

微信小程序把玩《九》： audio 组件， image 组件， video 组件

一： audio 组件



音频播放已经封装的很好！只需配合属性设置即可！（method 和 data 配合使用）

主要属性：

属性名	类型	默认值	说明
action	Object		控制音频的播放、暂停，播放速率、播放进度的对象，有method和data两个参数
src	String		要播放音频的资源地址
loop	Boolean	false	是否循环播放
controls	Boolean	true	是否显示默认控件
poster	String		默认控件上的音频封面的图片资源地址，如果controls属性值为false则设置poster无效
name	String	未知音频	默认控件上的音频名字，如果controls属性值为false则设置name无效
author	String	未知作者	默认控件上的作者名字，如果controls属性值为false则设置author无效
binderror	EventHandle		当发生错误时触发error事件，detail = {errMsg: MediaError.code}

binderror	EventHandle	当发生错误时触发error事件，detail = {errMsg: MediaError.code}
bindplay	EventHandle	当开始/继续播放时触发play事件
bindpause	EventHandle	当暂停播放时触发pause事件
bindratechange	EventHandle	当播放速率改变时触发ratechange事件
bindtimeupdate	EventHandle	当播放进度改变时触发timeupdate事件，detail = {currentTime, duration}
bindended	EventHandle	当播放到末尾时触发ended事件

返回错误码	描述
MEDIA_ERR_ABORTED	获取资源被用户禁止
MEDIA_ERR_NETWORK	网络错误
MEDIA_ERR_DECODE	解码错误
MEDIA_ERR_SRC_NOT_SUPPOERTED	不合适资源

method	描述	data
play	播放	
pause	暂停	
setPlaybackRate	调整速度	倍速
setCurrentTime	设置当前时间	播放时间

wxml

<audio

action="{{action}}"

src='http://sc1.111ttt.com/2016/1/09/26/202261732256.mp3'

poster= 'http://avatar.csdn.net/E/3/9/1_u014360817.jpg'

controls="true"

name="钢琴曲"

author="不详"/>

<button type="primary" bindtap="audioPlay">播放</button>

<button type="primary" bindtap="audioPause">暂停</button>

<!--正常速度-->

<button type="primary" bindtap="audioPlaybackRateNormal">调为 1 倍速</button>

<!--慢放-->

<button type="primary" bindtap="audioPlaybackRateSlowDown">调为 0.5 倍速</button>

js

Page({

```
data:{  
  // text:"这是一个页面"  
  action: {  
    method: "  
  }  
},
```

```
audioPlay:function() {  
  this.setData({  
    action: {  
      method: 'play'  
    }  
  })  
},
```

```
audioPause:function(){  
  this.setData({
```

```
        action: {  
            method: 'pause'  
        }  
    })  
},
```

```
audioPlaybackRateNormal:function() {  
    this.setData({  
        action:{  
            method: 'setPlaybackRate',  
            data: 1  
        }  
    })  
},
```

```
audioPlaybackRateSlowDown:function() {  
    this.setData({  
        action: {
```

```
    method: 'setPlaybackRate',  
    data: 0.5  
  }  
})  
  
}  
})
```

二 : image 组件

aspectFit 保持纵横比缩放图片，只保证图片的短边能完全显示出来

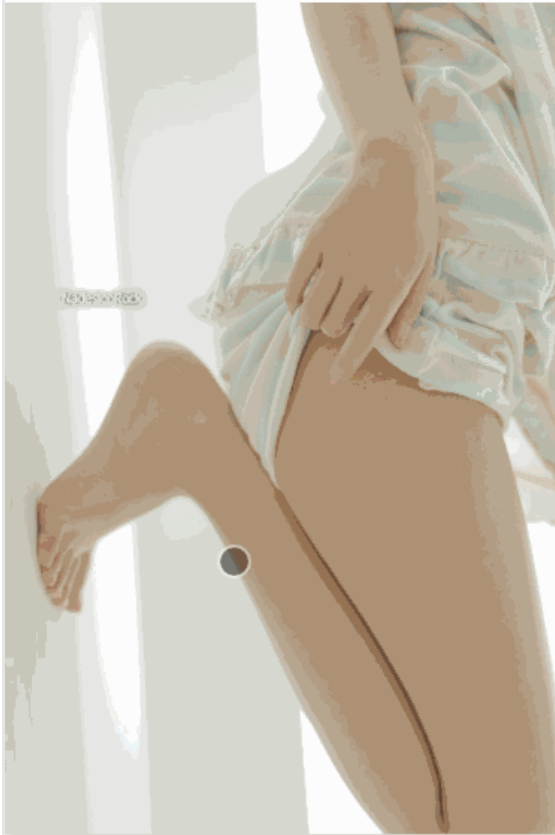


image 组件也是一个程序不可缺少的，可以说一个 app 中 image 组件随处可以看到，一般 image 有两种加载方式第一种是网络图片第二种是本地图片资源，都用 src 属性去指定。

重点属性：

属性名	类型	默认值	说明
src	String		图片资源地址
mode	String	'scaleToFill'	图片裁剪、缩放的模式
binderror	HandleEvent		当错误发生时，发布到AppService的事件名，事件对象event.detail = { errMsg: 'something wrong' }
bindload	HandleEvent		当图片载入完毕时，发布到AppService的事件名，事件对象event.detail = {}

三种缩放模式

模式	说明
scaleToFill	不保持纵横比缩放图片，使图片的宽高完全拉伸至填满image元素
aspectFit	保持纵横比缩放图片，使图片的长边能完全显示出来。也就是说，可以完整地将图片显示出来。
aspectFill	保持纵横比缩放图片，只保证图片的短边能完全显示出来。也就是说，图片通常只在水平或垂直方向是完整的，另一个方向将会发生截取。

九种剪切方式

模式	说明
top	不缩放图片， 只显示图片的顶部区域
bottom	不缩放图片， 只显示图片的底部区域
center	不缩放图片， 只显示图片的中间区域
left	不缩放图片， 只显示图片的左边区域
right	不缩放图片， 只显示图片的右边区域
top left	不缩放图片， 只显示图片的左上边区域
top right	不缩放图片， 只显示图片的右上边区域
bottom left	不缩放图片， 只显示图片的左下边区域
bottom right	不缩放图片， 只显示图片的右下边区域

wxml

<!--3 中是缩放模式

scaleToFill 不保持纵横比缩放图片，使图片的宽高完全拉伸至填满 image 元素

aspectFit 保持纵横比缩放图片，使图片的长边能完全显示出来。也就是说，可以完整地将图片显示出来。

aspectFill 保持纵横比缩放图片，只保证图片的短边能完全显示出来

-->

<view>aspectFit 保持纵横比缩放图片，只保证图片的短边能完全显示出来</view>

<image style="width: 100%; height:100%" mode="aspectFit" src="../../../image/image.jpg"/>

<!--9 种是裁剪模式

top 不缩放图片，只显示图片的顶部区域

bottom,同上

left

right

top right

top left

bottom right

bottom left

-->

<view>bottom 不缩放图片，只显示图片的底部区域</view>

<image style="width: 100%; height: 100%" mode="bottom" src="../../image/image.jpg"/>

<view>left 不缩放图片，只显示图片的左边区域</view>

<image style="width: 100%; height: 100%" mode="left" src="../../image/image.jpg"/>

<view>top right 不缩放图片，只显示图片的右上边区域</view>

<image style="width: 100%; height: 100%" mode="top right" src="../../image/image.jpg"/>

三：video 组件

点击显示视频组件

视频播放组件与图片加载组件也没啥差别，使用起来也没啥注意的

重要属性：

属性名	类型	默认值	说明
hidden	Boolean	false	设置视频的显示/隐藏，hidden值为true表示隐藏，值为false表示显示
src	String		要播放视频的资源地址
binderror	EventHandle		当发生错误时触发error事件，event.detail = { errMsg: 'something wrong' }

video标签默认宽度300px、高度225px，设置宽高需要通过wxss设置width和height。

wxml

<!--监听 button 点击事件-->

<button bindtap="listenerButton">点击显示视频组件</button>

<!--视频组件 src 资源地址，binderror 为监听错误信息-->

<video

src="http://mvvideo1.meitudata.com/575c2b652d7375255.mp4"

style="width: 100%; height: 100%"

hidden="{{hiddenVideo}}"

```
binderror="listenerVideo" />
```

```
js
```

```
Page({
```

```
  data:{
```

```
    // text:"这是一个页面"
```

```
    hiddenVideo: true
```

```
  },
```

```
  /**
```

```
   * 监听视频加载错误状态
```

```
   */
```

```
  listenerVideo:function(e) {
```

```
    console.log(e.detail.errMsg);
```

```
  },
```

```
  /**
```

```
   * 监听 button 点击事件
```

```
   */
```



```
listenerButton:function() {  
    this.setData({  
        hiddenVideo: !this.data.hiddenVideo  
    })  
}  
  
})
```