



黑客派

js 音乐播放器的实现以及可视化

作者: [maixiaojie](#)

原文链接: <https://hacpai.com/article/1562738358681>

来源网站: [黑客派](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

```
<p></p>
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></scr
pt>
<!-- 黑客派PC帖子内嵌-展示 -->
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342"
data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></in
>
<script>
  (adsbygoogle = window.adsbygoogle || []).push({});
</script>
<p></p>
<h2 id="前言">前言</h2>
<p>我们尝试用原生 js 写一个音乐播放器，对音频做一个简单的可视化。</p>
<p>最终的效果如下图：</p>
<p></p>
<p>思路还是比较明显的，第一，我们要拿到音频的什么数据进行可视化，如何获取？第二，如何可
化，第二个问题就比较简单了，用 canvas 来操作。</p>
<p>获取音频相关的数据可以用 Web Audio API，这里我们可以去 MDN 或者 W3C 上查看相关的
档。</p>
<h2 id="audio-API">audio API</h2>
<p>我们都知道在 Web 中播放音频用 <code>audio</code> 这个标签来操作。Web Audio API
对 <code>audio</code> 标签功能上的补充，它可以：</p>
<ol>
<li>一秒内同时发出多个声音</li>
<li>把音频流独立出来，进行复杂操作</li>
<li>将音频输出到多个地方，比如 canvas，从而实现可视化</li>
</ol>
<p>audio node 有入口和出口，多个节点构成类似链表一样的结构，从一个或者多个音源出发，经
一个或者多个处理节点，最终输出到输出节点。Web audio 的一个简单处理过程如下：</p>
<ol>
<li>创建 AudioContext 对象</li>
<li>在 AudioContext 对象内设置音源，例如
<code><audio preload="none">
  标签
</code></li>
<li>创建 affcet node(效果节点)</li>
<li>选择音频的最终输出节点</li>
<li>音频经过效果节点处理后，然后输入到下一个节点</li>
</ol>
<pre><code class="language-js highlight-chroma"><span class="highlight-c1">// 获取Audi
Context对象
</span><span class="highlight-c1"></span><span class="highlight-nb">window</span><
span class="highlight-p">.</span><span class="highlight-nx">AudioContext</span> <span c
lass="highlight-o">=</span><span class="highlight-nb">window</span><span class="high
light-p">.</span><span class="highlight-nx">AudioContext</span> <span class="highlight
o">||</span><span class="highlight-nb">window</span><span class="highlight-p">.</sp
n><span class="highlight-nx">webkitAudioContext</span> <span class="highlight-o">||</
pan> <span class="highlight-nb">window</span><span class="highlight-p">.</span><spa
class="highlight-nx">mozAudioContext</span> <span class="highlight-p">;</span>
<span class="highlight-c1">// 创建AudioContext对象实例
</pre>
```

```
</span><span class="highlight-c1"></span><span class="highlight-kd">var</span> <span class="highlight-nx">actx</span> <span class="highlight-o">=</span> <span class="highlight-k">new</span> <span class="highlight-nx">AudioContext</span><span class="highlight-p">(</span><span class="highlight-c1">// 创建AudioSource, 此处的audios为<&lt;audio>&gt;标签</span><span class="highlight-c1"></span><span class="highlight-kd">var</span> <span class="highlight-nx">audioSrc</span> <span class="highlight-o">=</span> <span class="highlight-nx">actx</span><span class="highlight-p">.</span><span class="highlight-nx">createMediaElementSource</span><span class="highlight-p">(</span><span class="highlight-nx">audios</span><span class="highlight-p">);</span><span class="highlight-kd">var</span> <span class="highlight-nx">analyser</span> <span class="highlight-o">=</span> <span class="highlight-nx">actx</span><span class="highlight-p">.</span><span class="highlight-nx">createAnalyser</span><span class="highlight-p">(</span><span class="highlight-c1">// 输入的连接</span><span class="highlight-c1"></span><span class="highlight-nx">audioSrc</span><span class="highlight-p">.</span><span class="highlight-nx">connect</span><span class="highlight-p">(</span><span class="highlight-nx">analyser</span><span class="highlight-p">);</span><span class="highlight-c1">// 输出的连接</span><span class="highlight-c1"></span><span class="highlight-nx">analyser</span><span class="highlight-p">.</span><span class="highlight-nx">connect</span><span class="highlight-p">(</span><span class="highlight-nx">actx</span><span class="highlight-p">.</span><span class="highlight-nx">destination</span><span class="highlight-p">);</span><span class="highlight-c1">// 本例子中我们获取音高数据来进行可视化</span><span class="highlight-c1"></span><span class="highlight-kd">var</span> <span class="highlight-nx">voiceHigh</span> <span class="highlight-o">=</span> <span class="highlight-k">new</span> <span class="highlight-nx">Uint8Array</span><span class="highlight-p">(</span><span class="highlight-nx">analyser</span><span class="highlight-p">.</span><span class="highlight-nx">frequencyBinCount</span><span class="highlight-p">);</span><span class="highlight-nx">analyser</span><span class="highlight-p">.</span><span class="highlight-nx">getByteFrequencyData</span><span class="highlight-p">(</span><span class="highlight-nx">voiceHigh</span><span class="highlight-p">);</span></code></pre><h2 id="canvas">canvas</h2><p>用 canvas 对音高数据进行可视化展示。 </p><pre><code class="language-js highlight-chroma"><span class="highlight-kd">function</span><span class="highlight-nx">draw</span><span class="highlight-p">()</span> <span class="highlight-p">{</span><span class="highlight-kd">var</span> <span class="highlight-nx">step</span> <span class="highlight-o">=</span> <span class="highlight-nb">Math</span><span class="highlight-p">.</span><span class="highlight-nx">round</span><span class="highlight-p">(</span><span class="highlight-nx">voiceHigh</span><span class="highlight-p">.</span><span class="highlight-nx">length</span><span class="highlight-p">/</span><span class="highlight-nx">num</span><span class="highlight-p">);</span><span class="highlight-nx">ctx</span><span class="highlight-p">.</span><span class="highlight-nx">clearRect</span><span class="highlight-p">(</span><span class="highlight-nx">0</span><span class="highlight-p">,</span><span class="highlight-nx">0</span><span class="highlight-p">,</span><span class="highlight-nx">canvas</span><span class="highlight-p">.</span><span class="highlight-nx">width</span><span class="highlight-p">,</span><span class="highlight-nx">canvas</span><span class="highlight-p">.</span><span class="highlight-nx">height</span><span class="highlight-p">);</span><span class="highlight-nx">ctx</span><span class="highlight-p">.</span><span class="highlight-nx">clearRect</span><span class="highlight-p">(</span><span class="highlight-nx">0</span><span class="highlight-p">,</span><span class="highlight-nx">0</span><span class="highlight-p">,</span><span class="highlight-nx">canvas</span><span class="highlight-p">.</span><span class="highlight-nx">width</span><span class="highlight-p">,</span><span class="highlight-nx">canvas</span><span class="highlight-p">.</span><span class="highlight-nx">height</span><span class="highlight-p">);</span></code></pre>
```

```
highlight-nx">beginPath</span> <span class="highlight-p">());</span>
  <span class="highlight-k">for</span> <span class="highlight-p">( </span> <span class="highlight-kd">var</span> <span class="highlight-nx">i</span> <span class="highlight-o">=
/ <span class="highlight-mi">0</span> <span class="highlight-p">;</span> <span class="highlight-nx">i</span> <span class="highlight-o">&lt;</span> <span class="highlight-nx">num</span> <span class="highlight-p">;</span> <span class="highlight-nx">i</span> <span class="highlight-o">++</span> <span class="highlight-p">)</span> <span class="highlight-p">{</span>
  <span class="highlight-kd">var</span> <span class="highlight-nx">value</span> <span class="highlight-o">= </span> <span class="highlight-nx">voiceHigh</span> <span class="highlight-p">[ </span> <span class="highlight-nx">step</span> <span class="highlight-o">*</span> <span class="highlight-nx">i</span> <span class="highlight-p">];</span> </span>
```

```
<span class="highlight-p">}</span>
```

```
</code> </pre>
```

```
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></script>
```

```
<!-- 黑客派PC帖子内嵌-展示 -->
```

```
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342" data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></ins>
```

```
<script>
  (adsbygoogle = window.adsbygoogle || []).push({});
</script>
```

```
<h2 id="进度条">进度条</h2>
```

<p>进度条算是比较简单好做的了，我们用一个背景 div 填充，另外一个 div 表示当前进度，宽度是动态变化的，根据音频播放的当前时间除以总时间，然后乘背景 div 的宽度，计算得到结果即可。</p>

```
<pre> <code class="language-js highlight-chroma"> <span class="highlight-kd">var</span>
<span class="highlight-nx">bglen</span> <span class="highlight-o">= </span> <span class="highlight-nb">document</span> <span class="highlight-p">.</span> <span class="highlight-nx">getElementById</span> <span class="highlight-p">(</span> <span class="highlight-p">.</span> <span class="highlight-nx">offsetW
dth</span> <span class="highlight-p">);</span>
```

```
<span class="highlight-kd">var</span> <span class="highlight-nx">fillbar</span> <span class="highlight-o">= </span> <span class="highlight-nb">document</span> <span class="highlight-p">.</span> <span class="highlight-nx">getElementById</span> <span class="highlight-p">(</span> <span class="highlight-s1">'fillbar'</span> <span class="highlight-p">);</span>
```

```
<span class="highlight-nx">fillbar</span> <span class="highlight-p">.</span> <span class="highlight-nx">style</span> <span class="highlight-p">.</span> <span class="highlight-nx">width</span> <span class="highlight-o">= </span> <span class="highlight-nb">parseFloat</span>
/ <span class="highlight-p">(</span> <span class="highlight-nx">cur</span> <span class="highlight-o">/ </span> <span class="highlight-nx">duration</span> <span class="highlight-p">)</span> <span class="highlight-o">*</span> <span class="highlight-nx">bglen</span>
<span class="highlight-p">+ </span> <span class="highlight-s1">'px'</span> <span class="highlight-p">);</span>
```

```
</code> </pre>
```

```
<h2 id="滚动歌词">滚动歌词</h2>
```

<p>歌词这一块，首先要找好思路，如何解析歌词？如何控制歌词滚动？如何判断当前是哪句歌词？</p>

清楚这几个问题后，事情就变得不困难了。

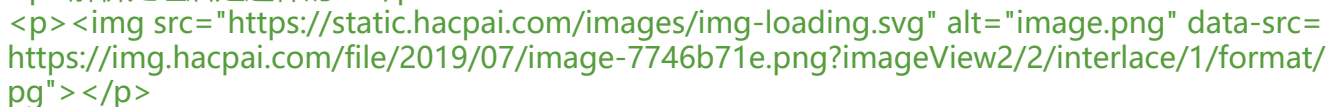
解析歌词

这一块可能是这部分内容里面最复杂的，需要一些简单的算法，这里用到了递归。

解析前的 lrc 歌词是这样的：

```
<code class="highlight-chroma">[ti:不要在我寂寞的时候说爱我] [ar:郑源] [by:] [00:21.57  
柔柔的晚风轻轻吹过 [00:25.39]我的心情平静而寂寞 [00:29.96]当我想忘记爱情去勇敢生活 [00:34.11  
是谁到我身边唱起了情歌 [01:48.86][00:39.01]当初的爱情匆匆走过 [01:52.69][00:42.79]除了伤口没  
下什么 [01:57.27][00:47.40]你总是在我寂寞流泪的时候 [02:01.67][00:51.80]用你的双臂紧紧抱着我  
02:59.75][02:05.19][00:55.27]不要在我寂寞的时候说爱我 [03:04.15][02:09.69][00:59.96]除非你真  
能给予我快乐 [03:09.30][02:14.78][01:04.97]那过去的伤总在随时提醒我 [03:13.94][02:19.40][01:09  
65]别再被那爱情折磨 [03:17.17][02:22.65][01:12.82]不要在我哭泣的时候说爱我 [03:21.62][02:27.1  
][01:17.32]除非你真的不让我难过 [03:25.65][02:31.13][01:21.33]我不想听太多那虚假的承诺 [03:30.  
0][02:35.44][01:25.69]让我为爱再次后悔犯下的错!
```

解析处理后是这样的：



```
<code class="highlight-chroma">function getNext(y, lrc) {  
  var rs = "";  
  var i = y + 1;  
  if(lrc[i]) {  
    t = lrc[i].split('');  
    if(t[1] == "") {  
      rs = getNext(i, lrc);  
    }else {  
      rs = t[1]  
    }  
  }  
  return rs;  
}
```

歌词滚动

歌词滚动最简单的一种方法，让包含歌词的盒子的 top 值不断减少，歌词就会一直往上走。通过前播放的时间和歌词的时间对个比较，判断当前时间对应的歌词，给对应的 dom 添加标识。

```
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></scr  
pt>
```

<!-- 黑客派PC帖子内嵌-展示 -->

```
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342"  
data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></in  
>
```

```
<script>  
  (adsbygoogle = window.adsbygoogle || []).push({});  
</script>
```

```
<code class="language-js highlight-chroma"><span class="highlight-nx">cur</span>  
<span class="highlight-o">=</span> <span class="highlight-nb">parseInt</span><span class="highlight-p">(</span><span class="highlight-nx">audio</span><span class="highlight-p">.</span><span class="highlight-nx">currentTime</span><span class="highlight-p">)</span><span class="highlight-nx">min</span> <span class="highlight-o">=</span> <span class="highlight-nb">parseInt</span><span class="highlight-p">(</span><span class="highlight-nx">cur</span><span class="highlight-o">/</span><span class="highlight-mi">60</span><span class="highlight-p">)</span><span class="highlight-nx">sec</span> <span class="highlight-o">=</span> <span class="highlight-nx">
```



```

highlight-nb">parseInt</span><span class="highlight-p">(</span><span class="highlight-
x">cur</span><span class="highlight-o">%</span><span class="highlight-mi">60</spa
><span class="highlight-p">);</span>
<span class="highlight-nx">sec</span><span class="highlight-o">=</span><span class=
highlight-nx">sec</span><span class="highlight-o">&lt;</span><span class="highlight-m
">10</span><span class="highlight-o">?</span><span class="highlight-s1">'0'</span>
span class="highlight-o">+</span><span class="highlight-nx">sec</span><span class="h
ghlight-o">:</span><span class="highlight-nx">sec</span><span class="highlight-p">);</
pan>
<span class="highlight-nx">currentTime</span><span class="highlight-p">.</span><span clas
="highlight-nx">innerHTML</span><span class="highlight-o">=</span><span class="highl
ight-nx">min</span><span class="highlight-o">+</span><span class="highlight-s1">'<
span><span class="highlight-o">+</span><span class="highlight-nx">sec</span><span c
ass="highlight-p">);</span>
<span class="highlight-nx">fillbar</span><span class="highlight-p">.</span><span class=
highlight-nx">style</span><span class="highlight-p">.</span><span class="highlight-nx"
width</span><span class="highlight-o">=</span><span class="highlight-nb">parseFloat
/span><span class="highlight-p">(</span><span class="highlight-nx">cur</span><span c
ass="highlight-o">/</span><span class="highlight-nx">duration</span><span class="high
ight-p">)</span><span class="highlight-o">*</span><span class="highlight-nx">bglen</
pan><span class="highlight-o">+</span><span class="highlight-s1">'px'</span><span c
ass="highlight-p">);</span>
<span class="highlight-k">if</span><span class="highlight-p">(</span><span class="highl
ght-nx">lrcTimeArr</span><span class="highlight-p">.</span><span class="highlight-nx">
ndexOf</span><span class="highlight-p">(</span><span class="highlight-nx">cur</span>
<span class="highlight-p">)</span><span class="highlight-o">&gt;</span><span class="
ighlight-o">-</span><span class="highlight-mi">1</span><span class="highlight-p">)</s
an><span class="highlight-p">{</span>
  <span class="highlight-nb">document</span><span class="highlight-p">.</span><span class=
class="highlight-nx">querySelectorAll</span><span class="highlight-p">(</span><span cla
s="highlight-s1">'#lrc li'</span><span class="highlight-p">).</span><span class="highlight
nx">forEach</span><span class="highlight-p">(</span><span class="highlight-kd">functi
n</span><span class="highlight-p">(</span><span class="highlight-nx">e</span><span class=
ass="highlight-p">)</span><span class="highlight-p">{</span>
  <span class="highlight-nx">e</span><span class="highlight-p">.</span><span class=
highlight-nx">className</span><span class="highlight-o">=</span><span class="highli
ht-s1">"</span><span class="highlight-p">);</span>
  <span class="highlight-p">})</span>
  <span class="highlight-nb">document</span><span class="highlight-p">.</span><span class=
class="highlight-nx">querySelector</span><span class="highlight-p">(</span><span class
"highlight-s1">'#lrc li[time="'</span><span class="highlight-o">+</span><span class="hi
hlight-nx">cur</span><span class="highlight-o">+</span><span class="highlight-s1">'"]'
/span><span class="highlight-p">).</span><span class="highlight-nx">className</span>
<span class="highlight-o">=</span><span class="highlight-s1">'active'</span><span class=
="highlight-p">);</span>
  <span class="highlight-nx">lrcEl</span><span class="highlight-p">.</span><span class
"highlight-nx">style</span><span class="highlight-p">.</span><span class="highlight-nx"
top</span><span class="highlight-o">=</span><span class="highlight-nx">lrcOffsetTop
/span><span class="highlight-o">-</span><span class="highlight-nx">lrcTimeArr</span>
<span class="highlight-p">.</span><span class="highlight-nx">indexOf</span><span class=
="highlight-p">(</span><span class="highlight-nx">cur</span><span class="highlight-p">
</span><span class="highlight-o">*</span><span class="highlight-mi">16</span><span class=
"highlight-o">+</span><span class="highlight-s1">'px'</span><span class="highli
ht-p">);</span>

```

```
<span class="highlight-p"></span>  
</code></pre>  
<h2 id="写在最后">写在最后</h2>  
<p>以上是在音乐播放器以及可视化方面的一些实验。 </p>
```