微信小程序组件文档

书栈(BookStack.CN)

目　录

- map

画布

- canvas

开放能力

- open-data
- web-view
- ad

致谢

当前文档 《微信小程序组件文档》 由 进击的皇虫 使用 书栈(BookStack.CN) 进行构建，生成于 2018-06-26。

书栈(BookStack.CN) 仅提供文档编写、整理、归类等功能，以及对文档内容的生成和导出工具。

文档内容由网友们编写和整理，书栈(BookStack.CN) 难以确认文档内容知识点是否错漏。如果您在阅读文档获取知识的时候，发现文档内容有不恰当的地方，请向我们反馈，让我们共同携手，将知识准确、高效且有效地传递给每一个人。

同时，如果您在日常工作、生活和学习中遇到有价值有营养的知识文档，欢迎分享到 书栈(BookStack.CN) ，为知识的传承献上您的一份力量！

如果当前文档生成时间太久，请到 书栈(BookStack.CN) 获取最新的文档，以跟上知识更新换代的步伐。

文档地址：http://www.bookstack.cn/books/mp-widget

书栈官网：http://www.bookstack.cn

书栈开源：https://github.com/TruthHun

分享，让知识传承更久远！ 感谢知识的创造者，感谢知识的分享者，也感谢每一位阅读到此处的读者，因为我们都将成为知识的传承者。

介绍

基础组件

框架为开发者提供了一系列基础组件，开发者可以通过组合这些基础组件进行快速开发。

什么是组件：

- 组件是视图层的基本组成单元。
- 组件自带一些功能与微信风格的样式。
- 一个组件通常包括开始标签和结束标签，属性用来修饰这个组件，内容在两个标签之内。

```
<tagname property="value">
  Content goes here ...
</tagname>
```

注意：所有组件与属性都是小写，以连字符-连接

属性类型

类型	描述	注解
Boolean	布尔值	组件写上该属性，不管该属性等于什么，其值都为 `true` ，只有组件上没有写该属性时，属性值才为 `false` 。如果属性值为变量，变量的值会被转换为Boolean类型
Number	数字	`1` , `2.5`
String	字符串	`"string"`
Array	数组	`[ 1, "string" ]`
Object	对象	`{ key: value }`
EventHandler	事件处理函数名	`"handlerName"` 是 Page中定义的事件处理函数名
Any	任意属性	

共同属性类型

所有组件都有的属性：

属性名	类型	描述	注解
id	String	组件的唯一标示	保持整个页面唯一
class	String	组件的样式类	在对应的 WXSS 中定义的样式类

style	String	组件的内联样式	可以动态设置的内联样式
hidden	Boolean	组件是否显示	所有组件默认显示
data-*	Any	自定义属性	组件上触发的事件时，会发送给事件处理函数
bind / *catch*	EventHandler	组件的事件	详见事件

特殊属性

几乎所有组件都有各自定义的属性，可以对该组件的功能或样式进行修饰，请参考各个组件的定义。

组件列表

基础组件分为以下七大类:

视图容器(View Container):

组件名	说明
view	视图容器
scroll-view	可滚动视图容器
swiper	滑块视图容器

基础内容(Basic Content):

组件名	说明
icon	图标
text	文字
progress	进度条

表单(Form):

标签名	说明
button	按钮
form	表单
input	输入框
checkbox	多项选择器
radio	单项选择器
picker	列表选择器
picker-view	内嵌列表选择器
slider	滚动选择器
switch	开关选择器
label	标签

导航(Navigation):

组件名	说明
navigator	应用链接

多媒体(Media):

组件名	说明
audio	音频
image	图片
video	视频

地图(Map):

组件名	说明
map	地图

画布(Canvas):

组件名	说明
canvas	画布

原文:

> *https://developers.weixin.qq.com/miniprogram/dev/component/*

视图容器

- view
- scroll-view
- swiper
- movable-view
- cover-view

view

view

视图容器。

属性名	类型	默认值	说明	最低版本
hover-class	String	none	指定按下去的样式类。当 `hover-class="none"` 时，没有点击态效果	
hover-stop-propagation	Boolean	false	指定是否阻止本节点的祖先节点出现点击态	1.5.0
hover-start-time	Number	50	按住后多久出现点击态，单位毫秒	
hover-stay-time	Number	400	手指松开后点击态保留时间，单位毫秒	

示例：

在开发者工具中预览效果

```
<view class="section">
  <view class="section__title">flex-direction: row</view>
  <view class="flex-wrp" style="flex-direction:row;">
    <view class="flex-item bc_green">1</view>
    <view class="flex-item bc_red">2</view>
    <view class="flex-item bc_blue">3</view>
  </view>
</view>
<view class="section">
  <view class="section__title">flex-direction: column</view>
  <view class="flex-wrp" style="height: 300px;flex-direction:column;">
    <view class="flex-item bc_green">1</view>
    <view class="flex-item bc_red">2</view>
    <view class="flex-item bc_blue">3</view>
  </view>
</view>
```

Bug & Tip

- tip: 如果需要使用滚动视图，请使用 scroll-view

原文:

> *https://developers.weixin.qq.com/miniprogram/dev/component/view.html*

scroll-view

scroll-view

可滚动视图区域。

属性名	类型	默认值	说明
scroll-x	Boolean	false	允许横向滚动
scroll-y	Boolean	false	允许纵向滚动
upper-threshold	Number	50	距顶部/左边多远时（单位px），触发 scrolltoupper 事件
lower-threshold	Number	50	距底部/右边多远时（单位px），触发 scrolltolower 事件
scroll-top	Number		设置竖向滚动条位置
scroll-left	Number		设置横向滚动条位置
scroll-into-view	String		值应为某子元素id（id不能以数字开头）。设置哪个方向可滚动，则在哪个方向滚动到该元素
scroll-with-animation	Boolean	false	在设置滚动条位置时使用动画过渡
enable-back-to-top	Boolean	false	iOS点击顶部状态栏、安卓双击标题栏时，滚动条返回顶部，只支持竖向
bindscrolltoupper	EventHandle		滚动到顶部/左边，会触发 scrolltoupper 事件
bindscrolltolower	EventHandle		滚动到底部/右边，会触发 scrolltolower 事件
bindscroll	EventHandle		滚动时触发，event.detail = {scrollLeft, scrollTop, scrollHeight, scrollWidth, deltaX, deltaY}

使用竖向滚动时，需要给 `<scroll-view/>` 一个固定高度，通过 WXSS 设置 height。

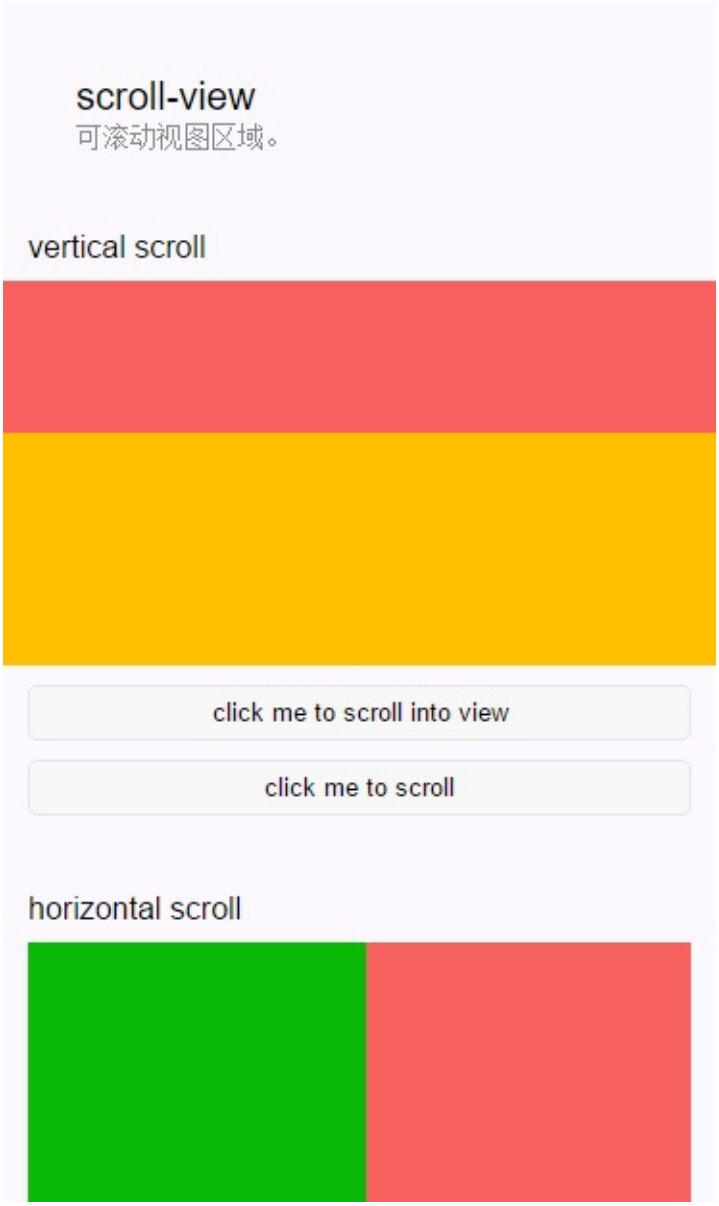
示例代码：

```
<view class="section">
  <view class="section__title">vertical scroll</view>
  <scroll-view scroll-y style="height: 200px;" bindscrolltoupper="upper" bindscrolltolower="lower" bindscroll="scroll" scroll-into-view="{{toView}}" scroll-top="{{scrollTop}}">
    <view id="green" class="scroll-view-item bc_green"></view>
    <view id="red"  class="scroll-view-item bc_red"></view>
    <view id="yellow" class="scroll-view-item bc_yellow"></view>
    <view id="blue" class="scroll-view-item bc_blue"></view>
  </scroll-view>

  <view class="btn-area">
    <button size="mini" bindtap="tap">click me to scroll into view </button>
    <button size="mini" bindtap="tapMove">click me to scroll</button>
  </view>
</view>
```

```
<view class="section section_gap">
  <view class="section__title">horizontal scroll</view>
  <scroll-view class="scroll-view_H" scroll-x style="width: 100%">
    <view id="green" class="scroll-view-item_H bc_green"></view>
    <view id="red"  class="scroll-view-item_H bc_red"></view>
    <view id="yellow" class="scroll-view-item_H bc_yellow"></view>
    <view id="blue" class="scroll-view-item_H bc_blue"></view>
  </scroll-view>
</view>
```

```
var order = ['red', 'yellow', 'blue', 'green', 'red']
Page({
  data: {
    toView: 'red',
    scrollTop: 100
  },
  upper: function(e) {
    console.log(e)
  },
  lower: function(e) {
    console.log(e)
  },
  scroll: function(e) {
    console.log(e)
  },
  tap: function(e) {
    for (var i = 0; i < order.length; ++i) {
      if (order[i] === this.data.toView) {
        this.setData({
          toView: order[i + 1]
        })
        break
      }
    }
  },
  tapMove: function(e) {
    this.setData({
      scrollTop: this.data.scrollTop + 10
    })
  }
})
```



Bug & Tip

- tip: 请勿在 scroll-view 中使用 textarea、map、canvas、video 组件
- tip: scroll-into-view 的优先级高于 scroll-top
- tip: 在滚动 scroll-view 时会阻止页面回弹，所以在 scroll-view 中滚动，是无法触发 onPullDownRefresh
- tip: 若要使用下拉刷新，请使用页面的滚动，而不是 scroll-view ，这样也能通过点击顶部状态栏回到页面顶部

原文：

> *https://developers.weixin.qq.com/miniprogram/dev/component/scroll-view.html*

swiper

swiper

滑块视图容器。

属性名	类型	默认值	说明	最低版本
indicator-dots	Boolean	false	是否显示面板指示点	
indicator-color	Color	rgba(0, 0, 0, .3)	指示点颜色	1.1.0
indicator-active-color	Color	#000000	当前选中的指示点颜色	1.1.0
autoplay	Boolean	false	是否自动切换	
current	Number	0	当前所在滑块的 index	
current-item-id	String	""	当前所在滑块的 item-id ，不能与 current 被同时指定	1.9.0
interval	Number	5000	自动切换时间间隔	
duration	Number	500	滑动动画时长	
circular	Boolean	false	是否采用衔接滑动	
vertical	Boolean	false	滑动方向是否为纵向	
previous-margin	String	"0px"	前边距，可用于露出前一项的一小部分，接受 px 和 rpx 值	1.9.0
next-margin	String	"0px"	后边距，可用于露出后一项的一小部分，接受 px 和 rpx 值	1.9.0
display-multiple-items	Number	1	同时显示的滑块数量	1.9.0
skip-hidden-item-layout	Boolean	false	是否跳过未显示的滑块布局，设为 true 可优化复杂情况下的滑动性能，但会丢失隐藏状态滑块的布局信息	1.9.0
bindchange	EventHandle		current 改变时会触发 change 事件，event.detail = {current: current, source: source}	
bindanimationfinish	EventHandle		动画结束时会触发 animationfinish 事件，event.detail 同上	1.9.0

从 1.4.0 开始， `change` 事件返回 `detail` 中包含一个 `source` 字段，表示导致变更的原因，可能值如下：

- autoplay 自动播放导致swiper变化；
- touch 用户划动引起swiper变化；
- 其他原因将用空字符串表示。

注意：其中只可放置 `<swiper-item/>` 组件，否则会导致未定义的行为。

swiper-item

仅可放置在 `<swiper/>` 组件中，宽高自动设置为100%。

属性名	类型	默认值	说明	最低版本
item-id	String	""	该 swiper-item 的标识符	1.9.0

示例代码：

在开发者工具中预览效果

```
<swiper indicator-dots="{{indicatorDots}}"
  autoplay="{{autoplay}}" interval="{{interval}}" duration="{{duration}}">
  <block wx:for="{{imgUrls}}">
    <swiper-item>
      <image src="{{item}}" class="slide-image" width="355" height="150"/>
    </swiper-item>
  </block>
</swiper>
<button bindtap="changeIndicatorDots"> indicator-dots </button>
<button bindtap="changeAutoplay"> autoplay </button>
<slider bindchange="intervalChange" show-value min="500" max="2000"/> interval
<slider bindchange="durationChange" show-value min="1000" max="10000"/> duration
```

```
Page({
  data: {
    imgUrls: [
      'http://img02.tooopen.com/images/20150928/tooopen_sy_143912755726.jpg',
      'http://img06.tooopen.com/images/20160818/tooopen_sy_175866434296.jpg',
      'http://img06.tooopen.com/images/20160818/tooopen_sy_175833047715.jpg'
    ],
    indicatorDots: false,
    autoplay: false,
    interval: 5000,
    duration: 1000
  },
  changeIndicatorDots: function(e) {
    this.setData({
      indicatorDots: !this.data.indicatorDots
    })
  },
  changeAutoplay: function(e) {
    this.setData({
      autoplay: !this.data.autoplay
    })
  },
  intervalChange: function(e) {
    this.setData({
      interval: e.detail.value
    })
  },
```

```
  durationChange: function(e) {
    this.setData({
      duration: e.detail.value
    })
  }
})
```

Bug & Tip

- tip: 如果在 bindchange 的事件回调函数中使用 setData 改变 current 值，则有可能导致 setData 被不停地调用，因而通常情况下请在改变 current 值前检测 source 字段来判断是否是由于用户触摸引起。

原文：

> *https://developers.weixin.qq.com/miniprogram/dev/component/swiper.html*

movable-view

movable-area

> 基础库 *1.2.0* 开始支持，低版本需做兼容处理

`movable-view` 的可移动区域

属性名	类型	默认值	说明	最低版本
scale-area	Boolean	false	当里面的movable-view设置为支持双指缩放时，设置此值可将缩放手势生效区域修改为整个movable-area	1.9.90

注意：**movable-area** 必须设置**width**和**height**属性，不设置默认为**10px**

movable-view

> 基础库 *1.2.0* 开始支持，低版本需做兼容处理

可移动的视图容器，在页面中可以拖拽滑动

属性名	类型	默认值	说明	最低版本
direction	String	none	movable-view的移动方向，属性值有all、vertical、horizontal、none	
inertia	Boolean	false	movable-view是否带有惯性	
out-of-bounds	Boolean	false	超过可移动区域后，movable-view是否还可以移动	
x	Number / String		定义x轴方向的偏移，如果x的值不在可移动范围内，会自动移动到可移动范围；改变x的值会触发动画	
y	Number / String		定义y轴方向的偏移，如果y的值不在可移动范围内，会自动移动到可移动范围；改变y的值会触发动画	
damping	Number	20	阻尼系数，用于控制x或y改变时的动画和过界回弹的动画，值越大移动越快	
friction	Number	2	摩擦系数，用于控制惯性滑动的动画，值越大摩擦力越大，滑动越快停止；必须大于0，否则会被设置成默认值	
disabled	Boolean	false	是否禁用	1.9.90
scale	Boolean	false	是否支持双指缩放，默认缩放手势生效区域是在movable-view内	1.9.90
scale-min	Number	0.5	定义缩放倍数最小值	1.9.90
scale-max	Number	10	定义缩放倍数最大值	1.9.90
scale-value	Number	1	定义缩放倍数，取值范围为 0.5 - 10	1.9.90
animation	Boolean	true	是否使用动画	2.1.0

bindchange	EventHandle		拖动过程中触发的事件，event.detail = {x: x, y: y, source: source}，其中source表示产生移动的原因，值可为touch（拖动）、touch-out-of-bounds（超出移动范围）、out-of-bounds（超出移动范围后的回弹）、friction（惯性）和空字符串（setData）	1.9.90
bindscale	EventHandle		缩放过程中触发的事件，event.detail = {x: x, y: y, scale: scale}，其中x和y字段在2.1.0之后开始支持返回	1.9.90

除了基本事件外，movable-view提供了两个特殊事件

类型	触发条件	最低版本
htouchmove	初次手指触摸后移动为横向的移动，如果catch此事件，则意味着touchmove事件也被catch	1.9.90
vtouchmove	初次手指触摸后移动为纵向的移动，如果catch此事件，则意味着touchmove事件也被catch	1.9.90

> *movable-view 必须设置width和height属性，不设置默认为10pxmovable-view 默认为绝对定位，top和left属性为0px当movable-view小于movable-area时，movable-view的移动范围是在movable-area内；当movable-view大于movable-area时，movable-view的移动范围必须包含movable-area（x轴方向和y轴方向分开考虑）*

注意：movable-view必须在 `<movable-area/>` 组件中，并且必须是直接子节点，否则不能移动。

示例代码：

在开发者工具中预览效果

```
<view class="section">
  <view class="section__title">movable-view区域小于movable-area</view>
  <movable-area style="height: 200px; width: 200px; background: red;">
    <movable-view style="height: 50px; width: 50px; background: blue;" x="{{x}}" y="{{y}}" direction="all">
    </movable-view>
  </movable-area>
  <view class="btn-area">
    <button size="mini" bindtap="tap">click me to move to (30px, 30px)</button>
  </view>
  <view class="section__title">movable-view区域大于movable-area</view>
  <movable-area style="height: 100px; width: 100px; background: red;">
    <movable-view style="height: 200px; width: 200px; background: blue;" direction="all">
    </movable-view>
  </movable-area>
  <view class="section__title">可放缩</view>
  <movable-area style="height: 200px; width: 200px; background: red;" scale-area>
    <movable-view style="height: 50px; width: 50px; background: blue;" direction="all" bindchange="onChange" bindscale="onScale" scale scale-min="0.5" scale-max="4" scale-value="2">
    </movable-view>
  </movable-area>
</view>
```

```
Page({
  data: {
```

```
    x: 0,
    y: 0
  },
  tap: function(e) {
    this.setData({
      x: 30,
      y: 30
    });
  },
  onChange: function(e) {
    console.log(e.detail)
  },
  onScale: function(e) {
    console.log(e.detail)
  }
})
```

原文：

> *https://developers.weixin.qq.com/miniprogram/dev/component/movable-view.html*

cover-view

cover-view

> 基础库 *1.4.0* 开始支持，低版本需做兼容处理

覆盖在原生组件之上的文本视图，可覆盖的原生组件包括 `map`、`video`、`canvas`、`camera`、`live-player`、`live-pusher`，只支持嵌套 `cover-view`、`cover-image`。

属性名	类型	默认值	说明	最低版本
scroll-top	Number		设置顶部滚动偏移量，仅在设置了 overflow-y: scroll 成为滚动元素后生效	2.1.0

cover-image

> 基础库 *1.4.0* 开始支持，低版本需做兼容处理

覆盖在原生组件之上的图片视图，可覆盖的原生组件同 `cover-view`，支持嵌套在cover-view里。

属性名	类型	默认值	说明	最低版本
src	String		图标路径，支持临时路径、网络地址（1.6.0起支持）。暂不支持base64格式。	
bindload	EventHandle		图片加载成功时触发	2.1.0
binderror	EventHandle		图片加载失败时触发	2.1.0

Bug & Tips

- tip: 基础库 2.1.0 起支持设置 scale rotate 的css样式，包括transition动画
- tip: 基础库 1.9.90 起 cover-view 支持 overflow: scroll，但不支持动态更新 overflow
- tip: 基础库 1.9.90 起最外层 cover-view 支持 position: fixed
- tip: 基础库 1.9.0 起支持插在 view 等标签下。在此之前只可嵌套在原生组件map、video、canvas、camera内，避免嵌套在其他组件内。
- tip: 基础库 1.6.0 起支持css transition动画，transition-property只支持transform(translateX, translateY)与opacity。
- tip: 基础库 1.6.0 起支持css opacity。
- tip: 事件模型遵循冒泡模型，但不会冒泡到原生组件。
- tip: 文本建议都套上cover-view标签，避免排版错误。
- tip: 只支持基本的定位、布局、文本样式。不支持设置单边的border、background-image、shadow、overflow: visible等。
- tip: 建议子节点不要溢出父节点
- tip: 默认设置的样式有: white-space: nowrap; line-height: 1.2; display: block;
- bug: 自定义组件嵌套 cover-view 时，自定义组件的 slot 及其父节点暂不支持通过 wx:if 控制显隐，否则会导致 cover-view 不显示

示例：

在开发者工具中预览效果

```
<video id="myVideo" src="http://wxsnsdy.tc.qq.com/105/20210/snsdyvideodownload?filekey=30280201010421301f0201690402534804102ca905ce620b1241b726bc41dcff44e00204012882540400&bizid=1023&hy=SH&fil
 controls="{{false}}" event-model="bubble">
  <cover-view class="controls">
    <cover-view class="play" bindtap="play">
      <cover-image class="img" src="/path/to/icon_play" />
    </cover-view>
    <cover-view class="pause" bindtap="pause">
      <cover-image class="img" src="/path/to/icon_pause" />
    </cover-view>
    <cover-view class="time">00:00</cover-view>
  </cover-view>
</video>
```

```
.controls {
  position: relative;
  top: 50%;
  height: 50px;
  margin-top: -25px;
  display: flex;
}
.play,.pause,.time {
  flex: 1;
  height: 100%;
}
.time {
  text-align: center;
  background-color: rgba(0, 0, 0, .5);
  color: white;
  line-height: 50px;
}
.img {
  width: 40px;
  height: 40px;
  margin: 5px auto;
}
```

```
Page({
  onReady() {
    this.videoCtx = wx.createVideoContext('myVideo')
  },
  play() {
    this.videoCtx.play()
  },
  pause() {
    this.videoCtx.pause()
```

```
  }
})
```

原文：

> *https://developers.weixin.qq.com/miniprogram/dev/component/cover-view.html*

基础内容

- icon
- text
- rich-text
- progress

icon

icon

图标。

属性名	类型	默认值	说明
type	String		icon的类型，有效值：success, success_no_circle, info, warn, waiting, cancel, download, search, clear
size	Number	23	icon的大小，单位px
color	Color		icon的颜色，同css的color

示例：

在开发者工具中预览效果

```
<view class="group">
  <block wx:for="{{iconSize}}">
    <icon type="success" size="{{item}}"/>
  </block>
</view>

<view class="group">
  <block wx:for="{{iconType}}">
    <icon type="{{item}}" size="40"/>
  </block>
</view>


<view class="group">
  <block wx:for="{{iconColor}}">
    <icon type="success" size="40" color="{{item}}"/>
  </block>
</view>
```

```
Page({
  data: {
    iconSize: [20, 30, 40, 50, 60, 70],
    iconColor: [
      'red', 'orange', 'yellow', 'green', 'rgb(0,255,255)', 'blue', 'purple'
    ],
    iconType: [
      'success', 'success_no_circle', 'info', 'warn', 'waiting', 'cancel', 'download', 'search', 'clear'
    ]
  }
})
```

```
![icon](http://static.bookstack.cn/projects/mp-widget/153b8804425958f2.png)
```

原文:

> *https://developers.weixin.qq.com/miniprogram/dev/component/icon.html*

text

text

文本。

属性名	类型	默认值	说明	最低版本
selectable	Boolean	false	文本是否可选	1.1.0
space	String	false	显示连续空格	1.4.0
decode	Boolean	false	是否解码	1.4.0

space 有效值:

值	说明
ensp	中文字符空格一半大小
emsp	中文字符空格大小
nbsp	根据字体设置的空格大小

Tips

- decode可以解析的有 < > & '
- 各个操作系统的空格标准并不一致。
- 组件内只支持 嵌套。
- 除了文本节点以外的其他节点都无法长按选中。
  示例:

在开发者工具中预览效果

```
<view class="btn-area">
  <view class="body-view">
    <text>{{text}}</text>
    <button bindtap="add">add line</button>
    <button bindtap="remove">remove line</button>
  </view>
</view>
```

```
var initData = 'this is first line\nthis is second line'
var extraLine = [];
Page({
  data: {
    text: initData
  },
  add: function(e) {
    extraLine.push('other line')
    this.setData({
      text: initData + '\n' + extraLine.join('\n')
```

```
    })
  },
  remove: function(e) {
    if (extraLine.length > 0) {
      extraLine.pop()
      this.setData({
        text: initData + '\n' + extraLine.join('\n')
      })
    }
  }
})
```

Bug & Tip

- bug : 基础库版本低于 2.1.0 时， `<text/>` 组件内嵌的 `<text/>` style 设置可能不会生效。

原文：

> *https://developers.weixin.qq.com/miniprogram/dev/component/text.html*

rich-text

rich-text

> 基础库 *1.4.0* 开始支持，低版本需做兼容处理

富文本。

属性	类型	默认值	说明	最低版本
nodes	Array / String	[]	节点列表 / HTML String	1.4.0

支持默认事件，包括：`tap`、`touchstart`、`touchmove`、`touchcancel`、`touchend`和`longtap`

nodes 属性推荐使用 **Array** 类型，由于组件会将 **String** 类型转换为 **Array** 类型，因而性能会有所下降

nodes

现支持两种节点，通过type来区分，分别是元素节点和文本节点，默认是元素节点，在富文本区域里显示的HTML节点

元素节点：**type = node**

属性	说明	类型	必填	备注
name	标签名	String	是	支持部分受信任的HTML节点
attrs	属性	Object	否	支持部分受信任的属性，遵循Pascal命名法
children	子节点列表	Array	否	结构和nodes一致

文本节点：**type = text**

属性	说明	类型	必填	备注
text	文本	String	是	支持entities

受信任的HTML节点及属性

全局支持class和style属性，不支持**id**属性。

节点	属性
a	
abbr	
b	
blockquote	
br	
code	
col	span, width
colgroup	span, width

dd	
del	
div	
dl	
dt	
em	
fieldset	
h1	
h2	
h3	
h4	
h5	
h6	
hr	
i	
img	alt, src, height, width
ins	
label	
legend	
li	
ol	start, type
p	
q	
span	
strong	
sub	
sup	
table	width
tbody	
td	colspan, height, rowspan, width
tfoot	
th	colspan, height, rowspan, width
thead	
tr	
ul	

示例:

在开发者工具中预览效果

```
<!-- rich-text.wxml -->
<rich-text nodes="{{nodes}}" bindtap="tap"></rich-text>
```

```
// rich-text.js
Page({
  data: {
    nodes: [{
      name: 'div',
      attrs: {
        class: 'div_class',
        style: 'line-height: 60px; color: red;'
      },
      children: [{
        type: 'text',
        text: 'Hello World!'
      }]
    }]
  },
  tap() {
    console.log('tap')
  }
})
```

Bug & Tip

- tip: nodes 不推荐使用 String 类型，性能会有所下降。
- tip: rich-text 组件内屏蔽所有节点的事件。
- tip: attrs 属性不支持 id ，支持 class 。
- tip: name 属性大小写不敏感。
- tip: 如果使用了不受信任的HTML节点，该节点及其所有子节点将会被移除。
- tip: img 标签仅支持网络图片。
- tip: 如果在自定义组件中使用 rich-text 组件，那么仅自定义组件的 wxss 样式对 rich-text 中的 class 生效。

原文：

> *https://developers.weixin.qq.com/miniprogram/dev/component/rich-text.html*

progress

progress

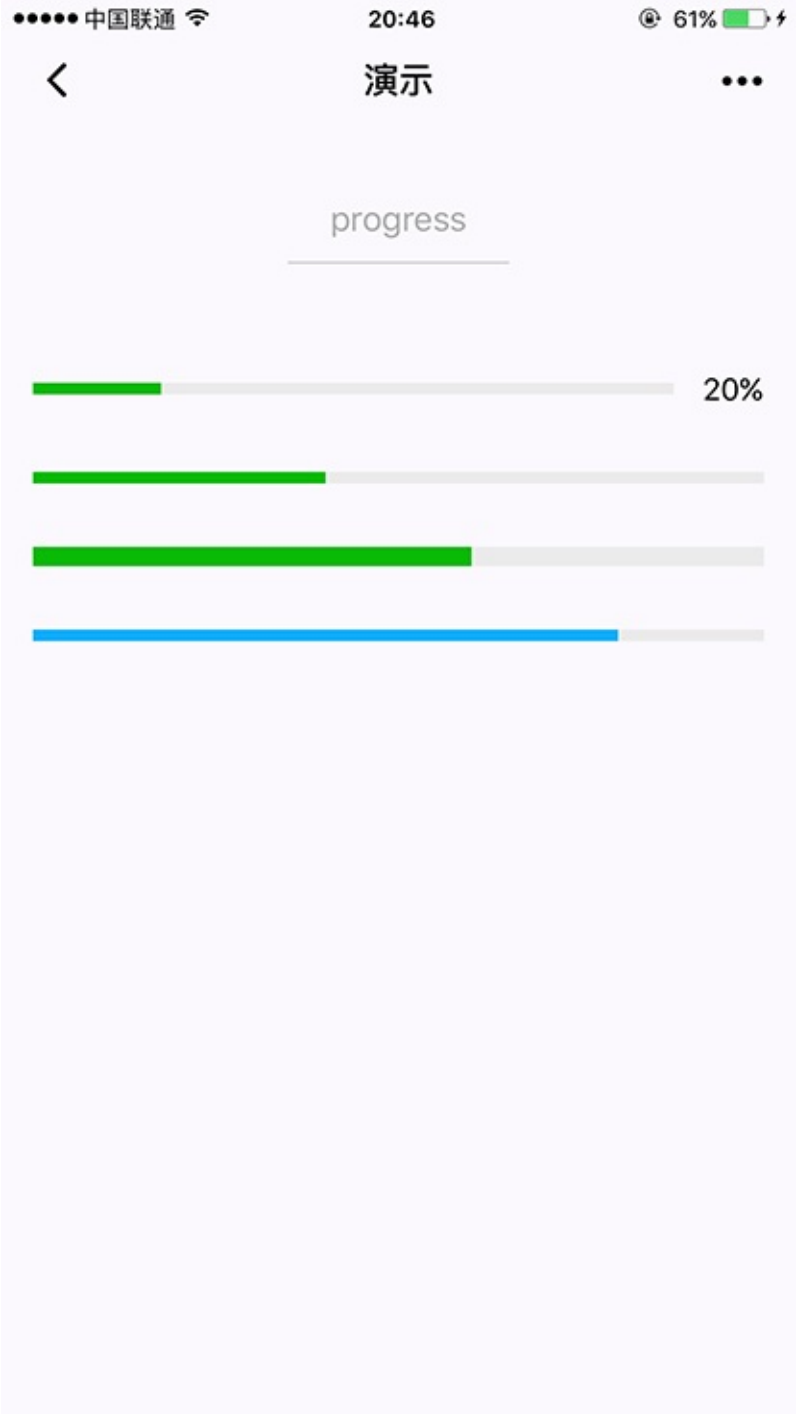
进度条。

属性名	类型	默认值	说明	最低版本
percent	Float	无	百分比0~100	
show-info	Boolean	false	在进度条右侧显示百分比	
stroke-width	Number	6	进度条线的宽度，单位px	
color	Color	#09BB07	进度条颜色 （请使用 activeColor）	
activeColor	Color		已选择的进度条的颜色	
backgroundColor	Color		未选择的进度条的颜色	
active	Boolean	false	进度条从左往右的动画	
active-mode	String	backwards	backwards：动画从头播；forwards：动画从上次结束点接着播	1.7.0

示例：

在开发者工具中预览效果

```
<progress percent="20" show-info />
<progress percent="40" stroke-width="12" />
<progress percent="60" color="pink" />
<progress percent="80" active />
```



原文：

> *https://developers.weixin.qq.com/miniprogram/dev/component/progress.html*

表单组件

- button
- checkbox
- form
- input
- label
- picker
- picker-view
- radio
- slider
- switch
- textarea

button

button

按钮。

属性名	类型	默认值	说明	生效时机
size	String	default	按钮的大小	
type	String	default	按钮的样式类型	
plain	Boolean	false	按钮是否镂空，背景色透明	
disabled	Boolean	false	是否禁用	
loading	Boolean	false	名称前是否带 loading 图标	
form-type	String		用于 `<form/>` 组件，点击分别会触发 `<form/>` 组件的 submit/reset 事件	
open-type	String		微信开放能力	
hover-class	String	button-hover	指定按钮按下去的样式类。当 `hover-class="none"` 时，没有点击态效果	
hover-stop-propagation	Boolean	false	指定是否阻止本节点的祖先节点出现点击态	
hover-start-time	Number	20	按住后多久出现点击态，单位毫秒	
hover-stay-time	Number	70	手指松开后点击态保留时间，单位毫秒	
lang	String	en	指定返回用户信息的语言，zh_CN 简体中文，zh_TW 繁体中文，en 英文。	open-type="getUserInfo"
bindgetuserinfo	Handler		用户点击该按钮时，会返回获取到的用户信息，回调的detail数据与wx.getUserInfo返回的一致	open-type="getUserInfo"
session-from	String		会话来源	open-type="contact"
send-message-title	String	当前标题	会话内消息卡片标题	open-type="contact"
send-message-path	String	当前分享路径	会话内消息卡片点击跳转小程序路径	open-type="contact"
send-message-img	String	截图	会话内消息卡片图片	open-type="contact"
show-message-card	Boolean	false	显示会话内消息卡片	open-type="contact"
bindcontact	Handler		客服消息回调	open-type="contact"

bindgetphonenumber	Handler		获取用户手机号回调	open-type="getPhoneNumber"
app-parameter	String		打开 APP 时，向 APP 传递的参数	open-type="launchApp"
binderror	Handler		当使用开放能力时，发生错误的回调	open-type="launchApp"
bindopensetting	Handler		在打开授权设置页后回调	open-type="openSetting"

- 注1: **button-hover** 默认为**{background-color: rgba(0, 0, 0, 0.1); opacity: 0.7;}**
- 注2: **bindgetphonenumber** 从**1.2.0** 开始支持，但是在**1.5.3**以下版本中无法使用**wx.canIUse**进行检测，建议使用基础库版本进行判断。
- 注3: 在**bindgetphonenumber** 等返回加密信息的回调中调用 **wx.login** 登录，可能会刷新登录态。此时服务器使用 **code** 换取的 **sessionKey** 不是加密时使用的 **sessionKey**，导致解密失败。建议开发者提前进行 **login**；或者在回调中先使用 **checkSession** 进行登录态检查，避免 **login** 刷新登录态。

size 有效值:

值	说明
default	默认大小
mini	小尺寸

type 有效值:

值	说明
primary	绿色
default	白色
warn	红色

form-type 有效值:

值	说明
submit	提交表单
reset	重置表单

open-type 有效值:

值	说明	最低版本
contact	打开客服会话	1.1.0
share	触发用户转发，使用前建议先阅读使用指引	1.2.0
getUserInfo	获取用户信息，可以从bindgetuserinfo回调中获取到用户信息	1.3.0
getPhoneNumber	获取用户手机号，可以从bindgetphonenumber回调中获取到用户信息，具体说明	1.2.0
launchApp	打开APP，可以通过app-parameter属性设定向APP传的参数具体说明	1.9.5
openSetting	打开授权设置页	2.0.7

feedback	打开“意见反馈”页面，用户可提交反馈内容并上传日志，开发者可以登录小程序管理后台后进入左侧菜单“客服反馈”页面获取到反馈内容	2.1.0

示例代码：

在开发者工具中预览效果

```
/** wxss **/
/** 修改button默认的点击态样式类**/
.button-hover {
  background-color: red;
}
/** 添加自定义button点击态样式类**/
.other-button-hover {
  background-color: blue;
}
```

```
<button type="default" size="{{defaultSize}}" loading="{{loading}}" plain="{{plain}}"
        disabled="{{disabled}}" bindtap="default" hover-class="other-button-hover"> default </button>
<button type="primary" size="{{primarySize}}" loading="{{loading}}" plain="{{plain}}"
        disabled="{{disabled}}" bindtap="primary"> primary </button>
<button type="warn" size="{{warnSize}}" loading="{{loading}}" plain="{{plain}}"
        disabled="{{disabled}}" bindtap="warn"> warn </button>
<button bindtap="setDisabled">点击设置以上按钮disabled属性</button>
<button bindtap="setPlain">点击设置以上按钮plain属性</button>
<button bindtap="setLoading">点击设置以上按钮loading属性</button>
<button open-type="contact">进入客服会话</button>
<button open-type="getUserInfo" lang="zh_CN" bindgetuserinfo="onGotUserInfo">获取用户信息</button>
<button open-type="openSetting">打开授权设置页</button>
```

```
var types = ['default', 'primary', 'warn']
var pageObject = {
  data: {
    defaultSize: 'default',
    primarySize: 'default',
    warnSize: 'default',
    disabled: false,
    plain: false,
    loading: false
  },
  setDisabled: function(e) {
    this.setData({
      disabled: !this.data.disabled
    })
  },
  setPlain: function(e) {
    this.setData({
      plain: !this.data.plain
    })
  },
  setLoading: function(e) {
```

```
    this.setData({
      loading: !this.data.loading
    })
  },
  onGotUserInfo: function(e) {
    console.log(e.detail.errMsg)
    console.log(e.detail.userInfo)
    console.log(e.detail.rawData)
  },
}

for (var i = 0; i < types.length; ++i) {
  (function(type) {
    pageObject[type] = function(e) {
      var key = type + 'Size'
      var changedData = {}
      changedData[key] =
        this.data[key] === 'default' ? 'mini' : 'default'
      this.setData(changedData)
    }
  })(types[i])
}

Page(pageObject)
```

原文：

> https://developers.weixin.qq.com/miniprogram/dev/component/button.html

checkbox

checkbox-group

多项选择器，内部由多个 `checkbox` 组成。

属性名	类型	默认值	说明
bindchange	EventHandle		`<checkbox-group/>` 中选中项发生改变是触发 change 事件，detail = {value:[选中的checkbox的value的数组]}

checkbox

多选项目。

属性名	类型	默认值	说明
value	String		`<checkbox/>` 标识，选中时触发 `<checkbox-group/>` 的 change 事件，并携带 `<checkbox/>` 的 value
disabled	Boolean	false	是否禁用
checked	Boolean	false	当前是否选中，可用来设置默认选中
color	Color		checkbox的颜色，同css的color

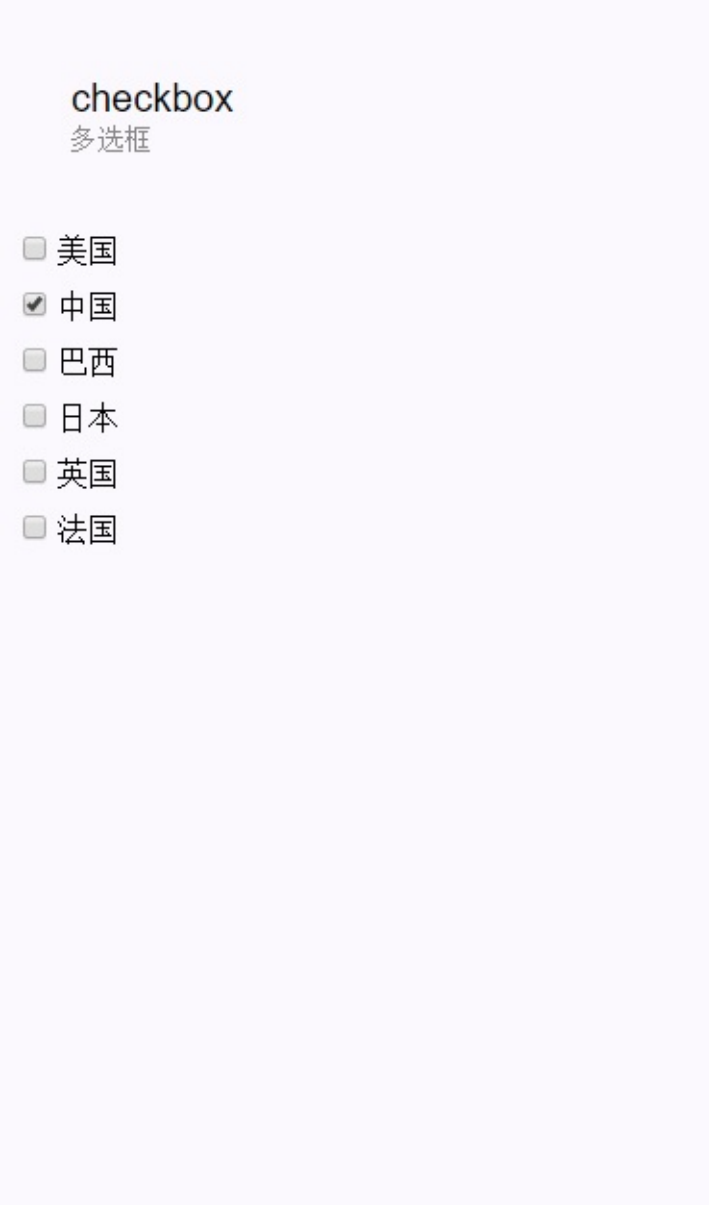
示例：

在开发者工具中预览效果

```
<checkbox-group bindchange="checkboxChange">
  <label class="checkbox" wx:for="{{items}}">
    <checkbox value="{{item.name}}" checked="{{item.checked}}"/>{{item.value}}
  </label>
</checkbox-group>
```

```
Page({
  data: {
    items: [
      {name: 'USA', value: '美国'},
      {name: 'CHN', value: '中国', checked: 'true'},
      {name: 'BRA', value: '巴西'},
      {name: 'JPN', value: '日本'},
      {name: 'ENG', value: '英国'},
      {name: 'TUR', value: '法国'},
    ]
  },
  checkboxChange: function(e) {
    console.log('checkbox发生change事件，携带value值为：', e.detail.value)
  }
```

```
15. })
```



原文：

> *https://developers.weixin.qq.com/miniprogram/dev/component/checkbox.html*

form

form

表单，将组件内的用户输入的 `<switch/>` `<input/>` `<checkbox/>` `<slider/>` `<radio/>` `<picker/>` 提交。

当点击 `<form/>` 表单中 formType 为 submit 的 `<button/>` 组件时，会将表单组件中的 value 值进行提交，需要在表单组件中加上 name 来作为 key。

属性名	类型	说明	最低版本
report-submit	Boolean	是否返回 formId 用于发送模板消息	
bindsubmit	EventHandle	携带 form 中的数据触发 submit 事件，event.detail = {value : {'name': 'value'} , formId: ''}	
bindreset	EventHandle	表单重置时会触发 reset 事件	

示例代码：

在开发者工具中预览效果

```
<form bindsubmit="formSubmit" bindreset="formReset">
  <view class="section section_gap">
    <view class="section__title">switch</view>
    <switch name="switch"/>
  </view>
  <view class="section section_gap">
    <view class="section__title">slider</view>
    <slider name="slider" show-value ></slider>
  </view>

  <view class="section">
    <view class="section__title">input</view>
    <input name="input" placeholder="please input here" />
  </view>
  <view class="section section_gap">
    <view class="section__title">radio</view>
    <radio-group name="radio-group">
      <label><radio value="radio1"/>radio1</label>
      <label><radio value="radio2"/>radio2</label>
    </radio-group>
  </view>
  <view class="section section_gap">
    <view class="section__title">checkbox</view>
    <checkbox-group name="checkbox">
      <label><checkbox value="checkbox1"/>checkbox1</label>
```

```
      <label><checkbox value="checkbox2"/>checkbox2</label>
    </checkbox-group>
  </view>
  <view class="btn-area">
    <button formType="submit">Submit</button>
    <button formType="reset">Reset</button>
  </view>
</form>
```

```
Page({
  formSubmit: function(e) {
    console.log('form发生了submit事件，携带数据为：', e.detail.value)
  },
  formReset: function() {
    console.log('form发生了reset事件')
  }
})
```

原文：

> *https://developers.weixin.qq.com/miniprogram/dev/component/form.html*

input

input

输入框。

属性名	类型	默认值	说明	最低版本
value	String		输入框的初始内容	
type	String	"text"	input 的类型	
password	Boolean	false	是否是密码类型	
placeholder	String		输入框为空时占位符	
placeholder-style	String		指定 placeholder 的样式	
placeholder-class	String	"input-placeholder"	指定 placeholder 的样式类	
disabled	Boolean	false	是否禁用	
maxlength	Number	140	最大输入长度，设置为 -1 的时候不限制最大长度	
cursor-spacing	Number	0	指定光标与键盘的距离，单位 px 。取 input 距离底部的距离和 cursor-spacing 指定的距离的最小值作为光标与键盘的距离	
auto-focus	Boolean	false	(即将废弃，请直接使用 focus)自动聚焦，拉起键盘	
focus	Boolean	false	获取焦点	
confirm-type	String	"done"	设置键盘右下角按钮的文字	1.1.0
confirm-hold	Boolean	false	点击键盘右下角按钮时是否保持键盘不收起	1.1.0
cursor	Number		指定focus时的光标位置	1.5.0
selection-start	Number	-1	光标起始位置，自动聚集时有效，需与selection-end搭配使用	1.9.0
selection-end	Number	-1	光标结束位置，自动聚集时有效，需与selection-start搭配使用	1.9.0
adjust-position	Boolean	true	键盘弹起时，是否自动上推页面	1.9.90
bindinput	EventHandle		键盘输入时触发，event.detail = {value, cursor, keyCode}，keyCode 为键值，2.1.0 起支持，处理函数可以直接 return 一个字符串，将替换输入框的内容。	
bindfocus	EventHandle		输入框聚焦时触发，event.detail = { value, height }，height 为键盘高度，在基础库 1.9.90 起支持	
			输入框失去焦点时触发，	

			event.detail = {value: value}
bindconfirm	EventHandle		点击完成按钮时触发，event.detail = {value: value}

type 有效值:

值	说明
text	文本输入键盘
number	数字输入键盘
idcard	身份证输入键盘
digit	带小数点的数字键盘

confirm-type 有效值:

值	说明
send	右下角按钮为“发送”
search	右下角按钮为“搜索”
next	右下角按钮为“下一个”
go	右下角按钮为“前往”
done	右下角按钮为“完成”

示例代码:

在开发者工具中预览效果

```
<!--input.wxml-->
<view class="section">
  <input placeholder="这是一个可以自动聚焦的input" auto-focus/>
</view>
<view class="section">
  <input placeholder="这个只有在按钮点击的时候才聚焦" focus="{{focus}}" />
  <view class="btn-area">
    <button bindtap="bindButtonTap">使得输入框获取焦点</button>
  </view>
</view>
<view class="section">
  <input  maxlength="10" placeholder="最大输入长度10" />
</view>
<view class="section">
  <view class="section__title">你输入的是：{{inputValue}}</view>
  <input  bindinput="bindKeyInput" placeholder="输入同步到view中"/>
</view>
<view class="section">
  <input  bindinput="bindReplaceInput" placeholder="连续的两个1会变成2" />
</view>
<view class="section">
  <input password type="number" />
</view>
```

```
</view>
<view class="section">
  <input password type="text" />
</view>
<view class="section">
  <input type="digit" placeholder="带小数点的数字键盘"/>
</view>
<view class="section">
  <input type="idcard" placeholder="身份证输入键盘" />
</view>
<view class="section">
  <input placeholder-style="color:red" placeholder="占位符字体是红色的" />
</view>
```

```
//input.js
Page({
  data: {
    focus: false,
    inputValue: ''
  },
  bindButtonTap: function() {
    this.setData({
      focus: true
    })
  },
  bindKeyInput: function(e) {
    this.setData({
      inputValue: e.detail.value
    })
  },
  bindReplaceInput: function(e) {
    var value = e.detail.value
    var pos = e.detail.cursor
    if(pos != -1){
      //光标在中间
      var left = e.detail.value.slice(0,pos)
      //计算光标的位置
      pos = left.replace(/11/g,'2').length
    }

    //直接返回对象，可以对输入进行过滤处理，同时可以控制光标的位置
    return {
      value: value.replace(/11/g,'2'),
      cursor: pos
    }

    //或者直接返回字符串,光标在最后边
    //return value.replace(/11/g,'2'),
  }
})
```

Bug & Tip

- bug : 微信版本 6.3.30, focus 属性设置无效;
- bug : 微信版本 6.3.30, placeholder 在聚焦时出现重影问题;
- tip : input 组件是一个 native 组件, 字体是系统字体, 所以无法设置 font-family;
- tip : 在 input 聚焦期间, 避免使用 css 动画; <

原文:

> *https://developers.weixin.qq.com/miniprogram/dev/component/input.html*

label

label

用来改进表单组件的可用性，使用 `for` 属性找到对应的 `id` ，或者将控件放在该标签下，当点击时，就会触发对应的控件。

`for` 优先级高于内部控件，内部有多个控件的时候默认触发第一个控件。

目前可以绑定的控件有： `<button/>` ， `<checkbox/>` ， `<radio/>` ， `<switch/>` 。

属性名	类型	说明
for	String	绑定控件的 id

示例代码：

在开发者工具中预览效果

```
<view class="section section_gap">
<view class="section__title">表单组件在label内</view>
<checkbox-group class="group" bindchange="checkboxChange">
  <view class="label-1" wx:for="{{checkboxItems}}">
    <label>
      <checkbox hidden value="{{item.name}}" checked="{{item.checked}}"></checkbox>
      <view class="label-1__icon">
        <view class="label-1__icon-checked" style="opacity:{{item.checked ? 1: 0}}"></view>
      </view>
      <text class="label-1__text">{{item.value}}</text>
    </label>
  </view>
</checkbox-group>
</view>

<view class="section section_gap">
<view class="section__title">label用for标识表单组件</view>
<radio-group class="group" bindchange="radioChange">
  <view class="label-2" wx:for="{{radioItems}}">
    <radio id="{{item.name}}" hidden value="{{item.name}}" checked="{{item.checked}}"></radio>
    <view class="label-2__icon">
      <view class="label-2__icon-checked" style="opacity:{{item.checked ? 1: 0}}"></view>
    </view>
    <label class="label-2__text" for="{{item.name}}"><text>{{item.name}}</text></label>
  </view>
</radio-group>
</view>
```

```
Page({
  data: {
    checkboxItems: [
```

```
      {name: 'USA', value: '美国'},
      {name: 'CHN', value: '中国', checked: 'true'},
      {name: 'BRA', value: '巴西'},
      {name: 'JPN', value: '日本', checked: 'true'},
      {name: 'ENG', value: '英国'},
      {name: 'TUR', value: '法国'},
    ],
    radioItems: [
      {name: 'USA', value: '美国'},
      {name: 'CHN', value: '中国', checked: 'true'},
      {name: 'BRA', value: '巴西'},
      {name: 'JPN', value: '日本'},
      {name: 'ENG', value: '英国'},
      {name: 'TUR', value: '法国'},
    ],
    hidden: false
  },
  checkboxChange: function(e) {
    var checked = e.detail.value
    var changed = {}
    for (var i = 0; i < this.data.checkboxItems.length; i ++) {
      if (checked.indexOf(this.data.checkboxItems[i].name) !== -1) {
        changed['checkboxItems['+i+'].checked'] = true
      } else {
        changed['checkboxItems['+i+'].checked'] = false
      }
    }
    this.setData(changed)
  },
  radioChange: function(e) {
    var checked = e.detail.value
    var changed = {}
    for (var i = 0; i < this.data.radioItems.length; i ++) {
      if (checked.indexOf(this.data.radioItems[i].name) !== -1) {
        changed['radioItems['+i+'].checked'] = true
      } else {
        changed['radioItems['+i+'].checked'] = false
      }
    }
    this.setData(changed)
  }
})
```

```
.label-1, .label-2{
    margin-bottom: 15px;
}
.label-1__text, .label-2__text {
    display: inline-block;
    vertical-align: middle;
}

.label-1__icon {
```

```
    position: relative;
    margin-right: 10px;
    display: inline-block;
    vertical-align: middle;
    width: 18px;
    height: 18px;
    background: #fcfff4;
}

.label-1__icon-checked {
    position: absolute;
    top: 3px;
    left: 3px;
    width: 12px;
    height: 12px;
    background: #1aad19;
}


.label-2__icon {
    position: relative;
    display: inline-block;
    vertical-align: middle;
    margin-right: 10px;
    width: 18px;
    height: 18px;
    background: #fcfff4;
    border-radius: 50px;
}

.label-2__icon-checked {
    position: absolute;
    left: 3px;
    top: 3px;
    width: 12px;
    height: 12px;
    background: #1aad19;
    border-radius: 50%;
}

.label-4_text{
    text-align: center;
    margin-top: 15px;
}
```

原文：

> https://developers.weixin.qq.com/miniprogram/dev/component/label.html

picker

picker

从底部弹起的滚动选择器，现支持五种选择器，通过mode来区分，分别是普通选择器，多列选择器，时间选择器，日期选择器，省市区选择器，默认是普通选择器。

普通选择器：mode = selector

属性名	类型	默认值	说明	最低版本
range	Array / Object Array	[]	mode为 selector 或 multiSelector 时，range 有效	
range-key	String		当 range 是一个 Object Array 时，通过 range-key 来指定 Object 中 key 的值作为选择器显示内容	
value	Number	0	value 的值表示选择了 range 中的第几个（下标从 0 开始）	
bindchange	EventHandle		value 改变时触发 change 事件，event.detail = {value: value}	
disabled	Boolean	false	是否禁用	
bindcancel	EventHandle		取消选择或点遮罩层收起 picker 时触发	1.9.90

多列选择器：mode = multiSelector（最低版本：1.4.0）

属性名	类型	默认值	说明	最低版本
range	二维Array / 二维Object Array	[]	mode为 selector 或 multiSelector 时，range 有效。二维数组，长度表示多少列，数组的每项表示每列的数据，如 `[["a","b"], ["c","d"]]`	
range-key	String		当 range 是一个 二维Object Array 时，通过 range-key 来指定 Object 中 key 的值作为选择器显示内容	
value	Array	[]	value 每一项的值表示选择了 range 对应项中的第几个（下标从 0 开始）	
bindchange	EventHandle		value 改变时触发 change 事件，event.detail = {value: value}	
bindcolumnchange	EventHandle		某一列的值改变时触发 columnchange 事件，event.detail = {column: column, value: value}，column 的值表示改变了第几列（下标从0开始），value 的值表示变更值的下标	
bindcancel	EventHandle		取消选择时触发	1.9.90
disabled	Boolean	false	是否禁用	

时间选择器：mode = time

属性名	类型	默认值	说明	最低版本
value	String		表示选中的时间，格式为"hh:mm"	
start	String		表示有效时间范围的开始，字符串格式为"hh:mm"	
end	String		表示有效时间范围的结束，字符串格式为"hh:mm"	
bindchange	EventHandle		value 改变时触发 change 事件，event.detail = {value: value}	
bindcancel	EventHandle		取消选择时触发	1.9.90
disabled	Boolean	false	是否禁用	

日期选择器：mode = date

属性名	类型	默认值	说明	最低版本
value	String	0	表示选中的日期，格式为"YYYY-MM-DD"	
start	String		表示有效日期范围的开始，字符串格式为"YYYY-MM-DD"	
end	String		表示有效日期范围的结束，字符串格式为"YYYY-MM-DD"	
fields	String	day	有效值 year,month,day，表示选择器的粒度	
bindchange	EventHandle		value 改变时触发 change 事件，event.detail = {value: value}	
bindcancel	EventHandle		取消选择时触发	1.9.90
disabled	Boolean	false	是否禁用	

fields 有效值：

值	说明
year	选择器粒度为年
month	选择器粒度为月份
day	选择器粒度为天

省市区选择器：mode = region（最低版本：1.4.0）

属性名	类型	默认值	说明	最低版本
value	Array	[]	表示选中的省市区，默认选中每一列的第一个值	
custom-item	String		可为每一列的顶部添加一个自定义的项	1.5.0
bindchange	EventHandle		value 改变时触发 change 事件，event.detail = {value: value}	
bindcancel	EventHandle		取消选择时触发	1.9.90
disabled	Boolean	false	是否禁用	

示例代码：

在开发者工具中预览效果

```
<view class="section">
  <view class="section__title">普通选择器</view>
  <picker bindchange="bindPickerChange" value="{{index}}" range="{{array}}">
    <view class="picker">
      当前选择：{{array[index]}}
    </view>
  </picker>
</view>
<view class="section">
  <view class="section__title">多列选择器</view>
  <picker mode="multiSelector" bindchange="bindMultiPickerChange" bindcolumnchange="bindMultiPickerColumnChange" value="{{multiIndex}}" range="{{multiArray}}">
    <view class="picker">
      当前选择：{{multiArray[0][multiIndex[0]]}}, {{multiArray[1][multiIndex[1]]}}, {{multiArray[2][multiIndex[2]]}}
    </view>
  </picker>
</view>
<view class="section">
  <view class="section__title">时间选择器</view>
  <picker mode="time" value="{{time}}" start="09:01" end="21:01" bindchange="bindTimeChange">
    <view class="picker">
      当前选择: {{time}}
    </view>
  </picker>
</view>

<view class="section">
  <view class="section__title">日期选择器</view>
  <picker mode="date" value="{{date}}" start="2015-09-01" end="2017-09-01" bindchange="bindDateChange">
    <view class="picker">
      当前选择: {{date}}
    </view>
  </picker>
</view>
<view class="section">
  <view class="section__title">省市区选择器</view>
  <picker mode="region" bindchange="bindRegionChange" value="{{region}}" custom-item="{{customItem}}">
    <view class="picker">
      当前选择：{{region[0]}}, {{region[1]}}, {{region[2]}}
    </view>
  </picker>
</view>
```

```
Page({
  data: {
    array: ['美国', '中国', '巴西', '日本'],
    objectArray: [
```

```
      {
        id: 0,
        name: '美国'
      },
      {
        id: 1,
        name: '中国'
      },
      {
        id: 2,
        name: '巴西'
      },
      {
        id: 3,
        name: '日本'
      }
    ],
    index: 0,
    multiArray: [['无脊柱动物', '脊柱动物'], ['扁性动物', '线形动物', '环节动物', '软体动物', '节肢动物'], ['猪肉绦虫', '吸血虫']],
    objectMultiArray: [
      [
        {
          id: 0,
          name: '无脊柱动物'
        },
        {
          id: 1,
          name: '脊柱动物'
        }
      ], [
        {
          id: 0,
          name: '扁性动物'
        },
        {
          id: 1,
          name: '线形动物'
        },
        {
          id: 2,
          name: '环节动物'
        },
        {
          id: 3,
          name: '软体动物'
        },
        {
          id: 3,
          name: '节肢动物'
        }
      ], [
        {
          id: 0,
```

```
          name: '猪肉绦虫'
        },
        {
          id: 1,
          name: '吸血虫'
        }
      ]
    ],
    multiIndex: [0, 0, 0],
    date: '2016-09-01',
    time: '12:01',
    region: ['广东省', '广州市', '海珠区'],
    customItem: '全部'
  },
  bindPickerChange: function(e) {
    console.log('picker发送选择改变，携带值为', e.detail.value)
    this.setData({
      index: e.detail.value
    })
  },
  bindMultiPickerChange: function (e) {
    console.log('picker发送选择改变，携带值为', e.detail.value)
    this.setData({
      multiIndex: e.detail.value
    })
  },
  bindMultiPickerColumnChange: function (e) {
    console.log('修改的列为', e.detail.column, '，值为', e.detail.value);
    var data = {
      multiArray: this.data.multiArray,
      multiIndex: this.data.multiIndex
    };
    data.multiIndex[e.detail.column] = e.detail.value;
    switch (e.detail.column) {
      case 0:
        switch (data.multiIndex[0]) {
          case 0:
            data.multiArray[1] = ['扁性动物', '线形动物', '环节动物', '软体动物', '节肢动物'];
            data.multiArray[2] = ['猪肉绦虫', '吸血虫'];
            break;
          case 1:
            data.multiArray[1] = ['鱼', '两栖动物', '爬行动物'];
            data.multiArray[2] = ['鲫鱼', '带鱼'];
            break;
        }
        data.multiIndex[1] = 0;
        data.multiIndex[2] = 0;
        break;
      case 1:
        switch (data.multiIndex[0]) {
          case 0:
            switch (data.multiIndex[1]) {
              case 0:
```

```
                data.multiArray[2] = ['猪肉绦虫', '吸血虫'];
                break;
              case 1:
                data.multiArray[2] = ['蛔虫'];
                break;
              case 2:
                data.multiArray[2] = ['蚂蚁', '蚂蟥'];
                break;
              case 3:
                data.multiArray[2] = ['河蚌', '蜗牛', '蛞蝓'];
                break;
              case 4:
                data.multiArray[2] = ['昆虫', '甲壳动物', '蛛形动物', '多足动物'];
                break;
            }
            break;
          case 1:
            switch (data.multiIndex[1]) {
              case 0:
                data.multiArray[2] = ['鲫鱼', '带鱼'];
                break;
              case 1:
                data.multiArray[2] = ['青蛙', '娃娃鱼'];
                break;
              case 2:
                data.multiArray[2] = ['蜥蜴', '龟', '壁虎'];
                break;
            }
            break;
        }
        data.multiIndex[2] = 0;
        console.log(data.multiIndex);
        break;
    }
    this.setData(data);
  },
  bindDateChange: function(e) {
    console.log('picker发送选择改变，携带值为', e.detail.value)
    this.setData({
      date: e.detail.value
    })
  },
  bindTimeChange: function(e) {
    console.log('picker发送选择改变，携带值为', e.detail.value)
    this.setData({
      time: e.detail.value
    })
  },
  bindRegionChange: function (e) {
    console.log('picker发送选择改变，携带值为', e.detail.value)
    this.setData({
      region: e.detail.value
    })
```

```
164.   }
165. })
```

原文：

> *https://developers.weixin.qq.com/miniprogram/dev/component/picker.html*

picker-view

picker-view

嵌入页面的滚动选择器

属性名	类型	说明	最低版本
value	NumberArray	数组中的数字依次表示 picker-view 内的 picker-view-column 选择的第几项（下标从 0 开始），数字大于 picker-view-column 可选项长度时，选择最后一项。	
indicator-style	String	设置选择器中间选中框的样式	
indicator-class	String	设置选择器中间选中框的类名	1.1.0
mask-style	String	设置蒙层的样式	1.5.0
mask-class	String	设置蒙层的类名	1.5.0
bindchange	EventHandle	当滚动选择，value 改变时触发 change 事件，event.detail = {value: value}；value为数组，表示 picker-view 内的 picker-view-column 当前选择的是第几项（下标从 0 开始）	

注意：其中只可放置 `<picker-view-column/>` 组件，其他节点不会显示。

picker-view-column

仅可放置于 `<picker-view />` 中，其孩子节点的高度会自动设置成与picker-view的选中框的高度一致

示例代码：

在开发者工具中预览效果

```
<view>
  <view>{{year}}年{{month}}月{{day}}日</view>
  <picker-view indicator-style="height: 50px;" style="width: 100%; height: 300px;" value="{{value}}" bindchange="bindChange">
    <picker-view-column>
      <view wx:for="{{years}}" style="line-height: 50px">{{item}}年</view>
    </picker-view-column>
    <picker-view-column>
      <view wx:for="{{months}}" style="line-height: 50px">{{item}}月</view>
    </picker-view-column>
    <picker-view-column>
      <view wx:for="{{days}}" style="line-height: 50px">{{item}}日</view>
    </picker-view-column>
  </picker-view>
</view>
```

```
const date = new Date()
const years = []
const months = []
const days = []

for (let i = 1990; i <= date.getFullYear(); i++) {
  years.push(i)
}

for (let i = 1 ; i <= 12; i++) {
  months.push(i)
}

for (let i = 1 ; i <= 31; i++) {
  days.push(i)
}

Page({
  data: {
    years: years,
    year: date.getFullYear(),
    months: months,
    month: 2,
    days: days,
    day: 2,
    value: [9999, 1, 1],
  },
  bindChange: function(e) {
    const val = e.detail.value
    this.setData({
      year: this.data.years[val[0]],
      month: this.data.months[val[1]],
      day: this.data.days[val[2]]
    })
  }
})
```

![picker_view(http://static.bookstack.cn/projects/mp-widget/153b884c700c7b7c.png)

Tips

- tip: 滚动时在iOS自带振动反馈，可在系统设置 -> 声音与触感 -> 系统触感反馈中关闭<

原文:

> *https://developers.weixin.qq.com/miniprogram/dev/component/picker-view.html*

radio

radio-group

单项选择器，内部由多个 `<radio/>` 组成。

属性名	类型	默认值	说明
bindchange	EventHandle		`<radio-group/>` 中的选中项发生变化时触发 change 事件，event.detail = {value: 选中项radio的value}

radio

单选项目

属性名	类型	默认值	说明
value	String		`<radio/>` 标识。当该 `<radio/>` 选中时，`<radio-group/>` 的 change 事件会携带 `<radio/>` 的value
checked	Boolean	false	当前是否选中
disabled	Boolean	false	是否禁用
color	Color		radio的颜色，同css的color

示例：

在开发者工具中预览效果

```
<radio-group class="radio-group" bindchange="radioChange">
  <label class="radio" wx:for="{{items}}">
    <radio value="{{item.name}}" checked="{{item.checked}}"/>{{item.value}}
  </label>
</radio-group>
```

```
Page({
  data: {
    items: [
      {name: 'USA', value: '美国'},
      {name: 'CHN', value: '中国', checked: 'true'},
      {name: 'BRA', value: '巴西'},
      {name: 'JPN', value: '日本'},
      {name: 'ENG', value: '英国'},
      {name: 'TUR', value: '法国'},
    ]
  },
  radioChange: function(e) {
    console.log('radio发生change事件，携带value值为：', e.detail.value)
  }
```

```
15. })
```

原文：

> *https://developers.weixin.qq.com/miniprogram/dev/component/radio.html*

slider

slider

滑动选择器。

属性名	类型	默认值	说明	最低版本
min	Number	0	最小值	
max	Number	100	最大值	
step	Number	1	步长，取值必须大于 0，并且可被(max - min)整除	
disabled	Boolean	false	是否禁用	
value	Number	0	当前取值	
color	Color	#e9e9e9	背景条的颜色（请使用 backgroundColor）	
selected-color	Color	#1aad19	已选择的颜色（请使用 activeColor）	
activeColor	Color	#1aad19	已选择的颜色	
backgroundColor	Color	#e9e9e9	背景条的颜色	
block-size	Number	28	滑块的大小，取值范围为 12 - 28	1.9.0
block-color	Color	#ffffff	滑块的颜色	1.9.0
show-value	Boolean	false	是否显示当前 value	
bindchange	EventHandle		完成一次拖动后触发的事件，event.detail = {value: value}	
bindchanging	EventHandle		拖动过程中触发的事件，event.detail = {value: value}	1.7.0

示例代码：

在开发者工具中预览效果

```
<view class="section section_gap">
  <text class="section__title">设置step</text>
  <view class="body-view">
    <slider bindchange="slider2change" step="5"/>
  </view>
</view>

<view class="section section_gap">
  <text class="section__title">显示当前value</text>
  <view class="body-view">
    <slider bindchange="slider3change" show-value/>
  </view>
</view>

```

```
<view class="section section_gap">
  <text class="section__title">设置最小/最大值</text>
  <view class="body-view">
    <slider bindchange="slider4change" min="50" max="200" show-value/>
  </view>
</view>
```

```
var pageData = {}
for (var i = 1; i < 5; i++) {
  (function (index) {
    pageData['slider' + index + 'change'] = function(e) {
      console.log('slider' + 'index' + '发生 change 事件，携带值为', e.detail.value)
    }
  })(i)
}
Page(pageData)
```

原文：

> *https://developers.weixin.qq.com/miniprogram/dev/component/slider.html*

switch

switch

开关选择器。

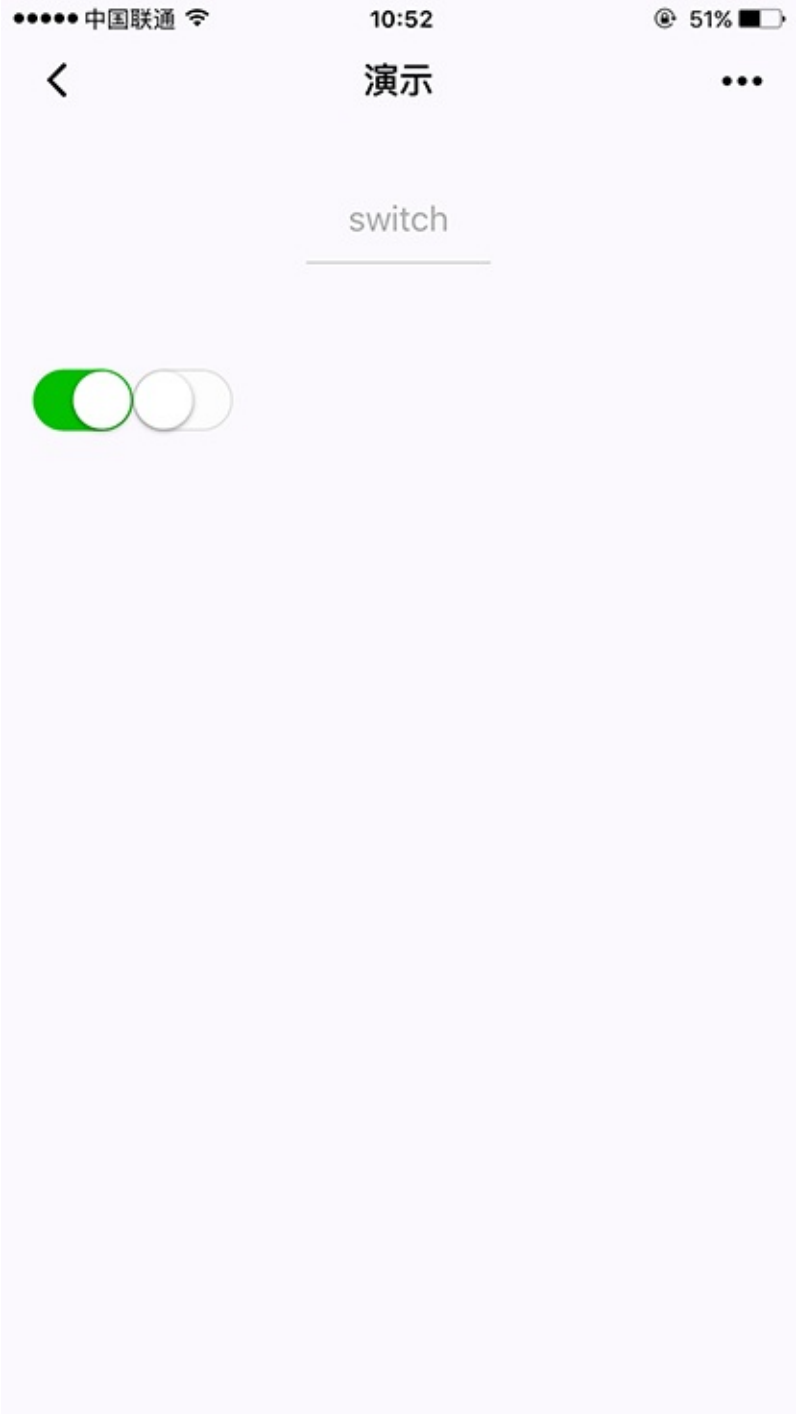
属性名	类型	默认值	说明
checked	Boolean	false	是否选中
type	String	switch	样式，有效值：switch，checkbox
bindchange	EventHandle		checked 改变时触发 change 事件，event.detail={value:checked}
color	Color		switch 的颜色，同 css 的 color

示例：

在开发者工具中预览效果

```
<view class="body-view">
    <switch checked bindchange="switch1Change"/>
    <switch bindchange="switch2Change"/>
</view>
```

```
Page({
  switch1Change: function (e){
    console.log('switch1 发生 change 事件，携带值为', e.detail.value)
  },
  switch2Change: function (e){
    console.log('switch2 发生 change 事件，携带值为', e.detail.value)
  }
})
```



Tips

- tip: switch类型切换时在iOS自带振动反馈，可在系统设置 -> 声音与触感 -> 系统触感反馈中关闭

原文：

> *https://developers.weixin.qq.com/miniprogram/dev/component/switch.html*

textarea

textarea

多行输入框。

属性名	类型	默认值	说明	最低版本
value	String		输入框的内容	
placeholder	String		输入框为空时占位符	
placeholder-style	String		指定 placeholder 的样式	
placeholder-class	String	textarea-placeholder	指定 placeholder 的样式类	
disabled	Boolean	false	是否禁用	
maxlength	Number	140	最大输入长度，设置为 -1 的时候不限制最大长度	
auto-focus	Boolean	false	自动聚焦，拉起键盘。	
focus	Boolean	false	获取焦点	
auto-height	Boolean	false	是否自动增高，设置auto-height时，style.height不生效	
fixed	Boolean	false	如果 textarea 是在一个 `position:fixed` 的区域，需要显示指定属性 fixed 为 true	
cursor-spacing	Number	0	指定光标与键盘的距离，单位 px 。取 textarea 距离底部的距离和 cursor-spacing 指定的距离的最小值作为光标与键盘的距离	
cursor	Number		指定focus时的光标位置	1.5.0
show-confirm-bar	Boolean	true	是否显示键盘上方带有”完成“按钮那一栏	1.6.0
selection-start	Number	-1	光标起始位置，自动聚集时有效，需与selection-end搭配使用	1.9.0
selection-end	Number	-1	光标结束位置，自动聚集时有效，需与selection-start搭配使用	1.9.0
adjust-position	Boolean	true	键盘弹起时，是否自动上推页面	1.9.90
bindfocus	EventHandle		输入框聚焦时触发，event.detail = { value, height }，height为键盘高度，在基础库 1.9.90 起支持	
bindblur	EventHandle		输入框失去焦点时触发，event.detail = {value, cursor}	
bindlinechange	EventHandle		输入框行数变化时调用，event.detail = {height: 0,	

			heightRpx: 0, lineCount: 0}
bindinput	EventHandle		当键盘输入时，触发 input 事件，event.detail = {value, cursor}， **bindinput** 处理函数的返回值并不会反映到 **textarea** 上
bindconfirm	EventHandle		点击完成时， 触发 confirm 事件，event.detail = {value: value}

示例代码：

在开发者工具中预览效果

```
<!--textarea.wxml-->
<view class="section">
  <textarea bindblur="bindTextAreaBlur" auto-height placeholder="自动变高" />
</view>
<view class="section">
  <textarea placeholder="placeholder颜色是红色的" placeholder-style="color:red;"  />
</view>
<view class="section">
  <textarea placeholder="这是一个可以自动聚焦的textarea" auto-focus />
</view>
<view class="section">
  <textarea placeholder="这个只有在按钮点击的时候才聚焦" focus="{{focus}}" />
  <view class="btn-area">
    <button bindtap="bindButtonTap">使得输入框获取焦点</button>
  </view>
</view>
<view class="section">
  <form bindsubmit="bindFormSubmit">
    <textarea placeholder="form 中的 textarea" name="textarea"/>
    <button form-type="submit"> 提交 </button>
  </form>
</view>
```

```
//textarea.js
Page({
  data: {
    height: 20,
    focus: false
  },
  bindButtonTap: function() {
    this.setData({
      focus: true
    })
  },
  bindTextAreaBlur: function(e) {
    console.log(e.detail.value)
  },
  bindFormSubmit: function(e) {
    console.log(e.detail.value.textarea)
```

```
  }
})
```

Bug & Tip

- bug: 微信版本 6.3.30，textarea 在列表渲染时，新增加的 textarea 在自动聚焦时的位置计算错误。
- tip: textarea 的 blur 事件会晚于页面上的 tap 事件，如果需要在 button 的点击事件获取 textarea，可以使用 form 的 bindsubmit。
- tip: 不建议在多行文本上对用户的输入进行修改，所以 textarea 的 bindinput 处理函数并不会将返回值反映到 textarea 上。
- tip: textarea 组件是由客户端创建的原生组件，它的层级是最高的，不能通过 z-index 控制层级。
- tip: 请勿在 scroll-view、swiper、picker-view、movable-view 中使用 textarea 组件。
- tip: css 动画对 textarea 组件无效。

原文：

> *https://developers.weixin.qq.com/miniprogram/dev/component/textarea.html*

导航

- navigator
- functional-page-navigator

navigator

navigator

页面链接。

属性名	类型	默认值	说明	最低版本
target	String		在哪个目标上发生跳转，默认当前小程序	2.0.7
url	String		当前小程序内的跳转链接	
open-type	String	navigate	跳转方式	
delta	Number		当 open-type 为 'navigateBack' 时有效，表示回退的层数	
app-id	String		当target="miniProgram"时有效，要打开的小程序 appId	2.0.7
path	String		当target="miniProgram"时有效，打开的页面路径，如果为空则打开首页	2.0.7
extra-data	Object		当target="miniProgram"时有效，需要传递给目标小程序的数据，目标小程序可在 `App.onLaunch()`，`App.onShow()` 中获取到这份数据。详情	2.0.7
version	version	release	当target="miniProgram"时有效，要打开的小程序版本，有效值 develop（开发版），trial（体验版），release（正式版），仅在当前小程序为开发版或体验版时此参数有效；如果当前小程序是正式版，则打开的小程序必定是正式版。	2.0.7
hover-class	String	navigator-hover	指定点击时的样式类，当 `hover-class="none"` 时，没有点击态效果	
hover-stop-propagation	Boolean	false	指定是否阻止本节点的祖先节点出现点击态	1.5.0
hover-start-time	Number	50	按住后多久出现点击态，单位毫秒	
hover-stay-time	Number	600	手指松开后点击态保留时间，单位毫秒	

open-type 有效值：

值	说明	最低版本
navigate	对应 `wx.navigateTo` 或 `wx.navigateToMiniProgram` 的功能	
redirect	对应 `wx.redirectTo` 的功能	
switchTab	对应 `wx.switchTab` 的功能	
reLaunch	对应 `wx.reLaunch` 的功能	1.1.0
navigateBack	对应 `wx.navigateBack` 的功能	1.1.0
exit	退出小程序，target="miniProgram"时生效	2.1.0

注： `navigator-hover` 默认为 `{background-color: rgba(0, 0, 0, 0.1); opacity: 0.7;}` ， `<navigator/>` 的子节点背景色应为透明色

示例代码：

在开发者工具中预览效果

```
/** wxss **/
/** 修改默认的navigator点击态 **/
.navigator-hover {
    color:blue;
}
/** 自定义其他点击态样式类 **/
.other-navigator-hover {
    color:red;
}
```

```
<!-- sample.wxml -->
<view class="btn-area">
  <navigator url="/page/navigate/navigate?title=navigate" hover-class="navigator-hover">跳转到新页面</navigator>
  <navigator url="../../redirect/redirect/redirect?title=redirect" open-type="redirect" hover-class="other-navigator-hover">在当前页打开</navigator>
  <navigator url="/page/index/index" open-type="switchTab" hover-class="other-navigator-hover">切换Tab</navigator>
  <navigator target="miniProgram" open-type="navigate" app-id="" path="" extra-data="" version="release">打开绑定的小程序</navigator>
</view>
```

```
<!-- navigator.wxml -->
<view style="text-align:center"> {{title}} </view>
<view> 点击左上角返回回到之前页面 </view>
```

```
<!-- redirect.wxml -->
<view style="text-align:center"> {{title}} </view>
<view> 点击左上角返回回到上级页面 </view>
```

```
// redirect.js navigator.js
Page({
  onLoad: function(options) {
    this.setData({
      title: options.title
    })
  }
})
```

原文：

> *https://developers.weixin.qq.com/miniprogram/dev/component/navigator.html*

functional-page-navigator

functional-page-navigator

这个组件从小程序基础库版本 2.1.0 开始支持。

仅在插件的自定义组件中有效，用于跳转到插件功能页。

属性名	类型	默认值	说明	最低版本
version	String	release	跳转到的小程序版本，有效值 develop（开发版），trial（体验版），release（正式版），仅在当前小程序为开发版或体验版时此参数有效；如果当前小程序是体验版或正式版，则打开的小程序必定是正式版	2.1.0
name	String		要跳转到的功能页	2.1.0
args	Object	null	功能页参数，参数格式与具体功能页相关	2.1.0
bindsuccess	EventHandle		功能页返回，且操作成功时触发， detail 格式与具体功能页相关	2.1.0
bindfail	EventHandle		功能页返回，且操作失败时触发， detail 格式与具体功能页相关	2.1.0

name 有效值：

值	说明	接受的 args	success 返回的 detail	fail 返 deta
loginAndGetUserInfo	获取用户信息，对应 wx.login 和 wx.getUserInfo	与 wx.getUserInfo 接受的 args 相同（除回调函数外）	wx.login 和 wx.getUserInfo 的结果的并集	与 wx.getUser 相同
requestPayment	支付，对应 wx.requestPayment	fee 字段，表示需要显示在页面中的金额，单位为分；paymentArgs 字段，任意数据，传递给功能页中的响应函数	与 wx.requestPayment 相同	与 wx.request 相同

示例代码：

```
<!-- sample.wxml -->
<functional-page-navigator name="loginAndGetUserInfo" bind:success="loginSuccess">
  <button>登录到插件</button>
</functional-page-navigator>
```

```
// redirect.js navigator.js
Component({
  methods: {
```

```
    loginSuccess: function(e) {
      console.log(e.detail.code) // wx.login 的 code
      console.log(e.detail.userInfo) // wx.getUserInfo 的 userInfo
    }
  }
})
```

Tips:

- 功能页是插件所有者小程序中的一个特殊页面，开发者不能自定义这个页面的外观。
- 在功能页展示时，一些与界面展示相关的接口将被禁用（接口调用返回 `fail` ）。
- 这个组件本身可以在开发者工具中使用，但功能页的跳转目前不支持在开发者工具中调试，请在真机上测试。

原文:

> *https://developers.weixin.qq.com/miniprogram/dev/component/functional-page-navigator.html*

媒体组件

- audio
- image
- video
- camera
- live-player
- live-pusher

audio

audio

注意：**1.6.0** 版本开始，该组件不再维护。建议使用能力更强的 **wx.createInnerAudioContext** 接口

音频。

属性名	类型	默认值	说明
id	String		audio 组件的唯一标识符
src	String		要播放音频的资源地址
loop	Boolean	false	是否循环播放
controls	Boolean	false	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}
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
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

MediaError.code

返回错误码	描述
1	获取资源被用户禁止
2	网络错误
3	解码错误
4	不合适资源

示例代码：

在开发者工具中预览效果

```
<!-- audio.wxml -->
<audio poster="{{poster}}" name="{{name}}" author="{{author}}" src="{{src}}" id="myAudio" controls loop>
</audio>

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

```
<button type="primary" bindtap="audioPause">暂停</button>
<button type="primary" bindtap="audio14">设置当前播放时间为14秒</button>
<button type="primary" bindtap="audioStart">回到开头</button>
```

```
// audio.js
Page({
  onReady: function (e) {
    // 使用 wx.createAudioContext 获取 audio 上下文 context
    this.audioCtx = wx.createAudioContext('myAudio')
  },
  data: {
    poster: 'http://y.gtimg.cn/music/photo_new/T002R300x300M000003rsKF44GyaSk.jpg?max_age=2592000',
    name: '此时此刻',
    author: '许巍',
    src: 'http://ws.stream.qqmusic.qq.com/M500001VfvsJ21xFqb.mp3?guid=ffffffff82def4af4b12b3cd9337d5e7&uin=346897220&vkey=6292F51E1E384E06DCBDC9AB7C49FD713D632D313AC4858BACB8DDD2
  },
  audioPlay: function () {
    this.audioCtx.play()
  },
  audioPause: function () {
    this.audioCtx.pause()
  },
  audio14: function () {
    this.audioCtx.seek(14)
  },
  audioStart: function () {
    this.audioCtx.seek(0)
  }
})
```

相关api: wx.createAudioContext

原文:

> https://developers.weixin.qq.com/miniprogram/dev/component/audio.html#audio

image

image

图片。

属性名	类型	默认值	说明	最低版本
src	String		图片资源地址	
mode	String	'scaleToFill'	图片裁剪、缩放的模式	
lazy-load	Boolean	false	图片懒加载。只针对page与scroll-view下的image有效	1.5.0
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 = {height:'图片高度px', width:'图片宽度px'}	

注：**image**组件默认宽度**300px**、高度**225px**

mode 有效值：

mode 有 13 种模式，其中 4 种是缩放模式，9 种是裁剪模式。

模式	值	说明
缩放	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
缩放	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。
缩放	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。
缩放	widthFix	宽度不变，高度自动变化，保持原图宽高比不变
裁剪	top	不缩放图片，只显示图片的顶部区域
裁剪	bottom	不缩放图片，只显示图片的底部区域
裁剪	center	不缩放图片，只显示图片的中间区域
裁剪	left	不缩放图片，只显示图片的左边区域
裁剪	right	不缩放图片，只显示图片的右边区域
裁	top left	不缩放图片，只显示图片的左上边区域

剪	top left	不缩放图片，只显示图片的左上边区域
裁剪	top right	不缩放图片，只显示图片的右上边区域
裁剪	bottom left	不缩放图片，只显示图片的左下边区域
裁剪	bottom right	不缩放图片，只显示图片的右下边区域

示例：

在开发者工具中预览效果

```
<view class="page">
  <view class="page__hd">
    <text class="page__title">image</text>
    <text class="page__desc">图片</text>
  </view>
  <view class="page__bd">
    <view class="section section_gap" wx:for="{{array}}" wx:for-item="item">
      <view class="section__title">{{item.text}}</view>
      <view class="section__ctn">
        <image style="width: 200px; height: 200px; background-color: #eeeeee;" mode="{{item.mode}}" src="{{src}}"></image>
      </view>
    </view>
  </view>
</view>
```

```
Page({
  data: {
    array: [{
      mode: 'scaleToFill',
      text: 'scaleToFill：不保持纵横比缩放图片，使图片完全适应'
    }, {
      mode: 'aspectFit',
      text: 'aspectFit：保持纵横比缩放图片，使图片的长边能完全显示出来'
    }, {
      mode: 'aspectFill',
      text: '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'
    }, {
      mode: 'top',
      text: 'top：不缩放图片，只显示图片的顶部区域'
    }, {
      mode: 'bottom',
      text: 'bottom：不缩放图片，只显示图片的底部区域'
    }, {
      mode: 'center',
      text: 'center：不缩放图片，只显示图片的中间区域'
    }, {
      mode: 'left',
      text: 'left：不缩放图片，只显示图片的左边区域'
```

```
    }, {
      mode: 'right',
      text: 'right：不缩放图片，只显示图片的右边边区域'
    }, {
      mode: 'top left',
      text: 'top left：不缩放图片，只显示图片的左上边区域'
    }, {
      mode: 'top right',
      text: 'top right：不缩放图片，只显示图片的右上边区域'
    }, {
      mode: 'bottom left',
      text: 'bottom left：不缩放图片，只显示图片的左下边区域'
    }, {
      mode: 'bottom right',
      text: 'bottom right：不缩放图片，只显示图片的右下边区域'
    }],
    src: '../../resources/cat.jpg'
  },
  imageError: function(e) {
    console.log('image3发生error事件，携带值为', e.detail.errMsg)
  }
})
```

原图

scaleToFill

不保持纵横比缩放图片，使图片完全适应

aspectFit

保持纵横比缩放图片，使图片的长边能完全显示出来

aspectFill

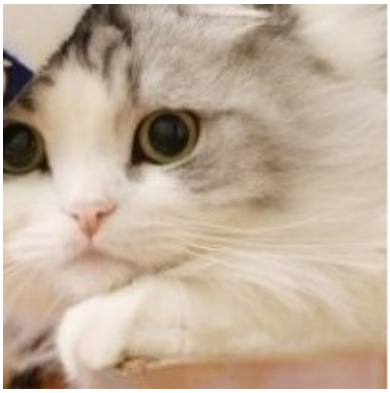
保持纵横比缩放图片，只保证图片的短边能完全显示出来

top

不缩放图片，只显示图片的顶部区域

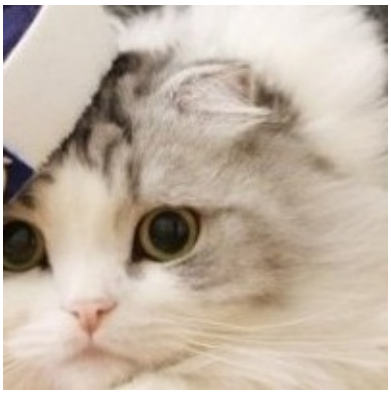
`bottom`

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

原文：

> *https://developers.weixin.qq.com/miniprogram/dev/component/image.html*

video

video

视频。

属性名	类型	默认值	说明	最低版本
src	String		要播放视频的资源地址	
initial-time	Number		指定视频初始播放位置	1.6.0
duration	Number		指定视频时长	1.1.0
controls	Boolean	true	是否显示默认播放控件（播放/暂停按钮、播放进度、时间）	
danmu-list	Object Array		弹幕列表	
danmu-btn	Boolean	false	是否显示弹幕按钮，只在初始化时有效，不能动态变更	
enable-danmu	Boolean	false	是否展示弹幕，只在初始化时有效，不能动态变更	
autoplay	Boolean	false	是否自动播放	
loop	Boolean	false	是否循环播放	1.4.0
muted	Boolean	false	是否静音播放	1.4.0
page-gesture	Boolean	false	在非全屏模式下，是否开启亮度与音量调节手势	1.6.0
direction	Number		设置全屏时视频的方向，不指定则根据宽高比自动判断。有效值为 0（正常竖向），90（屏幕逆时针90度），-90（屏幕顺时针90度）	1.7.0
show-progress	Boolean	true	若不设置，宽度大于240时才会显示	1.9.0
show-fullscreen-btn	Boolean	true	是否显示全屏按钮	1.9.0
show-play-btn	Boolean	true	是否显示视频底部控制栏的播放按钮	1.9.0
show-center-play-btn	Boolean	true	是否显示视频中间的播放按钮	1.9.0
enable-progress-gesture	Boolean	true	是否开启控制进度的手势	1.9.0
objectFit	String	contain	当视频大小与 video 容器大小不一致时，视频的表现形式。contain: 包含，fill: 填充，cover: 覆盖	
poster	String		视频封面的图片网络资源地址，如果 controls 属性值为 false 则设置 poster 无效	
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	
bindended	EventHandle		当播放到末尾时触发 ended 事件	

bindtimeupdate	EventHandle		播放进度变化时触发，event.detail = {currentTime, duration} 。触发频率 250ms 一次	
bindfullscreenchange	EventHandle		视频进入和退出全屏时触发，event.detail = {fullScreen, direction}，direction取为 vertical 或 horizontal	1.4.0
bindwaiting	EventHandle		视频出现缓冲时触发	1.7.0
binderror	EventHandle		视频播放出错时触发	1.7.0

`<video />` 默认宽度300px、高度225px，可通过wxss设置宽高。

示例代码：

在开发者工具中预览效果

```
<view class="section tc">
  <video src="{{src}}"   controls ></video>
  <view class="btn-area">
    <button bindtap="bindButtonTap">获取视频</button>
  </view>
</view>

<view class="section tc">
  <video id="myVideo" src="http://wxsnsdy.tc.qq.com/105/20210/snsdyvideodownload?filekey=30280201010421301f0201690402534804102ca905ce620b1241b726bc41dcff44e00204012882540400&bizid=1023&hy=SH&fil
   danmu-list="{{danmuList}}" enable-danmu danmu-btn controls></video>
  <view class="btn-area">
    <button bindtap="bindButtonTap">获取视频</button>
    <input bindblur="bindInputBlur"/>
    <button bindtap="bindSendDanmu">发送弹幕</button>
  </view>
</view>
```

```
function getRandomColor () {
  let rgb = []
  for (let i = 0 ; i < 3; ++i){
    let color = Math.floor(Math.random() * 256).toString(16)
    color = color.length == 1 ? '0' + color : color
    rgb.push(color)
  }
  return '#' + rgb.join('')
}

Page({
  onReady: function (res) {
    this.videoContext = wx.createVideoContext('myVideo')
  },
  inputValue: '',
    data: {
```

```
        src: '',
    danmuList: [
      {
        text: '第 1s 出现的弹幕',
        color: '#ff0000',
        time: 1
      },
      {
        text: '第 3s 出现的弹幕',
        color: '#ff00ff',
        time: 3
    }]
    },
  bindInputBlur: function(e) {
    this.inputValue = e.detail.value
  },
  bindButtonTap: function() {
    var that = this
    wx.chooseVideo({
      sourceType: ['album', 'camera'],
      maxDuration: 60,
      camera: ['front','back'],
      success: function(res) {
        that.setData({
          src: res.tempFilePath
        })
      }
    })
  },
  bindSendDanmu: function () {
    this.videoContext.sendDanmu({
      text: this.inputValue,
      color: getRandomColor()
    })
  }
})
```

video

相关api: wx.createVideoContext

Bug & Tip

- tip: video 组件是由客户端创建的原生组件，它的层级是最高的，不能通过 z-index 控制层级。
- tip: 请勿在 scroll-view、swiper、picker-view、movable-view 中使用 video 组件。
- tip: css 动画对 video 组件无效。

原文:

> *https://developers.weixin.qq.com/miniprogram/dev/component/video.html*

camera

camera

> 基础库 *1.6.0* 开始支持，低版本需做兼容处理

系统相机。

需要用户授权 scope.camera

属性名	类型	默认值	说明	最低版本
mode	String	normal	有效值为 normal, scanCode	2.1.0
device-position	String	back	前置或后置，值为front, back	
flash	String	auto	闪光灯，值为auto, on, off	
scan-area	Array		扫码识别区域，格式为[x, y, w, h]，x,y是相对于camera显示区域的左上角，w,h为区域宽度，单位px，仅在 mode="scanCode" 时生效	2.1.0
bindstop	EventHandle		摄像头在非正常终止时触发，如退出后台等情况	
binderror	EventHandle		用户不允许使用摄像头时触发	
bindscancode	EventHandle		在成功识别到一维码时触发，仅在 mode="scanCode" 时生效	2.1.0

相关api: wx.createCameraContext

Bug & Tip

- tip: camera 组件是由客户端创建的原生组件，它的层级是最高的，不能通过 z-index 控制层级。可使用 cover-view cover-image覆盖在上面。
- tip: 同一页面只能插入一个 camera 组件。
- tip: 请勿在 scroll-view、swiper、picker-view、movable-view 中使用 camera 组件。
- bug: scan-area 属性目前存在识别区域不准的问题，建议先不指定

示例:

在开发者工具中预览效果

```
<!-- camera.wxml -->
<camera device-position="back" flash="off" binderror="error" style="width: 100%; height: 300px;"></camera>
<button type="primary" bindtap="takePhoto">拍照</button>
<view>预览</view>
<image mode="widthFix" src="{{src}}"></image>
```

```
// camera.js
Page({
    takePhoto() {
```

```
        const ctx = wx.createCameraContext()
        ctx.takePhoto({
            quality: 'high',
            success: (res) => {
                this.setData({
                    src: res.tempImagePath
                })
            }
        })
    },
    error(e) {
        console.log(e.detail)
    }
})
```

原文：

> *https://developers.weixin.qq.com/miniprogram/dev/component/camera.html*

live-player

live-player

> 基础库 *1.7.0* 开始支持，低版本需做兼容处理

实时音视频播放。

暂只针对如下类目开放，需要先通过类目审核，再在小程序管理后台，“设置”-“接口设置”中自助开通该组件权限。

一级类目	二级类目
社交	直播
教育	在线教育
医疗	互联网医院，公立医院
政务民生	所有二级类目
金融	基金、信托、保险、银行、证券/期货、非金融机构自营小额贷款、征信业务、消费金融

属性名	类型	默认值	说明	最低版本
src	String		音视频地址。目前仅支持 flv, rtmp 格式	
mode	String	live	live（直播），RTC（实时通话）	
autoplay	Boolean	false	自动播放	
muted	Boolean	false	是否静音	
orientation	String	vertical	画面方向，可选值有 vertical, horizontal	
object-fit	String	contain	填充模式，可选值有 contain, fillCrop	
background-mute	Boolean	false	进入后台时是否静音（已废弃，默认退台静音）	
min-cache	Number	1	最小缓冲区，单位s	
max-cache	Number	3	最大缓冲区，单位s	
bindstatechange	EventHandle		播放状态变化事件，detail = {code}	
bindfullscreenchange	EventHandle		全屏变化事件，detail = {direction, fullScreen}	
bindnetstatus	EventHandle		网络状态通知，detail = {info}	1.9.0

注意：

- 默认宽度300px、高度225px，可通过wxss设置宽高。
- 开发者工具上暂不支持。
- 相关api：wx.createLivePlayerContext

状态码

代码	说明
2001	已经连接服务器
2002	已经连接服务器,开始拉流
2003	网络接收到首个视频数据包(IDR)
2004	视频播放开始
2005	视频播放进度
2006	视频播放结束
2007	视频播放Loading
2008	解码器启动
2009	视频分辨率改变
-2301	网络断连，且经多次重连抢救无效，更多重试请自行重启播放
-2302	获取加速拉流地址失败
2101	当前视频帧解码失败
2102	当前音频帧解码失败
2103	网络断连, 已启动自动重连
2104	网络来包不稳：可能是下行带宽不足，或由于主播端出流不均匀
2105	当前视频播放出现卡顿
2106	硬解启动失败，采用软解
2107	当前视频帧不连续，可能丢帧
2108	当前流硬解第一个I帧失败，SDK自动切软解
3001	RTMP -DNS解析失败
3002	RTMP服务器连接失败
3003	RTMP服务器握手失败
3005	RTMP 读/写失败

网络状态数据

键名	说明
videoBitrate	当前视频编/码器输出的比特率，单位 kbps
audioBitrate	当前音频编/码器输出的比特率，单位 kbps
videoFPS	当前视频帧率
videoGOP	当前视频 GOP,也就是每两个关键帧(I帧)间隔时长，单位 s
netSpeed	当前的发送/接收速度
netJitter	网络抖动情况，抖动越大，网络越不稳定
videoWidth	视频画面的宽度
videoHeight	视频画面的高度

示例代码：

在开发者工具中预览效果

```
<live-player src="https://domain/pull_stream" mode="RTC" autoplay bindstatechange="statechange"
binderror="error" style="width: 300px; height: 225px;" />
```

```
Page({
  statechange(e) {
    console.log('live-player code:', e.detail.code)
  },
  error(e) {
      console.error('live-player error:', e.detail.errMsg)
  }
})
```

Bug & Tip

- tip: live-player 组件是由客户端创建的原生组件，它的层级是最高的，不能通过 z-index 控制层级。可使用 cover-view cover-image覆盖在上面。
- tip: 请勿在 scroll-view、swiper、picker-view、movable-view 中使用 live-player 组件。
- tip: css 动画对 live-player 组件无效。

原文：

> *https://developers.weixin.qq.com/miniprogram/dev/component/live-player.html*

live-pusher

live-pusher

> 基础库 1.7.0 开始支持，低版本需做兼容处理

实时音视频录制。

需要用户授权 scope.camera、scope.record

暂只针对如下类目开放，需要先通过类目审核，再在小程序管理后台，“设置”-“接口设置”中自助开通该组件权限。

一级类目	二级类目
社交	直播
教育	在线教育
医疗	互联网医院，公立医院
政务民生	所有二级类目
金融	基金、信托、保险、银行、证券/期货、非金融机构自营小额贷款、征信业务、消费金融

属性名	类型	默认值	说明	最低版本
url	String		推流地址。目前仅支持 flv, rtmp 格式	
mode	String	RTC	SD（标清）, HD（高清）, FHD（超清）, RTC（实时通话）	
autopush	Boolean	false	自动推流	
muted	Boolean	false	是否静音	
enable-camera	Boolean	true	开启摄像头	
auto-focus	Boolean	true	自动聚集	
orientation	String	vertical	vertical, horizontal	
beauty	Number	0	美颜	
whiteness	Number	0	美白	
aspect	String	9:16	宽高比，可选值有 3:4, 9:16	
min-bitrate	Number	200	最小码率	
max-bitrate	Number	1000	最大码率	
waiting-image	String		进入后台时推流的等待画面	
waiting-image-hash	String		等待画面资源的MD5值	
zoom	Boolean	false	调整焦距	2.1.0
background-mute	Boolean	false	进入后台时是否静音	
bindstatechange	EventHandle		状态变化事件，detail = {code}	
bindnetstatus	EventHandle		网络状态通知，detail = {info}	1.9.0

binderror	EventHandle		渲染错误事件，detail = {errMsg, errCode}	1.7.4

注意：

- 默认宽度为100%、无默认高度，请通过wxss设置宽高。
- 开发者工具上暂不支持。
- 相关api：wx.createLivePusherContext

错误码（errCode）

代码	说明
10001	用户禁止使用摄像头
10002	用户禁止使用录音

状态码（code）

代码	说明
1001	已经连接推流服务器
1002	已经与服务器握手完毕,开始推流
1003	打开摄像头成功
1004	录屏启动成功
1005	推流动态调整分辨率
1006	推流动态调整码率
1007	首帧画面采集完成
1008	编码器启动
-1301	打开摄像头失败
-1302	打开麦克风失败
-1303	视频编码失败
-1304	音频编码失败
-1305	不支持的视频分辨率
-1306	不支持的音频采样率
-1307	网络断连，且经多次重连抢救无效，更多重试请自行重启推流
-1308	开始录屏失败，可能是被用户拒绝
-1309	录屏失败，不支持的Android系统版本，需要5.0以上的系统
-1310	录屏被其他应用打断了
-1311	Android Mic打开成功，但是录不到音频数据
-1312	录屏动态切横竖屏失败
1101	网络状况不佳：上行带宽太小，上传数据受阻
1102	网络断连，已启动自动重连
1103	硬编码启动失败,采用软编码

1104	视频编码失败
1105	新美颜软编码启动失败，采用老的软编码
1106	新美颜软编码启动失败，采用老的软编码
3001	RTMP -DNS解析失败
3002	RTMP服务器连接失败
3003	RTMP服务器握手失败
3004	RTMP服务器主动断开，请检查推流地址的合法性或防盗链有效期
3005	RTMP 读/写失败

网络状态数据（info）

键名	说明
videoBitrate	当前视频编/码器输出的比特率，单位 kbps
audioBitrate	当前音频编/码器输出的比特率，单位 kbps
videoFPS	当前视频帧率
videoGOP	当前视频 GOP,也就是每两个关键帧(I帧)间隔时长，单位 s
netSpeed	当前的发送/接收速度
netJitter	网络抖动情况，抖动越大，网络越不稳定
videoWidth	视频画面的宽度
videoHeight	视频画面的高度

示例代码：

在开发者工具中预览效果

```
<live-pusher url="https://domain/push_stream" mode="RTC" autopush bindstatechange="statechange" style="width: 300px; height: 225px;" />
```

```
Page({
  statechange(e) {
    console.log('live-pusher code:', e.detail.code)
  }
})
```

Bug & Tip

- tip: live-pusher 组件是由客户端创建的原生组件，它的层级是最高的，不能通过 z-index 控制层级。可使用 cover-view cover-image覆盖在上面。
- tip: 请勿在 scroll-view、swiper、picker-view、movable-view 中使用 live-pusher 组件。
- tip: css 动画对 live-pusher 组件无效。<

原文：

> *https://developers.weixin.qq.com/miniprogram/dev/component/live-pusher.html*

地图

- map

map

map

地图。

属性名	类型	默认值	说明	最低版本
longitude	Number		中心经度	
latitude	Number		中心纬度	
scale	Number	16	缩放级别，取值范围为5-18	
markers	Array		标记点	
covers	Array		即将移除，请使用 **markers**	
polyline	Array		路线	
circles	Array		圆	
controls	Array		控件	
include-points	Array		缩放视野以包含所有给定的坐标点	
show-location	Boolean		显示带有方向的当前定位点	
bindmarkertap	EventHandle		点击标记点时触发，会返回marker的id	
bindcallouttap	EventHandle		点击标记点对应的气泡时触发，会返回marker的id	1.2.0
bindcontroltap	EventHandle		点击控件时触发，会返回control的id	
bindregionchange	EventHandle		视野发生变化时触发	
bindtap	EventHandle		点击地图时触发	
bindupdated	EventHandle		在地图渲染更新完成时触发	1.6.0

注意：**covers** 属性即将移除，请使用 **markers** 替代

markers

标记点用于在地图上显示标记的位置

属性	说明	类型	必填	备注	最低版本
id	标记点id	Number	否	marker点击事件回调会返回此id。建议为每个**marker**设置上**Number**类型**id**，保证更新**marker**时有更好的性能。	
latitude	纬度	Number	是	浮点数，范围 -90 ~ 90	
longitude	经度	Number	是	浮点数，范围 -180 ~ 180	
title	标注点名	String	否		
iconPath	显示的图标	String	是	项目目录下的图片路径，支持相对路径写法，以'/'开头则表示相对小程序根目录；	

				也支持临时路径	
rotate	旋转角度	Number	否	顺时针旋转的角度，范围 0 ~ 360，默认为 0	
alpha	标注的透明度	Number	否	默认1，无透明，范围 0 ~ 1	
width	标注图标宽度	Number	否	默认为图片实际宽度	
height	标注图标高度	Number	否	默认为图片实际高度	
callout	自定义标记点上方的气泡窗口	Object	否	支持的属性见下表，可识别换行符。	1.2.0
label	为标记点旁边增加标签	Object	否	支持的属性见下表，可识别换行符。	1.2.0
anchor	经纬度在标注图标的锚点，默认底边中点	Object	否	{x, y}，x表示横向(0-1)，y表示竖向(0-1)。{x: .5, y: 1} 表示底边中点	1.2.0

marker 上的气泡 callout

属性	说明	类型	最低版本
content	文本	String	1.2.0
color	文本颜色	String	1.2.0
fontSize	文字大小	Number	1.2.0
borderRadius	callout边框圆角	Number	1.2.0
bgColor	背景色	String	1.2.0
padding	文本边缘留白	Number	1.2.0
display	'BYCLICK':点击显示; 'ALWAYS':常显	String	1.2.0
textAlign	文本对齐方式。有效值: left, right, center	String	1.6.0

marker 上的气泡 label

属性	说明	类型	最低版本
content	文本	String	1.2.0
color	文本颜色	String	1.2.0
fontSize	文字大小	Number	1.2.0
x	label的坐标（废弃）	Number	1.2.0
y	label的坐标（废弃）	Number	1.2.0
anchorX	label的坐标，原点是 marker 对应的经纬度	Number	2.1.0
anchorY	label的坐标，原点是 marker 对应的经纬度	Number	2.1.0
borderWidth	边框宽度	Number	1.6.0
borderColor	边框颜色	String	1.6.0
borderRadius	边框圆角	Number	1.6.0
bgColor	背景色	String	1.6.0
padding	文本边缘留白	Number	1.6.0
textAlign	文本对齐方式。有效值: left, right, center	String	1.6.0

polyline

指定一系列坐标点，从数组第一项连线至最后一项

属性	说明	类型	必填	备注	最低版本
points	经纬度数组	Array	是	[{latitude: 0, longitude: 0}]	
color	线的颜色	String	否	8位十六进制表示，后两位表示alpha值，如：#000000AA	
width	线的宽度	Number	否		
dottedLine	是否虚线	Boolean	否	默认false	
arrowLine	带箭头的线	Boolean	否	默认false，开发者工具暂不支持该属性	1.2.0
arrowIconPath	更换箭头图标	String	否	在arrowLine为true时生效	1.6.0
borderColor	线的边框颜色	String	否		1.2.0
borderWidth	线的厚度	Number	否		1.2.0

circles

在地图上显示圆

属性	说明	类型	必填	备注
latitude	纬度	Number	是	浮点数，范围 -90 ~ 90
longitude	经度	Number	是	浮点数，范围 -180 ~ 180
color	描边的颜色	String	否	8位十六进制表示，后两位表示alpha值，如：#000000AA
fillColor	填充颜色	String	否	8位十六进制表示，后两位表示alpha值，如：#000000AA
radius	半径	Number	是	
strokeWidth	描边的宽度	Number	否	

controls

在地图上显示控件，控件不随着地图移动。即将废弃，请使用 **cover-view**

属性	说明	类型	必填	备注
id	控件id	Number	否	在控件点击事件回调会返回此id
position	控件在地图的位置	Object	是	控件相对地图位置
iconPath	显示的图标	String	是	项目目录下的图片路径，支持相对路径写法，以'/'开头则表示相对小程序根目录；也支持临时路径

clickable	是否可点击	Boolean	否	默认不可点击

position

属性	说明	类型	必填	备注
left	距离地图的左边界多远	Number	否	默认为0
top	距离地图的上边界多远	Number	否	默认为0
width	控件宽度	Number	否	默认为图片宽度
height	控件高度	Number	否	默认为图片高度

地图组件的经纬度必填，如果不填经纬度则默认值是北京的经纬度。

示例：

在开发者工具中预览效果

```
<!-- map.wxml -->
<map id="map" longitude="113.324520" latitude="23.099994" scale="14" controls="{{controls}}" bindcontroltap="controltap" markers="{{markers}}" bindmarkertap="markertap" polyline="{{polyline}}" bindregionchange="regionchange" show-location style="width: 100%; height: 300px;"></map>
```

```
// map.js
Page({
  data: {
    markers: [{
      iconPath: "/resources/others.png",
      id: 0,
      latitude: 23.099994,
      longitude: 113.324520,
      width: 50,
      height: 50
    }],
    polyline: [{
      points: [{
        longitude: 113.3245211,
        latitude: 23.10229
      }, {
        longitude: 113.324520,
        latitude: 23.21229
      }],
      color:"#FF0000DD",
      width: 2,
      dottedLine: true
    }],
    controls: [{
      id: 1,
      iconPath: '/resources/location.png',
      position: {
        left: 0,
        top: 300 - 50,
```

```
        width: 50,
        height: 50
      },
      clickable: true
    }]
  },
  regionchange(e) {
    console.log(e.type)
  },
  markertap(e) {
    console.log(e.markerId)
  },
  controltap(e) {
    console.log(e.controlId)
  }
})
```

相关api: wx.createMapContext

Bug & Tip

- tip: map 组件是由客户端创建的原生组件，它的层级是最高的，不能通过 z-index 控制层级。
- tip: 请勿在 scroll-view、swiper、picker-view、movable-view 中使用 map 组件。
- tip: css 动画对 map 组件无效。
- tip: map 组件使用的经纬度是火星坐标系，调用 wx.getLocation 接口需要指定 type 为 gcj02

原文:

> *https://developers.weixin.qq.com/miniprogram/dev/component/map.html#map*

画布

- canvas

canvas

canvas

画布。

属性名	类型	默认值	说明
canvas-id	String		canvas 组件的唯一标识符
disable-scroll	Boolean	false	当在 canvas 中移动时且有绑定手势事件时，禁止屏幕滚动以及下拉刷新
bindtouchstart	EventHandle		手指触摸动作开始
bindtouchmove	EventHandle		手指触摸后移动
bindtouchend	EventHandle		手指触摸动作结束
bindtouchcancel	EventHandle		手指触摸动作被打断，如来电提醒，弹窗
bindlongtap	EventHandle		手指长按 500ms 之后触发，触发了长按事件后进行移动不会触发屏幕的滚动
binderror	EventHandle		当发生错误时触发 error 事件，detail = {errMsg: 'something wrong'}

注：

- **canvas** 标签默认宽度**300px**、高度**225px**
- 同一页面中的 **canvas-id** 不可重复，如果使用一个已经出现过的 **canvas-id**，该 **canvas** 标签对应的画布将被隐藏并不再正常工作

示例代码：下载
在开发者工具中预览效果

```
<!-- canvas.wxml -->
<canvas style="width: 300px; height: 200px;" canvas-id="firstCanvas"></canvas>
<!-- 当使用绝对定位时，文档流后边的 canvas 的显示层级高于前边的 canvas -->
<canvas style="width: 400px; height: 500px;" canvas-id="secondCanvas"></canvas>
<!-- 因为 canvas-id 与前一个 canvas 重复，该 canvas 不会显示，并会发送一个错误事件到 AppService -->
<canvas style="width: 400px; height: 500px;" canvas-id="secondCanvas" binderror="canvasIdErrorCallback">
</canvas>
```

```
// canvas.js
Page({
  canvasIdErrorCallback: function (e) {
    console.error(e.detail.errMsg)
  },
  onReady: function (e) {
    // 使用 wx.createContext 获取绘图上下文 context
    var context = wx.createCanvasContext('firstCanvas')

```

```
    context.setStrokeStyle("#00ff00")
    context.setLineWidth(5)
    context.rect(0, 0, 200, 200)
    context.stroke()
    context.setStrokeStyle("#ff0000")
    context.setLineWidth(2)
    context.moveTo(160, 100)
    context.arc(100, 100, 60, 0, 2 * Math.PI, true)
    context.moveTo(140, 100)
    context.arc(100, 100, 40, 0, Math.PI, false)
    context.moveTo(85, 80)
    context.arc(80, 80, 5, 0, 2 * Math.PI, true)
    context.moveTo(125, 80)
    context.arc(120, 80, 5, 0, 2 * Math.PI, true)
    context.stroke()
    context.draw()
  }
})
```

相关api: wx.createCanvasContext

Bug & Tip

- tip: canvas 组件是由客户端创建的原生组件，它的层级是最高的，不能通过 z-index 控制层级。
- tip: 请勿在 scroll-view、swiper、picker-view、movable-view 中使用 canvas 组件。
- tip: css 动画对 canvas 组件无效。
- bug: 避免设置过大的宽高，在安卓下会有crash的问题

原文:

> *https://developers.weixin.qq.com/miniprogram/dev/component/canvas.html#canvas*

开放能力

- open-data
- web-view
- ad

open-data

open-data

> 基础库 *1.4.0* 开始支持，低版本需做兼容处理

用于展示微信开放的数据。

属性名	类型	默认值	说明
type	String		开放数据类型
open-gid	String		当 type="groupName" 时生效，群id
lang	String	en	当 type="user*" 时生效，以哪种语言展示 userInfo，有效值有：en, zh_CN, zh_TW

type 有效值：

值	说明	最低版本
groupName	拉取群名称	1.4.0
userNickName	用户昵称	1.9.90
userAvatarUrl	用户头像	1.9.90
userGender	用户性别	1.9.90
userCity	用户所在城市	1.9.90
userProvince	用户所在省份	1.9.90
userCountry	用户所在国家	1.9.90
userLanguage	用户的语言	1.9.90

Tips： 只有当前用户在此群内才能拉取到群名称

示例：

在开发者工具中预览效果

```
<open-data type="groupName" open-gid="xxxxxx"></open-data>
<open-data type="userAvatarUrl"></open-data>
<open-data type="userGender" lang="zh_CN"></open-data>
```

Tips:关于open-gid的获取请查看 转发

原文：

> *https://developers.weixin.qq.com/miniprogram/dev/component/open-data.html*

web-view

web-view

> 基础库 *1.6.4* 开始支持，低版本需做兼容处理

web-view 组件是一个可以用来承载网页的容器，会自动铺满整个小程序页面。个人类型与海外类型的小程序暂不支持使用。

属性名	类型	默认值	说明
src	String		webview 指向网页的链接。可打开关联的公众号的文章，其它网页需登录小程序管理后台配置业务域名。
bindmessage	EventHandler		网页向小程序 postMessage 时，会在特定时机（小程序后退、组件销毁、分享）触发并收到消息。e.detail = { data }

示例代码：

```
<!-- wxml -->
<!-- 指向微信公众平台首页的web-view -->
<web-view src="https://mp.weixin.qq.com/"></web-view>
```

相关接口 1

`<web-view/>` 网页中可使用JSSDK 1.3.2提供的接口返回小程序页面。支持的接口有：

接口名	说明	最低版本
wx.miniProgram.navigateTo	参数与小程序接口一致	1.6.4
wx.miniProgram.navigateBack	参数与小程序接口一致	1.6.4
wx.miniProgram.switchTab	参数与小程序接口一致	1.6.5
wx.miniProgram.reLaunch	参数与小程序接口一致	1.6.5
wx.miniProgram.redirectTo	参数与小程序接口一致	1.6.5
wx.miniProgram.postMessage	向小程序发送消息	1.7.1
wx.miniProgram.getEnv	获取当前环境	1.7.1

示例代码：

在开发者工具中预览效果

```
<!-- html -->
<script type="text/javascript" src="https://res.wx.qq.com/open/js/jweixin-1.3.2.js"></script>

// javascript
wx.miniProgram.navigateTo({url: '/path/to/page'})
wx.miniProgram.postMessage({ data: 'foo' })
```

```
7. wx.miniProgram.postMessage({ data: {foo: 'bar'} })
8. wx.miniProgram.getEnv(function(res) { console.log(res.miniprogram) // true })
```

相关接口 2

`<web-view/>` 网页中仅支持以下**JSSDK**接口：

接口模块	接口说明	具体接口
判断客户端是否支持js		checkJSApi
图像接口	拍照或上传	chooseImage
	预览图片	previewImage
	上传图片	uploadImage
	下载图片	downloadImage
	获取本地图片	getLocalImgData
音频接口	开始录音	startRecord
	停止录音	stopRecord
	监听录音自动停止	onVoiceRecordEnd
	播放语音	playVoice
	暂停播放	pauseVoice
	停止播放	stopVoice
	监听语音播放完毕	onVoicePlayEnd
	上传接口	uploadVoice
	下载接口	downloadVoice
智能接口	识别音频	translateVoice
设备信息	获取网络状态	getNetworkType
地理位置	使用内置地图	getLocation
	获取地理位置	openLocation
摇一摇周边	开启ibeacon	startSearchBeacons
	关闭ibeacon	stopSearchBeacons
	监听ibeacon	onSearchBeacons
微信扫一扫	调起微信扫一扫	scanQRCode
微信卡券	拉取使用卡券列表	chooseCard
	批量添加卡券接口	addCard
	查看微信卡包的卡券	openCard
长按识别	小程序圆形码	无

相关接口 3

用户分享时可获取当前 `<web-view/>` 的URL，即在 `onShareAppMessage` 回调中返回 `webViewUrl` 参数。

示例代码：

```
Page({
  onShareAppMessage(options) {
    console.log(options.webViewUrl)
  }
})
```

相关接口 4

在网页内可通过 `window.__wxjs_environment` 变量判断是否在小程序环境，建议在 `WeixinJSBridgeReady` 回调中使用，也可以使用JSSDK 1.3.2提供的 `getEnv` 接口。

示例代码：

```
// web-view下的页面内
function ready() {
  console.log(window.__wxjs_environment === 'miniprogram') // true
}
if (!window.WeixinJSBridge || !WeixinJSBridge.invoke) {
  document.addEventListener('WeixinJSBridgeReady', ready, false)
} else {
  ready()
}

// 或者
wx.miniProgram.getEnv(function(res) {
  console.log(res.miniprogram) // true
})
```

Bug & Tip

- 网页内**iframe**的域名也需要配置到域名白名单。
- 开发者工具上，可以在 `<web-view/>` 组件上通过右键 - 调试，打开 `<web-view/>` 组件的调试。
- 每个页面只能有一个`<web-view/>`，`<web-view/>`会自动铺满整个页面，并覆盖其他组件。
- `<web-view/>`网页与小程序之间不支持除JSSDK提供的接口之外的通信。
- 在iOS中，若存在JSSDK接口调用无响应的情况，可在`<web-view>`的src后面加个`#wechat_redirect`解决。

原文：

> *https://developers.weixin.qq.com/miniprogram/dev/component/web-view.html*

ad

ad

> 基础库 *1.9.94* 开始支持，低版本需做兼容处理

广告。目前暂时以邀请制开放申请，请留意后续模板消息的通知

属性名	类型	默认值	说明
unit-id	String		广告单元id，可在小程序管理后台的流量主模块新建

注意

- 目前可以给 ad 标签设置 wxss 样式调整广告宽度，以使广告与页面更融洽，但请遵循小程序流量主应用规范
- 在无广告展示时，ad 标签不会占用高度
- ad 组件不支持触发 bindtap 等触摸相关事件

原文：

> *https://developers.weixin.qq.com/miniprogram/dev/component/ad.html*