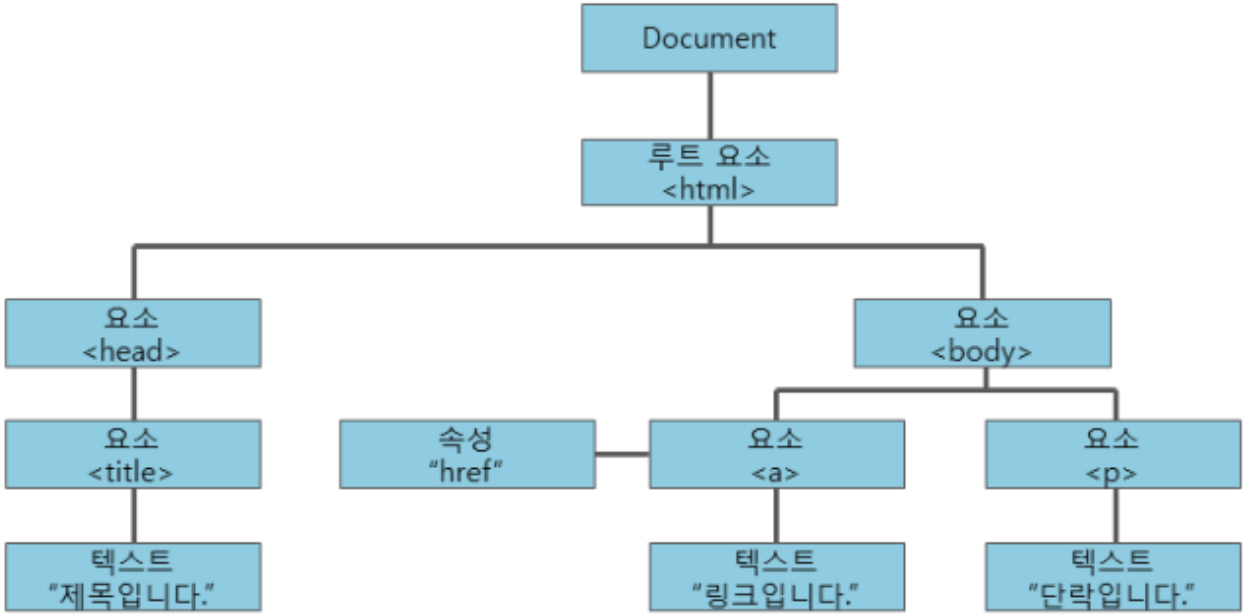
10강
DOM
(document Object Model)
문서 객체 모델

JavaScript

➢ DOM(Document Object Model) 이란 ?

- 문서 객체 모델(DOM, Document Object Model)은 XML이나 HTML 문서에 접근하기 위한 일종의 인터페이스이다.
- 웹 문서의 모든 요소를 자바스크립트를 이용하여 조작 할 수 있도록 객체를 사용해 문서를 해석하는 방법
- 이 객체 모델은 문서 내의 모든 요소를 정의하고, 각각의 요소에 접근하는 방법을 제공한다.
- 이러한 DOM은 W3C의 표준 객체 모델이며, 다음과 같이 계층 구조로 표현된다.



JavaScript

➢ Document

- Document는 웹 문서 자체를 가리키는 DOM 요소 중 하나이다.
- 콘솔 창에 document 입력 후 ▶ 클릭 후 결과 값 확인하면 웹 문서의 소스가 그대로 들어있다. 고로 document 를 사용하면 자바스크립트에서 웹 문서의 소스 전부를 인식 할 수 있기 때문에 수정도 가능하다.

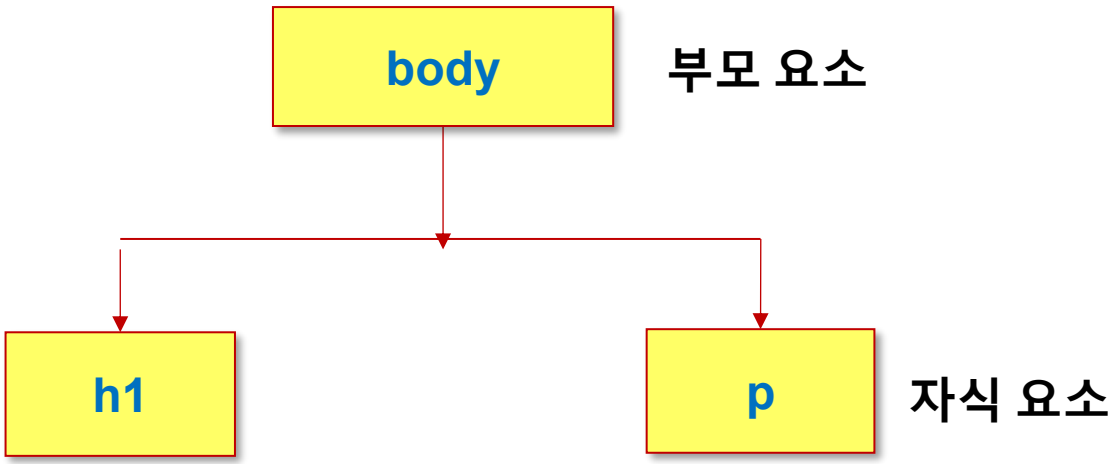
```
> document
<· ▼#document
       <!DOCTYPE html>
       <html lang="en">
       ▶ <head>…</head>
       ▶ <body>…</body>
       </html>
```

JavaScript

➢ DOM 트리(tree)

- DOM은 웹 문서의 요소를 부모요소와 자식 요소로 구분한다.

```
<body>
  <h1>제목</h1>
  <p>본문</p>
</body>
```

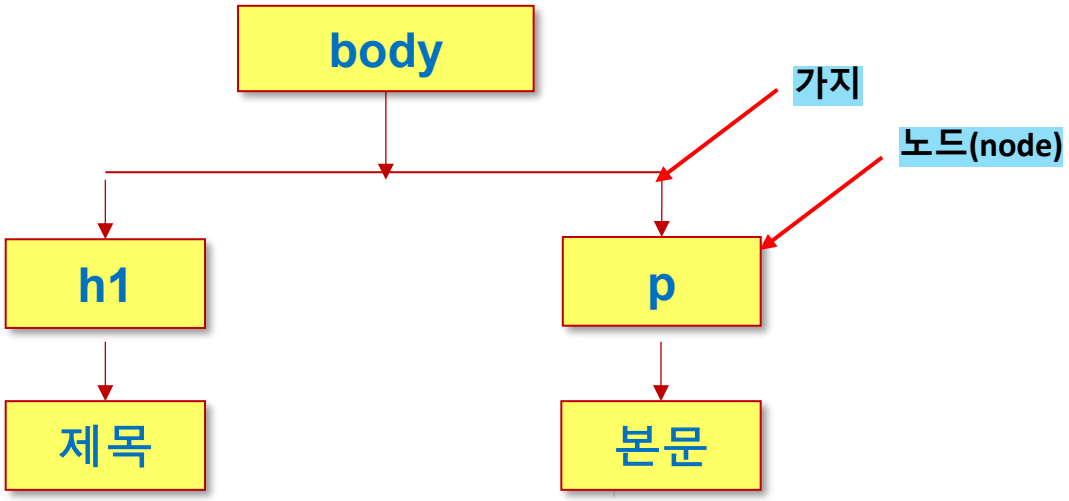


JavaScript

➢ DOM 트리(tree) – 가지, 노드

- DOM 트리는 가지와 노드(node)로 표현한다.
- 노드(node)는 그림에서 네모상자를 카리키고, 웹 문서의 요소나 속성을 나타낸다.
- 가지는 그림의 얇은 선을 카리키는 것으로 노드와 노드사이의 연결관계를 나타낸다.
- DOM 트리는 웹 문서의 HTML 요소만 표현하지 않는다. HTML요소가 품고 있는 텍스트, 이미지도 자식으로 간주한다.

```
<body>
  <h1>제목</h1>
  <p>본문</p>
</body>
```



JavaScript

➢ DOM 트리(tree) 웹 문서 요소 표현

- 웹 문서의 태그는 **요소(Element)노드**로 표현한다.
- 태그가 품고 있는 텍스트는 해당 요소 노드(태그)의 자식 노드인 **텍스트(Text)노드**로 표현한다.
- 태그의 속성은 모두 해당 요소 노드(태그)의 자식 노드인 **속성(Attribute)노드**로 표현한다.
- 주석은 **주석(Comment)노드**로 표현한다.

```
노드 종류: 요소 노드, 속성 노드, 텍스트 노드

<h1 id="title">Hello</h1>

요소 노드:   <h1>
속성 노드:    id="title"
텍스트 노드: "Hello"
```

➢ DOM을 사용하는 이유

- 자바스크립트를 통해 HTML에서 데이터를 가져오고 싶을 때
- 웹 페이지 데이터를 동적으로 변경하고 싶을 때
- interactive 한 웹 애플리케이션(Web App)을 만들고 싶을 때

➢ DOM 요소에 접근하기

- 요소들을 이용하기 위해서는 아래와 같은 메서드로 사용해 특정 태그에 접근해야 된다. 이 메서드를 이용해 선택한 요소를 변수에 대입하여 사용할 수 있다.

메서드	설명
document.getElementById("id명")	해당 id명을 가진 요소 하나를 반환합니다.
document.querySelector("선택자")	해당 선택자를 만족하는 요소 하나를 반환합니다.
document.getElementsByClassName("class명")[순서]	해당 class명을 가진 요소들을 배열에 담아 인덱스에 맞는 요소를 반환합니다.
document.getElementsByTagName("태그명")[순서]	해당 태그명을 가진 요소들을 배열에 담아 인덱스에 맞는 요소를 반환합니다.
document.querySelectorAll("선택자명")[순서]	해당 선택자를 만족하는 모든 요소들을 배열에 인덱스에 맞는 요소를 반환합니다.

JavaScript

➢ getElementById( )와 querySelector( ) 차이

- getElementById()함수를 비롯한 getElementsByClassName(), getElementsByTagName()함수들은 DOM의 요소 노드(Element Node)를 선택할 때 사용된다.
- 반면에 querySelector()와 querySelectorAll()함수도 마찬가지로 요소 노드를 선택하지만, CSS 선택자 문법을 사용할 수 있어 더 유연하고 세밀하게 요소를 찾을 수 있다.
- 자바스크립트 프로그램에서 단순히 특정 웹 요소를 선택해서 변경한다면 getElementById(), getElementsByClassName(), getElementsByTagName() 함수를 사용하면 되고, 보다 다양한 조건으로 요소를 선택하거나 세밀하게 제어하고 싶다면 querySelector(), querySelectorAll() 함수를 사용하면 된다.
- 요소의 텍스트나 속성에 접근하려면, 선택된 요소에서 .textContent, .innerText, .getAttribute(), .setAttribute() 등의 프로퍼티/메서드를 사용해야 한다.

➢ DOM 요소에 속성 노드 접근하기

- HTML 태그 속성을 가져오거나 수정하는 함수 – getAttribute(), setAttribute()

메서드	설명
document.querySelector("선택자"). getAttribute("속성명")	해당 선택자의 속성에 접근 한다.
document.querySelector("선택자"). setAttribute("속성명")	해당 선택자의 속성에 접근하여 수정한다.

JavaScript

➢ DOM 접근하기 주요 메서드

```
document.getElementById("id명")
document.querySelector("CSS선택자")
document.querySelectorAll("CSS선택자")
```

- `querySelector` → 가장 많이 쓰는 방식 (CSS 선택자 그대로 활용)

JavaScript

➢ 요소 탐색

```
let parent = el.parentNode; // 부모
let child = el.children[0]; // 첫번째 자식
let next = el.nextElementSibling; // 다음 형제
```

➢ DOM 조작 1: 텍스트 변경

```
<h1 id="title">안녕하세요</h1>
<script>
document.getElementById("title").textContent = "DOM 조작 성공!";
</script>
```

➢ DOM 조작 2: 속성 변경

```
<img id="photo" src="a.jpg" width="200">
<script>
document.getElementById("photo").setAttribute("src", "b.jpg");
</script>
```

➢ DOM 조작 3: 스타일 변경

```
<p id="msg">이 글자는 파란색으로 바뀝니다.</p>
<script>
document.getElementById("msg").style.color = "blue";
</script>
```

➢ DOM 조작 4: 요소 추가

```
<ul id="list">
<li>기존 항목</li>
</ul>
<script>
let li = document.createElement("li");
li.textContent = "새 항목";
document.getElementById("list").appendChild(li);
</script>
```

➢ DOM 조작 5: 요소 삭제

```
<ul id="list">
<li id="item">삭제될 항목</li>
</ul>
<script>
document.getElementById("item").remove();
</script>
```

JavaScript

➢ 이벤트와 DOM - addEventListener()란

- 특정 DOM 요소(HTML 태그)에 대해, 어떤 이벤트(클릭, 키 입력, 마우스 이동 등)가 발생했을 때 실행할 함수(이벤트 핸들러)를 등록하는 기능이다.
- 기존의 onclick="" 같은 인라인 방식 대신, JS 코드 안에서 이벤트를 깔끔하게 제어할 수 있다.

이벤트 종류 type, 문자열)

1. "click" : 클릭
2. "mouseover" : 마우스 올렸을 때
3. "mouseout" : 마우스 벗어났을 때
4. "keydown" : 키보드 눌렀을 때
5. "submit" : 폼 제출될 때

```
<button id="btn">눌러보세요</button>
<script>
document.getElementById("btn").addEventListener("click", () => {
alert("버튼 클릭됨!");
});
</script>
```

onClick 없애도됨

➢ 글자 색 변경 간단한 예제

```
<h1 id="title">Hello</h1>
<button onclick="changeText()">바꾸기</button>
<script>
function changeText() {
let el = document.getElementById("title");
    el.textContent = "안녕하세요!";
    el.style.color = "red";
}
</script>
```

➢ 리스트 추가 간단한 예제

```
<ul id="list">
<li>첫 번째</li>
</ul>
<button onclick="addItem()">추가</button>
<script>
function addItem() {
  let li = document.createElement("li");
  li.textContent = "새 항목";
  document.getElementById("list").appendChild(li);
}
</script>
```

➢ 이미지 교체 간단한 예제

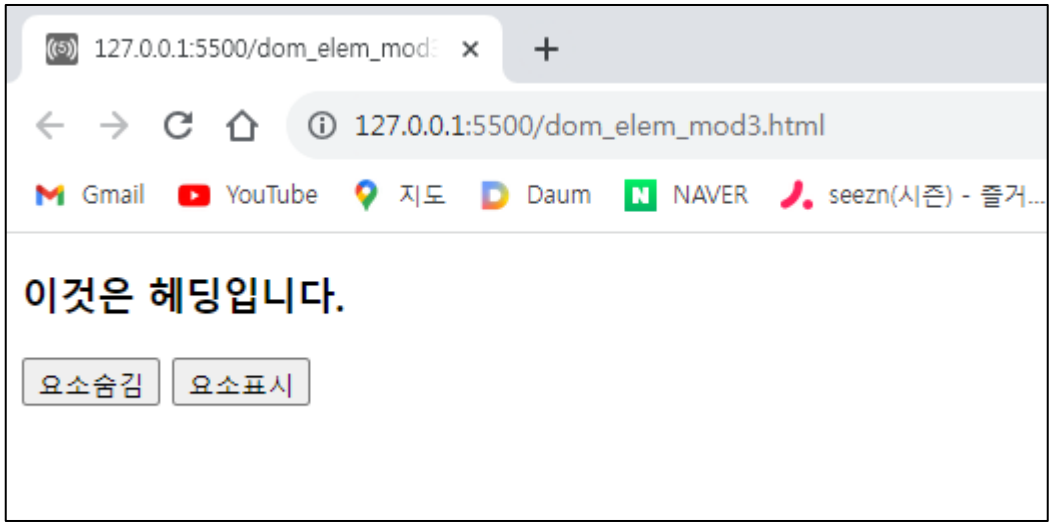
```
<img id="img" src="a.jpg" width="200">
<button onclick="changeImg()">교체</button>
<script>
function changeImg() {
    document.getElementById("img").src = "b.jpg";
}
</script>
```

DOM(document Object Model) 간단한 예제

JavaScript

```
<h3 id="heading">이것은 헤딩입니다.</h3>


<input type="button" value="요소숨김" onclick="document.getElementById('heading').style.visibility = 'hidden'" />
<input type="button" value="요소표시" onclick="document.getElementById('heading').style.visibility = 'visible'" />
```



DOM(document Object Model) 간단한 예제

JavaScript

```
<p id="p1">This is a paragraph.</p>
<script>
    function changeStyle() {
        document.getElementById("p1").style.color = "red";
        document.getElementById("p1").style.fontFamily = "Century Schoolbook";
        document.getElementById("p1").style.fontStyle = "italic";
    }
</script>
     <input type="button" onclick="changeStyle()" value="눌러보세요" />
```

➢ DOM요소 추가 함수 종류

- 웹 문서 상에 없던 요소를 추가해서 화면에 표시 할 수 있다.

함수이름	설명
createElement(생성할 요소노드 명);	새 요소노드를 생성함
createTextNode(생성할 텍스트노드 값);	텍스트노드를 생성함
appendChild(연결시킬 요소노드나 텍스트노드);	텍스트 노드를 요소노드에 자식노드로 추가함
removeChild(삭제할 노드 명);	객체에 연결된 노드를 삭제함

DOM(document Object Model) 간단한 예제

```
<a href="javascript:add();">보여주기</a>
<a href="javascript:remove();">사라지기</a>

    function add() {
      let tables = document.createElement('table'); //table태그요소노드 생성
      tables.setAttribute('id', 'target'); //table태그요소에 id 속성을 붙여줌(선택사항)
      let trs = document.createElement('tr'); //tr태그요소노드 생성
      let tds = document.createElement('td'); //td태그요소노드 생성
      let textNode = document.createTextNode('저는 현재 공부 중입니다.'); //텍스트노드 생성
      tables.appendChild(trs).appendChild(tds).appendChild(textNode); //table요소노드에 나머지 연결시킴
      //여기까지 코딩한다고 해서는 아무것도 안 나온다.

      document.body.appendChild(tables);
      //기존에 html문서안에서 상주하고 있는 body요소 노드에 tables요소를 연결시켜 줘야한다.
    }
    function remove() {
      let target = document.getElementById('target');
      target.parentNode.removeChild(target);
      //일반적으로 문서객체를 삭제할 때 사용하는 방법
    }
```

DOM(document Object Model) 간단한 예제

JavaScript

보여주기 사라지기

보여주기 사라지기
저는 현재 공부 중입니다.
저는 현재 공부 중입니다.

보여주기 사라지기
저는 현재 공부 중입니다.

보여주기 사라지기
저는 현재 공부 중입니다.
저는 현재 공부 중입니다.

DevTools is now available in Korean!

Always match Chrome's language | Switch DevTools to Korean | Don't show again

Elements | Console | Sources

```
<!DOCTYPE html>
<html lang="kr">
▶ <head> … </head>
▼ <body>
    <a href="javascript:add();">보여주기</a>
    <a href="javascript:remove();">사라지기</a>
  ▶ <script> … </script>
  ▼ <table id="target">
    ▼ <tr>
        <td>저는 현재 공부 중입니다.</td>
      </tr>
    </table>
  ▼ <table id="target">
    ▼ <tr>
        <td>저는 현재 공부 중입니다.</td>
      </tr>
    </table>
  </body>
</html> == $0
```

JavaScript

➢ ClassName

- 해당 요소의 class 값을 문자열로 가져오거나(return) 변경할 수 있는 DOM 프로퍼티이다.
- 새로운 값을 대입하면 기존 class 전체가 교체된다.
- 따라서 클래스 하나만 추가/삭제하는 작업에는 불편하다.

➢ ClassList

- 요소의 class 속성을 토큰(개별 클래스 단위)으로 관리할 수 있게 해주는 DOM 프로퍼티이다.
- 요소의 class 속성을 **토큰 단위 배열처럼 관리**하는 **DOMTokenList 객체**를 반환한다.
- 개별 클래스 조작에 편리하며, 여러 개의 클래스도 동시에 조작 가능하다.

➢ 주요 메서드

- add("클래스명1", "클래스명2") : 클래스 추가
- remove("클래스명") : 클래스 제거
- toggle("클래스명") : 있으면 제거, 없으면 추가
- contains("클래스명") : 클래스 존재 여부 확인

JavaScript

➢ ClassList 주요 메서드

```
// 클래스 전체 읽기
element.classList;  // DOMTokenList 반환 (유사 배열 객체)

// 특정 클래스 가져오기
element.classList.item(index);

// 클래스 추가
element.classList.add('이름1', '이름2');

// 클래스 삭제
element.classList.remove('이름');

// 클래스 존재 확인 (true/false)
element.classList.contains('이름');

// 클래스 토글 (있으면 제거, 없으면 추가)
element.classList.toggle('이름');

// 클래스 교체 (old → new)
element.classList.replace('oldClass', 'newClass');
```

DOM(document Object Model) – className 간단한 예제

JavaScript

```
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>dom 연습문제 - className, classList</title>
    <style>
      .blue{
        color: blue;
      }
    </style>
  </head>
  <body>
    <h1>
      <span>난 span 태그야</span>
    </h1>
    <script>
      let titleChange = document.getElementsByTagName('h1')[0];
      let p = document.createElement('p');

      p.className = 'blue';
      // p.classList.add('blue');
      p.innerHTML = '난 js로 추가된 p태그야';
      titleChange.appendChild(p);//h1태그의 자식 노드로 추가한다.
    </script>
  </body>
```

DOM(document Object Model) – className 간단한 예제 결과 화면

JavaScript

➢ 클래스 토글 간단한 예제

```
<p id="text">클릭하면 강조 표시</p>
<style>
.highlight { color: red; font-weight: bold; }
</style>
<script>
 let text = document.getElementById("text")
   text.onclick = function() {
       this.classList.toggle("highlight");
   }
</script>
```

기초문제

1. id 선택자를 사용해 텍스트를 "DOM 실습 중급"으로 바꾸고, 콘솔에도 출력

HTML

```
<p id="text">안녕하세요</p>
```

2. 클래스 선택자로 모든 div의 배경색을 "yellow"로 변경하세요.

HTML

```
<div class="box">1</div>
<div class="box">2</div>
<div class="box">3</div>
```

3. getElementsByTagName을 사용해 모든 <p> 글자색을 "blue"로 변경하세요.

HTML

```
<p>첫 번째 문장</p>
<p>두 번째 문장</p>
```

4. querySelector를 사용해 첫 번째 <li> 글자를 "포도"로 바꾸세요.

HTML

```
<ul>
  <li>사과</li>
  <li>바나나</li>
  <li>체리</li>
</ul>
```

5. querySelectorAll을 사용해 모든 <li> 글자를 "과일"로 바꾸세요.

HTML

```
<ul>
  <li>사과</li>
  <li>바나나</li>
  <li>체리</li>
</ul>
```

6. 새 <li> 요소를 생성하고 "새 항목" 텍스트를 넣어 에 추가하세요.

HTML

```
<ul id="list"></ul>
```

7. 첫 번째 <li> 요소를 삭제하세요.

HTML

```
<ul>
  <li>삭제될 항목</li>
  <li>남길 항목</li>
</ul>
```

8. innerHTML을 사용해 DOM 연습 을 추가하세요.

HTML

```
<div id="content"></div>
```

9. 버튼 클릭 시 input 값을 콘솔에 출력하세요.

HTML

```
<input type="text" id="userInput" value="안녕">
<button id="btn">확인</button>
```

10. JS로 input 값을 "DOM 실습"으로 변경하세요.

HTML

```
<input type="text" id="userInput">
```

11. JS에서 classList.add("active")를 사용해 클래스 추가하세요.

HTML

```
<div id="box"></div>
```

12. JS에서 classList.remove("active")를 사용해 제거하세요.

HTML

```
<div id="box" class="active"></div>
```

13. 버튼 클릭 시 classList.toggle("active") 기능 구현하세요.

HTML

```
<div id="box" class="active"></div>
```

14. JS에서 배경색을 "red"로 변경하세요.

HTML

```
<div id="box" style="width: 100px; height:
100px;"></div>
```

15. JS로 링크 주소를 "https://google.com"으로 변경하세요.

HTML

```
<a id="link" href="https://naver.com">네이버</a>
```

16. JS로 src 속성 값을 콘솔에 출력하세요.

HTML

```
<img id="logo" src="logo.png">
```

17. 자식 <p> 요소에서 부모 요소를 찾아 배경색을 "pink"로 바꾸세요.

HTML

```
<div class="parent">
  <p>자식 요소</p>
</div>
```

18. children을 사용해 첫 번째 <li> 글자를 "첫 번째"로 바꾸세요.

HTML

```
<ul id="list">
  <li>1</li>
  <li>2</li>
</ul>
```

19. 두 번째 요소 기준으로 이전 형제 요소의 글자를 "이전"으로 바꾸세요.

HTML

```
<p>첫 번째</p>
<p>두 번째</p>
<p>세 번째</p>
```

20. JS에서 "0"이라는 <li> 요소를 첫 번째 자리에 삽입하세요.

HTML

```
<ul id="list">
  <li>1</li>
  <li>2</li>
</ul>
```

연습문제

DOM 연습문제 1

예제1] 아래 주어진 조건에 만족하도록 JavaScript 문서를 작성하시오.

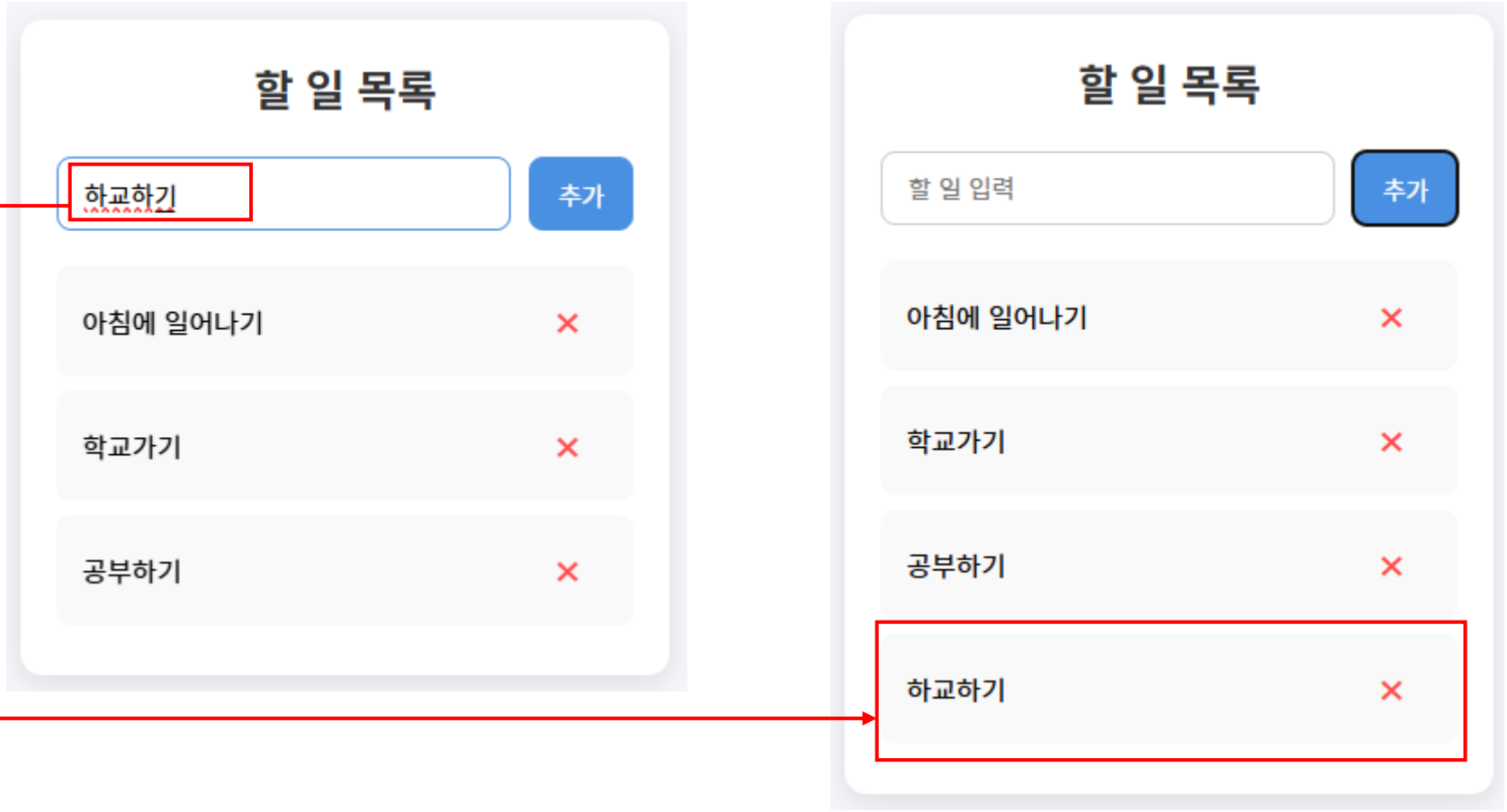
조건

① 아래 주어진 HTML을 이용해 Todo List를 작성하시오.
② CSS는 <출력 결과물>보고 알아서 작성하시오.
③ DOM 요소 추가 함수 (createElement, createTextNode, appendChild, removeChild) 이용해 적상하시오.
④ 다음 페이지의 <출력 결과물>처럼 출력되도록 작성하시오.
⑤ 저장할 파일명은 Ex_todo.html 로 작성하시오.

HTML

```
<div class="todo-container">
    <h1>할 일 목록</h1>
    <div class="input-area">
      <input id="task" type="text" placeholder="할 일 입력" />
      <button onclick="addTask()">추가</button>
    </div>
    <ul id="tasks"></ul>
</div>
```

<출력 결과>


할 일 목록
하교하기
추가
아침에 일어나기
학교가기
공부하기
할 일 목록
할 일 입력
추가
아침에 일어나기
학교가기
공부하기
하교하기

DOM 연습문제 3

예제3] 아래 주어진 조건에 만족하도록 html 문서를 작성하시오.

조건

① 모달(modal)창을 작성하시오.

② [뉴스레터 구독하기]버튼을 클릭하면 모달창이 화면에 출력되도록 작성하시오.

③ Office.jpg 파일을 이용하여 작성하시오.

④ CSS는 <출력 결과물>보고 알아서 작성하시오.

⑤ [X]버튼을 클릭하면 모달창이 화면에서 사라지도록 작성하시오.

⑥ 'active'라는 클래스명을 classList.add( )를 이용하여 추가하여 작성하시오.

⑦ classList.remove( ) 메서드를 이용하여 작성하시오.

⑧ 주어진 <html>파일을 이용하여 작성하시오.

⑨ <출력 결과물>처럼 출력되도록 작성하시오.

⑩ 저장할 파일명은 Ex_Modal.html 로 작성하시오.

DOM 연습문제 3

특수문자 싸이트 https://html.spec.whatwg.org/multipage/named-characters.html

HTML

```
<button class="btn-open">뉴스레터 구독하기</button>

    <div class="modal">
      <div class="modal-conetent">
        <div class="photo"></div>
        <div class="description">
          <div class="desc-header">
            <h2>지금 다양한 혜택을 받아보세요.</h2>
            <button class="btn-close">&times;</button>
          </div>
          <div class="desc-content">
            <input type="email" placeholder="이메일 주소를 입력하세요" />
            <button type="button">뉴스레터 구독하기</button>
            <p>
              스타트업메이트 뉴스룸의 다양한 소식과 혜택을 이메일로 받아
              보시려면 구독 신청 해주시기 바랍니다. 스타트업메이트에 대해 알고
              싶은 뉴스, 꼭 알아야 할 뉴스를 신속하고 정확하게 전합니다.
            </p>
          </div>
        </div>
      </div>
    </div>
    <div class="overlay"></div>
```

DOM 연습문제 3 출력 결과물