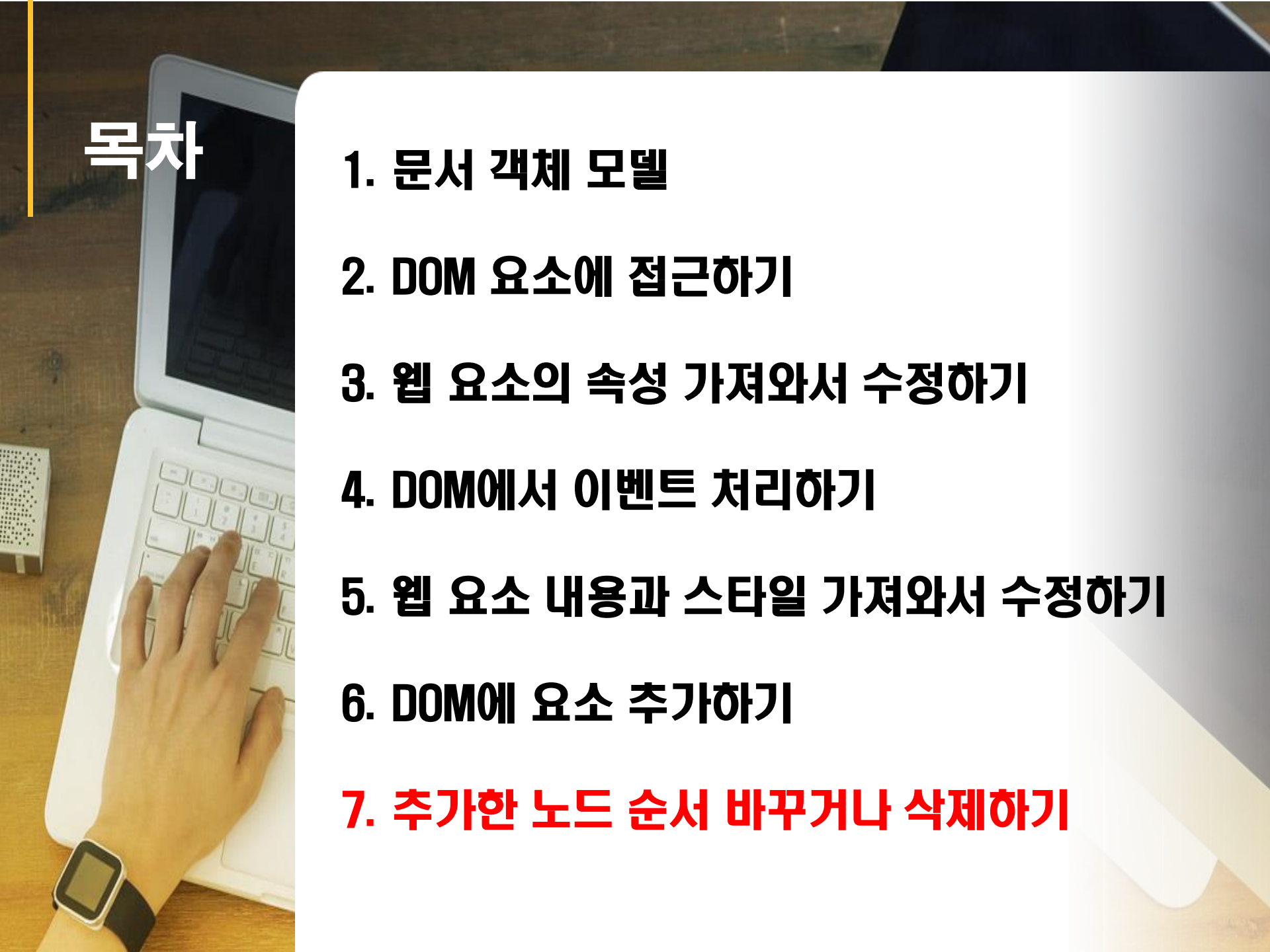
KOOKMIN UNIVERSITY

문서 객체 모델(DOM)

소프트웨어학부
진혜진



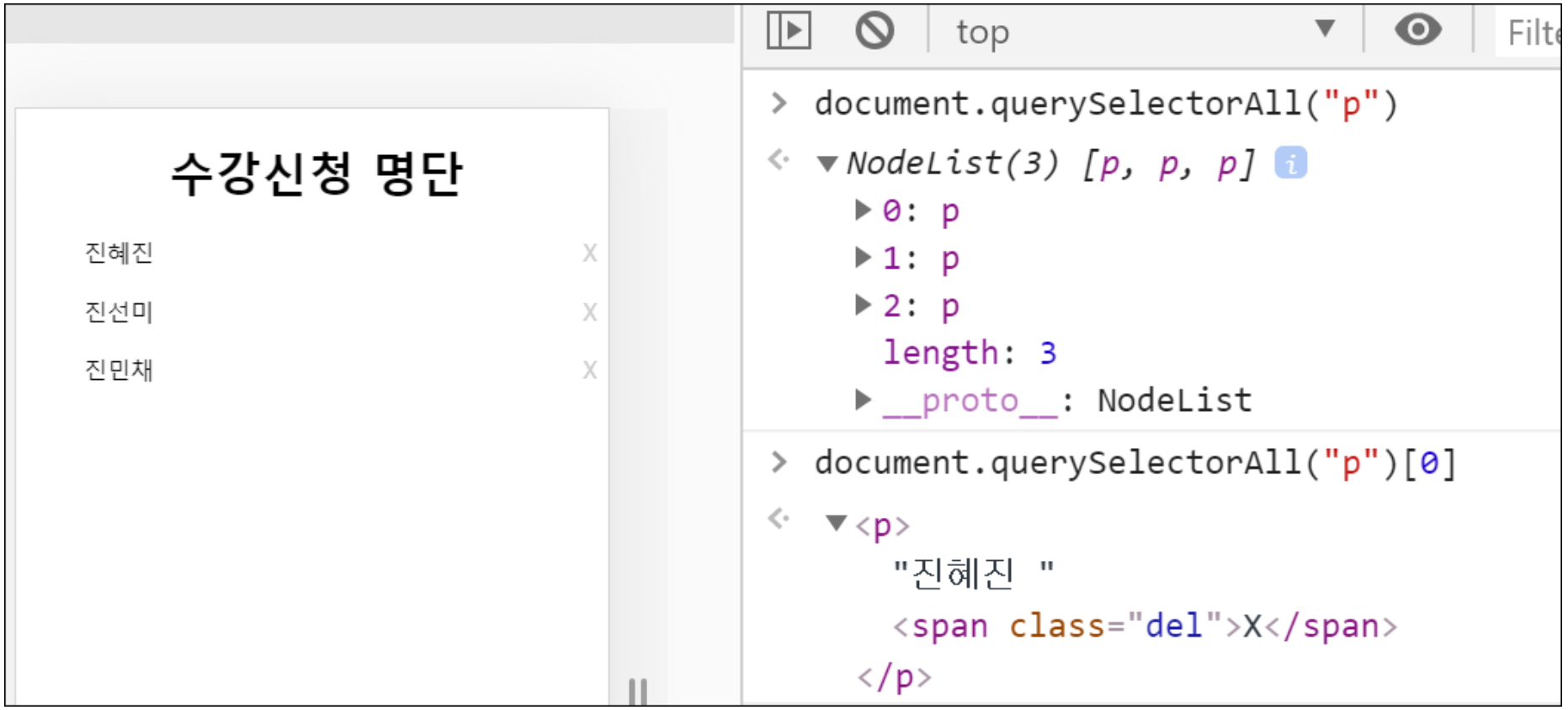
목차

7. 추가한 노드 순서 바꾸거나 삭제하기

- 여러 노드를 한꺼번에 저장하는 노드 리스트

```
<body>
    <div id="container">
            <h1>수강신청 명단</h1>
            <div id="nameList">
                <p>진혜진 <span class="del">X</span></p>
                <p>진선미 <span class="del">X</span></p>
                <p>진민채 <span class="del">X</span></p>
            </div>
    </div>
</body>
```

7. 추가한 노드 순서 바꾸거나 삭제하기



7. 추가한 노드 순서 바꾸거나 삭제하기

- DOM 트리를 활용해 원하는 노드 다루기
  - 자식 노드 확인하기
    - hasChildNodes( ) 함수
    - 특정 노드에 자식 노드가 있는지 확인하는 함수
    - 자식 노드가 있다면 true, 없다면 false 반환

```
> document.querySelectorAll("p")[0].hasChildNodes()
< true
```

7. 추가한 노드 순서 바꾸거나 삭제하기


```
<div id="nameList">
    <p>진혜진 <span class="del">X</span></p>
    <p>진선미 <span class="del">X</span></p>
    <p>진민채 <span class="del">X</span></p>
</div>
```

- 자식 노드에 접근하기
  - childNodes 속성
  - 현재 노드의 자식 노드를 가지고 있는 속성
  - 요소 노드 뿐만 아니라 빈 텍스트 노드도 자식 노드로 인식함

```
> document.querySelector("#nameList").childNodes
◂ ▼ NodeList(7) [text, p, text, p, text, p, text]
    ▶ 0: text
    ▶ 1: p
    ▶ 2: text
    ▶ 3: p
    ▶ 4: text
    ▶ 5: p
    ▶ 6: text
      length: 7
    ▶ __proto__: NodeList
```

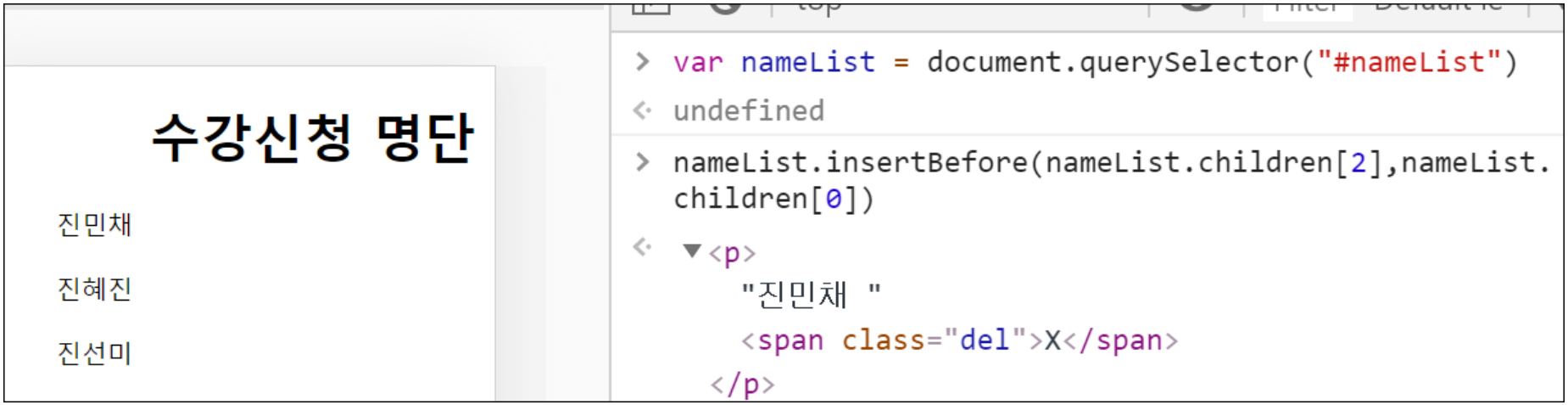
7. 추가한 노드 순서 바꾸거나 삭제하기

- 요소에만 접근하려면 children 속성 사용
  - children 속성
  - 현재 노드의 자식 노드 중 요소 노드만 가지고 있는 속성

```
> document.querySelector("#nameList").children
HTMLCollection(3) [p, p, p]
    0: p
    1: p
    2: p
    length: 3
    __proto__: HTMLCollection
```

7. 추가한 노드 순서 바꾸거나 삭제하기

- 원하는 위치에 노드 삽입하기
  - insertBefore(인수1, 인수2) 함수
    - 첫 번째 인수는 추가하는 노드
    - 두 번째 인수는 기준이 되는 노드
  - 부모 노드에 자식 노드를 추가할 때 기준 노드를 지정하고 그 앞에 자식 노드 추가 할 수 있음



7. 추가한 노드 순서 바꾸거나 삭제하기


```
<div id="nameList">
    <p>진혜진 <span class="del">X</span></p>
    <p>진선미 <span class="del">X</span></p>
    <p>진민채 <span class="del">X</span></p>
</div>
```

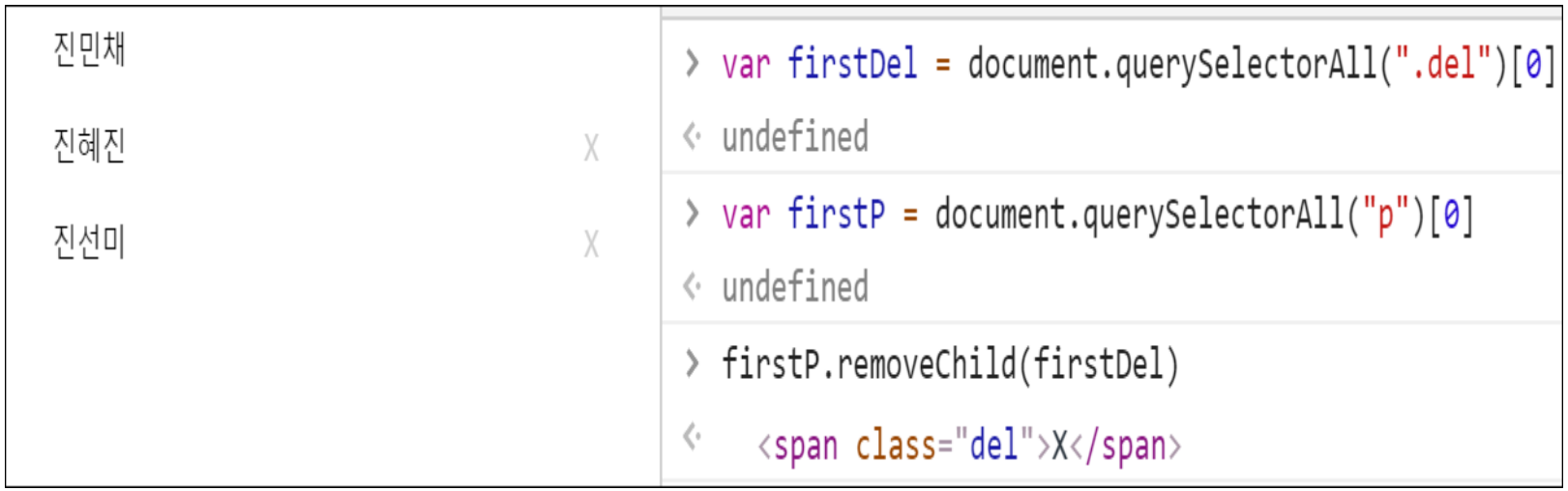
특정 노드 삭제하기

removeChild( ) 함수와 parentNode 속성

- 노드는 스스로 자신을 삭제할 수 없기 때문에 부모 노드에 접근한 후 부모 노드에서 삭제해야 함
- parentNode 속성
  - 현재 노드의 부모 노드 정보를 가지고 있는 속성
- removeChild( ) 함수
  - 부모 노드에서 자식 노드를 삭제하는 함수

```
> document.querySelectorAll(".del")[0].parentNode
<- ▼<p>
     "진민채 "
     <span class="del">X</span>
   </p>
```

7. 추가한 노드 순서 바꾸거나 삭제하기




```
▼<div id="container">
   <h1>수강신청 명단</h1>
  ▼<div id="nameList">
     <p>진민채 </p> == $0
    ▼<p>
       "진혜진 "
       <span class="del">X</span>
     </p>
```

7. 추가한 노드 순서 바꾸거나 삭제하기

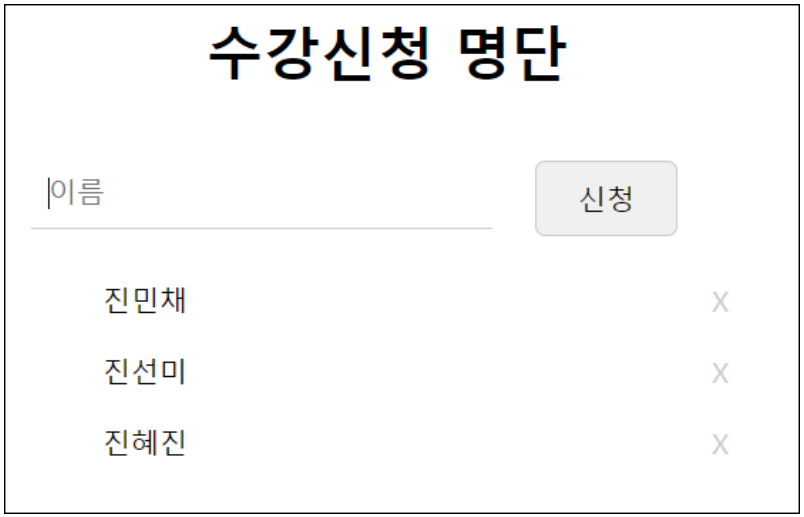
□ 맨위에 이름 추가하기

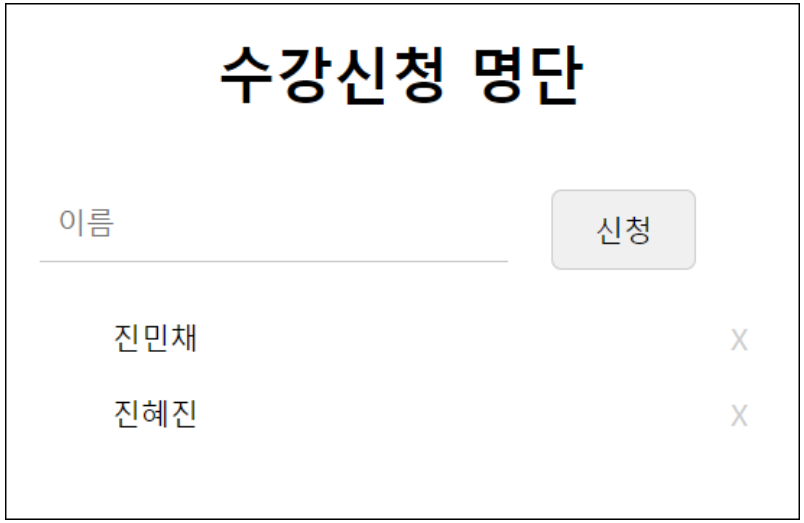
```
function newRegister() {
    var newP = document.createElement("p");  // 새 p 요소 만들기
    var userName = document.querySelector("#userName");
    var newText = document.createTextNode(userName.value);  // 새 텍스트 노드 만들기
    newP.appendChild(newText);  // 텍스트 노드를 p 요소의 자식 요소로 연결하기

  var nameList = document.querySelector("#nameList");
  nameList.insertBefore(newP, nameList.childNodes[0]);  // p 요소를 #nameList 맨 앞에 추가하기
    // nameList.appendChild(newP); // p 요소를 #nameList의 자식 요소로 만들기
    userName.value = "";  // 텍스트 필드 지우기
}
```

7. 추가한 노드 순서 바꾸거나 삭제하기

7. 추가한 노드 순서 바꾸거나 삭제하기

이름 삭제하기





```
function newRegister() {
  var newP = document.createElement("p");  // 새 p 요소 만들기
  var userName = document.querySelector("#userName");
  var newText = document.createTextNode(userName.value);  // 새 텍스트 노드 만들기
  newP.appendChild(newText);  // 텍스트 노드를 p 요소의 자식 요소로 연결하기

  var delBttn = document.createElement("span");   // 새 button 요소 만들기
  var delText = document.createTextNode("X");  // 새 텍스트 노드 만들기
  delBttn.setAttribute("class", "del");  // 버튼에 class 속성 설정하기
  delBttn.appendChild(delText);  // 텍스트 노드를 button 요소의 자식 요소로 연결하기
  newP.appendChild(delBttn);  //  del 버튼을 p 요소의 자식 요소로

  var nameList = document.querySelector("#nameList");
  nameList.insertBefore(newP, nameList.childNodes[0]);  // p 요소를 #nameList 맨 앞에 추가하기
  // nameList.appendChild(newP); // p 요소를 #nameList의 자식 요소로 만들기
  userName.value = "";  // 텍스트 필드 지우기

  var removeBttns = document.querySelectorAll(".del");

  for (var i=0; i<removeBttns.length; i++) {  // removeBttns에 있는 요소 전체를 반복
    removeBttns[i].addEventListener("click", function() {  // i번째 버튼을 클릭했을 때 실행할 함수 선언
      if (this.parentNode.parentNode)  // 현재 노드(this)의 부모 노드의 부모 노드가 있을 경우 실행
        this.parentNode.parentNode.removeChild(this.parentNode);
        // '현재 노드(this)의 부모 노드의 부모 노드'를 찾아 '현재 노드(this)의 부모 노드(p 노드)' 삭제
    });
  }
}
```