2

Updated my answer to a better one using youtube player API for more option and dynamic output. It has many option to select and you could easily control it using Javascript. I hope this could help. Credits to @wasikuss posted answer.

// Load the IFrame Player API code asynchronously.

setTimeout(function() {

player.playVideo();

}, 20000);

var tag = document.createElement('script');

tag.src = "https://www.youtube.com/player\_api";

var firstScriptTag = document.getElementsByTagName('script')[0];

firstScriptTag.parentNode.insertBefore(tag, firstScriptTag);

// Replace the 'ytplayer' element with an <iframe> and

// YouTube player after the API code downloads.

var player;

function onYouTubePlayerAPIReady() {

player = new YT.Player('ytplayer', {

height: '315',

width: '560',

videoId: 'iejOAPyooXs'

});

}

Reference <https://developers.google.com/youtube/player_parameters>

<iframe id="ytplayer" type="text/html" width="640" height="360"

src="https://www.youtube.com/embed/M7lc1UVf-VE?autoplay=1&origin=http://example.com"

frameborder="0"></iframe>

<div id="ytplayer"></div>

<script>

// Load the IFrame Player API code asynchronously.

var tag = document.createElement('script');

tag.src = "https://www.youtube.com/player\_api";

var firstScriptTag = document.getElementsByTagName('script')[0];

firstScriptTag.parentNode.insertBefore(tag, firstScriptTag);

// Replace the 'ytplayer' element with an <iframe> and

// YouTube player after the API code downloads.

var player;

function onYouTubePlayerAPIReady() {

player = new YT.Player('ytplayer', {

height: '360',

width: '640',

videoId: 'M7lc1UVf-VE'

});

}

</script>

Eg 2

<div id="player"></div>  
  
    <script>  
      // 2. This code loads the IFrame Player API code asynchronously.  
      var tag = document.createElement('script');  
  
      tag.src = "https://www.youtube.com/iframe\_api";  
      var firstScriptTag = document.getElementsByTagName('script')[0];  
      firstScriptTag.parentNode.insertBefore(tag, firstScriptTag);  
  
      // 3. This function creates an <iframe> (and YouTube player)  
      //    after the API code downloads.  
      var player;  
      function onYouTubeIframeAPIReady() {  
        player = new YT.Player('player', {  
          height: '390',  
          width: '640',  
          videoId: 'M7lc1UVf-VE',  
          events: {  
            'onReady': onPlayerReady,  
            'onStateChange': onPlayerStateChange  
          }  
        });  
      }  
  
      // 4. The API will call this function when the video player is ready.  
      function onPlayerReady(event) {  
        event.target.playVideo();  
      }  
  
      // 5. The API calls this function when the player's state changes.  
      //    The function indicates that when playing a video (state=1),  
      //    the player should play for six seconds and then stop.  
      var done = false;  
      function onPlayerStateChange(event) {  
        if (event.data == YT.PlayerState.PLAYING && !done) {  
          setTimeout(stopVideo, 6000);  
          done = true;  
        }  
      }  
      function stopVideo() {  
        player.stopVideo();  
      }

Mute

player.mute():Void

Setting the player size

player.setSize(width:Number, height:Number):Object

Sets the size in pixels of the <iframe> that contains the player.

**Example 3:**This example sets player parameters to automatically play the video when it loads and to hide the video player's controls. It also adds event listeners for several events that the API broadcasts.

function onYouTubeIframeAPIReady() {  
  var player;  
  player = new YT.Player('player', {  
    videoId: 'M7lc1UVf-VE',  
    playerVars: { 'autoplay': 1, 'controls': 0 },  
    events: {  
      'onReady': onPlayerReady,  
      'onStateChange': onPlayerStateChange,  
      'onError': onPlayerError  
    }  
  });  
}

**Example 2: Loud playback**

This example creates a 1280px by 720px video player. The event listener for the onReady event then calls the [setVolume](https://developers.google.com/youtube/iframe_api_reference" \l "setVolume) function to adjust the volume to the highest setting.

function onYouTubeIframeAPIReady() {  
  var player;  
  player = new YT.Player('player', {  
    width: 1280,  
    height: 720,  
    videoId: 'M7lc1UVf-VE',  
    events: {  
      'onReady': onPlayerReady,  
      'onStateChange': onPlayerStateChange,  
      'onError': onPlayerError  
    }  
  });  
}  
  
function onPlayerReady(event) {  
  event.target.setVolume(100);  
  event.target.playVideo();  
}

This code will work. Just set your time in milliseconds and write your JS code on loadAfterTime function:

<script>

window.onload = function(){

setTimeout(loadAfterTime, 5000)

};

function loadAfterTime() {

// code you need to execute goes here.

}

</script>

‘onReady’: function

}

});

}

window.onload = function(event){

setTimeout(loadAfterTime, 20000)

};

function loadAfterTime() {

// code you need to execute goes here.

function onPlayerReady(event) {  
  event.target.playVideo();  
}

}

<iframe class="delayed" width="486" height="273" frameborder="0" data-src="http://www.youtube.com/embed/I\_V\_kIzKKqM?enablejsapi=1&amp;version=3&amp;rel=0&amp;autoplay=1&amp;showsearch=0&amp;autohide=1&amp;border=0&amp;showinfo=0" allowfullscreen=""></iframe>

setTimeout(function() { $('iframe.delayed' delayed').attr('data-src')); }, 20000);).attr('src', $('iframe.

setTimeout(function() { $('iframe.delayed').attr('src', $('iframe.delayed').attr('data-src')); }, 5000);fs