

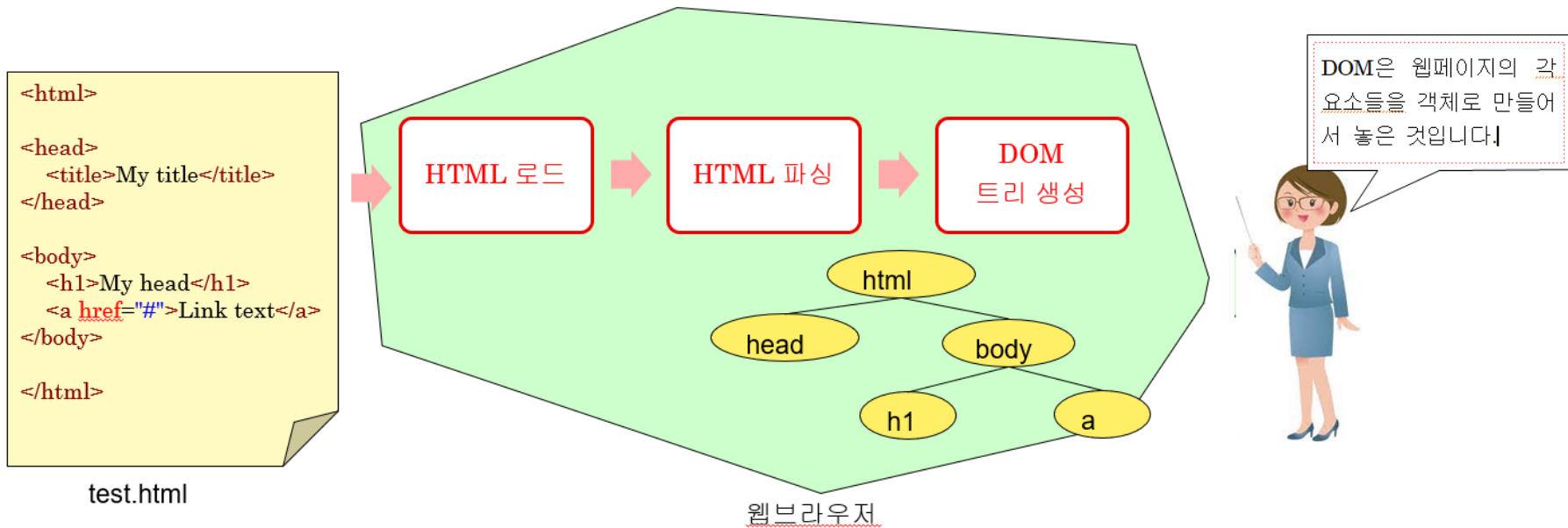
학습 목표

- **DOM(Document Object Model)**을 설명할 수 있나요?
- 웹페이지에서 **HTML** 요소의 내용을 자바스크립트 코드로 변경할 수 있나요?
- 웹페이지에서 **HTML** 요소의 스타일을 자바스크립트 코드로 변경할 수 있나요?
- 웹페이지에 새로운 요소를 추가할 수 있나요?
- 웹페이지의 요소를 삭제할 수 있나요?



문서 객체 모델(DOM)

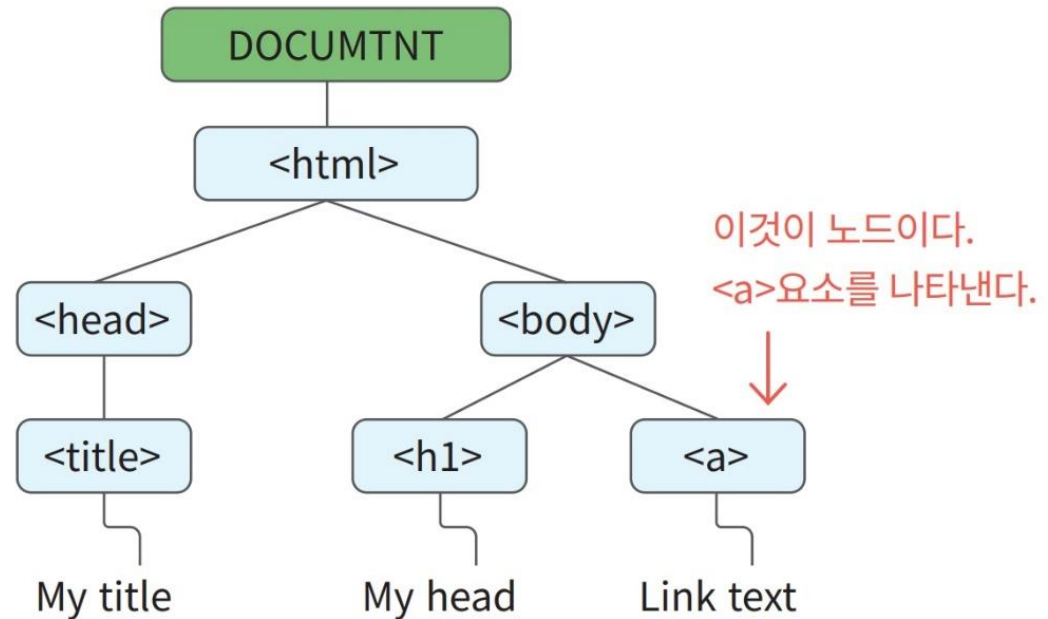
- **DOM(Document Object Model)**은 **HTML** 문서의 계층적인 구조를 트리(tree)로 표현
- 자바스크립트가 간편하게 **HTML** 문서에 접근하여 수정할 수 있도록, **HTML** 요소를 객체로 만들어 놓는 것이다.



HTML→DOM

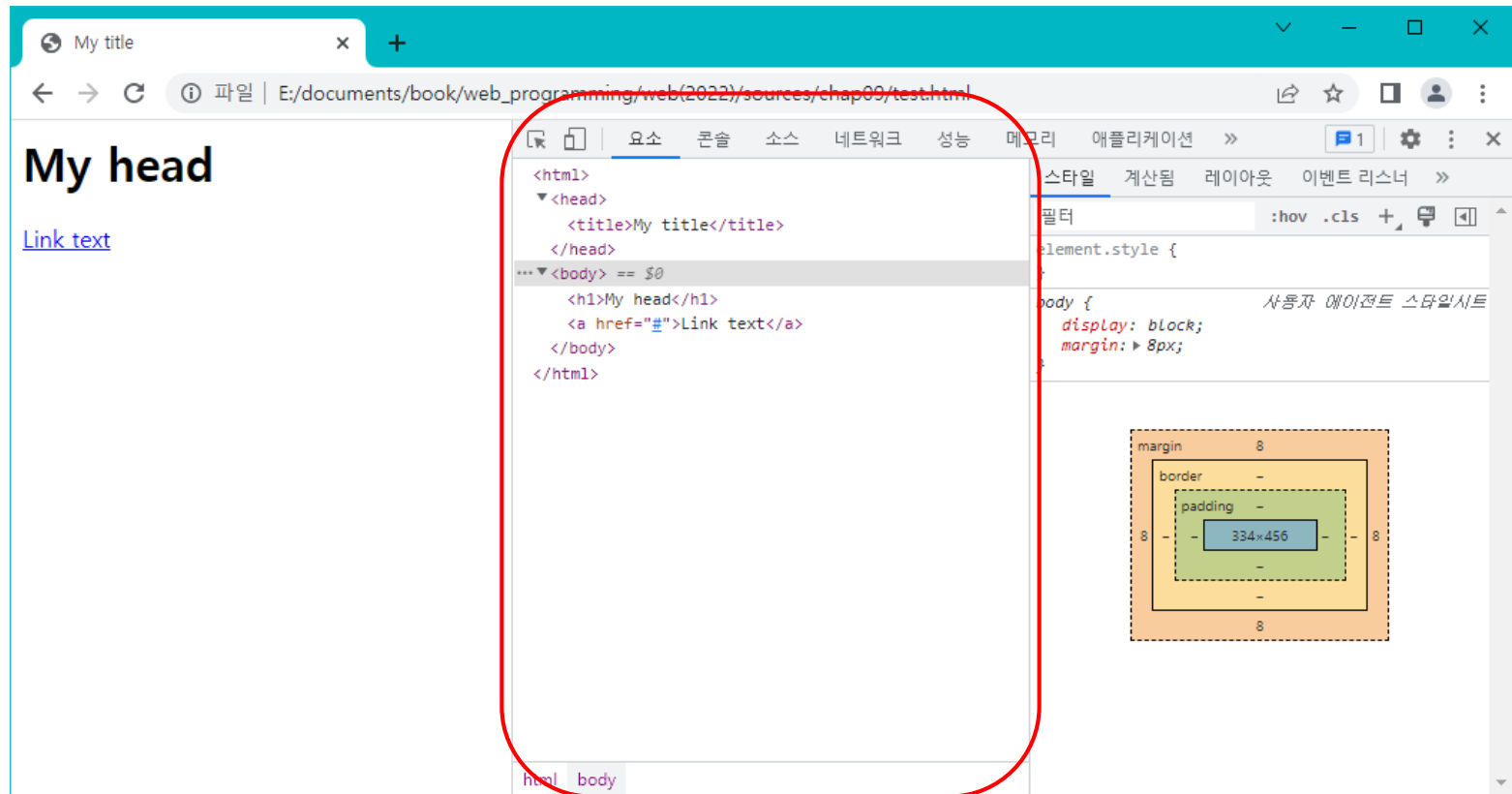
- 웹 페이지가 로드되면, 브라우저는 **DOM**을 생성한 후에 이 **DOM**을 순회하면서 웹페이지를 화면에 표시한다.

```
<html>
  <head>
    <title>My title</title>
  </head>
  <body>
    <h1>My head</h1>
    <a href="#" >Link text</a>
  </body>
</html>
```



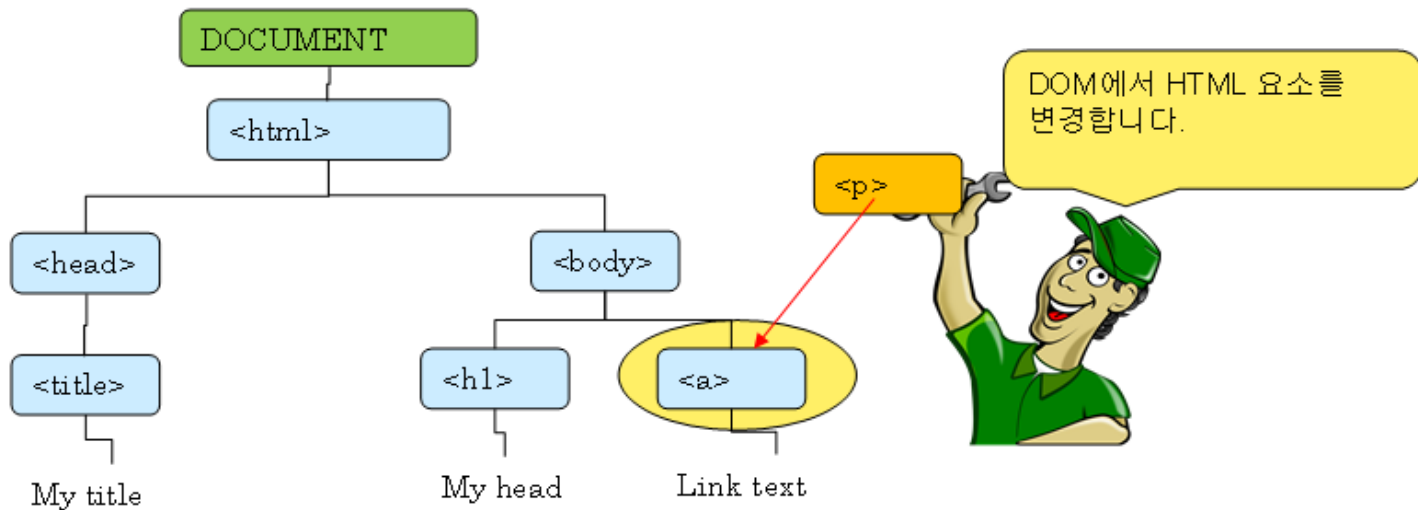
웹페이지의 DOM 트리 확인하는 방법

- 웹페이지의 **DOM** 트리는 어디서 볼 수 있을까? 크롬 브라우저에서 마우스 오른쪽 버튼을 누르고 **[검사]**를 선택하면 다음과 같은 화면이 등장한다



DOM을 이용하여 무엇을 할 수 있을까?

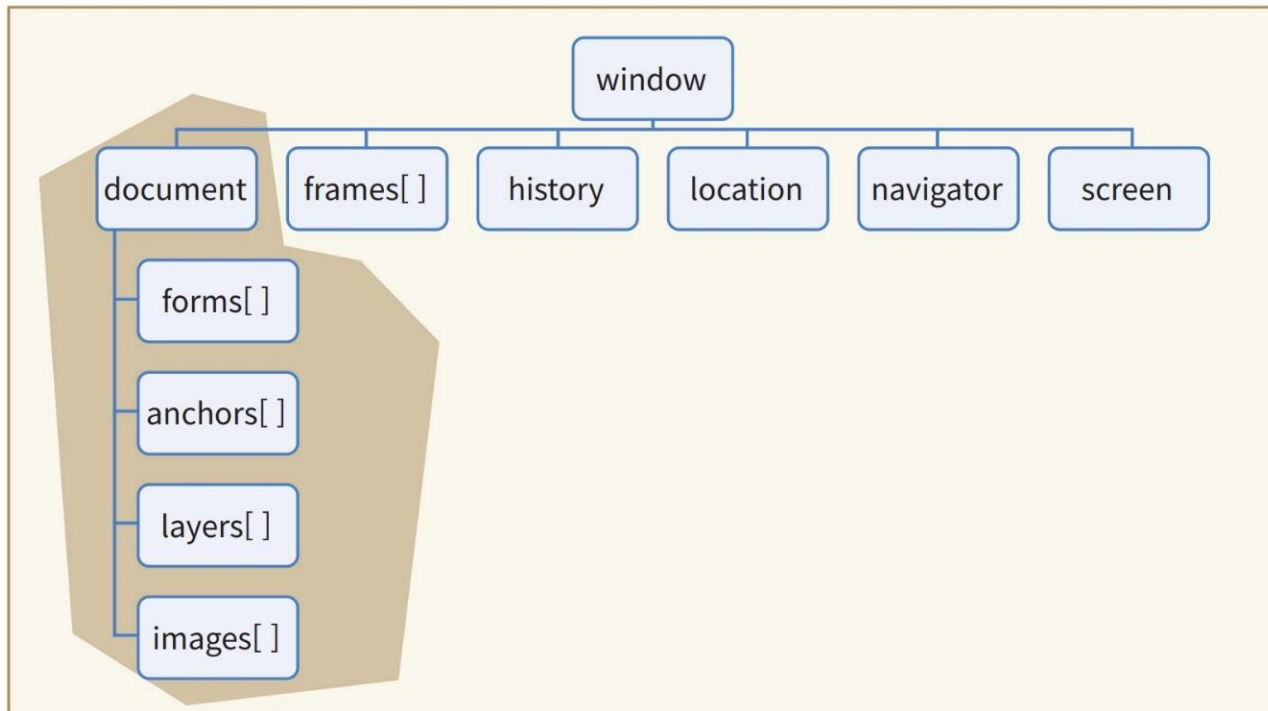
- 우리는 자바스크립트를 사용하여 **DOM** 트리를 순회하면서 페이지 안의 **HTML** 요소들의 내용(콘텐츠)이나 속성, **CSS** 스타일을 우리 마음대로 변경할 수 있다.
- 각각의 요소에서 발생하는 이벤트에 반응하는 코드를 작성할 수 있다.



자바스크립트에서 사용할 수 있는 객체들의 종류

- **HTML** 문서를 객체로 표현한 것을 **DOM**
- 웹브라우저를 객체로 표현한 것을 **BOM(Browser Object Model)**

BOM(Browsers Object Model)



DOM(Document Object Model)

HTML 요소 찾기

- 동적인 웹페이지를 작성하려면 원하는 요소를 찾아야 한다.
 - **ID**로 **HTML** 요소 찾기
 - 태그 이름으로 **HTML** 요소 찾기
 - 클래스 이름으로 **HTML** 요소 찾기
 - **CSS** 선택자로 **HTML** 요소 찾기

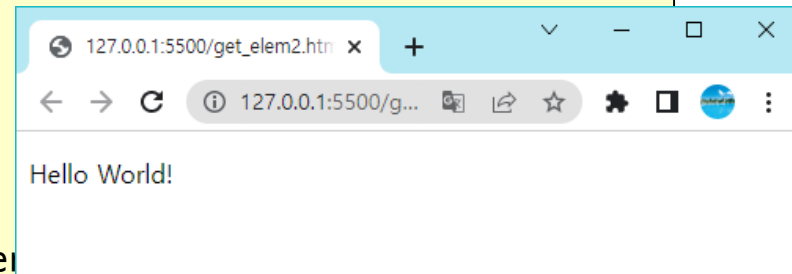
id로 HTML 요소 찾기

- **DOM에서 HTML 요소를 찾는 가장 쉬운 방법은 요소의 id를 이용하는 것이다.**

```
x = document.getElementById("main");
```

문서 안에서 id가 "main"인
요소를 찾아서 반환한다.

```
<!DOCTYPE html>
<html>
<body>
  <p id="special"></p>
  <script>
    document.getElementById("special").innerHTML
      = "Hello World!";
  </script>
</body>
</html>
```



태그 이름으로 HTML 요소 찾기

- 비슷한 메소드로 **getElementsByTagName()** 메소드가 있다. 이 메소드는 태그의 이름을 인수로 받아서 이 태그를 사용하는 모든 요소들을 배열에 넣어서 반환한다.

```
eleArray = document.getElementsByTagName("p");
```

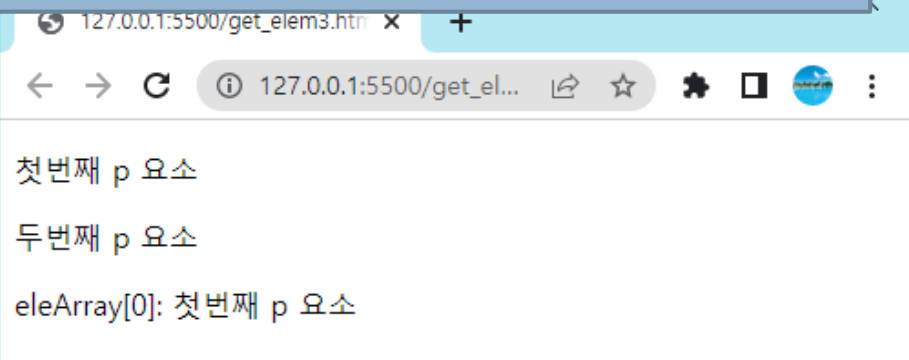
→ 문서 안의 <p> 태그를 모두 찾아서 반환한다.

→ 배열로 반환된다. 첫 번째 요소는 eleArray[0]이다.

태그 이름으로 HTML 요소 찾기

```
<!DOCTYPE html>
<html>
<body>
  <p>첫번째 p 요소</p>
  <p>두번째 p 요소</p>
  <p id="special"></p>
  <script>

</script>
</body>
</html>
```



예제

```
<!DOCTYPE ><html><body>
```

```
<ul>
```

```
<li>List item 1</li>
```

```
<li>List item 2</li>
```

```
<li>List item 3</li>
```

```
<li>List item 4</li>
```

```
<li>List item 5</li>
```

```
</ul>
```

```
<script>
```

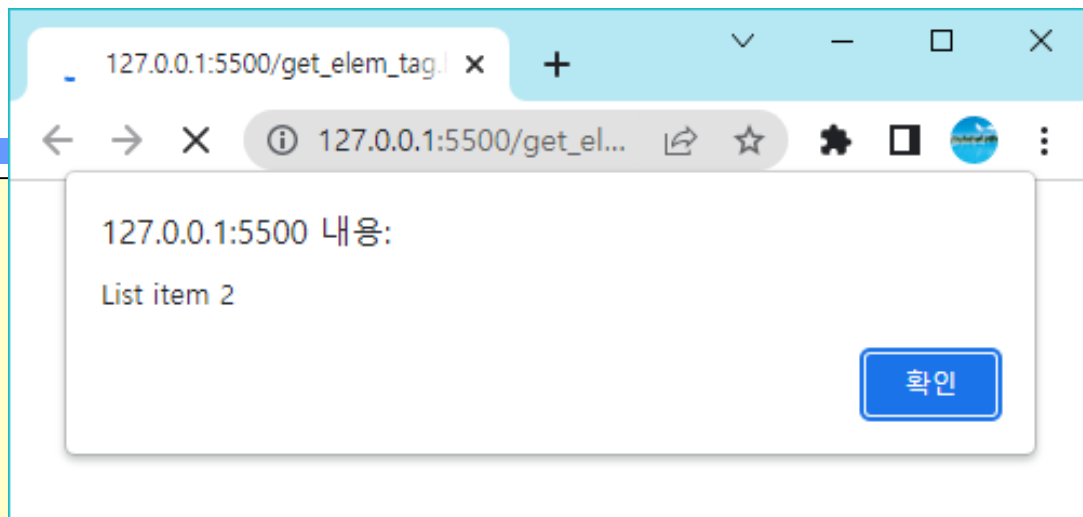
```
let list = document.getElementsByTagName('ul')[0];
```

```
let allItems = list.getElementsByTagName('li');
```

```
</script>
```


```
</body>
```

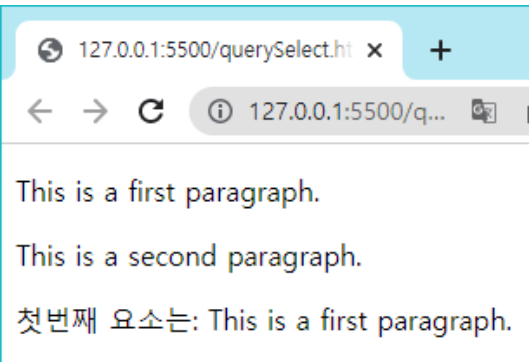
```
</html>
```



CSS 선택자로 HTML 요소 찾기

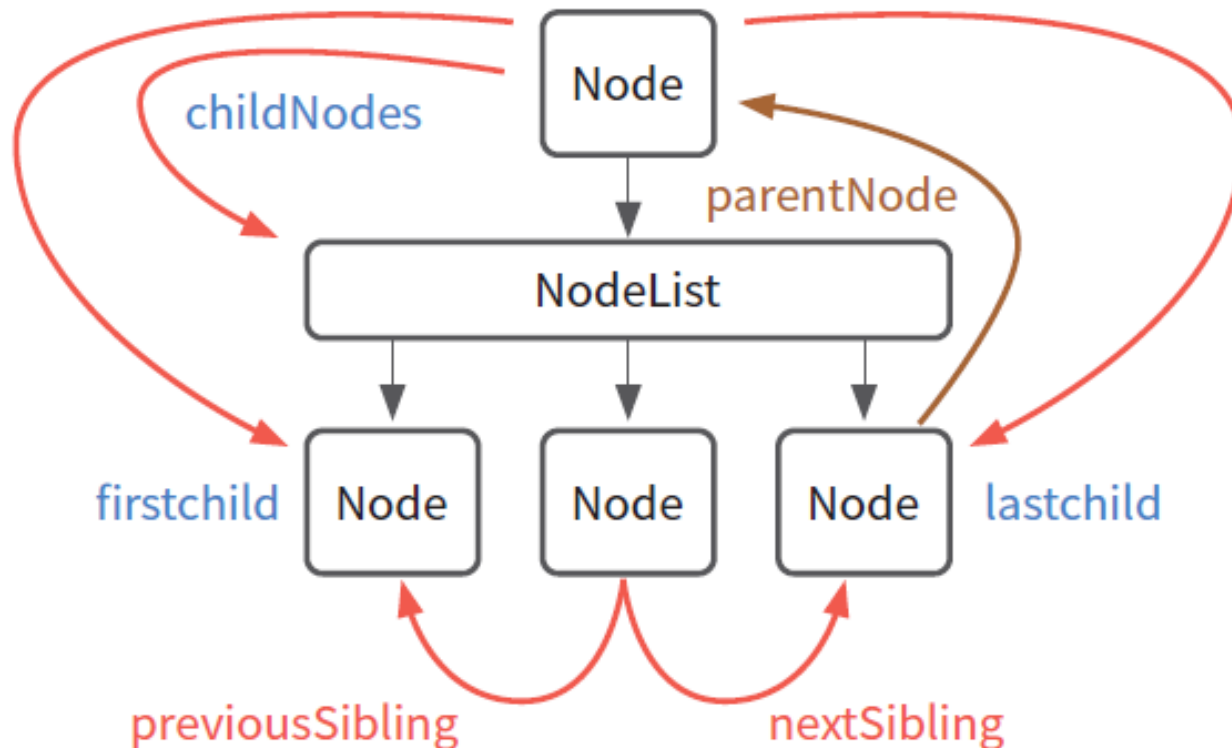
- 지정된 **CSS** 선택자와 일치하는 모든 **HTML** 요소를 찾으려면 **querySelectorAll()** 메소드를 사용한다.

```
<!DOCTYPE html>
<html>
<body>
  <p class="special">This is a first paragraph.</p>
  <p class="special">This is a second paragraph.</p>
  <p id="result"></p>
  <script>
    
  </script>
</body>
</html>
```



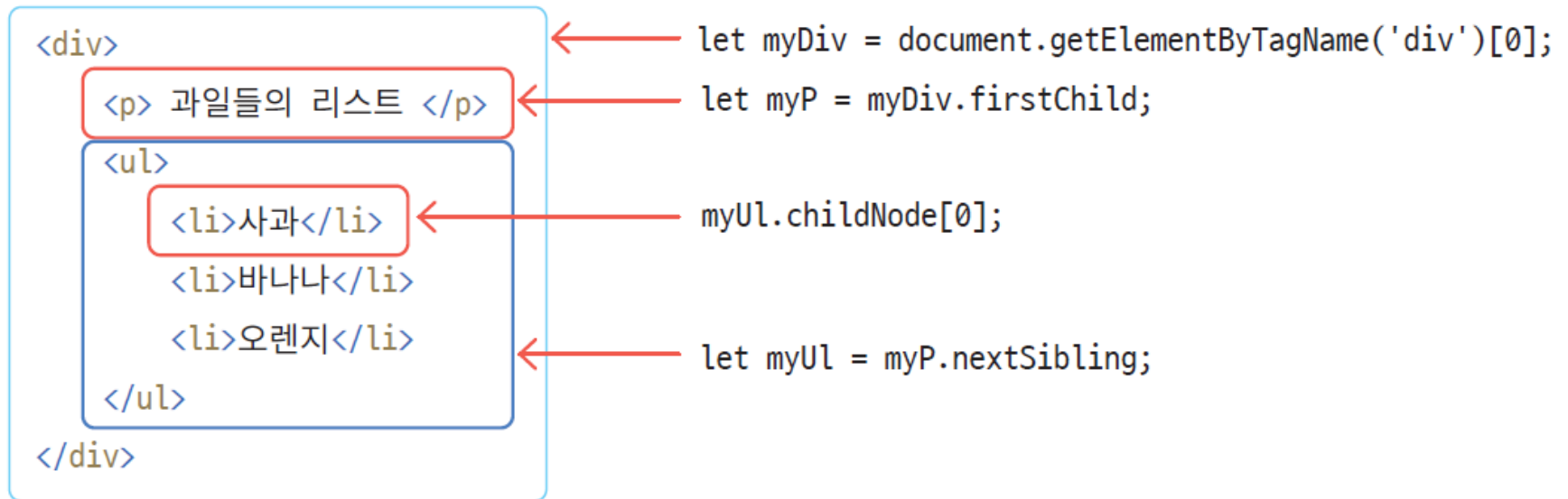
객체들 간의 관계를 이용한 DOM 트리 순회

- **DOM** 트리를 순회하는 한 가지 방법은 자식 노드와 부모 노드 관계를 이용하여 한 노드씩 방문하는 방법이다. 이 방법에서는 **childNodes[], nextSibling[], parentNode[]**을 이용한다.



childNodes[], nextSibling[], parentNode[]

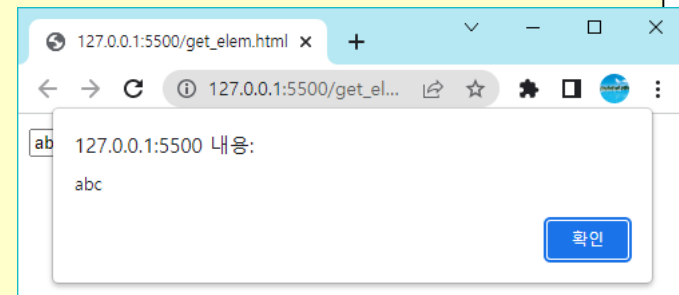
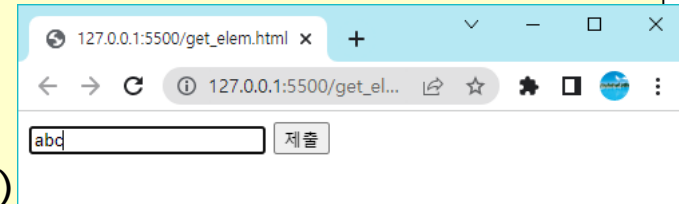
- **childNodes**: 한 요소의 모든 자식 요소에 접근할 수 있다. 배열이 반환된다.
- **firstChild**: 'childNodes' 배열의 첫 번째 자식 노드가 반환된다. 'childNodes[0]'과 같다.
- **lastChild**: 'childNodes' 배열의 마지막 자식 노드가 반환된다. 'childNodes[childNodes.length-1]'과 같다.
- **parentNode**: 현재 노드의 부모 노드를 반환한다.
- **nextSibling**: 현재 노드의 다음 형제 노드를 반환한다.
- **previousSibling**: 현재 노드의 이전 형제 노드를 반환한다.



입력 양식에서 사용자 입력값 읽기 #1

- 일반 **HTML** 요소에서는 **innerHTML** 속성을 이용하여 요소 안의 내용을 읽으면 된다. 하지만 **<input>**과 같은 입력 양식에서는 어떻게 읽어야 할까?

```
<!DOCTYPE >
<html>
<head>
  <script>
    function process() {
      let obj = document.getElementById("target")
      alert(obj.value);
    }
  </script>
</head>
<body>
  <form name="myform">
    <input type="text" id="target" name="text1">
    <input type="submit" value="제출" onclick="process()">
  </form>
</body>
</html>
```



입력 양식에서 사용자 입력값 읽기 #2

- 입력 요소를 찾을 때, 많이 사용되는 방법이 하나 더 있다.
document 객체 안에 **forms**라는 배열 객체가 있다.
forms 객체는 **document** 객체 안에 존재하는 모든 입력 양식을 저장한다.

```
<form> ← document.forms[0]
  <input type="text" value=''> ← document.forms[0].elements[0]
  <input type="text" value=''> ← document.forms[0].elements[1]
  <input type="submit">
</form>
```

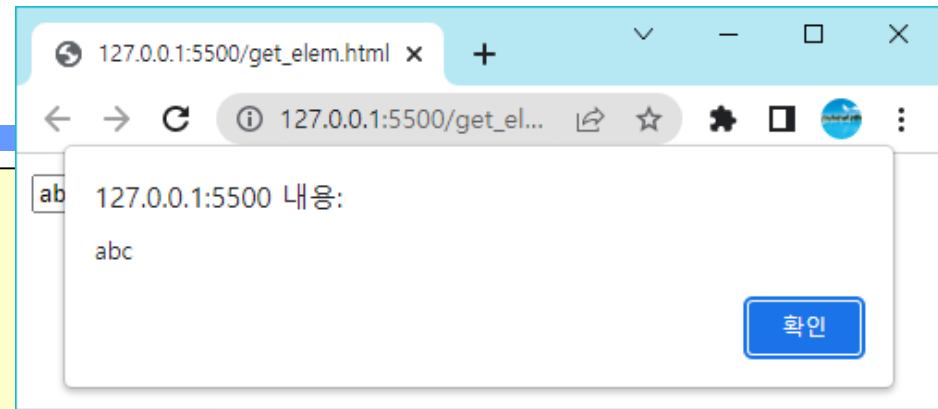
```
<form name="myform"> ← document.myform
  <input type="text" id="target1" name="text1"> ← document.myform.text1
  <input type="text" id="target2" name="text2"> ← document.myform.text2
  <input type="submit">
</form>
```


예제

```
<!DOCTYPE >  
<html>  
<head>  
  <script>
```



```
  </script>  
</head>  
<body>  
  <form name="myform">  
    <input type="text" id="target" name="text1">  
    <input type="submit" value="제출" onclick="process()>  
  </form>  
</body>  
</html>
```



텍스트 영역에서 읽기

```
<!DOCTYPE html>
<html lang="ko">
<head>
```

```
  <script>
```



```
  </script>
```

```
</head>
```

```
<body>
```

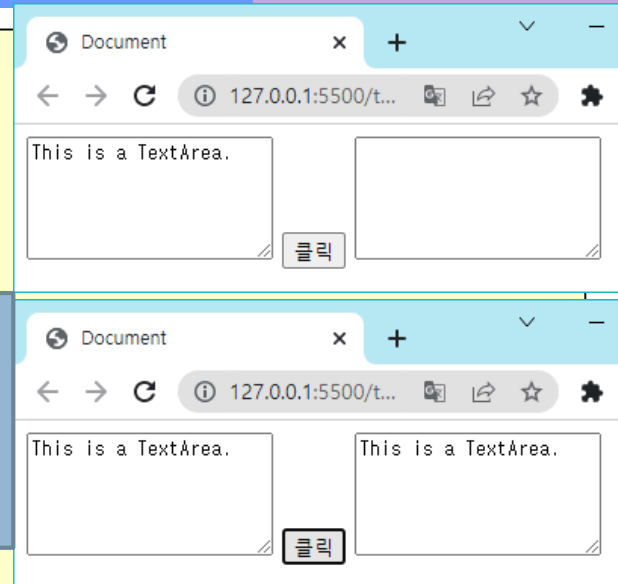
```
  <TEXTAREA id="textArea" rows=5 cols=20>This is a TextArea.</TEXTAREA>
```

```
  <input type="button" onClick="getTextAreaValue();" value="클릭">
```

```
  <TEXTAREA id="result" rows=5 cols=20></TEXTAREA>
```

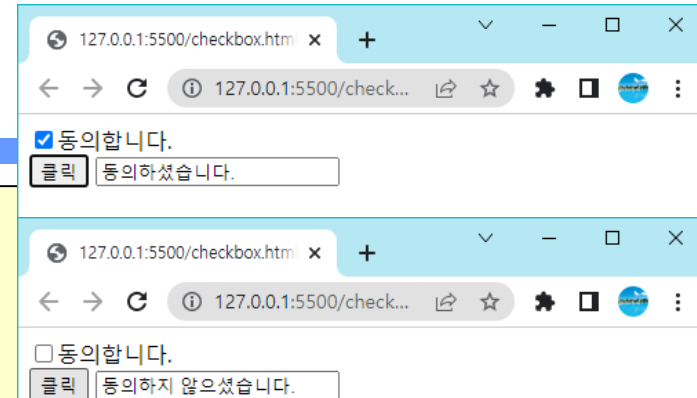
```
</body>
```

```
</html>
```



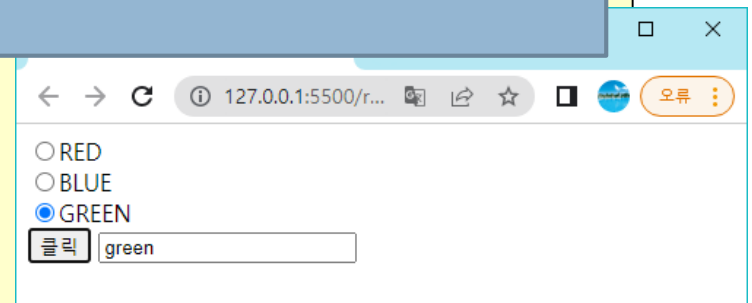
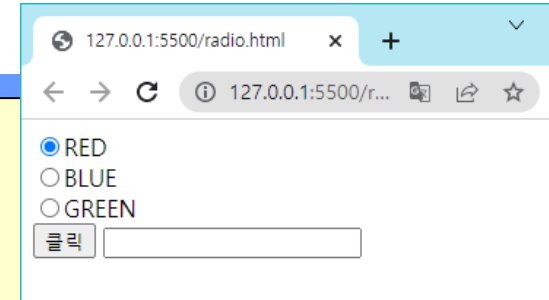
체크 박스에서 읽기

```
<!DOCTYPE html>
<html lang="ko"><head>
  <script>
    function get() {
      // [Redacted]
    }
  </script>
</head>
<body>
  <input type="checkbox" id="myCheckBox">동의합니다.<br>
  <input type="button" onClick="get();" value="클릭">
  <input type="text" id="result" value="">
</body>
</html>
```



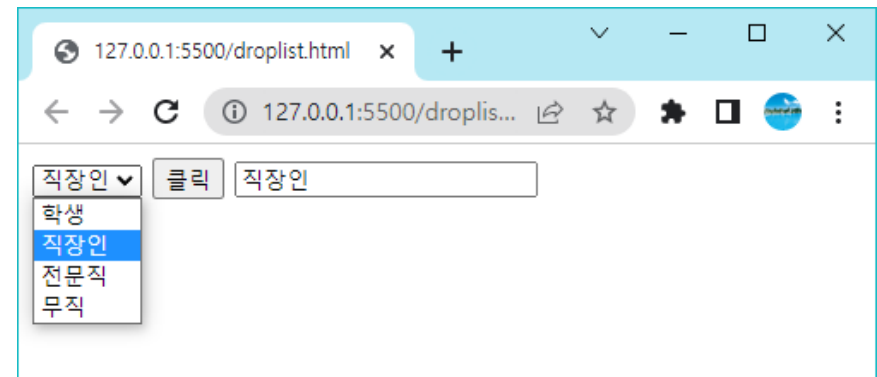
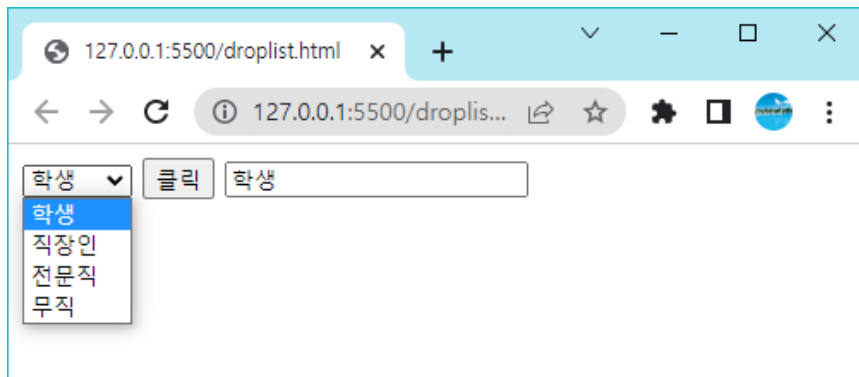
라디오 버튼에서 읽기

```
<!DOCTYPE html><html lang="en"><head>
  <script>
    function getValue() {
      let list = document.getElementsByName('color');
      for (let i = 0; i < list.length; i++) {
        if (list[i].checked) {
          document.getElementById('result').value = list[i].value;
        }
      }
    }
  </script>
</head>
<body>
  <input type="radio" name="color" CHECKED value="red">RED<BR>
  <input type="radio" name="color" value="blue">BLUE<BR>
  <input type="radio" name="color" value="green">GREEN<BR>
  <input type="button" onClick="getValue();" value="클릭">
  <input type="text" id="result" value="">
</body>
</html>
```



SelectList에서 값읽기

- **SelectList**는 풀다운 메뉴의 기능을 하는 입력 양식 요소이다. 여기서는 다음과 같이 사용자가 선택한 값을 읽을 수 있다.



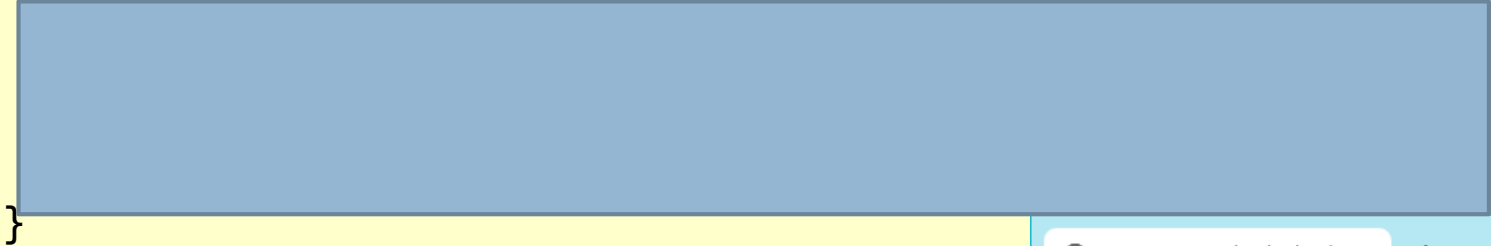
라디오 버튼에서 읽기

```
<!DOCTYPE html><html lang="en"><head>
```

```
<script>
```

```
function getValue() {
```

```
    let selected, s;
```



```
}
```

```
</script></head>
```

```
<body>
```

```
<select id="myList">
```

```
    <option value="학생">학생</option>
```

```
    <option value="직장인" SELECTED>직장인</option>
```

```
    <option value="전문직">전문직</option>
```

```
    <option value="무직">무직</option>
```

```
</select>
```

```
<input type="button" onClick="getValue()" value="클릭">
```

```
<input type="text" id="result" value="">
```

```
</body>
```

```
</html>
```

