

JavaScript

Video & Audio – 김근형 강사

Video & Audio

- Video & Audio
 - html5 에서 비디오 파일을 재생하는 video 요소, 오디오 파일을 재생하는 audio 요소가 새로 추가됨.
 - 새로운 요소를 알기 위해서 video, audio, source 요소의 마크업을 알 필요가 있다.

Video & Audio

○ Video 속성과 프로퍼티

속성/프로퍼티	설명
src	비디오 파일의 URL을 지정한다
poster	비디오 재생을 하기 전에 표시할 이미지 파일을 지정한다.
preload	사용자가 비디오를 볼 가능성을 유저 에이전트에 알려준다. 유저 에이전트는 이 정보를 참고로 미디어 데이터를 어느 정도 받아들이지 결정한다. 지정 가능한 값은 "none", "metadata", "auto" 이다. "none"은 비디오가 재생 가능해지기 전까지 아무것도 내려받지 않도록 한다. "metadata"는 비디오 파일의 메타 정보가 담긴 부분까지만 미리 내려받는다. "auto"는 미리 모든 부분을 내려 받는다. html5에서 preload 속성 값은 브라우저에 전달하는 참고 값이다. 반드시 지정한대로 브라우저가 동작하지는 않는다. autoplay 속성을 지정했다면 preload는 무시된다.
autoplay	비디오가 재생 가능해지면 즉시 재생하도록 지정하는 불리언 속성이다.
loop	비디오 재생이 끝나면 맨 앞으로 돌아가서 다시 재생하도록 지정하는 불리언 속성이다.
controls	비디오의 재생이나 장치 등의 UI 컨트롤 표시를 지정하는 불리언 속성이다.
width	비디오의 가로 길이를 css 픽셀로 지정한다. width 속성은 문자열로 반환된다. 값을 지정하여 길이를 변경할 수 있다.
height	비디오의 세로 길이를 css 픽셀로 지정한다. height 속성은 문자열로 반환된다. 값을 지정하여 길이를 변경할 수 있다.
videoWidth	실제 영상의 가로 길이를 반환한다. 속성은 존재하지 않으며 프로퍼티 호출만이 가능하다. 값 지정 불가
videoHeight	실제 영상의 세로 길이를 반환한다. 속성은 존재하지 않으며 프로퍼티 호출만이 가능하다. 값 지정 불가

Video & Audio

○ 비디오 속성 예제 - 1

```
<p><video src="video.mp4" controls="controls"></video></p>  
<p><button>반복 재생 해제 상태</button></p>
```

```
<script type="text/javascript">  
  // video 요소 오브젝트  
  var video = document.querySelector('video');  
  // button 요소 오브젝트  
  var button = document.querySelector('button');  
  // button 요소에 click이벤트 리스너를 지정  
  button.addEventListener("click", function() {  
    // loop 프로퍼티의 값을 반전  
    video.loop = video.loop ? false : true;  
    // button 요소의 텍스트를 변경  
    if( video.loop == true ) {  
      button.textContent = "반복 재생 ";  
    } else {  
      button.textContent = "반복 재생 해제 상태 ";  
    }  
  }, false);  
</script>
```



반복 재생

Video & Audio

○ 비디오 속성 예제 - 2

```
// video 요소
var video = null;
// 페이지가 로드 되었을 때의 처리
window.addEventListener("load", function() {
    // video 요소 오브젝트
    video = document.querySelector('video');
    // video 요소가 클릭되었을 때의 처리
    window.addEventListener("click", toggle_size, false);
    // 창의 크기가 변경 되었을 때의 처리
    window.addEventListener("resize", centering, false);
    // video 요소를 한가운데로 이동
    centering();
    video.style.display = "block";
}, false);
```

```
// video 요소가 클릭되었을 때의 처리
function toggle_size() {
    if( video.height == video.videoHeight ) {
        // 창의 크기에 맞춤
        video.width = document.documentElement.clientWidth;
        video.height = document.documentElement.clientHeight;
        // video요소의 위치를 왼쪽 상단으로
        video.style.left = "0px";
        video.style.top = "0px";
    } else {
        // 원래 크기로 돌리기
        video.width = video.videoWidth;
        video.height = video.videoHeight;
        // video요소를 한가운데로 이동
        centering();
    }
}
// video 요소를 한가운데로 이동
function centering() {
    // 창 크기
    var sw = document.documentElement.clientWidth;
    var sh = document.documentElement.clientHeight;
    // video 요소의 크기
    var vw = video.width;
    var vh = video.height;
    // video 요소를 가운데로 이동
    video.style.left = ( (sw - vw) / 2 ) + "px";
    video.style.top = ( (sh - vh) / 2 ) + "px";
}
```

Video & Audio

○ Audio 속성과 프로퍼티

속성/프로퍼티	설명
src	오디오 파일의 URL을 지정한다
preload	사용자가 오디오를 볼 가능성을 유저 에이전트에 알려준다. 유저 에이전트는 이 정보를 참고로 미디어 데이터를 어느 정도 받아들이지 결정한다. 지정 가능한 값은 "none", "metadata", "auto" 이다. "none"은 오디오가 재생 가능해지기 전까지 아무것도 내려받지 않도록 한다. "metadata"는 오디오 파일의 메타 정보가 담긴 부분까지만 미리 내려받는다. "auto"는 미리 모든 부분을 내려 받는다. html5에서 preload 속성 값은 브라우저에 전달하는 참고 값이다. 반드시 지정한대로 브라우저가 동작하지는 않는다. autoplay 속성을 지정했다면 preload는 무시된다.
autoplay	오디오가 재생 가능해지면 즉시 재생하도록 지정하는 불리언 속성이다.
loop	오디오 재생이 끝나면 맨 앞으로 돌아가서 다시 재생하도록 지정하는 불리언 속성이다.
controls	오디오의 재생이나 장치 등의 UI 컨트롤 표시를 지정하는 불리언 속성이다.

코덱(Codec)

- 코덱과 컨테이너
 - 코덱 : 비디오나 오디오의 인코딩 방식을 말한다.
 - 컨테이너 : 실제 비디오나 오디오를 저장하는 형식
 - 각 브라우저마다 재생 가능한 코덱과 컨테이너가 정해져 있다.
 - 대표적인 코덱과 그에 대응하는 브라우저는 다음 장에 있다.

코덱(Codec)

- 비디오 형식과 대응 브라우저
- H.264(.mp4), Theora(.ogv), VP8+VP9(.webm)

Browser	Operating system	Latest stable release	Video formats supported			
			Theora	H.264	VP8 (WebM)	VP9 (WebM)
Android browser	Android	4.2.1 "Jelly Bean" (November 27, 2012; 5 months ago) [±][32][33]	2.3[34]	3.0[34]	2.3[34]	
Chromium	All supported	N/A	r18297[35]	Manual install[^{note 1}]	r47759[37]	r172738[38]
Google Chrome	All supported	26.0.1410.64 (April 9, 2013; 25 days ago) [±][39]	3.0[40][41]	3.0[41][^{note 2}]	6.0[43][44]	25[45]
Internet Explorer	Windows	10.0.4 [46] (April 9, 2013; 25 days ago) [±]	Manual install[^{note 3}]	9.0[48]	Manual install[^{note 4}]	
	Windows Phone	10.0 (November 21, 2012; 5 months ago) [±]	No	9.0[^{citation needed}]	No	
	Windows RT	10.0		10.0[^{citation needed}]		
Konqueror	All supported	4.10.2 (2 April 2013; 32 days ago) [±][51]	4.4[^{note 5}]			
Mozilla Firefox	Windows 7+	20.0.1 (April 11, 2013; 23 days ago) [±][53]	3.5[54]	21.0 (Beta)[^{note 6}]	4.0[57][58]	
	Windows Vista			22.0 (Aurora)[59]		
	Android			17.0[60]		
	All other supported			No		
Opera	All supported	12.15 (April 4, 2013; 30 days ago) [±]	10.50[61]	No	10.60[62][63]	
Safari	iOS	6.0.4 (April 16, 2013; 18 days ago) [±]	No	3.1[64][65]	No	
	MacOS X		Manual install[^{note 7}]		Manual install[66]	
Web (previously Epiphany)	All supported	3.8.1 (April 18, 2013; 16 days ago) [±][67]	2.28[^{note 8}]			

코덱(Codec)

- 오디오 형식과 대응 브라우저
 - AAC(.aac), MP3(.mp3), Vorbis(.ogg), WAV(.wav)

Browser ⇄	Operating system ⇄	Formats supported by different web browsers						
		Ogg Vorbis ⇄	WAV PCM ⇄	MP3 ⇄	AAC ⇄	WebM Vorbis ⇄	Ogg Opus ⇄	WebM Opus ⇄
Google Chrome	All supported	9	Yes	Yes	Yes	Yes	25 (since v31 in Windows)	Yes
Internet Explorer	Windows	No	No	9	9	No	No	No
Mozilla Firefox	All supported	3.5	3.5	Windows (21.0), Linux (24.0, needs a gstreamer codec), OS X (26.0)	Windows (21.0) and Linux (24.0, needs a gstreamer codec) only	4.0	15.0	28.0 ^[4]
Opera	All supported	10.50	11.00	14	14	10.60	14	Yes
Safari	OS X	Yes	3.1	3.1	3.1	No	No	No

MIME 형식

- MIME Type

- 비디오나 오디오를 재생할 때 어떤 타입을 재생할 지 결정하고자 할 경우 MIME 타입 설정할 수 있다.
- 다음 아래와 같은 타입들의 MIME 타입을 결정할 수 있다.

```
AddType video/mp4 .mp4
AddType video/ogg .ogv
AddType video/webm .webm
AddType audio/mp4 .acc
AddType audio/mpeg .mp3
AddType audio/ogg .ogg
AddType audio/wav .wav
```

- source 같은 복수의 파일을 등록해서 브라우저의 호환성을 최대화 할 경우 MIME Type이 쓰일 수 있다.

Source 요소

○ Source

- source 요소는 다양한 브라우저에서 비디오나 오디오를 재생할 수 있도록 하기 위한 요소
- 모든 요소가 동일한 코덱을 지원하는 것이 아니기 때문에 각기 다른 스타일로 인코딩한 데이터를 한꺼번에 넣어 해당 브라우저에서 지원하는 인코딩 타입만 실행하도록 하는 역할을 수행한다.
- source 요소에는 해당 파일이 어떤 타입을 지원하는지 명시하기 위해 MIME Type을 쓰도록 지원한다.
- source 요소의 type 속성은 필수는 아니지만, 없으면 미디어 리소스를 무조건 내려받아 불필요한 트래픽을 발생시키므로 반드시 지정할 것을 권장한다.

```
<audio controls="controls">  
  <source src="audio.mp3" type="audio/mpeg" />  
  <source src="audio.ogg" type="audio/ogg" />  
</audio>
```

Video & Audio 관련 속성/메서드

- MIME 형식으로부터 재생 가능 여부를 판단
 - source 요소에 미디어 리소스를 여러 개 지정하면 자동으로 브라우저가 재생 가능한 미디어 리소스를 선택한다.
 - 하지만 미디어 요소를 HTML에 마크업하지 않으면 스크립트로 어떠한 형식이 재생 가능한지를 판단할 필요가 있다. 이럴때는 `canPlayType()` 함수를 사용한다.

메서드 / 속성	설명
<code>media.canPlayType(type)</code>	매개변수(type)에 MIME 형식을 지정하면 미디어 요소(video 또는 audio 요소)가 재생할 수 있는 형식인지를 문자열로 반환한다. 재생할 수 없다면 공백 문자열을 그렇지 않다면 "maybe" 또는 "probably"를 반환한다.

- `canPlayType()` 함수로 MIME 형식에 따라 재생할 수 있는지 판별할 수 있지만 MIME 형식만으로는 확실한 재생 여부를 보증할 수 없다. 그래서 함수는 "maybe" 또는 "probably" 라는 모호한 결과를 반환한다.

Video & Audio

○ Audio 프로퍼티 예제 - 1

```
<meta charset="UTF-8">
<title>Document</title>
<script type="text/javascript">
window.addEventListener("load", function() { // audio 요소를 생성
if( ! window.Audio ) { return; }
var audio = new Audio();
// 재생할 수 있는 오디오 형식을 판별하여 src에 지정
if( audio.canPlayType('audio/mpeg') ) {
    audio.src = "audio.mp3";
} else if( audio.canPlayType('audio/ogg') ) {
    audio.src = "audio.ogg";
} else {
    return;
}
// button 요소
var button = document.querySelector('#play');
// button 요소에 click 이벤트 리스너를 지정
button.addEventListener("click", function() {
// 효과음을 재생
    audio.play();
}, false);
}, false);
</script>
</head>
<body>
    <p> <button id="play">Press! </button> </p>
```

Press!

Video & Audio 관련 속성/메서드

○ 선택된 파일 확인

- source 요소로 미디어 리소스를 여러 개 지정했을 때 브라우저에 따라 선택하는 파일이 다르다.
- 상황에 따라서는 스크립트로 어떤 파일을 선택했는지 판별해야 할 때도 있다.

메서드 / 속성	설명
media.currentSrc	선택한 미디어 파일의 URL을 반환한다. 선택한 파일이 없다면 공백 문자열을 반환한다. 값 지정 불가

- currentSrc 는 해당 파일 경로일 경우 file:// 혹은 http:// 의 전체 경로를 반환한다.

Video & Audio 관련 속성/메서드

○ 선택된 파일 확인

```
<audio controls="controls">
  <source src="audio.ogg" type="audio/ogg" />
  <source src="audio.mp3" type="audio/mpeg" />
</audio>
<p>재생 파일 : <span id="url">-</span></p>
<script type="text/javascript">
  window.addEventListener("load", function() {
    // audio 요소
    var audio = document.querySelector('audio');
    // 선택된 파일의 URL
    var url = audio.currentSrc;
    // 선택된 파일 표시
    var span = document.querySelector('#url');
    span.textContent = url;
  }, false);
</script>
```

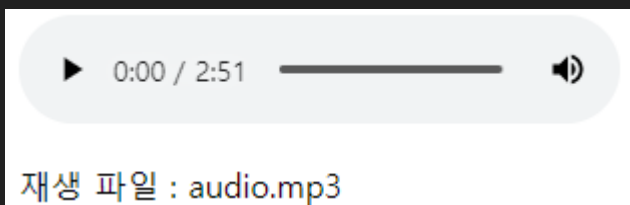
▶ 0:00 / 2:51



재생 파일 : file:///C:/Users/oklin/OneDrive/%EC%88%98%EC%97%85/%EC%88%98%EC%97%85/JavaScript&jQuery/2.0/source/83/audio.ogg

Video & Audio 관련 속성/메서드

- 선택된 파일 확인
 - 파일명만 호출하고 싶다면 src 속성에 접근해서 출력하는 방법이 있다.



```
<script type="text/javascript">
    window.addEventListener("load", function() {
        // audio 요소
        var audio = document.querySelector('audio');
        // 선택된 파일의 URL
        var url = audio.currentSrc;
        // audio요소와 source 요소의 리스트
        var list = document.querySelectorAll('audio,source');
        // 선택된 파일 판별
        var src = "";
        for( var i=0; i<list.length; i++ ) {
            var el = list.item(i);
            if( el.src == url ) {
                src = el.getAttribute("src");
                break;
            }
        }
        // 선택된 파일 표시
        var span = document.querySelector('#url');
        span.textContent = src;
    }, false);
</script>
</head>
<body>
    <audio controls="controls">
        <source src="audio.mp3" type="audio/mpeg" />
        <source src="audio.ogg" type="audio/ogg" />
    </audio>
    <p>재생 파일 : <span id="url">-</span></p>
</body>
```


Video & Audio 관련 속성/메서드

○ 네트워크 이용 상태 파악

- 미디어 요소의 객체에는 미디어 리소스의 선택에서부터 내려받기까지의 변화를 실시간으로 반영하는 `networkState` 프로퍼티가 있다.

메서드 / 속성	설명
<code>media.networkState</code>	미디어 요소(video, audio)의 네트워크 이용 상태를 수치로 반환한다. 값 지정 불가. 이 프로퍼티가 반환하는 값의 의미는 아래와 같다. 0 : 미디어 리소스 선택 전의 초기 상태(NETWORK_EMPTY) 1 : 유휴 상태(NETWORK_IDLE) 2 : 미디어 리소스를 다운로드 중(NETWORK_LOADING) 3 : 미디어 리소스를 찾을 수 없는 경우(NETWORK_NO_SOURCE)

Video & Audio 관련 속성/메서드

○ 네트워크 이용 상태 파악 예제



다운로드 : [MP4](#) | [WebM](#)

```
<script type="text/javascript">
  window.addEventListener("load", function() {
    // video 요소
    var video = document.querySelector('video');
    // a 요소 리스트
    var alist = document.querySelectorAll('#dn a');
    // a 요소에 click 이벤트 리스너를 지정
    for( var i=0; i<alist.length; i++ ) {
      alist.item(i).addEventListener("click", function(event) {
        // 기본 액션을 취소
        event.preventDefault();
        // networkState 프로퍼티의 값을 검사
        if( video.networkState == video.NETWORK_LOADING ) {
          // NETWORK_LOADING 상태라면 경고 창 표시
          alert("잠시후 다시 시도해 주세요");
        } else {
          // NETWORK_LOADING 상태가 아니라면 다운로드
          window.open( event.currentTarget.href, "download" );
        }
      }, false);
    }
  }, false);
</script>
</head>
<body>
  <video controls="controls" preload="auto" width="480" height="272">
    <source src="video.mp4" type="video/mp4" />
    <source src="video.webm" type="video/webm" />
  </video>
  <p id="dn">다운로드 : <a href="video.mp4">MP4</a> | <a href="video.webm">WebM</a></p>
```

Video & Audio 관련 속성/메서드

○ 미디어 리소스의 재생과 정지

- 미디어 요소의 API에는 미디어 리소스를 재생하거나 정지하는 함수, 그리고 정지 중 인지 재생 완료되었는지를 판별하는 프로퍼티가 있다.

메서드 / 속성	설명
media.play()	현재의 재생 위치에서 미디어 리소스를 재생한다. 현재의 재생 위치가 미디어 리소스의 맨 끝이라면 맨 처음으로 돌아가서 재생을 시작한다.
media.pause()	미디어 리소스의 재생을 멈춘다.
media.paused	재생이 정지되어 있다면 true를 반환한다. 실행 중이면 false를 반환한다. 값 지정 불가
media.ended	재생이 종료되었다면, 즉 재생 위치가 미디어 리소스의 맨 뒤라면 true를 반환한다. 그렇지 않으면 false를 반환한다. 값 지정 불가.

Video & Audio 관련 속성/메서드

○ 미디어 리소스의 재생과 정지 예제

play!

pause!

isPause!

ended!

true

true

```
<p> <button id="play">play! </button> </p>
<p> <button id="pause">pause! </button> </p>
<p> <button id="ispause">isPause! </button> </p>
<p> <button id="ended">ended! </button> </p>
<p id="printpause"></p>
<p id="printended"></p>
```

```
<script type="text/javascript">
window.addEventListener("load", function() { // audio 요소를 생성
if( ! window.Audio ) { return; }
var audio = new Audio();
// 재생할 수 있는 오디오 형식을 판별하여 src에 지정
if( audio.canPlayType('audio/mpeg') ) {
    audio.src = "audio.mp3";
} else if( audio.canPlayType('audio/ogg') ) {
    audio.src = "audio.ogg";
} else {
    return;
}

var button1 = document.querySelector('#play');
button1.addEventListener("click", function() { audio.play();});
var button2 = document.querySelector('#pause');
button2.addEventListener("click", function() { audio.pause();});
var button3 = document.querySelector('#ispause');
button3.addEventListener("click", function() {
    var print1 = document.getElementById("printpause");
    print1.innerHTML = audio.paused;});
var button4 = document.querySelector('#ended');
button4.addEventListener("click", function() {
    var print2 = document.getElementById("printended");
    print2.innerHTML = audio.ended;});
});
</script>
```

Video & Audio 관련 속성/메서드

- 강제로 미디어 요소를 재설정하기

- load() 함수를 사용하면 강제로 미디어 요소를 재설정할 수 있다.

메서드 / 속성	설명
media.load()	미디어 요소(video, audio) 요소를 재설정하여 미디어 리소스의 설정을 다시 지정하고 미디어 리소스를 로드한다.

- load() 함수에는 이름과는 다르게 미디어 요소를 리셋하는 기능이 있다.
 - 그리고 선택된 미디어 리소스를 필요에 따라 다시 읽어들이 수 있다.
 - 실제 읽어들이는 미디어의 데이터 양은 preload 속성값에 다르다.

Video & Audio 관련 속성/메서드

○ 강제로 미디어 요소를 재설정하기 예제



<p>비디오를 선택하고 비디오를 클릭해 주세요

```
<select name="list">
  <option value=""></option>
  <option value="video1">video1</option>
  <option value="video2">video2</option>
  <option value="video3">video3</option>
</select>
```

</p>

<p><video poster="poster.png" width="480" height="272"></video></p>

```
<script type="text/javascript">
  window.addEventListener("load", function() {
    var video = document.querySelector('video');

    var playback_toggle = function() {
      ( video.paused == true ) ? video.play() : video.pause();
    };

    video.addEventListener("click", playback_toggle, false);
    video.addEventListener("touchend", playback_toggle, false);
    var select = document.querySelector('select[name="list"]');
    select.addEventListener("change", set_video, false);
  }, false);

  function set_video() {
    var video = document.querySelector('video');
    var select = document.querySelector('select[name="list"]');
    var name = select.value;
    if( ! name.match(/^video[1-3]$/) ) { return; }
    if( video.canPlayType('video/mp4') ) {
      video.src = name + ".mp4";
    } else if( video.canPlayType('video/webm') ) {
      video.src = name + ".webm";
    } else if( video.canPlayType('video/ogg') ) {
      video.src = name + ".ogg";
    } else {
      return;
    }
    video.load();
  }
</script>
```


Video & Audio 관련 속성/메서드

- 미디어 데이터의 로드 상태를 실시간으로 파악하기
 - 미디어 요소의 객체에는 현재의 재생 위치에 관련된 미디어 데이터의 로드 상태를 실시간으로 반영하는 `readyState` 프로퍼티가 존재한다.

메서드 / 속성	설명
<code>media.readyState</code>	현재의 재생 위치에 관련 미디어 데이터의 로드 상태를 나타내는 숫자를 반환한다. 값 지정 불가. 0 : 이용 가능한 미디어 리소스의 정보를 얻지 못하거나 현재 재생 위치를 재생하기 위한 미디어 데이터를 얻을 수 없는 상태 (HAVE_NOTHING) 1 : 미디어 리소스의 길이(시간)와 크기(비디오의 경우)는 얻었으나 아직 현재의 재생 위치를 재생하기 위한 미디어 데이터를 얻지 못한 상태 (HAVE_METADATA) 2 : 현재의 재생 위치만을 표시할 수 있는 미디어 데이터는 얻었지만 아직 현재의 재생위치에서 재생을 진행하기 위한 데이터는 얻지 못한 상태 (HAVE_CURRENT_DATA) 3 : 현재의 재생 위치에서 재생을 진행하기 위한 미디어 데이터는 얻었지만 재생 속도에 맞춰 매끄럽게 재생할 수 있을 만큼의 데이터는 얻지 못한 상태 (HAVE_FUTURE_DATA) 4 : 재생 속도에 맞춰 매끄럽게 재생할 수 있을 만큼의 미디어 데이터를 얻은 상태 (HAVE_ENOUGH_DATA)

Video & Audio 관련 속성/메서드

- 미디어 데이터의 로드 상태를 실시간으로 파악하기 예제

```
<button onclick="myFunction()" type="button">Get ready state</button><br>

<video id="myVideo" width="320" height="176" preload="none" controls>
  <source src="long.mp4" type="video/mp4">
  Your browser does not support HTML5 video.
</video>

<script>
var vid = document.getElementById("myVideo");

function myFunction() {
  alert(vid.readyState);
}
</script>
```

이 페이지 내용:

0

확인

Get ready state

▶ 0:00



Video & Audio 관련 속성/메서드

○ 재생속도 변경하기

- 미디어 요소로 재생하는 리소스는 재생 속도를 변경할 수 있다.
- 빨리감기는 물론 되감기도 가능하다.

메서드 / 속성	설명
media.playbackRate	재생 속도의 비율을 나타내는 수치를 반환한다. 이 프로퍼티의 기본값은 1.0으로 보통속도를 나타낸다. 2.0 이라면 2배속을 나타낸다. 재생 중에 값을 지정하여 재생 속도를 변경할 수 있다. 음수를 지정하면 되감는다.
media.defaultPlaybackRate	기본 재생 속도의 비율을 나타내는 값을 반환한다. 값을 지정하여 기본 재생 속도를 변경할 수 있다. 다만 재생 중에 이 값을 변경해도 재생에는 아무런 영향을 미치지 않는다. 하지만 이 값을 지정 한 후 재생을 다시 시작하면 defaultPlaybackRate 프로퍼티의 값의 속도로 재생한다.

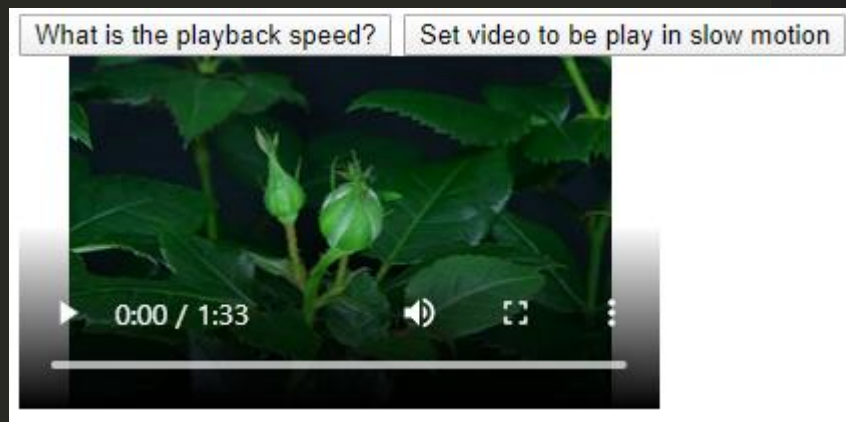
Video & Audio 관련 속성/메서드

○ 재생속도 변경하기 예제 - 1

```
<button onclick="getPlaySpeed()" type="button">What is the playback speed?</button>  
<button onclick="setPlaySpeed()" type="button">Set video to be play in slow motion</button><br>
```

```
<video id="myVideo" width="320" height="176" controls>  
  <source src="video.mp4" type="video/mp4">  
  <source src="video.ogv" type="video/ogg">  
  Your browser does not support HTML5 video.  
</video>
```

```
<script type="text/javascript">  
var vid = document.getElementById("myVideo");  
  
function getPlaySpeed() {  
  alert(vid.playbackRate);  
}  
  
function setPlaySpeed() {  
  vid.playbackRate = 0.5;  
}  
</script>
```



Video & Audio 관련 속성/메서드

○ 재생속도 변경하기 예제 - 2

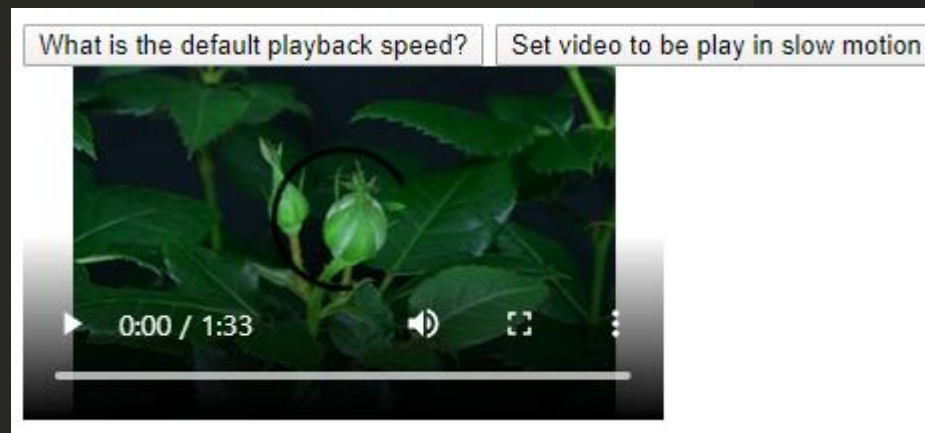
```
<button onclick="getPlaySpeed()" type="button">What is the default playback speed?</button>  
<button onclick="setPlaySpeed()" type="button">Set video to be play in slow motion</button><br>
```

```
<video id="myVideo" width="320" height="176" controls>  
  <source src="video.mp4" type="video/mp4">  
  <source src="video.ogv" type="video/ogg">  
  Your browser does not support HTML5 video.  
</video>
```

```
<script type="text/javascript">  
var vid = document.getElementById("myVideo");
```

```
function getPlaySpeed() {  
  alert(vid.defaultPlaybackRate);  
}
```

```
function setPlaySpeed() {  
  vid.defaultPlaybackRate = 0.5;  
  vid.load();  
}  
</script>
```



Video & Audio 관련 속성/메서드

○ 길이와 재생 위치 파악하기

- 미디어 요소에는 재생하는 미디어 리소스의 길이(시간)와 현재 재생 위치를 초 단위로 파악할 수 있는 프로퍼티가 존재한다.

메서드 / 속성	설명
media.duration	미디어 리소스의 길이(시간)를 초 단위로 반환한다. 미디어 리소스를 재생할 수 없을 때는 NaN을 반환한다. 끝을 알 수 없는 스트리밍과 같은 경우에는 infinity를 반환한다. 이 프로퍼티는 값 지정이 불가능하다.
media.currentTime	현재 재생 위치를 초 단위로 반환한다. 값을 지정하여 재생 위치를 변경할 수 있다.
media.initialTime	미디어 리소스의 로드를 완료했을 때 맨 처음에 재생할 위치를 초 단위로 반환한다. 미디어 파일 재생이라면 보통은 0을 반환한다. 이 프로퍼티는 값 지정이 불가능하다.

Video & Audio 관련 속성/메서드

○ 길이와 재생 위치 파악하기 예제 - 1

```
<button onclick="myFunction()" type="button">Get video length</button><br>
```

```
<video id="myVideo" width="320" height="176" controls>  
  <source src="video.mp4" type="video/mp4">  
  <source src="video.ogv" type="video/ogg">  
  Your browser does not support HTML5 video.  
</video>
```

```
<script>  
var vid = document.getElementById("myVideo");  
  
function myFunction() {  
  alert(vid.duration);  
}  
</script>
```

이 페이지 내용:

93.080956

확인

Get video length



Video & Audio 관련 속성/메서드

○ 길이와 재생 위치 파악하기 예제 - 2

```
<button onclick="getCurTime()" type="button">Get current time position</button>
<button onclick="setCurTime()" type="button">Set time position to 5 seconds</button><br>

<video id="myVideo" width="320" height="176" controls>
  <source src="video.mp4" type="video/mp4">
  <source src="video.ogv" type="video/ogg">
  Your browser does not support HTML5 video.
</video>

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

function getCurTime() {
  alert(vid.currentTime);
}

function setCurTime() {
  vid.currentTime=5;
}
</script>
```



Video & Audio 관련 속성/메서드

○ 길이와 재생 위치 파악하기 예제 - 3

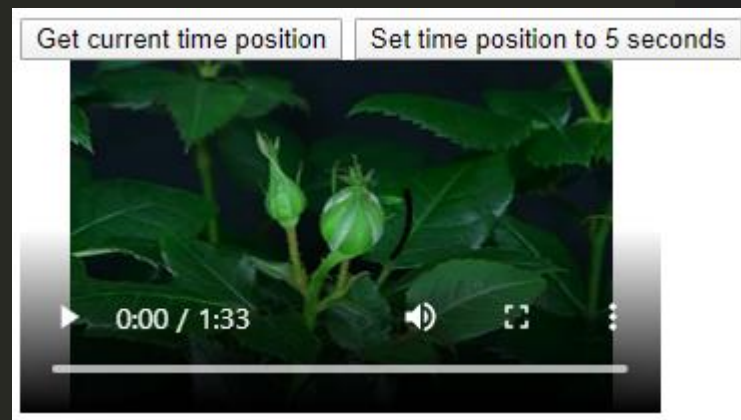
```
<button onclick="getCurTime()" type="button">Get current time position</button>
<button onclick="setCurTime()" type="button">Set time position to 5 seconds</button><br>

<video id="myVideo" width="320" height="176" controls>
  <source src="video.mp4" type="video/mp4">
  <source src="video.ogv" type="video/ogg">
  Your browser does not support HTML5 video.
</video>

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

function getCurTime() {
  alert(vid.currentTime);
}

function setCurTime() {
  vid.currentTime=5;
}
</script>
```



Video & Audio 관련 속성/메서드

- 재생된 범위와 버퍼링 완료 범위를 파악하기

- 미디어 요소로 미디어 리소스의 재생된 범위와 내려받기가 완료되어 버퍼링된 범위를 파악할 수 있다.

메서드 / 속성	설명
media.played	이미 재생이 완료된 미디어 리소스의 범위를 나타내는 TimeRanges 객체를 반환한다.
media.buffered	이미 내려받기가 완료되어 버퍼링된 미디어 리소스의 범위를 나타내는 TimeRanges 객체를 반환한다.

- TimeRanges 객체에서 다루는 프로퍼티는 아래와 같다.

메서드 / 속성	설명
TimeRanges.length	TimeRanges 객체에 담긴 범위 수를 반환한다.
TimeRanges.start(index)	TimeRanges 객체에 담긴 범위 중에 매개변수(index) 번째 범위의 시작 시간을 반환한다. index는 0부터 시작한다.
TimeRanges.end(index)	TimeRanges 객체에 담긴 범위 중에 매개변수(index) 번째 범위의 종료 시간을 반환한다. index는 0부터 시작한다.

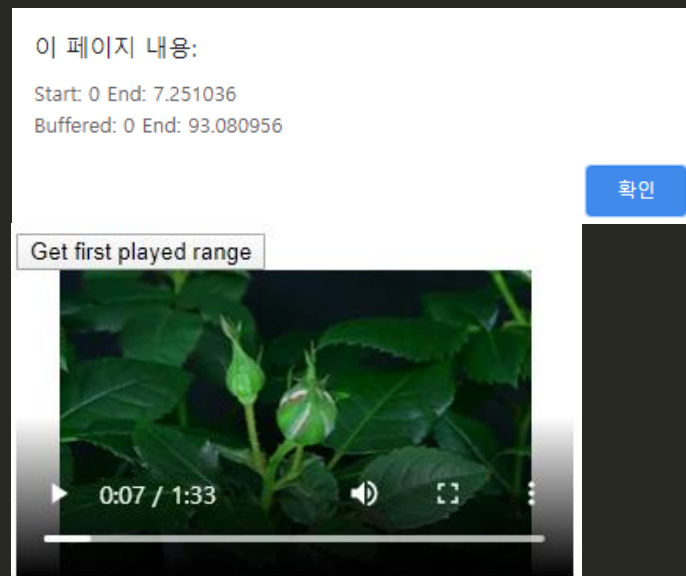
Video & Audio 관련 속성/메서드

- 재생된 범위와 버퍼링 완료 범위를 파악하기

```
<button onclick="myFunction()" type="button">Get first played range</button><br>
```

```
<video id="myVideo" width="320" height="176" controls>  
  <source src="video.mp4" type="video/mp4">  
  <source src="video.ogv" type="video/ogg">  
  Your browser does not support HTML5 video.  
</video>
```

```
<script type="text/javascript">  
  var vid = document.getElementById("myVideo");  
  
  function myFunction() {  
    alert("Start: " + vid.played.start(0) + " End: " + vid.played.end(0) + "\n" +  
      "Buffered: " + vid.buffered.start(0) + " End: " + vid.buffered.end(0));  
  }  
</script>
```



Video & Audio 관련 속성/메서드

○ 탐색 상태 파악하기

- 탐색이란 재생 위치를 찾는 행위를 의미한다.
- 탐색은 브라우저가 구현한 UI의 타임라인을 클릭해서 보고 싶은 위치로 이동하거나 자바스크립트로 미디어 요소의 `currentTime` 프로퍼티에 값을 지정하여 재생 위치를 변경하면 발생한다.
- 미디어 요소에는 탐색 중인지를 파악할 수 있는 프로퍼티와 탐색 가능한 범위를 파악할 수 있는 프로퍼티가 존재한다.

메서드 / 속성	설명
<code>media.seeking</code>	탐색 중이라면 <code>true</code> 를 그렇지 않으면 <code>false</code> 를 반환한다.
<code>media.seekable</code>	탐색 가능한 범위를 나타내는 <code>TimeRanges</code> 객체를 반환한다. 차이점은 <code>seekable</code> 프로퍼티가 반환하는 범위는 그 미디어 리소스 전체 시간의 길이를 나타내는 범위 하나만 반환한다. 즉 <code>seekable</code> 이 반환하는 <code>TimeRanges</code> 객체의 <code>length</code> 프로퍼티는 1이다. 그리고 <code>start(0)</code> 함수는 0을 반환하고 <code>end(0)</code> 함수는 미디어 요소의 <code>duration</code> 프로퍼티와 같은 값을 반환한다. 이러한 성질로 인해 audio 나 video 요소보다는 실시간 스트리밍일 경우 많이 사용된다.

Video & Audio 관련 속성/메서드

○ 음량 변경하기

- 미디어 요소는 volume 프로퍼티를 사용하여 음량을 스크립트로 변경할 수 있다.
- 또 muted 프로퍼티를 사용하면 음소거 상태로 만들 수 있다.

메서드 / 속성	설명
media.volume	현재 음량을 나타내는 수치를 반환한다. 음량을 나타내는 숫자는 0.0~1.0 사이의 실수로 0.0은 최소 음량(무음)을 1.0은 최대 음량을 나타낸다. 기본값은 1.0이며 값을 지정하여 음량을 변경할 수 있다.
media.muted	음소거 되었다면 true 아니면 false를 반환한다. muted 프로퍼티는 volume 프로퍼티보다 우선시된다. volume 프로퍼티가 0.0 보다 큰 값이라도 muted 프로퍼티의 값이 true 라면 음소거 상태가 된다. 값을 지정하여 음소거를 설정하거나 해지할 수 있다.

- muted를 true로 지정했을 시 volume 이 0.0이 되는 것은 아니다.

Video & Audio 관련 속성/메서드

○ 음량 변경하기 예제 - 1

```
<button onclick="getVolume()" type="button">What is the volume?</button>
<button onclick="setHalfVolume()" type="button">Set volume to 0.2</button>
<button onclick="setFullVolume()" type="button">Set volume to 1.0</button><br>
```

```
<video id="myVideo" width="320" height="176" controls>
  <source src="video.mp4" type="video/mp4">
  <source src="video.ogv" type="video/ogg">
  Your browser does not support HTML5 video.
</video>
```

```
<script type="text/javascript">
var vid = document.getElementById("myVideo");
```

```
function getVolume() {
  alert(vid.volume);
}
```

```
function setHalfVolume() {
  vid.volume = 0.2;
}
```

```
function setFullVolume() {
  vid.volume = 1.0;
}
</script>
```



Video & Audio 관련 속성/메서드

○ 음량 변경하기 예제 - 2

```
<button onclick="enableMute()" type="button">Mute sound</button>
<button onclick="disableMute()" type="button">Enable sound</button>
<button onclick="checkMute()" type="button">Check muted status</button><br>

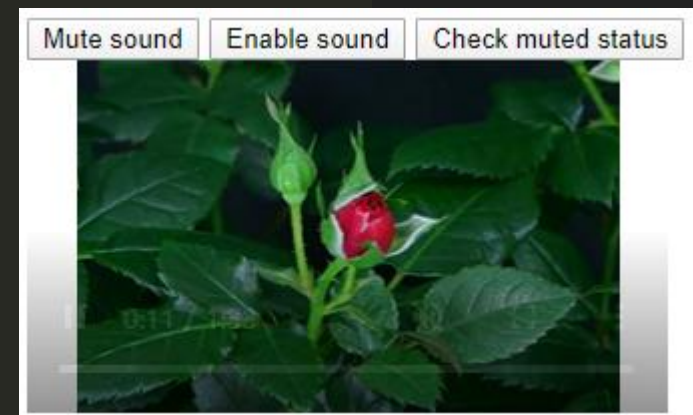
<video id="myVideo" width="320" height="176" controls>
  <source src="video.mp4" type="video/mp4">
  <source src="video.ogv" type="video/ogg">
  Your browser does not support HTML5 video.
</video>

<script>
var vid = document.getElementById("myVideo");

function enableMute() {
  vid.muted = true;
}

function disableMute() {
  vid.muted = false;
}

function checkMute() {
  alert(vid.muted);
}
</script>
```



Video & Audio 관련 속성/메서드

○ 오류 파악하기

- 비디오나 오디오는 파일 크기가 점점 커지는 경향이 있기 때문에 내려받는 도중에 네트워크에 문제 등으로 재생할 수 없을 때가 있다.
- 또 비디오나 오디오의 인코딩 문제로 브라우저에서 재생할 수 없을 수 있다.
- 미디어 요소에는 미디어 리소스의 내려받기부터 재생 준비가 완료되는 사이에 발생하는 어떠한 오류를 파악하는 프로퍼티가 존재한다.

메서드 / 속성	설명
media.error	오류를 나타내는 <code>MediaError</code> 객체를 반환한다. 오류가 없다면 <code>null</code> 을 반환한다.
<code>MediaError.code</code>	오류를 나타내는 코드 번호를 반환한다. 번호의 의미는 다음과 같다. <ol style="list-style-type: none">1. 사용자의 조작에 의해 미디어 리소스 얻기가 중단되었을 때(<code>MEDIA_ERR_ABORTED</code>)2. 네트워크 오류로 미디어 리소스 얻기가 중단되었을 때(<code>MEDIA_ERR_NETWORK</code>)3. 미디어 리소스의 디코딩 중에 오류가 발생했을 때(<code>MEDIA_ERR_DECODE</code>)4. <code>src</code> 속성에 지정된 미디어 리소스가 부적절할 때(<code>MEDIA_ERR_SRC_NOT_SUPPORTED</code>)

이벤트

- 미디어 요소의 이벤트

- 미디어 요소에는 미디어 리소스의 로드부터 재생 중일 때 음량 조정이나 탐색 등에 대한 이벤트가 있다.
- 앞에 설명한 `networkState` 프로퍼티와 `readyState` 프로퍼티는 이벤트의 발생과 함께 상태가 변한다.
- 다음에 설명할 이벤트는 이벤트가 발생했을 시 프로퍼티 값이 어떻게 변하는지도 같이 정리했다.

이벤트

○ 로딩부터 재생 전까지의 이벤트

- 밑의 표는 미디어 요소를 인식하고 나서 미디어 리소스를 내려받은 후 최초의 프레임을 표시하고 대기 상태가 되기까지 발생하는 이벤트를 발생 순서에 따라 나열했다.

이벤트명	설명	NetworkState	readyState
emptied	<ul style="list-style-type: none">- load() 함수를 호출해 미디어 요소가 초기화되었을 때- 네트워크 오류 등의 이유로 미디어 리소스를 전혀 얻지 못했을 때- 미디어 데이터의 파손 등의 이유로 디코딩할 수 없었을 때- 미디어 데이터를 내려받다가 페이지 이동 등의 이유로 사용자에게 의해 처리가 중지됐을 때	NETWORK_EMPTY(코드 : 0)	
loadstart	<ul style="list-style-type: none">- 해당 미디어 요소에 적용할 미디어 리소스를 선정했을 때(내려받기 시작 전)	NETWORK_LOADING(코드 : 2)	
progress	<ul style="list-style-type: none">- 미디어 데이터를 내려받는 중에 계속해서 발생 HTML5 명세에서 이 이벤트의 발생 빈도는 350밀리 초 또는 1바이트 수신할 때마다로 규정한다.	NETWORK_LOADING(코드 : 2)	
loadedmetadata	<ul style="list-style-type: none">- 미디어 리소스의 길이(시간)나 크기(비디오일 경우)가 판명 됐을 때		HAVE_METADATA(코드 1 혹은 그 이상)
loadeddata	<ul style="list-style-type: none">- 미디어 요소에 적용된 미디어 리소스의 길이(시간)나 크기(비디오일 경우)가 판명된 후 맨 처음 재생 위치의 미디어 데이터가 로드되었을 때		HAVE_CURRENT_DATA(코드 2 혹은 그 이상)

이벤트

○ 미디어 요소의 이벤트

이벤트명	설명	NetworkState	readyState
canplay	- 재생 준비가 완료되었으니 아직 재생 속도에 맞춰 자연스럽게 재생될 정도의 미디어 데이터를 내려받지 않은 상태		HAVE_FUTURE_DATA(코드 3) 또는 그 이상
canplaythrough	- 재생 속도에 맞춰 자연스럽게 재생할 수 있을 정도의 미디어 데이터를 내려받았을 때		HAVE_ENOUGH_DATA(코드 4)
progress	- 당장 필요한 미디어 데이터의 내려받기를 완료하고 나서 의도적으로 내려받기를 중단했을 때 - 모든 미디어 데이터를 내려받았는지의 여부는 상관이 없다.	NETWORK_IDLE_DATA(코드 1)	

이벤트

○ 미디어 요소의 이벤트 예제



```
<div class="example">
  <button type="button">Load video</button>
  <video controls width="250"></video>

  <div class="event-log">
    <label>Event log:</label>
    <textarea readonly class="event-log-contents"></textarea>
  </div>
</div>
```

```
<script type="text/javascript">
  const loadVideo = document.querySelector('button');
  const video = document.querySelector('video');
  const eventLog = document.querySelector('.event-log-contents');
  let source = null;

  function handleEvent(event) {
    eventLog.textContent = eventLog.textContent +
      `${event.type} net : ${video.networkState} ready : ${video.readyState}\n`;
  }

  video.addEventListener('emptied', handleEvent);
  video.addEventListener('loadstart', handleEvent);
  video.addEventListener('progress', handleEvent);
  video.addEventListener('loadedmetadata', handleEvent);
  video.addEventListener('loadeddata', handleEvent);
  video.addEventListener('canplay', handleEvent);
  video.addEventListener('canplaythrough', handleEvent);
  video.addEventListener('suspend', handleEvent);

  loadVideo.addEventListener('click', () => {
    if (source) {
      document.location.reload();
    } else {
      loadVideo.textContent = "Reset example";
      source = document.createElement('source');
      source.setAttribute('src', 'long.mp4');
      source.setAttribute('type', 'video/webm');

      video.appendChild(source);
    }
  });
});
```

이벤트

- 재생 및 일시 정지에서 발생하는 이벤트
 - 재생이 시작되고 나서 일시 정지 등 다양한 액션이 발생할 가능성이 있다.
 - 다음은 재생 시작, 재생 중, 정지 사이에 발생할 수 있는 이벤트이다.

이벤트명	설명	NetworkState	readyState
play	<ul style="list-style-type: none">- 재생이 시작되었을 때 또는 사용자가 재생 버튼을 눌러 play() 함수가 호출 되었을 때 발생한다. 미디어 요소에 autoplay 속성이 지정되어 있으면 재생이 시작된 시점에 자동으로 발생한다.- 이 이벤트가 발생하면 paused 프로퍼티는 false 가 된다. 이 이벤트는 재생이 시작되었을 때만 발생한다. 일단 재생이 시작되면 그 후에는 재생이 정지되어 다시 재생이 시작 될 때까지 발생하지 않는다.		
playing	<ul style="list-style-type: none">- 다음 프레임을 얻어 재생이 시작되거나 재시작 할 때 발생한다.- 이 이벤트가 발생했을 때 paused 프로퍼티는 false이다. 이 이벤트는 play 이벤트가 발생할 때도 같이 발생하지만 네트워크 상태가 좋지 않아 다음에 재생할 프레임이 없어 멈췄다 다시 재생될 때에도 발생한다.		HAVE_FUTURE_DATA(3 또는 그 이상)
timeupdate	<ul style="list-style-type: none">- 재생 중에 연속해서 발생한다. HTML5 명세에서는 15 ~ 250 밀리 초 간격으로 발생한다.- 비디오 프레임 단위로 발생하는 것이 아니므로 주의		

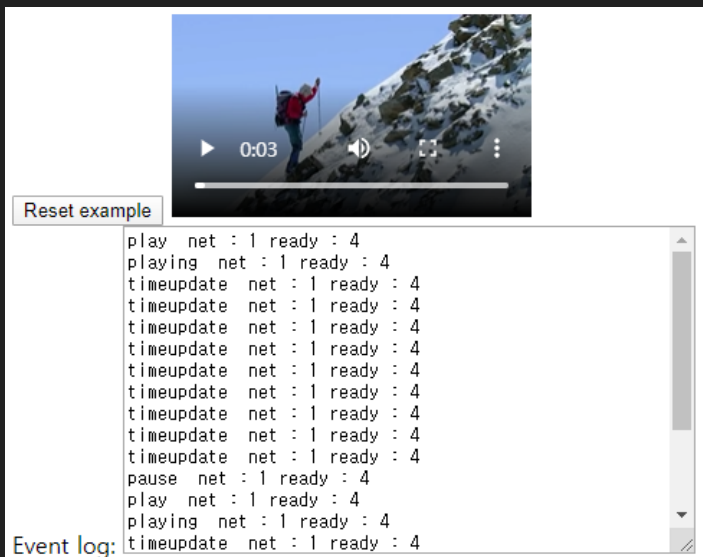
이벤트

○ 재생 및 일시 정지에서 발생하는 이벤트

이벤트명	설명	NetworkState	readyState
pause	<ul style="list-style-type: none">- 재생이 중지되었을 때 발생한다. 사용자가 중지 버튼을 누르거나 pause() 함수가 호출되었을 때 이다.- 이 이벤트가 발생하면 pause 프로퍼티 값이 true 가 된다.		
waiting	<ul style="list-style-type: none">- 다음 프레임을 얻지 못하여 대기하게 되면 발생한다. 이 때 브라우저는 네트워크 오류의 발생 여부는 인식하지 않는다. 대기하는 것으로 다음 미디어 데이터를 얻을 수 있다고 판단하고 있는 상황- 이 이벤트가 발생했을 때 중지 중이지만 paused 프로퍼티의 값은 false이다.		HAVE_FUTURE_DATA(3 또는 그 이상)
stalled	<ul style="list-style-type: none">- 미디어 데이터를 내려 받으려고 하나 얻지 못할 때 발생한다.	NETWORK_LOADING(2)	
ended	<ul style="list-style-type: none">- 미디어 데이터의 길이의 마지막에 닿으면 발생한다. 이 이벤트가 발생했을 때는 ended 프로퍼티 값이 true가 된다.		

이벤트

○ 재생 및 일시 정지에서 발생하는 이벤트 예제



```
<div class="example">
  <button type="button">Load video</button>
  <video controls width="250"></video>

  <div class="event-log">
    <label>Event log:</label>
    <textarea readonly class="event-log-contents"></textarea>
  </div>
</div>
```

```
<script type="text/javascript">
  const loadVideo = document.querySelector('button');
  const video = document.querySelector('video');
  const eventLog = document.querySelector('.event-log-contents');
  let source = null;

  function handleEvent(event) {
    eventLog.textContent = eventLog.textContent +
      `${event.type} net : ${video.networkState} ready : ${video.readyState}\n`;
  }

  video.addEventListener('play', handleEvent);
  video.addEventListener('playing', handleEvent);
  video.addEventListener('timeupdate', handleEvent);
  video.addEventListener('pause', handleEvent);
  video.addEventListener('waiting', handleEvent);
  video.addEventListener('stalled', handleEvent);
  video.addEventListener('ended', handleEvent);

  loadVideo.addEventListener('click', () => {
    if (source) {
      document.location.reload();
    } else {
      loadVideo.textContent = "Reset example";
      source = document.createElement('source');
      source.setAttribute('src', 'long.mp4');
      source.setAttribute('type', 'video/webm');

      video.appendChild(source);
    }
  });
</script>
```

이벤트

○ 조작에 의한 이벤트

- 탐색 중일 때, 재생 속도가 변경되었을 때, 음량이 변경되었을 때 등 사용자의 입력에 의한 변화도 이벤트로 발생한다.

이벤트명	설명
seeking	- 탐색을 시작하면 발생한다. seeking 프로퍼티는 true가 된다.
seeked	- 탐색을 종료할 때 발생한다. 이때 seeking 프로퍼티는 false가 된다.
ratechange	- 재생 속도가 변경되면(defaultPlaybackRate 프로퍼티나 playbackRate 프로퍼티가 변경되면) 발생한다.
durationchange	- 길이가 변경되면(duration 프로퍼티 값이 변경되면) 발생한다.
volumechange	- 음량이 변경되거나 음소거가 전환되면(volume 프로퍼티나 muted 프로퍼티가 변경되면) 발생한다.

이벤트

○ 조작에 의한 이벤트 예제 - 1

```
<video controls="controls" preload="auto" width="480" height="272">
  <source src="video.mp4" type="video/mp4" />
  <source src="video.webm" type="video/webm" />
</video>
<script type="text/javascript">
  const video = document.querySelector('video');

  video.addEventListener('seeking', (event) => {
    console.log('Video is seeking a new position.');
```

```
Video is seeking a new position.
Video is seeking a new position.
Video is seeking a new position.
Video is seeking a new position.
```

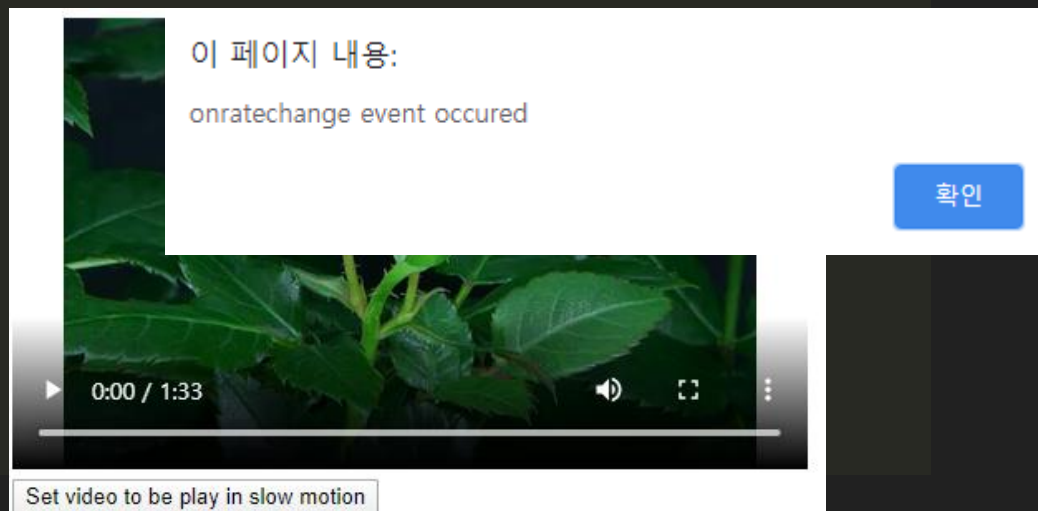

이벤트

○ 조작에 의한 이벤트 예제 - 2

```
<video id="myVideo" controls="controls" preload="auto" width="480" height="272">
  <source src="video.mp4" type="video/mp4" />
  <source src="video.ogv" type="video/webm" />
</video>
<br>
<button onclick="setPlaySpeed()" type="button">Set video to be play in slow motion</button>

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

  function setPlaySpeed() {
    vid.playbackRate = 0.3;
  }
  vid.onratechange = function() {myFunction()};
  function myFunction() {
    alert("onratechange event occured");
  }
</script>
```



이벤트

○ 조작에 의한 이벤트 예제 - 3

```
<video id="myVideo" width="320" height="176" controls>
  <source src="video.mp4" type="video/mp4">
  <source src="video.ogv" type="video/ogg">
</video>

<script type="text/javascript">
  var vid = document.getElementById("myVideo");
  vid.ondurationchange = function() {
    alert("The video duration has changed");
  };
</script>
```

이 페이지 내용:

The video duration has changed

확인

이벤트

○ 조작에 의한 이벤트 예제 - 4

```
<video id="myVideo" width="320" height="176" controls>
  <source src="video.mp4" type="video/mp4">
  <source src="video.ogv" type="video/ogg">
</video>

<script type="text/javascript">
  var vid = document.getElementById("myVideo");
  vid.onvolumechange = function() {
    alert("The volume has been changed");
  };
</script>
```

이 페이지 내용:

The volume has been changed

확인

이벤트

○ 오류에 의한 이벤트

- 미디어 요소에는 미디어 리소스의 로드에서 재생에 이르기까지 어떠한 문제가 발생하면 이벤트가 발생한다.

이벤트명	설명	networkState
error	- 미디어 데이터를 얻는데 어떠한 오류가 발생했을 때 발생한다. 이 이벤트가 발생하면 error 프로퍼티에서 얻을 수 있는 에러코드는 MEDIA_ERR_NETWORK(2) 또는 그 이상의 값이 된다.	NETWORK_EMPTY 또는 NETWORK_IDLE
abort	- 예러가 아닌 이유로 미디어 데이터 내려받기가 중단됐을 때 발생한다. 이 이벤트가 발생하면 error 프로퍼티에서 얻을 수 있는 에러 코드는 MEDIA_ERR_ABORTED(1) 이다.	NETWORK_EMPTY 또는 NETWORK_IDLE

커스텀 플레이어

○ 커스텀 플레이어 만들기



자막 넣기

- 커스텀 플레이어 만들기