```
<video width="320" height="240">
   <source src="video/1.mp4" type="video/mp4"> <!--경로는 source에서 지정한다, 대체소스들을 여러개 지정할 수 있다-->
   <source src="movie.ogg" type="video/ogg"><!--source들 중에서 브라우저가 처음 인식한 것을 사용한다(다음 사진에서 지원현황)-->
 Your browser does not support the video tag.<!--브라우저가 video태그를 지원하지 않으면 뜨는 텍스트이다-->
```

</video>

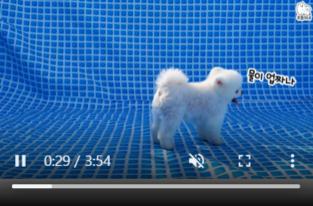
Note: Chromium browsers do not allow autoplay in most cases. However, muted autoplay is always allowed.

```
kvideo width="320" height="240" autoplay muted><!--진짜네-->
    <source src="video/1.mp4" type="video/mp4">
    <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
  </video>
```



```
kvideo width="320" height="240" autoplay muted controls
    <source src="video/1.mp4" type="video/mp4">
    <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
```

</video>



```
dir(document.guerySelector("video"))
▶ video
```

```
<video width="320" height="240" autoplay muted controls>
        <source src="video/1.mp4" type="video/mp4">
        <source src="movie.ogg" type="video/ogg">
      Your browser does not support the video tag.
      </video>
<script>
    const video = document.querySelector("video");
</script>
```

dir(video) ▶ video

```
▼ video 🚹
   accessKey: ""
   ariaAtomic: null
   ariaAutoComplete: null
   ariaBusy: null
   ariaChecked: null
   ariaColCount: null
   ariaColIndex: null
   ariaColSpan: null
   ariaCurrent: null
   ariaDescription: null
   ariaDisabled: null
   ariaExpanded: null
   ariaHasPopup: null
   ariaHidden: null
   ariaKeyShortcuts: null
   ariaLabel: null
   ariaLevel: null
   ariaLive: null
   ariaModal: null
   ariaMultiLine: null
   ariaMultiSelectable: null
   ariaOrientation: null
   ariaPlaceholder: null
   ariaPosInSet: null
   ariaPressed: null
   ariaReadOnly: null
   ariaRelevant: null
 Console What's New
                          Issues
```

```
autofocus: false
autoplay: true
baseURI: "file:///C:/Users/82102/Desktop/%EC%83%88%20%ED%8F%B4%EB%8D%94/1.html"
buffered: TimeRanges {length: 1}
childElementCount: 2
childNodes: NodeList(5) [text, source, text, source, text]
children: HTMLCollection(2) [source, source]
classList: DOMTokenList [value: '']
className: ""
clientHeight: 240
clientLeft: 0
clientTop: 0
clientWidth: 320
contentEditable: "inherit"
controls: true
controlsList: DOMTokenList [value: '']
crossOrigin: null
currentSrc: "file:///C:/Users/82102/Desktop/%EC%83%88%20%ED%8F%B4%EB%8D%94/video/1.mp4"
currentTime: 13.740023
dataset: DOMStringMap {}
defaultMuted: true
```

onmouseout: null onmouseover: null onmouseup: null onmousewheel: null onpaste: null onpause: null onplay: null onplaying: null onpointercancel: null onpointerdown: null onpointerenter: null onpointerleave: null onpointermove: null

OHIIIOUSEIIIOVE, HULL

```
const video = document.querySelector("video");
function method(){
video.addEventListener("play", method)//onxxx속성에서 on만 뺀 값이 파라미터로 들어감
```

```
outerHIML: "<video width=\"320\" he
 outerText:
▶ ownerDocument: document
▶ parentElement: body
▶ parentNode: body
▶ part: DOMTokenList [value: '']
 paused: false
 playbackRate: 1
▶ played: TimeRanges {length: 1}
 playsInline: false
 poster:
 nrefix: null
```

```
defaultPlaybackRate: 1
 dir:
 disablePictureInPicture: false
 disableRemotePlayback: false
 draggable: false
 duration: 234.451882
 elementTiming:
 ended: false
 enterKeyHint: ""
 error: null
▶ firstChild: text
```

```
translate: true
videoHeight: 720
videoWidth: 1280
virtualKevboardPolicv:
volume: 1
webkitAudioDecodedByteCount: 224748
webkitDecodedFrameCount: 416
webkitDisplayingFullscreen: false
webkitDroppedFrameCount: 0
webkitSupportsFullscreen: true
webkitVideoDecodedByteCount: 953986
width: 320
```

```
onfullscreenchange: (...)
onfullscreenerror: (...)
ongotpointercapture: (...)
oninput: (...)
oninvalid: (...)
onkevdown: (...)
onkeypress: (...)
onkeyup: (...)
onload: (...)
onloadeddata: (...)
onloadedmetadata: (...)
onloadstart: (...)
onlostpointercapture: (...)
onmousedown: (...)
onmouseenter: undefined
 nmouseleave: undefined
ODMOLICAMOVA: ( )
```

```
video.play();
video.pause();
video.requestFullscreen();
```

```
const videos = document.querySelectorAll(".video");
const videos2 = document.querySelectorAll(".video2");
//video가 추가 되었을 때 대비하여 querySelectorAll로 받음
//class = "video"(모든 비디오) : 구간조정, 볼륨조정, 전체화면, 단 하나재생
//class = "video2"(큰 거 하나) : 커지기, 색바꾸기
```

```
//add Event Lister"s"
for(let i=0;i<videos.length;i++){</pre>
    videos[i].addEventListener("keypress",fullScreen);
    videos[i].addEventListener("play", onlyOnePlay);
    videos[i].addEventListener("keydown",volumeAndTime);
for (let i = 0; i < videos2.length; i++) {
    videos2[i].addEventListener("click", sizeChange);
    videos2[i].addEventListener("play", colorChange);
    videos2[i].addEventListener("pause", colorChange);
```

```
function fullScreen(e){//전체화면
   const video = e.target;
   const paused = video.paused;
   if(e.keyCode==13){//enter
       //기존에 엔터키가 시작/정지(spacebar)기능을 하였고
       //preventDefault 안 먹히더라
       if(paused){
           video.play();
       else{//불가피
           video.pause();
       video.requestFullscreen();
```

```
function onlyOnePlay(e){//하나만 재생, 나머지 조용
   for(let i=0;i<videos.length;i++){</pre>
       const video = videos[i];
       if(video!=e.target){//방금 play눌린 것 빼고
           video.pause();//다 pause
```

```
function volumeAndTime(e){//방향키로 볼륨,구간 조정
   //keydown
   const key = e.keyCode;
   const VOLUME VALUE = 0.1;
   const video = e.target;
   const TIME VALUE = video.duration/10;//전체 영상 길이 기준으로
   //volume은 0~1.0사이,
   if(key==38&&video.volume<=0.9){//up
       video.volume+=VOLUME VALUE;
       return;
   //볼륨이랑 달리 오류가 안나더라
   if(key==37){//left
       video.currentTime -= TIME VALUE;
       return:
   if(key==39){//right
       video.currentTime += TIME VALUE;
       return:
   if(key==40&&video.volume>=0.1){//down
       video.volume-=VOLUME VALUE;
       return;
```

```
function colorChange(){
    //엘레먼트 보여주기
    document.body.classList.toggle("tomato");
//add Event Lister"s"
for(let i=0;i<videos.length;i++){</pre>
    videos[i].addEventListener("keypress",fullScreen);
    videos[i].addEventListener("play", onlyOnePlay);
    videos[i].addEventListener("keydown",volumeAndTime);
for (let i = 0; i < videos2.length; <math>i++) {
    videos2[i].addEventListener("click", sizeChange);
    videos2[i].addEventListener("play", colorChange);
    videos2[i].addEventListener("pause", colorChange);
```

.tomato{ background-color: =tomato;

```
function sizeChange(e){
   e.preventDefault();
   e.target.classList.toggle("full");
```