HTML Audio Player with Visualizer

Here, we will learn how to create a custom audio player with a visualizer in HTML.

HTML Audio Player with Visualizer

HTML features include native audio and video support without the need for Flash. The below code works based on HTML, CSS, and JavaScript. You can drag and drop your local MP3 files into the container.

We recommend going through these chapters to learn about embedding audio in HTML and the **<audio>** tag.

Code Files to Create HTML Audio Player

Let's look at the following example, where we are going to create a local audio visualizer

1. HTML File

```
<!DOCTYPE html>
<html>
   <style>
      #instructions {
         width: 100%;
         text-align: center;
         top: 50%;
         margin-top: -100px;
         color: #DE3163;
      }

      #container {
         position: absolute;
         width: 100%;
         height: 100%;
         background: #D5F5E3;
      }

      #canvas-container {
         width: 600px;
         height: 600px;
         margin: auto;
         position: relative;
         top: 50%;
         margin-top: -263px;
         margin-right: -61px;
```

```
      }

      #canvas-copy {
         opacity: 0.05;
         -webkit-transform: scaleY(-1);
         margin-top: -6px;
      }
   </style>
<body>
   <div id="container">
      <div id="canvas-container">
         <canvas width=600 height=300 id="canvas"></canvas>
         <canvas width=600 height=300 id="canvas-copy"></canvas>
      </div>
      <div id="instructions">
         <a href="https://www.tutorialspoint.com/index.htm" align="center"> Tutorials
Point</a>
         <h2 style="font-family:verdana"> Drag Your Local MP3 Files </h2>
      </div>
      <div id="button"></div>
   </div>
   <script src="js.html"></script>
</body>
</html>
```

Now, we are going to create a JavaScript file with the name mentioned in the above HTML file.

2. JavaScript File

```
<script>
   (function() {
      var requestAnimationFrame = window.requestAnimationFrame ||
window.mozRequestAnimationFrame || window.webkitRequestAnimationFrame ||
window.msRequestAnimationFrame;
      window.requestAnimationFrame = requestAnimationFrame;
   })();
   window.onload = function() {
      var element = document.getElementById('container')
      dropAndLoad(element, init, "ArrayBuffer")
   }

   function dropAndLoad(dropElement, callback, readFormat) {
      var readFormat = readFormat || "DataUrl"
      dropElement.addEventListener('dragover', function(e) {
         e.stopPropagation()
```

```
        e.preventDefault()
        e.dataTransfer.dropEffect = 'copy'
    }, false)
    dropElement.addEventListener('drop', function(e) {
        e.stopPropagation()
        e.preventDefault()
        loadFile(e.dataTransfer.files[0])
    }, false)

    function loadFile(files) {
        var file = files
        var reader = new FileReader()
        reader.onload = function(e) {
            callback(e.target.result)
        }
        reader['readAs' + readFormat](file)
    }
}

function init(arrayBuffer) {
    document.getElementById('instructions').innerHTML = 'Audio Loading'
    window.audioCtx = new AudioContext()
    window.analyser = audioCtx.createAnalyser()
    if (window.source) source.noteOff(0)
    audioCtx.decodeAudioData(arrayBuffer, function(buffer) {
        window.source = audioCtx.createBufferSource()
        source.buffer = buffer
        source.connect(analyser)
        analyser.connect(audioCtx.destination)
        source.start(0)
        var viz = new simpleViz()
        new visualizer(viz['update'], analyser)
        document.getElementById('instructions').innerHTML = ''
    })
}

function visualizer(visualization, analyser) {
    var self = this
    this.visualization = visualization
    var last = Date.now()
    var loop = function() {
        var dt = Date.now() - last
        var byteFreq = new Uint8Array(analyser.frequencyBinCount)
        analyser.getByteFrequencyData(byteFreq)
        last = Date.now()
        self.visualization(byteFreq, dt)
```

```
            requestAnimationFrame(loop)
        }
        requestAnimationFrame(loop)
    }

    function simpleViz(canvas) {
        var self = this
        this.canvas = document.getElementById('canvas')
        this.ctx = this.canvas.getContext("2d")
        this.copyCtx = document.getElementById('canvas-copy').getContext("2d")
        this.ctx.fillStyle = '#fff'
        this.barWidth = 10
        this.barGap = 4
        this.bars = Math.floor(this.canvas.width / (this.barWidth + this.barGap))
        this.update = function(byteFreq) {
            self.ctx.clearRect(0, 0, self.canvas.width, self.canvas.height)
            var step = Math.floor(byteFreq.length / self.bars)
            for (var i = 0; i < self.bars; i++) {
                var barHeight = byteFreq[i * step]
                self.ctx.fillRect(i * (self.barWidth + self.barGap), self.canvas.height -
barHeight, self.barWidth, barHeight)
                self.copyCtx.clearRect(0, 0, self.canvas.width, self.canvas.height)
                self.copyCtx.drawImage(self.canvas, 0, 0)
            }
        }
    }
</script>
```


Complete Code to Create HTML Audio Player

Let's combine both the file and observe the output we are going to get.

</> Open Compiler

```
<!DOCTYPE html>
<html>
    <style>
        #instructions {
            width: 100%;
            text-align: center;
            top: 50%;
            margin-top: -100px;
            color: #DE3163;
```

```
      }

      #container {
         position: absolute;
         width: 100%;
         height: 100%;
         background: #D5F5E3;
      }

      #canvas-container {
         width: 600px;
         height: 600px;
         margin: auto;
         position: relative;
         top: 50%;
         margin-top: -263px;
         margin-right: -61px;
      }

      #canvas-copy {
         opacity: 0.05;
         -webkit-transform: scaleY(-1);
         margin-top: -6px;
      }
   </style>
<body>
   <div id="container">
      <div id="canvas-container">
         <canvas width=600 height=300 id="canvas"></canvas>
         <canvas width=600 height=300 id="canvas-copy"></canvas>
      </div>
      <div id="instructions">
         <a href="https://www.tutorialspoint.com/index.htm" align="center"> Tutorials
Point</a>
         <h2 style="font-family:verdana"> Drag Your Local MP3 Files </h2>
      </div>
      <div id="button"></div>
   </div>
   <script>
      (function() {
         var requestAnimationFrame = window.requestAnimationFrame ||
window.mozRequestAnimationFrame || window.webkitRequestAnimationFrame ||
window.msRequestAnimationFrame;
         window.requestAnimationFrame = requestAnimationFrame;
      })();
      window.onload = function() {
```

```
    var element = document.getElementById('container')
    dropAndLoad(element, init, "ArrayBuffer")
}

function dropAndLoad(dropElement, callback, readFormat) {
    var readFormat = readFormat || "DataUrl"
    dropElement.addEventListener('dragover', function(e) {
        e.stopPropagation()
        e.preventDefault()
        e.dataTransfer.dropEffect = 'copy'
    }, false)
    dropElement.addEventListener('drop', function(e) {
        e.stopPropagation()
        e.preventDefault()
        loadFile(e.dataTransfer.files[0])
    }, false)

    function loadFile(files) {
        var file = files
        var reader = new FileReader()
        reader.onload = function(e) {
            callback(e.target.result)
        }
        reader['readAs' + readFormat](file)
    }
}

function init(arrayBuffer) {
    document.getElementById('instructions').innerHTML = 'Audio Loading'
    window.audioCtx = new AudioContext()
    window.analyser = audioCtx.createAnalyser()
    if (window.source) source.noteOff(0)
    audioCtx.decodeAudioData(arrayBuffer, function(buffer) {
        window.source = audioCtx.createBufferSource()
        source.buffer = buffer
        source.connect(analyser)
        analyser.connect(audioCtx.destination)
        source.start(0)
        var viz = new simpleViz()
        new visualizer(viz['update'], analyser)
        document.getElementById('instructions').innerHTML = ''
    })
}

function visualizer(visualization, analyser) {
    var self = this
```

```
            this.visualization = visualization
            var last = Date.now()
            var loop = function() {
               var dt = Date.now() - last
               var byteFreq = new Uint8Array(analyser.frequencyBinCount)
               analyser.getByteFrequencyData(byteFreq)
               last = Date.now()
               self.visualization(byteFreq, dt)
               requestAnimationFrame(loop)
            }
            requestAnimationFrame(loop)
         }

         function simpleViz(canvas) {
            var self = this
            this.canvas = document.getElementById('canvas')
            this.ctx = this.canvas.getContext("2d")
            this.copyCtx = document.getElementById('canvas-copy').getContext("2d")
            this.ctx.fillStyle = '#fff'
            this.barWidth = 10
            this.barGap = 4
            this.bars = Math.floor(this.canvas.width / (this.barWidth + this.barGap))
            this.update = function(byteFreq) {
               self.ctx.clearRect(0, 0, self.canvas.width, self.canvas.height)
               var step = Math.floor(byteFreq.length / self.bars)
               for (var i = 0; i < self.bars; i++) {
                  var barHeight = byteFreq[i * step]
                  self.ctx.fillRect(i * (self.barWidth + self.barGap), self.canvas.height
- barHeight, self.barWidth, barHeight)
                  self.copyCtx.clearRect(0, 0, self.canvas.width, self.canvas.height)
                  self.copyCtx.drawImage(self.canvas, 0, 0)
               }
            }
         }
      </script>
   </body>
</html>
```

When we run the above code, it will generate an output consisting of the text applied with CSS indicating to drag the local MP3 file to play music.

HTML Audio Player with Playlist

Consider the following example, where we are allowing the user to upload the multiple local MP3s that act as a playlist.

</>

Open Compiler

```
<!DOCTYPE html>
<html>
<body style="background-color:#ABEBC6;">
   <audio controls id="y" autoplay></audio>
   <br>
   <br>
   <br>
   <input type="file" id="x" multiple>
   <script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js">
</script>
   <script>
      var x = document.getElementById("x"),
         y = document.getElementById("y");

      function next(n) {
         var a = URL.createObjectURL(z[n]);
         y.setAttribute('src', a);
         y.play();
      }
      var _next = 0,
         z,
         len;
      x.addEventListener('Change', function() {
         z = x.z;
         len = z.length;
         if (len) {
            next(_next);
         }
      });
      y.addEventListener("Completed", function() {
         _next += 1;
         next(_next);
         console.log(len, _next);
         if ((len - 1) == _next) {
            _next = -1;
         }
      });
   </script>
</body>
</html>
```

On running the above code, the output window will pop up, allowing the user to upload multiple mp3 files and play them automatically on the webpage.