

知识点:

音频

想要使用createAudio页面只能在服务器环境下运行

案例效果图:

代码:

```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <style>
5        *{
6          margin: 0;
7          padding: 0;
8        }
9        .controls {
10         display: flex;
11         justify-content: space-around;
12         align-items: center;
13         margin-bottom: 16px;
14       }
15       .music-wrapper {
16         position: relative;
17         display: flex;
18         flex-direction: row;
19         justify-content: space-between;
20         list-style: none;
21         width: 100%;
22         height: 148px;
23         padding: 16px;
24         border-top: 1px solid #ccc;
25       }
26       .music-draw{
27         width: 20px;
28         height: 50px;
29         transition: .5s;
30       }
31       #controll{
32         position: absolute;
33         top: 50%;
34         left: 50%;
```

```

35         transform: translate(-50%, -50%);
36         padding: 8px;
37         background-color: #64c239;
38         border: none;
39         color: #fff;
40         width: 80px;
41         height: 80px;
42         border-radius: 50%;
43         cursor: pointer;
44         border: 2px solid #ccc;
45     }
46     #controll:hover{
47         opacity: .8;
48     }
49 </style>
50 </head>
51 <body>
52     <!-- 控制音量 -->
53     <div class="controlls">
54         <div>
55             <h3>控制音量</h3>
56             <input type="range" id="volume" min="0" max="2" value="1"
step="0.01">
57         </div>
58         <div>
59             <h3>控制立体声方向</h3>
60             <input type="range" id="panner" min="-1" max="1" value="0"
step="0.01">
61         </div>
62     </div>
63     <ul class="music-wrapper">
64         <button id="controll" data-playing="false">播放</button>
65     </ul>
66     <!-- 如果你要加载的声音文件保留在其他域中，则需要使用 crossorigin 属性 -->
67     <audio src="./audio.mp3" type="audio/mpeg" id="source"></audio>
68     <script>
69         const pillarNumber = 128;
70         /** 创建负责动效的长方形的个数 */
71         const ulWrapper = document.querySelector(".music-wrapper");
72         new Array(pillarNumber).fill("").forEach(()=>{
73             const li = document.createElement("li");
74             li.classList.add("music-draw");
75             li.style.backgroundColor = "#"+new Array(6).fill(0).map(()=>
[0,1,2,3,4,5,6,7,8,9,"a","b","c","d","e","f"]
[Math.floor(Math.random()*16)]).join("");
76             ulWrapper.appendChild(li);
77         });
78         /** 创建音频上下文 */
79         const AudioContext = window.AudioContext ||
window.webkitAudioContext;
80         const audioContext = new AudioContext();
81
82         /** 获取音频源 */

```

```

83     const audioElement = document.querySelector("#source");
84     /** 讲音频源放在音频上下文中 */
85     const track = audioContext.createMediaElementSource(audioElement);
86
87     /** 修改音量 */
88     const gainNode = audioContext.createGain();
89
90     /** 为应用程序增加立体声平移 */
91     const pannerOptions = { pan: 0 };
92     const panner = new StereoPannerNode(audioContext, pannerOptions);
93
94     /** 获取数据分析 */
95     const analyser = audioContext.createAnalyser();
96
97     /** 连接 */
98
99     track.connect(gainNode).connect(panner).connect(analyser).connect(audioContext.
100     destination);
101
102     /** 控制播放 */
103     const controllBtn = document.querySelector("#controll");
104     controllBtn.onclick = function () {
105         if(audioContext.state === "suspended"){
106             audioContext.resume();
107         }
108
109         if(this.dataset.playing === "false"){
110             audioElement.play();
111             this.dataset.playing = "true";
112             this.innerHTML = "暂停";
113             this.style.backgroundColor = "#e61021";
114         }else {
115             audioElement.pause();
116             this.dataset.playing = "false";
117             this.innerHTML = "播放";
118             this.style.backgroundColor = "#64c239";
119         }
120     }
121
122     /** 音频播放结束，将状态更新 */
123     audioElement.addEventListener("ended", ()=>{
124         controllBtn.dataset.playing = "false";
125     });
126
127     /** 修改音量 */
128     const volumeInput = document.querySelector("#volume");
129     volumeInput.oninput = function () {
130         gainNode.gain.value = this.value;
131     }
132
133     /** 立体声平移 */
134     const pannerInput = document.querySelector("#panner");

```

```

134     pannerInput.oninput = function (){
135         panner.pan.value = this.value;
136     }
137
138
139     /** 获取音频数据 */
140     function getAudioData (){
141         const freqArray = new Uint8Array(analyser.frequencyBinCount);
142         analyser.getByteFrequencyData(freqArray);
143         const step = Math.round(freqArray.length / pillarNumber);
144         const results = freqArray.slice(0,pillarNumber).map(i => i *
step);
145
146         if(audioElement.currentTime >= audioElement.duration){
147             audioElement.currentTime = 0;
148         }
149
150         results.currentTime = audioElement.currentTime ;
151         results.duration = audioElement.duration;
152
153         return results;
154     }
155     const musicDraws = document.querySelectorAll(".music-draw");
156
157     const timer = setInterval(()=>{
158         if(controllBtn.dataset.playing === "true"){
159             const data = getAudioData();
160             console.log('data:',data);
161             musicDraws.forEach((item,index)=>{
162                 item.style.height = 20+Number(data[index])+ "px";
163             })
164         }
165     },20)
166
167     </script>
168 </body>
169 </html>

```