

웹 프로그래밍 강의노트

DOM 모델

2024년 2학기

DOM의 정의 및 문서 구조

- DOM (Document Object Model)
 - 웹 문서를 자바스크립트 입장에서 구조적 문서 (document) 객체 형태로 다루는 모델
 - 자바스크립트는 HTML 문서를 객체(Object)로 바라보고 다룬다
 - HTML 문서 뿐만 아니라 CSS 속성도 변경 가능
- DOM 표준
 - 2000년에 DOM2 제정
 - 대부분의 웹브라우저가 지원
 - 2004년에 DOM3 까지 제정된 상태
- 자바스크립트 활용
 - 자바스크립트를 이용하여 HTML 문서의 내용을 변경
 - 사용자 입력을 받아 처리

DOM과 HTML 웹 문서의 관계

- DOM을 이용해 웹문서를 프로그램 언어에서의 변수나 구조체와 같은 데이터로 처리할 수 있도록 한다

자바스크립트를 이용해 DOM의 내용을 추가/변경

```
graph TD; A[자바스크립트를 이용해 DOM의 내용을 추가/변경] --> B[HTML 문서의 태그/콘텐츠가 변경되는 효과]; B --> C[화면에 디스플레이 되는 내용도 변경]; C --> D[HTML 문서의 내용을 동적으로 변경];
```

HTML 문서의 태그/콘텐츠가 변경되는 효과

화면에 디스플레이 되는 내용도 변경

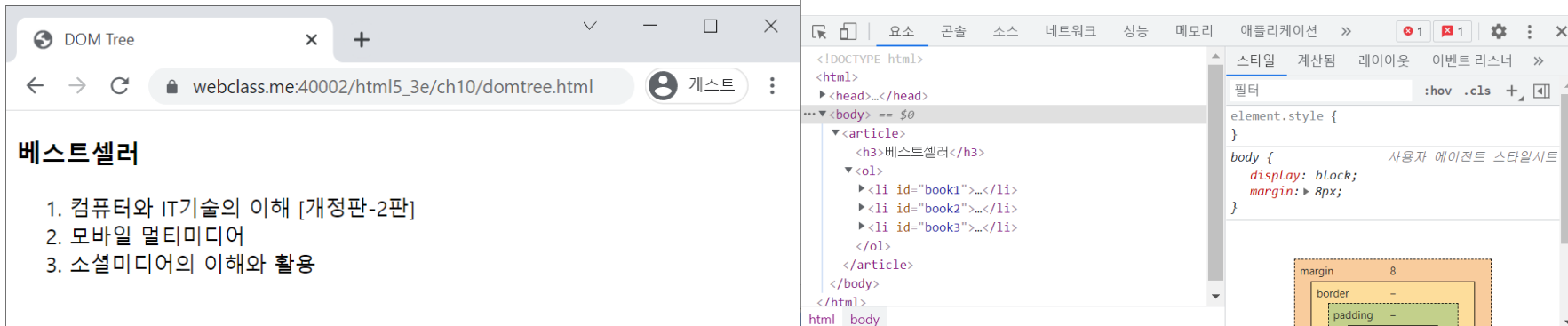
HTML 문서의 내용을 동적으로 변경

트리 구조의 DOM

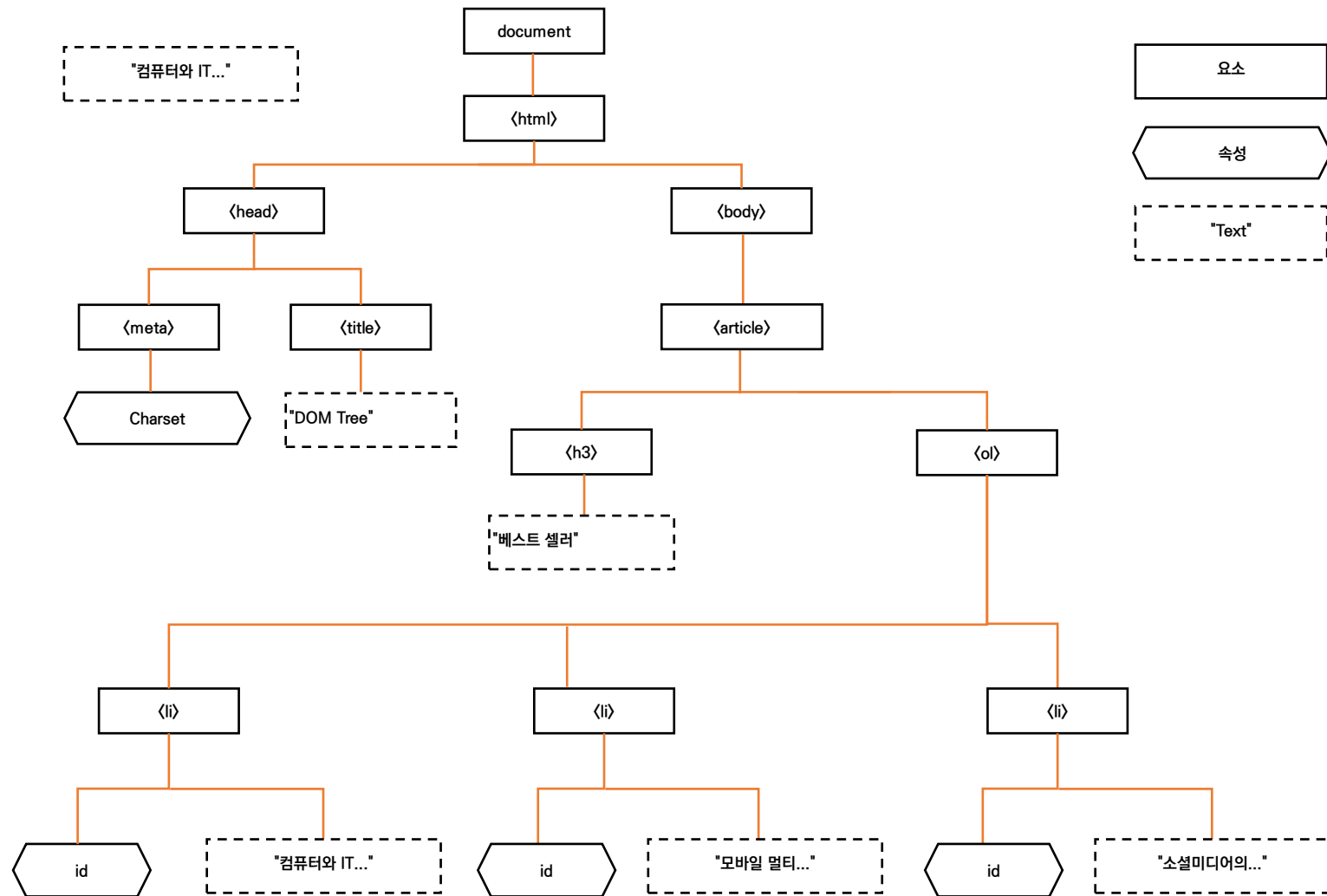
- HTML 문서는 태그 요소의 계층적 구조
 - DOM도 트리 구조의 형태
 - 트리의 노드: HTML 태그 요소
 - 노드는 속성과 속성값을 가짐
- DOM 구조는 웹 브라우저에서 확인 가능
 - 개발자 도구 이용

웹 브라우저에서 DOM 구조 확인

```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <meta charset="utf-8"/>
5 <title> DOM Tree </title>
6 </head>
7 <body>
8   <article>
9     <h3>베스트셀러</h3>
10    <ol>
11      <li id="book1"> 컴퓨터와 IT기술의 이해 [개정판-2판] </li>
12      <li id="book2"> 모바일 멀티미디어 </li>
13      <li id="book3"> 소셜미디어의 이해와 활용</li>
14    </ol>
15  </article>
16 </body>
17 </html>
```



DOM 트리 구조



HTML 태그 요소와 DOM

- 태그 요소는 DOM의 객체로 표현 됨
 - 태그 요소에 포함된 다른 요소는 객체내에 소속된 객체 형태로 표현 (하위객체)
- 태그 속성은 DOM 객체의 속성으로 표현 됨
- 요소 전체가 하나의 객체

```
<input type = "text" name = "username"/>
```

- type과 name은 속성
- "text"와 "username"은 type과 name의 속성값

DOM을 통한 HTML 문서 접근

DOM을 통한 HTML 문서 접근

- 자바스크립트 입장에서는 웹 브라우저 환경과 HTML 문서를 모두 객체로 바라보고 처리
 - 일반 프로그래밍에서 처럼 객체에 접근해서 값을 읽어내거나, 저장하고 수정하는 작업을 수행함으로써 웹 브라우저나 HTML 문서에 대한 처리를 수행
- 여러가지 작업 수행 가능
 - HTML 문서의 요소를 변경할 수 있다
 - HTML 문서의 요소의 속성을 변경할 수 있다
 - HTML 문서의 요소의 스타일 속성을 변경할 수 있다
 - HTML 문서의 요소를 삭제하거나 속성을 삭제할 수 있다
 - 새로운 HTML 요소나 속성을 생성해서 기존의 DOM에 추가할 수 있다

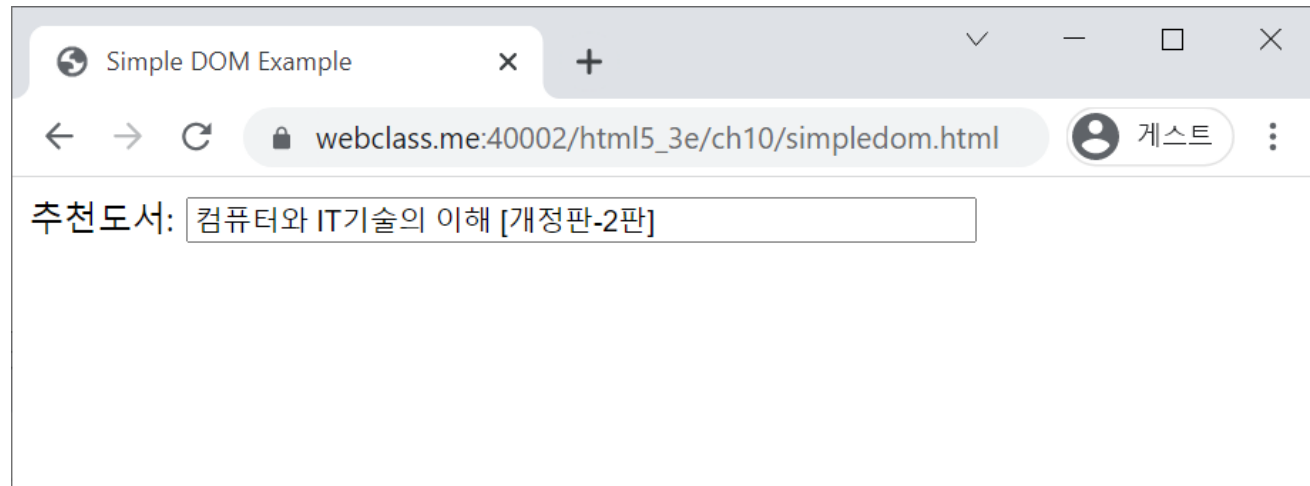
DOM을 통한 HTML 문서 접근

- DOM 접근 방법

1. document의 forms나 images 속성을 이용해서 접근하는 방법
2. document 객체가 제공하는 getElementByTagName(), getElementByClassName() 메소드를 이용해 태그나 클래스 이름으로 접근하는 방법
3. document 객체가 제공하는 getElementById() 등의 메소드를 이용해서 접근하는 방법
 - 가장 사용이 쉽고 많이 사용되고 방법인 getElementById() 메소드 방법을 중심으로 설명

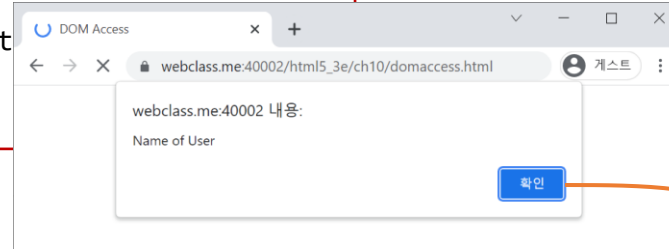
DOM 접근 방법 예제 (1)

```
1 <form>
2     추천도서: <input id="recommend" type="text" size="50">
3 </form>
4
5 <script type="text/javascript">
6     dom = document.getElementById("recommend");
7     dom.value = "컴퓨터와 IT기술의 이해 [개정판-2판]";
8 </script>
```

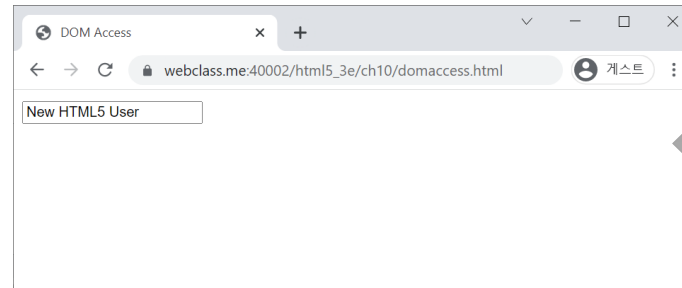
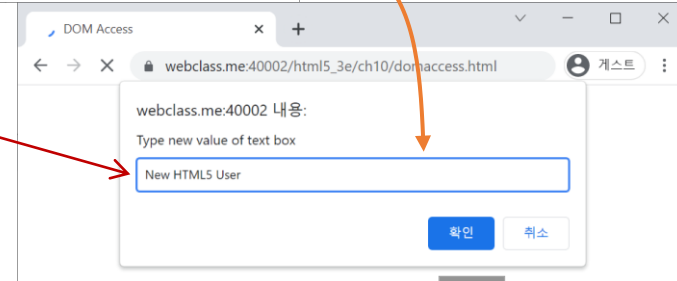


DOM 접근 방법 예제 (2)

```
1 <form action="">
2   <input id="username" type="text" value="Name of User" />
3 </form>
4
5 <script type="text/javascript">
6   var dom = document.getElementById("username");
7   alert(dom.value);
8   var newValue = prompt("Type new value of t
9   dom.value = newValue;
10 </script>
```

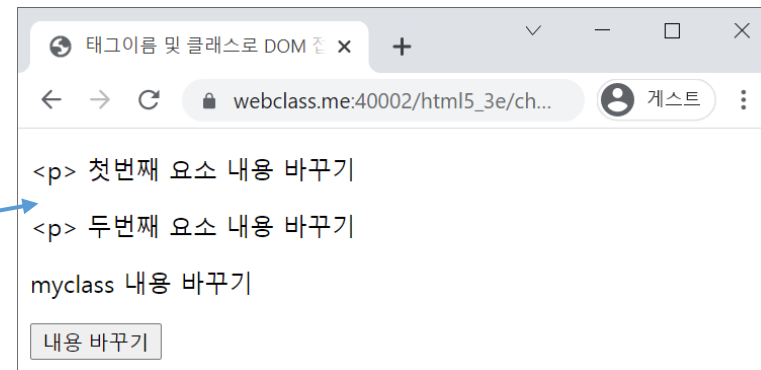
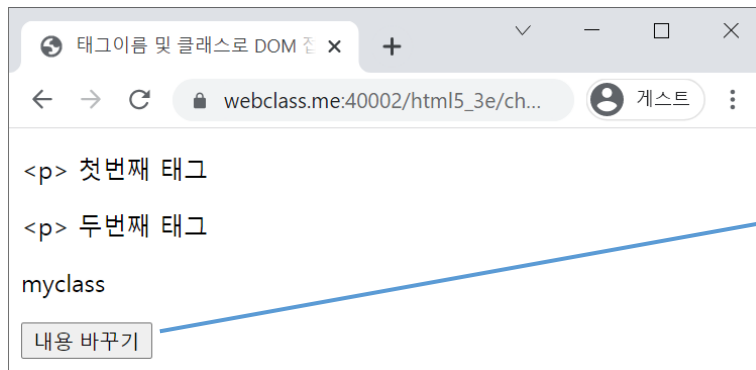


사용자 입력



태그 이름이나 클래스 이름으로 DOM에 접근하는 방법

```
1 <p> &lt;p&gt; 첫번째 태그 </p>
2 <p> &lt;p&gt; 두번째 태그 </p>
3 <p class="myclass"> myclass </p>
4 <input type="button" value="내용 바꾸기" onclick="changecontent();" />
5 <script type="text/javascript">
6     function changecontent() {
7         p_elements = document.getElementsByTagName("p")
8         myclass_elements = document.getElementsByClassName("myclass")
9         p_elements[0].innerText = "<p> 첫번째 요소 내용 바꾸기";
10        p_elements[1].innerText = "<p> 두번째 요소 내용 바꾸기";
11        myclass_elements[0].innerText = "myclass 내용 바꾸기"
12    }
13 </script>
```

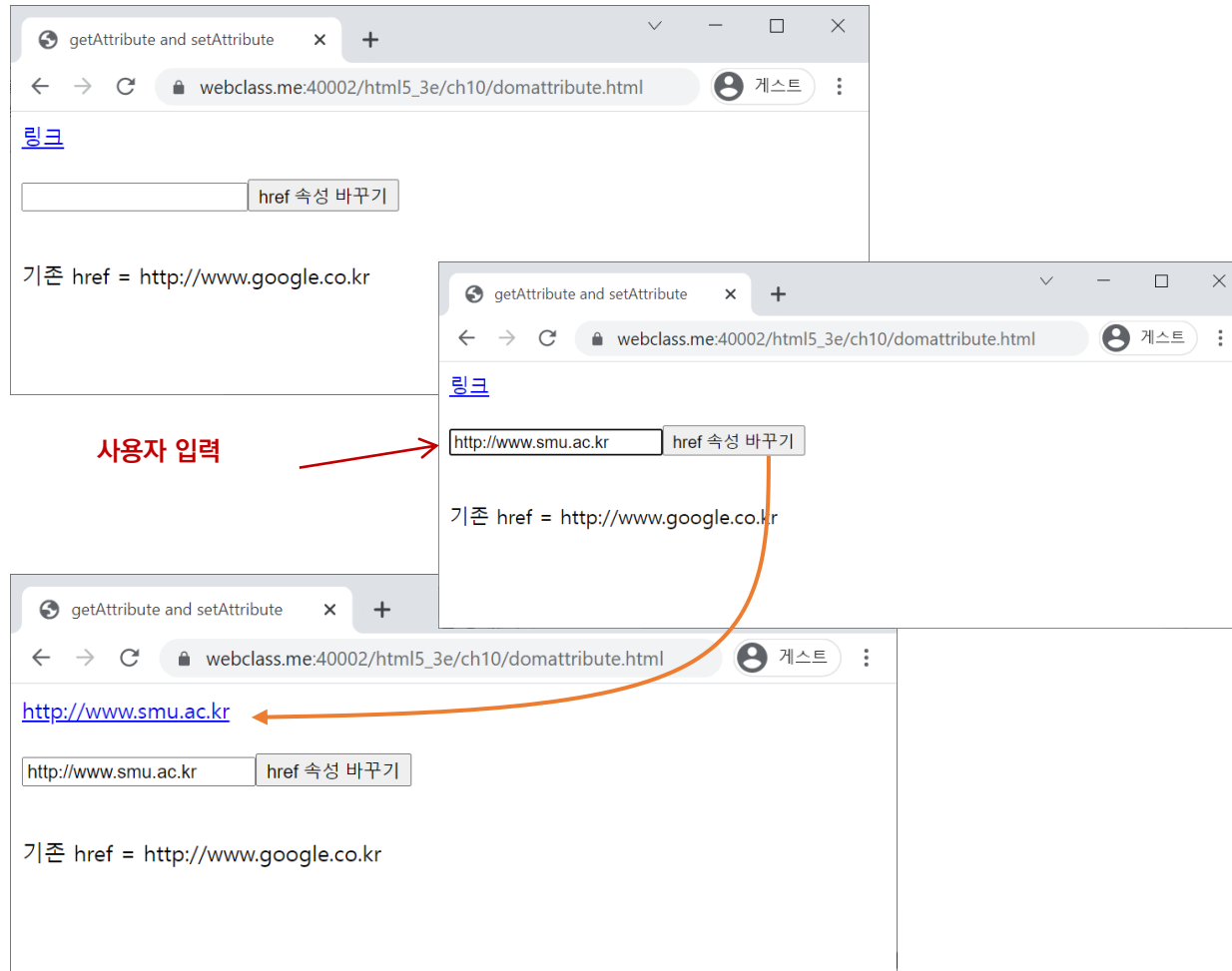


메소드를 이용한 DOM 속성 접근 방법

- `getAttribute()`: 속성 값을 읽어내는 메소드
- `setAttribute()`: 속성 값을 수정하는 메소드

```
1 <a id="link" href="http://www.google.co.kr"> 링크 </a>
2 <br/><br/>
3 <input type="text" id="text"></input><button id="btn">href 속성 바꾸기</button>
4 <br/><br/>
5 <p id = "curr_href"></p>
6
7 <script type="text/javascript">
8     function changehref() {
9         new_href = "https://" + document.getElementById("text").value;
10        document.getElementById("link").setAttribute("href", new_href);
11        document.getElementById("link").innerText = new_href;
12        console.log(document.getElementById("link").innerText)
13    }
14    document.getElementById("btn").onclick = changehref;
15    document.getElementById("curr_href").innerText =
16        "기존 href = " + document.getElementById("link").getAttribute("href");
17 </script>
18
19 </body>
20 </html>
```

메소드를 이용한 DOM 속성 접근 방법

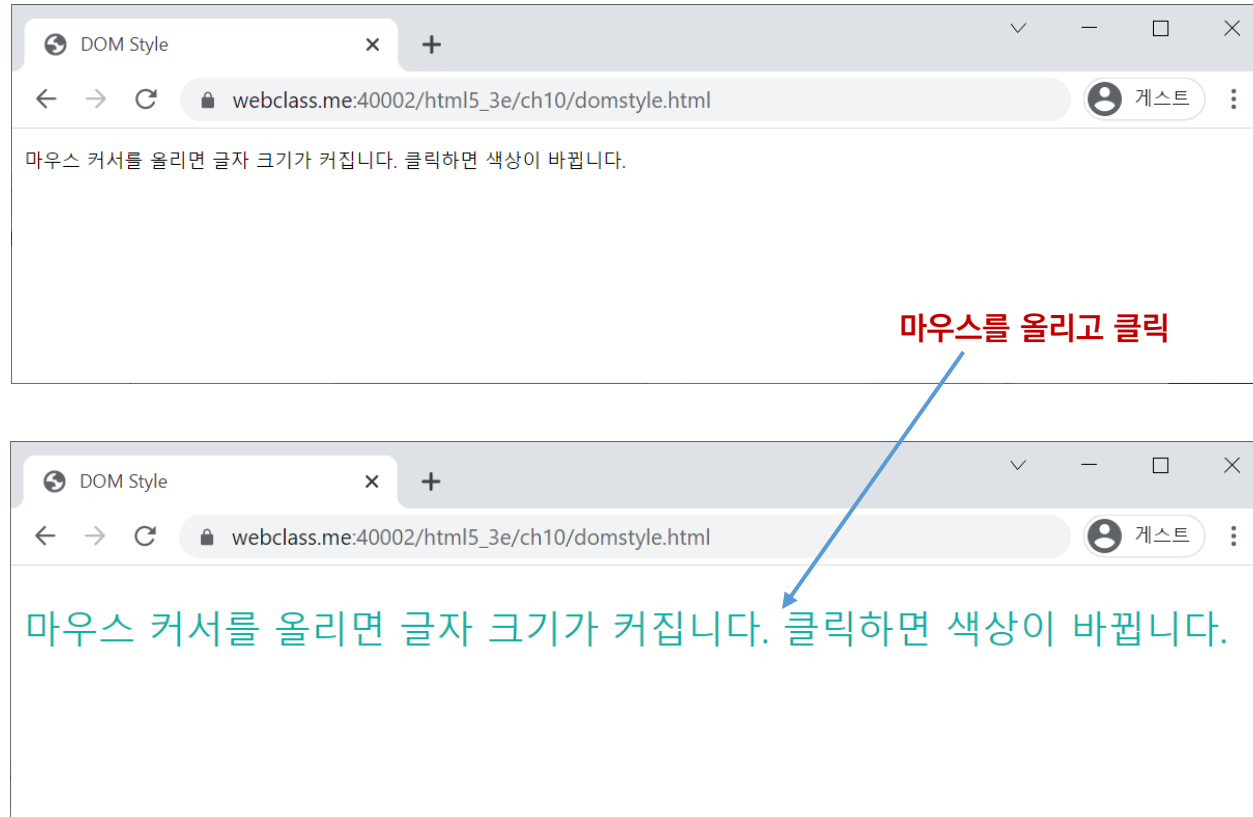


DOM을 통한 스타일 속성 접근 방법

- `dom.style.fontSize` 과 같이 스타일 속성에 접근 가능

```
1 <p id="text" style="font-size: 12px;"> 마우스 커서를 올리면 글자 크기가 커집니다. 클릭하면 색상이 바뀝니다.</p>
2
3 <script type="text/javascript">
4     function largefont() {
5         document.getElementById("text").style.fontSize = "24px"
6     }
7     function normalfont() {
8         document.getElementById("text").style.fontSize = "12px"
9     }
10    function changecolor() {
11        r = Math.round(Math.random() * 255);
12        g = Math.round(Math.random() * 255);
13        b = Math.round(Math.random() * 255);
14        document.getElementById("text").style.color = "rgb(" + r + "," + g + "," + b + ")";
15    }
16    document.getElementById("text").onmouseover = largefont;
17    document.getElementById("text").onmouseout = normalfont;
18    document.getElementById("text").onclick = changecolor;
19 </script>
```


DOM을 통한 스타일 속성 접근 방법



자바스크립트를 이용한 DOM 생성, 수정 및 삭제

자바스크립트를 이용한 DOM의 생성, 수정 및 삭제

- createElement(): HTML 요소를 생성해 내는 메소드

```
new_div_element= document.createElement("div");
```

```
document.getElementById("my_area").appendChild(new_div_element);
```

- removeChild(): 자식 요소를 삭제하는 메소드
- replaceChild(): 자식 요소를 교체하는 메소드
- firstChild: 자식 요소들 중 첫번째 요소를 가리키는 속성 값

자바스크립트를 이용한 DOM의 생성, 수정 및 삭제

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title> DOM 생성 및 삭제 예제 </title>
5 <script type="text/javascript">
6   function button_append() {
7     btn = document.createElement("button");
8     btn.innerText = document.getElementById("button_name").value;
9     btn.onclick = remove_self;
10    document.getElementById("button_area").appendChild(btn);
11  }
12
13  function delete_first() {
14    first_child = document.getElementById("button_area").firstChild;
15    if (first_child) {
16      document.getElementById("button_area").removeChild(first_child);
17    }
18  }
19
20  function remove_self(event) {
21    btn = event.target;
22    document.getElementById("button_area").removeChild(btn);
23  }
24 </script>
25 </head>
26 <body>
27   <h3> DOM 생성 및 삭제 예제 </h3>
28   생성된 버튼 클릭 시 삭제됩니다.
29   <br/>
30   <input id="button_name" type="text"><button onclick="button_append();"> 버튼 요소 생성 </button>
31   <button onclick="delete_first();"> firstChild 요소 삭제 </button>
32   <div id="button_area" style="border: solid; height: 300px;"></div>
33 </body>
34 </html>
```

자바스크립트를 이용한 DOM의 생성, 수정 및 삭제

