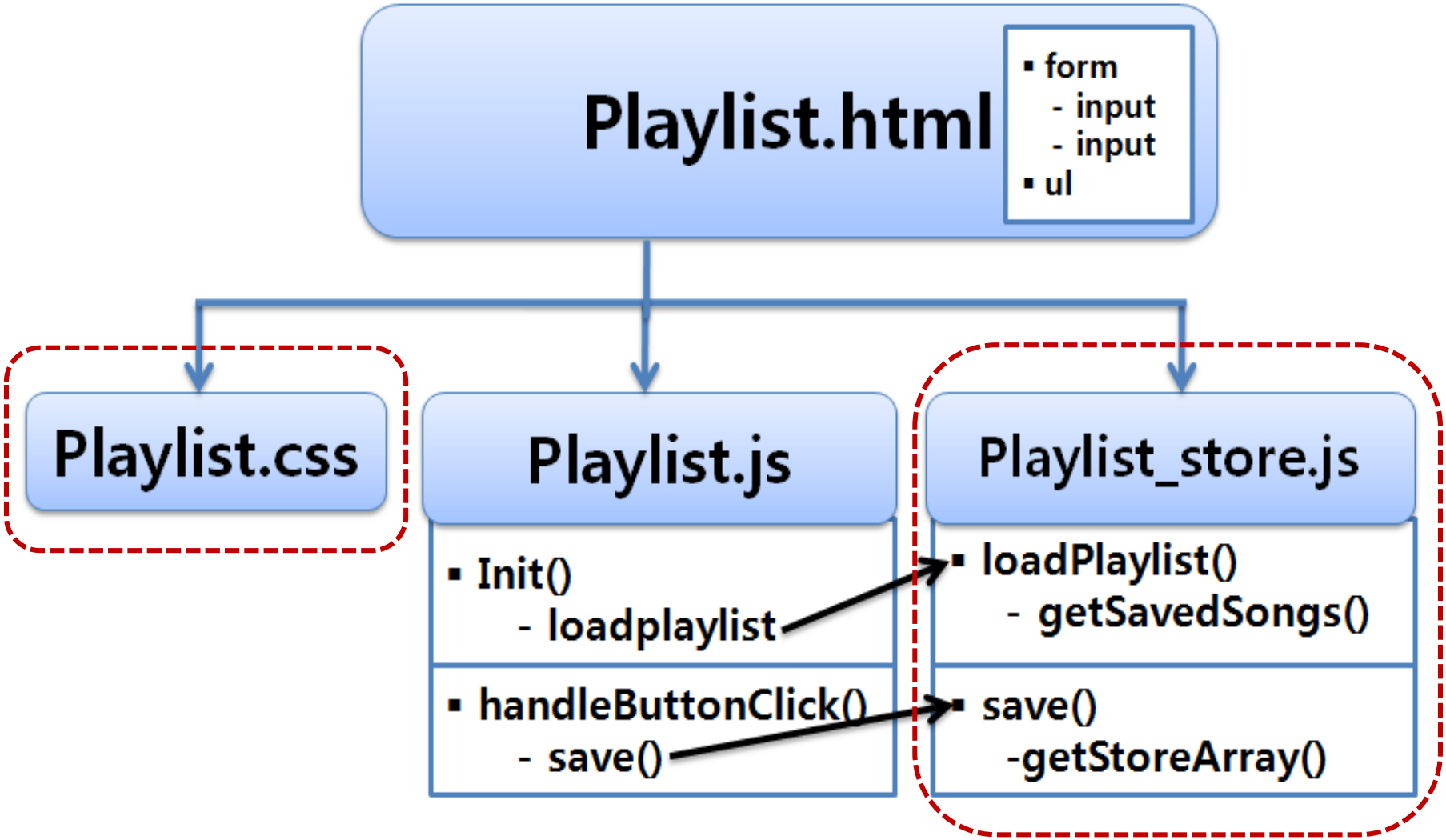
playlist.js 을 완성하시오.





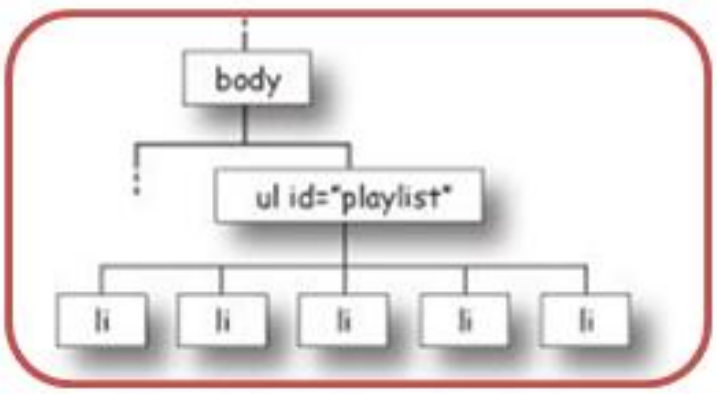

```
function handleButtonClick(e) {
  var textInput = document.getElementById("songTextInput");
  var songName = textInput.value;
  //alert("Adding " + songName);

  if(songName == ""){
    alert("곡을 입력하세요");
  }else{
    var li = document.createElement("li");
    li.innerHTML = songName;
    var ul = document.getElementById("playlist");
    ul.appendChild(li);
    save(songName);
  }
}
```

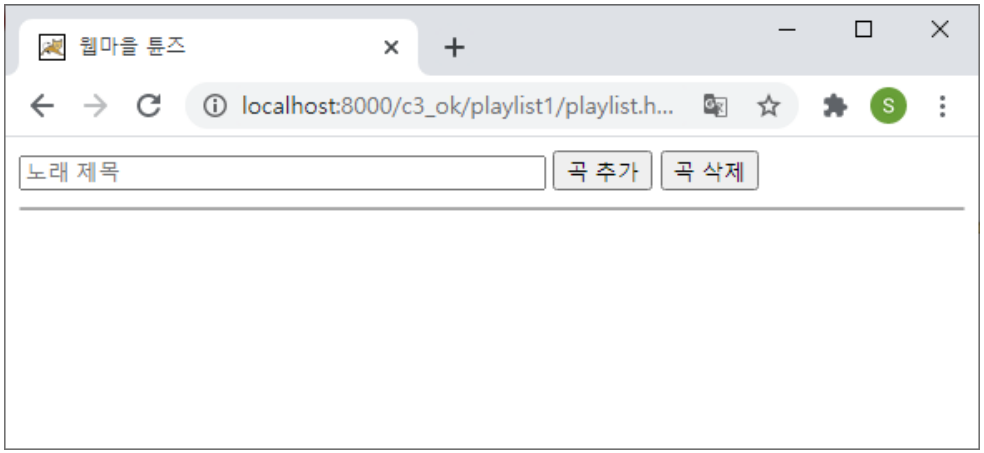
프로그램 구조도 – 준비된 코드



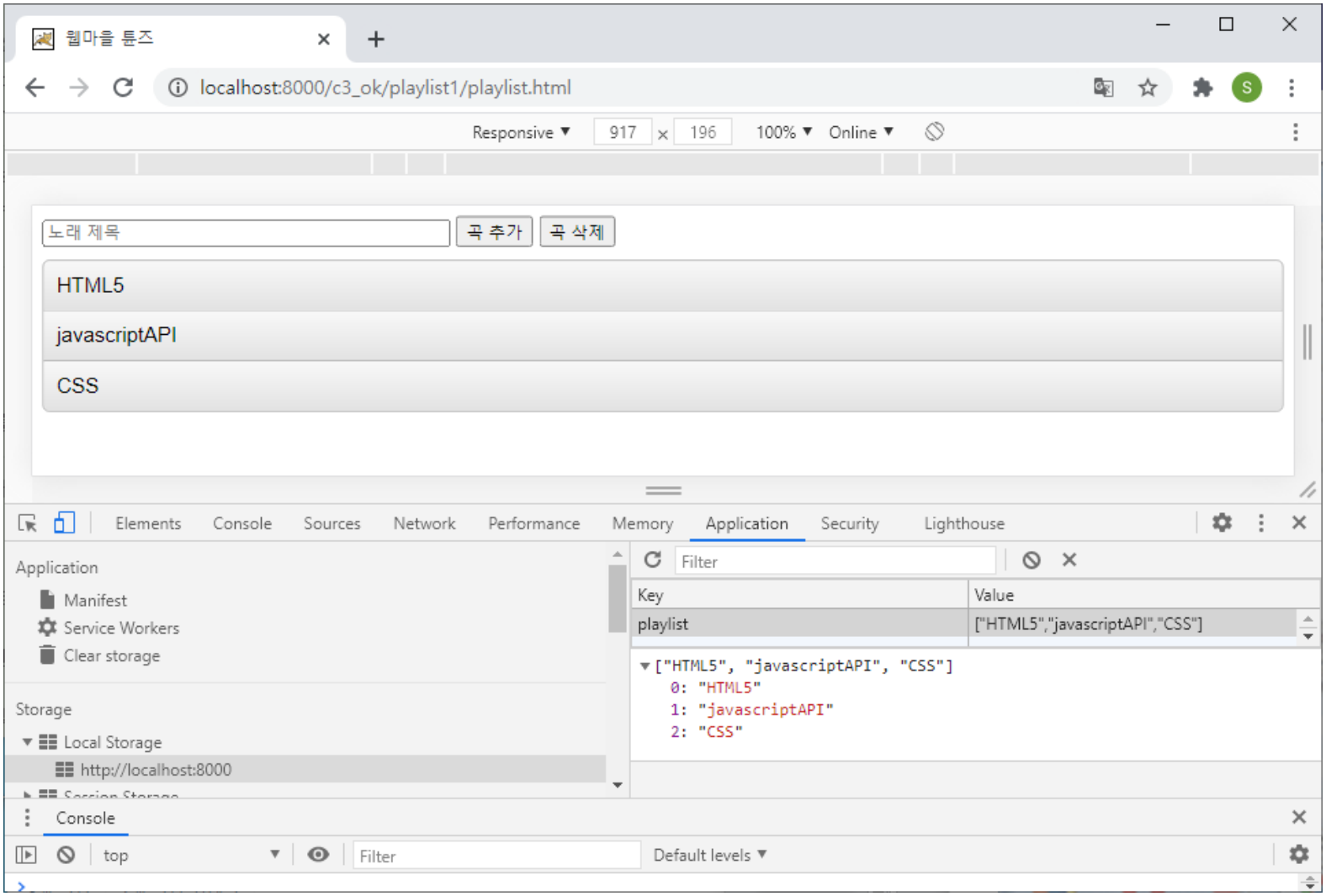
<실습6> playlist1> playlist.html , playlist.js 만들기 – 모든 곡삭제(1/3)

playlist.html, playlist.js를 완성하시오.





<실습6> 모든 곡을 삭제하는 프로그램 실행화면(2/3)



<실습6> 모든 곡을 삭제하기 위한 작업 – localStorage.clear()- (3/3)

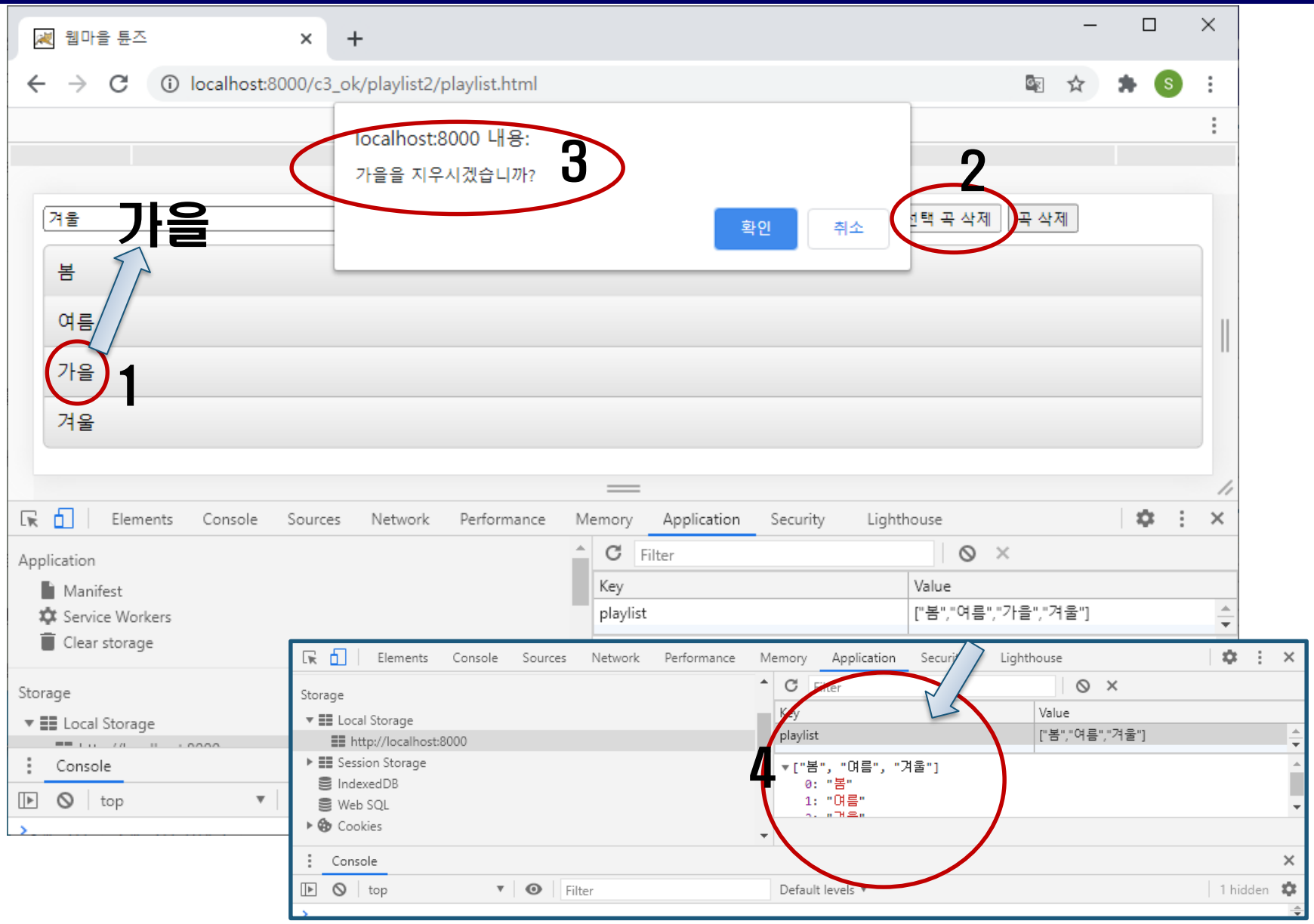
```
<input type="button" id="deleteAllButton" value="곡 모두 삭제">
```

```
  var button = document.getElementById("deleteAllButton");
  button.onclick = handleButtonClick1;
  loadPlaylist();
```

```
function handleButtonClick1(e){
  removeAll();
}
```

```
function removeAll(){
  if(confirm('모두 지울까요?')){
    localStorage.clear();
  }
}
```

<실습7> playlist2> playlist.html , playlist.js 만들기 – 한곡씩삭제 –(1/2)



<실습7> playlist2> 한곡씩 삭제하기 위한 작업 – (2/2)

```
<input type="button" id="deleteButton" value="곡 선택 삭제">
```

```
  var button = document.getElementById("deleteButton");
  button.onclick = handleButtonClick2;
```

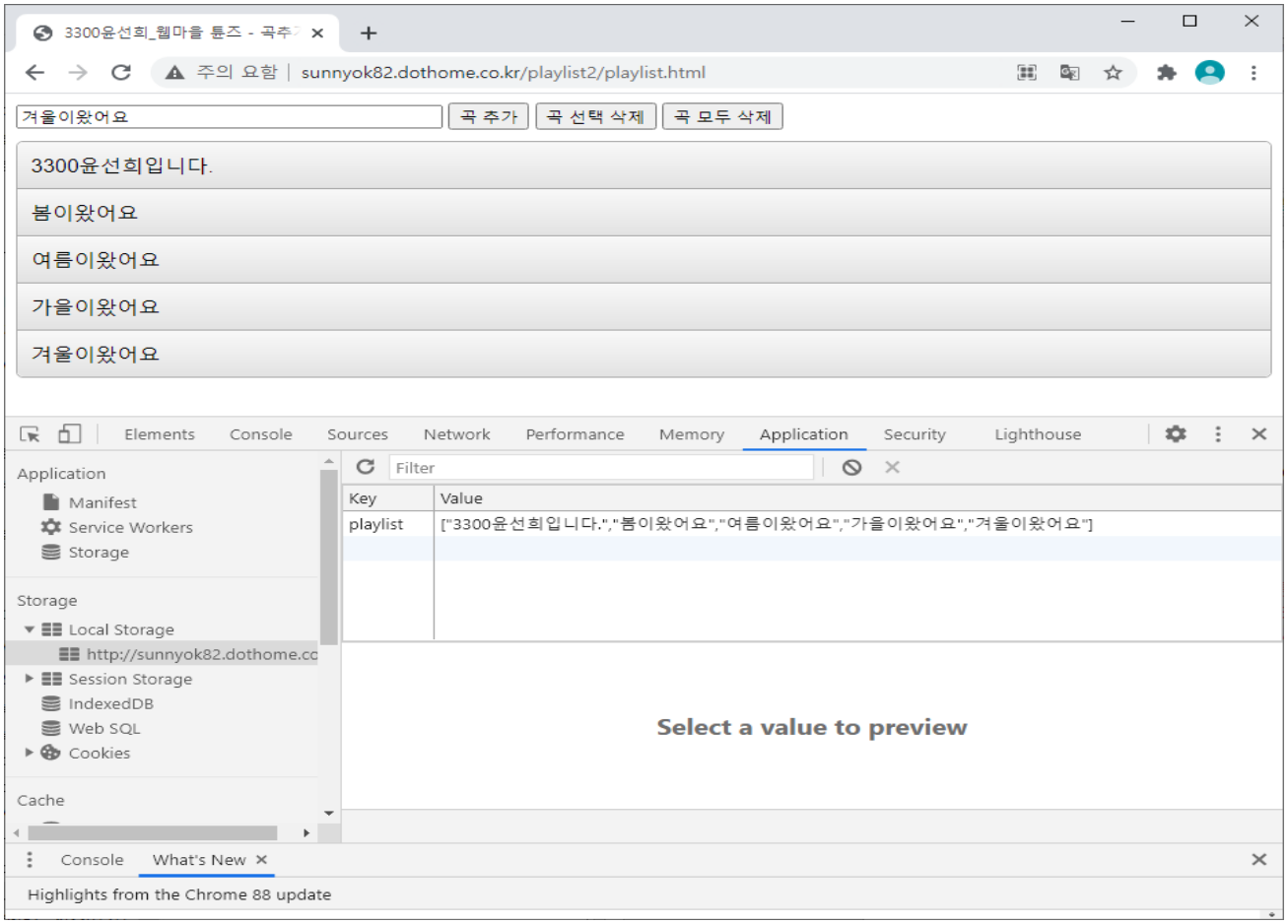
```
  function handleButtonClick2(e){
    var textInput = document.getElementById("songTextInput");
    var songName = textInput.value;
    if(songName==""){
      alert("곡을 입력하세요");
    }else{
      remove(songName);
    }
  }
```

```
function remove(songName){
  if(confirm(songName+'을 지울까요?')){
    var playlistArray = getSavedSongs();

    if (playlistArray != null) {
      for (var i = 0; i < playlistArray.length; i++) {//저장된 songName을 모두 지우기 위해
        if(playlistArray[i]==songName)
           playlistArray.splice(i--,1);
         }
      localStorage.setItem("playlist", JSON.stringify(playlistArray));
    }
  }
location.reload();
}
```

location.reload(); 활용

<실습8> netlify에 올려 웹호스팅 하여보자

- netlify_webhosting 참조



playlist.html=> index.html 로 수정

Q & A