

08. 문서 객체 모델

문서 객체 모델(Document Object Model, DOM)

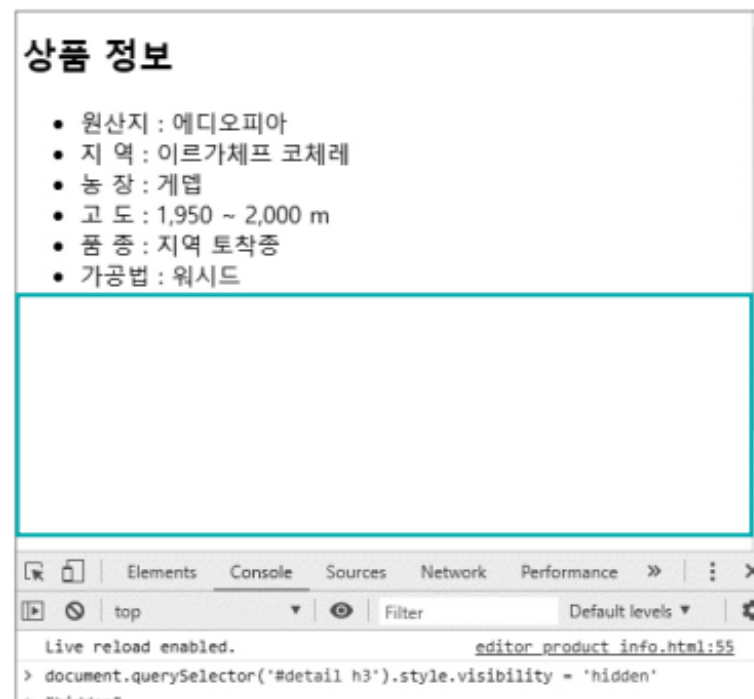
자바스크립트가 웹 문서를 동적으로 다룰 수 있도록 문서에 있는 각 요소를 객체 형태로 처리하는 것

실습 파일 : 08Wdom.html

예) 문서에서 특정 부분을 가져와서 visibility 속성 값을 바꾸면?

→ 화면에서 사라짐

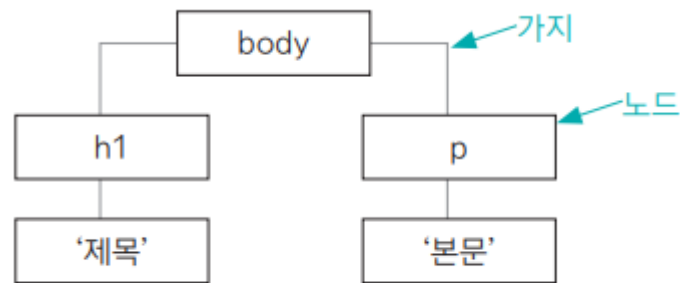
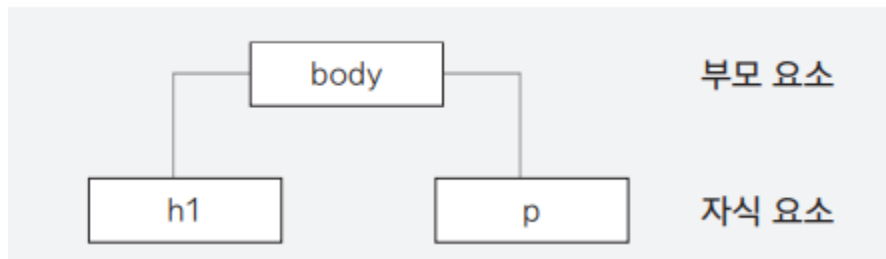
```
> document.querySelector('#detail p').style.visibility = 'hidden'
```



DOM 트리

- DOM은 웹 문서의 요소를 부모 요소와 자식 요소로 구분
- 웹 문서 구조를 부모/자식 관계로 표시하면 나무를 거꾸로 뒤집어 놓은 모습 → DOM 트리

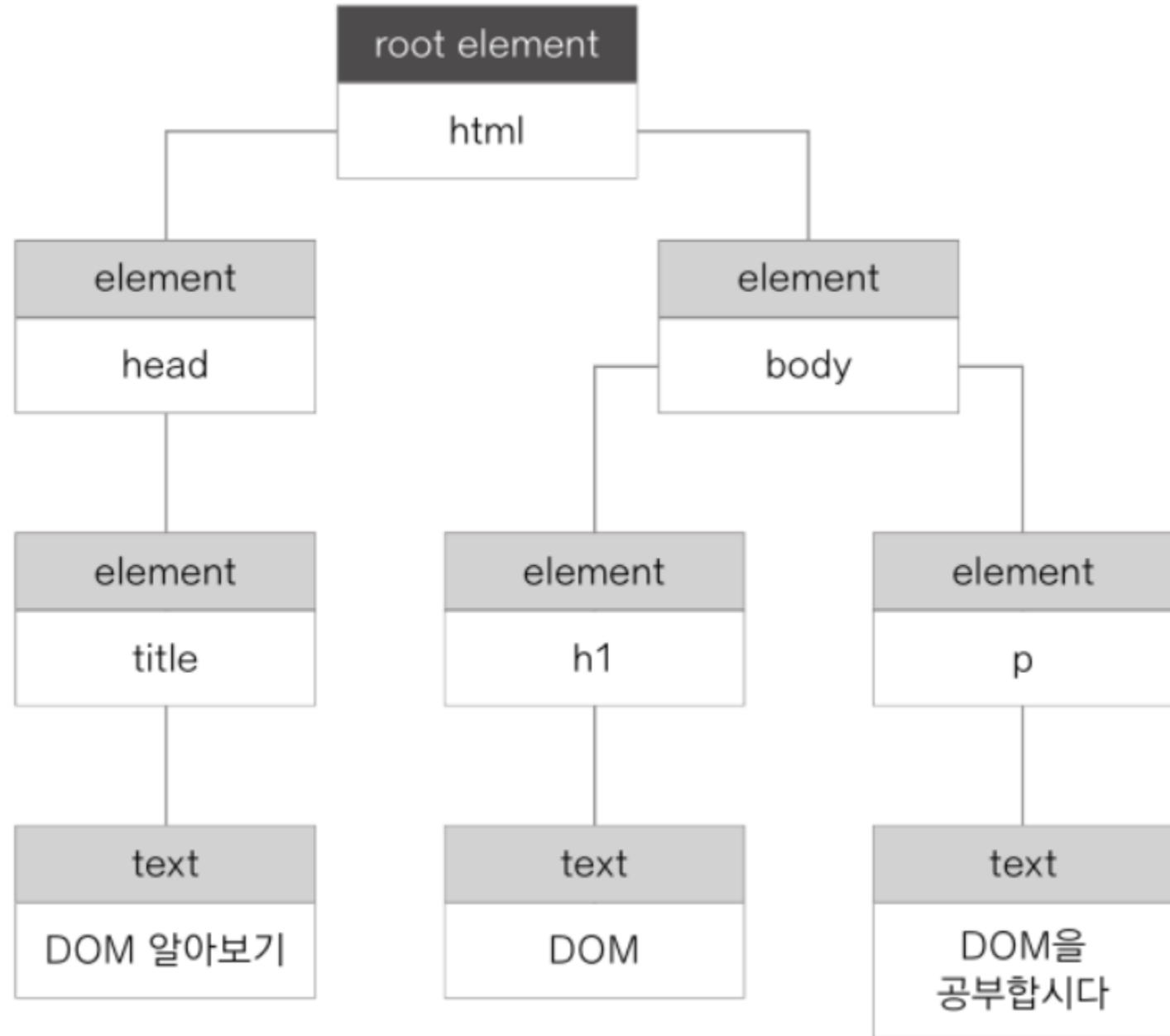
```
<body>  
  <h1>제목</h1>  
  <p>본문</p>  
</body>
```



DOM 트리의 노드

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

live dom viewer



getElementById() – id 선택자를 사용해 접근하기

```
10 <body>
11   <div id="container">
12     <h1 id="heading">에디오피아 게덱</h1>
13     <div id="prod-img">
```

```
> document.getElementById("heading")
< <h1 id="heading">에디오피아 게덱</h1>
```

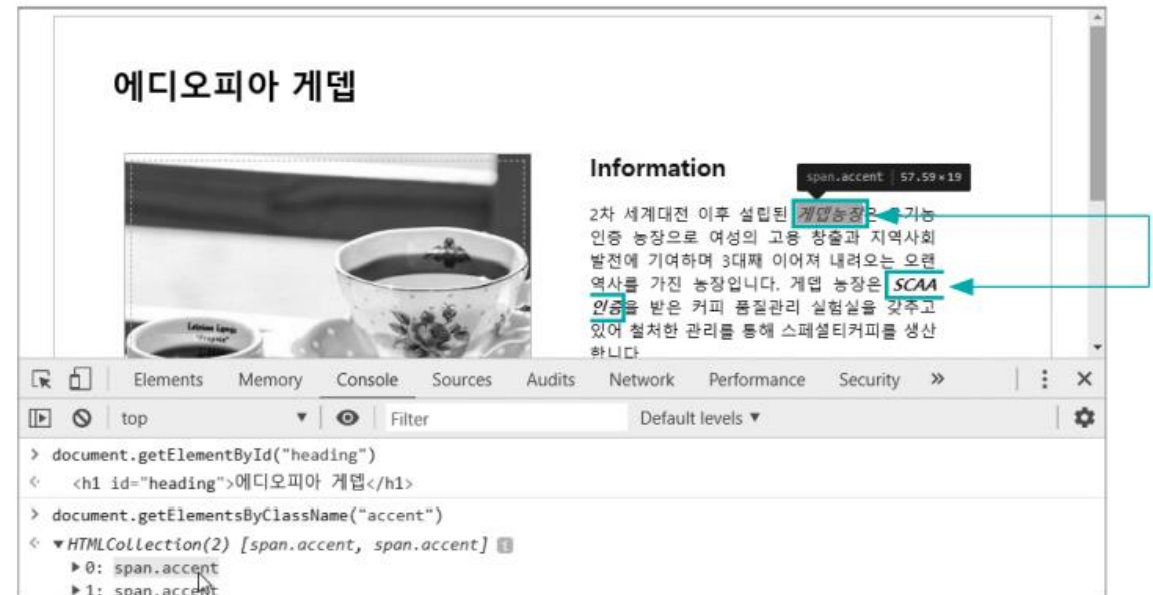


getElementsClassName() – class 선택자를 사용해 접근하기

18 <p>2차세계대전 이후 설립된 게덱농장은 ... SCAA 인증을 받은 ... 스페셜티커피를 생산합니다.</p>

```
> document.getElementsByClassName("accent")  
< ▶ HTMLCollection(2) [span.accent, span.accent]
```

```
> document.getElementsByClassName("accent")[0]  
< <span class="accent">게덱농장</span>
```



getElementsByTagName() – 태그 이름을 사용해 접근하기

```
16 <div id="desc">
17   <h2>Information</h2>
18   <p>2차 세계대전 이후 ..... 생산합니다.</p>
19   <h2>Flavor Note</h2>
20   <p>은은하고 ..... 향이 입안 가득.</p>
21 </div>
```

```
> document.getElementsByTagName("h2")
< ▶ HTMLCollection(2) [h2, h2]
  ▶ 0: h2
  ▶ 1: h2
  length: 2
  ▶ __proto__: HTMLCollection
```



querySelector(), querySelectorAll()

```
document.getElementById("container")
```

```
document.querySelector("#container")
```

id 값은 # 기호와 함께
class 값은 . 기호와 함께
태그 이름은 기호 없음

```
> document.querySelector("#heading")  
< <h1 id="heading">에디오피아 게덱</h1>
```

```
> document.querySelectorAll(".accent")  
< ▶ NodeList(2) [span.accent, span.accent]
```

[미리보기] 선택한 상품 이미지 표시하기

에디오피아 게덱



상품명 : 에디오피아 게덱
판매가 : 9,000원
배송비 : 3,000원
(50,000원 이상 구매시 무료)
적립금 : 180원(2%)
로스팅 : 2019.06.17

장바구니 담기

상세 설명 보기



에디오피아 게덱



상품명 : 에디오피아 게덱
판매가 : 9,000원
배송비 : 3,000원
(50,000원 이상 구매시 무료)
적립금 : 180원(2%)
로스팅 : 2019.06.17

장바구니 담기

상세 설명 보기



태그 속성에 접근하기 및 수정하기

getAttribute() – 태그의 속성에 접근해서 값을 가져옴

setAttribute() – 태그의 속성 값을 지정함

속성값 가져오기

```
13 <div id="prod-img">  
14     
15 </div>
```

```
> document.querySelector("#prod-img > img").getAttribute("src")  
< "images/coffee-pink.jpg"
```

속성값 수정하기

```
> document.querySelector("#prod-img > img").setAttribute("src", "images/coffee-blue.jpg")
```

선택한 상품 이미지 표시하기

```
1  var bigPic = document.querySelector("#cup");
2  var smallPics = document.querySelectorAll(".small");
3
4  for(var i=0; i<smallPics.length; i++) {  //노드 리스트의 각 요소에 접근
5      smallPics[i].onclick = showBig;  //요소를 누르면 showBig( ) 함수 실행
6  }
7
8  function showBig( ) {
9      var newPic = this.src;  //click 이벤트가 발생한 대상의 src 속성 값 가져옴
10     bigPic.setAttribute("src", newPic);  //newPic 값을 큰 이미지의 src 속성에 할당
11 }
```

DOM에서 이벤트 처리하기

HTML 태그 안에서 이벤트 처리기 연결하기

```
11 <div id="container">
12   . //누르면 change-
    Pic( ) 함수 실행
13 </div>
14 <script>
15   var pic = document.querySelector('#pic');
16   function changePic( ) {
17     pic.src = "images/boy.png";
18   }
19 </script>
```

DOM에서 이벤트 처리하기

DOM 요소에 이벤트 처리기 연결하기

```
11 <div id="container">
12   
13 </div>
14 <script>
15   var pic = document.querySelector('#pic');
16   pic.onclick = changePic; //pic 요소를 누르면 changePic() 함수 실행
17
18   function changePic( ) {
19     pic.src = "images/boy.png";
20   }
21 </script>
```

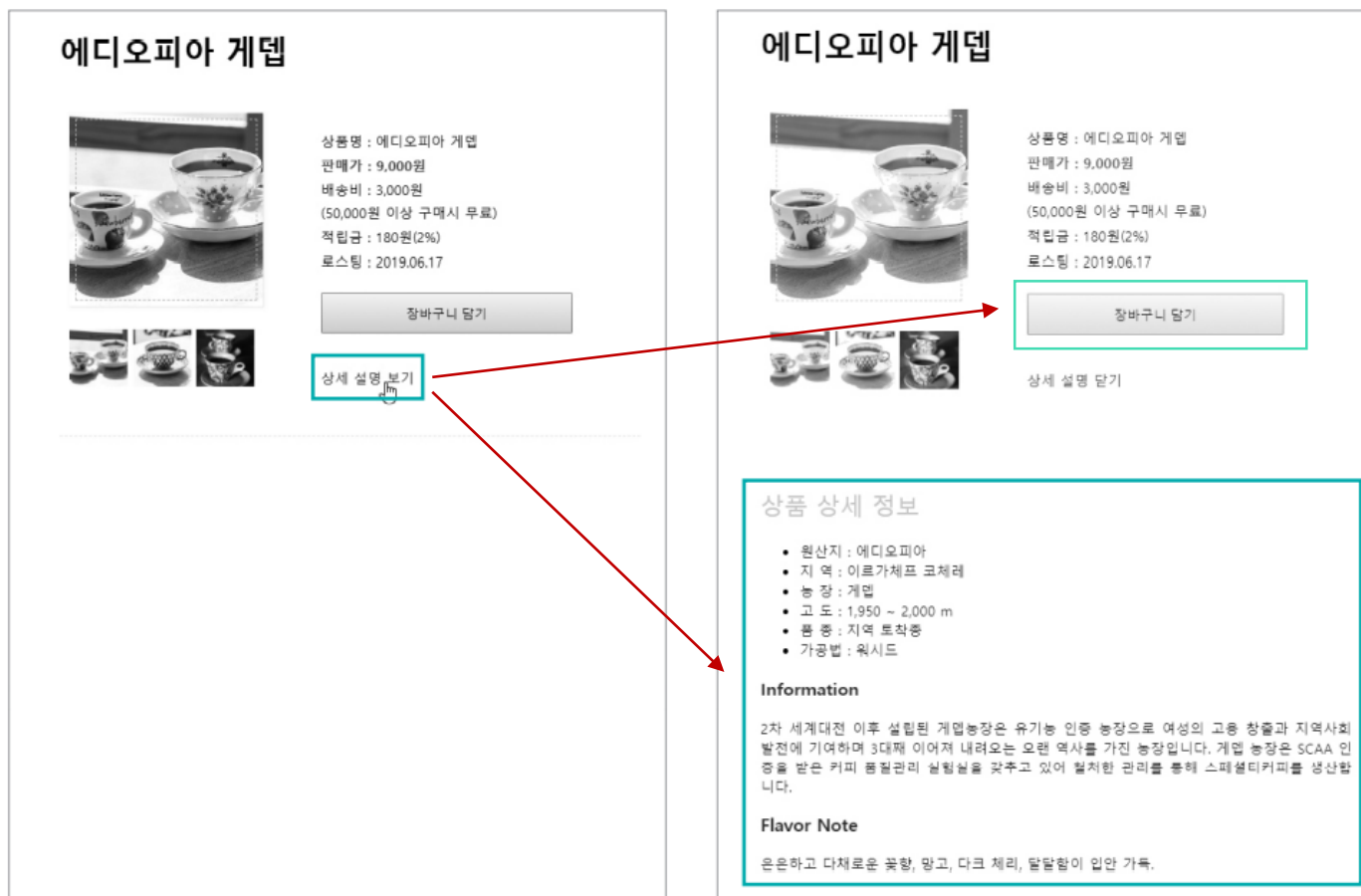
DOM에서 이벤트 처리하기

addEventListener() 사용하기

```
1  var pic = document.querySelector('#pic');
2  pic.addEventListener("mouseover", changePic, false);
3
4  function changePic( ) {
5      pic.src = "images/boy.png";
6  }
7  function originPic( ) {
8      pic.src = "images/girl.png";
9  }
```

- ① 이벤트 유형 : 처리할 이벤트 유형 지정.
"on"을 붙이지 않는다.
- ② 함수 : 이벤트가 발생했을 때 실행할 명령
또는 함수
- ③ 캡처 여부 : 이벤트를 캡처링할지 여부.
기본 값 false

[미리보기] 상세 설명 보기/닫기 버튼



버튼 클릭하면

- 화면에 없던 내용이 나타남
- 버튼 내용이 바뀜

웹 요소의 스타일 가져와서 수정하기

DOM으로 CSS 속성에 접근하고 수정하기

- style 속성 다음에 속성 이름 작성
- 텍스트 스타일 바꾸려면 (요소).style.color

```
> document.querySelector("#heading").style.color = "white"  
< "white"
```

- background-color처럼 두 단어로 된 속성은 backgroundColor처럼 사용

```
> document.querySelector("#heading").style.backgroundColor = "gray"  
< "gray"
```

웹 요소의 스타일 가져와서 수정하기

웹 요소를 화면에 표시하기 / 감추기 - CSS display 속성 사용

```
12  var view = document.querySelector("#view"); // #view 요소를 가져옴
13  view.addEventListener("click", function() { // #view를 눌렀을 때 실행할 함수 선언
14      if (isOpen == false) { // 상세 정보가 감춰져 있다면
15          document.querySelector("#detail").style.display = "block"; // 상세 정보를 화면에 표시
16          view.innerText = "상세 설명 닫기"; // 링크 내용 수정
17          isOpen = true; // 표시 상태로 지정
18      }
19      else { // 상세 정보가 표시되어 있다면
20          document.querySelector("#detail").style.display = "none"; // 상세 정보를 화면에서 감춤
21          view.innerText = "상세 설명 보기"; // 링크 내용 수정
22          isOpen = false; // 감춰진 상태로 지정
23      }
24  });
```

[미리보기] 참가 신청 명단 표시하기

참가 신청

홍길동

신청

참가 신청

이름

홍길동

신청

DOM 트리와 노드

- 모든 HTML 태그는 '요소(Element) 노드'로 표현합니다.
- HTML 태그에서 사용하는 텍스트 내용은 '텍스트(Text) 노드'로 표현합니다.
- HTML 태그에 있는 속성은 모두 '속성(Attribute) 노드'로 표현합니다.
- 주석은 '주석(Comment) 노드'로 표현합니다.

`<p class="accent">주문이 완료되었습니다.</p>`

속성 노드

텍스트 노드

p 요소 노드

DOM에 요소 추가하기



옵션 선택 전



선택 내용 표시

DOM에 요소 추가하기

새로운 노드를 만들거나 부모 노드에 연결할 때

	함수	설명
1	<code>createElement()</code>	새 요소 노드를 만듭니다.
2-1	<code>createTextNode()</code>	텍스트 내용이 있을 경우 텍스트 노드를 만듭니다.
	<code>appendChild()</code>	텍스트 노드를 요소 노드에 자식 노드로 추가합니다.
2-2	<code>createAttribute()</code>	요소에 속성이 있을 경우 속성 노드를 만듭니다.
	<code>setAttributeNode()</code>	속성 노드를 요소 노드에 연결합니다.
3	<code>appendChild()</code>	새로 만든 요소 노드를 부모 노드에 추가합니다.

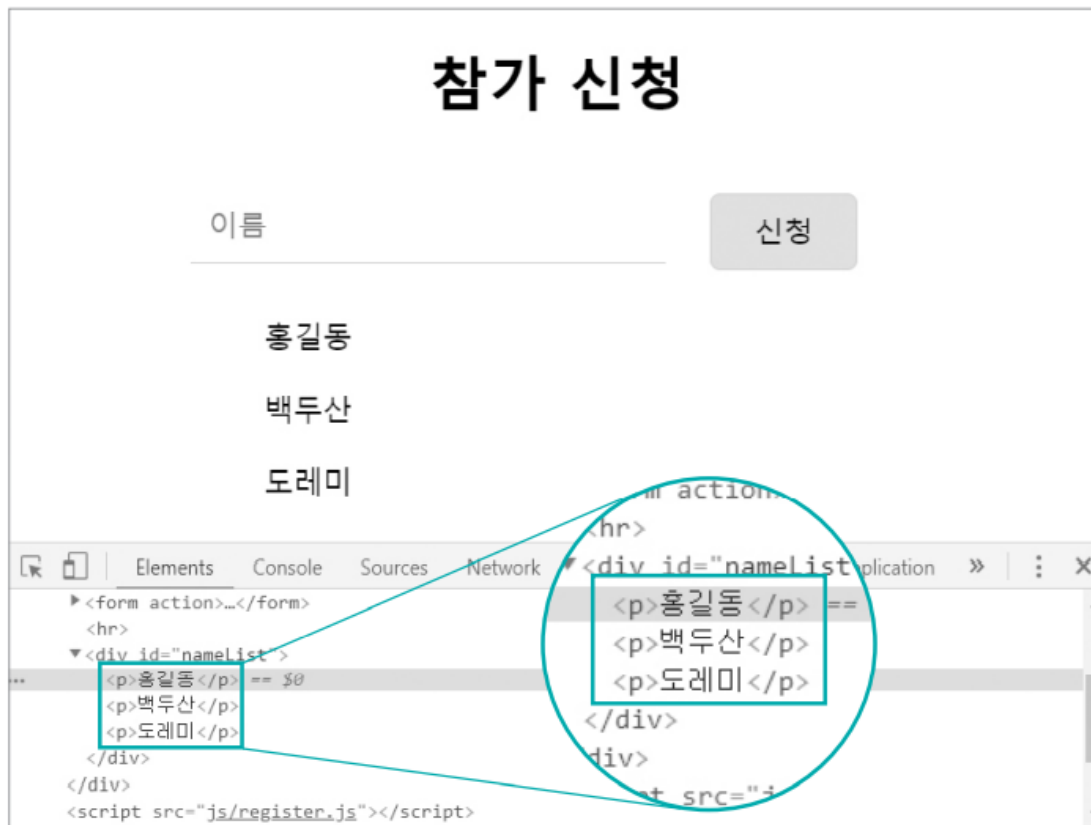
버튼에 이벤트 처리 함수 연결

```
10 <body>
11   <div id = "container">
12     <h1>참가 신청</h1>
13     <form action = "">
14       <input type = "text" id = "userName" placeholder = "이름" required>
15       <button onclick = "newRegister( );return false;">신청</button>
16     </form>
17     <hr>
18     <div id = "nameList"></div> //신청 명단이 표시될 영역
19   </div>
20 </body>
```

노드 추가 & 표시

```
1  function newRegister() {  
2      var newP = document.createElement("p"); // 새 p 요소 만들기  
3      var userName = document.querySelector("#userName");  
4      var newText = document.createTextNode(userName.value); // 새 텍스트 노드 만들기  
5      newP.appendChild(newText); // 텍스트 노드를 p 요소의 자식 요소로 연결하기  
6  
7      var nameList = document.querySelector("#nameList");  
8      nameList.appendChild(newP); // p 요소를 #nameList의 자식 요소로 만들기  
9      userName.value = ""; // 텍스트 필드 지우기  
10 }
```


노드 추가 & 표시



노드 순서 바꾸기 & 삭제

hasChildNodes() 함수

특정 노드에 자식 노드가 있는지 확인하는 함수. 자식 노드가 있다면 true, 없다면 false 반환

childNodes 속성

현재 노드의 자식 노드를 가지고 있는 속성. 요소 노드 뿐만 아니라 빈 텍스트 노드도 자식으로 인식.

children 속성

현재 노드의 자식 노드 중 요소 노드만 가지고 있는 속성

노드 순서 바꾸기 & 삭제

insertBefore() 함수

부모 노드에 자식 노드를 추가할 때 기준 노드를 지정하고 그 앞에 자식 노드 추가

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

노드 순서 바꾸기 & 삭제

```
13  var nameList = document.querySelector("#nameList");
14  nameList.insertBefore(newP, nameList.childNodes[0]); // p 요소를 #nameList 맨 앞에 추가하기
15  // nameList.appendChild(newP); // p 요소를 #nameList의 자식 요소로 만들기
16  userName.value = ""; // 텍스트 필드 지우기
17
18  var removeBtns = document.querySelectorAll(".del");
19
20  for (var i=0; i<removeBtns.length; i++) { // removeBtns에 있는 요소 전체를 반복
21      removeBtns[i].addEventListener("click", function() { // i번째 버튼을 클릭했을 때 실행할 함수 선언
22          if (this.parentNode.parentNode) // 현재 노드(this)의 부모 노드의 부모 노드가 있을 경우 실행
23              this.parentNode.parentNode.removeChild(this.parentNode);
24          // '현재 노드(this)의 부모 노드의 부모 노드'를 찾아 '현재 노드(this)의 부모 노드(p 노드)' 삭제
25      });
26  }
27 }
```