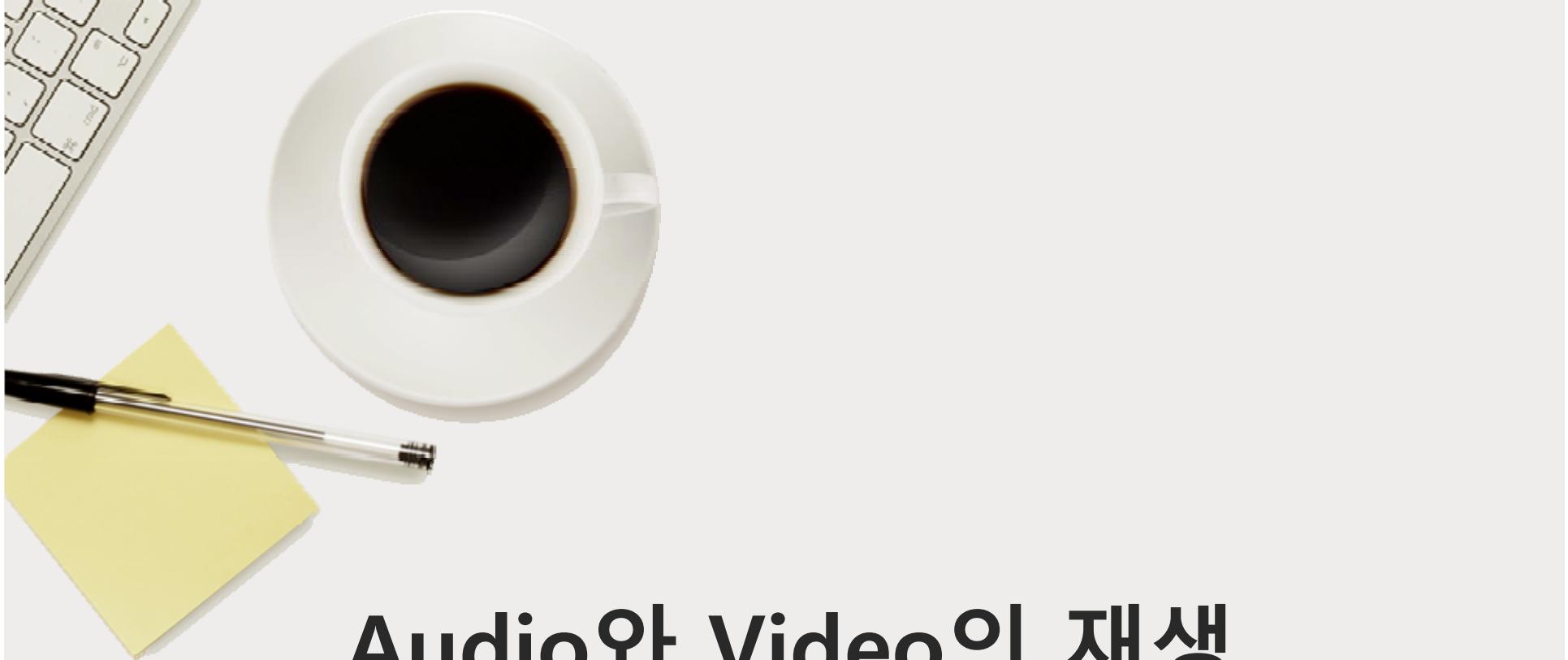


Chapter 9

Audio & Video



Audio와 Video의 재생



▣ 기존 HTML4에서의 오디오와 비디오 처리

- **embed** 요소나 **object** 요소를 사용
- 외부 plug-in인 플래시나 silverlight 등을 불러와 사용하였음
 - 다양한 plug-in이 존재하기 때문에 개발자는 plug-in 선택에 어려움

▣ HTML5에서의 오디오와 비디오 처리

- **오디오의 경우**
 - 외부 플러그인 없이 오디오를 재생할 수 있음
- **비디오의 경우**
 - 비디오 플레이를 위해서는 코덱(codec)이 필요함
 - 초기 HTML5에서는 코덱 표준화를 시도했지만 코덱이 표준화되지 못함
- **가장 많이 사용되는 H264 코덱은 특허에 따른 로열티 문제를 안고 있음**
 - 대안으로 오픈 코덱이 제시되었으나 현재까지 결정된 사항은 없음





■ <audio> 요소

- 웹 페이지에 오디오 재생 플레이어를 삽입하는 기능 수행

```
<audio src="오디오파일의 주소">  
    오디오 요소의 사용이 불가능한 브라우저의 경우 출력할 내용  
</audio>
```

- **src 속성**
 - 재생하고자 하는 오디오 파일의 URL이나 파일의 경로를 지정
- **브라우저가 <audio> 요소를 지원하지 못할 경우**
 - 현재의 브라우저가 <audio> 요소를 지원하지 않는다는 이미지나 텍스트 출력

```
<audio src="audio_sample.ogg">  
    현재의 브라우저는 audio 요소를 지원하지 않습니다.  
</audio>
```





▣ <audio> 요소의 주요 속성

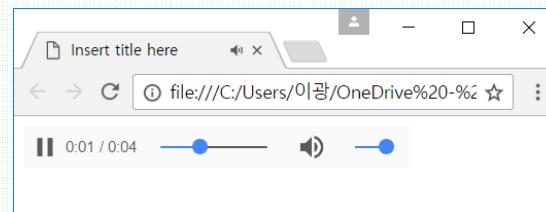
속성	설명
src	재생할 오디오 파일의 경로를 나타냄
preload	웹 문서 로딩시에 오디오의 로딩 상태를 나타냄. 다음과 같이 3가지의 값을 지정할 수 있음 (기본 값은 auto) - none : 오디오를 재생하기 위해 미리 다운로드 하지 않음 - metadata : 오디오에 대한 기본적인 정보들을(크기, 첫프레임, 오디오 길이 등) 가져옴 - auto : 오디오를 자동으로 재생하기 위해 다운로드를 수행
autoplay	웹 문서 로딩시에 오디오를 자동으로 재생할지를 지정
loop	재생할 오디오의 반복 여부를 나타낸다.
controls	기본으로 제공되는 재생 컨트롤 집합을 표시할지를 나타낸다. 일반적으로 표시되는 컨트롤은 재생, 일시 정지, 검색 및 볼륨 설정이 포함된다.





■ <audio> 요소의 예

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Insert title here</title>
</head>
<body>
    <audio src="source/audio_sample.ogg" controls="controls" autoplay="autoplay" loop="loop">
        <p>이 브라우저는 audio 요소를 지원하지 않습니다.</p>
    </audio>
</body>
</html>
```



- 각 속성의 값은 별도로 지정하지 않아도 정상적으로 동작함

```
<audio src="audio_sample.ogg" controls="controls" autoplay="autoplay" loop="loop">  
= <audio src="audio sample.ogg" controls autoplay loop>
```





■ <audio>요소 내에 <source> 요소의 사용

- **브라우저마다 지원하는 오디오 파일의 포맷이 다를 수 있음**
 - <source> 요소를 사용해 하나 이상의 오디오 미디어를 지정할 수 있음
 - type 속성에 MIME 타입을 지정
- **다양한 오디오 파일을 한번에 지정하는 것은 아님**
 - 브라우저에 따라 지원하는 오디오 형식이 다르므로 동일한 오디오를 다양한 형태의 파일로 작성해 지정하는 데 사용됨

```
<source src="오디오파일이름" type="MIME type">
```

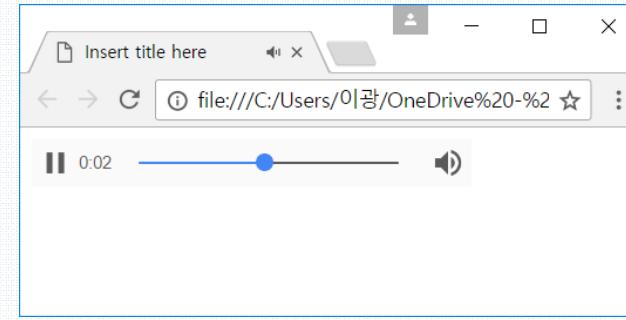
속성	설명
src	재생할 오디오 파일의 URL이나 경로를 지정
type	해당 오디오 파일의 MIME 타입 지정





□ <source> 요소의 예

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Insert title here</title>
</head>
<body>
    <audio controls autoplay loop>
        <source src="source/audio_sample.mp3" type="audio/mpeg">
        <source src="source/audio_sample.ogg" type="audio/ogg">
    </audio>
</body>
</html>
```





■ <video> 요소

- 웹 페이지에 비디오 재생 플레이어를 삽입하는 기능 수행

```
<video src="비디오파일의 주소">  
    비디오 요소의 사용이 불가능한 브라우저의 경우 출력할 내용  
</video>
```

- **src 속성**
 - 재생하고자 하는 비디오 파일의 URL이나 파일의 경로를 지정
- **브라우저가 <video> 요소를 지원하지 못할 경우**
 - 현재의 브라우저가 <video> 요소를 지원하지 않는다는 이미지나 텍스트 출력

```
<video src="bunny_small_ogg.ogg">  
    현재의 브라우저는 video 요소를 지원하지 않습니다.  
</video>
```





■ <video> 요소의 주요 속성

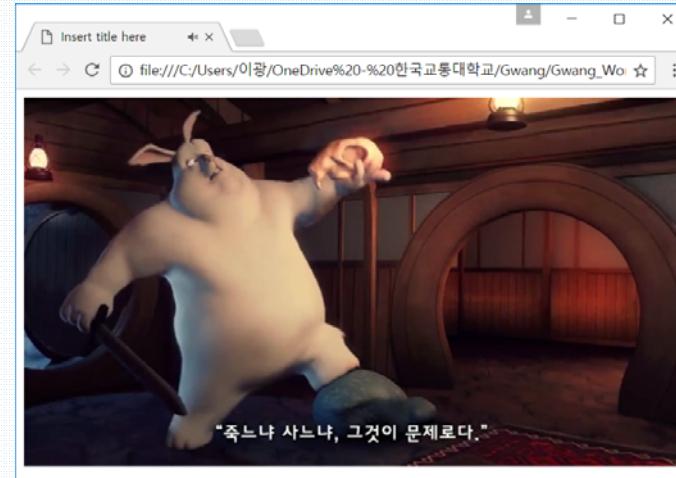
속성	설명
src	재생할 비디오 파일을 URL이나 경로를 지정
poster	비디오 데이터가 없는 경우, 브라우저에서 보여줄 수 있는 포스터 이미지 파일을 지정 - 지정한 비디오 데이터가 설정되어 있지 않거나 콘텐츠를 로드하는 중이어서 동영상을 사용할 수 없을 경우에만 포스터 이미지가 표시
preload	웹 문서 로딩 시에 비디오 데이터의 로딩 상태를 다음과 같이 3가지의 값으로 지정함 - none : 비디오를 재생하기 위해 미리 다운로드 하지 않음 - metadata : 비디오에 대한 기본적인 정보들을(크기, 첫 프레임, 비디오 길이 등) 가져옴 - auto : 기본 값으로, 비디오를 자동으로 재생하기 위해 다운로드
autoplay	웹 문서 로딩시에 비디오를 자동으로 재생할지를 지정
loop	재생할 비디오의 반복 여부를 지정
controls	기본으로 제공되는 재생 컨트롤 집합을 표시할지를 지정 - 일반적으로 표시되는 컨트롤은 재생, 일시 정지, 검색 및 볼륨 설정이 포함됨
width	비디오의 폭을 지정
height	비디오의 높이를 지정





■ <video> 요소의 예

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Insert title here</title>
</head>
<body>
    <video src="source/bunny_sd.mp4" controls autoplay>
        <p>이 브라우저는 video 요소를 지원하지 않습니다.</p>
    </video>
</body>
</html>
```





■ 코덱(codec) 지정

- **코더(coder) + 디코더(decoder)**
 - 영상 또는 음성 등의 아날로그 신호를 디지털 방식으로 변환하는 코더(Coder)와 디지털 신호를 영상이나 음성으로 바꿔주는 디코더(Decoder)의 합성어
- **동영상 재생을 위해서는 음성 코덱과 영상 코덱이 필요함**
 - [ogg 컨테이너의 예]

컨테이너 포맷 : ogg

영상 데이터
코덱 : Theora

음성 데이터
코덱 : Vorbis

컨테이너	포맷	파일 확장자	MIME 타입	코덱 문자열
MP3	MP3	.mp3	audio/mpeg	mp3
MP4	AAC, AAC+	.mp4, .m4a, .aac	audio/mp4	mp4a.40.5
OGA/OGG	Ogg Vorbis	.oga, .ogg	audio/ogg	vorbis
WAV	PCM	.wav	audio/wav	1





■ <video>요소 내에 <source> 요소의 사용

- <source> 요소의 type 속성에 값(컨테이너) 다음에 코덱을 지정
 - type 속성과 codec은 세미콜론으로 구분

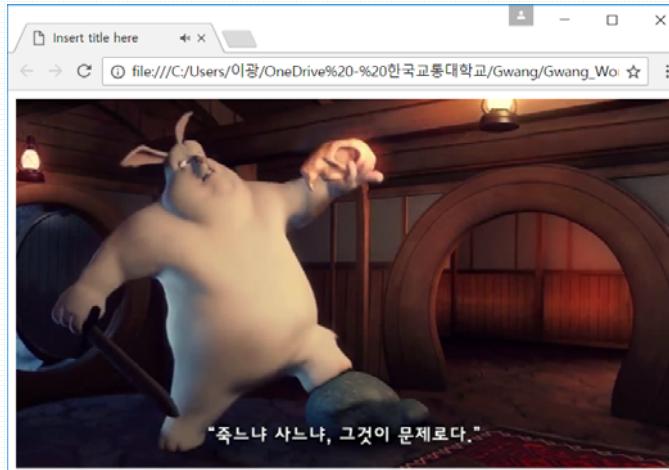
```
<source src="a.ogg" type="video/ogg" ; codec="theora, Vorbis"></source>
```

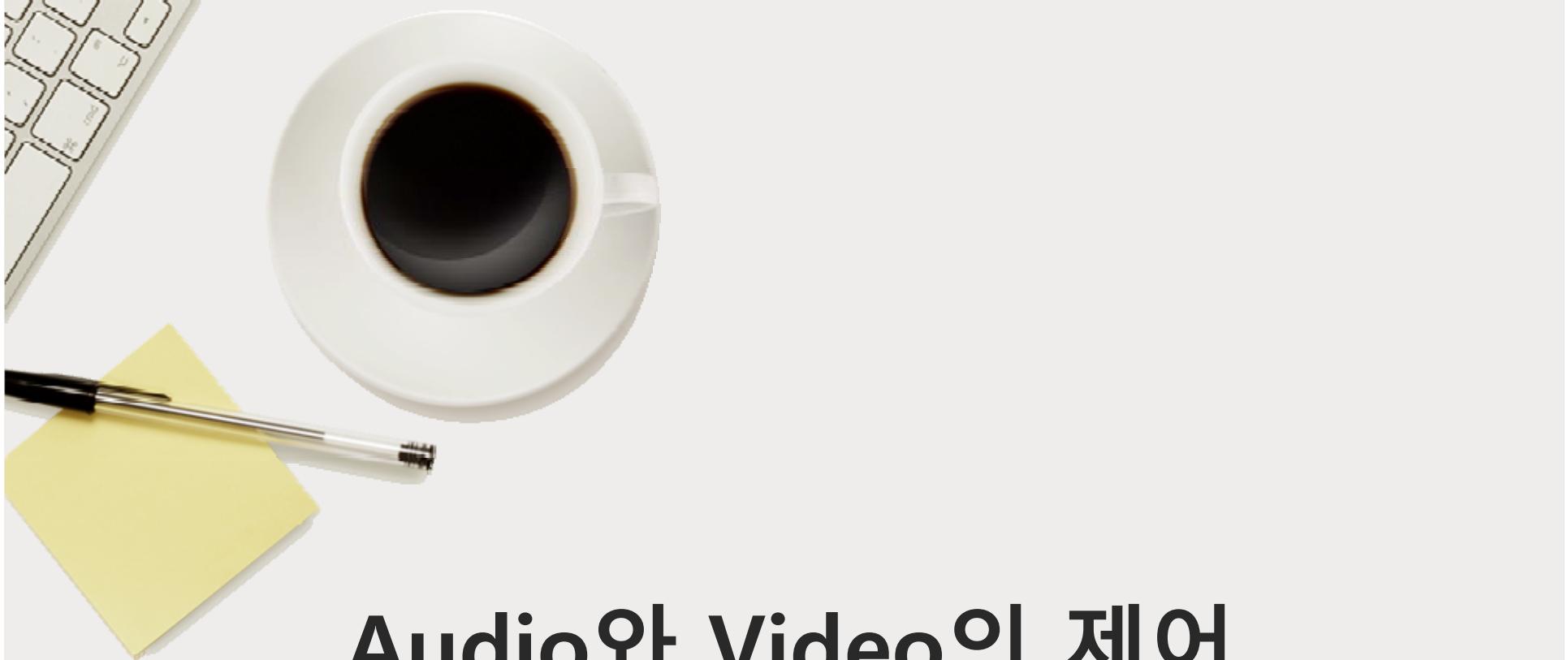




■ <source> 요소의 예

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Insert title here</title>
</head>
<body>
    <video controls autoplay width="600" height="400">
        <source src="source/bunny_sd.mp4" type="video/mp4; codecs="avc1.4D401E, mp4a40.2""></source>
        <source src="source/bunny_sd.ogg" type="video/ogg; codecs="theora, vorbis""></source>
    </video>
</body>
</html>
```





Audio와 Video의 제어



▣ **getElementsByName(태그)[[인덱스]]** 메서드

- **HTML 태그의 이름을 사용해 참조하는 방식**
 - 지정한 태그가 나타나는 순서에 따라 인덱스로 표현 가능

- **[예]**

- `document.getElementsByName('p')`
 - : HTML 문서 내에 존재하는 <p> 태그를 의미
 - `document.getElementsByName('p')[0]`
 - : HTML 문서 내에 존재하는 <p> 태그들 중 첫 번째 <p> 태그를 의미

▣ **getElementsById(아이디)** 메서드

- **HTML 문서 내에서 id 속성에 의해 지정된 부분을 참조하는 방식**

- **[예]**

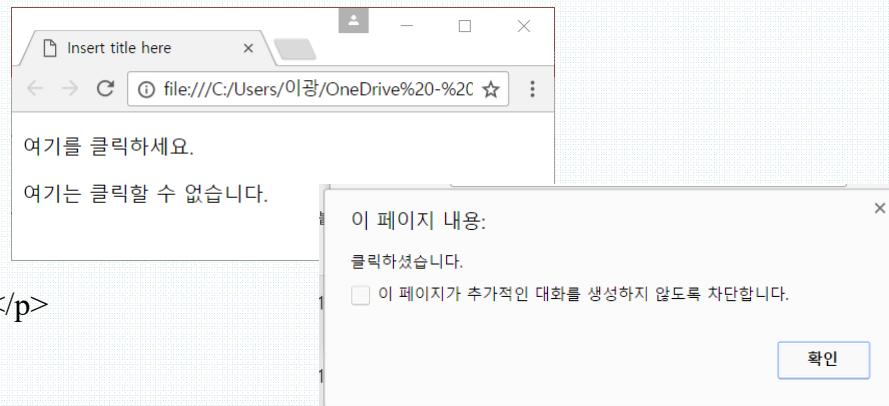
- `document.getElementById('main')`
 - : HTML 문서 내에서 id 속성의 값이 main으로 지정된 부분을 의미





▣ getElementsByTagName() 메서드의 예

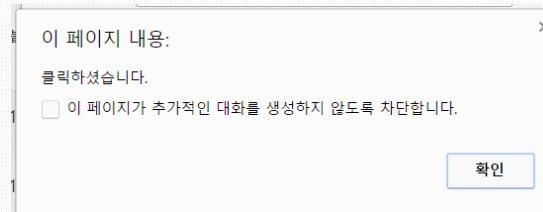
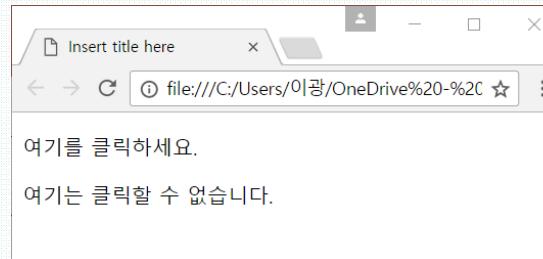
```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Insert title here</title>
    <script>
        function showalert() {
            alert('클릭하셨습니다.');
        }
        function clickme() {
            document.getElementsByTagName('p')[0].onclick=showalert;
        }
        window.onload=clickme;
    </script>
</head>
<body>
    <div id="main">
        <p>여기클릭하세요.</p>
        <p>여기는 클릭할 수 없습니다.</p>
    </div>
</body>
</html>
```





▣ getElementsById() 메서드의 예

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Insert title here</title>
    <script>
        function showalert() {
            alert('클릭하셨습니다.');
        }
        function clickme() {
            document.getElementById('p2').onclick=showalert;
        }
        window.onload=clickme;
    </script>
</head>
<body>
    <div id="main">
        <p id="p2">여기지를 클릭하세요.</p>
        <p>여기는 클릭할 수 없습니다</p>
    </div>
</body>
</html>
```





■ **querySelector()**

- **HTML5에서 추가된 메서드**
- **인자로 가지는 선택자 그룹과 일치하는 첫 번째 요소를 리턴**
- **[예]**

- `document.querySelector("#main p")`
: HTML 문서의 main 선택자 그룹에서 첫번째 <p> 요소를 리턴

■ **querySelectorAll() 메서드**

- **HTML5에서 추가된 메서드**
- **인자로 가지는 선택자 그룹과 일치하는 모든 요소를 리턴**
 - 리턴된 요소들은 HTML 문서에 나타나는 순서에 따라 지정한 배열에 반환됨

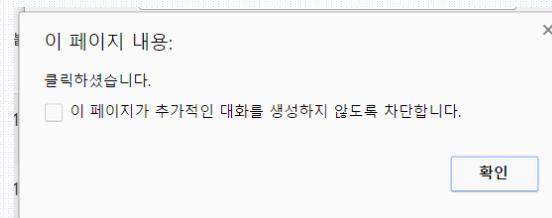
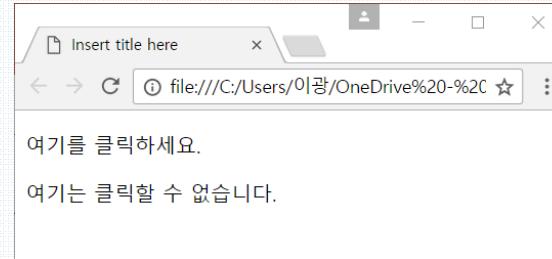
- **[예]**
 - `var list = document.querySelectorAll("#main p")`
: HTML 문서의 main 선택자 그룹에서 <p> 요소들이 list[] 라는 배열에 반환됨
: list[0]은 첫 번째 <p> 요소를 의미함





▣ querySelector() 메서드의 예

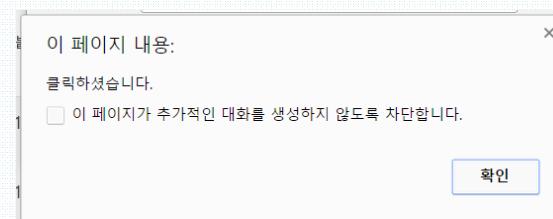
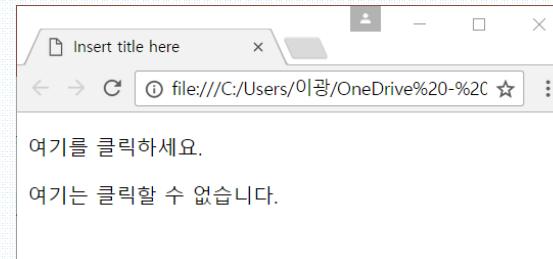
```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Insert title here</title>
    <script>
        function showalert() {
            alert('클릭하셨습니다.');
        }
        function clickme() {
            document.querySelector("#main p").onclick=showalert;
        }
        window.onload=clickme;
    </script>
</head>
<body>
    <div id="main">
        <p id="p2">여기를 클릭하세요.</p>
        <p>여기는 클릭할 수 없습니다</p>
    </div>
</body>
</html>
```





▣ querySelectorAll() 메서드의 예 - 1

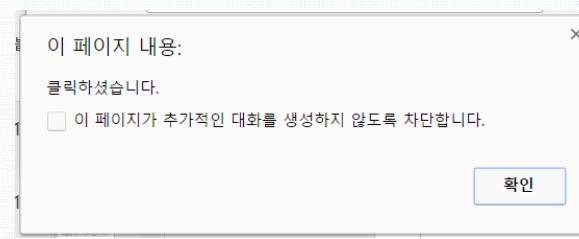
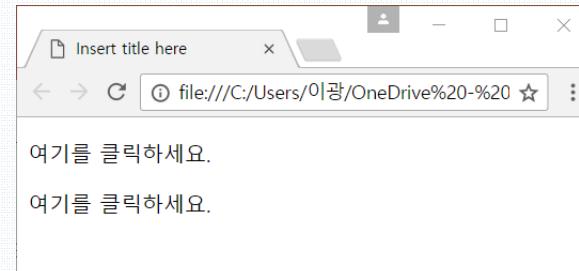
```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Insert title here</title>
    <script>
        function showalert() {
            alert('클릭하셨습니다.');
        }
        function clickme() {
            var list = document.querySelectorAll("#main p");
            list[0].onclick=showalert;
        }
        window.onload=clickme;
    </script>
</head>
<body>
    <div id="main">
        <p id="p2">여기지를 클릭하세요.</p>
        <p>여기는 클릭할 수 없습니다</p>
    </div>
</body>
</html>
```





▣ querySelectAll() 메서드의 예 - 2

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Insert title here</title>
    <script>
        function showalert() {
            alert('클릭하셨습니다.');
        }
        function clickme() {
            var list = document.querySelectorAll("#main p");
            for(var i=0; i<list.length; i++) {
                list[i].onclick=showalert
            }
        }
        window.onload=clickme;
    </script>
</head>
<body>
    <div id="main">
        <p>여기지를 클릭하세요.</p>
        <p>여기지를 클릭하세요.</p>
    </div>
</body>
</html>
```





▣ addEventListener() 메서드

- 특정 요소에 이벤트를 등록하는 기능을 수행

```
object.addEventListener('이벤트', 이벤트함수, 처리방식)
```

```
window.addEventListener('click', clickme, false)
```

- 이벤트**
 - 웹페이지에서 발생되는 이벤트(on이 붙지 않음에 유의)
- 이벤트함수**
 - 해당 이벤트가 발생했을 때 수행되는 함수
- 처리방식**
 - 이벤트 처리방식
 - true : 캡처링(Capturing)
 - false : 버블링(Bubbling) (주로 사용됨)





▣ **removeEventListener()** 메서드

- 특정 요소에 등록된 이벤트를 해제하는 기능 수행

```
object.removeEventListener( [이벤트], 이벤트함수)
```

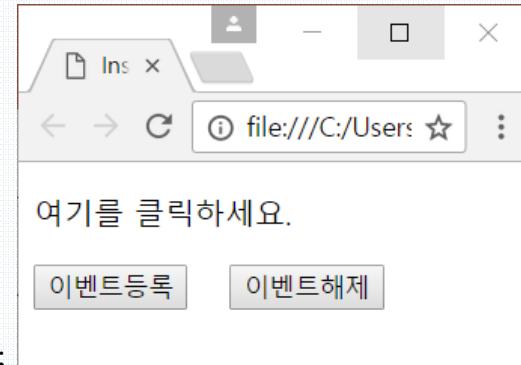
```
window.removeEventListener('click', clickme)
```





▣ 이벤트 등록과 해제의 예

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <script>
        function showalert() {
            alert('클릭하셨습니다.');
        }
        function add_event() {
            var clickobj = document.getElementsByTagName('p')[0];
            clickobj.addEventListener('click', showalert, false);
        }
        function remove_event() {
            var clickobj = document.getElementsByTagName('p')[0];
            clickobj.removeEventListener('click', showalert);
        }
    </script>
</head>
<body>
    <div id="main">
        <p>여기를 클릭하세요.</p>
        <input type="button" value="이벤트등록" onClick="add_event()"> &ampnbsp&ampnbsp
        <input type="button" value="이벤트해제" onClick="remove_event()">
    </div>
</body>
</html>
```





▣ 자바스크립트 API를 사용하여 오디오/비디오 제어 가능

- 오디오와 비디오를 제어하기 위한 자바스크립트 메서드는 같음
- <audio>와 <video> 요소는 단순히 태그를 이용하여 미디어를 재생하는 기능을 수행할 뿐임
 - 자바스크립트 API를 사용해 오디오와 비디오를 정밀하게 제어 가능
 - 제어를 위한 메서드와 다수의 속성이 존재함
- 오디오/비디오 제어를 위한 주요 메서드

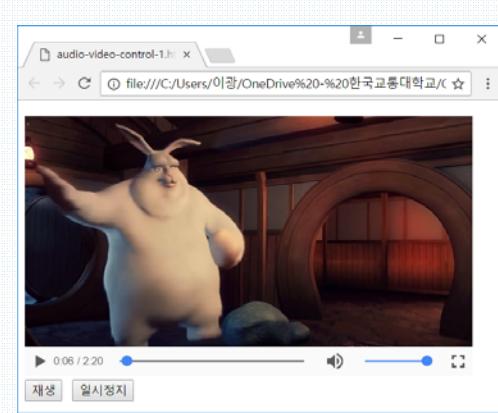
메서드	설명
load()	미디어 파일을 다시 읽어드림 (재생하지는 않음)
play()	미디어 파일을 재생
pause()	미디어 파일의 재생을 일시 정지함





▣ 오디오와 비디오의 예 - 1

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <script type="text/javascript">
        function play_media()
        {
            var videocontrol = document.getElementById("videoplay");
            videocontrol.play();
        }
        function paused_media()
        {
            var videocontrol = document.getElementById("videoplay");
            videocontrol.pause();
        }
    </script>
</head>
<body>
    <video id="videoplay" src="source/bunny_sd.mp4" controls width="500" height="300">
        <p>이 브라우저는 video 요소를 지원하지 않습니다.</p>
    </video>
    <br>
    <input type="button" value="재생" onclick="play_media();">&nbsp;
    <input type="button" value="일시정지" onclick="paused_media();">
</body>
</html>
```





▣ 오디오와 비디오 제어를 위한 주요 속성

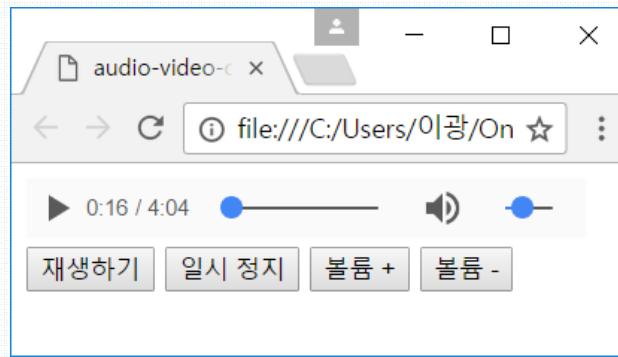
메서드	설명
src	src 속성 값을 의미
currentSrc	현재 읽어 들여 재생중인 미디어 데이터의 URL
currentTime	현재 재생 위치 (초 단위)
startTime	재생할 위치 (초 단위)
duration	미디어 데이터의 길이 (초 단위)
paused	일시 정지인지 아닌지를 반환 (일시 정지 시 true)
defaultPlaybackRate	기본 재생 속도 (기본 값은 1.0)
playbackRate	재생 속도 (기본:1.0, 0이상:앞으로 재생, 음수:뒤로 재생)
ended	재생이 종료되었는지를 반환 (종료 시 true)
muted	음 소거 상태인지 반환 (음 소거 상태 시 true)
volumen	음량 (0~1.0까지 값을 가짐, 1.0이 최대)





▣ 오디오와 비디오의 예 - 2

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
</head>
<body>
    <audio id="player" src="source/mylove.mp3" controls>
        <p>이 브라우저는 audio 요소를 지원하지 않습니다.</p>
    </audio>
    <br>
    <input type=button value="재생하기" onclick="document.getElementById('player').play()">
    <input type=button value="일시 정지" onclick="document.getElementById('player').pause()">
    <input type=button value="볼륨 +" onclick="document.getElementById('player').volume += 0.1">
    <input type=button value="볼륨 -" onclick="document.getElementById('player').volume -= 0.1">
    <br>
</body>
</html>
```





▣ 오디오와 비디오의 예 - 3

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">

<script type="text/javascript">
    var myVideo;
    function initEvents() {
        myVideo = document.getElementById('player');
        btn = document.getElementById('btn_play');
    }

    function playVideo() {
        if (myVideo.paused) {
            myVideo.play(); btn.value = "일시 멈춤";
        } else {
            myVideo.pause(); btn.value = "재생 하기";
        }
    }

    function restartVideo() {
        myVideo.currentTime = 0;
    }

```



오디오와 비디오의 제어



```
function volumeUp() {  
    myVideo.volume += 0.1;  
}  
  
function volumeDown() {  
    myVideo.volume -= 0.1;  
}  
  
function increaseSpeed() {  
    myVideo.playbackRate += 0.1;  
}  
  
function decreaseSpeed() {  
    myVideo.playbackRate -= 0.1;  
}  
  
function defaultSpeed() {  
    myVideo.playbackRate = 1;  
}  
  
window.addEventListener("DOMContentLoaded", initEvents, false);  
  
    </script>  
</head>
```



오디오와 비디오의 제어



```
<body>
  <video width="600" controls id="player" src="source/bunny_sd.mp4">
    <p>이 브라우저는 비디오 요소를 지원하지 않습니다.</p>
  </video>
  <br>
  <input type=button id="btn_play" value="재생하기" onClick="playVideo()">
  <input type=button id="btn_restart" value="다시재생" onClick="restartVideo()">
  <input type=button id="btn_volup" value="볼륨증가" onClick="volumnUp()">
  <input type=button id="btn_voldown" value="볼륨감소" onClick="volumnDown()">
  <input type=button id="btn_incspeed" value="속도증가" onClick="increaseSpeed()">
  <input type=button id="btn_decspeed" value="속도감소" onClick="decreaseSpeed()">
  <input type=button id="btn_defspeed" value="기본속도" onClick="defaultSpeed()">
</body>

</html>
```





▣ 오디오와 비디오 관련 이벤트 (미디어 로드 관련)

이벤트	설명
loadstart	미디어 데이터 요청이 시작됨
progress	미디어 데이터 로딩 중 지속적으로 발생
loadedmetadata	메타데이터(재생시간, 동영상의 너비와 높이 등)가 로드되었을 때 발생
loadeddata	브라우저가 처음으로 미디어 데이터를 표시할 수 있을 때 발생
canplay	미디어 재생을 시작할 수 있는 충분한 데이터가 로드되었을 때 (추가적인 버퍼링 데이터가 요구될 수 있음)
canplaythrough	추가적인 데이터를 버퍼링하기 위해 재생을 일시정지 시키지 않고도 전체 미디어를 재생 할 수 있을 만큼 충분한 데이터가 로드되었을 때 발생
stalled	브라우저가 데이터를 요청하지만 예상치 않은 이유로 데이터를 로드할 수 없을 때 발생 (에러 때문이 아님)
suspend	브라우저에서 의도적으로 데이터 로드를 일시중지 시킬 때 발생 (에러 때문이 아님)
abort	사용자에 의해 미디어 데이터의 로드가 중지되었을 때 발생
error	에러에 의해 데이터 로드가 중지되었을 때 발생
loadend	데이터 로딩 완료 시 발생 (load, abort, error 이후에 모두 발생)





▣ 오디오와 비디오 관련 이벤트 (미디어 재생 관련)

이벤트	설명
play	미디어 데이터의 재생이 시작되었을 때 발생
playing	미디어 데이터가 재생되는 경우 발생 (Play 이벤트 발생 후 미디어 데이터 충분히 로드되었을 경우 발생)
timeupdate	미디어 데이터가 재생되는 동안 currentTime 프로퍼티가 변경될 때마다 일어날 때마다 발생
waiting	play 이벤트 발생 후 미디어 데이터 충분히 로드되지 않았을 경우 발생
ended	미디어 데이터의 재생이 종료되었을 때 발생





▣ 오디오와 비디오의 예 - 4

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
</head>

<body>
    <video id="videoplay" controls width="500" height="300">
        <source src="source/bunny_sd.mp4" type="video/mp4; codecs="avc1.4D401E, mp4a40.2""></source>
    </video>
    <br>
    <div id="playtime"></div>

    <script type="text/javascript">
        var videocontrol = document.getElementById("videoplay");

        videocontrol.addEventListener("timeupdate", PlayTime, false);
        videocontrol.addEventListener("ended", PlayEnd, false);

        function PlayTime(e)
        {
            document.getElementById("playtime").innerHTML
            = "재생 상태 : " + Math.floor(videocontrol.currentTime) + "/" + Math.floor(videocontrol.duration);
        }
    </script>
</body>
```

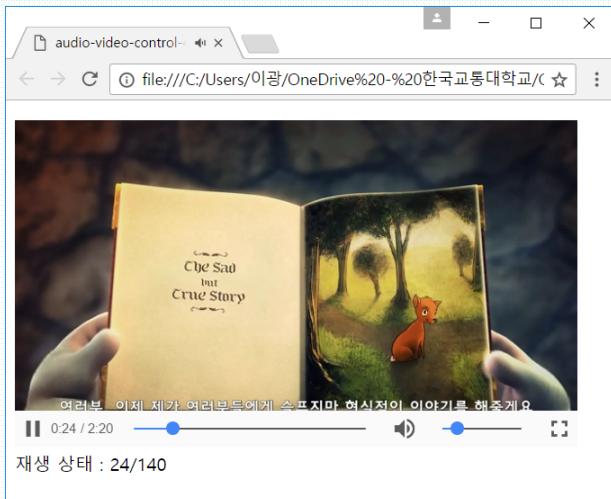


오디오와 비디오의 제어



```
function PlayEnd(e)
{
    alert("미디어 재생이 완료되었습니다.");
}
</script>

</body>
</html>
```



이 페이지 내용:

미디어 재생이 완료되었습니다.

확인

