

Custom HTML Video Player

Creating a Custom Video Player in HTML

We can **create a custom HTML video player** using the HTML `<video>` tag, some other HTML tags for design, CSS, and JavaScript. HTML features include native video support without the need for Flash.

In this tutorial, we will learn how we can **create a video player in HTML**. The code provided in this tutorial works based on HTML, CSS, and JavaScript. You can drag and drop your local video files into the container.

We recommend going through the chapter to learn about [embedding Video Player in HTML](#).

Complete Code to Create a Custom HTML Video Player

Let's look at the following example, where we are going to allow the user to upload the local video file:

[Open Compiler](#)

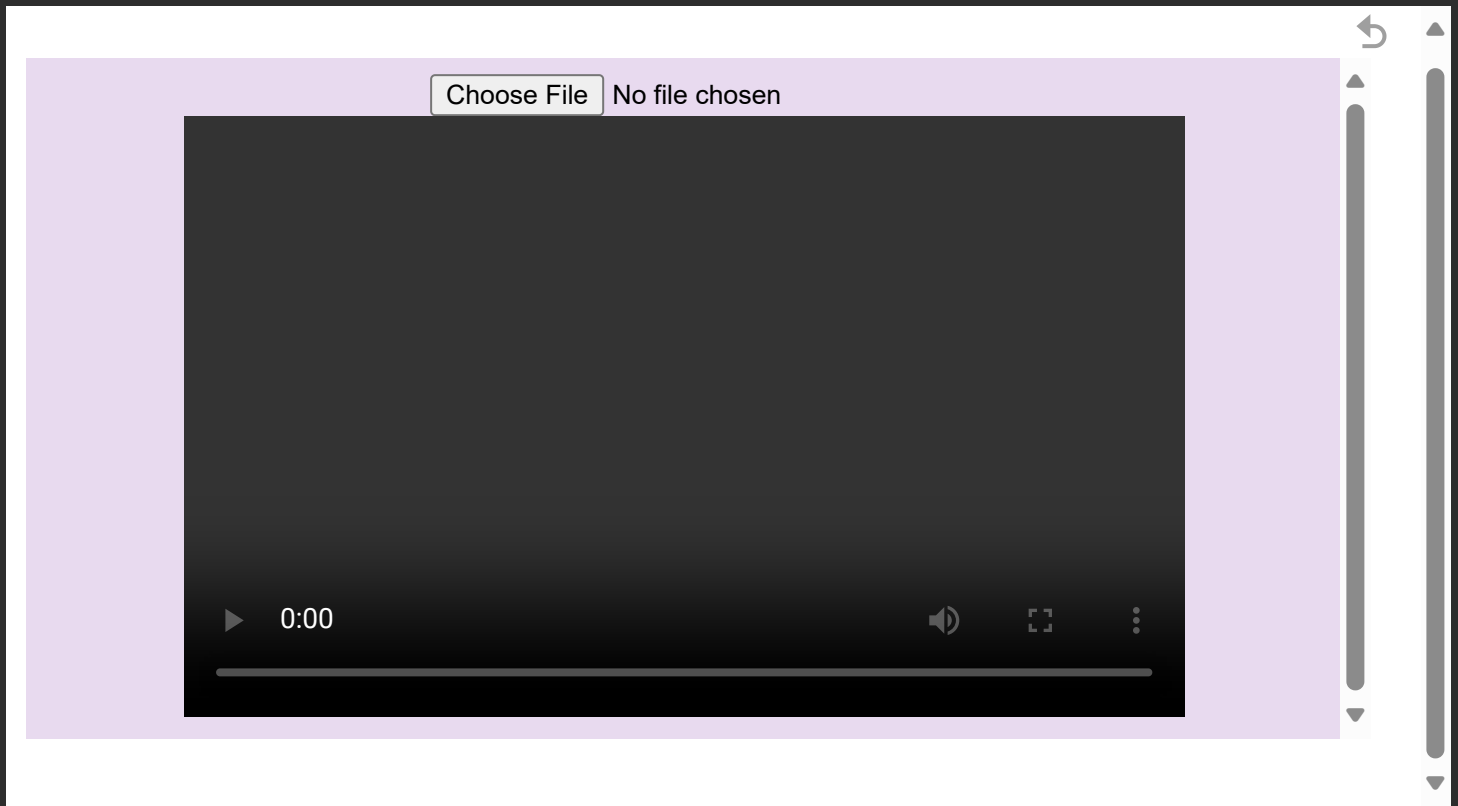
```
<!DOCTYPE html>
<html>
<head>
  <style>
    body {
      background-color: #E8DAEF;
      text-align: center;
    }
  </style>
</head>
<body>
  <input type="file" accept="video/*">
  <br>
  <video controls height="300" width="500"></video>
  <script>
    (function localFileVideoPlayer() {
      'WELCOME'
      var x = function(event) {
        var vid = this.files[0]
        var a = window.a || window.webkitURL
        var y = a.createObjectURL(vid)
        var videoNode = document.querySelector('video')
        videoNode.src = y
      }
    })
  </script>

```

```

    var z = document.querySelector('input')
    z.addEventListener('change', x, false)
  })()
</script>
</body>
</html>

```



When we run the above code, it will generate an output displaying the video controls, allowing the user to upload the local video file on the webpage.

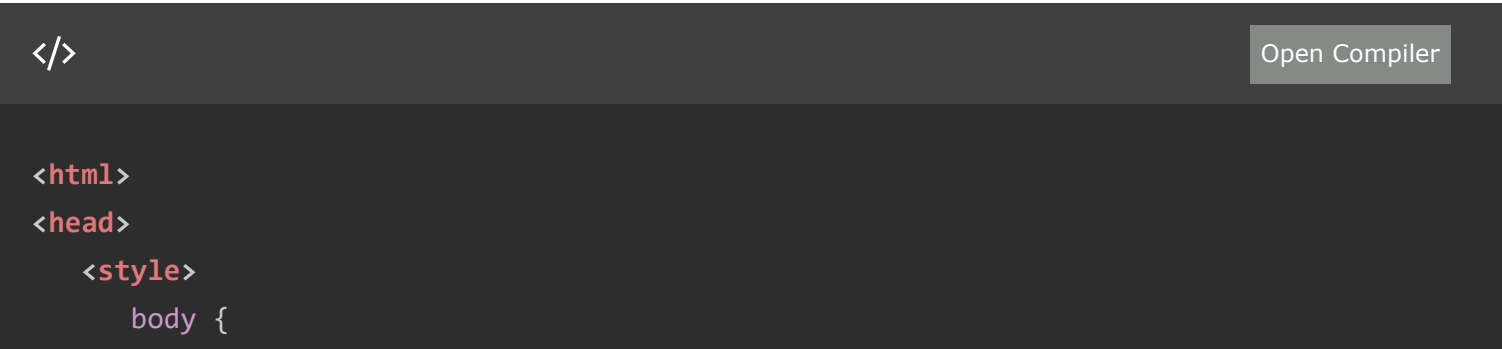
Explore our **latest online courses** and learn new skills at your own pace. Enroll and become a certified expert to boost your career.

Capture and Play Video Using Camera Accessibility

Below is a recorder that works based on HTML, CSS, and JavaScript. Before entering this page, the user must allow camera accessibility to cost images.

Example

Consider the following example, where we are going to use the user camera and capture the image:



```

        text-align: center;
        background-color: #EAF1F5;
    }
</style>
</head>
<body>
    <video id="vid" controls autoplay></video>
    <canvas id="mytutorial" width="600" height="300" style="border:1px solid #d3d3d3;">
</canvas>
    <button id="snapshot">Take Picture</button>
    <script>
        const x = document.getElementById('vid');
        const mycanvas = document.getElementById('mytutorial');
        const y = mycanvas.getContext('2d');
        const clickpicture = document.getElementById('snapshot');
        const a = {
            video: true,
        };
        clickpicture.addEventListener('click', () => {
            y.drawImage(x, 1, 0, mycanvas.width, mycanvas.height);
        });
        navigator.mediaDevices.getUserMedia(a).then((stream) => {
            x.srcObject = stream;
        });
    </script>
</body>
</html>

```

On running the above code, the output window will pop up, displaying the canvas along with a capture button on the webpage.