

HTML 표준화 문서를 기반으로 하는 지침서



속이 깊은

HTML5 & CSS3

김명진 지음

13강

오디오와 비디오

학습 목표

기존 웹 문서에서는 오디오 및 비디오를 재생하기 위하여 embed 요소나 object 요소의 도움을 받아서 외부 플러그인 플래시 또는 실버라이트 등을 불러와서 사용했었다. 그러나 HTML5에서는 자체적으로 오디오 및 비디오재생이 가능하도록 audio 요소와 video 요소를 추가하였기 때문에 별도의 플러그인 없이도 오디오 및 비디오의 재생이 가능하게 된 것이다. 따라서 이번 장에서는 미디어에 대한 기본적인 내용들과 오디오 및 비디오에 대한 기능들에 대하여 살펴보도록 한다.

Section

1 미디어 요소

2 오디오

3 비디오



표준화 문서	HTML 5 - http://www.w3.org/html/wg/drafts/html/CR/
표준화 단계	Last Call Working Draft(Editor's Draft) (2014-06-27)
문서 위치	4.7.10 Media elements

■ 미디어 요소의 인터페이스

enum	CanPlayTypeEnum { "" /* 빈 문자열을 나타냄 */, "maybe", "probably" };	
Interface	HTMLMediaElement : HTMLElement	
미디어 오류 상태	속성	error;
네트워크 상태	속성	src, currentSrc, crossOrigin, networkState, preload, buffered
	메서드	load(), canPlayType(type);
	상수값	NETWORK_EMPTY = 0 / NETWORK_IDLE = 1 NETWORK_LOADING = 2 / NETWORK_NO_SOURCE = 3
준비 상태	속성	readyState, seeking
	상수값	HAVE_NOTHING = 0 / HAVE_METADATA = 1 / HAVE_CURRENT_DATA = 2 HAVE_FUTURE_DATA = 3 / HAVE_ENOUGH_DATA = 4
재생 상태	속성	currentTime, duration, paused, defaultPlaybackRate, playbackRate, played, seekable, ended, autoplay, loop
	메서드	getStartDate(), play(), pause()
미디어 컨트롤러	속성	mediaGroup, controller
재생 도구	속성	controls, volume, muted, defaultMuted
트랙	속성	audioTracks, videoTracks, textTracks,
	메서드	addTextTrack(kind, label, language)



■ 미디어 요소의 인터페이스

<i>media.currentSrc</i>	현재 미디어 자원의 주소를 반환한다. 미디어 자원이 없을 경우 빈 문자열을 반환한다.
<i>media.networkState</i>	다음 코드 중에서 해당 요소에 대한 현재 네트워크 활동 상태를 나타내는 코드를 반환
NETWORK_EMPTY	요소가 아직 초기화되어 있지 않아서 속성 모두 각각의 초기 상태에 있다는 것을 나타낸다.
NETWORK_EMPTY	요소가 자원을 선택했지만, 현재 실제로 네트워크를 사용하고 있지 않다는 것을 나타낸다.
NETWORK_LOADING	브라우저에서 활발히 데이터의 다운로드 시도를 하고 있는 상태를 나타낸다.
NETWORK_NO_SOURCE	요소가 사용할 자원을 아직 찾지 못했다는 것을 나타낸다.



■ 미디어 요소의 인터페이스



미디어 자원의 로딩, 길이 및 현재 위치

media.load()

미디어 요소를 재설정하고 처음부터 다시 새 미디어 자원을 선택하고 다운로드하기 시작한다.



buffered 속성

- 브라우저가 버퍼링 한 미디어 자원의 범위를 나타내는 TimeRanges 객체를 반환



preload 속성

- 미디어 자원이 버퍼링 되거나 재생되는 때에도 다음과 같은 키워드를 사용하여 변경될 수 있다.

키워드	설명
none	버퍼링이 어쨌든 시작되면(예를 들어, 사용자가 "재생" 버튼을 누를 때) 미디어 자원을 실제로 얼마나 다운로드 해야 하는지에 관한 어떤 정보도 제공하지 않는다.
metadata	자원에 대한 메타데이터(크기, 트랙 목록, 지속 시간 등)와 처음 몇 프레임이라도 가져오는 것이 합리적이라는 힌트를 브라우저에게 준다. 브라우저에서는 이 메타데이터를 가져오면, 미디어 요소의 readyState 속성은 HAVE_METADATA로 설정될 것이다. 그러나 미디어 자원이 재생 중일 때는, 일반적으로 프레임 몇 개가 따라올 것이고, 속성은 HAVE_CURRENT_DATA 또는 HAVE_FUTURE_DATA가 될 것이다.
auto	리소스 전체를 안전하게 다운로드 받을 수 있으면서도 서버에 위험 부담을 주지 않고 사용자의 필요를 우선적으로 처리해야 한다는 정보를 브라우저에게 준다.

media.duration

초 단위로 미디어 자원의 길이를 반환한다.

media.currentTime [=value]

초 단위로 재생 위치를 반환한다. 주어진 시간을 찾아 가도록 설정될 수 있다.



■ 미디어 요소의 인터페이스



미디어의 준비 상태

media.readyState	현재 재생 위치와 관련하여 요소의 현재 상태를 아래 목록으로 표현하는 값을 반환한다.
HAVE_NOTHING	미디어 자원에 관한 정보가 없기 때문에 현재 재생 위치에 대한 데이터가 없는 상태를 나타낸다.
HAVE_METADATA	자원을 충분히 얻었기(메타 데이터 다운로드가 완료된 상태) 때문에 자원의 지속 시간을 알 수 있다.
HAVE_CURRENT_DATA	현재 재생 위치에서 더 이상 진행할 수 없을 정도로 데이터가 충분히 있지 않거나 재생 방향으로 가도 더 이상 얻을 데이터가 없는 상태를 나타낸다.
HAVE_FUTURE_DATA	현재 재생 위치에서 재생 방향으로 앞으로 더 나갈 수 있을 정도로 충분한 데이터가 있는 상태(조금 더 재생할 수 있는 만큼 다운 로드가 완료된 상태)를 나타낸다.
HAVE_ENOUGH_DATA	<p>HAVE_FUTURE_DATA 조건을 충족하고 다음 중 하나를 만족해야 한다.</p> <ul style="list-style-type: none"> ✓ 데이터가 유효 재생 속도로 진행할 때 재생 미디어 자원의 끝에 도달하기 전에 현재의 재생 위치가 이용 가능한 데이터를 추월하지 않을 것으로 브라우저가 추정한다. ✓ 브라우저가 더 이상 추가적으로 획득하는 데이터가 발생하지 않는다는 상태를 입력한다(예들 들어, 버퍼가 가득차 있을 때).



■ 미디어 요소의 인터페이스



미디어의 오류 상태

<i>media.error</i>	미디어 요소의 현재 오류 상태를 나타내는 객체인 <code>MediaError</code> 를 반환한다.	
<i>media.error.code</i>	아래 목록에서 현재 오류 상태를 나타내는 오류 코드를 반환한다.	
	MEDIA_ERR_ABORTED	미디어 자원을 가져오는 프로세스가 사용자에게 의해서 중지됨.
	MEDIA_ERR_NETWORK	자원이 완전히 만들어져서 사용할 수 있더라도, 어떤 네트워크 오류가 발생하여 브라우저에서 미디어 자원을 더 이상 가져오지 못함.
	MEDIA_ERR_DECODE	자원이 완전히 만들어져서 사용할 수 있더라도, 미디어 자원을 디코딩하는 중에 어떤 오류가 발생함.
	MEDIA_ERR_SRC_NOT_SUPPORTED	src 속성이 나타내는 미디어 자원이 적절하지 않음.



■ 미디어 요소의 인터페이스



미디어 자원의 컨트롤러 속성

<i>media.volume</i> [=value]	현재의 재생되는 미디어 자원의 볼륨을 반환한다.
<ul style="list-style-type: none"> ✓ 0.0 ~ 1.0 사이의 값으로 현재 재생되는 미디어 자원 볼륨 값을 변경할 수 있다. ✓ 만일, 0.0 ~ 1.0 사이의 값이 아닌 경우에는 <code>IndexSizeError</code> 오류를 발생시킨다. 	
<i>media.muted</i> [=value]	현재 오디오가 묵음 처리되었다면, <code>true</code> 를 반환한다.
<ul style="list-style-type: none"> ✓ 오디오의 묵음처리 여부를 바꾸도록 설정할 수 있다. 컨트롤러나 이 컨트롤러에 종속된 모든 미디어 요소의 다른 속성과 관계없이 오디오의 묵음 처리 여부를 반환한다. 	
<i>media.defaultMuted</i>	<code>muted</code> 콘텐츠 속성을 반영한다.
<ul style="list-style-type: none"> ✓ 어떤 동적인 효과도 없이 요소의 기본 상태를 제어만 한다. 다음의 예제는 자동으로 재생되지만, 사용자에게 방해가 되지 않도록 무음 상태에서 재생되고 사용자가 소리를 켤 수 있도록 한다. 	
<pre><video src="adverts.cgi?kind=video" controls autoplay loop muted> </video></pre>	



■ 미디어 요소의 인터페이스



미디어 자원의 재생과 검색을 위한 메서드 및 속성

<i>media.played</i>	재생한 미디어 자원의 범위를 나타내는 TimeRanges 객체를 반환한다.
<i>media.paused</i>	재생이 멈추면 true값을 반환한다. 그렇지 않으면 false 값을 반환한다.
<i>media.ended</i>	미디어 자원의 재생이 끝까지 완료되면 true 값을 반환한다.
<i>media.seeking</i>	브라우저에서 현재 재생 위치를 찾고 있으면, true 값을 반환한다.
<i>media.seekable</i>	브라우저에서 찾아갈 수 있는 미디어 자원의 범위를 나타내는 TimeRanges 객체를 반환한다.
<i>media.defaultPlaybackRate</i> [=value]	미디어를 재생하기 위한 재생 기본 속도를 반환한다.
✓ 재생 기본 속도를 변경할 수 있도록 설정될 수 있다. 요소에 현재 미디어 컨트롤러가 있을 때, 이 속성은 무시되고 현재 미디어 컨트롤러의 defaultPlaybackRate가 대신 사용된다.	
<i>media.playbackRate</i> [=value]	현재 재생 속도를 반환한다(1.0이 보통 속도)
✓ 재생 속도를 변경할 수 있도록 설정될 수 있다. 요소에 현재 미디어 컨트롤러가 있을 때, 이 속성은 무시되고 현재 미디어 컨트롤러의 playbackRate가 대신 사용된다.	
<i>media.play()</i>	미디어 자원을 로딩하고 재생을 시작하며 paused 속성을 false로 설정한다. 재생이 끝나면, 처음부터 다시 재생을 시작한다.
<i>media.pause()</i>	미디어 자원을 로딩하며 속성값을 true로 설정한다.



■ 미디어 컨트롤러의 인터페이스

enum	MediaControllerPlaybackState { "waiting", "playing", "ended" };
Interface	MediaController : EventTarget
속성	readyState, buffered, seekable, duration, currentTime, paused, playbackState, played, defaultPlaybackRate, playbackRate, volume, muted
메서드	pause(), unpause(), play()
이벤트 핸들러	onemptied, onloadedmetadata, onloadeddata, oncanplay, oncanplaythrough, onplaying, onended, onwaiting, ondurationchange, ontimeupdate, onplay, onpause, onratechange, onvolumechange



미디어 컨트롤러 객체 생성

```
controller = new MediaController()
```



■ 미디어 컨트롤러의 인터페이스



미디어 컨트롤러 객체의 속성과 메서드

<code>media.controller [= controller]</code>	미디어 요소를 위한 현재의 미디어 컨트롤러가 있다면, 이 컨트롤러를 반환한다. 그렇지 않으면 null을 반환한다.
✓	명시적으로 미디어 컨트롤러를 지정함으로써, 미디어 컨트롤러를 변경할 수 있다. 이런 경우, mediagroup 속성은 제거된다.
<code>controller.buffered</code>	TimeRanges 객체를 반환한다.
✓	브라우저에서는 종속된 모든 미디어 요소를 위해 관련되어 있는 모든 미디어 데이터를 갖고 있다. 그리고 이 데이터의 시간 범위들이 교차하는 부분을 나타낸 것이 TimeRanges 객체이다.
<code>controller.seekable</code>	TimeRanges 객체를 반환한다.
✓	브라우저에서는 종속된 모든 미디어 요소를 위해 찾아 들어가는 시간 범위들이 교차하는 부분을 나타낸 것이 TimeRanges 객체이다.
<code>controller.duration</code>	미디어 자원의 재생 가능한 시간을 반환한다. 미디어가 없으면, 0을 반환한다.
✓	최초로 재생이 가능한 때와 가장 마지막으로 재생이 가능한 때 사이의 차이를 나타낸다
✓	현대 데이터가 실제로 버퍼링되어 있는지 또는 직접적으로 찾아 갈 수 있는지는 고려하지 않는다.
✓	무한한 스트림에서 나중에 재생되는 부분의 시간은 포함하지 않는다.
<code>controller.paused</code>	미디어 자원의 재생이 멈추면, true를 반환한다. 그렇지 않으면 false를 반환한다.
✓	이 속성 값이 true이면, 이 컨트롤러에 종속된 모든 미디어 요소도 멈춘다.
<code>controller.currentTime [=value]</code>	초 단위로 현재 재생 위치를 반환한다.
✓	0 ~ 현재 duration 사이의 위치를 나타내는 것으로, 주어진 시간으로 찾아 갈 수 있도록 설정될 수 있다.



■ 미디어 컨트롤러의 인터페이스



미디어 컨트롤러 객체의 속성과 메서드

<code>controller.defaultPlaybackRate [=value]</code>	기본 재생 속도를 반환하며, 기본 재생 속도를 변경할 수 있다.
<ul style="list-style-type: none"> ✓ 요소에 현재 미디어 컨트롤러가 있을 때, 이 속성은 무시되고 현재 미디어 컨트롤러의 <code>defaultPlaybackRate</code>가 대신 사용된다. ✓ 이 속성 값으로는 재생에 아무런 영향을 주지 않지만, 사용자가 빠른 속도로 재생하였다가, 다시 보통의 속도로 재생한다면, 재생 속도(<code>playbackRate</code>)은 이 기본 속도로 돌아간다. 	
<code>controller.playbackRate [=value]</code>	현재 재생 속도를 반환한다. 재생 속도를 변경할 수 있다.
<code>controller.volume [=value]</code>	현재의 재생되는 미디어 자원의 볼륨을 반환한다.
<ul style="list-style-type: none"> ✓ 0.0 ~ 1.0 사이의 값으로 현재 재생되는 미디어 자원 볼륨 값을 변경할 수 있다. 	
<code>controller.muted [=value]</code>	현재 오디오가 묵음 처리되었다면, <code>true</code> 를 반환한다.
<ul style="list-style-type: none"> ✓ 오디오의 묵음처리 여부를 바꾸도록 설정할 수 있다. 컨트롤러나 이 컨트롤러에 종속된 모든 미디어 요소의 다른 속성과 관계없이 오디오의 묵음 처리 여부를 반환한다. 	
<code>controller.pause()</code>	<code>paused</code> 속성 값을 <code>true</code> 로 설정한다.
<code>controller.unpause()</code>	<code>paused</code> 속성 값을 <code>false</code> 로 설정한다.
<code>controller.play()</code>	<code>paused</code> 속성 값을 <code>false</code> 로 설정하고 종속되어 있는 각 미디어 요소의 <code>play()</code> 메서드를 호출한다.



■ 미디어 컨트롤러의 인터페이스



미디어 컨트롤러 객체의 이벤트 핸들러

이벤트 이름	이벤트 발생 시점
onemptied	종속된 미디어 요소 모두 최근에 readyState를 HAVE_NOTHING이나 더 큰 값으로 설정할 때, 종속된 미디어 요소가 더 이상 없을 때
onloadedmetadata	종속된 미디어 요소 모두 최근에 readyState를 HAVE_NOTHING이나 더 큰 값으로 설정할 때
onloadeddata	종속된 미디어 요소 모두 최근에 readyState를 HAVE_CURRENT_DATA나 더 큰 값으로 설정할 때
oncanplay	종속된 미디어 요소 모두 최근에 readyState를 HAVE_FUTURE_DATA나 더 큰 값으로 설정할 때
oncanplaythrough	종속된 미디어 요소 모두 최근에 readyState를 HAVE_ENOUGH_DATA이나 더 큰 값으로 설정할 때
onplaying	MediaController가 더 이상 차단된 미디어 컨트롤러가 아닐 때
onwaiting	MediaController가 차단된 미디어 컨트롤러일 때
onended	종속된 미디어 요소 모두 최근에 재생을 마친 때
ondurationchange	duration 속성이 방금 업데이트되었을 때
ontimeupdate	미디어 컨트롤러 위치가 변경되었을 때
onplay	paused 속성이 최근에 false로 변경되었을 때
onpause	paused 속성이 최근에 true로 변경되었을 때
onratechange	defaultPlaybackRate 속성이나 playbackRate 속성이 방금 업데이트되었을 때
onvolumechange	volume 속성이나 muted 속성이 방금 업데이트되었을 때



■ 미디어 컨트롤러의 인터페이스



선언적으로 미디어 컨트롤러 할당하기

```
<article>
<style>
  div { margin: 1em auto; position: relative; width: 400px; height: 300px; }
  video { position: absolute; bottom: 0; right: 0; }
  video:first-child { width: 100%; height: 100%; }
  video:last-child { width: 30%; }
</style>
<div>
  <video src="movie.vid#track=Video&track=English" autoplay
    controls mediagroup=movie>
  </video>
  <video src="movie.vid#track=sign" autoplay mediagroup=movie> </video>
</div>
</article>
```

- 위 예제는, video 요소 2개와 함께 묵시적으로 미디어 컨트롤러를 사용하여 영화 파일의 수화 해설 트랙이 같은 비디오 파일의 주요 비디오 트랙 위에 오버레이 형태로(위에 겹쳐서 나타나는 형태) 나타나도록 하고 있다.



■ 텍스트 트랙(TextTrack)

- 텍스트 트랙은 미디어와 관련된 시간 및 텍스트 콘텐츠를 제공하는 신호 모음
 - 각 텍스트 트랙에는 해당 TextTrack 객체가 있다.
 - 텍스트 트랙의 종류로는 subtitles, captions, descriptions, chapters, 그리고 metadata 문자열로 구분될 수 있다.
 - 텍스트 트랙 신호(cue)는 텍스트 트랙에서 시간의 민감한 데이터 단위로서, 특정 시간에 표시되고 다른 시간에 사라지는 텍스트의 자막과 제목에 적용된다.
 - 각 텍스트 트랙 신호는 해당하는 TextTrackCue 객체가 있으며, 특정 텍스트 트랙과 연관될 수 있다.



■ 텍스트 트랙의 신호 구성

신호	설명
식별자	임의의 문자열이다.
시작 시간	신호가 적용되는 시작 부분을 초와 그 이하의 단위로 지정한다.
종료 시간	신호가 적용되는 끝 부분을 초와 그 이하의 단위로 지정한다.
pause-on-exit 플래그	신호가 적용되는 범위의 끝에 도달할 때, 미디어 요소의 재생이 멈추어야 하는지를 나타내는 논리 값
추가된 형식별 데이터	포맷을 위해 필요에 따라서 추가된 필드. 예를 들어, WebVTT는 텍스트 트랙 신호의 쓰기 방향을 가지고 있다.
신호 데이터	신호의 가공되지 않은 텍스트와 이를 표현하기 위한 규칙이다. 이 데이터는 포맷에 의해 정의된다. 예를 들어, WebVTT 텍스트를 사용한다.



■ 텍스트 트랙(TextTrack)



텍스트 트랙 목록 인터페이스

Interface	TextTrackList : EventTarget
속성	length
메서드	TextTrack(index), getTrackById(id)
이벤트 핸들러	onchange, onaddtrack, onremovetrack



텍스트 트랙 목록 객체의 속성 및 메서드

media.textTracks.length	미디어 요소의 텍스트 트랙으로 구성된 목록 내에 있는 텍스트 트랙의 개수를 반환한다.
media.textTracks.[n]	n번째 텍스트 트랙을 나타내는 TextTrack 객체를 반환한다.
track.track	track 요소의 텍스트 트랙을 나타내는 TextTrack 객체를 반환한다.
textTrack.media.textTracks.getTrackById(id)	주어진 식별자로 TextTrack 객체를 반환하고 객체가 없으면 null을 반환한다.



■ 텍스트 트랙(TextTrack)



텍스트 트랙 인터페이스

enum	TextTrackMode { "disabled", "hidden", "showing" };
enum	TextTrackKind { "subtitles", "captions", "descriptions", "chapters", "metadata" };
Interface	TextTrack : EventTarget
속성	kind, label, language, id, inBandMetadataTrackDispatchType, mode, cues, activeCues
메서드	addCue(cue), removeCue(cue)
이벤트 핸들러	oncuechange;



텍스트 트랙 객체의 속성 및 메서드

textTrack.kind	텍스트 트랙의 종류를 문자열로 반환한다.
textTrack.label	텍스트 트랙의 레이블을 반환한다.
textTrack.language	텍스트 트랙의 언어 문자열을 반환한다.
textTrack.id	주어진 트랙에 대한 식별자(ID)를 반환한다.
textTrack.mode	텍스트 트랙의 모드를 반환한다.
textTrack.cues	TextTrackCueList 객체로써, 신호로 구성된 텍스트 트랙 목록을 반환한다.
textTrack.activeCues	TextTrackCueList 객체로써, 현재 활성화되어 있는 신호로 구성된 텍스트 트랙 목록에서 텍스트 트랙 신호를 반환한다.
textTrack.inBandMetadataTrackDispatchType	문서 내에서 다른 스크립트에 보내질 수 있는 대역내(in-band) 메타데이터 트랙에 대한 특별한 미디어 자원에서 추출된 문자열을 반환한다.
textTrack.addCue(cue)	주어진 신호를 textTrack의 신호로 구성된 텍스트 트랙 목록에 추가한다.
textTrack.removeCue(cue)	주어진 신호를 textTrack의 신호로 구성된 텍스트 트랙 목록에서 제거한다.
textTrack.media.addTextTrack(kind, [label [, language]])	미디어 요소의 텍스트 트랙으로 구성된 목록에 추가되는 새로운 TextTrack 객체를 생성하고 반환한다.



■ 텍스트 트랙(TextTrack)



텍스트 트랙 신호 목록 인터페이스

Interface	TextTrackCueList {
속성	length
메서드	TextTrackCue(index), getTrackById(id)



텍스트 트랙 신호 목록 객체의 속성 및 메서드

<i>cuelist.length</i>	목록에 있는 신호들의 개수를 반환한다.
<i>cuelist[index]</i>	목록에서 index인 텍스트 트랙 신호를 반환한다.
<i>cuelist.getCueById(id)</i>	텍스트 트랙 신호 식별자가 id인 첫 번째 텍스트 트랙 신호를 반환한다.



■ 텍스트 트랙(TextTrack)



텍스트 트랙 신호 인터페이스

[Constructor(double startTime, double endTime, DOMString text)]

Interface	TextTrackCue : EventTarget {
속성	track, id, startTime, endTime, pauseOnExit,
이벤트 핸들러	onenter, onexit;



텍스트 트랙 신호 객체의 속성

cue.track	텍스트 트랙 신호가 속해 있는 TextTrack 객체가 있으면 이 객체를 반환한다. 그렇지 않으면 null을 반환한다.
cue.id [= value]	텍스트 트랙의 신호 식별자를 반환하며, 식별자를 설정할 수 있다.
cue.startTime [= value]	초 단위로, 텍스트 트랙 신호의 시작 시간을 반환한다.
cue.endTime [= value]	초 단위로, 텍스트 트랙 신호의 끝 시간을 반환한다.
cue.pauseOnExit [= value]	지정한 endTime에 도달할 경우에 미디어 재생을 중지할지의 여부를 나타낸다. 텍스트 트랙 신호 pause-on-exit 플래그를 설정할 수 있으며, 이 플래그가 설정되어 있으면, true를 반환하고 그렇지 않으면 false를 반환한다.



■ 텍스트 트랙(TextTrack)



데이터 신호 인터페이스

[Constructor(double startTime, double endTime, ArrayBuffer data)]

Interface	DataCue : TextTrackCue
속성	data, text;



데이터 신호 객체의 생성자 및 속성

cue = new **DataCue**(*startTime*, *endTime*, *data*)

- ✓ addCue() 메서드와 함께 사용하기 위해 새로운 TextTrackCue 객체를 반환한다.
- ✓ 전달 인자 startTime은 텍스트 트랙 신호의 시작 시간을 지정한다.
- ✓ 전달 인자 endTime은 텍스트 트랙 신호의 종료 시간을 지정한다.
- ✓ 전달 인자 data는 텍스트 트랙 신호의 가공되지 않은 데이터를 설정한다.

cue.data [= value]	가공 및 구문 분석되지 않은 형태의 텍스트 트랙 신호 데이터를 반환한다
cue.text [= value]	data로부터 UTF-16 텍스트로 변환된 텍스트 트랙 신호 데이터를 반환한다.



■ 텍스트 트랙(TextTrack)



장(chapter)을 설명하는 텍스트 트랙 예

WEBVTT			
01	00:00:00.000 --> 00:50:00.000	18	00:55:00.000 --> 01:30:00.000
02	Astrophysics	19	Data Structures
03		20	
04	00:00:00.000 --> 00:10:00.000	21	01:30:00.000 --> 01:35:00.000
05	Introduction to Astrophysics	22	Answers to Last Exam
06		23	
07	00:10:00.000 --> 00:45:00.000	24	01:35:00.000 --> 01:40:00.000
08	The Solar System	25	Coursework Description
09		26	
10	00:00:00.000 --> 00:10:00.000	27	01:40:00.000 --> 02:30:00.000
11	Coursework Description	28	General Relativity
12		29	
13	00:50:00.000 --> 01:40:00.000	30	01:40:00.000 --> 02:00:00.000
14	Computational Physics	31	Tensor Algebra
15		32	
16	00:50:00.000 --> 00:55:00.000	33	02:00:00.000 --> 02:30:00.000
17	Introduction to Programming	34	The General Relativistic Field Equations



■ 여러 미디어 트랙이 있는 미디어 자원



오디오 및 비디오 트랙에 대한 인터페이스

Interface	AudioTrackList : EventTarget		
속성	length	메서드	AudioTrack(index), getTrackById(id)
이벤트 핸들러	onchange, onaddtrack, onremovetrack		
Interface	AudioTrack {		
속성	id, kind, label, language, enabled		
Interface	VideoTrackList : EventTarget {		
속성	length	메서드	VideoTrack(index), getTrackById(id)
이벤트 핸들러	onchange, onaddtrack, onremovetrack		
Interface	VideoTrack {		
속성	id, kind, label, language, selected		



■ 여러 미디어 트랙이 있는 미디어 자원



오디오 및 비디오 트랙 객체의 속성 및 메서드

<i>media.audioTracks.length</i>	리스트 내에 있는 트랙의 수를 반환한다.
<i>media.videoTracks.length</i>	
<i>audioTrack.id</i>	주어진 트랙의 아이디(ID)를 반환한다. 아이디는 트랙의 포맷이 미디어 조각 URI 문법을 지원하면 조각 식별자와 함께 사용할 수 있고, <code>getTrackById()</code> 메서드와 함께 사용할 수 있는 아이디이다.
<i>videoTrack.id</i>	
<i>audioTrack.kind</i>	주어진 트랙이 해당하는 범주를 반환한다. 가능한 트랙 범주는 아래 트랙 범주에서 확인할 수 있다.
<i>videoTrack.kind</i>	
<i>audioTrack.label</i>	주어진 트랙의 레이블을 알고 있으면 그 레이블을 반환한다. 그렇지 않으면 빈 문자열을 반환한다.
<i>videoTrack.label</i>	
<i>audioTrack.language</i>	주어진 트랙의 언어를 알고 있으면, 그 언어를 반환하거나, 그렇지 않으면 빈 문자열을 반환한다.
<i>videoTrack.language</i>	
<i>audioTrack = media.audioTracks[index]</i>	지정한 <code>AudioTrack</code> 이나 <code>VideoTrack</code> 객체를 반환한다.
<i>videoTrack = media.videoTracks[index]</i>	
<i>audioTrack.enabled [=value]</i>	주어진 트랙이 활성화되어 있으면 <code>true</code> 값을 반환하고 그렇지 않으면 <code>false</code> 값을 반환한다. 트랙의 활성화 여부를 변경할 수 있다.
✓ 만일, 여러 오디오 트랙이 동시에 활성화되어 있다면, 모두 다 들린다.	
<i>audioTrack.selected [=value]</i>	주어진 트랙이 활성화되어 있으면 <code>true</code> 값을 반환하고 그렇지 않으면 <code>false</code> 값을 반환한다. 트랙의 선택 여부를 변경할 수 있도록 설정될 수 있다.
✓ 앞의 트랙이 선택되어 있는 동안, 새 트랙을 선택하면 앞의 트랙은 선택 해제된다.	
<i>media.videoTracks.selectedIndex</i>	현재 선택된 트랙이 있을 경우, 그 트랙의 인덱스를 반환하거나, 그렇지 않으면 <code>-1</code> 을 반환한다.



■ 여러 미디어 트랙이 있는 미디어 자원



오디오 및 비디오 트랙 객체의 속성 및 메서드

audioTrack = **media.audioTracks.getTrackById(id)**

videoTrack = **media.videoTracks.getTrackById(id)**

주어진 식별자(id)에 해당하는 AudioTrack이나 VideoTrack 객체를 반환한다. 해당되는 식별자가 없으면 null을 반환한다.

media.audioTracks

오디오 트랙을 나타내는 AudioTrackList 객체를 반환한다.

media.videoTracks

비디오 트랙을 나타내는 VideoTrackList 객체를 반환한다.

```
<script>
function loadVideo(url, container) {
  var controller = new MediaController();
  var video = document.createElement('video');
  video.src = url;
  video.autoplay = true;
  video.controls = true;
  video.controller = controller;
  container.appendChild(video);
  video.onloadedmetadata = function (event) {
    for (var i = 0; i < video.videoTracks.length; i += 1) {
      if (video.videoTracks[i].kind == 'sign') {
        var sign = document.createElement('video');
        sign.src = url + '#track=' + video.videoTracks[i].id;
        sign.autoplay = true;
        sign.controller = controller;
        container.appendChild(sign);
        return;
      }
    }
  };
}
</script>
```



■ 재생 시간 범위 및 이벤트



재생 시간 범위 인터페이스

Interface	TimeRanges		
속성	length	메서드	start(index), end(index)
<i>media.length</i>	객체 내에 있는 범위들의 개수를 나타낸다.		
<i>time = media.start(index)</i>	주어진 인덱스로 범위 시작 부분의 시간을 반환한다.		
	✓ 인덱스가 범위 밖에 있으면, <code>IndexSizeError</code> 오류를 발생시킨다.		
<i>time = media.end(index)</i>	주어진 인덱스로 범위 끝 부분의 시간을 반환한다.		
	✓ 인덱스가 범위 밖에 있으면, <code>IndexSizeError</code> 오류를 발생시킨다.		



트랙에 대한 이벤트 객체

[Constructor(DOMString type, optional TrackEventInit eventInitDict)]			
Interface	TrackEvent : Event		
속성	readonly attribute (VideoTrack or AudioTrack or TextTrack) track ;		
dictionary	TrackEventInit : EventInit		
속성	(VideoTrack or AudioTrack or TextTrack) track ;		



■ 재생 시간 범위 및 이벤트



트랙에 대한 이벤트 객체

event.track	이벤트가 관련되어 있는 트랙 객체(TextTrack이나 AudioTrack, VideoTrack)를 반환한다.
이벤트 이름	이벤트 발생 시점
loadstart	미디어 데이터를 찾기 시작할 때
progress	미디어 데이터를 가져오고 있을 때
suspend	브라우저에서 의도적으로 미디어 데이터를 가져오고 있지 않을 때
abort	브라우저에서 미디어 데이터를 완전히 다운로드되기 전에 데이터를 가져오는 것을 중지할 때(오류 때문에 중지하는 것은 아님)
error	미디어 데이터를 가져오는 동안 발생할 때,
emptied	미디어 데이터의 networkState가 NETWORK_EMPTY 상태가 되었을 때
stalled	브라우저에서 미디어 데이터를 가져오려고 시도하지만, 예상치 않게 데이터가 오지 않을 때
loadedmetadata	브라우저에서 미디어 자원과의 지속 시간과 크기를 방금 알아냈고 텍스트 트랙이 이용될 수 있도록 준비된 때
loadeddata	브라우저가 처음으로 현재 재생 위치에서 미디어 데이터를 표시할 수 있을 때
canplay	브라우저에서 미디어 데이터를 다시 재생할 수 있지만, 지금 재생해야 할 경우 멈추고 콘텐츠를 더 버퍼링하지 않고서는 끝까지 현재 속도로 미디어 자원을 표시할 수 없다고 판단한 때
canplaythrough	브라우저에서 지금 재생해야 할 경우, 콘텐츠를 더 버퍼링하기 위해서 멈추지 않으면 끝까지 현재 속도로 미디어 자원을 표시할 수 없다고 판단한 때
playing	미디어 데이터가 없어서 잠시 멈추거나 지연된 후 재생을 시작할 준비가 되었을 때



■ 재생 시간 범위 및 이벤트



트랙에 대한 이벤트 객체

이벤트 이름	이벤트 발생 시점
waiting	다음 프레임이 없기 때문에 재생이 멈추었지만, 브라우저에서 적절한 때에 프레임이 생길 것으로 보는 때
seeking	seeking IDL 속성이 true로 바뀌었을 때
seeked	seeking IDL 속성이 false로 바뀌었을 때
ended	재생이 미디어 자원의 끝에 도달하여 멈추었을 때
durationchange	duration 속성이 방금 업데이트되었을 때
timeupdate	정상적인 재생 내용의 일부로서, 또는 중단되는 것과 같이 특이한 방식으로 현재 재생 위치가 바뀌었을 때
play	요소가 더 이상 멈춰 있지 않은 때. 이 이벤트는 play() 메서드가 반환된 이후나 autoplay() 속성으로 인해 재생이 시작되었을 때 발생한다.
pause	요소가 멈췄을 때. 이 이벤트는 pause() 메서드가 반환된 후에 발생한다.
ratechange	defaultPlaybackRate 또는 playbackRate 속성이 막 변경되었을 때
resize	videoWidth 또는 videoHeight 속성의 값이 변경되었을 때
volumechange	volume 또는 muted 속성이 변경되었을 때. 관련 속성의 설정 메서드(setter)가 반환된 후 발생한다.



오디오 요소



audio 요소

- 웹 브라우저에서 플러그인의 도움 없이 오디오를 재생할 수 있도록 한다
 - HTML5에서 추가된 기능이기 때문에 IE8이하에서는 사용할 수 없다.
 - audio 요소는 미디어 데이터가 표면적으로 오디오 데이터인 미디어 요소이며, 다음과 같은 속성 값들을 사용할 수 있다.

속성 값	설명
src	재생할 오디오 파일의 경로를 나타낸다.
preload	<p>웹 문서 로딩시에 오디오의 로딩 상태를 나타낸다. 다음과 같이 3가지의 값을 지정할 수 있으며, 기본 값은 auto이다.</p> <ul style="list-style-type: none"> ✓ none – 오디오를 재생하기 위해 미리 다운로드 하지 않는다. ✓ metadata – 오디오에 대한 기본적인 정보들을(크기, 첫프레임, 오디오 길이 등) 가져온다. ✓ auto – 오디오를 자동으로 재생하기 위해 다운로드를 한다.
autoplay	웹 문서 로딩시에 오디오를 자동으로 재생할 지를 나타낸다.
loop	재생할 오디오의 반복 여부를 나타낸다.
controls	기본으로 제공되는 재생 컨트롤 집합을 표시할 지를 나타낸다. 일반적으로 표시되는 컨트롤은 재생, 일시 정지, 검색 및 볼륨 설정이 포함된다.



오디오 요소



audio 요소

예제 Audio_simplePlay.html

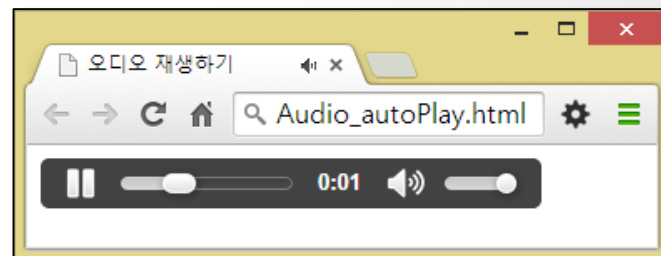
```
<audio src="media/audio_sample.ogg">
  <p>당신의 브라우저는 audio 요소를 지원하지 않습니다. </p>
</audio>
```

예제 Audio_autoPlay.html

```
<audio src="media/audio_sample.ogg" controls="controls" autoplay="autoplay" loop="loop">
  <p>당신의 브라우저는 audio 요소를 지원하지 않습니다. </p>
</audio>
```

```
<audio src="media/audio_sample.ogg" controls autoplay loop>
  <p>당신의 브라우저는 audio 요소를 지원하지 않습니다. </p>
</audio>
```

브라우저	모양
Chrome	
Firefox	
Opera	
Safari	
IE	





오디오 요소

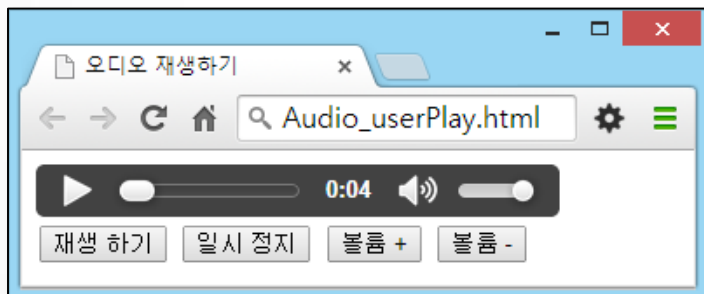


audio 요소

예제

Audio_userPlay.html

```
<audio id="player" src="vincent.mp3"></audio> <br>
<button onclick="document.getElementById('player').play()">Play</button>
<button onclick="document.getElementById('player').pause()">Pause</button>
<button onclick="document.getElementById('player').volume += 0.1">Vol+ </button>
<button onclick="document.getElementById('player').volume -= 0.1">Vol- </button>
```





오디오 요소



오디오 포맷

오디오 포맷	브라우저				
	Chrome Canary	IE 11	Firefox 30	Opera 22	Safari 8.0
PCM	Yes	No	Yes	Yes	No
AAC	No	Yes	No	No	Yes
MP3	Yes	Yes	Yes	No	Yes
Ogg Vorbis	Yes	No	Yes	Yes	No
Ogg Opus	Yes	No	Yes	Yes	No
WebM(Vorbis)	Yes	No	Yes	Yes	No
WebM(Opus)	Yes	No	Yes	Yes	No

※ 자료 참조 - HTML5test.com 브라우저 비교



■ 다양한 포맷 지정



source 요소

- 브라우저마다 지원하는 오디오/비디오 포맷이 다르기 때문에 모든 브라우저에서 오디오/비디오가 실행되기 위해서는 source 요소를 사용하는 것이 효율적이다.
- source 요소는 미디어 요소(audio 요소와 video 요소)를 위해서 여러 대체 미디어 자원을 명시할 수 있다.

속성 값	설명
src	미디어 자원의 주소를 지정하는 것으로, 재생할 파일의 경로를 지정하면 된다. 지정되는 값은 앞으로 공백이 없는 유효한 URL이어야 하고 반드시 지정되어야 한다.
type	미디어 자원을 가져오기 전에 재생할 수 있는지에 대한 브라우저의 판단을 돕기 위한 것으로, 미디어 자원의 MIME 타입을 지정한다. 유효한 MIME 타입에 대해서는 다음의 RFC 문서와 사이트를 참고하기 바란다. [RFC4281] The Codecs Parameter for "Bucket" Media Types (http://tools.ietf.org/html/rfc4281) [Media Types] http://www.iana.org/assignments/media-types/media-types.xhtml
media	브라우저는 미디어 자원을 가져오기 전에 이 자원이 사용자에게 유용한지를 판단할 수 있도록 도와주기 위해 이 속성 값을 사용하여, 미디어 자원에 대한 유효한 미디어 쿼리를 지정한다. 이 값이 생략되면 기본적으로 "all" 값으로 지정되고, 지정한 미디어 자원이 모든 미디어에 적합하다는 뜻이다.



다양한 포맷 지정



source 요소

MIME 타입

- 인터넷 미디어 유형(Internet Media Types)으로 알려진 MIME 타입은 시스템이 이를 처리하는 방법을 알 수 있도록 파일 형식을 정의하는 방법

Ogg 컨테이너 내에 단독으로 있는 보비스(Vorbis) 오디오

```
<source src='audio.ogg' type='audio/ogg; codecs="vorbis" '
```

Ogg 컨테이너 내에 단독으로 있는 스피크스(Speex) 오디오

```
<source src='audio.spx' type='audio/ogg; codecs="speex" '
```

Ogg 컨테이너 내에 단독으로 있는 FLAC 오디오

```
<source src='audio.oga' type='audio/ogg; codecs="flac" '
```

예제

Audio_sourcePlay.html

```
<audio controls>
  <source src="media/audio_sample.ogg" type="audio/ogg">
  <source src="media/audio_sample.mp3" type="audio/mpeg">
  <p>당신의 브라우저는 audio 요소를 지원하지 않습니다. </p>
</audio>
```



■ 다양한 포맷 지정



source 요소

■ 지원하는 오디오 MIME 타입을 사전에 알 수 있는 방법

1. MIME 타입의 지원 여부를 알기 전에 오디오 요소를 가져온다.

- 첫 번째 방법 - 오디오의 아이디를 사용해서 오디오 요소를 가져올 수 있다

```
var audio = document.getElementById('오디오 아이디');
```

- 두 번째 방법 - HTML5에서 새롭게 추가된 오디오 API를 사용한다.

```
var audio = new Audio();
```

2. 오디오 포맷을 지원하는지 canPlayType() 메서드를 사용하여 조사한다.

- 첫 번째 방법 - 오디오 포맷을 지정해서 확인할 수 있다.

```
audio.canPlayType('audio/ogg');
```

- 두 번째 방법 - codec을 명확히 지정한다.

```
audio.canPlayType('audio/ogg; codecs="vorbis" ');
```



■ 다양한 포맷 지정



source 요소

- 오디오 컨테이너, 포맷, 파일 확장자, MIME 타입, 그리고 코덱 문자열

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



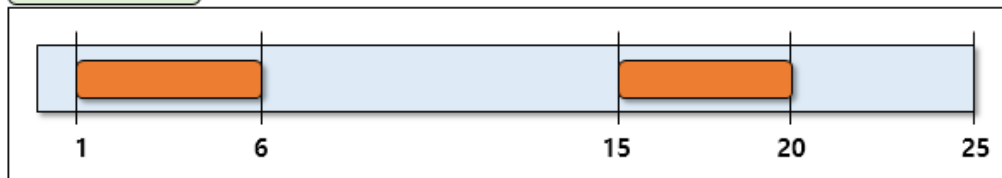
오디오 제어하기



재생 시간 범위 객체(TimeRanges)

```
var buffered = audio.buffered; //버퍼링 된 미디어의 TimeRanges 객체를 반환한다.
var buuferedEnd = audio.buffered.end(); //마지막으로 버퍼링된 시간을 초 단위로 반환한다.
```

TimeRanges



메서드 호출	반환되는 값
audio.buffered.leng	2를 반환한다.
audio.buffered.end()	20을 반환한다.
audio.buffered.start(0)	1을 반환한다.
audio.buffered.end(0)	6을 반환한다.
audio.buffered.start(1)	15을 반환한다.
audio.buffered.end(1)	20을 반환한다.

```
var isSeeking = audio.seeking; //플레이어가 검색이 가능한가?
var isSeekable = audio.seekable && audio.seekable.length > 0; //미디어가 검색 가능한가?
var seekableEnd = audio.seekable.end(); //미디어가 검색 가능한 끝 지점을 초 단위로 반환한다.
```



오디오 제어하기



재생 상태 업데이트하기

- 이벤트 핸들러를 이용하여, 이벤트 발생에 따른 적절한 처리를 하도록 한다.

예제	Audio_userPlay.html
HTML	<pre> 01 <audio id="player" src="media/audio_sample.ogg" controls> </audio>
 02 <p>재생 상태 : </p> 03 <button onclick="document.getElementById('player').play()">재생 하기</button> 04 <button onclick="document.getElementById('player').pause()">일시 정지</button> 05 <button onclick="document.getElementById('player').volume += 0.1">볼륨 +</button> 06 <button onclick="document.getElementById('player').volume -= 0.1">볼륨 -</button> </pre>
Script	<pre> 01 var player = document.getElementById('player'); 02 player.addEventListener("timeupdate", PlayTime, false); 03 player.addEventListener("ended", PlayEnd, false); 04 05 function play() { 06 if(player.paused) player.play(); 07 else player.pause(); 08 } 09 function PlayTime(e) { 10 document.getElementById("playtime").innerHTML = Math.floor(player.currentTime) + "/" + 11 Math.floor(player.duration); 12 } 13 function PlayEnd(e) { 14 document.getElementById("playtime").innerHTML = "재생이 종료되었습니다."; 15 } </pre>



오디오 제어하기



실습하기 - Audio_Study01_SimplePlayer.html

Script

```

01 var currentFile = "";
02 var myAudio, canvas, btn, audioURL, duration;
03 function progressBar() { //업데이트되는 상태바를 표시한다.
04     var elapsedTime = Math.round(myAudio.currentTime);
05     var context = canvas.getContext("2d");
06     context.clearRect(0, 0, canvas.clientWidth, canvas.clientHeight);
07     context.fillStyle = "red";
08     var fWidth = (elapsedTime / duration) * (canvas.clientWidth);
09     if (fWidth > 0) { context.fillRect(0, 0, fWidth, canvas.clientHeight); }
10 }
11 function playAudio() {
12     duration = Math.round(myAudio.duration);
13     if (audioURL.value !== currentFile) { //오디오 파일이 변경되었으면
14         myAudio.src = audioURL.value; //오디오 파일의 src를 수정한다.
15         currentFile = audioURL.value;
16     }
17     if (myAudio.paused) { myAudio.play(); btn.textContent = "일시 멈춤"; }
18     else { myAudio.pause(); btn.textContent = "재생 하기"; }
19 }
20
21 function rewindAudio() { myAudio.currentTime -= 1.0; } //재생 위치를 앞으로 1초씩 옮긴다.
22 function restartAudio() { myAudio.currentTime = 0; } //재생 위치를 처음으로 옮긴다.
23 function volumeUp() { myAudio.volume += 0.1; } //볼륨을 0.1씩 올린다.
24 function volumeDown() { myAudio.volume -= 0.1; } //볼륨을 0.1씩 내린다.

```



오디오 제어하기



실습하기 - Audio_Study01_SimplePlayer.html

Script

```

25 function increaseSpeed() { myAudio.playbackRate += 1; } //재생 속도를 1씩 올린다.
26 function decreaseSpeed() {                               //재생 속도를 1씩 내린다.
27     if (myAudio.playbackRate <= 1) { //재생 속도가 1보다 작거나 같으면,
28         //재생 속도가 1보다 작으면 중간 속도로 변경한다.
29             myAudio.playbackRate = (myAudio.playbackRate / 2);
30         }
31     else { myAudio.playbackRate -= 1; }
32 }
33
34 function defaultSpeed() { myAudio.playbackRate = 1; } //기본 재생 속도로 초기화
35 function PlayEnd() { btn.textContent = "재생 하기"; } //재생이 끝나면 버튼의 레이블을 변경
36
37 function initEvents() {
38     canvas = document.getElementById('myCanvas');
39     myAudio = document.getElementById('myAudio');
40     btn = document.getElementById('play');
41     audioURL = document.getElementById('audioFile');
42     myAudio.addEventListener("timeupdate", progressBar, true);
43     myAudio.addEventListener("ended", PlayEnd, false);
44 }
45 //웹 문서가 로드 되었을 때 이벤트를 초기화 하도록 해준다.
46 window.addEventListener("DOMContentLoaded", initEvents, false);

```

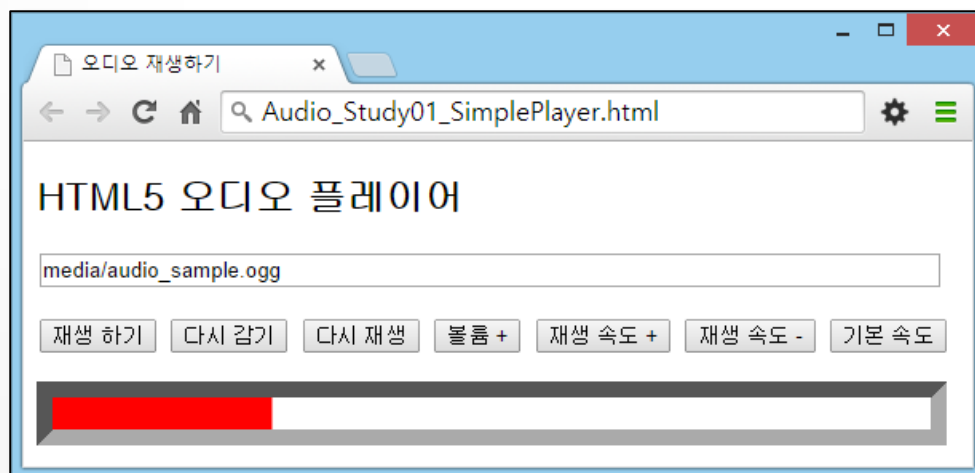



오디오 제어하기



실습하기 - Audio_Study01_SimplePlayer.html

HTML	01	<audio id="myAudio" src="media/audio_sample.ogg">오디오 요소를 지원하지 않음</audio>
	02	<p> <input type="text" id="audioFile" size="80" value="media/audio_sample.ogg" /> </p>
	03	<p> <button id="play" onclick="playAudio();">재생 하기</button>
	04	<button id="rewind" onclick="rewindAudio();">다시 감기</button>
	05	<button id="restart" onclick="restartAudio();">다시 재생</button>
	06	<button id="volumeUp" onclick="volumeUp();">볼륨 +</button>
	07	<button id="increaseSpeed" onclick="increaseSpeed();">재생 속도 +</button>
	08	<button id="decreaseSpeed" onclick="decreaseSpeed();">재생 속도 -</button>
	09	<button id="defaultSpeed" onclick="defaultSpeed();">기본 속도</button>
	10	</p>
	11	<canvas id="myCanvas" width="550" height="20" style="border: 10px inset #aaa">캔 버 스 를 지원하지 않음</canvas>





비디오 요소



video 요소

- 웹 브라우저에서 플러그인의 도움 없이 비디오를 재생할 수 있도록 한다
 - HTML5에서 추가된 기능으로, 웹 페이지 안에서 비디오를 내장하도록 지원한다.

속성 값	설명
src	재생할 비디오 파일을 가리키는 URL을 나타내는 문자열이다.
poster	비디오 데이터가 없는 경우, 브라우저에서 보여줄 수 있는 포스터 이미지 파일의 유효하고 앞뒤로 공백이 없는 URL을 지정한다. 지정한 비디오 데이터가 설정되어 있지 않거나 콘텐츠를 로드하는 중이어서 동영상을 사용할 수 없을 경우에만 포스터 이미지가 표시된다.
preload	웹 문서 로딩시에 비디오 데이터의 로딩 상태를 다음과 같이 3가지의 값으로 지정한다. <ul style="list-style-type: none"> ✓ none – 비디오를 재생하기 위해 미리 다운로드 하지 않는다. ✓ metadata – 비디오에 대한 기본적인 정보들을(크기, 첫 프레임, 비디오 길이 등) 가져온다. ✓ auto – 기본 값으로, 비디오를 자동으로 재생하기 위해 다운로드를 한다.
autoplay	웹 문서 로딩시에 비디오를 자동으로 재생할 지를 나타낸다.
loop	재생할 비디오의 반복 여부를 나타낸다.
controls	기본으로 제공되는 재생 컨트롤 집합을 표시할 지를 나타낸다. 일반적으로 표시되는 컨트롤은 재생, 일시 정지, 검색 및 볼륨 설정이 포함된다.
width	비디오의 폭을 지정한다.
height	비디오의 높이를 지정한다.

video.videoWidth

비디오 본래의 폭을 CSS 픽셀 단위로 반환한다.

video.videoHeight

비디오 본래의 높이를 CSS 픽셀 단위로 반환한다.



비디오 요소



video 요소

간단한 비디오 재생

예제 Video_simplePlay1.html

```
<video src="media/bunny_small_ogg.ogg">
  <p>당신의 브라우저는 video 요소를 지원하지 않습니다. </p>
</video>
```



예제 Audio_simplePlay2.html

```
<video src="media/bunny_small_ogg.ogg" width="400px" controls autoplay loop >
  <p>당신의 브라우저는 video 요소를 지원하지 않습니다. </p>
</video>
```



■ 비디오 제어하기



브라우저별 비디오 포맷 지원 현황

※ 자료 참조 - 위키백과(HTML5 비디오)

브라우저	최신 안정판(공개일)	비디오 포맷		
		Ogg	MP4	WebM
IE	11.0.9600.16476 (2013년 10월 17일)	수동 설치	9.0	수동 설치
Firefox	30.0(2014년 06월 10일)	3.5	No	4.0
Chrome	PC용 : 34.0.1847.116(2014년 4월 8일) 안드로이드용 ARM, x86: 34.0.1847.114(2014년 4월 2일) iOS용 : 33.0.1750.21(2014년 3월 27일)	3.0	예	6.0
Safari	7.0.1(2013년 12월 16일)	수동 설치	3.1	No
Opera	Blink (21.0.1432.48)(2014년 4월 24일) Presto (12.17)(2014년 4월 23일)	10.50	No	10.60

- ✓ MP4 ⇒ H264 비디오 코덱과 AAC 오디오 코덱을 사용하는 MPEG 4 파일
- ✓ WebM ⇒ VP8 비디오 코덱과 Vorbis 오디오 코덱을 사용하는 WebM 파일
- ✓ Ogg ⇒ Theora 비디오 코덱과 Vorbis 오디오 코덱을 사용하는 Ogg 파일



■ 비디오 제어하기



source 요소 사용

예제 Video_sourcePlay.html

```
<video controls>
  <source src="media/bunny_small_ogg.ogg" type=" video/ogg">
  <source src="media/bunny_small_avi.avi" type=" video/mp4">
  <p>당신의 브라우저는 audio 요소를 지원하지 않습니다. </p>
</video>
```

MP4 컨테이너 내에 있는 H.264 베이스라인 프로파일 비디오 레벨 3와 복잡성이 낮은 AAC 오디오

```
<source src='video.mp4' type='video/mp4; codecs="avc1.42E01E, mp4a.40.2" '
```

MP4 컨테이너 내에 있는 MPEG-4 시각적인 단순 프로파일 레벨 0와 복잡성이 낮은 AAC 오디오

```
<source src='video.mp4' type='video/mp4; codecs="mp4v.20.8, mp4a.40.2" '
```

Ogg 컨테이너 내에 있는 테오라(Theora) 비디오와 보비스(Vorbis) 오디오

```
<source src='video.ogv' type='video/ogg; codecs="theora, vorbis" '
```



■ 비디오 제어하기



재생 시간 지정

#t=[시작 시간] [,종료 시간]	#t=시간:분:초, 시간:분:초
src="media/bunny_small_ogg.ogg#t=1:20:30, 2:05:00"	1시간 20분 30초부터 2시간 5분까지 재생
src="media/bunny_small_ogg.ogg#t=10,20"	10초부터 20초까지 재생
src="media/bunny_small_ogg.ogg#t=,10"	10초까지 재생
src="media/bunny_small_ogg.ogg#t=20"	20초 부터 재생

예제	Video_timePlay.html
<pre><video src="media/bunny_small_ogg.ogg#t=10, 20" controls autoplay loop> <p>당신의 브라우저는 audio 요소를 지원하지 않습니다. </p> </video></pre>	



비디오 제어하기



실습하기 - Video_Study01_SimplePlayer.html

Script

```

01 var myVideo, video_list, btn;
02 function playVideo() {
03     if (myVideo.paused) { myVideo.play(); btn.textContent = "일시 멈춤"; }
04     else { myVideo.pause(); btn.textContent = "재생 하기"; }
05 }
06 function rewindVideo() { myVideo.currentTime -= 10.0; } //재생 위치를 앞으로 10초씩 옮긴다.
07 function restartVideo() { myVideo.currentTime = 0; } //재생 위치를 처음으로 옮긴다.
08 function volumeUp() { myVideo.volume += 0.1; } //볼륨을 0.1씩 올린다.
09 function volumeDown() { myVideo.volume -= 0.1; } //볼륨을 0.1씩 내린다.
10 function increaseSpeed() { myVideo.playbackRate += 0.1; } //재생 속도를 0.1씩 올린다.
11 function decreaseSpeed() { //재생 속도를 0.1씩 내린다.
12     if (myVideo.playbackRate <= 1) {
13         //재생 속도가 1보다 작으면 중간 속도로 변경한다.
14         myVideo.playbackRate = (myVideo.playbackRate / 2);
15     }
16     else { myVideo.playbackRate -= 0.1; }
17 }
18 function defaultSpeed() { myVideo.playbackRate = 1; } //기본 재생 속도로 초기화한다.
19 function swapVideo(e) {
20     if (myVideo.canPlayType("video/ogg") == 'maybe' ||
21         myVideo.canPlayType("video/ogg") == 'probably') {
22         myVideo.src = e.target.getAttribute('data-file')+'.ogg';
23     } else if (myVideo.canPlayType("video/mp4") == 'maybe' ||
24         myVideo.canPlayType("video/mp4") == 'probably') {
25         myVideo.src = e.target.getAttribute('data-file')+'.mp4';
26     }
27     myVideo.load();
28     playVideo();
29 }

```




비디오 제어하기



실습하기 - Video_Study01_SimplePlayer.html

Script	<pre> 30 function initEvents() { 31 myVideo = document.getElementById('player'); 32 btn = document.getElementById('play'); 33 video_list = document.getElementsByTagName('input'); 34 35 for(i = 0; i<video_list.length; i++) { 36 video_list[i].addEventListener('click', swapVideo, true); 37 } 38 } 39 //웹 문서가 로드되었을 때 이벤트를 초기화 하도록 해준다. 40 window.addEventListener("DOMContentLoaded", initEvents, false); </pre>
HTML	<pre> 01 <h2>HTML5 비디오 플레이어</h2> 02 <video width="600" controls id="player">비디오 요소를 지원하지 않음</video>
 03 <p> 04 <button id="play" onclick="playVideo();">재생 하기</button> 05 <button id="rewind" onclick="rewindVideo();">다시 감기</button> 06 <button id="restart" onclick="restartVideo();">다시 재생</button> 07 <button id="volumeUp" onclick="volumeUp();">볼륨 +</button> 08 <button id="volumeDown" onclick="volumeDown();">볼륨 -</button> 09 <button id="increaseSpeed" onclick="increaseSpeed();">재생 속도 +</button> 10 <button id="decreaseSpeed" onclick="decreaseSpeed();">재생 속도 -</button> 11 <button id="defaultSpeed" onclick="defaultSpeed();">기본 속도</button> 12 </p> 13 <input data-file="media/video_Wildlife" type="button" value="WildLife 보기"> </input> 14 <input data-file="media/bunny_small_ogg" type="button" value="Bunny 보기"> </input> </pre>



비디오 제어하기



실습하기 - Video_Study01_SimplePlayer.html

HTML5 비디오 플레이어



재생 하기

다시 감기

다시 재생

볼륨 +

볼륨 -

재생 속도 +

재생 속도 -

기본 속도

WildLife 보기

Bunny 보기



자막처리



track 요소

- 트랙(Track) 요소는 사용자에게 여러 언어나 비디오의 설명을 제공할 때, 시간이 지정된 텍스트 파일을 지정할 때 사용한다.

속성 값	설명
kind	track 요소의 상태를 나타내는 것으로, 생략할 수 있으며 기본 값은 subtitles(자막) 이다. 각 상태를 나타내는 키워드는 해당 예제의 설명을 참고하도록 한다.
src	텍스트 트랙 데이터의 주소를 나타낸다. 앞뒤로 공백이 들어갈 수 없는 유효한 URL이어야 하며, 반드시 지정해야 한다. 지정한 URL이 WebVTT(Web Video Text Track) 자원을 식별하고 kind 속성이 메타데이터(metadata) 상태에 있지 않으면, WebVTT 파일은 신호 텍스트를 사용하는 WebVTT가 된다. 지정한 URL이 WebVTT 자원을 식별하고 kind 속성이 장(chapters) 상태에 있으면, WebVTT 파일은 장 제목 텍스트를 사용하는 WebVTT 파일이 되고 오직 중첩된 신호만을 사용하는 WebVTT 파일이어야 한다.
srclang	텍스트 트랙 데이터의 언어를 제공한다. kind 속성이 자막 상태이면, 반드시 지정되어야 한다.
label	사용자가 읽을 수 있는 트랙의 제목을 제공한다. 브라우저에서는 자신의 인터페이스에 자막, 캡션, 오디오 설명 트랙을 열거할 때 이 제목을 사용해야 한다.
default	이 속성 값이 지정되면, 사용자가 다른 트랙을 별도로 지정하지 않는 경우, 기본 트랙이 활성화되어야 한다는 것을 나타낸다.



자막처리



track 요소

```
<video src="media/bunny_small_ogg.ogg" controls autoplay loop>
  <track src="media/bunny_small_ogg.vtt" srclang="ko" label="한국어" kind="subtitles" default>
</video>
```

키워드	상태	간단한 설명
subtitles	자막	대화를 글로 옮긴 것이나 번역한 것으로, 소리가 들리기는 하지만, 이해할 수 없을 때 적절하다. 비디오 위에 작성된다. ✓ 예: 사용자가 미디어 자원의 오디오에서 들리는 언어를 이해하지 못할 때
captions	캡션	대화를 글로 옮긴 것이나 번역한 것으로, 사운드 효과, 관련된 음악적 신호, 다른 관련된 오디오 정보로 소리가 들리지만, 또렷하게 알아들을 수 없을 때 적절하다. 비디오 위에 작성되며 청력이 약한 사람들에게 적합하다. ✓ 예: 소리가 묵음 처리되거나 주위 소음으로 상쇄되거나, 사용자가 청각 장애인일 때
descriptions	설명	미디어 자원의 비디오 구성 요소에 대한 텍스트 형식의 설명으로, 시각적 구성 요소가 알아보기 힘들거나, 없거나 사용할 수 없을 때 오디오 합성을 위한 것이다. 오디오로 합성된다. ✓ 예: 사용자가 운전 시에 애플리케이션과 상호작용할 때나 사용자가 앞을 못 볼 때
chapters	장	장의 제목으로 미디어 자원을 탐색하기 위한 목적으로 만든 것이다. 브라우저의 인터페이스에서 상호작용하는 목록으로 나타난다.
metadata	메타데이터	스크립트에서 사용하기 위해 만든 트랙이다. 브라우저에는 나타나지 않는다.



자막처리



track 요소

- 트랙 파일 형식 - 간소화된 WebVTT(Web Video Text Track) 텍스트 파일 형식 사용

WEBVTT

00:00.000 --> 00:07.031

지금 보고 계시는 것은 예제로 만든 자막입니다.

...

타이밍 신호는 HH:MM:SS.sss 형식

- 다중 트랙 파일

```
<video id="myVideo" controls autoplay loop>
  <source src="video.mp4" type="video/mp4">
  <track id="Track_en" src="video_en.vtt" label="영어" kind="subtitles" srclang="en" >
  <track id="Track_kr" src="video_ko.vtt" label="한국어" kind="subtitles" srclang="ko" default>
  <track id="Track_de" src="video_de.vtt" label="독일어" kind="subtitles" srclang="de">
</video>
```




■ 자막처리



track 요소

■ 트랙을 이용하여 비디오 재생하기

예제	Video_trackSimple.html
Script	<pre> 01 function loadCaptions(track) 02 { 03 var cues = track.track.cues; //트랙의 신호 객체를 가져온다 04 var list = document.getElementById('results'); //트랙의 텍스트들을 담은 객체를 가져온다. 05 06 for (i = 0; i < cues.length; i++) { 07 var track_text = cues[i].getCueAsHTML(); //트랙의 각 신호에 대한 텍스트를 가져온다. 08 var option = document.createElement("option"); //임의의 요소를 만들고 09 option.text = track_text.textContent; //임의의 요소에 가져온 텍스트를 지정한다. 10 //data-time 속성을 만들고 그 속성에 시간을 지정한다. 11 option.setAttribute('data-time', cues[i].startTime); 12 list.add(option); //목록 요소에 가져온 트랙 신호 텍스트를 추가한다. 13 } 14 } 15 function playCaption(control) { 16 var obj = control.options[control.options.selectedIndex]; //목록 요소의 선택된 아이템 객체 17 var track_cue = obj.getAttribute('data-time'); //아이템에 설정되어 있는 시간을 가져온다. 18 var video = document.getElementById('player'); 19 video.currentTime = track_cue - 0.1; //비디오에 가져온 시간으로 설정하여 재생하도록 한다. 20 } </pre>
HTML	<pre> 01 <video id="player" src="media/video_Wildlife.mp4" width="500px" controls autoplay loop> 02 <track src='media/video_caption1.vtt' srclang='ko' label='한국어' kind='subtitles' default 03 onload='loadCaptions(this)'> 04 </video> 05 <div><select size='5' id='results' onchange='playCaption(this)' style='width:500px; margin- top:8px' ></select></div> 06 <div>위 자막을 클릭하시면 자막이 있는 곳으로 이동하면서 재생됩니다.</div> </pre>



자막처리



track 요소

- 트랙을 이용하여 비디오 재생하기



※ track 요소의 경우는 로컬 파일시스템인 file://URL에서는 테스트를 해볼 수가 없고, 웹 서버에서 테스트를 해야만 한다(CORS; Cross-Origin Resource Sharing 문제).



다음 학습



- 1 오프라인 웹 애플리케이션 API
- 2 파일 API
- 3 웹 스토리지