**Video tag** [**video**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/video)

* videoTag.play()
* videoTag.play()
* 'timeupdate' event // it is an event on every change in the time of video

media.addEventListener('timeupdate' , function() {

})

* videoTag.currentTime // in seconds, you can use math to calculate it in minutes, but you have to add attach a zero for less than 1 second

let minutes = Math.floor(time / 60);

let seconds = Math.floor(time - ( minutes \* 60 ))

* videoTag.duration //in seconds
* for progress-bar and audio bar you can use input range with an eventListener. For input type range it's better to use **input** event than **change**, change has some latency.

timerBar.addEventListener(**'input'** , function() {//input instead of change

media.currentTime = (this.value / 100) \* media.duration // value is in percentage of duration

})

* Fullscreen [Fullscreen\_API](https://developer.mozilla.org/en-US/docs/Web/API/Fullscreen_API) . We have to use moz and ms and webkit APIs to make it work with every browser.
  + Document.fullscreenElemnt
  + document.requestFullscreen
  + document.exitFullscreen

fullscreen.addEventListener('click' , function() {

console.log(document.fullscreenElement)

if (!document.fullscreenElement) {

if(playerArea.requestFullscreen) {

playerArea.requestFullscreen();

} else if(playerArea.mozFullScreenElement) {

playerArea.mozFullScreenElement()

} else if(playerArea.msFullscreenElement) {

playerArea.msFullscreenElement()

} else if(playerArea.webkitFullscreenElement) {

playerArea.webkitFullscreenElement()

}

} else {

if (document.exitFullscreen) {

document.exitFullscreen();

} else if(document.mozCancelFullscreen) {

document.mozCancelFullscreen();

} else if(document.msCancelFullscreen) {

document.msCancelFullscreen();

} else if(document.webkitCancelFullscreen) {

document.webkitCancelFullscreen();

}

}

})