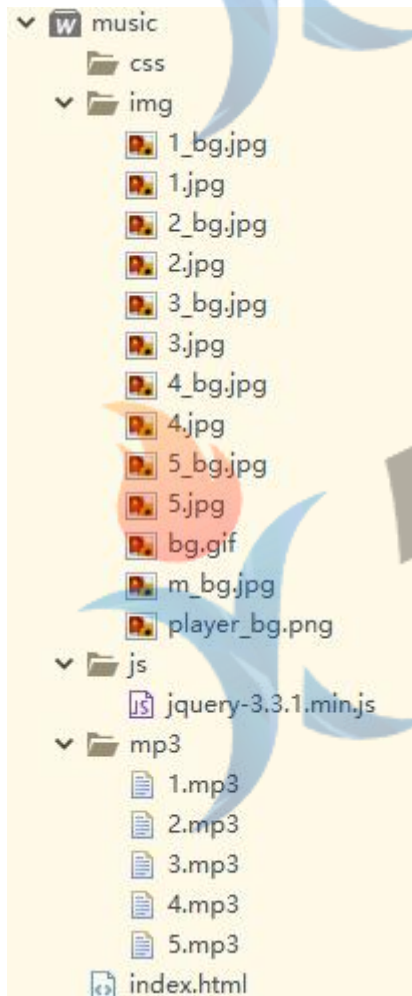


# 爱尚音乐开发流程

为了方便大家学习，特将爱尚音乐的开发流程写成文档，方便大家学习参考。

## 一、创建项目

创建 html 项目，导入项目所需图片素材和 jQuery 文件。



编写 index.html，导入 jQuery 文件。

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>爱尚音乐</title>
  </head>

  <body>
    <!-- 引入外部js文件 -->
```

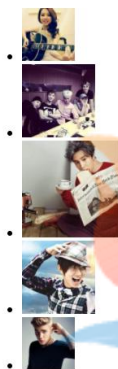
```
<script type="text/javascript" src="js/jquery-3.3.1.min.js"></script>
</body>
</html>
```

## 二、开发步骤

### 1、布局横幅 banner

使用 div ul li 实现横幅 banner 的布局，并设置图片的宽度。  
index.html

```
<body>
<!-- -----banner start----- -->
<div class="banner">
  <ul>
    <li class="m1" datasrc="mp3/1.mp3" title="龙卷风 - 邓紫棋">
      
    </li>
    <li class="m2" datasrc="mp3/2.mp3" title="Sober - bigbang">
      
    </li>
    <li class="m3" datasrc="mp3/3.mp3" title="绅士 - 薛之谦">
      
    </li>
    <li class="m4" datasrc="mp3/4.mp3" title="杀手 - 林俊杰">
      
    </li>
    <li class="m5" datasrc="mp3/5.mp3" title="baby - justin bieber">
      
    </li>
  </ul>
</div>
<!-- -----banner end----- -->
</body>
```



我们需要修改样式来实现横幅的效果，目前使用的是默认的纵向无序列表样式。

## 2、创建 css 文件

创建 css 文件，将文件导入 html，设置横幅背景图片，横幅背景阴影；取消无序列表默认样式，使列表横向显示。

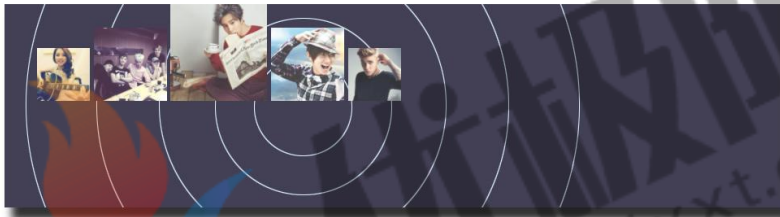
music.css

```
/* 采用什么编码解析文件 */
@charset "utf-8";

/* -----横幅 begin----- */
/* 设置横幅的宽高背景 */
.banner {
    width: 960px; /* 宽 */
    height: 250px; /* 高 */
    background: url(../img/m_bg.jpg) no-repeat; /* 背景图片 不重复 */
    margin: 80px; /* 外边距 */
    opacity: 0.8; /* 透明度 */
    box-shadow: 10px 15px 15px 1px black; /* 阴影部分 */
}

/* 设置横幅的li的样式 */
.banner ul li {
    list-style: none; /* 取消列表默认样式 */
    display: inline-block; /* 类似于导航栏的效果 */
}

/* -----横幅 end----- */
```



### 3、修改图片样式

将图片使用定位移动，修改图片样式，大小。

music.css

```
/* -----横幅 begin----- */
/* 设置横幅的宽高背景 */
.banner {
    width: 960px; /* 宽 */
    height: 250px; /* 高 */
    background: url(../img/m_bg.jpg); /* 设置背景图片 */
    margin: 80px; /* 外边距 */
    opacity: 0.8; /* 透明度 */
    box-shadow: 10px 15px 15px 1px black; /* 阴影部分 */
    position: relative; /* 让子元素使用定位相对于其父元素定位 */
}

/* 设置横幅的li的样式 */
.banner ul li {
    list-style: none; /* 取消列表默认样式 */
    display: inline-block; /* 类似于导航栏的效果 */
    position: absolute; /* 使用绝对定位，相对于其父元素定位 */
}

/* 操作图片 */
.banner ul li img {
    border-radius: 50%; /* 将图片菱角变圆 */
    cursor: pointer; /* 鼠标变手势 */
}
```

```
/* 使用类选择器 改变图片的位置 */
```

```
.m1 {top: 30px; left: 90px;}
```

```
.m2 {top: 140px; left: 160px;}
```

```
.m3 {top: 67px; left: 308px;}
```

```
.m4 {top: 50px; left: 500px;}
```

```
.m5 {top: 160px; left: 580px;}
```

```
/* -----横幅 end----- */
```



## 4、布局播放器 music

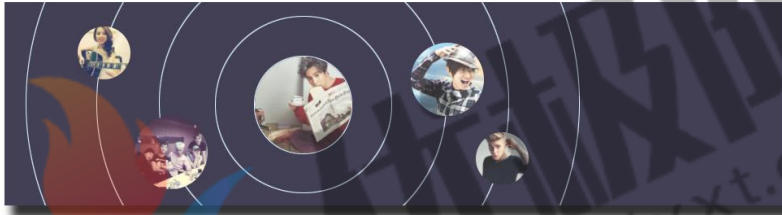
使用 div 实现播放器 music 的布局，并设置图片的宽度。

index.html

```
<!-- -----music start----- -->
<div class="music">
  <div class="m_img">
    
  </div>
  <div class="m_text">龙卷风 - 邓紫棋</div>
  <div class="m_btn">
    <a href="#" class="m_prev" title="上一首"></a>
    <a href="#" class="m_play" title="播放"></a>
    <a href="#" class="m_next" title="下一首"></a>
  </div>
  <div class="m_close"></div>
  <audio src="mp3/1.mp3" class="m_mp3"></audio>
</div>
<!-- -----music end----- -->
```



龙卷风 - 邓紫棋



## 5、修改播放器样式

设置播放器宽，高，背景色，使用定位，设置透明度及背景阴影。

music.css

```
/* -----播放器 begin----- */
/* 修改播放器的样式 */
.music {
    width: 550px; /* 宽 */
    height: 120px; /* 高 */
    background: black; /* 背景色黑色 */
    position: fixed; /* 相对浏览器的绝对定位 */
    left: 0px; /* 最左边开始 */
    bottom: 50px; /* 从底部开始移动 */
    opacity: 0.8; /* 透明度 */
    box-shadow: 10px 15px 15px 1px black; /* 阴影部分 */
}
/* -----播放器 end----- */
```





## 6、优化播放器

修改播放器内的图片，文字，及上一首、下一首、暂停、打开关闭各按钮的位置及样式。

music.css

```
/* -----播放器 begin----- */
/* 修改播放器的样式 */
.music {
    width: 550px; /* 宽 */
    height: 120px; /* 高 */
    background: black; /* 背景色黑色 */
    position: fixed; /* 相对浏览器的绝对定位 */
    left: 0px; /* 最左边开始 */
    bottom: 50px; /* 从底部开始移动 */
    opacity: 0.8; /* 透明度 */
    box-shadow: 10px 15px 15px 1px black; /* 阴影部分 */
}

/* 调整音乐盒图片的位置 */
.music .m_img {
    float: left; /* 左浮动 */
    margin-top: 15px;
    margin-left: 10px;
}

/* 修改音乐盒文本的显示 */
.music .m_text {
    float: left; /* 左浮动 */
    color: white; /* 颜色白色 */
    font-size: 20px; /* 字体大小 */
}
```

```
font-weight: bold ; /* 加粗 */
margin-top: 50px;
margin-left: 20px;
}

/* 使得所有a标签一起移动 */
.music .m_btn {
    float: left; /* 左浮动 */
    /* 使用绝对定位, 防止歌曲名称过长挤出按钮, div发生样式改变 */
    position: absolute;
    left: 320px;
}

/* 修改a标签 设置宽高及背景图片 */
.music .m_btn a {
    float: left; /* 左浮动 */
    width: 35px; /* 宽 */
    height: 35px; /* 高 */
    background: url(../img/player_bg.png);
    margin-top: 50px;
    margin-left: 20px;
}

/* 改变上一首的图片 */
.music .m_btn .m_prev {background-position: -68px 0px;}
/* 改变暂停的图片 */
.music .m_btn .m_play {background-position: -104px -5px;}
/* 改变下一首的图片 */
.music .m_btn .m_next {background-position: -145px 0px;}

/* 改变上一首的图片 */
.music .m_btn .m_prev:hover {background-position: -68px -32px;}
/* 改变暂停的图片 */
.music .m_btn .m_play:hover {background-position: -104px -47px;}
/* 改变下一首的图片 */
.music .m_btn .m_next:hover {background-position: -145px -32px;}

/* 设置最右边的关闭样式 */
.music .m_close {
    float: right; /* 右浮动 */
    width: 23px; /* 宽 */
    height: 120px; /* 高 */
    background: url(../img/player_bg.png);
    margin-top: 2px;
```



```

        cursor: pointer; /* 鼠标变手势 */
    }
    /* -----播放器 end----- */

```

这部分的代码使用到了对图片坐标的调整和修改，大家需要掌握一下。

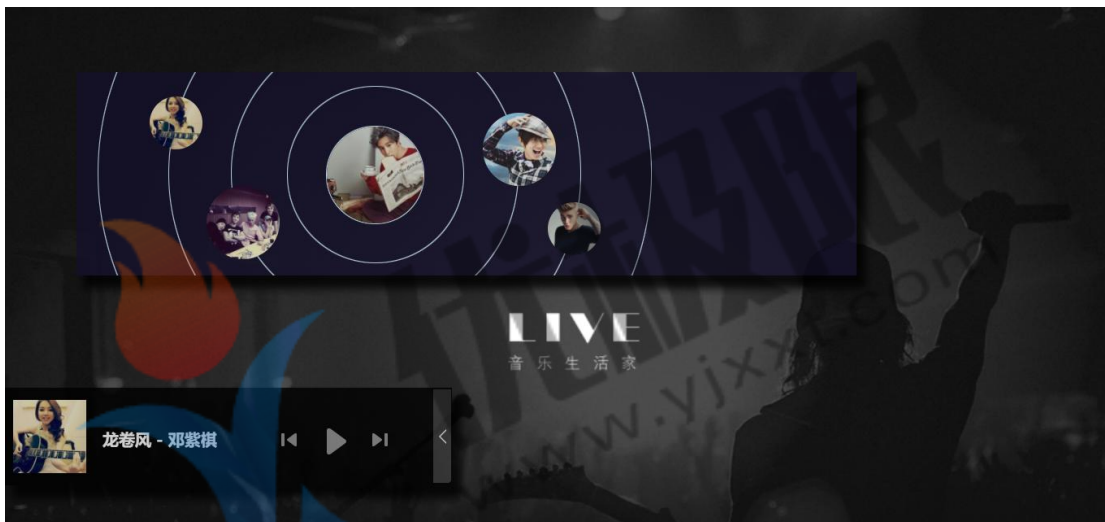
## 7、添加背景图片

music.css

```

/* 使用元素选择器 设置body背景 */
body {
    background: url(../img/bg.gif) no-repeat; /* 图片不重复 */
    background-size: cover; /* 自适应 */
}

```



接下来就是使用 js 处理事件，将页面动起来，更换背景，使歌曲播放和暂停，图片旋转等效果。

## 8、创建 js 文件并更换背景

点击横幅图片后，第一步先更换背景图片。

music.js

```

/**
 * 分析:
 * 1. 获取li的index()
 * 2. 更换背景图片
 * 3. 更换播放器图片、文本
 * 4. 更换播放按钮及title为暂停
 * 5. 图片旋转
 * 6. 播放歌曲

```

```

*
* 其他：
*   1. 暂停/播放歌曲
*   2. 上一首 下一首
*   3. 播放器可以显示与隐藏
*/

// 准备工作 收集数据
var index = 0; // li初始化索引
var li = $('.banner ul li'); // 获取所有的li元素
var img = $('.music .m_img img'); // 获取播放器img元素
var text = $('.music .m_text'); // 获取播放器m_text元素
var prev = $('.music .m_btn .m_prev'); // 获取播放器上一首a元素按钮
var play = $('.music .m_btn .m_play'); // 获取播放器暂停/播放a元素按钮
var next = $('.music .m_btn .m_next'); // 获取播放器下一首a元素按钮
var mp3 = $('.music .m_mp3'); // 媒体元素
var flag = false; // 歌曲是否播放的标记 true播放 false暂停
var close = true; // 播放器是否显示 true显示 false隐藏

/*
* 获取点击的li索引
*/
li.click(function() {
    //console.log($(this).index());
    // 获取当前点击的li索引
    index = $(this).index();

    // 播放歌曲
    show();
});

/*
* 封装一个函数 方便上一首 下一首调用
*/
function show() {
    // 更换背景图片 +1是因为索引从0开始，图片名称是从1开始的
    change_bg(index+1);

    // 更换播放器图片、文本

    // 更换播放按钮及title为暂停

    // 图片旋转

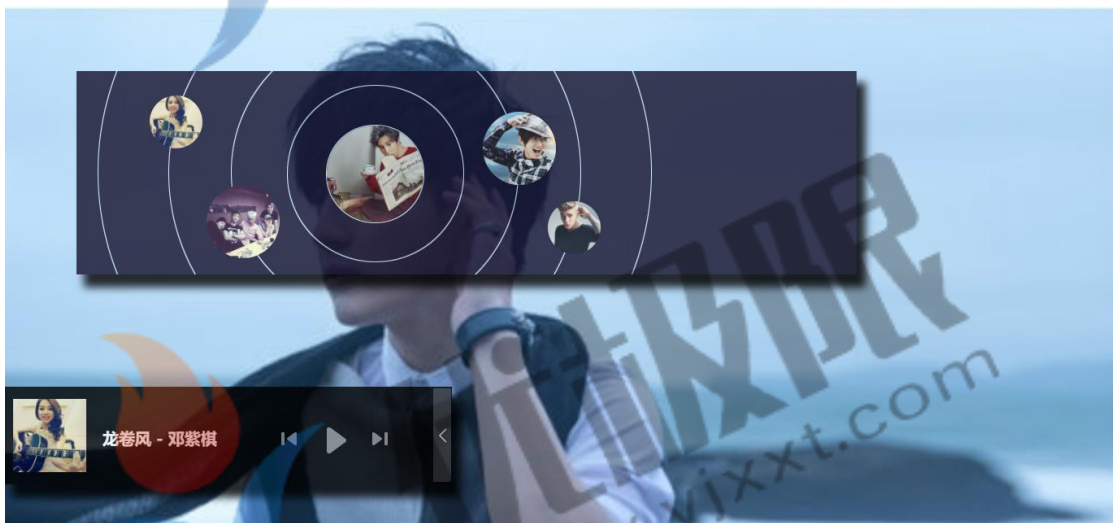
```

```

// 播放歌曲
}

/*
 * 更换背景图片
 */
function change_bg(idx) {
    $('body').css({
        'background': 'url(img/'+idx+'_bg.jpg) no-repeat',
        'background-size': 'cover'
    });
}

```



## 9、更换播放器图片、文本

点击横幅图片后，第二步更换播放器内图片和文本。

music.js

```

/*
 * 封装一个函数 方便上一首 下一首调用
 */
function show() {
    // 更换背景图片 +1是因为索引从0开始，图片名称是从1开始的
    change_bg(index+1);

    // 更换播放器图片、文本
    change_img_text(index+1);

    // 更换播放按钮及title为暂停

```

```

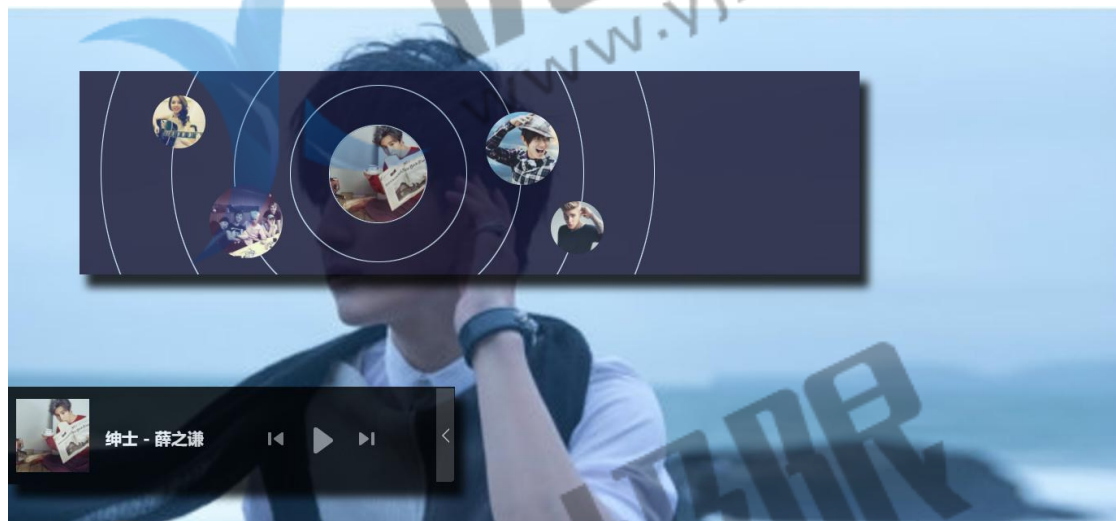
// 图片旋转

// 播放歌曲
}

/*
 * 更换背景图片
 */
function change_bg(idx) {
    $('body').css({
        'background': 'url(img/'+idx+'_bg.jpg) no-repeat',
        'background-size': 'cover'
    });
}

/*
 * 更换播放器图片、文本
 */
function change_img_text(idx) {
    img.attr('src', 'img/'+idx+'.jpg');// 更换播放器图片
    text.html(li.eq(index).attr('title'));// 获取li的title属性然后替换文本
}

```



## 10、更换播放按钮及 title 为暂停

点击横幅图片后，第三步更换播放按钮为暂停及 title，先编写 css 样式，然后使用 js 动态更换样式。

music.css

```

/* -----播放器 begin----- */

...

/* 编写js时，更换播放按钮为暂停时编写的代码 */
.music .m_btn .m_pause {background-position: -292px -93px;}
.music .m_btn .m_pause:hover {background-position: -334px -93px;}
/* -----播放器 end----- */

```

music.js

```

/*
 * 封装一个函数 方便上一首 下一首调用
 */
function show() {
    // 更换背景图片 +1是因为索引从0开始，图片名称是从1开始的
    change_bg(index+1);

    // 更换播放器图片、文本
    change_img_text(index+1);

    // 更换播放按钮及title为暂停
    change_btn_title();

    // 图片旋转

    // 播放歌曲
}

/*
 * 更换背景图片
 */
function change_bg(id) {
    $('body').css({
        'background': 'url(img/'+id+'_bg.jpg) no-repeat',
        'background-size': 'cover'
    });
}

/*
 * 更换播放器图片、文本
 */
function change_img_text(id) {
    img.attr('src', 'img/'+id+'.jpg');// 更换播放器图片
    text.html(li.eq(index).attr('title'));// 获取li的title属性然后替换文本
}

```

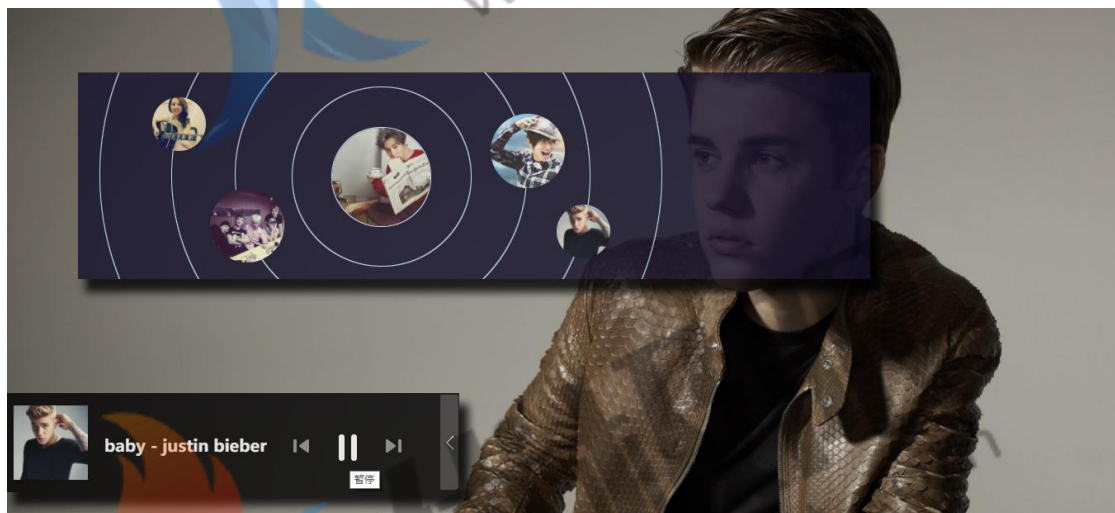


```

}

/*
 * 更换播放按钮及title为暂停
 */
function change_btn_title() {
    play.removeClass('m_play'); // 移除原有的播放样式
    play.addClass('m_pause'); // 添加新的暂停样式
    play.attr('title', '暂停'); // 更换title
}

```



## 11、图片旋转

点击横幅图片后，第四步图片旋转，先编写 css 样式，然后使用 js 动态更换样式。

music.css

图片旋转参考：

<http://www.w3school.com.cn/cssref/index.asp#animation>

<http://www.w3school.com.cn/cssref/index.asp#transform>

```

/* -----横幅 begin----- */
...

/* 图片旋转的动画 */
@keyframes img_rotate{
    from{
        transform:rotate(0deg);
    } to{
        transform:rotate(360deg);
    }
}

```



```

}

/* 动画执行的时间 次数 */
.banner .img_rotate {
    animation: img_rotate 2s infinite linear;
}
/* -----横幅 end----- */

```

music.js

```

/*
 * 封装一个函数 方便上一首 下一首调用
 */
function show() {
    // 更换背景图片 +1是因为索引从0开始, 图片名称是从1开始的
    change_bg(index+1);

    // 更换播放器图片、文本
    change_img_text(index+1);

    // 更换播放按钮为暂停及title
    change_btn_title();

    // 图片旋转
    img_rotate();

    // 播放歌曲
}

/*
 * 更换背景图片
 */
function change_bg(id) {
    $('body').css({
        'background': 'url(img/'+id+'_bg.jpg) no-repeat',
        'background-size': 'cover'
    });
}

/*
 * 更换播放器图片、文本
 */
function change_img_text(id) {
    img.attr('src', 'img/'+id+'.jpg');// 更换播放器图片
    text.html(li.eq(index).attr('title'));// 获取li的title属性然后替换文本
}

```

```

}

/*
 * 更换播放按钮及title为暂停
 */
function change_btn_title() {
    play.removeClass('m_play'); // 移除原有的播放样式
    play.addClass('m_pause'); // 添加新的暂停样式
    play.attr('title', '暂停'); // 更换title
}

/*
 * 图片旋转
 */
function img_rotate() {
    // 移除所有img的图片旋转样式
    li.children().removeClass('img_rotate');
    // 给当前li的img添加图片旋转样式
    li.eq(index).children().addClass('img_rotate');
}

```



## 12、播放歌曲

点击横幅图片后，第五步播放歌曲，修改歌曲是否播放的标记。

music.js

```

/*
 * 封装一个函数 方便上一首 下一首调用
 */
function show() {

```

```

// 更换背景图片 +1是因为索引从0开始, 图片名称是从1开始的
change_bg(index+1);

// 更换播放器图片、文本
change_img_text(index+1);

// 更换播放按钮为暂停及title
change_btn_title();

// 图片旋转
img_rotate();

// 播放歌曲
play_mp3();
}

/*
 * 更换背景图片
 */
function change_bg(idx) {
    $('body').css({
        'background': 'url(img/'+idx+'_bg.jpg) no-repeat',
        'background-size': 'cover'
    });
}

/*
 * 更换播放器图片、文本
 */
function change_img_text(idx) {
    img.attr('src', 'img/'+idx+'.jpg');// 更换播放器图片
    text.html(li.eq(index).attr('title'));// 获取li的title属性然后替换文
本
}

/*
 * 更换播放按钮及title为暂停
 */
function change_btn_title() {
    play.removeClass('m_play');// 移除原有的播放样式
    play.addClass('m_pause');// 添加新的暂停样式
    play.attr('title', '暂停');// 更换title
}

```

```

/*
 * 图片旋转
 */
function img_rotate() {
    // 移除所有img的图片旋转样式
    li.children().removeClass('img_rotate');
    // 给当前li的img添加图片旋转样式
    li.eq(index).children().addClass('img_rotate');
}

/*
 * 播放歌曲
 */
function play_mp3() {
    // 获取选中的li的datasrc属性
    mp3.attr('src', li.eq(index).attr('datasrc'));
    mp3.get(0).play(); // 设置媒体元素 播放歌曲
    flag = true; // 修改歌曲是否播放的标记 true播放 false暂停
}

```

至此功能已经开发的差不多了，歌曲也可以正常播放了，但是没有办法暂停，我们需要继续改进案例，实现歌曲暂停功能及相关样式的修改。

## 13、暂停歌曲

music.js

```

/*
 * 暂停或者播放歌曲
 */
play.click(function() {
    /*
     * 如果歌曲播放
     * 1. 暂停歌曲
     * 2. 图片旋转停止
     * 3. 暂停按钮更换为播放
     * 4. title更换为播放
     */
    if (flag) {
        mp3.get(0).pause(); // 暂停歌曲
        li.eq(index).children().removeClass('img_rotate'); // 移除图片旋转
        // 暂停按钮更换为播放按钮，title更换为播放
        play.removeClass('m_pause').addClass('m_play').attr('title', '播放');
        flag = false; // 修改歌曲是否播放的标记 true播放 false暂停
    }
}

```

```

    } else {
        /*
         * 如果歌曲暂停
         * 1.播放歌曲
         * 2.图片旋转
         * 3.播放按钮更换为暂停
         * 4.title更换为暂停
         */
        mp3.get(0).play(); // 播放歌曲
        li.eq(index).children().addClass('img_rotate'); // 图片旋转
        // 暂停按钮更换为播放title更换为播放
        play.removeClass('m_play').addClass('m_pause').attr('title', '暂停');
        flag = true; // 修改歌曲是否播放的标记 true播放 false暂停
        // 第一次进入页面直接点击播放按钮时的背景处理
        change_bg(index+1);
    }
});

```

## 14、上一首下一首

接下来实现上一首和下一首切换歌曲的功能，和之前个人相册网站实现思路是一样的，如果是第一首歌曲，再点击上一首时切换最后一首，如果是最后一首，再点击下一首时切换第一首。

music.js

```

// 上一首
prev.click(function() {
    index--;
    if (0 > index) {
        index = li.length-1;
    }
    // 播放歌曲
    show();
});

// 下一首
next.click(function() {
    index++;
    if (li.length-1 < index) {
        index = 0;
    }
    // 播放歌曲
    show();
});

```

```
});
```

最后让播放器可以隐藏与显示，所有功能就都开发完了。

## 15、隐藏播放器

点击 `class="close"` 的 div 后，隐藏或者显示播放器，先编写 css 样式，然后使用 js 动态更换样式。

music.css

```
/* -----播放器 begin----- */  
  
...  
  
/* 编写js时，隐藏或者显示播放器时编写的代码 */  
.music .m_open {background-position: -45px 0px;}  
/* -----播放器 end----- */
```

music.js

```
/*  
 * 播放器可以隐藏与显示  
 */  
$(".m_close").click(function() {  
    // 如果显示则隐藏 添加样式  
    if (close) {  
        $(this).addClass("m_open");  
        // 添加向左移动样式0.8秒完成  
        $(".music").animate({"left": "-520px"}, 800);  
        close = false; // 播放器是否显示 true显示 false隐藏  
    } else {  
        // 如果隐藏则显示 移除样式  
        $(this).removeClass("m_open");  
        // 恢复初始值  
        $(".music").animate({"left": "0px"}, 800);  
        close = true; // 播放器是否显示 true显示 false隐藏  
    }  
});
```





大功告成，抓紧去体验一把自己写的爱尚爱尚音乐播放器吧，可以自行找图片和 mp3 素材，编写自己喜欢的案例。

### 三、案例完成

最终完整的代码如下：

index.html

```
<!DOCTYPE html>
<html>

  <head>
    <meta charset="UTF-8">
    <title>爱尚音乐</title>
    <!-- 引入外部css文件 -->
    <link rel="stylesheet" href="css/music.css" />
  </head>
```

```

<body>
  <!-- -----banner start----- -->
  <div class="banner">
    <ul>
      <li class="m1" datasrc="mp3/1.mp3" title="龙卷风 - 邓紫棋">
        
      </li>
      <li class="m2" datasrc="mp3/2.mp3" title="Sober - bigbang">
        
      </li>
      <li class="m3" datasrc="mp3/3.mp3" title="绅士 - 薛之谦">
        
      </li>
      <li class="m4" datasrc="mp3/4.mp3" title="杀手 - 林俊杰">
        
      </li>
      <li class="m5" datasrc="mp3/5.mp3" title="baby - justin bieber">
        
      </li>
    </ul>
  </div>
  <!-- -----banner end----- -->
  <!-- -----music start----- -->
  <div class="music">
    <div class="m_img">
      
    </div>
    <div class="m_text">龙卷风 - 邓紫棋</div>
    <div class="m_btn">
      <a href="#" class="m_prev" title="上一首"></a>
      <a href="#" class="m_play" title="播放"></a>
      <a href="#" class="m_next" title="下一首"></a>
    </div>
    <div class="m_close"></div>
    <audio src="mp3/1.mp3" class="m_mp3"></audio>
  </div>
  <!-- -----music end----- -->
  <!-- 引入外部js文件 -->
  <script type="text/javascript" src="js/jquery-3.3.1.min.js" ></script>
  <script type="text/javascript" src="js/music.js" ></script>
</body>

```

```
</html>
```

```
music.css
```

```
/* 采用什么编码解析文件 */
@charset "utf-8";

/* -----横幅 begin----- */
/* 设置横幅的宽高背景 */
.banner {
    width: 960px; /* 宽 */
    height: 250px; /* 高 */
    background: url(../img/m_bg.jpg); /* 设置背景图片 */
    margin: 80px; /* 外边距 */
    opacity: 0.8; /* 透明度 */
    box-shadow: 10px 15px 15px 1px black; /* 阴影部分 */
    position: relative; /* 让子元素使用定位相对于其父元素定位 */
}

/* 设置横幅的li的样式 */
.banner ul li {
    list-style: none; /* 取消列表默认样式 */
    display: inline-block; /* 类似于导航栏的效果 */
    position: absolute; /* 使用绝对定位，相对于其父元素定位 */
}

/* 操作图片 */
.banner ul li img {
    border-radius: 50%; /* 将图片菱角变圆 */
    cursor: pointer; /* 鼠标变手势 */
}

/* 使用类选择器 改变图片的位置 */
.m1 {top: 30px; left: 90px;}
.m2 {top: 140px; left: 160px;}
.m3 {top: 67px; left: 308px;}
.m4 {top: 50px; left: 500px;}
.m5 {top: 160px; left: 580px;}

/* 图片旋转的动画 */
@keyframes img_rotate{
    from{
        transform: rotate(0deg);
    } to{
        transform: rotate(360deg);
    }
}
```

```

}

/* 动画执行的时间 次数 */
.banner .img_rotate {
    animation: img_rotate 2s infinite linear;
}
/* -----横幅 end----- */

/* -----播放器 begin----- */
/* 修改播放器的样式 */
.music {
    width: 550px; /* 宽 */
    height: 120px; /* 高 */
    background: black; /* 背景色黑色 */
    position: fixed; /* 相对浏览器的绝对定位 */
    left: 0px; /* 最左边开始 */
    bottom: 50px; /* 从底部开始移动 */
    opacity: 0.8; /* 透明度 */
    box-shadow: 10px 15px 15px 1px black; /* 阴影部分 */
}

/* 调整音乐盒图片的位置 */
.music .m_img {
    float: left; /* 左浮动 */
    margin-top: 15px;
    margin-left: 10px;
}

/* 修改音乐盒文本的显示 */
.music .m_text {
    float: left; /* 左浮动 */
    color: white; /* 颜色白色 */
    font-size: 20px; /* 字体大小 */
    font-weight: bold; /* 加粗 */
    margin-top: 50px;
    margin-left: 20px;
}

/* 使得所有a标签一起移动 */
.music .m_btn {
    float: left; /* 左浮动 */
    /* 使用绝对定位，防止歌曲名称过长挤出按钮，div发生样式改变 */
    position: absolute;
    left: 320px;

```

```
}

/* 修改a标签 设置宽高及背景图片 */
.music .m_btn a {
    float: left; /* 左浮动 */
    width: 35px; /* 宽 */
    height: 35px; /* 高 */
    background: url(../img/player_bg.png);
    margin-top: 50px;
    margin-left: 20px;
}

/* 改变上一首的图片 */
.music .m_btn .m_prev {background-position: -68px 0px;}
/* 改变暂停的图片 */
.music .m_btn .m_play {background-position: -104px -5px;}
/* 改变下一首的图片 */
.music .m_btn .m_next {background-position: -145px 0px;}

/* 改变上一首的图片 */
.music .m_btn .m_prev:hover {background-position: -68px -32px;}
/* 改变暂停的图片 */
.music .m_btn .m_play:hover {background-position: -104px -47px;}
/* 改变下一首的图片 */
.music .m_btn .m_next:hover {background-position: -145px -32px;}

/* 设置最右边的关闭样式 */
.music .m_close {
    float: right; /* 右浮动 */
    width: 23px; /* 宽 */
    height: 120px; /* 高 */
    background: url(../img/player_bg.png);
    margin-top: 2px;
    cursor: pointer; /* 鼠标变手势 */
}

/* 编写js时，更换播放按钮为暂停时编写的代码 */
.music .m_btn .m_pause {background-position: -292px -93px;}
.music .m_btn .m_pause:hover {background-position: -334px -93px;}

/* 编写js时，隐藏或者显示播放器时编写的代码 */
.music .m_open {background-position: -45px 0px;}
/* -----播放器 end----- */
```



```
/* 使用元素选择器 设置body背景 */
body {
    background: url(../img/bg.gif) no-repeat; /* 图片不重复 */
    background-size: cover; /* 自适应 */
}
```

music.js

```
/**
 * 分析:
 * 1. 获取li的index()
 * 2. 更换背景图片
 * 3. 更换播放器图片、文本
 * 4. 更换播放按钮及title为暂停
 * 5. 图片旋转
 * 6. 播放歌曲
 *
 * 其他:
 * 1. 暂停/播放歌曲
 * 2. 上一首 下一首
 * 3. 播放器可以显示与隐藏
 */

// 准备工作 收集数据
var index = 0; // li初始化索引
var li = $('.banner ul li'); // 获取所有的li元素
var img = $('.music .m_img img'); // 获取播放器img元素
var text = $('.music .m_text'); // 获取播放器m_text元素
var prev = $('.music .m_btn .m_prev'); // 获取播放器上一首a元素按钮
var play = $('.music .m_btn .m_play'); // 获取播放器暂停/播放a元素按钮
var next = $('.music .m_btn .m_next'); // 获取播放器下一首a元素按钮
var mp3 = $('.music .m_mp3'); // 媒体元素
var flag = false; // 歌曲是否播放的标记 true播放 false暂停
var close = true; // 播放器是否显示 true显示 false隐藏

/*
 * 获取点击的li索引
 */
li.click(function() {
    // 获取当前点击的li索引
    index = $(this).index();
    //console.log(index);

    // 播放歌曲
    show();
});
```



```

/*
 * 封装一个函数 方便上一首 下一首调用
 */
function show() {
    // 更换背景图片 +1是因为索引从0开始，图片名称是从1开始的
    change_bg(index+1);

    // 更换播放器图片、文本
    change_img_text(index+1);

    // 更换播放按钮及title为暂停
    change_btn_title();

    // 图片旋转
    img_rotate();

    // 播放歌曲
    play_mp3();
}

/*
 * 更换背景图片
 */
function change_bg(idx) {
    $('body').css({
        'background': 'url(img/'+idx+'_bg.jpg) no-repeat',
        'background-size': 'cover'
    });
}

/*
 * 更换播放器图片、文本
 */
function change_img_text(idx) {
    img.attr('src', 'img/'+idx+'.jpg');// 更换播放器图片
    text.html(li.eq(index).attr('title'));// 获取li的title属性然后替换文本
}

/*
 * 更换播放按钮及title为暂停
 */
function change_btn_title() {
    play.removeClass('m_play');// 移除原有的播放样式
}

```

```

        play.addClass('m_pause'); // 添加新的暂停样式
        play.attr('title', '暂停'); // 更换title
    }

    /*
     * 图片旋转
     */
    function img_rotate() {
        // 移除所有img的图片旋转样式
        li.children().removeClass('img_rotate');
        // 给当前li的img添加图片旋转样式
        li.eq(index).children().addClass('img_rotate');
    }

    /*
     * 播放歌曲
     */
    function play_mp3() {
        // 获取选中的li的datasrc属性
        mp3.attr('src', li.eq(index).attr('datasrc'));
        mp3.get(0).play(); // 设置媒体元素 播放歌曲
        flag = true; // 修改歌曲是否播放的标记 true播放 false暂停
    }

    /*
     * 暂停或者播放歌曲
     */
    play.click(function() {
        /*
         * 如果歌曲播放
         * 1. 暂停歌曲
         * 2. 图片旋转停止
         * 3. 暂停按钮更换为播放
         * 4. title更换为播放
         */
        if (flag) {
            mp3.get(0).pause(); // 暂停歌曲
            li.eq(index).children().removeClass('img_rotate'); // 移除图片旋转
            // 暂停按钮更换为播放按钮, title更换为播放
            play.removeClass('m_pause').addClass('m_play').attr('title', '播放');
            flag = false; // 修改歌曲是否播放的标记 true播放 false暂停
        } else {
            /*
             * 如果歌曲暂停

```

```

    * 1.播放歌曲
    * 2.图片旋转
    * 3.播放按钮更换为暂停
    * 4.title更换为暂停
    */
    mp3.get(0).play(); // 播放歌曲
    li.eq(index).children().addClass('img_rotate'); // 图片旋转
    // 暂停按钮更换为播放title更换为播放
    play.removeClass('m_play').addClass('m_pause').attr('title', '暂停');
    flag = true; // 修改歌曲是否播放的标记 true播放 false暂停
    // 第一次进入页面直接点击播放按钮时的背景处理
    change_bg(index+1);
  }
});

// 上一首
prev.click(function() {
  index--;
  if (0 > index) {
    index = li.length-1;
  }
  // 播放歌曲
  show();
});

// 下一首
next.click(function() {
  index++;
  if (li.length-1 < index) {
    index = 0;
  }
  // 播放歌曲
  show();
});

/*
 * 播放器可以隐藏与显示
 */
$(".m_close").click(function() {
  // 如果显示则隐藏 添加样式
  if (close) {
    $(this).addClass("m_open");
    // 添加向左移动样式0.8秒完成
    $(".music").animate({"left": "-520px"}, 800);
  }
});

```

```
        close = false; // 播放器是否显示 true显示 false隐藏
    } else {
        // 如果隐藏则显示 移除样式
        $(this).removeClass("m_open");
        // 恢复初始值
        $(".music").animate({"left": "0px"}, 800);
        close = true; // 播放器是否显示 true显示 false隐藏
    }
});
```