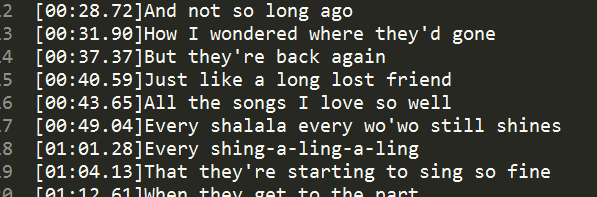
HTML5的最强大之处莫过于对媒体文件的处理，如利用一个简单的vedio标签就可以实现视频播放。类似地,在HTML5中也有对应的处理音频文件的标签，那就是audio标签   
HTML5出来这么久了，但是关于它里面的audio标签也就用过那么一次，当然还仅仅只是把这个标签插入到了页面中。这次呢就刚好趁着帮朋友做几个页面，拿这个audio标签来练练手。   
首先你需要向页面中插入一个audio标签，注意这里最好不要设置loop='loop',这个属性使用来设置循环播放的，因为到后面需要使用ended属性的时候，如果loop被设置为loop的话，ended属性将一直是false。   
autoplay='autoplay'设置页面加载后自动播放音乐，preload和autoplay属性的作用是一样的，如果标签中出现了autoplay属性，那么preload属性将被忽略。   
controls='controls'设置显示音乐的控制条。

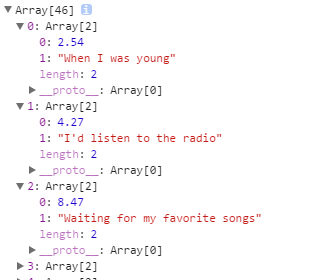
**XML/HTML Code**复制内容到剪贴板

1. **<audio** src="music/Yesterday Once More.mp3" id="aud" autoplay="autoplay" controls="controls" preload="auto"**>**
2. 您的浏览器不支持audio属性，请更换浏览器在进行浏览。
3. **</audio>**

有了这个标签之后，那么恭喜你，你的页面已经可以播放音乐了。但是这样会不会显得页面太过于单调了，于是我又给页面添加了一些东西，让歌词能够同步的显示在页面上，还能够选择要播放的音乐。那么先要做成这样的效果，我们就得要去下载一些lrc格式的歌词文件，然后你需要把这些音乐格式化一番。因为刚开始的音乐文件是这样的



我们需要把每一句歌词插入到一个二位数组里面，经过格式化之后歌词就变成这样的格式了   
这里附上格式化歌词的代码



**XML/HTML Code**复制内容到剪贴板

1. //歌词同步部分
2. function parseLyric(text) {
3. //将文本分隔成一行一行，存入数组
4. var lines = text.split('\n'),
5. //用于匹配时间的正则表达式，匹配的结果类似[xx:xx.xx]
6. pattern = /

\d2:\d2.\d2\d2:\d2.\d2

1. /g,
2. //保存最终结果的数组
3. result = [];
4. //去掉不含时间的行
5. while (!pattern.test(lines[0])) {
6. lineslines = lines.slice(1);
7. };
8. //上面用'\n'生成生成数组时，结果中最后一个为空元素，这里将去掉
9. lines[lines.length - 1].length === 0 && lines.pop();
10. lines.forEach(function(v /\*数组元素值\*/ , i /\*元素索引\*/ , a /\*数组本身\*/ ) {
11. //提取出时间[xx:xx.xx]
12. var time = v.match(pattern),
13. //提取歌词
14. vvalue = v.replace(pattern, '');
15. //因为一行里面可能有多个时间，所以time有可能是[xx:xx.xx][xx:xx.xx][xx:xx.xx]的形式，需要进一步分隔
16. time.forEach(function(v1, i1, a1) {
17. //去掉时间里的中括号得到xx:xx.xx
18. var t = v1.slice(1, -1).split(':');
19. //将结果压入最终数组
20. result.push([parseInt(t[0], 10) \* 60 + parseFloat(t[1]), value]);
21. });
22. });
23. //最后将结果数组中的元素按时间大小排序，以便保存之后正常显示歌词
24. result.sort(function(a, b) {
25. return a[0] - b[0];
26. });
27. return result;
28. }

到了这里我们就能够很容易的使用每首音乐的歌词了，我们需要有一个function来获得歌词，并且让他同步的显示在页面上，能够正常的切换音乐。下面附上代码。

**XML/HTML Code**复制内容到剪贴板

1. function fn(sgname){
2. $.get('music/'+sgname+'.lrc',function(data){
3. var str=parseLyric(data);
4. for(var i=0,li;i**<str.length**;i++){
5. li=$('**<li>**'+str[i][1]+'**</li>**');
6. $('#gc ul').append(li);
7. }
8. $('#aud')[0].ontimeupdate=function(){//视屏 音频当前的播放位置发生改变时触发
9. for (var i = 0, l = str.length; i **<** **l**; i++) {
10. if (this.currentTime /\*当前播放的时间\*/ **>** str[i][0]) {
11. //显示到页面
12. $('#gc ul').css('top',-i\*40+200+'px'); //让歌词向上移动
13. $('#gc ul li').css('color','#fff');
14. $('#gc ul li:nth-child('+(i+1)+')').css('color','red'); //高亮显示当前播放的哪一句歌词
15. }
16. }
17. if(this.ended){ //判断当前播放的音乐是否播放完毕
18. var songslen=$('.songs\_list li').length;
19. for(var i= 0,val;i**<songslen**;i++){
20. val=$('.songs\_list li:nth-child('+(i+1)+')').text();
21. if(val==sgname){
22. i=(i==(songslen-1))?1:i+2;
23. sgname=$('.songs\_list li:nth-child('+i+')').text(); //音乐播放完毕之后切换下一首音乐
24. $('#gc ul').empty(); //清空歌词
25. $('#aud').attr('src','music/'+sgname+'.mp3');
26. fn(sgname);
27. return;
28. }
29. }
30. }
31. };
32. });
33. } fn($('.songs\_list li:nth-child(1)').text());

那么到了这里你的音乐歌词已经能够正常的同步显示在页面上了。不过还缺少一个东西，就是一个音乐的列表，我希望能够点击这个列表里的音乐，从而播放该音乐，下面附上代码。   
****HTML代码****

**XML/HTML Code**复制内容到剪贴板

1. **<div** class="songs\_cnt"**>**
2. **<ul** class="songs\_list"**>**
3. **<li>**Yesterday Once More**</li>**
4. **<li>**You Are Beautiful**</li>**
5. **</ul>**
6. **<button** class="sel\_song"**>**点**<br/><br/>**我**</button>**
7. **</div>**
8. **<div** id="gc"**>**
9. **<ul></ul>**
10. **</div>**

****css代码****

**XML/HTML Code**复制内容到剪贴板

1. #gc{
2. width: 400px;
3. height: 400px;
4. background: transparent;
5. margin: 100px auto;
6. color: #fff;
7. font-size: 18px;
8. overflow: hidden;
9. position: relative;
10. }
11. #gc ul{
12. position: absolute;
13. top: 200px;
14. }
15. #gc ul li{
16. text-align: center;
17. height: 40px;
18. line-height: 40px;
19. }
20. .songs\_cnt{
21. float: left;
22. margin-top: 200px;
23. position: relative;
24. }
25. .songs\_list{
26. background-color: rgba(0,0,0,.2);
27. border-radius: 5px;
28. float: left;
29. width: 250px;
30. padding: 15px;
31. margin-left: -280px;
32. }
33. .songs\_list li{
34. height: 40px;
35. line-height: 40px;
36. font-size: 16px;
37. color: rgba(255,255,255,.8);
38. cursor: pointer;
39. }
40. .songs\_list li:hover{
41. font-size: 20px;
42. color: rgba(255,23,140,.6);
43. }
44. .sel\_song{
45. position: absolute;
46. top: 50%;
47. width: 40px;
48. height: 80px;
49. margin-top: -40px;
50. font-size: 16px;
51. text-align: center;
52. background-color: transparent;
53. border: 1px solid #2DCB70;
54. font-weight: bold;
55. cursor: pointer;
56. border-radius: 3px;
57. font-family: sans-serif;
58. transition:all 2s;
59. -moz-transition:all 2s;
60. -webkit-transition:all 2s;
61. -o-transition:all 2s;
62. }
63. .sel\_song:hover{
64. color: #fff;
65. background-color: #2DCB70;
66. }
67. .songs\_list li.active{
68. color: #f00;
69. }

****js代码****

**XML/HTML Code**复制内容到剪贴板

1. $('.songs\_list li:nth-child(1)').addClass('active');
2. $('.songs\_cnt').mouseenter(function () {
3. var e=event||window.event;
4. var tag= e.target||e.srcElement;
5. if(tag.nodeName=='BUTTON'){
6. $('.songs\_list').animate({'marginLeft':'0px'},'slow');
7. }
8. });
9. $('.songs\_cnt').mouseleave(function () {
10. $('.songs\_list').animate({'marginLeft':'-280px'},'slow');
11. });
12. $('.songs\_list li').each(function () {
13. $(this).click(function () {
14. $('#aud').attr('src','music/'+$(this).text()+'.mp3');
15. $('#gc ul').empty();
16. fn($(this).text());
17. $('.songs\_list li').removeClass('active');
18. $(this).addClass('active');
19. });
20. })