

**01**

**문서 객체모델 이해하기**

# 01 문서 객체모델 이해하기

## ■ 문서 객체모델

### 1. HTML 태그가 객체 형태인 이유

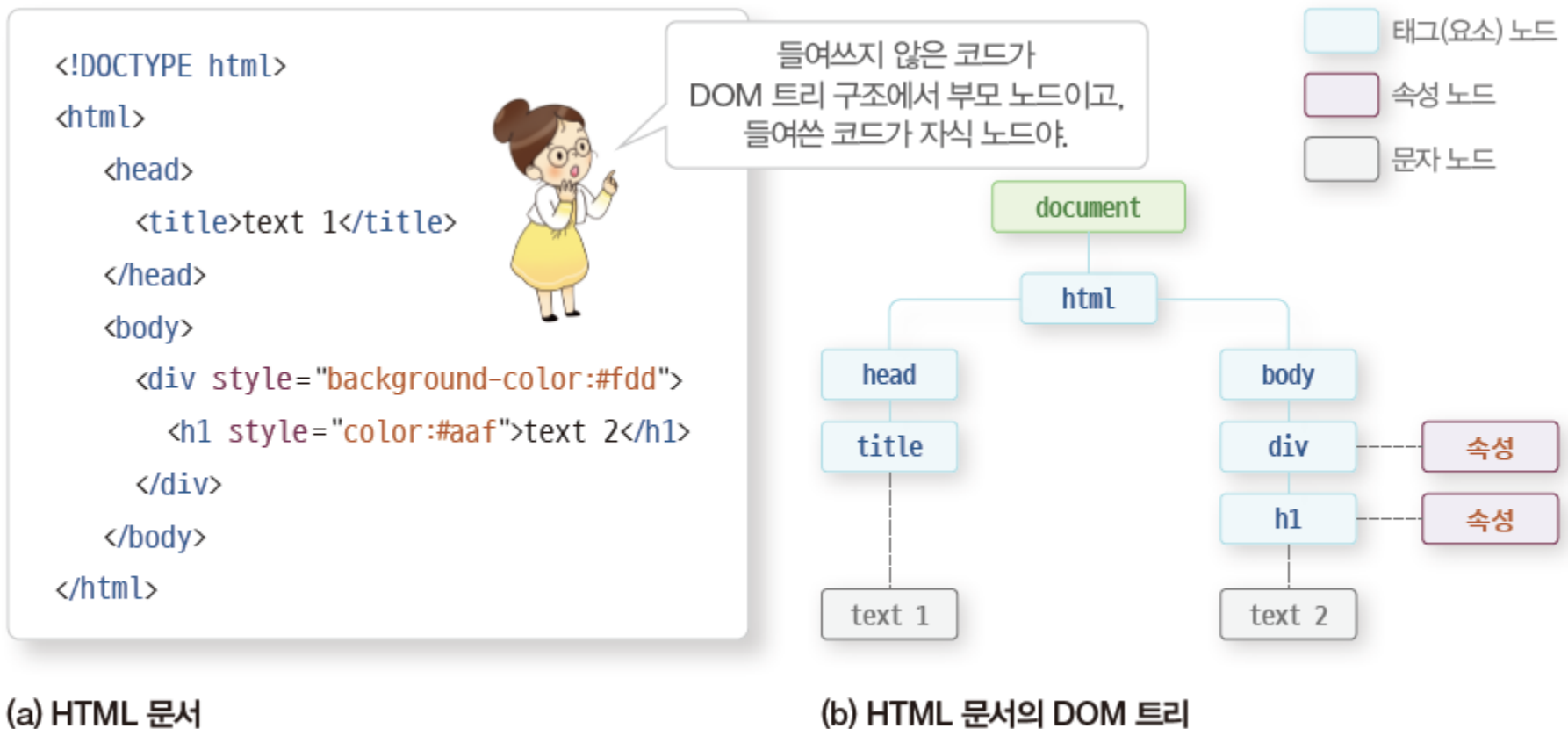


그림 8-1 DOM 트리 구조

# 01 문서 객체모델 이해하기

---

## ■ 웹 시스템 관련 객체

- Math, Date 객체
- 문서 객체모델의 객체(DOM)
- 브라우저 객체모델의 객체(BOM)
- 이벤트 객체

## ■ 공통 CSS 파일

- 이 장에서 사용하는 외부 CSS 파일 `style_js.css`

**02**

**DOM 객체 다루기**

## 02 DOM 객체 다루기

### ■ HTML 요소에 접근하기

#### 1. 요소 하나에 접근하기

표 8-2 요소 하나에 접근하는 메서드

노드	특징
<code>document.getElementById("id")</code>	id를 가진 태그 요소 하나에 접근

## 02 DOM 객체 다루기

### ■ HTML 요소에 접근하기

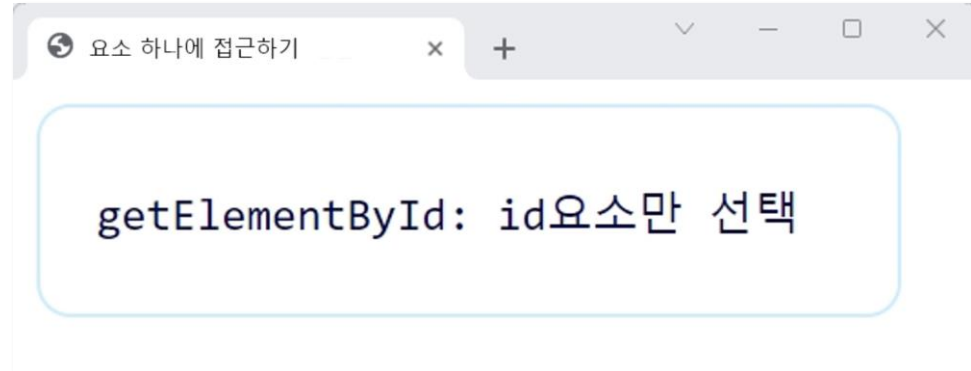
#### 1. 요소 하나에 접근하기

예제 8-1

요소 하나에 접근하기

ex8-1.html

```
</head>
<body>
  <p id="ide"></p>
  <script>
    document.getElementById("ide").innerHTML
      = "getElementById: id요소만 선택";
  </script>
</body>
</html>
```



## 02 DOM 객체 다루기

### ■ HTML 요소에 접근하기

#### 2. 요소 여러 개에 접근하기

표 8-3 요소 여러 개에 접근하는 메서드

메서드	내용
<code>document.getElementsByClassName("class")</code>	class를 가진 모든 태그 요소에 접근
<code>document.getElementsByTagName("tag")</code>	tag를 가진 모든 태그 요소에 접근
<code>document.getElementsByName("name")</code>	name을 가진 태그 요소에 접근

## 02 DOM 객체 다루기

### 예제 8-2

### 요소 여러 개에 접근하기

```
</head>
<body>
  <p>p & No class</p>
  <p class="elm">p class="elm" 1</p>
  <p class="elm">p class="elm" 2</p><hr>
  <script>
    const elms = document.getElementsByClassName("elm");
    for (let i = 0; i < elms.length; i++)
      document.write(`ByClassName: elms[${i}] = ${elms[i].innerHTML}<br>`);
    document.write("<hr>");
    const tags = document.getElementsByTagName("p");
    for (let i = 0; i < tags.length; i++)
      document.write(`ByTagName P: tags[${i}] = ${tags[i].innerHTML}<br>`);
  </script>
</body>
</html>
```

요소 여러 개에 접근하기

p & No class

p class="elm" 1

p class="elm" 2

ByClassName: elms[0] = p class="elm" 1

ByClassName: elms[1] = p class="elm" 2

ByTagName P: tags[0] = p & No class

ByTagName P: tags[1] = p class="elm" 1

ByTagName P: tags[2] = p class="elm" 2



## 02 DOM 객체 다루기

### ■ HTML 요소에 접근하기

#### 3. 태그 선택자로 요소에 접근하기

메서드	내용
<code>document.querySelectorAll("s")</code>	s 태그 선택자를 사용하여 태그 요소에 접근

## 02 DOM 객체 다루기

### 예제 8-3

### 태그 선택자로 요소에 접근하기

```
</head>
<body>
  <p class="elm">p class="elm" 1</p>
  <p class="elm">p class="elm" 2</p>
  <p>p & No class 1</p>
  <p id="ide">p id="ide"</p><hr>
  <script>
    const dots = document.querySelectorAll(".elm");
    for (let i = 0; i < dots.length; i++)
      document.write(`dots[${i}] = ${dots[i].innerHTML}<br>`);
    document.write("<hr>");
    const hash = document.querySelectorAll("#ide");
    for (let i = 0; i < hash.length; i++)
      document.write(`hash[${i}] = ${hash[i].innerHTML}<br>`);
  </script>
</body>
</html>
```

p class="elm" 1

p class="elm" 2

p & No class 1

p id="ide"

---

dots[0] = p class="elm" 1

dots[1] = p class="elm" 2

---

hash[0] = p id="ide"

## 02 DOM 객체 다루기

### ■ HTML 요소에서 값 가져오기

#### 1. 웹 폼 태그의 문자열 읽기

- input 태그의 값 읽기

```
<form name="myform">
  <input type="text" name="userId">
</form>
<script>
  const id1 = document.myform.userId;
  alert(`${id1.value}님 환영합니다`);
</script>
```

## 02 DOM 객체 다루기

### ■ HTML 요소에서 값 가져오기

#### 1. 웹 폼 태그의 문자열 읽기

- input 태그의 값 읽기

예제 8-4

input 태그의 값 읽기

ex8-4.html

```
</head>
<body>
  <form name="myform">
    사용자ID <input type="text" name="userId"><br>
    비밀번호 <input type="password" name="passwd"><br>
    <input type="submit" value="log in" onClick="getInput()">
  </form>
```

사용자ID

zoch

패스워드

...

log in

## 02 DOM 객체 다루기

### ■ HTML 요소에서 값 가져오기

#### 1. 웹 폼 태그의 문자열 읽기

- input 태그의 값 읽기

```
<script>
function getInput() {
  const id = document.myform.userId;
  const pw = document.myform.passwd;
  if (pw.value.length <= 4)
    alert(`${id.value}님 패스워드를 변경하세요`);
  else
    alert(`${id.value}님 환영합니다`);
}
</script>
</body>
</html>
```

이 페이지 내용:

zoch님 패스워드를 변경하세요

확인

## 02 DOM 객체 다루기

- textarea에서 값 읽기

예제 8-5

textarea에서 값 읽기

ex8-5.html

```
</head>
<body>
  <textarea id="input" rows="5" cols="20"></textarea><br>
  <input type="button" value="Get Text" onClick="getTextarea()"><hr>
  <p id="result"></p>
  <script>
    function getTextarea() {
      const st = document.getElementById("input").value;
      document.getElementById("result").innerHTML = st;
    }
  </script>
</body>
</html>
```



## 02 DOM 객체 다루기

- textarea에서 값 읽기

```
</head>
```

```
<body>
```

```
<textarea id="input" rows="5" cols="20"></textarea><br>
```

```
<input type="button" value="Get Text" onClick="getTextarea()"><hr>
```

```
<p id="result"></p>
```



```
<pre id="result"></pre>
```

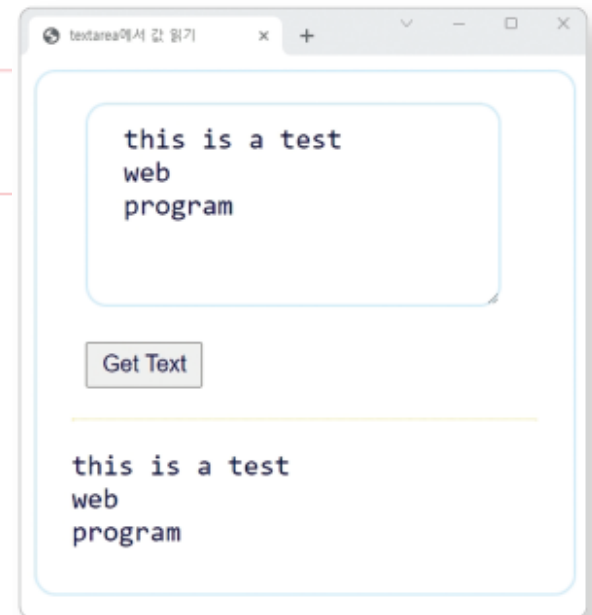


그림 8-2 <pre> 태그로 줄 바꿈을 출력한 결과

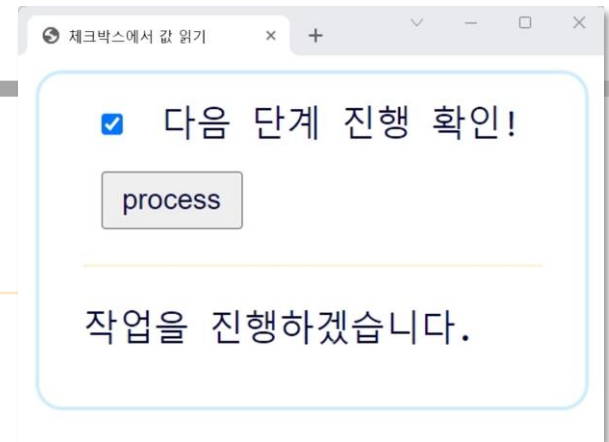
## 02 DOM 객체 다루기

- 체크박스에서 값 읽기

### 예제 8-6

### 체크박스에서 값 읽기

```
</head>
<body>
  <input type="checkbox" id="myCheckbox"> 다음 단계 진행 확인!<br>
  <input type="button" value="process" onClick="getCheckbox()"><hr>
  <p id="result"></p>
  <script>
    function getCheckbox() {
      if (document.getElementById("myCheckbox").checked == true)
        document.getElementById("result").innerHTML="작업을 진행하겠습니다.";
      else
        document.getElementById("result").innerHTML="사용자의 요구로 중단!";
    }
  </script>
</body>
</html>
```





## 02 DOM 객체 다루기

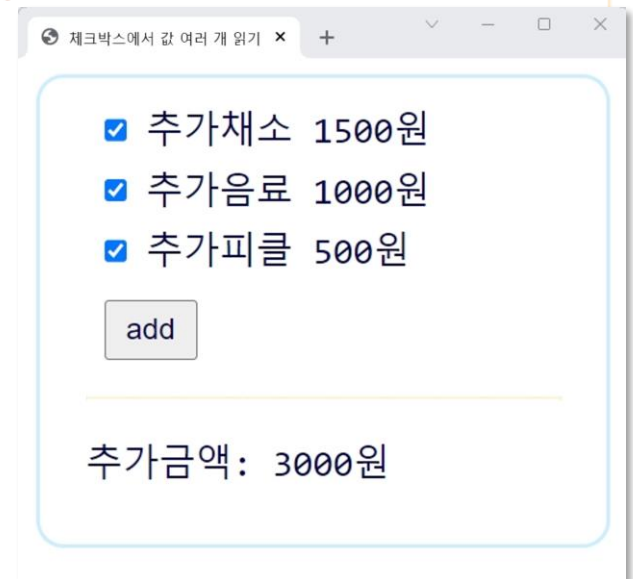
- 체크박스에서 값 여러 개 읽기

예제 8-7

체크박스에서 값 여러 개 읽기

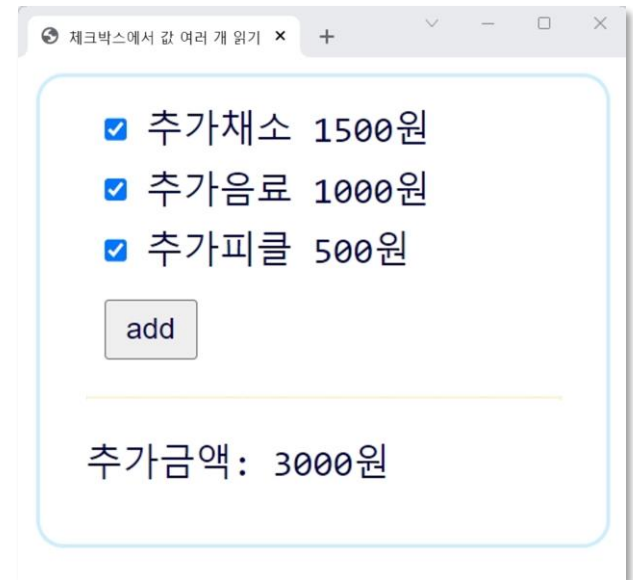
ex8-7.html

```
</head>
<body>
  <input type="checkbox" name="myAdd" value="1500">추가채소 1500원<br>
  <input type="checkbox" name="myAdd" value="1000">추가음료 1000원<br>
  <input type="checkbox" name="myAdd" value="500">추가피클 500원<br>
  <input type="button" value="add" onClick="getCheckbox()"><br>
  <p id="result"></p>
  <script>
    function getCheckbox() {
      let sum = 0;
      const adds = document.getElementsByName("myAdd");
```



## 02 DOM 객체 다루기

```
for (let i = 0; i < adds.length; i++) {  
  if (adds[i].checked)  
    sum = sum + parseInt(adds[i].value);  
}  
document.getElementById("result").innerHTML = `추가금액: ${sum}원`;  
}  
</script>  
</body>  
</html>
```



## 02 DOM 객체 다루기

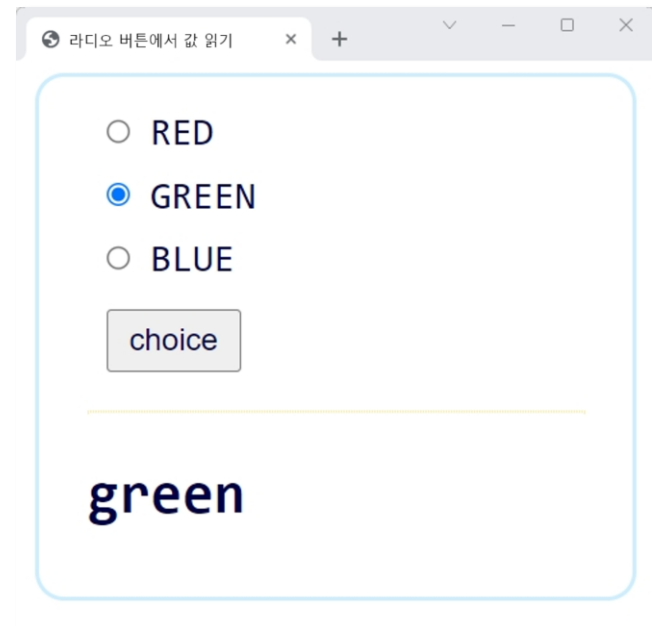
- 라디오 버튼에서 값 읽기

예제 8-8

라디오 버튼에서 값 읽기

ex8-8.html

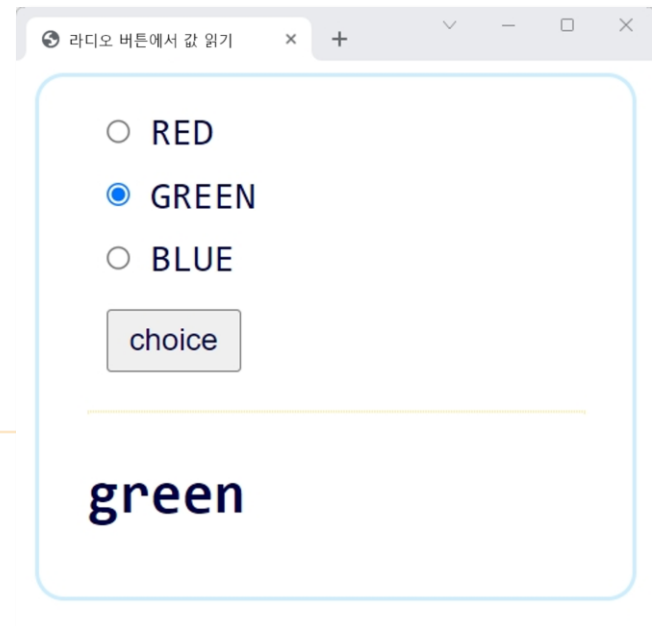
```
</head>
<body>
  <input type="radio" name="color" value="red" checked>RED<br>
  <input type="radio" name="color" value="green">GREEN<br>
  <input type="radio" name="color" value="blue">BLUE<br>
  <input type="button" value="choice" onClick="getRadio()"><br>
  <h2 id="result"></h2>
```



## 02 DOM 객체 다루기

- 라디오 버튼에서 값 읽기

```
<script>
function getRadio() {
  const choice = document.querySelectorAll("[type='radio']");
  for (let i = 0; i < choice.length; i++) {
    if (choice[i].checked) {
      document.getElementById("result").innerHTML=`${choice[i].value}`;
      break;
    }
  }
}
</script>
</body>
</html>
```



## 02 DOM 객체 다루기

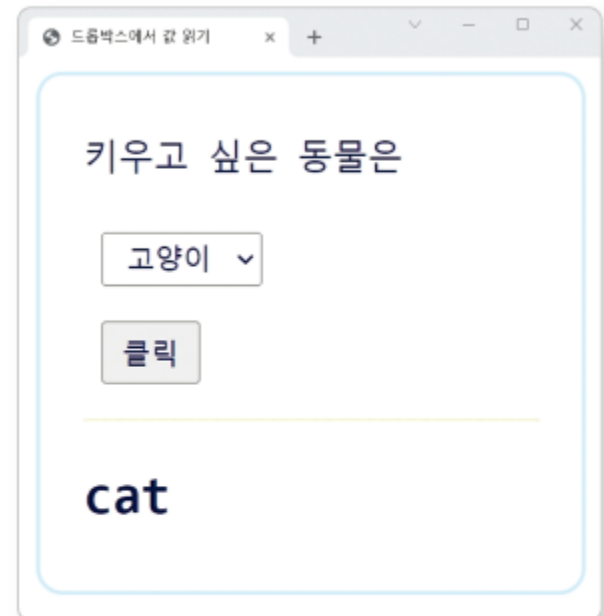
- 드롭박스에서 값 읽기

예제 8-9

드롭박스에서 값 읽기

ex8-9.html

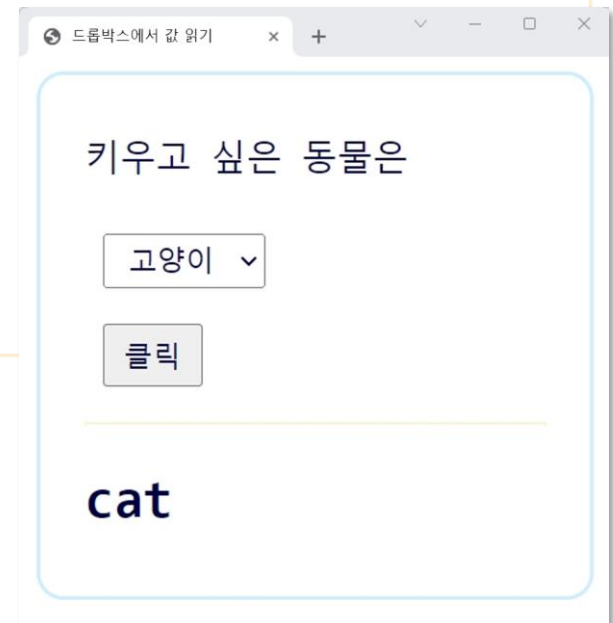
```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>드롭박스에서 값 읽기</title>
  <link rel="stylesheet" href="style_js.css">
</head>
<body>
  <p>키우고 싶은 동물은</p>
  <select id="myOption">
    <option value="dog">개</option>
    <option value="cat">고양이</option>
    <option value="duck">오리</option>
    <option value="koala">코알라</option>
  </select><br>
```



## 02 DOM 객체 다루기

- 드롭박스에서 값 읽기

```
<input type="button" value="클릭" onClick="getSelect()"><hr>
<h2 id="result"></h2>
<script>
  function getSelect() {
    const index = document.getElementById("myOption").selectedIndex;
    const st = document.getElementById("myOption").options[index].value;
    document.getElementById("result").innerHTML = st;
  }
</script>
</body>
</html>
```



## 02 DOM 객체 다루기

### ■ HTML 요소 변경하기

#### 1. 내용 변경하기

- 텍스트를 출력하는 메서드

표 8-4 HTML에 텍스트를 출력하는 메서드

메서드	내용
<code>document.write()</code>	HTML 페이지를 새로 만들고 글자를 출력
<code>document.writeln()</code>	글자 끝에 줄 바꿈 문자를 추가하여 출력

- 문자 노드에 접근하는 속성

표 8-5 문자 노드에 접근하는 속성

속성	내용
<code>innerHTML</code>	HTML 태그 요소를 직접 변경함. HTML 태그를 인식함
<code>innerText</code>	HTML이 적용된 상태로 불러옴. 따라서 <code>display:none</code> 의 문자는 읽을 수 없음
<code>textContent</code>	HTML을 제외한 모든 문자를 가져옴. 따라서 <code>display:none</code> 의 문자도 가져옴

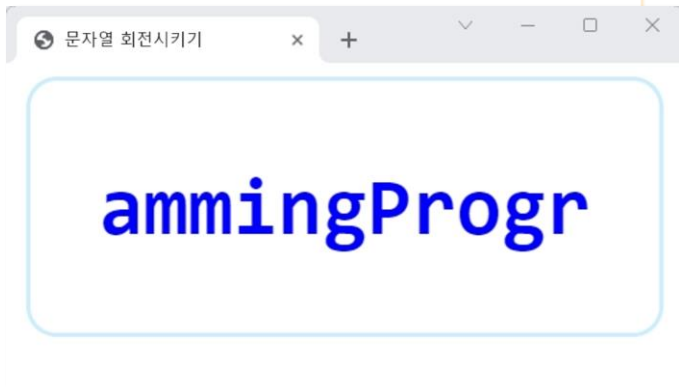
## 02 DOM 객체 다루기

예제 8-10

문자열 회전시키기

ex8-10.html

```
</head>
<body style="text-align: center">
  <h1 id="rotate">Programming</h1>
  <script>
    const rotate = document.getElementById("rotate");
    function rotateString() {
      const text = rotate.textContent;
      const firstChar = text[0];
      const rotatedText = text.slice(1) + firstChar;
      rotate.textContent = rotatedText;
    };
    setInterval(rotateString, 500); // 500ms마다 문자열을 회전
  </script>
</body>
</html>
```





## 02 DOM 객체 다루기

### ■ HTML 요소 변경하기

#### 2. 속성 변경하기

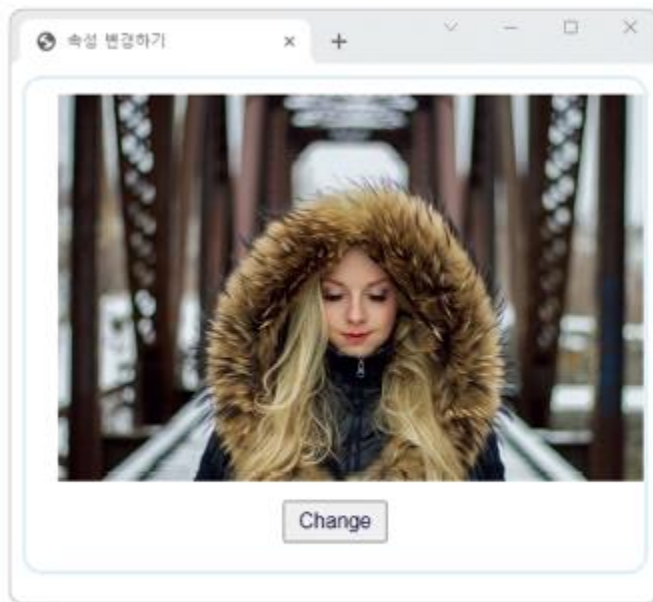
예제 8-11

속성 변경하기

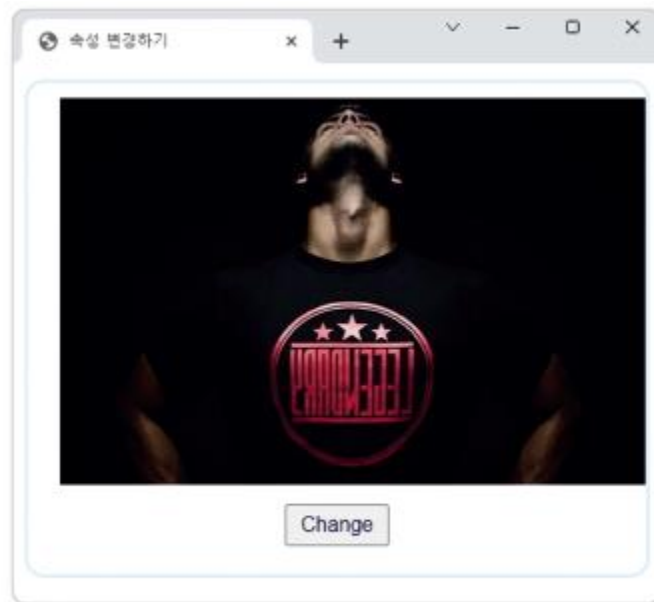
ex8-11.html

```
</head>
<body style="text-align: center">
  <br>
  <input type="button" value="Change" onClick="changeImage()" >
  <script>
    let toggle = 1;
    function changeImage() {
      if (++toggle % 2 == 0)
        document.getElementById("image").src = "../media/photo2.jpg";
      else
        document.getElementById("image").src = "../media/photo1.jpg";
    }
  </script>
```

## 02 DOM 객체 다루기



번갈아 나타나는 사진(1)



번갈아 나타나는 사진(2)

## 02 DOM 객체 다루기

### ■ HTML 요소 변경하기

#### 3. 스타일 변경하기

예제 8-12

스타일 변경하기

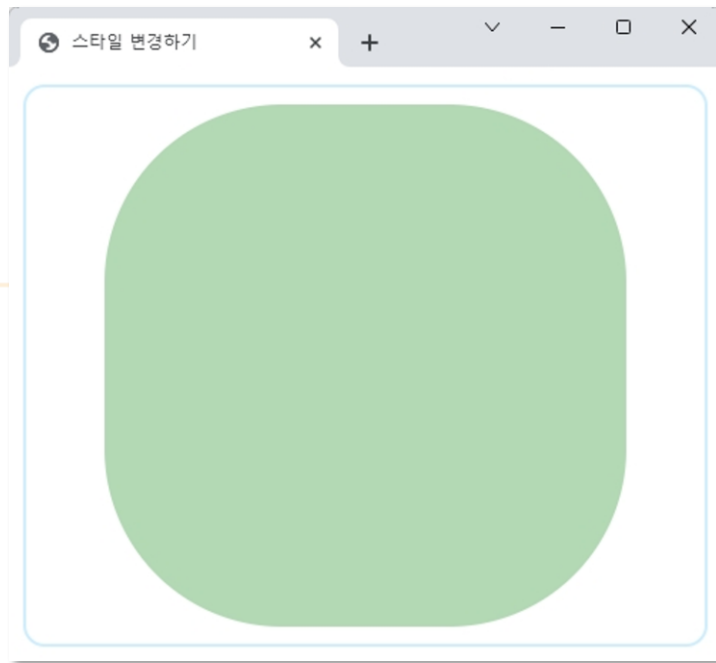
ex8-12.html

```
</head>
<body>
  <div id="box" style="width: 300px; height: 300px; margin: auto"></div>
  <script>
    const letters = "0123456789ABCDEF";
    let color = "#";
    let count = 1;
    let intervalID = setInterval(changeColor, 250);
```

## 02 DOM 객체 다루기

```
function changeColor() {  
  if (++count > 50)  
    clearInterval(intervalID);  
  for (let k = 0; k < 6; k++)  
    color = color + letters[Math.floor(Math.random() * 16)];  
  document.getElementById("box").style.backgroundColor = color;  
  document.getElementById("box").style.borderRadius = `${count}%`;  
  color = "#";  
}  
</script>  
</body>  
</html>
```

// 16진수 문자열 초기화



## 02 DOM 객체 다루기

### ■ HTML 요소 추가와 삭제하기

#### 1. 노드 추가하기

표 8-6 DOM에서 노드를 추가하거나 삭제하는 메서드

메서드	내용
<code>createElement("tag")</code>	<tag> 노드를 생성한다.
<code>appendChild(node)</code>	부모 노드에 node를 추가한다.
<code>removeChild(node)</code>	부모 노드에서 node를 삭제한다.

## 02 DOM 객체 다루기

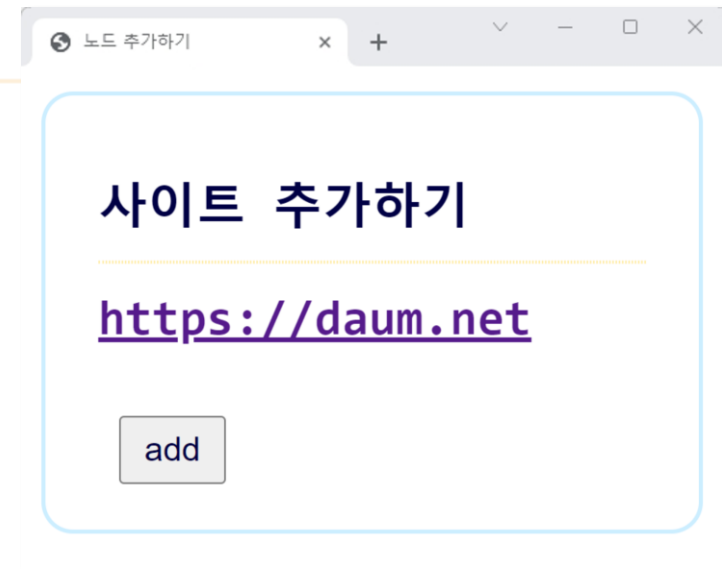
### ■ HTML 요소 추가와 삭제하기

#### 1. 노드 추가하기

예제 8-13

노드 추가하기

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>노드 추가하기</title>
  <link rel="stylesheet" href="style_js.css">
</head>
<body>
  <h3 id="target">사이트 추가하기</h3>
  <input type="button" value="add" onClick="addNode('https://daum.net')">
```

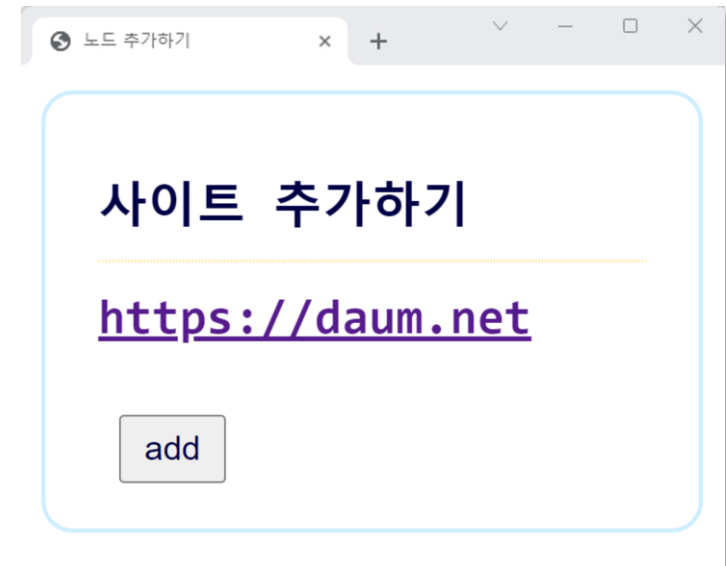


## 02 DOM 객체 다루기

### ■ HTML 요소 추가와 삭제하기

#### 1. 노드 추가하기

```
<script>
  function addNode(t) {
    let node = document.createElement("a");
    node.href = t;
    node.innerHTML = `<hr>${t}`;
    document.getElementById("target").appendChild(node);
  }
</script>
</body>
</html>
```



## 02 DOM 객체 다루기

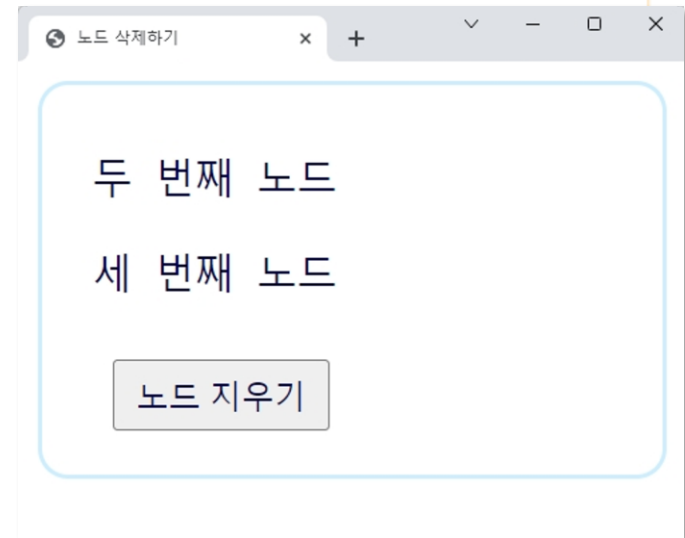
### ■ HTML 요소 추가와 삭제하기

#### 2. 노드 삭제하기

예제 8-14    노드 삭제하기

ex8-14.html

```
<body>
  <div id="target">
    <p id="p1">첫 번째 노드</p>
    <p id="p2">두 번째 노드</p>
    <p id="p3">세 번째 노드</p>
  </div>
  <button onClick="removeNode()">노드 지우기</button>
```





## 02 DOM 객체 다루기

### ■ HTML 요소 추가와 삭제하기

#### 2. 노드 삭제하기

```
<script>
  let num = 1;
  function removeNode() {
    const parent = document.getElementById("target");
    const child = document.getElementById(`p${num}`);
    parent.removeChild(child);
    num++;
  }
</script>
</body>
</html>
```

