# 试图容器

#### [view](https://uniapp.dcloud.io/component/view?id=view)

视图容器。

**属性说明**

|  | **类型** | **默认值** | **说明** |
| --- | --- | --- | --- |
| 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 |
| 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** [查看演示](https://uniapp.dcloud.io/h5/pages/component/view/view)

<template>

<view>

<view class="uni-padding-wrap uni-common-mt">

<view class="uni-title uni-common-mt">

flex-direction: row

<text>\n横向布局</text>

</view>

<view class="uni-flex uni-row">

<view class="flex-item uni-bg-red">A</view>

<view class="flex-item uni-bg-green">B</view>

<view class="flex-item uni-bg-blue">C</view>

</view>

<view class="uni-title uni-common-mt">

flex-direction: column

<text>\n纵向布局</text>

</view>

<view class="uni-flex uni-column">

<view class="flex-item flex-item-V uni-bg-red">A</view>

<view class="flex-item flex-item-V uni-bg-green">B</view>

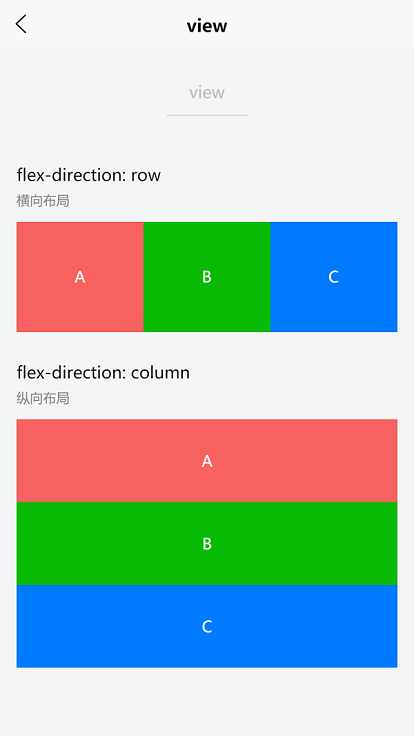
<view class="flex-item flex-item-V uni-bg-blue">C</view>

</view>

</view>

</view>

</template>



**Tips**

* 如果需要使用滚动视图，请使用 [scroll-view](https://uniapp.dcloud.io/component/scroll-view)
* 如果使用 <div> 组件编译时会被转换为 <view>。

#### [scroll-view](https://uniapp.dcloud.io/component/scroll-view?id=scroll-view)

可滚动视图区域。

**属性说明**

|  | **类型** | **默认值** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| 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 | 微信小程序 |
| show-scrollbar | Boolean | false | 控制是否出现滚动条 | App-nvue 2.1.5+ |
| refresher-enabled | Boolean | false | 开启自定义下拉刷新 | app-vue 2.5.12+,微信小程序基础库2.10.1+ |
| refresher-threshold | number | 45 | 设置自定义下拉刷新阈值 | app-vue 2.5.12+,微信小程序基础库2.10.1+ |
| refresher-default-style | string | "black" | 设置自定义下拉刷新默认样式，支持设置 black，white，none，none 表示不使用默认样式 | app-vue 2.5.12+,微信小程序基础库2.10.1+ |
| refresher-background | string | "#FFF" | 设置自定义下拉刷新区域背景颜色 | app-vue 2.5.12+,微信小程序基础库2.10.1+ |
| refresher-triggered | boolean | false | 设置当前下拉刷新状态，true 表示下拉刷新已经被触发，false 表示下拉刷新未被触发 | app-vue 2.5.12+,微信小程序基础库2.10.1+ |
| @scrolltoupper | EventHandle |  | 滚动到顶部/左边，会触发 scrolltoupper 事件 |  |
| @scrolltolower | EventHandle |  | 滚动到底部/右边，会触发 scrolltolower 事件 |  |
| @scroll | EventHandle |  | 滚动时触发，event.detail = {scrollLeft, scrollTop, scrollHeight, scrollWidth, deltaX, deltaY} |  |
| @refresherpulling | EventHandle |  | 自定义下拉刷新控件被下拉 | app-vue 2.5.12+,微信小程序基础库2.10.1+ |
| @refresherrefresh | EventHandle |  | 自定义下拉刷新被触发 | app-vue 2.5.12+,微信小程序基础库2.10.1+ |
| @refresherrestore | EventHandle |  | 自定义下拉刷新被复位 | app-vue 2.5.12+,微信小程序基础库2.10.1+ |
| @refresherabort | EventHandle |  | 自定义下拉刷新被中止 | app-vue 2.5.12+,微信小程序基础库2.10.1+ |

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

**示例** [查看演示](https://uniapp.dcloud.io/h5/pages/component/scroll-view/scroll-view)

<template>

<view>

<view class="uni-padding-wrap uni-common-mt">

<view class="uni-title uni-common-mt">

Vertical Scroll

<text>\n纵向滚动</text>

</view>

<view>

<scroll-view :scroll-top="scrollTop" scroll-y="true" class="scroll-Y" @scrolltoupper="upper" @scrolltolower="lower"

@scroll="scroll">

<view id="demo1" class="scroll-view-item uni-bg-red">A</view>

<view id="demo2" class="scroll-view-item uni-bg-green">B</view>

<view id="demo3" class="scroll-view-item uni-bg-blue">C</view>

</scroll-view>

</view>

<view @tap="goTop" class="uni-link uni-center uni-common-mt">

点击这里返回顶部

</view>

<view class="uni-title uni-common-mt">

Horizontal Scroll

<text>\n横向滚动</text>

</view>

<view>

<scroll-view class="scroll-view\_H" scroll-x="true" @scroll="scroll" scroll-left="120">

<view id="demo1" class="scroll-view-item\_H uni-bg-red">A</view>

<view id="demo2" class="scroll-view-item\_H uni-bg-green">B</view>

<view id="demo3" class="scroll-view-item\_H uni-bg-blue">C</view>

</scroll-view>

</view>

</view>

</view>

</template>

export default {

data() {

return {

scrollTop: 0,

old: {

scrollTop: 0

}

}

},

methods: {

upper: function(e) {

console.log(e)

},

lower: function(e) {

console.log(e)

},

scroll: function(e) {

console.log(e)

this.old.scrollTop = e.detail.scrollTop

},

goTop: function(e) {

this.scrollTop = this.old.scrollTop

this.$nextTick(function() {

this.scrollTop = 0

});

uni.showToast({

icon:"none",

title:"纵向滚动 scrollTop 值已被修改为 0"

})

}

}

}

**自定义下拉刷新**

<template>

<view>

<scroll-view style="height: 300px;" scroll-y="true" refresher-enabled="true" :refresher-triggered="triggered"

:refresher-threshold="100" refresher-background="lightgreen" @refresherpulling="onPulling"

@refresherrefresh="onRefresh" @refresherrestore="onRestore" @refresherabort="onAbort"></scroll-view>

</view>

</template>

<script>

export default {

data() {

return {

triggered: false

}

},

onLoad() {

this.\_freshing = false;

setTimeout(() => {

this.triggered = true;

}, 1000)

},

methods: {

onPulling(e) {

console.log("onpulling", e);

},

onRefresh() {

if (this.\_freshing) return;

this.\_freshing = true;

setTimeout(() => {

this.triggered = false;

this.\_freshing = false;

}, 3000)

},

onRestore() {

this.triggered = 'restore'; // 需要重置

console.log("onRestore");

},

onAbort() {

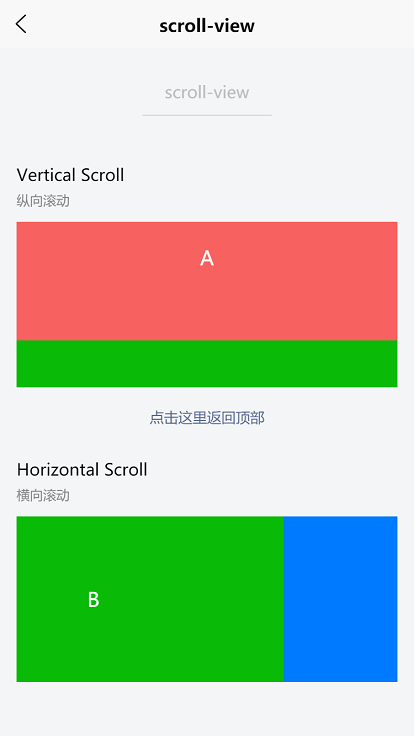
console.log("onAbort");

}

}

}

</script>



**Tips**

* APP-vue和小程序中，请勿在 scroll-view 中使用 map、video 等原生组件。小程序中 scroll-view 中也不要使用 canvas、textarea 原生组件。更新：微信基础库2.4.4起支持了原生组件在 scroll-view、swiper、movable-view 中的使用。app-nvue无此限制。
* scroll-view 不适合放长列表，有性能问题。长列表滚动和下拉刷新，应该使用原生导航栏搭配页面级的滚动和下拉刷新实现。包括在app-nvue页面，长列表应该使用list而不是scroll-view
* scroll-into-view 的优先级高于 scroll-top。
* scroll-view是区域滚动，不会触发页面滚动，无法触发pages.json配置的下拉刷新、页面触底onReachBottomDistance、titleNView的transparent透明渐变。
* 若要使用下拉刷新，建议使用页面的滚动，而不是 scroll-view 。插件市场有前端模拟的下拉刷新，但性能不佳。
* 如果遇到scroll-top、scroll-left属性设置不生效的问题参考：[组件属性设置不生效解决办法](https://uniapp.dcloud.io/use?id=%e5%b8%b8%e8%a7%81%e9%97%ae%e9%a2%98)
* scroll-view的滚动条设置，可通过css的-webkit-scrollbar自定义，包括隐藏滚动条。（nvue无此css）

#### [swiper](https://uniapp.dcloud.io/component/swiper?id=swiper)

滑块视图容器。一般用于左右滑动或上下滑动，比如banner轮播图，注意是滑动切换而不是滚动。swiper下的每个swiper-item是一个滑动切换区域，不能停留在2个滑动区域之间。

**属性说明**

| **属性名** | **类型** | **默认值** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| 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 |  |
| indicator-active-color | Color | #000000 | 当前选中的指示点颜色 |  |
| active-class | String |  | swiper-item 可见时的 class | 支付宝小程序 |
| changing-class | String |  | acceleration 设置为 {{true}} 时且处于滑动过程中，中间若干屏处于可见时的class | 支付宝小程序 |
| 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 | 支付宝小程序不支持 |
| interval | Number | 5000 | 自动切换时间间隔 |  |
| duration | Number | 500 | 滑动动画时长 | app-nvue不支持 |
| 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 | app-nvue、头条小程序不支持 |
| 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 | app-nvue、头条小程序不支持 |
| acceleration | Boolean | false | 当开启时，会根据滑动速度，连续滑动多屏 | 支付宝小程序 |
| disable-programmatic-animation | Boolean | false | 是否禁用代码变动触发 swiper 切换时使用动画。 | 支付宝小程序 |
| display-multiple-items | Number | 1 | 同时显示的滑块数量 | 支付宝小程序不支持 |
| 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 | App、微信小程序 |
| disable-touch | Boolean | false | 是否禁止用户 touch 操作 | App 2.5.5+、H5 2.5.5+、支付宝小程序、头条小程序（只在初始化时有效，不能动态变更） |
| touchable | Boolean | true | 是否监听用户的触摸事件，只在初始化时有效，不能动态变更 | 头条小程序（uni-app 2.5.5+ 推荐统一使用 disable-touch） |
| easing-function | String | default | 指定 swiper 切换缓动动画类型，有效值：default、linear、easeInCubic、easeOutCubic、easeInOutCubic | 微信小程序 |
| @change | EventHandle |  | current 改变时会触发 change 事件，event.detail = {current: current, source: source} |  |
| @transition | EventHandle |  | swiper-item 的位置发生改变时会触发 transition 事件，event.detail = {dx: dx, dy: dy} | App、H5、微信小程序、支付宝小程序、头条小程序、QQ小程序 |
| @animationfinish | EventHandle |  | 动画结束时会触发 animationfinish 事件，event.detail = {current: current, source: source} | 头条小程序不支持 |

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。

**swiper做左右拖动的长列表问题**

* swiper是单页组件，适合做banner图轮播和简单列表左右滑动。
* 因为性能问题，用swiper做复杂长列表，需要较高的优化技巧以及接受一些限制。
* 这是一个范例，[插件市场新闻模板示例](https://ext.dcloud.net.cn/plugin?id=103)，它在App端使用了nvue的原生渲染，实现高性能的左右拖动长列表；并支持可自定义的任何形式的下拉刷新。它在非App端使用的模式是只缓存左右一共3列的数据，dom中的数据过多时，它会自动释放。就是说App上，只要看过这一页，再进去时内容是还在的。而在非App上，只能做到缓存3页数据，其他页即便看过，再进去也会重新加载。并且非App的这种情况下，不再提供下拉刷新。虽然插件市场也有其他前端模拟的下拉刷新，但性能不佳。一般小程序的大厂案例里，提供左右拖长列表的，都是这种做法。

**Tips**

* 使用竖向滚动时，需要给 <scroll-view> 一个固定高度，通过 css 设置 height。
* 注意：其中只可放置 <swiper-item> 组件，否则会导致未定义的行为。
* 如果遇到current、current-item-id属性设置不生效的问题参考：[组件属性设置不生效解决办法](https://uniapp.dcloud.io/use?id=%e5%b8%b8%e8%a7%81%e9%97%ae%e9%a2%98)
* banner图的切换效果和指示器的样式，有多种风格可自定义，可在[uni-app插件市场](https://ext.dcloud.net.cn/search?q=%E8%BD%AE%E6%92%AD)搜索
* swiper在App的vue中、百度支付宝头条QQ小程序中，不支持内嵌video、map等原生组件。在微信基础库2.4.4起和App nvue2.1.5起支持内嵌原生组件。竖向的swiper内嵌视频可实现抖音、映客等视频垂直拖动切换效果。
* 同时监听 change transition，开始滑动时触发transition, 放开手后，在ios平台触发顺序为 transition... change，Android/微信小程序/支付宝为 transition... change transition...
* [swiper-item](https://uniapp.dcloud.io/component/swiper?id=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** |
| --- | --- | --- | --- |
| item-id | String |  | 该 swiper-item 的标识符 |

**示例** [查看演示](https://uniapp.dcloud.io/h5/pages/component/swiper/swiper)

<template>

<view>

<view class="uni-padding-wrap">

<view class="page-section swiper">

<view class="page-section-spacing">

<swiper class="swiper" :indicator-dots="indicatorDots" :autoplay="autoplay" :interval="interval" :duration="duration">

<swiper-item>

<view class="swiper-item uni-bg-red">A</view>

</swiper-item>

<swiper-item>

<view class="swiper-item uni-bg-green">B</view>

</swiper-item>

<swiper-item>

<view class="swiper-item uni-bg-blue">C</view>

</swiper-item>

</swiper>

</view>

</view>

</view>

<view class="swiper-list">

<view class="uni-list-cell uni-list-cell-pd">

<view class="uni-list-cell-db">指示点</view>

<switch :checked="indicatorDots" @change="changeIndicatorDots" />

</view>

<view class="uni-list-cell uni-list-cell-pd">

<view class="uni-list-cell-db">自动播放</view>

<switch :checked="autoplay" @change="changeAutoplay" />

</view>

</view>

<view class="uni-padding-wrap">

<view class="uni-common-mt">

<text>幻灯片切换时长(ms)</text>

<text class="info">{{duration}}</text>

</view>

<slider @change="durationChange" :value="duration" min="500" max="2000" />

<view class="uni-common-mt">

<text>自动播放间隔时长(ms)</text>

<text class="info">{{interval}}</text>

</view>

<slider @change="intervalChange" :value="interval" min="2000" max="10000" />

</view>

</view>

</template>

export default {

data() {

return {

background: ['color1', 'color2', 'color3'],

indicatorDots: true,

autoplay: true,

interval: 2000,

duration: 500

}

},

methods: {

changeIndicatorDots(e) {

this.indicatorDots = !this.indicatorDots

},

changeAutoplay(e) {

this.autoplay = !this.autoplay

},

intervalChange(e) {

this.interval = e.target.value

},

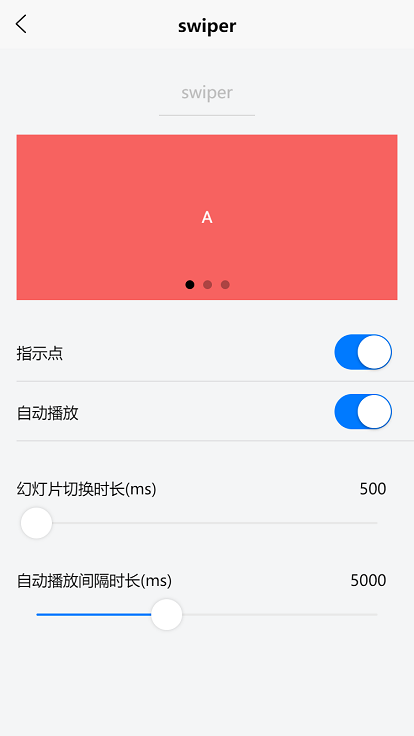
durationChange(e) {

this.duration = e.target.value

}

}

}



#### [movable-area](https://uniapp.dcloud.io/component/movable-view?id=movable-area)

movable-view 的可移动区域

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | √ | √ | √ | √ | x | √ |

**属性说明**

| **属性名** | **类型** | **默认值** | **说明** |
| --- | --- | --- | --- |
| 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 |

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

* movable-area app-nvue平台 暂不支持手势缩放，并且和滚动冲突。
* PC平台不支持 movable-area。

#### [movable-view](https://uniapp.dcloud.io/component/movable-view?id=movable-view)

可移动的视图容器，在页面中可以拖拽滑动

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | √ | √ | √ | √ | x | √ |

**属性说明**

| **属性名** | **类型** | **默认值** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| 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 | 微信小程序、App、H5、百度小程序 |
| out-of-bounds | Boolean | false | 超过可移动区域后，movable-view是否还可以移动 | 微信小程序、App、H5、百度小程序 |
| 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 | 微信小程序、App、H5、百度小程序 |
| 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 | 微信小程序、App、H5、百度小程序 |
| disabled | Boolean | false | 是否禁用 |  |
| 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 | 微信小程序、App、H5 |
| scale-min | Number | 0.5 | 定义缩放倍数最小值 | 微信小程序、App、H5 |
| scale-max | Number | 10 | 定义缩放倍数最大值 | 微信小程序、App、H5 |
| scale-value | Number | 1 | 定义缩放倍数，取值范围为 0.5 - 10 | 微信小程序、App、H5 |
| animation | Boolean | true | 是否使用动画 | 微信小程序、App、H5、百度小程序 |
| @change | 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） |  |
| @scale | EventHandle |  | 缩放过程中触发的事件，event.detail = {x: x, y: y, scale: scale}， | 微信小程序、App、H5、百度小程序 |

除了基本事件外，movable-view提供了两个特殊事件

| **类型** | **触发条件** |
| --- | --- |
| 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 |
| 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 |

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

**movable-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）**

**Tips**

* 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。
* 如果遇到x、y、scale属性设置不生效的问题参考：[组件属性设置不生效解决办法](https://uniapp.dcloud.io/use?id=%e5%b8%b8%e8%a7%81%e9%97%ae%e9%a2%98)
* 除了H5端和app-nvue端，其他平台不支持内嵌video、map等原生组件。更新：微信基础库2.4.4起支持了原生组件在 scroll-view、swiper、movable-view 中的使用

**示例**

<template>

<view class="page-body">

<view class="uni-padding-wrap uni-common-mt">

<view class="uni-title uni-common-mt">

示例 1

<text>\nmovable-view 区域小于 movable-area</text>

</view>

<movable-area>

<movable-view :x="x" :y="y" direction="all" @change="onChange">text</movable-view>

</movable-area>

<view @tap="tap" class="uni-link uni-center uni-common-mt">

点击这里移动至 (30px, 30px)

</view>

<view class="uni-title uni-common-mt">

示例 2

<text>\nmovable-view区域大于movable-area</text>

</view>

<movable-area>

<movable-view class="max" direction="all">text</movable-view>

</movable-area>

</view>

</view>

</template>

export default {

data() {

return {

x: 0,

y: 0,

old: {

x: 0,

y: 0

}

}

},

methods: {

tap: function(e) {

this.x = this.old.x

this.y = this.old.y

this.$nextTick(function() {

this.x = 30

this.y = 30

})

},

onChange: function(e) {

this.old.x = e.detail.x

this.old.y = e.detail.y

}

}

}



#### [cover-view](https://uniapp.dcloud.io/component/cover-view?id=cover-view)

覆盖在原生组件上的文本视图。

小程序框架为了优化体验，部分组件如map、video、textarea、canvas通过原生控件实现，原生组件层级高于前端组件，为了能正常覆盖原生组件，设计了cover-view。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | √ | √ | √ | √ | x | √ |

支持的事件：click

#### [cover-image](https://uniapp.dcloud.io/component/cover-view?id=cover-image)

覆盖在原生组件上的图片视图。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | √ | √ | √ | √ | x | √ |

**属性说明**

| **属性名** | **类型** | **默认值** | **说明** |
| --- | --- | --- | --- |
| src | String |  | 图标路径。支持本地路径、网络路径。不支持 base64 格式。 |

可覆盖的原生组件：<video>、<map>

支持的事件：click

**不支持的 CSS**

* position: fixed
* opacity
* overflow
* padding
* linebreak
* white-space

注意：nvue的cover-view不在上述限制中，它仅支持且全部支持nvue的所有css。

**Tips**

* App端vue页面 cover-view、cover-image 中不支持嵌套其它组件，包括再次嵌套cover-view，仅可覆盖video、map。App端nvue页面自2.1.5起没有这些限制。
* App端还可以使用plus.nativeObj.view绘制原生内容，参考:[uni-app中使用5+界面控件](https://ask.dcloud.net.cn/article/35036" \t "_blank)、[plus.nativeObj.view规范](https://www.html5plus.org/doc/zh_cn/nativeobj.html#plus.nativeObj.View)
* App端还提供了更灵活和强大的subNvue，参考[原生子窗体subNvue](https://uniapp.dcloud.io/api/window/subNVues)
* 在 video 组件中使用时，若想在全屏模式下使用cover-view，只有在微信小程序、App端的nvue页面可实现。
* 百度小程序iOS端暂不支持一个页面有多个video时嵌套cover-view。
* 支付宝小程序中 cover-view 不支持嵌套。

**微信小程序的cover-view使用注意：**

* cover-view和cover-image的aria-role仅可设置为button，读屏模式下才可以点击，并朗读出“按钮”；为空时可以聚焦，但不可点击
* 基础库 2.2.4 起支持 touch 相关事件，也可使用 hover-class 设置点击态
* 基础库 2.1.0 起支持设置 scale rotate 的 css 样式，包括 transition 动画
* 基础库 1.9.90 起 cover-view 支持 overflow: scroll，但不支持动态更新 overflow
* 基础库 1.9.90 起最外层 cover-view 支持 position: fixed
* 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。
* 基础库 1.6.0 起支持css transition动画，transition-property只支持transform (translateX, translateY)与opacity。
* 基础库 1.6.0 起支持css opacity。
* 事件模型遵循冒泡模型，但不会冒泡到原生组件。
* 文本建议都套上cover-view标签，避免排版错误。
* 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。
* 建议子节点不要溢出父节点
* 支持使用 z-index 控制层级
* 默认设置的样式有：white-space: nowrap; line-height: 1.2; display: block;
* 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** [查看演示](https://uniapp.dcloud.io/h5/pages/component/cover-view/cover-view)

<template>

<view class="page">

<video class="video" id="demoVideo" :controls="disable" :show-fullscreen-btn="disable" :show-play-btn="disable"

:show-center-play-btn="disable" :enable-progress-gesture="disable" @fullscreenchange="fullscreenchange" src="https://img.cdn.aliyun.dcloud.net.cn/guide/uniapp/%E7%AC%AC1%E8%AE%B2%EF%BC%88uni-app%E4%BA%A7%E5%93%81%E4%BB%8B%E7%BB%8D%EF%BC%89-%20DCloud%E5%AE%98%E6%96%B9%E8%A7%86%E9%A2%91%E6%95%99%E7%A8%8B@20181126.mp4">

<cover-view class="controls-title">简单的自定义 controls</cover-view>

<cover-image class="controls-play img" @click="play" src="../../../static/play.png"></cover-image>

<cover-image class="controls-pause img" @click="pause" src="../../../static/pause.png"></cover-image>

</video>

</view>

</template>

<script>

export default {

data() {

return {

videoCtx: null,

disable: false

}

},

mounted() {

this.videoCtx = uni.createVideoContext('demoVideo')

},

methods: {

play(event) {

this.videoCtx.play();

uni.showToast({

title: '开始播放',

icon: 'none'

});

},

pause(event) {

this.videoCtx.pause();

uni.showToast({

title: '暂停播放',

icon: 'none'

});

}

}

}

</script>

<style>

.page {

display: flex;

justify-content: center;

}

.video {

position: relative;

}

cover-view,

cover-image {

display: inline-block;

}

.img {

position: absolute;

width: 100rpx;

height: 100rpx;

top: 50%;

margin-top: -50rpx;

}

.controls-play {

left: 50rpx;

}

.controls-pause {

right: 50rpx;

}

.controls-title {

width: 100%;

text-align: center;

color: #FFFFFF;

}

</style>

# 基础内容

#### [icon](https://uniapp.dcloud.io/component/icon?id=icon)

图标。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | √(2.2.3+) | √ | √ | √ | √ | √ |

**Tips**

* 由于 icon 组件各端表现存在差异，可以通过使用 [字体图标](https://uniapp.dcloud.io/frame?id=%e5%ad%97%e4%bd%93%e5%9b%be%e6%a0%87) 的方式来弥补各端差异。

**属性说明**

| **属性名** | **类型** | **默认值** | **说明** |
| --- | --- | --- | --- |
| type | String |  | icon的类型 |
| size | Number | 23 | icon的大小，单位px |
| color | Color |  | icon的颜色，同css的color |

各平台 type 有效值说明：

| **平台** | **type 有效值** |
| --- | --- |
| App、H5、微信小程序、QQ小程序 | success, success\_no\_circle, info, warn, waiting, cancel, download, search, clear |
| 支付宝小程序 | info, warn, waiting, cancel, download, search, clear, success, success\_no\_circle,loading |
| 百度小程序 | success, info, warn, waiting, success\_no\_circle, clear, search, personal, setting, top, close, cancel, download, checkboxSelected, radioSelected, radioUnselect |

**示例**

<view class="item" v-for="(value,index) in iconType" :key="index">

<icon :type="value" size="26"/>

<text>{{value}}</text>

</view>

export default {

data() {

return {

iconType: ['success']

}

},

onLoad() {

// #ifdef APP-PLUS|| MP-WEIXIN

this.iconType = ['success', 'success\_no\_circle', 'info', 'warn', 'waiting', 'cancel', 'download', 'search','clear']

// #endif

// #ifdef MP-ALIPAY

this.iconType = ['info', 'warn', 'waiting', 'cancel', 'download', 'search', 'clear', 'success', 'success\_no\_circle', 'loading']

// #endif

// #ifdef MP-BAIDU

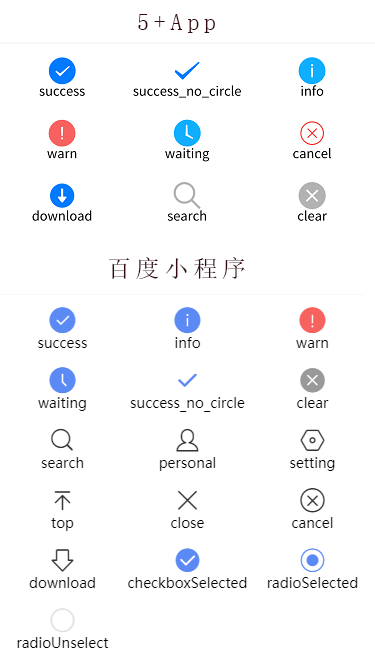
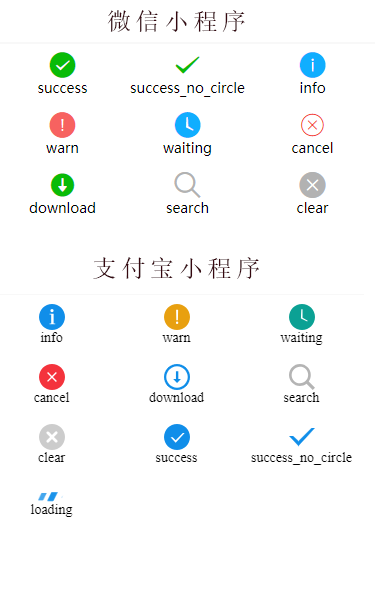
this.iconType = ['success', 'info', 'warn', 'waiting', 'success\_no\_circle', 'clear', 'search', 'personal', 'setting', 'top', 'close', 'cancel', 'download', 'checkboxSelected', 'radioSelected', 'radioUnselect']

// #endif

}

}

**效果展示**



#### [text](https://uniapp.dcloud.io/component/text?id=text)

文本。

**属性说明**

| **属性名** | **类型** | **默认值** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| selectable | Boolean | false | 文本是否可选 |  |
| space | String |  | 显示连续空格 | App、H5、微信小程序 |
| decode | Boolean | false | 是否解码 | App、H5、微信小程序 |

**space 值说明**

| **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**

* <text> 组件内只支持嵌套 <text>，不支持其它组件或自定义组件，否则会引发在不同平台的渲染差异。。
* decode 可以解析的有 &nbsp; &lt; &gt; &amp; &apos; &ensp; &emsp;。
* 各个操作系统的空格标准并不一致。
* 除了文本节点以外的其他节点都无法长按选中。
* 支持 \n 方式换行。
* 如果使用 <span> 组件编译时会被转换为 <text>。

**示例** [查看演示](https://uniapp.dcloud.io/h5/pages/component/text/text)

<template>

<view>

<view class="uni-padding-wrap uni-common-mt">

<view class="text-box" scroll-y="true">

<text>{{text}}</text>

</view>

<view class="uni-btn-v">

<button type="primary" :disabled="!canAdd" @click="add">add line</button>

<button type="warn" :disabled="!canRemove" @click="remove">remove line</button>

</view>

</view>

</view>

</template>

export default {

data() {

return {

texts: [

'HBuilder，200万开发者选择的IDE',

'MUI，轻巧、漂亮的前端开源框架',

'wap2app，M站快速转换原生体验的App',

'5+Runtime，为HTML5插上原生的翅膀',

'HBuilderX，轻巧、极速，极客编辑器',

'uni-app，终极跨平台方案',

'HBuilder，200万开发者选择的IDE',

'MUI，轻巧、漂亮的前端开源框架',

'wap2app，M站快速转换原生体验的App',

'5+Runtime，为HTML5插上原生的翅膀',

'HBuilderX，轻巧、极速，极客编辑器',

'uni-app，终极跨平台方案',

'......'

],

text: '',

canAdd: true,

canRemove: false,

extraLine: []

}

},

methods: {

add: function(e) {

this.extraLine.push(this.texts[this.extraLine.length % 12]);

this.text = this.extraLine.join('\n');

this.canAdd = this.extraLine.length < 12;

this.canRemove = this.extraLine.length > 0;

},

remove: function(e) {

if (this.extraLine.length > 0) {

this.extraLine.pop();

this.text = this.extraLine.join('\n');

this.canAdd = this.extraLine.length < 12;

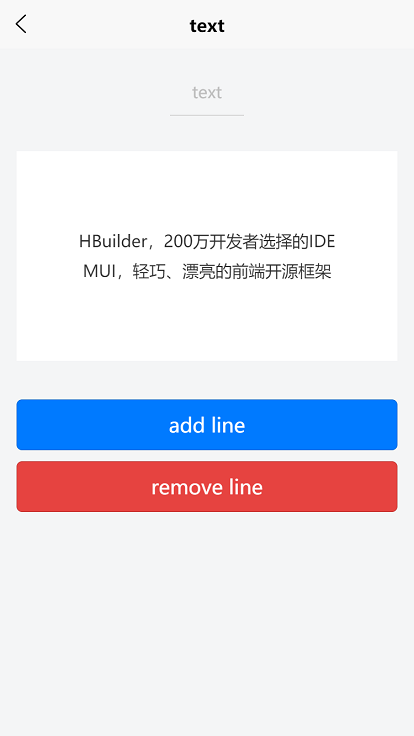
this.canRemove = this.extraLine.length > 0;

}

}

}

}



#### [rich-text](https://uniapp.dcloud.io/component/rich-text?id=rich-text)

富文本。

**属性说明**

| **属性名** | **类型** | **默认值** | **说明** | **平台兼容** |
| --- | --- | --- | --- | --- |
| nodes | Array / String | [] | 节点列表 / HTML String |  |
| [space](https://developers.weixin.qq.com/miniprogram/dev/component/rich-text.html) | string |  | 显示连续空格 | 微信基础库2.4.1+、QQ小程序 |
| selectable | Boolean | false | 富文本是否可以长按选中，可用于复制，粘贴等场景 | 百度APP 11.10+ |

**注意**

* app-nvue 平台 nodes 属性只支持使用 Array 类型
* 支付宝小程序 nodes 属性只支持使用 Array 类型。

如果需要支持 HTML String，则需要自己将 HTML String转化为 nodes 数组，可使用 [html-parser](https://github.com/dcloudio/hello-uniapp/blob/master/common/html-parser.js) 转换。

支持默认事件，包括：click、touchstart、touchmove、touchcancel、touchend、longpress。

**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**](https://uniapp.dcloud.io/component/rich-text?id=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** [查看演示](https://uniapp.dcloud.io/h5/pages/component/rich-text/rich-text)

<template>

<view class="content">

<page-head :title="title"></page-head>

<view class="uni-padding-wrap">

<view class="uni-title uni-common-mt">

数组类型

<text>\nnodes属性为Array</text>

</view>

<view class="uni-common-mt" style="background:#FFF; padding:20rpx;">

<rich-text :nodes="nodes"></rich-text>

</view>

<view class="uni-title uni-common-mt">

字符串类型

<text>\nnodes属性为String</text>

</view>

<view class="uni-common-mt" style="background:#FFF; padding:20rpx;">

<rich-text :nodes="strings"></rich-text>

</view>

</view>

</view>

</template>

export default {

data() {

return {

nodes: [{

name: 'div',

attrs: {

class: 'div-class',

style: 'line-height: 60px; color: red; text-align:center;'

},

children: [{

type: 'text',

text: 'Hello&nbsp;uni-app!'

}]

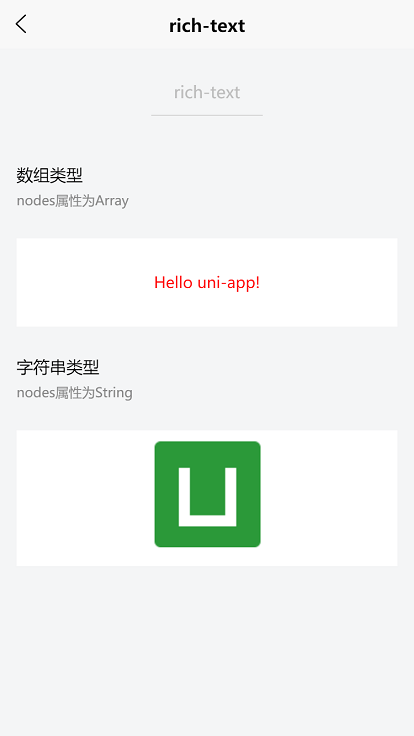
}],

strings: '<div style="text-align:center;"><img src="https://img-cdn-qiniu.dcloud.net.cn/uniapp/images/uni@2x.png"/></div>'

}

}

}



**Tips**

* nodes 不推荐使用 String 类型，性能会有所下降。
* rich-text 组件内屏蔽所有节点的事件。所以如果内容中有链接、图片需要点击，则不能使用rich-text，此时可在[uni-app插件市场](https://ext.dcloud.net.cn/)搜索uparse插件使用。app-nvue的rich-text组件支持链接图片点击，且nvue下不能使用uparse等插件。
* attrs 属性不支持 id ，支持 class 。
* name 属性大小写不敏感。
* 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。
* img 标签仅支持网络图片。
* 如果在自定义组件中使用 rich-text 组件，那么仅自定义组件的 css 样式对 rich-text 中的 class 生效。

#### [progress](https://uniapp.dcloud.io/component/progress?id=progress)

进度条。

**属性说明**

| **属性名** | **类型** | **默认值** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| percent | Float | 无 | 百分比0~100 |  |
| show-info | Boolean | false | 在进度条右侧显示百分比 |  |
| border-radius | number/string | 0 | 圆角大小 | 微信基础库2.3.1+、QQ小程序 |
| font-size | number/string | 16 | 右侧百分比字体大小 | 微信基础库2.3.1+、QQ小程序 |
| stroke-width | Number | 6 | 进度条线的宽度，单位px |  |
| activeColor | Color | #09BB07（百度为#E6E6E6） | 已选择的进度条的颜色 |  |
| backgroundColor | Color | #EBEBEB | 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 | App、H5、微信小程序、QQ小程序 |
| duration | number | 30 | 进度增加1%所需毫秒数 | 微信基础库2.8.2+ |
| @activeend | EventHandle |  | 动画完成事件 | 微信小程序 |

**示例** [查看演示](https://uniapp.dcloud.io/h5/pages/component/progress/progress)

<template>

<view>

<view class="uni-padding-wrap uni-common-mt">

<view class="progress-box">

<progress percent="20" show-info stroke-width="3" />

</view>

<view class="progress-box">

<progress percent="40" active stroke-width="3" />

</view>

<view class="progress-box">

<progress percent="60" active stroke-width="3" backgroundColor="#999"/>

</view>

<view class="progress-box">

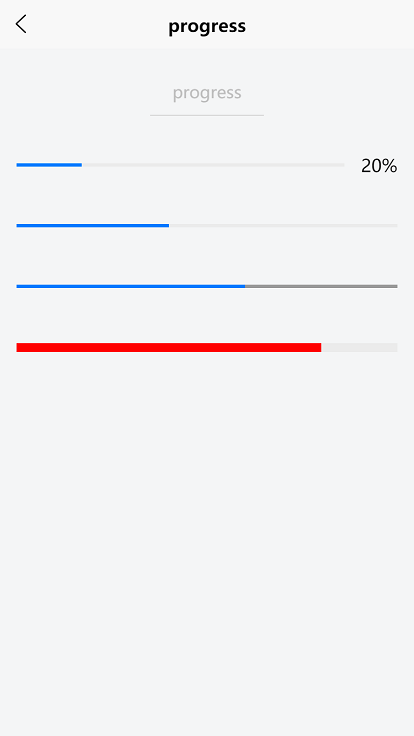
<progress percent="80" activeColor="red" active stroke-width="8" />

</view>

</view>

</view>

</template>



# 表单组件

#### [button](https://uniapp.dcloud.io/component/button?id=button)

按钮。

**属性说明**

| **属性名** | **类型** | **默认值** | **说明** | **生效时机** | **平台差异说明** |
| --- | --- | --- | --- | --- | --- |
| 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 |  | App-nvue 平台，在 ios 上为雪花，Android上为圆圈 |
| 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 |  | 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 |  | App-nvue 平台暂不支持 |
| 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 |  |  |
| @getphonenumber | Handler |  | 获取用户手机号回调 | open-type="getPhoneNumber" | 微信小程序 |
| @getuserinfo | Handler |  | 用户点击该按钮时，会返回获取到的用户信息，从返回参数的detail中获取到的值同uni.getUserInfo | open-type="getUserInfo" | 微信小程序 |
| @error | Handler |  | 当使用开放能力时，发生错误的回调 | open-type="launchApp" | 微信小程序 |
| @opensetting | Handler |  | 在打开授权设置页并关闭后回调 | open-type="openSetting" | 微信小程序 |
| @launchapp | Handler |  | 打开 APP 成功的回调 | open-type="launchApp" | 微信小程序 |

* **注1：button-hover 默认为 {background-color: rgba(0, 0, 0, 0.1); opacity: 0.7;}**
* open-type="launchApp"时需要调起的APP接入微信OpenSDK[详见](https://developers.weixin.qq.com/miniprogram/dev/framework/open-ability/launchApp.html" \t "_blank)

**size 有效值**

| **值** | **说明** |
| --- | --- |
| default | 默认大小 |
| mini | 小尺寸 |

**type 有效值**

| **值** | **说明** |
| --- | --- |
| primary | 微信小程序为绿色，App、H5、百度小程序、支付宝小程序为蓝色，头条小程序为红色，QQ小程序为浅蓝色 |
| 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** | **平台差异说明** |
| --- | --- | --- |
| feedback | 打开“意见反馈”页面，用户可提交反馈内容并上传日志 | App、微信小程序、QQ小程序 |
| share | 触发用户转发 | 微信小程序、百度小程序、支付宝小程序、头条小程序、QQ小程序 |
| getUserInfo | 获取用户信息，可以从@getuserinfo回调中获取到用户信息，包括手机号、头像、昵称等信息 | 微信小程序、百度小程序、QQ小程序 |
| contact | 打开客服会话，如果用户在会话中点击消息卡片后返回应用，可以从 @contact 回调中获得具体信息 | 微信小程序、百度小程序 |
| getPhoneNumber | 获取用户手机号，可以从@getphonenumber回调中获取到用户信息 | 微信小程序、百度小程序、头条小程序 |
| launchApp | 打开APP，可以通过app-parameter属性设定向APP传的参数 | [微信小程序](https://developers.weixin.qq.com/miniprogram/dev/framework/open-ability/launchApp.html)、[QQ小程序](https://q.qq.com/wiki/develop/miniprogram/frame/open_ability/open_app.html) |
| openSetting | 打开授权设置页 | 微信小程序、百度小程序 |
| getAuthorize | 支持小程序授权 | 支付宝小程序 |
| contactShare | 分享到通讯录好友 | 支付宝小程序 |
| lifestyle | 关注生活号 | 支付宝小程序 |
| openGroupProfile | 呼起QQ群资料卡页面，可以通过group-id属性设定需要打开的群资料卡的群号，同时manifest中必须配置groupIdList | QQ小程序基础库1.4.7版本+ |

**注意**

* 在小程序中，开发者可以登录 [微信小程序管理后台](https://mp.weixin.qq.com/) 、[QQ小程序后台](https://q.qq.com/" \l "/" \t "_blank)后，进入菜单“客服反馈”页面获取反馈内容。
* 在 App 中，开发者登录 [DCloud开发者中心](https://dev.dcloud.net.cn/" \t "_blank) 后点击应用名称，进入左侧菜单“用户反馈”页面获取反馈内容。
* 点击 share 分享按钮时会触发 [onShareAppMessage](https://uniapp.dcloud.io/api/plugins/share)
* 支付宝小程序平台，获取用户手机号时，建议先通过条件编译的方式，调用支付宝原生API，[参考](https://docs.alipay.com/mini/api/getphonenumber" \t "_blank)

**示例** [查看演示](https://uniapp.dcloud.io/h5/pages/component/button/button)

<template>

<view>

<view class="uni-padding-wrap uni-common-mt">

<button type="primary">页面主操作 Normal</button>

<button type="primary" loading="true">页面主操作 Loading</button>

<button type="primary" disabled="true">页面主操作 Disabled</button>

<button type="default">页面次要操作 Normal</button>

<button type="default" disabled="true">页面次要操作 Disabled</button>

<button type="warn">警告类操作 Normal</button>

<button type="warn" disabled="true">警告类操作 Disabled</button>

<view class="button-sp-area">

<button type="primary" plain="true">按钮</button>

<button type="primary" disabled="true" plain="true">不可点击的按钮</button>

<button type="default" plain="true">按钮</button>

<button type="default" disabled="true" plain="true">按钮</button>

<button class="mini-btn" type="primary" size="mini">按钮</button>

<button class="mini-btn" type="default" size="mini">按钮</button>

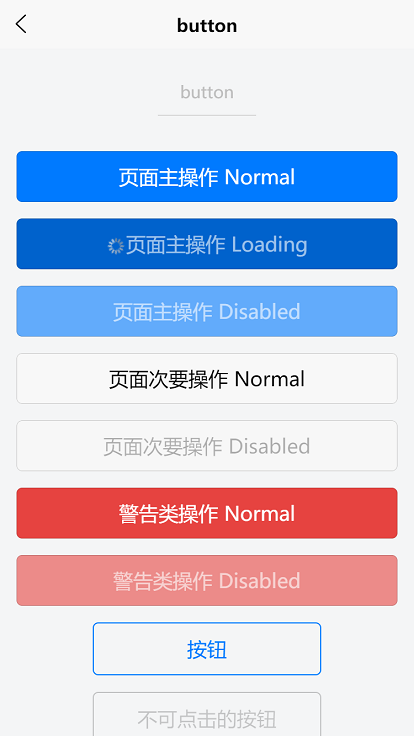
<button class="mini-btn" type="warn" size="mini">按钮</button>

</view>

</view>

</view>

</template>



**checkbox-group**

多项选择器，内部由多个 checkbox 组成。

**属性说明**

| **属性名** | **类型** | **默认值** | **说明** |
| --- | --- | --- | --- |
| @change | 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](https://uniapp.dcloud.io/component/checkbox?id=checkbox)

多选项目。

**属性说明**

| **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** [查看演示](https://uniapp.dcloud.io/h5/pages/component/checkbox/checkbox)

<template>

<view>

<view class="uni-padding-wrap uni-common-mt">

<view class="uni-title uni-common-mt">默认样式</view>

<view>

<checkbox-group>

<label>

<checkbox value="cb" checked="true" />选中

</label>

<label>

<checkbox value="cb" />未选中

</label>

</checkbox-group>

</view>

<view class="uni-title uni-common-mt">不同颜色和尺寸的checkbox</view>

<view>

<checkbox-group>

<label>

<checkbox value="cb" checked="true" color="#FFCC33" style="transform:scale(0.7)" />选中

</label>

<label>

<checkbox value="cb" color="#FFCC33" style="transform:scale(0.7)" />未选中

</label>

</checkbox-group>

</view>

</view>

<view class="uni-padding-wrap">

<view class="uni-title uni-common-mt">

推荐展示样式

<text>\n使用 uni-list 布局</text>

</view>

</view>

<view class="uni-list">

<checkbox-group @change="checkboxChange">

<label class="uni-list-cell uni-list-cell-pd" v-for="item in items" :key="item.value">

<view>

<checkbox :value="item.value" :checked="item.checked" />

</view>

<view>{{item.name}}</view>

</label>

</checkbox-group>

</view>

</view>

</template>

<script>

export default {

data() {

return {

title: 'checkbox 复选框',

items: [{

value: 'USA',

name: '美国'

},

{

value: 'CHN',

name: '中国',

checked: 'true'

},

{

value: 'BRA',

name: '巴西'

},

{

value: 'JPN',

name: '日本'

},

{

value: 'ENG',

name: '英国'

},

{

value: 'FRA',

name: '法国'

}

]

}

},

methods: {

checkboxChange: function (e) {

var items = this.items,

values = e.detail.value;

for (var i = 0, lenI = items.length; i < lenI; ++i) {

const item = items[i]

if(values.includes(item.value)){

this.$set(item,'checked',true)

}else{

this.$set(item,'checked',false)

}

}

}

}

}

</script>

<style>

.uni-list-cell {

justify-content: flex-start

}

</style>



示例代码说明：以上示例代码从hello uni-app示例中复制，涉及的css样式在hello uni-app的app.vue和uni.css中

预览H5效果：使用浏览器的手机模式访问<https://uniapp.dcloud.io/h5/pages/component/checkbox/checkbox>

**注意**

* checkbox的默认颜色，在不同平台不一样。微信小程序是绿色的，头条小程序为红色，其他平台是蓝色的。更改颜色使用color属性。
* 如需调节checkbox大小，可通过css的scale方法调节，如缩小到70%style="transform:scale(0.7)"

#### [editor](https://uniapp.dcloud.io/component/editor?id=editor)

富文本编辑器，可以对图片、文字格式进行编辑和混排。

编辑器导出内容支持带标签的 html和纯文本的 text，编辑器内部采用 delta 格式进行存储。

通过setContents接口设置内容时，解析插入的 html 可能会由于一些非法标签导致解析错误，建议开发者在应用内使用时通过 delta 进行插入。

富文本组件内部引入了一些基本的样式使得内容可以正确的展示，开发时可以进行覆盖。需要注意的是，在其它组件或环境中使用富文本组件导出的html时，需要额外引入[这段样式](https://github.com/dcloudio/uni-app/blob/master/src/core/view/components/editor/editor.css)，并维护<ql-container><ql-editor></ql-editor></ql-container>的结构，参考：[使用 editor 组件导出的 html](https://ask.dcloud.net.cn/article/36205)。

图片控件仅初始化时设置有效。

相关 api：[editorContext](https://uniapp.dcloud.io/api/media/editor-context)

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| 2.0.0+ [自定义组件编译模式](https://ask.dcloud.net.cn/article/35843)，不含nvue | 2.4.5+ | 基础库 2.7.0+ | x | x | x | x |

本功能自HBuilderX2.0起支持。运行到微信小程序工具时，注意在微信工具里选择最新的基础库。

editor组件目前只有H5、App的vue页面和微信支持，其他端的富文本编辑解决方案，可使用web-view加载web页面，也可搜索[插件市场](https://ext.dcloud.net.cn/search?q=%E5%AF%8C%E6%96%87%E6%9C%AC%E7%BC%96%E8%BE%91) 获取简单的markdown富文本编辑器

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| read-only | boolean | false | 否 | 设置编辑器为只读 |
| placeholder | string |  | 否 | 提示信息 |
| show-img-size | boolean | false | 否 | 点击图片时显示图片大小控件 |
| show-img-toolbar | boolean | false | 否 | 点击图片时显示工具栏控件 |
| show-img-resize | boolean | false | 否 | 点击图片时显示修改尺寸控件 |
| @ready | eventhandle |  | 否 | 编辑器初始化完成时触发 |
| @focus | eventhandle |  | 否 | 编辑器聚焦时触发，event.detail = {html, text, delta} |
| @blur | eventhandle |  | 否 | 编辑器失去焦点时触发，detail = {html, text, delta} |
| @input | eventhandle |  | 否 | 编辑器内容改变时触发，detail = {html, text, delta} |
| @statuschange | eventhandle |  | 否 | 通过 Context 方法改变编辑器内样式时触发，返回选区已设置的样式 |

编辑器内支持部分 HTML 标签和内连样式，不支持**class**和**id**

支持的标签

不满足的标签会被忽略，<div>会被转行为<p>储存。

| **类型** | **节点** |
| --- | --- |
| 行内元素 | <span> <strong> <b> <ins> <em> <i> <u> <a> <del> <s> <sub> <sup> <img> |
| 块级元素 | <p> <h1> <h2> <h3> <h4> <h5> <h6> <hr> <ol> <ul> <li> |

支持的内连样式

内联样式仅能设置在行内元素或块级元素上，不能同时设置。例如 font-size` 归类为行内元素属性，在 p 标签上设置是无效的。

| **类型** | **样式** |
| --- | --- |
| 块级样式 | text-align direction margin margin-top margin-left margin-right margin-bottom padding padding-top padding-left padding-right padding-bottom line-height text-indent |
| 行内样式 | font font-size font-style font-variant font-weight font-family letter-spacing text-decoration color background-color |

**注意事项**

1. 插入的 html 中事件绑定会被移除
2. formats 中的 color 属性会统一以 hex 格式返回
3. 粘贴时仅纯文本内容会被拷贝进编辑器
4. 插入 html 到编辑器内时，编辑器会删除一些不必要的标签，以保证内容的统一。例如<p><span>xxx</span></p>会改写为<p>xxx</p>
5. 编辑器聚焦时页面会被上推，系统行为以保证编辑区可见

**示例代码**

<template>

<view class="container">

<editor id="editor" class="ql-container" :placeholder="placeholder" @ready="onEditorReady"></editor>

<button type="warn" @tap="undo">撤销</button>

</view>

</template>

<script>

export default {

data() {

return {

placeholder: '开始输入...'

}

},

methods: {

onEditorReady() {

uni.createSelectorQuery().select('#editor').context((res) => {

this.editorCtx = res.context

}).exec()

},

undo() {

this.editorCtx.undo()

}

}

}

</script>

<style>

.container {

padding: 10px;

}

#editor {

width: 100%;

height: 300px;

background-color: #CCCCCC;

}

button {

margin-top: 10px;

}

</style>

#### [form](https://uniapp.dcloud.io/component/form?id=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。

**属性说明**

| **属性名** | **类型** | **说明** | **平台差异说明** |
| --- | --- | --- | --- |
| report-submit | Boolean | 是否返回 formId 用于发送[模板消息](https://developers.weixin.qq.com/miniprogram/dev/framework/open-ability/template-message.html) | 微信小程序、支付宝小程序 |
| @submit | EventHandle | 携带 form 中的数据触发 submit 事件，event.detail = {value : {'name': 'value'} , formId: ''}，report-submit 为 true 时才会返回 formId |  |
| @reset | EventHandle | 表单重置时会触发 reset 事件 |  |

**示例** [查看演示](https://uniapp.dcloud.io/h5/pages/component/form/form)

<template>

<view>

<view>

<form @submit="formSubmit" @reset="formReset">

<view class="uni-form-item uni-column">

<view class="title">switch</view>

<view>

<switch name="switch" />

</view>

</view>

<view class="uni-form-item uni-column">

<view class="title">radio</view>

<radio-group name="radio">

<label>

<radio value="radio1" /><text>选项一</text>

</label>

<label>

<radio value="radio2" /><text>选项二</text>

</label>

</radio-group>

</view>

<view class="uni-form-item uni-column">

<view class="title">checkbox</view>

<checkbox-group name="checkbox">

<label>

<checkbox value="checkbox1" /><text>选项一</text>

</label>

<label>

<checkbox value="checkbox2" /><text>选项二</text>

</label>

</checkbox-group>

</view>

<view class="uni-form-item uni-column">

<view class="title">slider</view>

<slider value="50" name="slider" show-value></slider>

</view>

<view class="uni-form-item uni-column">

<view class="title">input</view>

<input class="uni-input" name="input" placeholder="这是一个输入框" />

</view>

<view class="uni-btn-v">

<button form-type="submit">Submit</button>

<button type="default" form-type="reset">Reset</button>

</view>

</form>

</view>

</view>

</template>

<script>

export default {

data() {

return {

}

},

methods: {

formSubmit: function(e) {

console.log('form发生了submit事件，携带数据为：' + JSON.stringify(e.detail.value))

var formdata = e.detail.value

uni.showModal({

content: '表单数据内容：' + JSON.stringify(formdata),

showCancel: false

});

},

formReset: function(e) {

console.log('清空数据')

}

}

}

</script>

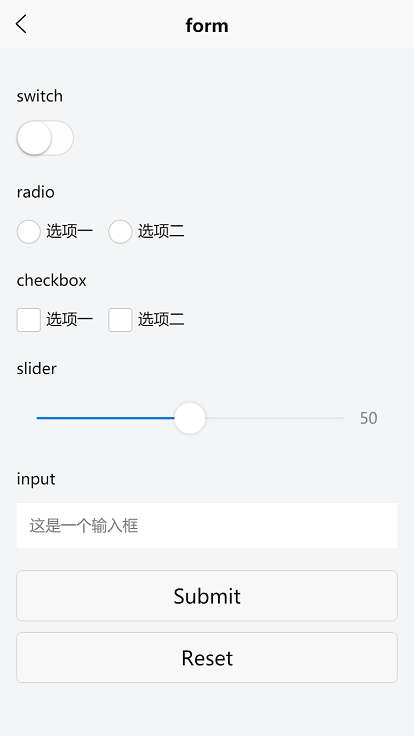
<style>

.uni-form-item .title {

padding: 20rpx 0;

}

</style>



**tips**

* [插件市场](http://ext.dcloud.net.cn/search?q=%E8%A1%A8%E5%8D%95%E6%A0%A1%E9%AA%8C)有表单校验插件

#### [input](https://uniapp.dcloud.io/component/input?id=input)

输入框。

**属性说明**

| **属性名** | **类型** | **默认值** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| 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 | App、微信小程序、百度小程序、QQ小程序 |
| focus | Boolean | false | 获取焦点。 | 在 H5 平台能否聚焦以及软键盘是否跟随弹出，取决于当前浏览器本身的实现。 |
| confirm-type | String | done | 设置键盘右下角按钮的文字，仅在 type="text" 时生效。 |  |
| confirm-hold | Boolean | false | 点击键盘右下角按钮时是否保持键盘不收起 | App、微信小程序、支付宝小程序、百度小程序、QQ小程序 |
| cursor | Number |  | 指定focus时的光标位置 |  |
| 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 |  |
| 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 |  |
| adjust-position | Boolean | true | 键盘弹起时，是否自动上推页面 | App（softinputMode 为 adjustResize 时无效）、微信小程序、百度小程序、QQ小程序 |
| @input | EventHandle |  | 当键盘输入时，触发input事件，event.detail = {value} | 差异见下方 Tips |
| @focus | EventHandle |  | 输入框聚焦时触发，event.detail = { value, height }，height 为键盘高度 | 仅微信小程序、App（2.2.3+） 、QQ小程序支持 height |
| @blur | EventHandle |  | 输入框失去焦点时触发，event.detail = {value: value} |  |
| @confirm | EventHandle |  | 点击完成按钮时触发，event.detail = {value: value} |  |

**Tips**

* input 事件处理函数可以直接 return 一个字符串，将替换输入框的内容。仅微信小程序支持。
* input 事件处理函数内实时修改当前值不生效，可以延迟设置，例如：setTimeout(() => { this.value = 100 }, 0)。
* input 组件上有默认的 min-height 样式，如果 min-height 的值大于 height 的值那么 height 样式无效。

**type 有效值**

| **值** | **说明** | **平台差异说明** |
| --- | --- | --- |
| text | 文本输入键盘 |  |
| number | 数字输入键盘 | 注意iOS的vue页面弹出的数字键盘并非9宫格方式 |
| idcard | 身份证输入键盘 | 微信、支付宝、百度、QQ小程序 |
| digit | 带小数点的数字键盘 | App的nvue页面、微信、支付宝、百度、头条、QQ小程序 |

**注意事项**

* 小程序平台，number 类型只支持输入整型数字。微信开发者工具上体现不出效果，请使用真机预览。
* 如果需要在小程序平台输入浮点型数字，请使用 digit 类型。
* 小程序端input在置焦时，会表现为原生控件，此时会层级变高。如需前端组件遮盖input，需让input失焦，或使用cover-view等覆盖原生控件的方案，[参考](https://uniapp.dcloud.io/component/native-component)。具体来讲，阿里小程序的input为text且置焦为原生控件；微信、头条、QQ所有input置焦均为原生控件；百度小程序置焦时仍然是非原生的。也可以参考[原生控件](https://uniapp.dcloud.io/component/native-component)文档
* input组件若不想弹出软键盘，可设置为disable
* App平台的vue页面及 H5平台 的弹出键盘使用的是浏览器控制的键盘，丰富程度略低。可通过插件市场找到更多类型的键盘。
* 插件市场有各种类型的模拟键盘，比如车牌键盘、身份证键盘，可去插件市场搜索 [键盘](https://ext.dcloud.net.cn/search?q=%E9%94%AE%E7%9B%98)。
* 搜索框ui组件，可参考uni ui的search组件或封装好的页面模板。[详见](https://ext.dcloud.net.cn/search?q=search)
* uni-app插件市场有输入文字后自动提示候选的组件，可搜索 [autocomplete](https://ext.dcloud.net.cn/search?q=autocomplete) 查看。

**confirm-type 有效值**

| **值** | **说明** | **平台差异说明** |
| --- | --- | --- |
| send | 右下角按钮为“发送” | 微信、支付宝、百度小程序、App的nvue |
| search | 右下角按钮为“搜索” |  |
| next | 右下角按钮为“下一个” | 微信、支付宝、百度小程序、App的nvue |
| go | 右下角按钮为“前往” |  |
| done | 右下角按钮为“完成” | 微信、支付宝、百度小程序、App的nvue |

* App平台的nvue页面，如果是uni-app编译模式，直接使用此属性设置即可生效。如果是weex编译模式，需通过weex的api设置，[weex相关文档参考](https://weex.apache.org/zh/docs/components/input.html#%E5%B1%9E%E6%80%A7)
* App平台的vue页面不支持控制键盘右下角为“发送”，涉及聊天的建议使用nvue。

App平台ios端软键盘上方横条出去方案

App平台在iOS上，webview中的软键盘弹出时，默认在软键盘上方有一个横条，显示着：上一项、下一项和完成等按钮。如不想显示这个横条，可以配置softinputNavBar: 'none'

配置方式，在 pages.json 中配置 style

"app-plus": {

"softinputNavBar": "none"

}

* 如需使用js动态设置softinputNavBar
* this.$mp.page.$getAppWebview().setStyle({
* softinputNavBar: 'none'

})

如果是nvue页面，默认就没有键盘上方的横条，无需任何设置。

关于软键盘弹出的逻辑说明

App平台软键盘弹出有 adjustResize|adjustPan 两种模式，默认为 adjustPan 模式，小程序平台只支持 adjustPan 模式，H5平台因不同浏览器而异

* adjustResize：软键盘弹出时，webview窗体高度挤压。屏幕高度=webview窗体高度+软键盘高度
* adjustPan：软键盘弹出时，webview窗体高度不变，但窗体上推，以保证输入框不被软键盘盖住

配置方式，在 pages.json 中配置 style

"app-plus": {

"softinputMode": "adjustResize"

}

**注意**

* adjustResize模式在Android App上，弹起键盘和收回键盘时，因为要重设webview窗体高度，可能会在个别安卓机型闪现灰屏或漏出下层页面内容。
* 小程序端在 input 聚焦期间，避免使用 css 动画。
* H5平台只能在用户交互时修改 focus 生效。
* 如果遇到 focus 属性设置不生效的问题参考：[组件属性设置不生效解决办法](https://uniapp.dcloud.io/use?id=%e5%b8%b8%e8%a7%81%e9%97%ae%e9%a2%98)

关于软键盘收起的逻辑说明

* Android上在软键盘弹出后，点击back或点击非置焦区域可收起软键盘。
* iOS上如果软键盘上方有带有“完成”的横条，则需要点完成才能收起键盘；如果没有软键盘上方横条，则点击非input/textarea区域即可收起软键盘

以上为默认逻辑，uni-app同时提供了隐藏软键盘的api：[uni.hideKeyboard()](https://uniapp.dcloud.io/api/key?id=hidekeyboard" \t "_blank)

App平台原生输入框的说明

在app平台，有titleNView配置的[searchinput](https://uniapp.dcloud.io/collocation/pages?id=app-titlenview)原生输入框和plus.nativeObj.view的drawinput。这两种方式的输入框都是原生的，不是webview里的。

* 原生输入框在iOS上不会有软键盘上方的横条
* 原生输入框一样受配置的adjustPan|adjustResize模式影响

**input示例** [查看演示](https://uniapp.dcloud.io/h5/pages/component/input/input)

<template>

<view>

<view class="uni-common-mt">

<view class="uni-form-item uni-column">

<view class="title">可自动聚焦的input</view>

<input class="uni-input" focus placeholder="自动获得焦点" />

</view>

<view class="uni-form-item uni-column">

<view class="title">键盘右下角按钮显示为搜索</view>

<input class="uni-input" confirm-type="search" placeholder="键盘右下角按钮显示为搜索" />

</view>

<view class="uni-form-item uni-column">

<view class="title">控制最大输入长度的input</view>

<input class="uni-input" maxlength="10" placeholder="最大输入长度为10" />

</view>

<view class="uni-form-item uni-column">

<view class="title">实时获取输入值：{{inputValue}}</view>

<input class="uni-input" @input="onKeyInput" placeholder="输入同步到view中" />

</view>

<view class="uni-form-item uni-column">

<view class="title">控制输入的input</view>

<input class="uni-input" @input="replaceInput" v-model="changeValue" placeholder="连续的两个1会变成2" />

</view>

<!-- #ifndef MP-BAIDU -->

<view class="uni-form-item uni-column">

<view class="title">控制键盘的input</view>

<input class="uni-input" ref="input1" @input="hideKeyboard" placeholder="输入123自动收起键盘" />

</view>

<!-- #endif -->

<view class="uni-form-item uni-column">

<view class="title">数字输入的input</view>

<input class="uni-input" type="number" placeholder="这是一个数字输入框" />

</view>

<view class="uni-form-item uni-column">

<view class="title">密码输入的input</view>

<input class="uni-input" password type="text" placeholder="这是一个密码输入框" />

</view>

<view class="uni-form-item uni-column">

<view class="title">带小数点的input</view>

<input class="uni-input" type="digit" placeholder="带小数点的数字键盘" />

</view>

<view class="uni-form-item uni-column">

<view class="title">身份证输入的input</view>

<input class="uni-input" type="idcard" placeholder="身份证输入键盘" />

</view>

<view class="uni-form-item uni-column">

<view class="title">控制占位符颜色的input</view>

<input class="uni-input" placeholder-style="color:#F76260" placeholder="占位符字体是红色的" />

</view>

</view>

</view>

</template>

export default {

data() {

return {

title: 'input',

focus: false,

inputValue: '',

changeValue: ''

}

},

methods: {

onKeyInput: function(event) {

this.inputValue = event.target.value

},

replaceInput: function(event) {

var value = event.target.value;

if (value === '11') {

this.changeValue = '2';

}

},

hideKeyboard: function(event) {

if (event.target.value === '123') {

uni.hideKeyboard();

}

}

}

}



#### [label](https://uniapp.dcloud.io/component/label?id=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>。

**属性说明**

| **属性名** | **类型** | **说明** |
| --- | --- | --- |
| for | String | 绑定控件的 id |

**注：**

* app-nvue平台 暂不支持for属性

**示例** [查看演示](https://uniapp.dcloud.io/h5/pages/component/label/label)

<template>

<view>

<view class="uni-common-mt">

<view class="uni-form-item uni-column">

<view class="title">表单组件在label内</view>

<checkbox-group class="uni-list" @change="checkboxChange">

<label class="uni-list-cell uni-list-cell-pd" v-for="item in checkboxItems" :key="item.name">

<view>

<checkbox :value="item.name" :checked="item.checked"></checkbox>

</view>

<view>{{item.value}}</view>

</label>

</checkbox-group>

</view>

<view class="uni-form-item uni-column">

<view class="title">label用for标识表单组件</view>

<radio-group class="uni-list" @change="radioChange">

<label class="uni-list-cell uni-list-cell-pd" v-for="(item,index) in radioItems" :key="index">

<view>

<radio :id="item.name" :value="item.name" :checked="item.checked"></radio>

</view>

<view>

<label class="label-2-text" :for="item.name">

<text>{{item.value}}</text>

</label>

</view>

</label>

</radio-group>

</view>

<view class="uni-form-item uni-column">

<view class="title">label内有多个时选中第一个</view>

<checkbox-group class="uni-list" @change="checkboxChange">

<label class="label-3">

<view class="uni-list-cell uni-list-cell-pd">

<checkbox class="checkbox-3">选项一</checkbox>

</view>

<view class="uni-list-cell uni-list-cell-pd">

<checkbox class="checkbox-3">选项二</checkbox>

</view>

<view class="uni-link uni-center" style="margin-top:20rpx;">点击该label下的文字默认选中第一个checkbox</view>

</label>

</checkbox-group>

</view>

</view>

</view>

</template>

export default {

data() {

return {

checkboxItems: [{

name: 'USA',

value: '美国'

},

{

name: 'CHN',

value: '中国',

checked: 'true'

}

],

radioItems: [{

name: 'USA',

value: '美国'

},

{

name: 'CHN',

value: '中国',

checked: 'true'

}

],

hidden: false

}

},

methods: {

checkboxChange: function(e) {

console.log(e)

var checked = e.target.value

var changed = {}

for (var i = 0; i < this.checkboxItems.length; i++) {

if (checked.indexOf(this.checkboxItems[i].name) !== -1) {

changed['checkboxItems[' + i + '].checked'] = true

} else {

changed['checkboxItems[' + i + '].checked'] = false

}

}

},

radioChange: function(e) {

var checked = e.target.value

var changed = {}

for (var i = 0; i < this.radioItems.length; i++) {

if (checked.indexOf(this.radioItems[i].name) !== -1) {

changed['radioItems[' + i + '].checked'] = true

} else {

changed['radioItems[' + i + '].checked'] = false

}

}

}

}

}



#### [picker](https://uniapp.dcloud.io/component/picker?id=picker)

从底部弹起的滚动选择器。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

**属性说明**

| **属性名** | **类型** | **默认值** | **说明** |
| --- | --- | --- | --- |
| range | Array / Array＜Object＞ | [] | mode为 selector 或 multiSelector 时，range 有效 |
| range-key | String |  | 当 range 是一个 Array＜Object＞ 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） |
| @change | EventHandle |  | value 改变时触发 change 事件，event.detail = {value: value} |
| disabled | Boolean | false | 是否禁用 |
| @cancel | EventHandle |  | 取消选择或点遮罩层收起 picker 时触发 |

* picker在各平台的实现是有UI差异的，有的平台如百度、支付宝小程序的Android端是从中间弹出的；有的平台支持循环滚动如微信、百度小程序；有的平台没有取消按钮如App端。但均不影响功能使用。

**多列选择器**

mode = multiSelector

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| vue支持，nvue自2.4起支持 | √ | √ | x | √ | √ | √ |

支付宝小程序 picker 组件不支持多列选择，可以使用 picker-view 组件替代。

**属性说明**

| **属性名** | **类型** | **默认值** | **说明** |
| --- | --- | --- | --- |
| range | 二维 Array / 二维 Array＜Object＞ | [] | 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 Array＜Object＞ 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） |
| @change | EventHandle |  | value 改变时触发 change 事件，event.detail = {value: value} |
| @columnchange | 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 |
| @cancel | EventHandle |  | 取消选择时触发 |
| disabled | Boolean | false | 是否禁用 |

**bug & tips**

* 由于 JavaScript 的限制 vue 不能观测如下方式设置 value：this.value[0] = 0 （[vue 注意事项](https://cn.vuejs.org/v2/guide/list.html#%E6%B3%A8%E6%84%8F%E4%BA%8B%E9%A1%B9)），解决方式参考：[hello-uniapp 示例](https://github.com/dcloudio/hello-uniapp/commit/59264474172a591c865431d02a2a1e3583978827)
* 微信开发工具的pc模拟器有可能出现拖动数据错乱，使用真机正常

**时间选择器**

mode = time

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | √ | √ | √ | √ | √ | √ |

* 时间选择在App端调用的是os的原生时间选择控件，在不同平台有不同的ui表现

**属性说明**

| **属性名** | **类型** | **默认值** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| 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" | App 不支持 |
| end | String |  | 表示有效时间范围的结束，字符串格式为"hh:mm" | App 不支持 |
| @change | EventHandle |  | value 改变时触发 change 事件，event.detail = {value: value} |  |
| @cancel | EventHandle |  | 取消选择时触发 |  |
| disabled | Boolean | false | 是否禁用 |  |

**日期选择器**

mode = date

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | √ | √ | √ | √ | √ | √ |

* 日期选择在App端调用的是os的原生日期选择控件，在不同平台有不同的ui表现。

**属性说明**

| **属性名** | **类型** | **默认值** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| 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 | H5、微信小程序、百度小程序、头条小程序 |
| @change | EventHandle |  | value 改变时触发 change 事件，event.detail = {value: value} |  |
| @cancel | EventHandle |  | 取消选择时触发 |  |
| 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

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x | x | √ | x | √ | √ | √ |

* 因省市区选择器包含大量数据，占用体积，并非所有应用都需要，且很多城市数据有自维护需求，所以在App和H5平台没有内置。可以基于多列picker或picker-view，自行填充城市数据，插件市场有较多类似插件，[详见](https://ext.dcloud.net.cn/search?q=%E5%9F%8E%E5%B8%82%E9%80%89%E6%8B%A9)。注意基于多列picker方式的地区选择不能运行在支付宝小程序上，只有基于picker-view的可以全端运行。这些插件中，比较推荐的是[SimpleJalon的地址联动选择插件](https://ext.dcloud.net.cn/plugin?id=1084)，它可以同时兼容app-nvue、app-vue、h5、及各端小程序。

| **属性名** | **类型** | **默认值** | **说明** |
| --- | --- | --- | --- |
| 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 |
| @change | EventHandle |  | value 改变时触发 change 事件，event.detail = {value: value} |
| @cancel | EventHandle |  | 取消选择时触发 |
| disabled | Boolean | false | 是否禁用 |

**示例** [查看演示](https://uniapp.dcloud.io/h5/pages/component/picker/picker)

<template>

<view>

<view class="uni-title uni-common-pl">地区选择器</view>

<view class="uni-list">

<view class="uni-list-cell">

<view class="uni-list-cell-left">

当前选择

</view>

<view class="uni-list-cell-db">

<picker @change="bindPickerChange" :value="index" :range="array">

<view class="uni-input">{{array[index]}}</view>

</picker>

</view>

</view>

</view>

<view class="uni-title uni-common-pl">时间选择器</view>

<view class="uni-list">

<view class="uni-list-cell">

<view class="uni-list-cell-left">

当前选择

</view>

<view class="uni-list-cell-db">

<picker mode="time" :value="time" start="09:01" end="21:01" @change="bindTimeChange">

<view class="uni-input">{{time}}</view>

</picker>

</view>

</view>

</view>

<view class="uni-title uni-common-pl">日期选择器</view>

<view class="uni-list">

<view class="uni-list-cell">

<view class="uni-list-cell-left">

当前选择

</view>

<view class="uni-list-cell-db">

<picker mode="date" :value="date" :start="startDate" :end="endDate" @change="bindDateChange">

<view class="uni-input">{{date}}</view>

</picker>

</view>

</view>

</view>

</view>

</template>

export default {

data() {

const currentDate = this.getDate({

format: true

})

return {

title: 'picker',

array: ['中国', '美国', '巴西', '日本'],

index: 0,

date: currentDate,

time: '12:01'

}

},

computed: {

startDate() {

return this.getDate('start');

},

endDate() {

return this.getDate('end');

}

},

methods: {

bindPickerChange: function(e) {

console.log('picker发送选择改变，携带值为', e.target.value)

this.index = e.target.value

},

bindDateChange: function(e) {

this.date = e.target.value

},

bindTimeChange: function(e) {

this.time = e.target.value

},

getDate(type) {

const date = new Date();

let year = date.getFullYear();

let month = date.getMonth() + 1;

let day = date.getDate();

if (type === 'start') {

year = year - 60;

} else if (type === 'end') {

year = year + 2;

}

month = month > 9 ? month : '0' + month;;

day = day > 9 ? day : '0' + day;

return `${year}-${month}-${day}`;

}

}

}



示例代码说明：以上示例代码从hello uni-app示例中复制，涉及的css样式在hello uni-app的app.vue和uni.css中

预览H5效果：使用浏览器的手机模式访问<https://uniapp.dcloud.io/h5/pages/component/picker/picker>

**注意**

* 在picker内容还在滚动时或滚动回弹动画还未结束时，点确定关闭弹出的picker，数据无法及时更新。需等待一下，或手动触停滚动再点确定。所有平台均如此
* 如果需要在PC端使用picker，需注意pc端没有touchmove事件，可以配置[H5模版](https://uniapp.dcloud.io/collocation/manifest?id=h5-template)，并引入[touch-emulator.js](https://github.com/dcloudio/touchemulator)来解决。hello uni-app的pc版也是使用了这个方案。

#### [picker-view](https://uniapp.dcloud.io/component/picker-view?id=picker-view)

嵌入页面的滚动选择器

**属性说明**

| **属性名** | **类型** | **默认值** | **平台差异说明** |
| --- | --- | --- | --- |
| value | Array＜Number＞ | 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 | app-nvue和头条小程序不支持 |
| mask-style | String | 设置蒙层的样式 |  |
| mask-class | String | 设置蒙层的类名 | app-nvue和头条小程序不支持 |
| @change | 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

<picker-view /> 的子组件，仅可放置于 <picker-view /> 中，其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** [查看演示](https://uniapp.dcloud.io/h5/pages/component/picker-view/picker-view)

<template>

<view>

<view class="uni-padding-wrap">

<view class="uni-title">日期：{{year}}年{{month}}月{{day}}日</view>

</view>

<picker-view v-if="visible" :indicator-style="indicatorStyle" :value="value" @change="bindChange">

<picker-view-column>

<view class="item" v-for="(item,index) in years" :key="index">{{item}}年</view>

</picker-view-column>

<picker-view-column>

<view class="item" v-for="(item,index) in months" :key="index">{{item}}月</view>

</picker-view-column>

<picker-view-column>

<view class="item" v-for="(item,index) in days" :key="index">{{item}}日</view>

</picker-view-column>

</picker-view>

</view>

</template>

export default {

data: function () {

const date = new Date()

const years = []

const year = date.getFullYear()

const months = []

const month = date.getMonth() + 1

const days = []

const day = date.getDate()

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)

}

return {

title: 'picker-view',

years,

year,

months,

month,

days,

day,

value: [9999, month - 1, day - 1],

visible: true,

indicatorStyle: `height: ${Math.round(uni.getSystemInfoSync().screenWidth/(750/100))}px;`

}

},

methods: {

bindChange: function (e) {

const val = e.detail.value

this.year = this.years[val[0]]

this.month = this.months[val[1]]

this.day = this.days[val[2]]

}

}

}



**Tips**

* 微信小程序端，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
* 如果需要在PC端使用picker-view，可以配置[H5模版](https://uniapp.dcloud.io/collocation/manifest?id=h5-template)，并引入[touch-emulator.js](https://github.com/dcloudio/touchemulator)

#### [radio-group](https://uniapp.dcloud.io/component/radio?id=radio-group)

单项选择器，内部由多个 <radio> 组成。

**属性说明**

| **属性名** | **类型** | **默认值** | **说明** |
| --- | --- | --- | --- |
| @change | 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](https://uniapp.dcloud.io/component/radio?id=radio)

单选项目。

**属性说明**

| **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** [查看演示](https://uniapp.dcloud.io/h5/pages/component/radio/radio)

<template>

<view>

<view class="uni-padding-wrap">

<view class="uni-title">默认样式</view>

<view>

<label class="radio"><radio value="r1" checked="true" />选中</label>

<label class="radio"><radio value="r2" />未选中</label>

</view>

</view>

<view class="uni-title uni-common-mt uni-common-pl">推荐展示样式</view>

<view class="uni-list">

<radio-group @change="radioChange">

<label class="uni-list-cell uni-list-cell-pd" v-for="(item, index) in items" :key="item.value">

<view>

<radio :value="item.value" :checked="index === current" />

</view>

<view>{{item.name}}</view>

</label>

</radio-group>

</view>

</view>

</template>

export default {

data() {

return {

items: [{

value: 'USA',

name: '美国'

},

{

value: 'CHN',

name: '中国',

checked: 'true'

},

{

value: 'BRA',

name: '巴西'

},

{

value: 'JPN',

name: '日本'

},

{

value: 'ENG',

name: '英国'

},

{

value: 'FRA',

name: '法国'

},

],

current: 0

}

},

methods: {

radioChange: function(evt) {

for (let i = 0; i < this.items.length; i++) {

if (this.items[i].value === evt.target.value) {

this.current = i;

break;

}

}

}

}

}



示例代码说明：以上示例代码从hello uni-app示例中复制，涉及的css样式在hello uni-app的app.vue和uni.css中

预览H5效果：使用浏览器的手机模式访问<https://uniapp.dcloud.io/h5/pages/component/radio/radio>

**注意**

* radio的默认颜色，在不同平台不一样。微信小程序是绿色的，头条小程序为红色，其他平台是蓝色的。更改颜色使用color属性。
* 如需调节radio大小，可通过css的scale方法调节，如缩小到70%style="transform:scale(0.7)"
* radio不是checkbox，点击一个已经选中的radio，不会将其取消选中

#### [slider](https://uniapp.dcloud.io/component/slider?id=slider)

滑动选择器。

**属性说明**

| **属性名** | **类型** | **默认值** | **说明** |
| --- | --- | --- | --- |
| 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 |
| activeColor | Color | 各个平台不同，详见下 | 滑块左侧已选择部分的线条颜色 |
| backgroundColor | Color | #e9e9e9 | 滑块右侧背景条的颜色 |
| block-size | Number | 28 | 滑块的大小，取值范围为 12 - 28 |
| block-color | Color | #ffffff | 滑块的颜色 |
| show-value | Boolean | false | 是否显示当前 value |
| @change | EventHandle |  | 完成一次拖动后触发的事件，event.detail = {value: value} |
| @changing | EventHandle |  | 拖动过程中触发的事件，event.detail = {value: value} |

**Tips**

* activeColor默认值在不同平台不一样，微信是绿色(#1aad19)，头条是红色，其他平台是蓝色
* 如需要区间滑块，即一根横条上使用2个滑块选择一段范围，可见[插件市场](https://ext.dcloud.net.cn/search?q=%E5%8C%BA%E9%97%B4%E6%BB%91%E5%9D%97)

**示例** [查看演示](https://uniapp.dcloud.io/h5/pages/component/slider/slider)

<template>

<view>

<view class="uni-padding-wrap uni-common-mt">

<view class="uni-title">设置step</view>

<view>

<slider value="60" @change="sliderChange" step="5" />

</view>

<view class="uni-title">显示当前value</view>

<view>

<slider value="50" @change="sliderChange" show-value />

</view>

<view class="uni-title">设置最小/最大值</view>

<view>

<slider value="100" @change="sliderChange" min="50" max="200" show-value />

</view>

<view class="uni-title">不同颜色和大小的滑块</view>

<view>

<slider value="50" @change="sliderChange" activeColor="#FFCC33" backgroundColor="#000000" block-color="#8A6DE9" block-size="20" />

</view>

</view>

</view>

</template>

export default {

data() {

return {}

},

methods: {

sliderChange(e) {

console.log('value 发生变化：' + e.detail.value)

}

}

}



示例代码说明：以上示例代码从hello uni-app示例中复制，涉及的css样式在hello uni-app的app.vue和uni.css中

预览H5效果：使用浏览器的手机模式访问<https://uniapp.dcloud.io/h5/pages/component/slider/slider>

**Tips**

* 如果需要在PC端使用slider，可以配置[H5模版](https://uniapp.dcloud.io/collocation/manifest?id=h5-template)，并引入[touch-emulator.js](https://github.com/dcloudio/touchemulator)

#### [switch](https://uniapp.dcloud.io/component/switch?id=switch)

开关选择器。

**属性说明**

| **属性名** | **类型** | **默认值** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| checked | Boolean | false | 是否选中 |  |
| disabled | Boolean | false | 是否禁用 | 头条小程序不支持 |
| type | String | switch | 样式，有效值：switch, checkbox |  |
| @change | 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** [查看演示](https://uniapp.dcloud.io/h5/pages/component/switch/switch)

<template>

<view>

<view class="uni-padding-wrap uni-common-mt">

<view class="uni-title">默认样式</view>

<view>

<switch checked @change="switch1Change" />

<switch @change="switch2Change" />

</view>

<view class="uni-title">推荐展示样式</view>

</view>

<view class="uni-list">

<view class="uni-list-cell uni-list-cell-pd">

<view class="uni-list-cell-db">开启中</view>

<switch checked />

</view>

<view class="uni-list-cell uni-list-cell-pd">

<view class="uni-list-cell-db">关闭</view>

<switch />

</view>

</view>

</view>

</template>

export default {

data() {

return {}

},

methods: {

switch1Change: function (e) {

console.log('switch1 发生 change 事件，携带值为', e.target.value)

},

switch2Change: function (e) {

console.log('switch2 发生 change 事件，携带值为', e.target.value)

}

}

}



示例代码说明：以上示例代码从hello uni-app示例中复制，涉及的css样式在hello uni-app的app.vue和uni.css中

预览H5效果：使用浏览器的手机模式访问<https://uniapp.dcloud.io/h5/pages/component/switch/switch>

**注意**

* switch的默认颜色，在不同平台不一样。微信小程序是绿色的，头条小程序为红色，其他平台是蓝色的。更改颜色使用color属性。
* 如需调节switch大小，可通过css的scale方法调节，如缩小到70%style="transform:scale(0.7)"

#### [textarea](https://uniapp.dcloud.io/component/textarea?id=textarea)

多行输入框。

**属性说明**

| **属性名** | **类型** | **默认值** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| 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 | 头条小程序不支持 |
| 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 |  |
| focus | Boolean | false | 获取焦点 | 在 H5 平台能否聚焦以及软键盘是否跟随弹出，取决于当前浏览器本身的实现。 |
| 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 | 微信小程序、百度小程序、头条小程序、QQ小程序 |
| 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 | App、微信小程序、百度小程序、头条小程序、QQ小程序 |
| cursor | Number |  | 指定focus时的光标位置 | 微信小程序、App、H5、百度小程序、头条小程序、QQ小程序 |
| show-confirm-bar | Boolean | true | 是否显示键盘上方带有”完成“按钮那一栏 | 微信小程序、百度小程序、QQ小程序 |
| selection-start | Number | -1 | 光标起始位置，自动聚焦时有效，需与selection-end搭配使用 | 微信小程序、App、H5、百度小程序、头条小程序、QQ小程序 |
| selection-end | Number | -1 | 光标结束位置，自动聚焦时有效，需与selection-start搭配使用 | 微信小程序、App、H5、百度小程序、头条小程序、QQ小程序 |
| adjust-position | Boolean | true | 键盘弹起时，是否自动上推页面 | App、微信小程序、百度小程序、QQ小程序 |
| @focus | EventHandle |  | 输入框聚焦时触发，event.detail = { value, height }，height 为键盘高度 | 仅微信小程序、App（HBuilderX 2.0+ [nvue uni-app模式](http://ask.dcloud.net.cn/article/36074" \t "_blank)） 、QQ小程序支持 height |
| @blur | EventHandle |  | 输入框失去焦点时触发，event.detail = {value, cursor} |  |
| @linechange | EventHandle |  | 输入框行数变化时调用，event.detail = {height: 0, heightRpx: 0, lineCount: 0} | 头条小程序不支持 |
| @input | EventHandle |  | 当键盘输入时，触发 input 事件，event.detail = {value, cursor}， @input 处理函数的返回值并不会反映到 textarea 上 |  |
| @confirm | EventHandle |  | 点击完成时， 触发 confirm 事件，event.detail = {value: value} | 微信小程序、百度小程序、QQ小程序 |

**示例** [查看示例](https://uniapp.dcloud.io/h5/pages/component/textarea/textarea)

<template>

<view>

<view class="uni-title uni-common-pl">输入区域高度自适应，不会出现滚动条</view>

<view class="uni-textarea">

<textarea @blur="bindTextAreaBlur" auto-height />

</view>

<view class="uni-title uni-common-pl">占位符字体是红色的textarea</view>

<view class="uni-textarea">

<textarea placeholder-style="color:#F76260" placeholder="占位符字体是红色的"/>

</view>

</view>

</template>

export default {

data() {

return {}

},

methods: {

bindTextAreaBlur: function (e) {

console.log(e.detail.value)

}

}

}



**Tips**

* 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 @submit。
* input 事件处理函数内实时修改当前值不生效，可以延迟设置，例如：setTimeout(() => { this.value = 100 }, 0)。
* 微信小程序、百度小程序、头条小程序中，textarea是原生组件，层级高于前端组件，请勿在 scroll-view、swiper、picker-view、movable-view 中使用 textarea 组件。覆盖textarea需要使用cover-view。[详见](https://uniapp.dcloud.io/component/native-component)
* 小程序端 css 动画对 textarea 组件无效。
* H5 平台只能在用户交互时修改 focus 生效。
* 如果遇到 focus 属性设置不生效的问题参考：[组件属性设置不生效解决办法](https://uniapp.dcloud.io/use?id=%e5%b8%b8%e8%a7%81%e9%97%ae%e9%a2%98)
* 软键盘的弹出和收起逻辑，详见[input的文档](https://uniapp.dcloud.io/component/input?id=app%e5%b9%b3%e5%8f%b0ios%e7%ab%af%e8%bd%af%e9%94%ae%e7%9b%98%e4%b8%8a%e6%96%b9%e6%a8%aa%e6%9d%a1%e5%8e%bb%e9%99%a4%e6%96%b9%e6%a1%88)

**富文本编辑的解决方案** 在输入框里图文混排内容，在web上该功能依赖document，而小程序和app的正常页面又没有document。

* 方式一：使用uni-app自带的editor组件，该组件支持App、H5、微信小程序，其他家小程序自身未提供这类解决方案。
* 方式二：采用markdown编辑器方案，输入区输入markdown语法，预览区提供预览。这种方式可以跨端。插件市场搜[富文本编辑](https://ext.dcloud.net.cn/search?q=%E5%AF%8C%E6%96%87%E6%9C%AC%E7%BC%96%E8%BE%91" \t "_blank)，有不少插件。
* 方式三：使用web-view组件，挂在html页面，此时可使用web中常见的各种富文本编辑器，插件市场也有这类插件。

# 导航

#### [navigator](https://uniapp.dcloud.io/component/navigator?id=navigator)

页面链接。

**属性说明**

| **属性名** | **类型** | **默认值** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| url | String |  | 应用内的跳转链接，值为相对路径或绝对路径，如："../first/first"，"/pages/first/first"，注意不能加 .vue 后缀 |  |
| 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 |  |
| animation-type | String | pop-in/out | 当 open-type 为 navigate、navigateBack 时有效，窗口的显示/关闭动画效果，详见：[窗口动画](https://uniapp.dcloud.io/api/router?id=animation) | App |
| animation-duration | Number | 300 | 当 open-type 为 navigate、navigateBack 时有效，窗口显示/关闭动画的持续时间。 | App |
| 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 | 微信小程序 |
| 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 |  |
| target | String | self | 在哪个小程序目标上发生跳转，默认当前小程序，值域self/miniProgram | 微信2.0.7+、百度2.5.2+、QQ |

**open-type 有效值**

| **值** | **说明** | **平台差异说明** |
| --- | --- | --- |
| navigate | 对应 uni.navigateTo 的功能 |  |
| redirect | 对应 uni.redirectTo 的功能 |  |
| switchTab | 对应 uni.switchTab 的功能 |  |
| reLaunch | 对应 uni.reLaunch 的功能 | 头条小程序不支持 |
| navigateBack | 对应 uni.navigateBack 的功能 |  |
| exit | 退出小程序，target="miniProgram"时生效 | 微信2.1.0+、百度2.5.2+、QQ1.4.7+ |

**注意**

* 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。\*\*
* app-nvue 平台只有纯nvue项目（render为native）才支持 <navigator>。非render为native的情况下，nvue暂不支持navigator组件，请使用API跳转。
* app下退出应用，Android平台可以使用[plus.runtime.quit](https://www.html5plus.org/doc/zh_cn/runtime.html#plus.runtime.quit)

**示例** [查看示例](https://uniapp.dcloud.io/h5/pages/component/navigator/navigator)

<template>

<view>

<view class="page-body">

<view class="btn-area">

<navigator url="navigate/navigate?title=navigate" hover-class="navigator-hover">

<button type="default">跳转到新页面</button>

</navigator>

<navigator url="redirect/redirect?title=redirect" open-type="redirect" hover-class="other-navigator-hover">

<button type="default">在当前页打开</button>

</navigator>

<navigator url="/pages/tabBar/extUI/extUI" open-type="switchTab" hover-class="other-navigator-hover">

<button type="default">跳转tab页面</button>

</navigator>

</view>

</view>

</view>

</template>

// navigate.vue页面接受参数

export default {

onLoad: function (option) { //option为object类型，会序列化上个页面传递的参数

console.log(option.id); //打印出上个页面传递的参数。

console.log(option.name); //打印出上个页面传递的参数。

}

}

url有长度限制，太长的字符串会传递失败，可使用[窗体通信](https://uniapp.dcloud.io/collocation/frame/communication)、[全局变量](https://ask.dcloud.net.cn/article/35021)，或encodeURIComponent等多种方式解决，如下为encodeURIComponent示例。

<navigator :url="'/pages/navigate/navigate?item='+ encodeURIComponent(JSON.stringify(item))"></navigator>

// navigate.vue页面接受参数

onLoad: function (option) {

const item = JSON.parse(decodeURIComponent(option.item));

}

**注意**

* 跳转tabbar页面，必须设置open-type="switchTab"

# 媒体组件

#### [audio](https://uniapp.dcloud.io/component/audio?id=audio)

音频。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | √ | √ | x | √ | x | x |

**注意：** 微信小程序平台自基础库 1.6.0 版本开始，不再维护 audio 组件，推荐使用API方式而不是组件方式来播放音频。API见 [uni.createInnerAudioContext](https://uniapp.dcloud.io/api/media/audio-context?id=createinneraudiocontext) 替代。

app-nvue也不支持此组件。

**属性说明**

| **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 |
| @error | EventHandle |  | 当发生错误时触发 error 事件，detail = {errMsg: MediaError.code} |
| @play | EventHandle |  | 当开始/继续播放时触发play事件 |
| @pause | EventHandle |  | 当暂停播放时触发 pause 事件 |
| @timeupdate | EventHandle |  | 当播放进度改变时触发 timeupdate 事件，detail = {currentTime, duration} |
| @ended | 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：** [查看示例](https://uniapp.dcloud.io/h5/pages/component/audio/audio)

<template>

<view>

<view class="page-body">

<view class="page-section page-section-gap" style="text-align: center;">

<audio style="text-align: left" :src="current.src" :poster="current.poster" :name="current.name" :author="current.author" :action="audioAction" controls></audio>

</view>

</view>

</view>

</template>

export default {

data() {

return {

current: {

poster: 'https://img-cdn-qiniu.dcloud.net.cn/uniapp/audio/music.jpg',

name: '致爱丽丝',

author: '暂无',

src: 'https://img-cdn-qiniu.dcloud.net.cn/uniapp/audio/music.mp3',

},

audioAction: {

method: 'pause'

}

}

}

}



#### [camera](https://uniapp.dcloud.io/component/camera?id=camera)

页面内嵌的区域相机组件。注意这不是点击后全屏打开的相机。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x | x | √ | x | √ | x | √ |

* 在 App 和 H5 端，可以使用API方式来调用全屏摄像头，而不是组件内嵌方式。并且App端和H5端目前无法通过cover-view覆盖摄像头，有相关需求见下方插件。
  + [uni.chooseImage](https://uniapp.dcloud.io/api/media/image?id=chooseimage) 和 [uni.chooseVideo](https://uniapp.dcloud.io/api/media/video?id=choosevideo) 可以拍照或录像，但它们会同时允许用户选择相册。
  + 使用plus.camera的API可以直接调起全屏的摄像头，详见：[https://www.html5plus.org/doc/zh\_cn/camera.html](http://www.html5plus.org/doc/zh_cn/camera.html) 调用原生摄像头进行相关操作。
* 如开发身份证扫描、银行卡识别等ocr识别需求，在微信小程序和百度小程序中使用本camera组件，将图片发送给服务器识别，插件市场有封装好的[模板](https://ext.dcloud.net.cn/search?q=%E5%B0%8F%E7%A8%8B%E5%BA%8F%E7%9B%B8%E6%9C%BA)；在App端使用[原生插件](https://ext.dcloud.net.cn/search?q=ocr)
* 活体检测、人脸识别另见文档[生物认证](https://uniapp.dcloud.io/api/other/authentication)
* app-nvue下支持barcode组件，可实现自定义扫码。[参考](https://uniapp.dcloud.io/component/barcode)

**属性说明**

| **属性名** | **类型** | **默认值** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| mode | String | normal | 有效值为 normal, scanCode | 微信小程序 |
| device-position | String | back | 前置或后置，值为front, back |  |
| flash | String | auto | 闪光灯，值为auto, on, off |  |
| @stop | EventHandle |  | 摄像头在非正常终止时触发，如退出后台等情况 |  |
| @error | EventHandle |  | 用户不允许使用摄像头时触发 |  |
| @scancode | EventHandle |  | 在扫码识别成功时触发，仅在 mode="scanCode" 时生效 | 微信小程序 |

**Tips：**

* 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。
* 请勿在 scroll-view、swiper、picker-view、movable-view 中使用 camera 组件。
* 同一页面只能插入一个 camera 组件。
* 相关API：[createCameraContext](https://uniapp.dcloud.io/api/media/camera-context)

**代码示例**

<template>

<view>

<camera device-position="back" flash="off" @error="error" style="width: 100%; height: 300px;"></camera>

<button type="primary" @click="takePhoto">拍照</button>

<view>预览</view>

<image mode="widthFix" :src="src"></image>

</view>

</template>

export default {

data() {

return {

src:""

}

},

methods: {

takePhoto() {

const ctx = uni.createCameraContext();

ctx.takePhoto({

quality: 'high',

success: (res) => {

this.src = res.tempImagePath

}

});

},

error(e) {

console.log(e.detail);

}

}

}

#### [image](https://uniapp.dcloud.io/component/image?id=image)

图片。

| **属性名** | **类型** | **默认值** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| 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 | 微信小程序、App、百度小程序、头条小程序 |
| fade-show | Boolean | true | 图片显示动画效果 | 仅App-nvue 2.3.4+ Android有效 |
| @error | HandleEvent |  | 当错误发生时，发布到 AppService 的事件名，事件对象event.detail = {errMsg: 'something wrong'} |  |
| @load | 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'} |  |

**Tips**

* <image> 组件默认宽度 300px、高度 225px；app-nvue平台，暂时默认为屏幕宽度
* src 仅支持相对路径、绝对路径，支持 base64 码；
* 页面结构复杂，css样式太多的情况，使用 image 可能导致样式生效较慢，出现 “闪一下” 的情况，此时设置 image{will-change: transform} ,可优化此问题。
* 自定义组件里面使用 <image>时，若 src 使用相对路径可能出现路径查找失败的情况，故建议使用绝对路径。
* webp格式的图片，app-vue下，iOS不支持，Android支持；app-nvue下，iOS和Android均支持。app-vue下也支持gif。

**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：** [查看示例](https://uniapp.dcloud.io/h5/pages/component/image/image)

<template>

<view class="page">

<view class="image-list">

<view class="image-item" v-for="(item,index) in array" :key="index">

<view class="image-content">

<image style="width: 200px; height: 200px; background-color: #eeeeee;" :mode="item.mode" :src="src"

@error="imageError"></image>

</view>

<view class="image-title">{{item.text}}</view>

</view>

</view>

</view>

</template>

export default {

data() {

return {

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: 'https://img-cdn-qiniu.dcloud.net.cn/uniapp/images/shuijiao.jpg'

}

},

methods: {

imageError: function(e) {

console.error('image发生error事件，携带值为' + e.detail.errMsg)

}

}

}

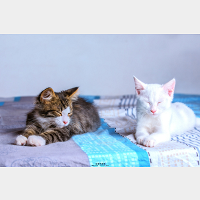
原图



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



#### [video](https://uniapp.dcloud.io/component/video?id=video)

视频。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | √ | √ | 基础库1.10.0+ | √ | √ | √ |

**属性说明**

| **属性名** | **类型** | **默认值** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| src | String |  | 要播放视频的资源地址 |  |
| autoplay | Boolean | false | 是否自动播放 |  |
| loop | Boolean | false | 是否循环播放 | 头条小程序不支持 |
| muted | Boolean | false | 是否静音播放 | 头条小程序不支持 |
| initial-time | Number |  | 指定视频初始播放位置，单位为秒（s）。 | 头条小程序不支持 |
| duration | Number |  | 指定视频时长，单位为秒（s）。 | 头条小程序不支持 |
| 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） | 头条小程序不支持 |
| danmu-list | Object Array |  | 弹幕列表 | 头条小程序不支持 |
| 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 | 头条小程序不支持 |
| enable-danmu | Boolean | false | 是否展示弹幕，只在初始化时有效，不能动态变更 | 头条小程序不支持 |
| page-gesture | Boolean | false | 在非全屏模式下，是否开启亮度与音量调节手势 | 微信小程序、H5 |
| 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） | H5和头条小程序不支持 |
| show-progress | Boolean | true | 若不设置，宽度大于240时才会显示 | 头条小程序不支持 |
| show-fullscreen-btn | Boolean | true | 是否显示全屏按钮 | 头条小程序不支持 |
| show-play-btn | Boolean | true | 是否显示视频底部控制栏的播放按钮 | 头条小程序不支持 |
| show-center-play-btn | Boolean | true | 是否显示视频中间的播放按钮 | 头条小程序不支持 |
| enable-progress-gesture | Boolean | true | 是否开启控制进度的手势 | 头条小程序不支持 |
| 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 | 微信小程序、H5 |
| 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 | 头条小程序不支持 |
| show-mute-btn | Boolean | false | 是否显示静音按钮 | 微信小程序 |
| title | String |  | 视频的标题，全屏时在顶部展示 | 微信小程序 |
| play-btn-position | String | bottom | 播放按钮的位置 | 微信小程序 |
| enable-play-gesture | Boolean | false | 是否开启播放手势，即双击切换播放/暂停 | 微信小程序 |
| auto-pause-if-navigate | Boolean | true | 当跳转到其它小程序页面时，是否自动暂停本页面的视频 | 微信小程序 |
| auto-pause-if-open-native | Boolean | true | 当跳转到其它微信原生页面时，是否自动暂停本页面的视频 | 微信小程序 |
| vslide-gesture | Boolean | false | 在非全屏模式下，是否开启亮度与音量调节手势（同 page-gesture） | 微信小程序 |
| vslide-gesture-in-fullscreen | Boolean | true | 在全屏模式下，是否开启亮度与音量调节手势 | 微信小程序 |
| ad-unit-id | String |  | 视频前贴广告单元ID，更多详情可参考开放能力[视频前贴广告](https://developers.weixin.qq.com/miniprogram/dev/framework/open-ability/ad/video-patch-ad.html) | 微信小程序 |
| poster-for-crawler | String |  | 用于给搜索等场景作为视频封面展示，建议使用无播放 icon 的视频封面图，只支持网络地址 | 微信小程序 |
| @play | EventHandle |  | 当开始/继续播放时触发play事件 | 头条小程序不支持 |
| @pause | EventHandle |  | 当暂停播放时触发 pause 事件 | 头条小程序不支持 |
| @ended | EventHandle |  | 当播放到末尾时触发 ended 事件 | 头条小程序不支持 |
| @timeupdate | EventHandle |  | 播放进度变化时触发，event.detail = {currentTime, duration} 。触发频率 250ms 一次 | 头条小程序不支持 |
| @fullscreenchange | EventHandle |  | 当视频进入和退出全屏时触发，event.detail = {fullScreen, direction}，direction取为 vertical 或 horizontal | 头条小程序不支持 |
| @waiting | EventHandle |  | 视频出现缓冲时触发 | 头条小程序不支持 |
| @error | EventHandle |  | 视频播放出错时触发 | 头条小程序不支持 |
| @progress | EventHandle |  | 加载进度变化时触发，只支持一段加载。event.detail = {buffered}，百分比 | 微信小程序、H5 |
| @loadedmetadata | EventHandle |  | 视频元数据加载完成时触发。event.detail = {width, height, duration} | 微信小程序、H5 |

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

**示例** [查看示例](https://uniapp.dcloud.io/h5/pages/component/video/video)

<template>

<view>

<view class="uni-padding-wrap uni-common-mt">

<view>

<video id="myVideo" src="https://img.cdn.aliyun.dcloud.net.cn/guide/uniapp/%E7%AC%AC1%E8%AE%B2%EF%BC%88uni-app%E4%BA%A7%E5%93%81%E4%BB%8B%E7%BB%8D%EF%BC%89-%20DCloud%E5%AE%98%E6%96%B9%E8%A7%86%E9%A2%91%E6%95%99%E7%A8%8B@20181126.mp4"

@error="videoErrorCallback" :danmu-list="danmuList" enable-danmu danmu-btn controls></video>

</view>

<!-- #ifndef MP-ALIPAY -->

<view class="uni-list uni-common-mt">

<view class="uni-list-cell">

<view>

<view class="uni-label">弹幕内容</view>

</view>

<view class="uni-list-cell-db">

<input v-model="danmuValue" class="uni-input" type="text" placeholder="在此处输入弹幕内容" />

</view>

</view>

</view>

<view class="uni-btn-v">

<button @click="sendDanmu" class="page-body-button">发送弹幕</button>

</view>

<!-- #endif -->

</view>

</view>

</template>

export default {

data() {

return {

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

}

],

danmuValue: ''

}

},

onReady: function(res) {

// #ifndef MP-ALIPAY

this.videoContext = uni.createVideoContext('myVideo')

// #endif

},

methods: {

sendDanmu: function() {

this.videoContext.sendDanmu({

text: this.danmuValue,

color: this.getRandomColor()

});

this.danmuValue = '';

},

videoErrorCallback: function(e) {

uni.showModal({

content: e.target.errMsg,

showCancel: false

})

},

getRandomColor: function() {

const 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)

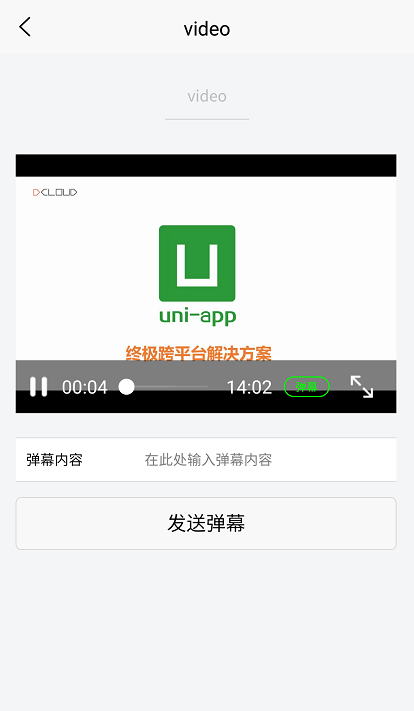
}

return '#' + rgb.join('')

}

}

}



相关api：[uni.createVideoContext](https://uniapp.dcloud.io/api/media/video-context?id=createvideocontext)

**注意**

* 视频播放格式说明：
  + H5平台：支持支持的视频格式视浏览器而定，一般通用的都支持：mp4、webm 和 ogg。（<video/> 组件编译到 H5 时会替换为标准 html 的 video 标签）。H5端也可以自行在条件编译里使用video.js等三方库，这些库可以自动判断环境兼容以决定使用标准video或flash来播放。
  + 小程序平台：各小程序平台支持程度不同，详见各家文档：[微信小程序视频组件文档](https://developers.weixin.qq.com/miniprogram/dev/component/video.html)、[支付宝小程序video组件](https://docs.alipay.com/mini/component/video)、[百度小程序视频组件文档](https://smartprogram.baidu.com/docs/develop/component/media/#video/)、[头条小程序视频组件文档](https://developer.toutiao.com/dev/cn/mini-app/develop/component/media-component/video)
  + App平台： 支持本地视频(mp4/flv)、网络视频地址（mp4/flv/m3u8）及流媒体（rtmp/hls/rtsp）。
* video全屏后，如何自行绘制界面？比如加个标题、加个分享按钮
  + 微信基础库 2.4.0 以上可通过cover-view来绘制界面覆盖元素
  + app端 2.1.5 以上nvue页面的video也可以通过cover-view来绘制界面覆盖元素
  + H5端可通过通用h5做法实现
  + 其他端无法全屏后自行绘制内容
* 如何实现抖音、映客等全屏视频垂直滑动切换效果？
  + 微信基础库 2.4.0 和 app端nvue 2.1.5 以上，可通过在垂直的swiper中内嵌video来实现。原生导航栏设置为custom，视频长宽设为手机屏幕大小，通过cover-view覆盖视频内容。插件市场有相关[示例](https://ext.dcloud.net.cn/search?q=%E6%8A%96%E9%9F%B3)
* <video/> 组件在非H5端是原生组件，层级高于普通前端组件，覆盖其需要使用[cover-view](https://uniapp.dcloud.io/component/cover-view?id=cover-view)组件或plus.nativeObj.view、subNVue。微信基础库 2.4.0 起已支持 video 组件的同层渲染，也就是video在非全屏时，可以被前端元素通过调节zindex来遮挡，但video全屏时，仍需要cover-view覆盖。
* 除微信基础库 2.4.0 和app端nvue页面 2.1.5 以上，其他情况下非H5的video不能放入scroll-view和swiper。注意参考 [原生组件使用限制](https://uniapp.dcloud.io/component/native-component)。
* App平台：使用 <video/> 组件，打包 App 时必须勾选 manifest.json->App 模块权限配置->VideoPlayer 模块。
* App平台：如果使用的视频路径为本地路径，需要配置资源为释放模式：在 manifest.json 文件内 app-plus 节点下新增 runmode 配置，设置值为liberate。
* App平台：如果想使用非原生的video，即原来普通的html5自带video，可使用web-view组件load html页面，在其中使用普通h5 video。
* H5平台： 在部分浏览器中会强制调用原生播放器播放（如：微信内置浏览器、UC浏览器等），在 x5 内核的浏览器中支持配置[同层播放器](https://x5.tencent.com/tbs/guide/video.html)。

#### [live-player](https://uniapp.dcloud.io/component/live-player?id=live-player)

实时音视频播放，也称直播拉流。

使用live-player 组件需注意：如果发布到小程序，需要先通过各家小程序的审核。指定类目的小程序才能使用（[微信小程序类目](https://developers.weixin.qq.com/miniprogram/dev/component/live-player.html)、[百度小程序类目](https://smartprogram.baidu.com/docs/develop/component/media/#live-player/)），审核通过后在各家小程序管理后台自助开通该组件权限。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x(见下) | x | √ | x | √ | √(基础库版本>=1.52.0) | √ |

* App的实时音视频播放，不是使用 live-player，而是直接使用 video 组件。
* H5 下可用 video 播放符合 HTML5 规范的流媒体，rtmp 等非 HTML5 标准的流媒体格式，仅在部分支持 flash 的国内手机浏览器上可播放。在 pc 浏览器上，需要安装 flash 插件才能播放 rtmp 等格式。

**属性说明**

| **属性名** | **类型** | **默认值** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| id | String |  | live-player 属性的唯一标志符 |  |
| src | String |  | 音视频地址。百度小程序支持 m3u8 格式；微信小程序支持 flv, rtmp 格式 |  |
| mode | String | live | live（直播），RTC（实时通话，该模式时延更低） | 微信小程序 |
| 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:contain、fillCrop |  |
| background-mute | Boolean | false | 进入后台时是否静音 |  |
| min-cache | Number | 1 | 最小缓冲区，单位s |  |
| max-cache | Number | 3 | 最大缓冲区，单位s |  |
| @statechange | EventHandle |  | 播放状态变化事件，detail = {code} |  |
| @netstatus | EventHandle |  | 网络状态通知，detail = {info} |  |
| @fullscreenchange | EventHandle |  | 全屏变化事件，detail = {direction, fullScreen}。 |  |

**Tips**

* 百度小程序 iOS 端不支持设置 orientation 属性；
* 微信小程序已废弃 background-mute 属性，默认为进入后台静音；
* live-player 默认宽度 300px、高度 225px；
* live-player 是原生组件，层级高于前端组件，请勿在 scroll-view、swiper、picker-view、movable-view 中使用
* 小程序下覆盖live-player需要使用cover-view。[详见](https://uniapp.dcloud.io/component/native-component)
* live-player 组件相关 JS API：[createLivePlayerContext](https://uniapp.dcloud.io/api/media/live-player-context)
* 小程序平台使用live-player有审核限制，请注意参考各家文档，

**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**

<live-player

src="https://domain/pull\_stream"

autoplay

@statechange="statechange"

@error="error"

style="width: 300px; height: 225px;"

/>

export default {

methods:{

statechange(e){

console.log('live-player code:', e.detail.code)

},

error(e){

console.error('live-player error:', e.detail.errMsg)

}

}

}

#### [live-pusher](https://uniapp.dcloud.io/component/live-pusher?id=live-pusher)

实时音视频录制，也称直播推流。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √(仅nvue) | x | √ | x | x | x | x |

如app平台的vue页面需要支持直播推流，需编写条件编译代码，使用 plus.video.LivePusher，[业务指南](https://ask.dcloud.net.cn/article/13416" \t "_blank)、[规范文档](http://www.html5plus.org/doc/zh_cn/video.html#plus.video.LivePusher)

使用nvue做直播，比使用vue的优势有：

1. nvue可一套代码直接编译到App和微信
2. nvue的cover-view比vue的cover-view更强大，在视频上绘制元素更容易。如果只考虑App端的话，不用cover-view，任意组件都可以覆盖live-pusher组件，因为nvue没有层级问题。
3. 若需要视频内嵌在swiper里上下滑动（类抖音、映客首页模式），App端只有nvue才能实现 当然nvue相比vue的坏处是css写法受限，如果只开发微信小程序，不考虑App，那么使用vue页面也是一样的。

**参数说明**

设置live-pusher组件的推流地址，推流视频模式等。

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| url | string |  | 是 | 推流地址，支持RTMP协议。 |
| mode | string | SD | 否 | 推流视频模式，可取值：SD（标清）, HD（高清）, FHD（超清）。 |
| aspect | string | 3:2 | 否 | 视频宽高比例 |
| muted | Boolean | false | 否 | 是否静音。 |
| enable-camera | Boolean | true | 否 | 开启摄像头。 |
| auto-focus | Boolean | true | 否 | 自动聚集。 |
| beauty | Number | 0 | 否 | 美颜，取值范围 0-9（iOS取值范围为1） ，0 表示关闭。 |
| whiteness | Number | 0 | 否 | 美白，取值范围 0-9（iOS取值范围为1） ，0 表示关闭。 |

<template>

<view>

<live-pusher id='livePusher1' ref="livePusher" class="livePusher" url=""

mode="SD" :muted="true" :enable-camera="true" :auto-focus="true" :beauty="1" whiteness="2"

aspect="9:16" @statechange="statechange" @netstatus="netstatus" @error = "error"

></live-pusher>

<button class="btn" @click="start">开始推流</button>

<button class="btn" @click="pause">暂停推流</button>

<button class="btn" @click="resume">resume</button>

<button class="btn" @click="stop">停止推流</button>

<button class="btn" @click="snapshot">快照</button>

<button class="btn" @click="startPreview">开启摄像头预览</button>

<button class="btn" @click="stopPreview">关闭摄像头预览</button>

<button class="btn" @click="switchCamera">切换摄像头</button>

</view>

</template>

<script>

export default {

data: {

fil: true

},

onReady() {

// 注意：需要在onReady中 或 onLoad 延时

this.context = uni.createLivePusherContext("livePusher", this);

},

methods: {

statechange(e) {

console.log("statechange:" + JSON.stringify(e));

},

netstatus(e) {

console.log("netstatus:" + JSON.stringify(e));

},

error(e) {

console.log("error:" + JSON.stringify(e));

},

start: function() {

this.context.start({

success: (a) => {

console.log("livePusher.start:" + JSON.stringify(a));

}

});

},

close: function() {

this.context.close({

success: (a) => {

console.log("livePusher.close:" + JSON.stringify(a));

}

});

},

snapshot: function() {

this.context.snapshot({

success: (e) => {

console.log(JSON.stringify(e));

}

});

},

resume: function() {

this.context.resume({

success: (a) => {

console.log("livePusher.resume:" + JSON.stringify(a));

}

});

},

pause: function() {

this.context.pause({

success: (a) => {

console.log("livePusher.pause:" + JSON.stringify(a));

}

});

},

stop: function() {

this.context.stop({

success: (a) => {

console.log(JSON.stringify(a));

}

});

},

switchCamera: function() {

this.context.switchCamera({

success: (a) => {

console.log("livePusher.switchCamera:" + JSON.stringify(a));

}

});

},

startPreview: function() {

this.context.startPreview({

success: (a) => {

console.log("livePusher.startPreview:" + JSON.stringify(a));

}

});

},

stopPreview: function() {

this.context.stopPreview({

success: (a) => {

console.log("livePusher.stopPreview:" + JSON.stringify(a));

}

});

}

}

}

</script>

相关api：[uni.createLivePusherContext](https://uniapp.dcloud.io/api/media/live-pusher-context?id=createlivepushercontext)

**注意**

* live-pusher 是原生组件，在小程序端层级高于前端组件，需使用cover-view覆盖。在低版本微信中，live-pusher无法内嵌于 scroll-view、swiper、picker-view、movable-view 中。在App端的nvue文件中，live-pusher没有这类限制。
* App平台：使用 <live-pusher/> 组件，打包 App 时必须勾选 manifest.json->App 模块权限配置->LivePusher(直播推流) 模块。

# 地图

#### [map](https://uniapp.dcloud.io/component/map?id=map)

地图。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | √ | √ | √ | √ | x | x |

**属性说明**

| **属性名** | **类型** | **默认值** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| 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 |  |
| 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 | App-nvue 2.1.5+、微信小程序、H5、百度小程序、支付宝小程序 |
| show-location | Boolean |  | 显示带有方向的当前定位点 | 微信小程序、H5、百度小程序、支付宝小程序 |
| polygons | Array.<polygon> |  | 多边形 | App-nvue 2.1.5+、微信小程序、百度小程序、支付宝小程序 |
| @markertap | EventHandle |  | 点击标记点时触发 | App-nvue 2.3.3+, App平台需要指定 marker 对象属性 id |
| @callouttap | EventHandle |  | 点击标记点对应的气泡时触发 |  |
| @controltap | EventHandle |  | 点击控件时触发 |  |
| @regionchange | EventHandle |  | 视野发生变化时触发 | 微信小程序、H5、百度小程序、支付宝小程序 |
| @tap | EventHandle |  | 点击地图时触发 |  |
| @updated | EventHandle |  | 在地图渲染更新完成时触发 | 微信小程序、H5、百度小程序 |

**注意**

* <map> 组件的宽/高推荐写直接量，比如：750rpx，不要设置百分比值。
* uni-app 只支持 gcj02 坐标

**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** | **平台差异说明** |
| --- | --- | --- | --- | --- | --- |
| id | 标记点id | Number | 是 | 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 | 点击时显示，callout存在时将被忽略 | App-nvue 2.1.5+、微信小程序、H5、支付宝小程序、百度小程序 |
| 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 | App-nvue 2.1.5+、微信小程序、支付宝小程序、百度小程序 |
| alpha | 标注的透明度 | Number | 否 | 默认1，无透明，范围 0 ~ 1 | App-nvue 2.1.5+、微信小程序、支付宝小程序、百度小程序 |
| width | 标注图标宽度 | Number | 否 | 默认为图片实际宽度 | App-nvue 2.1.5+、微信小程序、H5、支付宝小程序、百度小程序 |
| height | 标注图标高度 | Number | 否 | 默认为图片实际高度 | App-nvue 2.1.5+、微信小程序、H5、支付宝小程序、百度小程序 |
| 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。 | App-nvue 2.1.5+ |
| 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。 | App-nvue 2.1.5+、微信小程序、H5、App、百度小程序 |
| 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 | App-nvue 2.1.5+、微信小程序、H5、百度小程序 |

**marker 上的气泡 callout**

| **属性** | **说明** | **类型** | **平台差异说明** |
| --- | --- | --- | --- |
| content | 文本 | String |  |
| color | 文本颜色 | String | App-nvue 2.1.5+、微信小程序、H5、百度小程序 |
| fontSize | 文字大小 | Number | App-nvue 2.1.5+、微信小程序、H5、百度小程序 |
| borderRadius | callout边框圆角 | Number | App-nvue 2.1.5+、微信小程序、H5、百度小程序 |
| bgColor | 背景色 | String | App-nvue 2.1.5+、微信小程序、H5、百度小程序 |
| padding | 文本边缘留白 | Number | App-nvue 2.1.5+、微信小程序、H5、百度小程序 |
| display | 'BYCLICK':点击显示; 'ALWAYS':常显 | String | App-nvue 2.1.5+、微信小程序、H5、百度小程序 |
| textAlign | 文本对齐方式。有效值: left, right, center | String | App-nvue 2.1.5+、微信小程序、百度小程序 |

**marker 上的标签 label**

| **属性** | **说明** | **类型** | **平台差异说明** |
| --- | --- | --- | --- |
| content | 文本 | String |  |
| color | 文本颜色 | String | App-nvue 2.1.5+、微信小程序、H5、百度小程序 |
| fontSize | 文字大小 | Number | App-nvue 2.1.5+、微信小程序、H5、百度小程序 |
| x | label的坐标，原点是 marker 对应的经纬度 | Number | App-nvue 2.1.5+、微信小程序、H5、百度小程序 |
| y | label的坐标，原点是 marker 对应的经纬度 | Number | App-nvue 2.1.5+、微信小程序、H5、百度小程序 |
| borderWidth | 边框宽度 | Number | App-nvue 2.1.5+、微信小程序、百度小程序 |
| borderColor | 边框颜色 | String | App-nvue 2.1.5+、微信小程序、百度小程序 |
| borderRadius | 边框圆角 | Number | App-nvue 2.1.5+、微信小程序、百度小程序 |
| bgColor | 背景色 | String | App-nvue 2.1.5+、微信小程序、百度小程序 |
| padding | 文本边缘留白 | Number | App-nvue 2.1.5+、微信小程序、百度小程序 |
| textAlign | 文本对齐方式。有效值: left, right, center | String | App-nvue 2.1.5+、微信小程序、百度小程序 |

**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** | **平台差异说明** |
| --- | --- | --- | --- | --- | --- |
| 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：#0000AA |  |
| width | 线的宽度 | Number | 否 |  |  |
| dottedLine | 是否虚线 | Boolean | 否 | 默认false | App-nvue 2.1.5+、微信小程序、H5、百度小程序、支付宝小程序 |
| arrowLine | 带箭头的线 | Boolean | 否 | 默认false，微信小程序开发者工具暂不支持该属性 | App-nvue 2.1.5+、微信小程序、百度小程序 |
| arrowIconPath | 更换箭头图标 | String | 否 | 在arrowLine为true时生效 | App-nvue 2.1.5+、微信小程序、百度小程序 |
| borderColor | 线的边框颜色 | String | 否 |  | App-nvue 2.1.5+、微信小程序、H5、百度小程序 |
| borderWidth | 线的厚度 | Number | 否 |  | App-nvue 2.1.5+、微信小程序、H5、百度小程序 |

**polygon**  
指定一系列坐标点，根据 points 坐标数据生成闭合多边形

| **属性** | **说明** | **类型** | **必填** | **备注** |
| --- | --- | --- | --- | --- |
| points | 经纬度数组 | array | 是 | [{latitude: 0, longitude: 0}] |
| strokeWidth | 描边的宽度 | Number | 否 |  |
| strokeColor | 描边的颜色 | String | 否 | 十六进制 |  |
| fillColor | 填充颜色 | String | 否 | 十六进制 |  |
| zIndex | 设置多边形 Z 轴数值 | Number | 否 |  |  |

**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：#0000AA |
| fillColor | 填充颜色 | String | 否 | 8位十六进制表示，后两位表示alpha值，如：#0000AA |
| 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

| **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'/'开头则表示相对项目根目录；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** [查看示例](https://uniapp.dcloud.io/h5/pages/component/map/map)

<template>

<view>

<view class="page-body">

<view class="page-section page-section-gap">

<map style="width: 100%; height: 300px;" :latitude="latitude" :longitude="longitude" :markers="covers">

</map>

</view>

</view>

</view>

</template>

export default {

data() {

return {

title: 'map',

latitude: 39.909,

longitude: 116.39742,

covers: [{

latitude: 39.909,

longitude: 116.39742,

iconPath: '../../../static/location.png'

}, {

latitude: 39.90,

longitude: 116.39,

iconPath: '../../../static/location.png'

}]

}

},

methods: {

}

}

map 组件相关操作的 JS API：[uni.createMapContext](https://uniapp.dcloud.io/api/location/map?id=createmapcontext)

**注意事项**

* 小程序和app-vue中，<map> 组件是由引擎创建的原生组件，它的层级是最高的，不能通过 z-index 控制层级。在<map>上绘制内容，可使用组件自带的marker、controls等属性，也可以使用<cover-view>组件。App端还可以使用plus.nativeObj.view 或 subNVue 绘制原生内容，[参考](https://uniapp.dcloud.io/component/native-component)。另外App端nvue文件不存在层级问题。从微信基础库2.8.3开始，支持map组件的同层渲染，不再有层级问题。
* App端nvue文件的map和小程序拉齐度更高。vue里的map则与plus.map功能一致，和小程序的地图略有差异。**App端使用map推荐使用nvue。**
* 在涉及层级问题的小程序中和app-vue中，请勿在 scroll-view、swiper、picker-view、movable-view 中使用 <map> 组件。
* 小程序和 app-vue 中，css 动画对 <map> 组件无效。
* map 组件使用的经纬度是国测局坐标，调用 uni.getLocation 接口需要指定 type 为 gcj02。
* <map> 组件在不同平台的底层引擎是不同的：H5、微信小程序为腾讯地图；App、支付宝小程序为高德地图；百度小程序端为百度地图。app-vue也可以使用百度地图，在manifest中配置，但app-nvue只支持高德地图。另外选择地图、查看地图位置的API也仅支持高德地图。App端如无特殊必要，建议使用高德地图。
* map 组件默认的api是参考微信小程序的，如需要使用plus.map，可以通过$getAppMap获取原生地图对象，[详见](https://uniapp.dcloud.io/api/location/map)。注意nvue的map组件不是plus.map对象，无法使用$getAppMap
* H5 端获取定位信息，需要部署在 **https** 服务上，本地预览（localhost）仍然可以使用 http 协议。
* 无 GPS 模块的 PC 设备使用 Chrome 浏览器的时候，位置信息是连接谷歌服务器获取的，国内用户可能获取位置信息失败。
* App 端使用地图组件需要向高德或百度等三方服务商申请SDK资质，获取AppKey，打包时需要在manifest文件中勾选相应模块，在SDK配置中填写Appkey。注意申请包名和打包时的包名需匹配一致，证书信息匹配。在manifest可视化界面有详细申请指南。

**FAQ**

Q：应用中使用了 <map> 组件，打包为App时，提示缺少权限模块怎么办？  
A：App端原生地图，依赖第三方的 SDK，因此打包移动应用时，需要在manifest中勾选相关的权限并填写 key 信息。详见：<https://ask.dcloud.net.cn/article/29>

Q：国外应用想使用谷歌地图/google地图怎么办？  
A：1. 可以在web-view下调用谷歌的web地图；2. 可以写一个原生插件封装谷歌原生地图，具体参考uni-app插件市场的原生插件开发教程

# 画布

#### [canvas](https://uniapp.dcloud.io/component/canvas?id=canvas)

画布。

**属性说明**

| **属性名** | **类型** | **默认值** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| 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 | 头条小程序不支持 |
| @touchstart | EventHandle |  | 手指触摸动作开始 | 头条小程序不支持 |
| @touchmove | EventHandle |  | 手指触摸后移动 | 头条小程序不支持 |
| @touchend | EventHandle |  | 手指触摸动作结束 | 头条小程序不支持 |
| @touchcancel | EventHandle |  | 手指触摸动作被打断，如来电提醒，弹窗 | 头条小程序不支持 |
| @longtap | 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 | 头条小程序不支持 |
| @error | EventHandle |  | 当发生错误时触发 error 事件，detail = {errMsg: 'something wrong'} | 头条小程序不支持 |

**注意事项：**

* canvas 标签默认宽度 300px、高度 225px，动态修改 canvas 大小后需要重新绘制。
* 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。
* canvas 在微信小程序、百度小程序、QQ小程序中为原生组件，层级高于前端组件，请勿内嵌在 scroll-view、swiper、picker-view、movable-view 中使用。解决 canvas 层级过高无法覆盖，参考 [native-component](https://uniapp.dcloud.io/component/native-component)。其他小程序端的 canvas 仍然为 webview 中的 canvas。
* app-vue 中的 canvas 仍然是 webview 的 canvas。app-nvue下如需使用canvas，需下载插件，暂未封装为uni API，可参考[文档](https://github.com/dcloudio/NvueCanvasDemo)使用。目前对nvue的开发者仍然建议在使用canvas时单独起一个vue页面，或者用web-view组件。
* app-vue的canvas虽然是webview自带的canvas，但却比nvue和微信小程序的原生canvas性能更高。注意并非原生=更快。canvas动画的流畅，关键不在于渲染引擎的速度，而在于减少从逻辑层向视图层频繁通信造成的折损。
* 小程序、app-nvue，因为通信阻塞，难以绘制非常流畅的canvas动画。h5和app-vue不存在此问题。但注意，app-vue下若想流畅的绘制canvas动画，需要使用[renderjs](https://uniapp.dcloud.io/frame?id=renderjs)技术，把操作canvas的js逻辑放到视图层运行，避免逻辑层和视图层频繁通信。hello uni-app的canvas示例很典型，在相同手机运行该示例，可以看出在h5端和app端非常流畅，而小程序端做不到这么流畅的动画。

**示例：**

<template>

<view>

<canvas style="width: 300px; height: 200px;" canvas-id="firstCanvas"></canvas>

<canvas style="width: 400px; height: 500px;" canvas-id="secondCanvas"></canvas>

<canvas style="width: 400px; height: 500px;" canvas-id="secondCanvas" @error="canvasIdErrorCallback"></canvas>

</view>

</template>

export default {

onReady: function (e) {

var context = uni.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()

},

methods: {

canvasIdErrorCallback: function (e) {

console.error(e.detail.errMsg)

}

}

}

相关 api：[uni.createCanvasContext](https://uniapp.dcloud.io/api/canvas/createCanvasContext)

**扩展阅读**

canvas的常用用途有图表和图片处理，在uni-app插件市场有大量基于canvas的插件，可搜索 [图表](https://ext.dcloud.net.cn/search?q=%E5%9B%BE%E8%A1%A8) 、 [头像裁剪](https://ext.dcloud.net.cn/search?q=%E5%A4%B4%E5%83%8F%E8%A3%81%E5%89%AA) 、 [海报](https://ext.dcloud.net.cn/search?q=%E6%B5%B7%E6%8A%A5) 、 [二维码](https://ext.dcloud.net.cn/search?q=%E4%BA%8C%E7%BB%B4%E7%A0%81) 。

关于图表

* H5端流行的echart报表因为涉及大量dom操作，无法跨端使用，而wx-chart在跨端和更新方面都不足。如考虑小程序，推荐使用全端可用的[uChart](https://ext.dcloud.net.cn/plugin?id=271)。
* 如只考虑H5端，也可以继续使用echart、f2等常规web图表。
* 如不考虑小程序，那么App端和H5，也可以通过renderjs技术来使用echart、f2等web图表，功能性能比uchart更好。[什么是renderjs](https://uniapp.dcloud.io/frame?id=renderjs)、[基于renderjs使用echart的示例](https://ext.dcloud.net.cn/plugin?id=1207" \t "_blank)

**nvue页面如何使用canvas**

HBuilderX 2.2.5（alpha）开始 nvue 页面支持 Canvas，支持 W3C WebGL API [WebGL 1.0](https://www.khronos.org/registry/webgl/specs/latest/1.0/)

示例工程地址：[NvueCanvasDemo](https://github.com/dcloudio/NvueCanvasDemo)

# Webview

#### [web-view](https://uniapp.dcloud.io/component/web-view?id=web-view)

web-view 是一个 web 浏览器组件，可以用来承载网页的容器，会自动铺满整个页面（nvue 使用需要手动指的宽高）。

**各小程序平台，web-view 加载的 url 需要在后台配置域名白名单，包括内部再次 iframe 内嵌的其他 url 。**

**属性说明**

| **属性名** | **类型** | **说明** | **平台差异说明** |
| --- | --- | --- | --- |
| src | String | webview 指向网页的链接 |  |
| webview-styles | Object | webview 的样式 | App |
| @message | EventHandler | 网页向应用 postMessage 时，会在特定时机（后退、组件销毁、分享）触发并收到消息。 | H5 暂不支持 |

**src**

| **来源** | **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 网络 | √ | √ | √ | √ | √ | √ | √ |
| 本地 | √ | 暂不支持 | x | x | x | x | x |

**webview-styles**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| progress | Object/Boolean | 进度条样式。仅加载网络 HTML 时生效，设置为 false 时禁用进度条。 |

**progress**

| **属性** | **类型** | **默认值** | **说明** |
| --- | --- | --- | --- |
| color | String | #00FF00 | 进度条颜色 |

<template>

<view>

<web-view :webview-styles="webviewStyles" src="https://uniapp.dcloud.io/static/web-view.html"></web-view>

</view>

</template>

<script>

export default {

data() {

return {

webviewStyles: {

progress: {

color: '#FF3333'

}

}

}

}

}

</script>

<style>

</style>

注意：

* 小程序仅支持加载网络网页，不支持本地html
* 小程序端 web-view 组件一定有原生导航栏，下面一定是全屏的 web-view 组件，navigationStyle: custom 对 web-view 组件无效。
* App 端使用 自定义组件模式 时，uni.web-view.js 的最低版为 [uni.webview.1.5.2.js](https://js.cdn.aliyun.dcloud.net.cn/dev/uni-app/uni.webview.1.5.2.js)
* App 平台同时支持网络网页和本地网页，但本地网页及相关资源（js、css等文件）必须放在 uni-app 项目根目录->hybrid->html 文件夹下，如下为一个加载本地网页的uni-app项目文件目录示例：
* nvue web-view 必须指定样式宽高, @message 暂时写成 @onPostMessage，示例:
* V3 编译模式，网页向应用 postMessage 为实时消息

┌─components

├─hybrid

│ └─html

│ ├─css

│ │ └─test.css

│ ├─img

│ │ └─icon.png

│ ├─js

│ │ └─test.js

│ └─local.html

├─pages

│ └─index

│ └─index.vue

├─static

├─main.js

├─App.vue

├─manifest.json

└─pages.json

**示例** [查看示例](https://uniapp.dcloud.io/h5/pages/component/web-view/web-view)

<template>

<view>

<web-view src="/hybrid/html/local.html"></web-view>

</view>

</template>

<web-view> 加载的网页中支持调用部分 uni 接口：

| **方法名** | **说明** | **平台差异说明** |
| --- | --- | --- |
| uni.navigateTo | [navigateTo](https://uniapp.dcloud.io/api/router?id=navigateto) |  |
| uni.redirectTo | [redirectTo](https://uniapp.dcloud.io/api/router?id=redirectto) |  |
| uni.reLaunch | [reLaunch](https://uniapp.dcloud.io/api/router?id=relaunch) |  |
| uni.switchTab | [switchTab](https://uniapp.dcloud.io/api/router?id=switchtab) |  |
| uni.navigateBack | [navigateBack](https://uniapp.dcloud.io/api/router?id=navigateback) |  |
| uni.postMessage | 向应用发送消息 | 头条小程序不支持 |
| uni.getEnv | 获取当前环境 | 头条小程序不支持 |

##### [uni.postMessage(OBJECT)](https://uniapp.dcloud.io/component/web-view?id=postmessage)

网页向应用发送消息，在 <web-view> 的 message 事件回调 event.detail.data 中接收消息。

**Tips**

* 传递的消息信息，必须写在 data 对象中。
* event.detail.data 中的数据，以数组的形式接收每次 post 的消息。

##### [uni.getEnv(CALLBACK)](https://uniapp.dcloud.io/component/web-view?id=getenv)

**callback 返回的对象**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| plus | Boolean | App |
| miniprogram | Boolean | 微信小程序 |
| smartprogram | Boolean | 百度小程序 |
| miniprogram | Boolean | 支付宝小程序 |

**示例**

在 <web-view> 加载的 HTML 中，添加以下代码：

<script type="text/javascript">

var userAgent = navigator.userAgent;

if (userAgent.indexOf('AlipayClient') > -1) {

// 支付宝小程序的 JS-SDK 防止 404 需要动态加载，如果不需要兼容支付宝小程序，则无需引用此 JS 文件。

document.writeln('<script src="https://appx/web-view.min.js"' + '>' + '<' + '/' + 'script>');

} else if (/QQ/i.test(userAgent) && /miniProgram/i.test(userAgent)) {

// QQ 小程序

document.write('<script type="text/javascript" src="https://qqq.gtimg.cn/miniprogram/webview\_jssdk/qqjssdk-1.0.0.js"><\/script>');

} else if (/miniProgram/i.test(userAgent)) {

// 微信小程序 JS-SDK 如果不需要兼容微信小程序，则无需引用此 JS 文件。

document.write('<script type="text/javascript" src="https://res.wx.qq.com/open/js/jweixin-1.4.0.js"><\/script>');

} else if (/toutiaomicroapp/i.test(userAgent)) {

// 头条小程序 JS-SDK 如果不需要兼容头条小程序，则无需引用此 JS 文件。

document.write('<script type="text/javascript" src="https://s3.pstatp.com/toutiao/tmajssdk/jssdk-1.0.1.js"><\/script>');

} else if (/swan/i.test(userAgent)) {

// 百度小程序 JS-SDK 如果不需要兼容百度小程序，则无需引用此 JS 文件。

document.write('<script type="text/javascript" src="https://b.bdstatic.com/searchbox/icms/searchbox/js/swan-2.0.18.js"><\/script>');

}

if (!/toutiaomicroapp/i.test(userAgent)) {

document.querySelector('.post-message-section').style.visibility = 'visible';

}

</script>

<!-- uni 的 SDK -->

<script type="text/javascript" src="https://js.cdn.aliyun.dcloud.net.cn/dev/uni-app/uni.webview.1.5.2.js"></script>

待触发 UniAppJSBridgeReady 事件后，即可调用 uni 的 API。

<script>

document.addEventListener('UniAppJSBridgeReady', function() {

uni.postMessage({

data: {

action: 'message'

}

});

uni.getEnv(function(res) {

console.log('当前环境：' + JSON.stringify(res));

});

});

</script>

##### [****App端web-view的扩展****](https://uniapp.dcloud.io/component/web-view?id=app%e7%ab%afweb-view%e7%9a%84%e6%89%a9%e5%b1%95)

App端的webview是非常强大的，可以更灵活的控制和拥有更丰富的API。

每个vue页面，其实都是一个webview，而vue页面里的web-view组件，其实是webview里的一个子webview。这个子webview被append到父webview上。

通过以下方法，可以获得这个web-view组件对应的js对象，然后参考<https://www.html5plus.org/doc/zh_cn/webview.html#plus.webview.WebviewObject>，可以进一步重设这个web-view组件的样式，比如调整大小

<template>

<view>

<web-view src="https://www.baidu.com"></web-view>

</view>

</template>

<script>

var wv;//计划创建的webview

export default {

onReady() {

// #ifdef APP-PLUS

var currentWebview = this.$mp.page.$getAppWebview() //获取当前页面的webview对象

setTimeout(function() {

wv = currentWebview.children()[0]

wv.setStyle({top:150,height:300})

}, 1000); //如果是页面初始化调用时，需要延时一下

// #endif

}

};

</script>

甚至可以不用web-view组件，直接js创建一个子webview来加载html。比如不希望远程网页使用plus的API，不管是因为安全原因还是因为back监听冲突，可以使用如下代码：

<template>

<view>

</view>

</template>

<script>

var wv;//计划创建的webview

export default {

onLoad() {

// #ifdef APP-PLUS

wv = plus.webview.create("","custom-webview",{

plusrequire:"none", //禁止远程网页使用plus的API，有些使用mui制作的网页可能会监听plus.key，造成关闭页面混乱，可以通过这种方式禁止

'uni-app': 'none', //不加载uni-app渲染层框架，避免样式冲突

top:uni.getSystemInfoSync().statusBarHeight+44 //放置在titleNView下方。如果还想在webview上方加个地址栏的什么的，可以继续降低TOP值

})

wv.loadURL("https://www.baidu.com")

var currentWebview = this.$mp.page.$getAppWebview() //获取当前页面的webview对象

currentWebview.append(wv);//一定要append到当前的页面里！！！才能跟随当前页面一起做动画，一起关闭

setTimeout(function() {

console.log(wv.getStyle())

}, 1000);//如果是首页的onload调用时需要延时一下，二级页面无需延时，可直接获取

// #endif

}

};

</script>

如果想设置web-view组件可双指缩放，可参考如下代码：

onReady() {

// #ifdef APP-PLUS

var currentWebview = this.$mp.page.$getAppWebview() //获取当前页面的webview对象

setTimeout(function() {

wv = currentWebview.children()[0]

wv.setStyle({scalable:true})

}, 1000); //如果是页面初始化调用时，需要延时一下

// #endif

}

};

##### [web-view组件的层级问题解决](https://uniapp.dcloud.io/component/web-view?id=web-view%e7%bb%84%e4%bb%b6%e7%9a%84%e5%b1%82%e7%ba%a7%e9%97%ae%e9%a2%98%e8%a7%a3%e5%86%b3)

web-view组件在App和小程序中层级较高，如需要在vue页面中写代码为web-view组件覆盖内容，小程序端无解，只能由web-view的组件自己弹出div。App端有如下若干方案：

1. 比较简单的方式是actionsheet等原生弹出菜单
2. 也可以使用plus.nativeObj.view。这里有一个底部图标菜单的示例，可参考<https://ext.dcloud.net.cn/plugin?id=69>
3. 也可以使用HBuilderX1.9.10以后新出的[原生子窗体subNvue](https://uniapp.dcloud.io/api/window/subNVues)
4. 也可以在web-view组件内嵌的网页中弹出z-index更高的div。如果是外部网页，可以在vue中获得子webview对象后，通过[evalJS](https://www.html5plus.org/doc/zh_cn/webview.html#plus.webview.WebviewObject.evalJS)为这个子webview注入一段js，操作其弹出div层。

##### [web-view组件的浏览器内核说明](https://uniapp.dcloud.io/component/web-view?id=web-view%e7%bb%84%e4%bb%b6%e7%9a%84%e6%b5%8f%e8%a7%88%e5%99%a8%e5%86%85%e6%a0%b8%e8%af%b4%e6%98%8e)

* H5端的web-view其实是被转为iframe运行，使用的是当前的浏览器
* 小程序的web-view使用的是小程序自带的浏览器内核，不同厂商不一样，[详见](https://ask.dcloud.net.cn/article/1318)
* App端，Android，使用的是os自带的浏览器内核，在设置-所有应用里，显示系统服务，可查看Android System Webview的版本。在Android5+，系统webview支持安装升级。
* App端，iOS，是分为UIWebview和WKWebview的，2.2.5+起默认为WKWebview，之前版本[详见](https://ask.dcloud.net.cn/article/36348)

**注意事项**

* <web-view> 组件默认铺满全屏并且层级高于前端组件。App端想调节大小或再其上覆盖内容需使用plus规范。
* <web-view> 组件所在窗口的标题，跟随页面的 <title> 值的变化而变化。
* App端，web-view加载的html页面可以运行plus的api，但注意如果该页面调用了plus.key的API监听了back按键（或使用mui的封装），会造成back监听冲突。需要该html页面移除对back的监听。或按照上面的示例代码禁止网页使用plus对象
* uni.webview.js 最新版地址：<https://js.cdn.aliyun.dcloud.net.cn/dev/uni-app/uni.webview.1.5.2.js>
* 小程序平台，个人类型与海外类型的小程序使用 web-view 组件，提交审核时注意微信等平台是否允许使用
* 小程序平台， src 指向的链接需登录小程序管理后台配置域名白名单。App和H5 无此限制。

##### [FAQ](https://uniapp.dcloud.io/component/web-view?id=faq)

Q：web-view 的页面怎么和应用内的页面交互？  
A：调用 uni 相关的 API，就可以实现页面切换及发送消息。参考：[在 web-view 加载的 HTML 中调用 uni 的 API](https://ask.dcloud.net.cn/article/35083)

Q：web-view 加载的 HTML 中，能够调用 5+ 的能力么？  
A：加载的 HTML 中是有 5+ 环境的，在 plusready 后调用即可。参考：[一个简单实用的 plusready 方法](https://ask.dcloud.net.cn/article/34922)

# 广告

#### [ad](https://uniapp.dcloud.io/component/ad?id=ad)

应用内展示的广告组件，可用于banner或信息流。

* app端的广告源由腾讯广点通、头条穿山甲、360广告联盟提供，DCloud负责聚合
* 小程序端的广告由小程序平台提供

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √（2.5.2+） | x | √ | x | √ | √ | √ |

**开通配置广告**

开通广告步骤：

1. 开通广告 需在广告平台后台操作：
   * App平台：<https://uniad.dcloud.net.cn/>
   * 小程序平台：在各自的小程序管理后台操作。
2. 申请广告位id 在各位后台申请广告位id
3. 在页面合适位置编写代码，放置ad组件，配上广告位id（app是adpid，微信、头条、qq小程序是unit-id，百度小程序是apid）
4. App端打包后生效，打包时必须选择要集成的广告SDK（穿山甲、广点通、360联盟）。

**属性说明**

| **属性名** | **类型** | **默认值** | **说明** | **平台差异** |
| --- | --- | --- | --- | --- |
| data | Object | 可选 | 广告数据，通过 plus.ad.getAds (参考示例代码)，设置后adpid将无效 | App |
| adpid | String |  | uni-AD App广告位id，在[uni-AD官网](https://uniad.dcloud.net.cn/)申请广告位 | App |
| 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 | 微信小程序、头条小程序(最低版本1.19.0+)、QQ小程序 |
| ad-intervals | number |  | 广告自动刷新的间隔时间，单位为秒，参数值必须大于等于30（该参数不传入时 Banner 广告不会自动刷新） | 微信小程序(基础库2.3.1+) |
| appid | String |  | 小程序应用 ID | 百度小程序 |
| apid | String |  | 小程序广告位 ID | 百度小程序 |
| ad-left | Number |  | type为feeds时广告左边距（px），必须大于0 | QQ小程序 |
| ad-top | Number |  | type为feeds时广告上边距（px），必须大于0 | QQ小程序 |
| ad-width | Number |  | type为feeds时广告宽度（px），默认100%，最大值为屏幕宽度，最小值为265 | QQ小程序 |
| ad-height | Number |  | type为feeds时广告高度（px），最小85，最大160 | QQ小程序 |
| type | String | feed |  | QQ小程序、百度小程序、头条小程序 |
| @load | EventHandle |  | 广告加载成功的回调 |  |
| @error | EventHandle |  | 广告加载失败的回调，event.detail = {errCode: } |  |
| @close | EventHandle |  | 广告关闭的回调 |  |
| @downloadchange | EventHandle |  | 下载类广告的状态, event.detail = {status: } | 仅App Android 支持 |

**type属性 百度**

广告类型：banner/feed，需和百青藤平台上的代码位类型相匹配。

**type属性 头条**

广告的类型，默认 bannner，具体类型有：banner、video（视频）、large（大图）、lImg（左图右文）、rImg（右图左文），默认值为 banner

**type属性 QQ**

| **值** | **说明** |
| --- | --- |
| banner | banner广告 分 1 图和 3 图 1 文。3 图 1 文广告的背景色、文字颜色会根据祖先节点的背景色调整，分三种情况深色背景、浅色背景和白色背景 |
| swip | 翻页广告，1 图 1 文，会覆盖整个小程序，显示、隐藏逻辑需开发者自行控制 |
| card | 卡片广告，1 图，可关闭 |
| feeds | 自定义广告，可灵活控制广告上、左边距和宽高，以适应界面其他内容。可监听size事件获取实际宽高 |

App和微信小程序的ad组件没有type属性，可以用于banner，也可以用于信息流。

**注意**

* 无广告时没有高度，关闭广告时释放高度，宽度由父容器决定
* App 平台，因广告组件内部获得广告数据计算后设置组件大小，会出现界面抖动问题，可以提前通过 plus.ad.getAds 获得广告数据，设置 data 后 adpid 将无效
* 微信小程序 <ad> 组件不支持触发 tap 等触摸相关事件
* Android 平台 nvue的 <list> 组件中使用 <ad> 时，必须指定宽度属性<ad width="750rpx" />，因为 <list> 有自动的内存回收机制，不在屏幕范围的组件不被创建，组件内部无法获取大小
* 广点通概率出现重复广告，可根据需求请求广告数据，推荐单次大于1条(plus.ad.getAds) 来降低重复率
* vue 页面使用 <ad> 暂不支持非 V3 编译，manifest.json 可配置
* <recycle-list> 暂不支持 <ad>

**@error 错误码**

* App端聚合的穿山甲：[错误码](https://ad.oceanengine.com/union/media/doc?id=5de4cc6d78c8690012a90aa5)
* App端聚合的广点通：[错误码](https://developers.adnet.qq.com/doc/ios/union/union_debug#%E9%94%99%E8%AF%AF%E7%A0%81)

**@downloadchange status**

| **代码** | **说明** |
| --- | --- |
| -1 | 非下载 |
| 0 | 未开始下载 |
| 1 | 下载中 |
| 2 | 下载暂停 |
| 3 | 下载完成后 |
| 4 | 下载失败 |
| 5 | 下载取消 |
| 6 | 已安装 |

**示例：**

<template>

<view class="content">

<!-- App平台 示例 1 -->

<view class="ad-view">

<ad adpid="1111111111" @load="onload" @close="onclose" @error="onerror" @downloadchange="ondownloadchange"></ad>

</view>

<!-- App平台 示例 2 -->

<!-- 需要时可自定义属性，监听到 error 回调后(e.target可以取到)，开发者可以针对性的处理，比如隐藏广告组件的父容器，以保证用户体验 -->

<view class="ad-view" v-for="adItem in adList" :key="adItem.id">

<ad :adpid="adItem.adpid" :data-xx="adItem.id"></ad>

</view>

<!-- App平台 示例 3 (手动请求广告数据 仅App平台支持) -->

<view>

<button @click="getAdData">Get ad data</button>

</view>

<view class="ad-view">

<ad :data="adData"></ad>

</view>

<!-- 微信小程序 -->

<view class="ad-view">

<ad unit-id="" ad-intervals="100"></ad>

</view>

<!-- 百度小程序 -->

<view class="ad-view">

<ad appid="" apid="" type="feed"></ad>

</view>

<!-- 多平台兼容 -->

<view class="ad-view">

<ad adpid="1111111111" unit-id="" appid="" apid="" type="feed"></ad>

</view>

</view>

</template>

export default {

data() {

return {

title: 'uni-app ad',

adList: [],

adData: {}

}

},

onReady: function (e) {

// 显示 4 个广告

for (let i = 0; i < 4; i++) {

this.adList.push({

id: i,

adpid: "1111111111"

})

}

},

methods: {

getAdData: function (e) {

// 仅APP平台支持

plus.ad.getAds({

adpid: '', // dcloud 后台创建

count: 1, // 广告数量，默认 3

width: 300 // 根据宽度获取合适的广告(单位px)

},

(res) => {

this.adData = res.ads[0];

console.log(this.adData);

},

(err) => {

console.log(err);

}

)

},

onload(e) {

console.log("onload");

},

onclose(e) {

console.log("onclose: " + e.detail);

},

onerror(e) {

console.log("onerror: " + e.detail.errCode + " message:: " + e.detail.errMsg);

},

ondownloadchange(e) {

console.log("downloadchanged: " + JSON.stringify(e.detail));

}

}

}

<style>

.content {

background-color: #DBDBDB;

padding: 10px;

}

.ad-view {

background-color: #FFFFFF;

margin-bottom: 10px;

}

</style>

**激励视频广告** 文档地址：<https://uniapp.dcloud.io/api/ad/rewarded-video-ad>

**注意**

* App端广告开通指南和收益相关问题：<https://ask.dcloud.net.cn/article/36769>
* App端除了ad组件，还支持开屏。详见[uni-AD官网](https://uniad.dcloud.net.cn/)
* App端uni-AD聚合了腾讯广点通、头条穿山甲、360广告联盟等服务，打包时必须勾选相应的sdk，详见：<https://ask.dcloud.net.cn/article/36718> 

# 小程序开放能力组件

#### [official-account](https://uniapp.dcloud.io/component/official-account?id=official-account)

微信公众号关注组件。当用户扫小程序码打开小程序时，开发者可在小程序内配置公众号关注组件，方便用户快捷关注公众号，可嵌套在原生组件内。

仅微信小程序平台支持，[规范详情](https://developers.weixin.qq.com/miniprogram/dev/component/official-account.html)

支付宝平台另提供了lifestyle组件，可关注支付宝生活号，[规范详情](https://docs.alipay.com/mini/component/lifestyle)

#### [open-data](https://uniapp.dcloud.io/component/open-data?id=open-data)

用于展示平台开放的数据。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x | x | √ | x | √ | x | √ |

**Tips**

该功能为各小程序平台提供的开放能力。App端和H5端不涉及此概念。

支付宝和头条没有open-data组件，但提供了API方式获取相关信息。支付宝[参考](https://docs.alipay.com/mini/api/ch8chh)、头条[参考](https://developer.toutiao.com/dev/cn/mini-app/develop/open-capacity/user-information/getuserinfo)

**属性说明**

| **属性名** | **类型** | **默认值** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| type | String |  | 开放数据类型 |  |
| open-gid | String |  | 当 type="groupName" 时生效, 群id | 微信小程序、QQ小程序 |
| 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 | 微信小程序、QQ小程序 |

**type 有效值**

| **值** | **说明** | **平台差异说明** |
| --- | --- | --- |
| userNickName | 用户昵称 |  |
| userAvatarUrl | 用户头像 |  |
| userGender | 用户性别 |  |
| groupName | 拉取群名称 | 微信小程序、QQ小程序 |
| userCity | 用户所在城市 | 微信小程序、QQ小程序 |
| userProvince | 用户所在省份 | 微信小程序、QQ小程序 |
| userCountry | 用户所在国家 | 微信小程序、QQ小程序 |
| userLanguage | 用户的语言 | 微信小程序、QQ小程序 |

**示例**

<open-data type="userNickName"></open-data>

<open-data type="userAvatarUrl"></open-data>

<open-data type="userGender"></open-data>

# App nvue专用组件

#### [Barcode](https://uniapp.dcloud.io/component/barcode?id=barcode)

app端nvue专用的扫码组件。

* 此组件用于app端nvue页面实现内嵌到界面上的扫码。其他场景、其他平台，请使用全屏扫码API：[uni.scanCode](https://uniapp.dcloud.io/api/system/barcode" \t "_blank)
* App下纯nvue项目（manifest中renderer为native），暂不支持uni.scanCode API，此时只能使用barcode组件来替代。
* 此组件自HBuilderX 2.1.5+起支持。

**属性说明** 设置Barcode扫码控件的属性，如扫码框、扫码条的颜色等。

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| autostart | boolean | false | 否 | 是否自动开始扫码 |
| background | string |  | 否 | 条码识别控件背景颜色,颜色值支持(参考CSS颜色规范)：颜色名称(参考CSS Color Names)/十六进制值/rgb值，默认值为黑色。 |
| frameColor | string |  | 否 | 扫码框颜色,颜色值支持(参考CSS颜色规范)：颜色名称(参考CSS Color Names)/十六进制值/rgb值/rgba值，默认值为红色。 |
| scanbarColor | string |  | 否 | 扫码条颜色,颜色值支持(参考CSS颜色规范)：颜色名称(参考CSS Color Names)/十六进制值/rgb值/rgba值，默认值为红色。 |
| filters | Array[Number] | [0,1,2] | 否 | 条码类型过滤器，条码类型常量数组，默认情况支持QR、EAN13、EAN8类型。 通过此参数可设置扫码识别支持的条码类型（注意：设置支持的条码类型越多，扫描识别速度可能将会降低）。 |

**码类型常量**

* QR: QR二维码，数值为0
* EAN13: EAN条形码标准版，数值为1
* EAN8: ENA条形码简版，数值为2
* AZTEC: Aztec二维码，数值为3
* DATAMATRIX: Data Matrix二维码，数值为4
* UPCA: UPC条形码标准版，数值为5
* UPCE: UPC条形码缩短版，数值为6
* CODABAR: Codabar条形码，数值为7
* CODE39: Code39条形码，数值为8
* CODE93: Code93条形码，数值为9
* CODE128: Code128条形码，数值为10
* ITF: ITF条形码，数值为11
* MAXICODE: MaxiCode二维码，数值为12
* PDF417: PDF 417二维条码，数值为13
* RSS14: RSS 14条形组合码，数值为14
* RSSEXPANDED: 扩展式RSS条形组合码，数值为15

##### [start(object)](https://uniapp.dcloud.io/component/barcode?id=start)

**开始扫码识别**

###### [Object object](https://uniapp.dcloud.io/component/barcode?id=object-object)

| **属性** | **说明** | **类型** | **必填** | **备注** |
| --- | --- | --- | --- | --- |
| conserve | 是否保存扫码成功时的截图 | Boolean | 否 | 如果设置为true则在扫码成功时将图片保存，并通过onmarked回调函数的file参数返回保存文件的路径。 默认值为false，不保存截图。 |
| filename | 保存扫码成功时图片保存路径 | String | 否 | 可通过此参数设置保存截图的路径和名称，如果设置图片文件名称则必须指定文件的后缀名（必须是.png），否则认为是指定目录，文件名称则自动生成。 |
| vibrate | 扫码成功时是否需要震动提醒 | Boolean | 否 | 如果设置为true则在扫码成功时震动设备，false则不震动。 默认值为true。 |
| sound | 扫码成功时播放的提示音 | String | 否 | 可取值： "none" - 不播放提示音； "default" - 播放默认提示音（5+引擎内置）。 默认值为"default"。 |

##### [cancel()](https://uniapp.dcloud.io/component/barcode?id=cancel)

**取消扫码识别**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| 无 | 无 | 无 | 结束对摄像头获取图片数据进行条码识别操作，同时关闭摄像头的视频捕获。 结束后可调用start方法重新开始识别。 |

##### [setFlash(boolean)](https://uniapp.dcloud.io/component/barcode?id=setflash)

**操作闪光灯**

###### [Boolean boolean](https://uniapp.dcloud.io/component/barcode?id=boolean-boolean)

| **类型** | **必填** | **说明** | **备注** |
| --- | --- | --- | --- |
| Boolean | 是 | 是否开启闪光灯 | 可取值true或false，true表示打开闪光灯，false表示关闭闪光灯。 |

##### [事件](https://uniapp.dcloud.io/component/barcode?id=%e4%ba%8b%e4%bb%b6)

##### [marked](https://uniapp.dcloud.io/component/barcode?id=marked)

**条码识别成功事件 {event.detail}**

###### [返回参数说明](https://uniapp.dcloud.io/component/barcode?id=%e8%bf%94%e5%9b%9e%e5%8f%82%e6%95%b0%e8%af%b4%e6%98%8e)

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| type | string | "success" 表示成功 |
| message | string | 识别到的条码数据，扫码识别出的数据内容，字符串类型，采用UTF8编码格式。 |
| code | Number | 识别到的条码类型，与Barcode组件的条码类型常量一致。 |
| file | string | 扫码成功的截图文件路径，扫码识别到的截图，png格式文件，如果设置为不保存截图，则返回undefined。 |

##### [error](https://uniapp.dcloud.io/component/barcode?id=error)

**条码识别错误事件**

###### [返回参数说明](https://uniapp.dcloud.io/component/barcode?id=%e8%bf%94%e5%9b%9e%e5%8f%82%e6%95%b0%e8%af%b4%e6%98%8e-1)

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| type | string | "fail" 表示失败 |
| code | number | 相应 code 码 |
| message | string | 失败描述 |

**示例：**

<template>

<view>

<barcode id='1' class="barcode" autostart="true" ref="barcode" background="rgb(0,0,0)" frameColor="#1C86EE" scanbarColor="#1C86EE" :filters="fil" @marked="success1" @error="fail1"></barcode>

<button class="btn" @click="toStart">开始扫码识别</button>

<button class="btn" @click="tocancel">取消扫码识别</button>

<button class="btn" @click="toFlash">开启闪光灯</button>

<button class="btn" @click="toscan">预览</button>

</view>

</template>

<script>

export default {

onLoad() {

},

data: {

fil: [0, 2, 1]

},

methods: {

success1(e) {

console.log("success1:" + JSON.stringify(e));

},

fail1(e) {

console.log("fail1:" + JSON.stringify(e));

},

toStart: function() {

this.$refs.barcode.start({

conserve: true,

filename: '\_doc/barcode/'

});

},

tocancel:function(){

this.$refs.barcode.cancel();

},

toFlash: function() {

this.$refs.barcode.setFlash(true);

},

toscan: function() {

console.log("scan:");

const barcodeModule = uni.requireNativePlugin('barcodeScan');

barcodeModule.scan("/static/barcode1.png"

,(e)=>{

console.log("scan\_error:"+JSON.stringify(e));

});

}

}

}

</script>

<style>

.barcode {

width: 750rpx;

height: 700rpx;

background-color: #808080;

}

.btn {

top: 20rpx;

width: 730rpx;

margin-left: 10rpx;

margin-top: 10rpx;

background-color: #458B00;

border-radius: 10rpx;

}

</style>

#### [list](https://uniapp.dcloud.io/component/list?id=list)

app端nvue专用组件。在app-nvue下，如果是长列表，使用list组件的性能高于使用view或scroll-view的滚动。原因在于list在不可见部分的渲染资源回收有特殊的优化处理。

原生渲染的资源回收机制，与webview渲染不同。webview不需要数据有规则格式，长页面处于不可视的部分，其渲染资源会自动回收，除非webview使用区域滚动而不是页面滚动。所以vue页面只要不用scroll-view，就不需要关注这个问题。而原生渲染则必须注意。

**如果需要跨端，建议使用uni-ui的uni-list组件，它会自动处理webview渲染和原生渲染的情况，在app-nvue下使用list组件，而在其他平台则使用页面滚动。详见**<https://ext.dcloud.net.cn/plugin?id=24>

<list> 组件是提供垂直列表功能的核心组件，拥有平滑的滚动和高效的内存管理，非常适合用于长列表的展示。最简单的使用方法是在 <list> 标签内使用一组由简单数组循环生成的 <cell> 标签填充。

<template>

<list>

<cell v-for="num in lists">

<text>{{num}}</text>

</cell>

</list>

</template>

<script>

export default {

data () {

return {

lists: ['A', 'B', 'C', 'D', 'E']

}

}

}

</script>

**注意**

* 相同方向 <list> 或者 <scroll-view> 互相嵌套时，Android 平台子 <list> 不可滚动，iOS 可以，iOS 有Bounce效果， Android仅可滚动时有
* <list> 需要显式的设置其宽高，可使用 position: absolute; 定位或 width、height 设置其宽高值。
* list是区域滚动，不会触发页面滚动，无法触发pages.json配置的下拉刷新、页面触底onReachBottomDistance、titleNView的transparent透明渐变。
* Android 平台，因 <list> 高效内存回收机制，不在屏幕可见区域的组件不会被创建，导致一些内部需要计算宽高的组件无法正常工作，例如 <slider>、<progress>、<swiper>

自组件

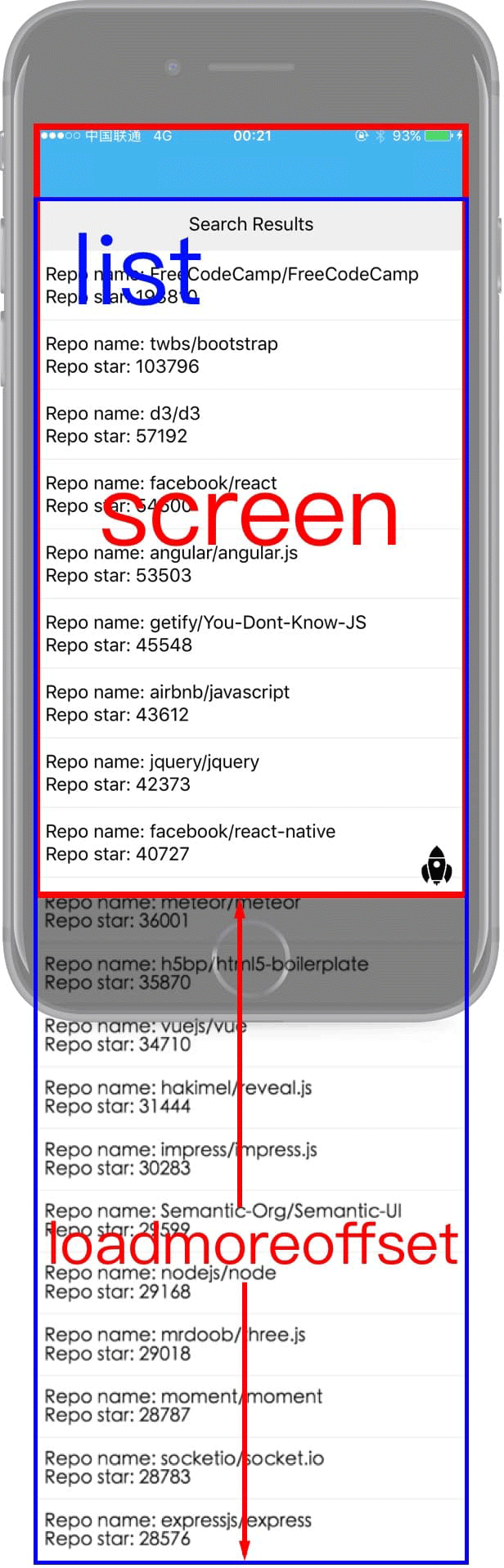
<list> 的子组件只能包括以下四种组件或是 fix 定位的组件，其他形式的组件将不能被正确渲染。

* <cell>  
  用于定义列表中的子列表项，类似于 HTML 中的 ul 之于 li。<list> 会对 <cell> 进行高效的内存回收以达到更好的性能。
* <header>  
  当 <header> 到达屏幕顶部时，吸附在屏幕顶部。
* <refresh>  
  用于给列表添加下拉刷新的功能。
* <loading>  
  <loading> 用法与特性和 <refresh> 类似，用于给列表添加上拉加载更多的功能。

属性

| **属性名** | **说明** | **类型** | **默认值** |
| --- | --- | --- | --- |
| show-scrollbar | 控制是否出现滚动条 | boolean | true |
| loadmoreoffset | 触发 loadmore 事件所需要的垂直偏移距离（设备屏幕底部与 list 底部之间的距离），建议手动设置此值，设置大于0的值即可 | number | 0 |
| offset-accuracy | 控制 onscroll 事件触发的频率：表示两次onscroll事件之间列表至少滚动了10px。注意，将该值设置为较小的数值会提高滚动事件采样的精度，但同时也会降低页面的性能 | number | 10 |
| pagingEnabled | 是否按分页模式显示List，默认值false | boolean | true/false |
| scrollable | 是否允许List滚动 | boolean | true/false |

loadmoreoffset 示意图：



事件

* loadmore 事件 如果列表滚动到底部将会立即触发这个事件，你可以在这个事件的处理函数中加载下一页的列表项。 如果未触发，请检查是否设置了loadmoreoffset的值，建议此值设置大于0

如何重置 loadmore

<template>

<list ref="list">

<cell v-for="num in lists">

<text>{{num}}</text>

</cell>

</list>

</template>

<script>

export default {

data () {

return {

lists: ['A', 'B', 'C', 'D', 'E']

}

},

methods: {

// 重置 loadmore

resetLoadMore() {

this.$refs["list"].resetLoadmore();

}

}

}

</script>

* scroll 事件 列表发生滚动时将会触发该事件，事件的默认抽样率为 10px，即列表每滚动 10px 触发一次，可通过属性 offset-accuracy 设置抽样率。

事件中的 event 对象属性：

* + contentSize {Object}：列表的内容尺寸
    - width {number}：列表内容宽度
    - height {number}：列表内容高度
  + contentOffset {Object}：列表的偏移尺寸
    - x {number}：x轴上的偏移量
    - y {number}：y轴上的偏移量
  + isDragging {boolean}: 用户是否正在拖动列表

#### [cell](https://uniapp.dcloud.io/component/cell?id=cell)

app端nvue专用组件。它的重要价值在于告诉原生引擎，哪些部分是可重用的。

<cell> Cell 必须以一级子组件的形式存在于 list recycler waterfall 中。

子组件

Cell 支持添加任意类型的组件作为自己的子组件，但是请不要再内部添加滚动容器了。

属性

* keep-scroll-position boolean. 控制当 Cell 被添加到列表中时，列表的滚动位置是否要保持不变。
* insert-animation string, cell 的插入动画。当前只支持 none 和 default。
* delete-animation string, cell 的删除动画。当前只支持 none 和 default。
* recycle boolean, 默认值 true。这个属性控制这个 Cell 的 view 和子 views 是否在列表滚动时进行回收，在 iOS 上通常必须指定为 true （因为默认为 true，所以一般不需要写这个属性），如果设置为 false，列表滚动时，页面会占用非常高的内存。Android上默认是true，设置为false可以防止Image和Text上数据重新绑定。
* 样式
* 通用样式.

**TIP**

* 不要指定 <cell> 的 flex 值。Cell 的宽度是由它的父容器决定的，你也不需要指定它的高度。
* Cell 的排版的位置是由父容器控制的，所以一般不要为其指定 margin 样式

事件

* 通用事件.

#### [recycle-list](https://uniapp.dcloud.io/component/recycle-list?id=recycle-list)

app端nvue专用组件。

<recycle-list> 是一个新的列表容器，具有回收和复用的能力，可以大幅优化内存占用和渲染性能。它的性能比list组件更高，但写法受限制。它除了会释放不可见区域的渲染资源，在非渲染的数据结构上也有更多优化。

此组件自HBuilderX 2.2.6+起支持。

[**子组件**](https://uniapp.dcloud.io/component/recycle-list?id=%e5%ad%90%e7%bb%84%e4%bb%b6)

<recycle-list> 只能使用 <cell-slot> 作为其直接子节点，使用其他节点无效。

代表的是列表每一项的模板，它只用来描述模板的结构，并不对应实际的节点。<cell-slot> 的个数只表示模板的种类数，真实列表项的个数是由数据决定的。

**注意**

* <cell-slot> 包含自定义组件时，在 Android 上有性能问题
* recycle-list 是区域滚动，不会触发页面滚动，无法触发pages.json配置的下拉刷新、页面触底onReachBottomDistance、titleNView的transparent透明渐变。

| **属性** | **说明** |
| --- | --- |
| case | 声明了当前模板的类型，只有和数据中的类型与当前类型匹配时才会渲染，语义和编程语言里的 case 一致。所有模板中最多只会匹配到一项，按照模板的顺序从上到下匹配，一旦匹配成功就不在继续匹配下一个。 |
| default | 表示当前模板为默认模板类型，不需要指定值。如果数据项没有匹配到任何 case 类型，则渲染带有 default 模板。如果存在多个 default，则只会使用第一个默认模板。 |
| key | 可选属性，用于指定列表数据中可以作为唯一标识的键值，可以优化渲染性能。 |

* warning 属性的省略 - 如果没写 switch，无论有没有写 case 或 default，都只使用第一个模板 - 在写了 switch 的情况下，case 和 default 必须写一个，否则该模板将会被忽略

属性

* for 在 <recycle-list> 添加 for 属性即可描述如何循环展开列表的数据，语法和 Vue 的 v-for 指令类似，但是它循环的是自己内部的子节点，并不是当前节点。写法：
  + alias in expression
  + (alias, index) in expression
* switch 在 <recycle-list> 添加 switch 属性可以用来指定数据中用于区分子模板类型的字段名，语义和编程语言里的 switch 一致，配合 <cell-slot> 中的 case 和 default 属性一起使用。 如果省略了 switch 属性，则只会将第一个 <cell-slot> 视为模板，多余的模板将会被忽略。

<recycle-list for="(item, i) in longList" switch="type">

<cell-slot case="A">

<text>- A {{i}} -</text>

</cell-slot>

<cell-slot case="B">

<text>- B {{i}} -</text>

</cell-slot>

</recycle-list>

可复用的组件

在 <recycle-list> 中使用的子组件也将被视为模板，在开发组件时给 <template> 标签添加 recyclable 属性，才可以用在 <recycle-list> 中。

<template recyclable>

<div>

<text>...</text>

</div>

</template>

<script>

// ...

</script>

**添加了 recyclable 属性并不会影响组件本身的功能，它仍然可以用在其他正常的组件里。**

注意事项

[**属性和文本的绑定**](https://uniapp.dcloud.io/component/recycle-list?id=%e5%b1%9e%e6%80%a7%e5%92%8c%e6%96%87%e6%9c%ac%e7%9a%84%e7%bb%91%e5%ae%9a)

绑定属性或者文本时，仅支持表达式，不支持函数调用，也不支持使用 filter，可以参考 Implementation.md#支持的表达式。

例如，下列写法不可用：

<div :prop="capitalize(card.title)">

<text>{{ card.title | capitalize }}</text>

</div>

针对这种场景，推荐使用 computed 属性来实现。 因为模板的取值是由客户端实现的，而函数的定义在前端（filter 可以认为是在模板里调用函数的语法糖），如果每次取值都走一次通信的话，会大幅降低渲染性能。

[**不可用**](https://uniapp.dcloud.io/component/recycle-list?id=%e4%b8%8d%e5%8f%af%e7%94%a8)

<cell-slot> 的功能和 <slot> 有部分重叠，而且更为激进，在概念上有冲突，存在很多边界情况无法完全支持。不要在 <cell-slot> 及其子组件里使用 <slot>。

[**v-once 不会优化渲染性能**](https://uniapp.dcloud.io/component/recycle-list?id=v-once-%e4%b8%8d%e4%bc%9a%e4%bc%98%e5%8c%96%e6%b8%b2%e6%9f%93%e6%80%a7%e8%83%bd)

和前端框架中的理解不同，客户端里要实现复用的逻辑，会标记模板节点的状态，添加了 v-once 能保证节点只渲染一次，但是并不一定能优化渲染性能，反而可能会拖慢客户端复用节点时的比对效率。

[**样式功能的限制**](https://uniapp.dcloud.io/component/recycle-list?id=%e6%a0%b7%e5%bc%8f%e5%8a%9f%e8%83%bd%e7%9a%84%e9%99%90%e5%88%b6)

计划支持。目前版本里还不支持绑定样式类名（v-bind:class），原因和进展可以参考 #14。

[**双向绑定**](https://uniapp.dcloud.io/component/recycle-list?id=%e5%8f%8c%e5%90%91%e7%bb%91%e5%ae%9a)

计划支持。v-model 还未调通，暂时不要使用。

[**子组件的限制**](https://uniapp.dcloud.io/component/recycle-list?id=%e5%ad%90%e7%bb%84%e4%bb%b6%e7%9a%84%e9%99%90%e5%88%b6)

没有 Virtual DOM！ 使用在 <recycle-list> 中的组件没有 Virtual DOM！与 Virtual DOM 相关的功能也不支持。在开发过程中尽量只处理数据，不要操作生成后的节点。

下列这些属性都不再有意义，请不要使用：

* vm.$el
* vm.$refs.xxx
* vm.$vnode
* vm.#slots
* vm.#scopedSlots

vm.$refs 里的值可能是数组、子组件的实例、DOM 元素，在前端里比较常用，如果不支持，对 Weex 里的 dom 模块和 animation 模块的功能也有影响。

目前正在讨论技术方案，部分接口可能会重新设计，或者是在 vm 上透出专为 <recycle-list> 设计的接口。

组件的属性 目前子组件的属性不支持函数。（正在讨论实现方案）

<sub-component :prop="item.xxx" />

因为子组件的属性值需要在前端和客户端之间传递，所以仅支持可序列化的值。item.xxx 的类型可以是对象、数组、字符串、数字、布尔值等，不支持函数。

生命周期的行为差异 由于列表的渲染存在回收机制，节点渲染与否也与用户的滚动行为有关，组件的生命周期行为会有一些不一致。

可回收长列表不会立即渲染所有节点，只有即将滚动到可视区域（以及可滚动的安全区域）内时才开始渲染，组件生命周期的语义没变，但是会延迟触发。

假设有 100 条数据，一条数据了对应一个组件。渲染首屏时只能展示 8 条数据的节点，那就只有前 8 个组件被创建了，也只有前 8 个组件的生命周期被触发。

组件的 beforeCreate 和 created 也只有在组件即将创建和创建完成时才会触发 同理，组件的 beforeMount 和 mounted 也只有页面真正渲染到了该组件，在即将挂载和已经挂载时才会触发 组件的自定义事件 计划支持。vm.$on, vm.$once, vm.$emit, vm.$off 等功能还未完全调通，接口可用，但是行为可能有些差异（参数丢失），暂时不要使用。

[**示例**](https://uniapp.dcloud.io/component/recycle-list?id=%e7%a4%ba%e4%be%8b)

<recycle-list for="(item, i) in longList" switch="type">

<cell-slot case="A">

<text>- A {{i}} -</text>

</cell-slot>

<cell-slot case="B">

<text>- B {{i}} -</text>

</cell-slot>

</recycle-list>

如果有如下数据：

const longList = [

{ type: 'A' },

{ type: 'B' },

{ type: 'B' },

{ type: 'A' },

{ type: 'B' }

]

则会生成如下等价节点：

<text>- A 0 -</text>

<text>- B 1 -</text>

<text>- B 2 -</text>

<text>- A 3 -</text>

<text>- B 4 -</text>

如果将模板合并成一个，也可以省略 switch 和 case，将例子进一步简化：

<recycle-list for="(item, i) in longList">

<cell-slot>

<text>- {{item.type}} {{i}} -</text>

</cell-slot>

</recycle-list>

#### [waterfall](https://uniapp.dcloud.io/component/waterfall?id=waterfall)

app端nvue专用组件。

<waterfall> 组件是提供瀑布流布局的核心组件。瀑布流，又称瀑布流式布局是比较流行的一种页面布局，视觉表现为参差不齐的多栏布局。随着页面滚动条向下滚动，这种布局还可以不断加载数据块并附加至当前尾部。

在nvue中，使用普通view做瀑布流，无法实现重用和不可见渲染资源释放。使用组件，指定cell后，原生引擎会自动优化性能。

<template>

<waterfall column-count="2" column-width="auto">

<cell v-for="num in lists" >

<text>{{num}}</text>

</cell>

</waterfall>

</template>

<script>

export default {

data () {

return {

lists: ['A', 'B', 'C', 'D', 'E']

}

}

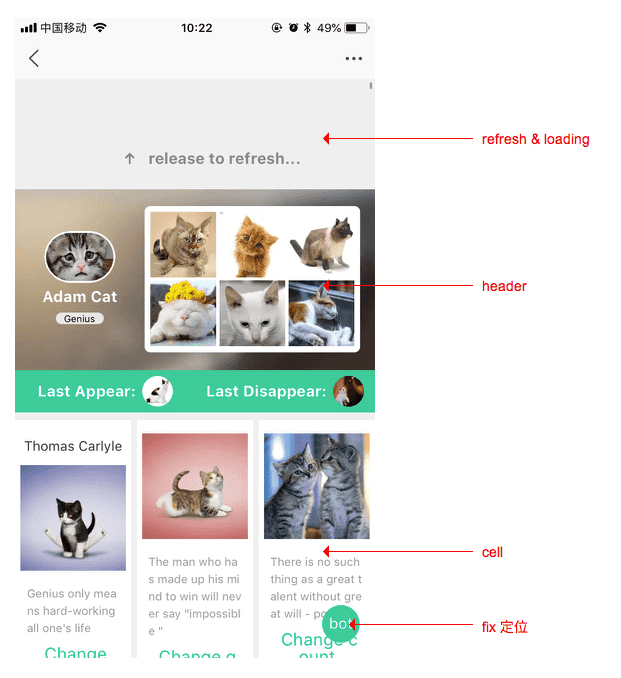
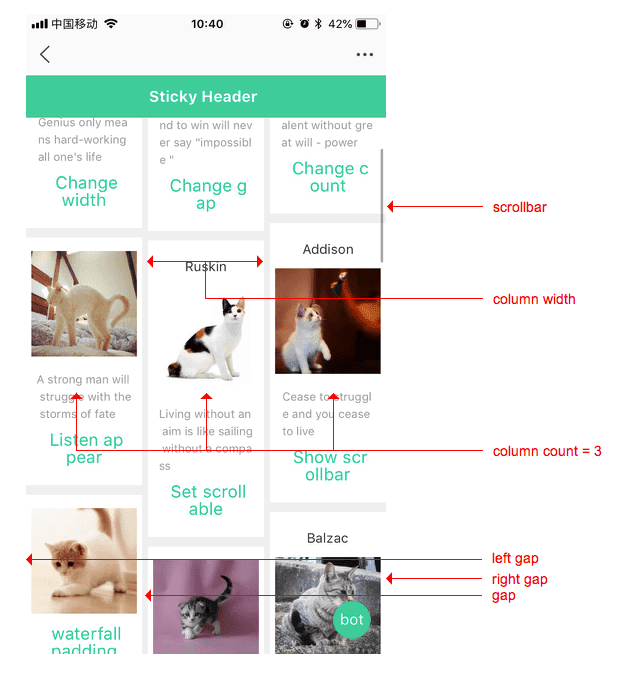
}

</script>

<style></style>

[**子组件**](https://uniapp.dcloud.io/component/waterfall?id=%e5%ad%90%e7%bb%84%e4%bb%b6)

和 <list> 组件一样, <waterfall> 组件的子组件只能包括以下四种组件或是 fix 定位的组件，其他形式的组件将不能被正确渲染。

* <cell>：用于定义列表中的子列表项，类似于 HTML 中的 ul 之于 li。<waterfall> 会对 <cell> 进行高效的内存回收以达到更好的性能。
* <header>：当 <header> 到达屏幕顶部时，吸附在屏幕顶部。
* <refresh>：用于给列表添加下拉刷新的功能。
* <loading>：<loading> 用法与特性和 <refresh> 类似，用于给列表添加上拉加载更多的功能。 
* [属性](https://uniapp.dcloud.io/component/waterfall?id=%e5%b1%9e%e6%80%a7)
* show-scrollbar : [可选] 可选值为 true/ false，默认值为 true。控制是否出现滚动条。
* column-count: [可选]描述瀑布流的列数
  + auto: 意味着列数是被其他属性所决定的(比如 column-width)
  + <integer>: 最佳列数，column-width 和 column-count 都指定非0值， 则 column-count 代表最大列数。
* column-width : [可选]描述瀑布流每一列的列宽
  + auto: 意味着列宽是被其他属性所决定的(比如 column-count)
  + <length>: 最佳列宽，实际的列宽可能会更宽(需要填充剩余的空间)， 或者更窄(如果剩余空间比列宽还要小)。 该值必须大于0
* column-gap: [可选]列与列的间隙. 如果指定了 normal ，则对应 32.
* left-gap: [可选]左边cell和列表的间隙. 如果未指定 ，则对应 0
* right-gap: [可选]右边cell和列表的间隙. 如果未指定，则对应 0 

其他支持的属性参见 <list> 组件属性部分

[**事件**](https://uniapp.dcloud.io/component/waterfall?id=%e4%ba%8b%e4%bb%b6)

支持所有通用事件：

* click：用于监听点击事件。（例如：一般绑定于子组件之上触发跳转）。
* longpress：用于监听长按事件（一般绑定于子组件之上例如：长按可删除）。
* appear：用于监听子组件出现事件（一般绑定于子组件之上例如：监听最后一个元素出现，加载新的数据）
* disappear：用于监听子组件滑出屏幕事件（一般绑定于子组件之上）

**注意**

* waterfall是区域滚动，不会触发页面滚动，无法触发pages.json配置的下拉刷新、页面触底onReachBottomDistance、titleNView的transparent透明渐变。

#### [refresh](https://uniapp.dcloud.io/component/refresh?id=refresh)

app端nvue专用组件。

<refresh> 为容器提供下拉刷新功能。在nvue上，可通过此组件实现灵活的、自定义的、高性能的下拉刷新。

**注意**

* <refresh> 是 <scroll-view>、<list>、<waterfall> 的子组件，只能在被它们包含时才能被正确渲染。

<scroll-view>

<refresh>

<text>Refreshing...</text>

</refresh>

<view v-for="num in lists">

<text>{{num}}</text>

</view>

</scroll-view>

[**子组件**](https://uniapp.dcloud.io/component/refresh?id=%e5%ad%90%e7%bb%84%e4%bb%b6)

* 诸如 <text>、<image> 之类的任何组件，都可以放到 <loading> 进行渲染。
* 特殊子组件 <loading-indicator>: 只能作为 <refresh> 和 <loading> 的子组件使用，拥有默认的动画效果实现。

<refresh>

<text>Refreshing</text>

<loading-indicator></loading-indicator>

</refresh>

[**属性**](https://uniapp.dcloud.io/component/refresh?id=%e5%b1%9e%e6%80%a7)

display 控制 <refresh> 组件显示、隐藏。display 的设置必须成对出现，即设置 display="show", 必须有对应的 display="hide"。可选值为 show / hide，默认值为 show。

[**事件**](https://uniapp.dcloud.io/component/refresh?id=%e4%ba%8b%e4%bb%b6)

* refresh 事件：当 <scroll-view>、<list>、<waterfall> 被下拉完成时触发。
* pullingdown 事件：当 <scroll-view>、<list>、<waterfall> 被下拉时触发。 可以从 event 参数对象中获取以下数据：
  + dy: 前后两次回调滑动距离的差值
  + pullingDistance: 下拉的距离
  + viewHeight: refresh 组件高度
  + type: “pullingdown” 常数字符串

<refresh @refresh="onrefresh" @pullingdown="onpullingdown" :display="refreshing ? 'show' : 'hide'">

<text>Refreshing ...</text>

<loading-indicator></loading-indicator>

</refresh>

# 扩展组件(uni ui)

### [扩展组件（uni-ui）：](https://uniapp.dcloud.io/component/README?id=uniui)

uni-ui是DCloud提供的一个跨端ui库，它是基于vue组件的、flex布局的、无dom的跨全端ui框架。

uni-ui不包括基础组件，**它是基础组件的补充**。

#### [uni ui产品特点](https://uniapp.dcloud.io/component/README?id=uni-ui%e4%ba%a7%e5%93%81%e7%89%b9%e7%82%b9)

1、高性能

目前为止，在小程序和混合app领域，uni ui是性能的标杆。

* 自动差量更新数据

虽然uni-app支持小程序自定义组件，所有小程序的ui库都可以用。但小程序自定义组件的ui库都需要使用setData手动更新数据，在大数据量时、或高频更新数据时，很容易产生性能问题。

而uni ui属于vue组件，uni-app引擎底层自动diff更新数据。当然其实插件市场里众多vue组件都具备这个特点。

* 优化逻辑层和视图层通讯折损

非H5，不管是小程序还是App，不管是app的webview渲染还是原生渲染，全都是逻辑层和视图层分离的。这里就有一个逻辑层和视图层通讯的折损问题。 比如在视图层拖动一个可跟手的组件，由于通讯的损耗，用js监听很难做到实时跟手。

这时就需要使用css动画以及平台底层提供的wxs、bindingx等技术。不过这些技术都比较复杂，所以uni ui里做了封装，在需要跟手式操作的ui组件，比如swiperaction列表项左滑菜单，就在底层使用了这些技术，实现了高性能的交互体验

* 背景停止

很多ui组件是会一直动的，比如轮播图、跑马灯。即便这个窗体被新窗体挡住，它在背景层仍然在消耗着硬件资源。在Android的webview版本为chrome66以上，背景操作ui会引发很严重的性能问题，造成前台界面明显卡顿。

而uni ui的组件，会自动判断自己的显示状态，在组件不再可见时，不会再消耗硬件资源。

2、全端

uni ui的组件都是多端自适应的，底层会抹平很多小程序平台的差异或bug。

比如导航栏navbar组件，会自动处理不同端的状态栏。 比如swiperaction组件，在app和微信小程序上会使用交互体验更好的wxs技术，但在不支持wxs的其他小程序端会使用js模拟类似效果。

uni ui还支持nvue原生渲染，[详见](https://github.com/dcloudio/uni-ui/tree/nvue-uni-ui)

未来uni ui还会支持pc等大屏设备。

3、与uni统计自动集成实现免打点

uni统计是优秀的多端统计平台，见[tongji.dcloud.net.cn](https://tongji.dcloud.net.cn/)。

除了一张报表看全端，它的另一个重要特点是免打点。 比如使用uni ui的navbar标题栏、收藏、购物车等组件，均可实现自动打点，统计页面标题等各种行为数据。 当然你也可以关闭uni统计，这不是强制的。

4、风格扩展

uni ui的默认风格是中型的，与uni-app基础组件风格一致。但它支持[uni.scss](https://uniapp.dcloud.io/collocation/uni-scss)，可以方便的扩展和切换App的风格。

ui是一种需求非常发散的产品，DCloud官方也无意用uni ui压制第三方ui插件的空间，但官方有义务在性能和多端方面提供一个开源的标杆给大家。

我们欢迎更多优秀的ui组件出现，也欢迎更多人贡献uni ui的主题风格，满足更多用户的需求。

uni-ui支持npm安装和zip下载安装2种方式。

#### [uni ui的使用方式](https://uniapp.dcloud.io/component/README?id=uni-ui%e7%9a%84%e4%bd%bf%e7%94%a8%e6%96%b9%e5%bc%8f)

uni ui支持npm安装，和HBuilderX单独导入个别组件等多种使用方式

1. npm安装参考：<https://ext.dcloud.net.cn/plugin?id=55>
2. 下表为uni-ui的扩展组件清单，点击每个组件可单独安装。

| **组件名** | **说明** |
| --- | --- |
| [Badge](https://ext.dcloud.net.cn/plugin?id=21) | 数字角标 |
| [Calendar](https://ext.dcloud.net.cn/plugin?id=56) | 日历 |
| [Card](https://ext.dcloud.net.cn/plugin?id=22) | 卡片 |
| [Collapse](https://ext.dcloud.net.cn/plugin?id=23) | 折叠面板 |
| [Combox](https://ext.dcloud.net.cn/plugin?id=1261) | 可选择输入框 |
| [CountDown](https://ext.dcloud.net.cn/plugin?id=25) | 倒计时 |
| [Drawer](https://ext.dcloud.net.cn/plugin?id=26) | 抽屉 |
| [Fab](https://ext.dcloud.net.cn/plugin?id=144) | 悬浮按钮 |
| [Fav](https://ext.dcloud.net.cn/plugin?id=864) | 收藏按钮 |
| [GoodsNav](https://ext.dcloud.net.cn/plugin?id=865) | 底部购物导航 |
| [Grid](https://ext.dcloud.net.cn/plugin?id=27) | 宫格 |
| [Icons](https://ext.dcloud.net.cn/plugin?id=28) | 图标 |
| [IndexedList](https://ext.dcloud.net.cn/plugin?id=375) | 索引列表 |
| [List](https://ext.dcloud.net.cn/plugin?id=24) | 列表 |
| [LoadMore](https://ext.dcloud.net.cn/plugin?id=29) | 加载更多 |
| [NavBar](https://ext.dcloud.net.cn/plugin?id=52) | 自定义导航栏 |
| [NoticeBar](https://ext.dcloud.net.cn/plugin?id=30) | 通告栏 |
| [NumberBox](https://ext.dcloud.net.cn/plugin?id=31) | 数字输入框 |
| [Pagination](https://ext.dcloud.net.cn/plugin?id=32) | 分页器 |
| [PopUp](https://ext.dcloud.net.cn/plugin?id=329) | 弹出层 |
| [Rate](https://ext.dcloud.net.cn/plugin?id=33) | 评分 |
| [SearchBar](https://ext.dcloud.net.cn/plugin?id=866) | 搜索栏 |
| [SegmentedControl](https://ext.dcloud.net.cn/plugin?id=54) | 分段器 |
| [Steps](https://ext.dcloud.net.cn/plugin?id=34) | 步骤条 |
| [SwipeAction](https://ext.dcloud.net.cn/plugin?id=181) | 滑动操作 |
| [SwiperDot](https://ext.dcloud.net.cn/plugin?id=284) | 轮播图指示点 |
| [Tag](https://ext.dcloud.net.cn/plugin?id=35) | 标签 |

**更多组件**

除了基础组件、uni-ui，插件市场还有更多扩展组件、模板，包括前端组件和原生扩展组件，具体见[插件市场](https://ext.dcloud.net.cn/)。

关于其他vue的web组件库、小程序组件库是否能在uni-app中使用的问题，参考<https://ask.dcloud.net.cn/article/35489>

其他的参考 插件市场 ! ! ! ! ! (<https://ext.dcloud.net.cn/>)

# 配置微信小程序插件

#### [配置微信小程序插件](https://uniapp.dcloud.io/component/mp-weixin-plugin?id=%e9%85%8d%e7%bd%ae%e5%be%ae%e4%bf%a1%e5%b0%8f%e7%a8%8b%e5%ba%8f%e6%8f%92%e4%bb%b6)

#### [注意事项](https://uniapp.dcloud.io/component/mp-weixin-plugin?id=%e6%b3%a8%e6%84%8f%e4%ba%8b%e9%a1%b9)

* 正式开始使用微信小程序插件之前需先在微信公众平台 -> 第三方设置 -> 插件管理处添加插件
* 要保证项目内微信小程序appid和申请插件的微信小程序appid一致插件才可使用

#### [引入插件代码包](https://uniapp.dcloud.io/component/mp-weixin-plugin?id=%e5%bc%95%e5%85%a5%e6%8f%92%e4%bb%b6%e4%bb%a3%e7%a0%81%e5%8c%85)

使用插件之前开发者需要在manifest.json中的mp-weixin内声明使用的插件，具体配置参照所用插件的开发文档

**代码示例**

"plugins": {

"pluginName": {

"version": "1.0.0",

"provider": "wxidxxxxxxxxxxxxxxxx"

}

}

#### [在分包内引入插件代码包](https://uniapp.dcloud.io/component/mp-weixin-plugin?id=%e5%9c%a8%e5%88%86%e5%8c%85%e5%86%85%e5%bc%95%e5%85%a5%e6%8f%92%e4%bb%b6%e4%bb%a3%e7%a0%81%e5%8c%85)

如果插件只在（同一个插件不能被多个分包同时引用）一个分包用到，可以单独配置到分包中，这样插件不会随主包加载，开发者可以在pages.json的[subPackages](https://uniapp.dcloud.io/collocation/pages?id=subpackages)中声明插件

**代码示例**

"subPackages": [{

"root": "pagesA",

"pages": [{

"path": "list/list"

}]

"plugins": {

"pluginName": {

"version": "1.0.0",

"provider": "wxidxxxxxxxxxxxxxxxx"

}

}

}]

**在分包内使用插件的限制**

* 仅能在这个分包内使用该插件；
* 同一个插件不能被多个分包同时引用；
* 不能从分包外的页面直接跳入分包内的插件页面，需要先跳入分包内的非插件页面、再跳入同一分包内的插件页面。

#### [在页面中使用](https://uniapp.dcloud.io/component/mp-weixin-plugin?id=%e5%9c%a8%e9%a1%b5%e9%9d%a2%e4%b8%ad%e4%bd%bf%e7%94%a8)

请参照[微信小程序-使用插件](https://developers.weixin.qq.com/miniprogram/dev/framework/plugin/using.html)

#### [可能遇到的问题](https://uniapp.dcloud.io/component/mp-weixin-plugin?id=%e5%8f%af%e8%83%bd%e9%81%87%e5%88%b0%e7%9a%84%e9%97%ae%e9%a2%98)

* 某些插件可能会需要一些权限才能正常运行，请在manifest.json中的mp-weixin内配置permission[详见](https://developers.weixin.qq.com/miniprogram/dev/reference/configuration/app.html#permission)
* 微信开发工具提示 “插件版本不存在”，可能是插件开发文档示例代码中使用的版本已经不存在，请在声明插件处更改版本

# 原生组件说明

#### [原生组件说明](https://uniapp.dcloud.io/component/native-component?id=%e5%8e%9f%e7%94%9f%e7%bb%84%e4%bb%b6%e8%af%b4%e6%98%8e)

小程序和App的vue页面，主体是webview渲染的。为了提升性能，小程序和App的vue页面下部分ui元素，比如导航栏、tabbar、video、map使用了原生控件。这种方式被称为混合渲染。

虽然提升了性能，但原生组件带来了其他问题：

1. 前端组件无法覆盖原生控件的层级问题
2. 原生组件不能嵌入特殊前端组件(如scroll-view)
3. 原生控件ui无法灵活自定义
4. 原生控件在Android上，字体会渲染为rom的主题字体，而webview如果不经过单独改造不会使用rom主题字体

H5、App的nvue页面，不存在混合渲染的情况，它们或者全部是前端渲染、或者全部是原生渲染，不涉及层级问题。

uni-app 中原生组件清单如下：

* [map](https://uniapp.dcloud.io/component/map)
* [video](https://uniapp.dcloud.io/component/video)
* [camera](https://uniapp.dcloud.io/component/camera)（仅微信小程序、百度小程序支持）
* [canvas](https://uniapp.dcloud.io/component/canvas)（仅在微信小程序、百度小程序表现为原生组件）
* [input](https://uniapp.dcloud.io/component/input)（仅在微信小程序、支付宝小程序、头条小程序、QQ小程序中且input置焦时表现为原生组件，其中支付宝小程序的input仅为text且置焦时才表现为原生组件）
* [textarea](https://uniapp.dcloud.io/component/textarea)（仅在微信小程序、百度小程序、头条小程序表现为原生组件）
* [live-player](https://uniapp.dcloud.io/component/live-player)（仅微信小程序、百度小程序支持，App端直接使用video组件可同时实现拉流）
* [live-pusher](https://uniapp.dcloud.io/component/live-pusher)（仅微信小程序、百度小程序、app-nvue支持，app-vue使用plus.video.LivePusher可实现推流）
* [cover-view](https://uniapp.dcloud.io/component/cover-view)
* [cover-image](https://uniapp.dcloud.io/component/cover-view?id=cover-image)

#### [混合渲染模式下原生组件的使用限制](https://uniapp.dcloud.io/component/native-component?id=%e6%b7%b7%e5%90%88%e6%b8%b2%e6%9f%93%e6%a8%a1%e5%bc%8f%e4%b8%8b%e5%8e%9f%e7%94%9f%e7%bb%84%e4%bb%b6%e7%9a%84%e4%bd%bf%e7%94%a8%e9%99%90%e5%88%b6)

由于原生组件脱离在 WebView 渲染流程外，因此在使用时有以下限制：

* 原生组件的层级是**最高**的，所以页面中的其他组件无论设置 z-index 为多少，都无法盖在原生组件上。后插入的原生组件可以覆盖之前的原生组件。
* 原生组件无法在 scroll-view、swiper、picker-view、movable-view 中使用。
* 同层渲染支持：微信基础库2.4.4起支持了video的同层渲染、微信基础库2.8.3支持map的同层渲染。支持同层渲染后，相关组件的时候不再有层级问题，无需再使用cover-view覆盖，也可以内嵌入swiper等组件。app-nvue 不涉及层级问题，天然所有组件都是同层渲染。
* 部分CSS样式无法应用于原生组件，例如：
  + 无法对原生组件设置 CSS 动画；
  + 无法定义原生组件为 position: fixed；
  + 不能在父级节点使用 overflow: hidden 来裁剪原生组件的显示区域。
* 在小程序端真机上，原生组件会遮挡 vConsole 弹出的调试面板。

#### [其他原生界面元素](https://uniapp.dcloud.io/component/native-component?id=%e5%85%b6%e4%bb%96%e5%8e%9f%e7%94%9f%e7%95%8c%e9%9d%a2%e5%85%83%e7%b4%a0)

除了原生组件外，uni-app在非H5端还有其他原生界面元素，清单如下：

* 原生navigationBar和tabbar（pages.json里配置的）。
* web-view组件虽然不是原生的，但这个组件相当于一个原生webview覆盖在页面上，并且小程序上web-view组件是强制全屏的，无法在上面覆盖前端元素
* 弹出框：picker、showModal、showToast、showLoading、showActionSheet、previewImage、chooseImage、chooseVideo等弹出元素也无法被前端组件覆盖
* plus下的plus.nativeObj.view、plus.video.LivePusher、plus.nativeUI、plus.webview，层级均高于前端元素

注意：app的nvue页面里的组件虽然不涉及map、video等原生组件的层级遮挡问题，但pages.json中配置的原生tabbar、原生navigationBar，一样是nvue里的组件也无法遮挡的。

#### [vue页面层级覆盖解决方案](https://uniapp.dcloud.io/component/native-component?id=vue%e9%a1%b5%e9%9d%a2%e5%b1%82%e7%ba%a7%e8%a6%86%e7%9b%96%e8%a7%a3%e5%86%b3%e6%96%b9%e6%a1%88)

为了解决webview渲染中原生组件层级最高的限制，uni-app提供了 [cover-view](https://uniapp.dcloud.io/component/cover-view) 和 [cover-image](https://uniapp.dcloud.io/component/cover-view?id=cover-image) 组件，让其覆盖在原生组件上。

除了跨端的cover-view，App端还提供了3种方案：plus.nativeObj.view、subNVue、新开半透明nvue页面。详述如下

* [cover-view](https://uniapp.dcloud.io/component/cover-view?id=cover-view)

cover-view只能覆盖原生组件，不能覆盖其他原生界面元素。比如cover-view可以覆盖video、map，但无法覆盖原生导航栏、tabbar、web-view。

微信小程序在基础库 2.4.0 起已支持 video 组件的同层渲染，2.7.0 起支持 map 组件的同层渲染。可以被前端元素通过调节zindex来遮挡，也支持在scroll-view等组件中内嵌这2个原生组件。但video全屏时，仍需要cover-view覆盖。

app-vue的cover-view相比小程序端还有一些限制，1) 无法嵌套、 2) 无法内部滚动，即cover-view无法内部出现滚动条、 3) 无法覆盖到视频的全屏界面。 app-nvue的cover-view无这些限制。

另外cover-view无论如何都无法解决原生导航栏、tabbar、web-view组件的覆盖，为此App端补充了2个层级覆盖方案plus.nativeObj.view和subNVue

* [plus.nativeObj.view](https://www.html5plus.org/doc/zh_cn/nativeobj.html#plus.nativeObj.View)

简称nview，它是一个原生的类画布的控件，其实cover-view也是用plus.nativeObj.view封装的。API文档详见：<https://www.html5plus.org/doc/zh_cn/nativeobj.html#plus.nativeObj.View>

plus.nativeObj.view的API比较原生，可以画出任何界面，但plus.nativeObj.view有3个问题：1. api很底层，开发比较复杂；2. 不支持动画；3. 不支持内部滚动。

* [subNVue](https://ask.dcloud.net.cn/article/35948)

subNVue是把weex渲染的原生子窗体，覆盖在vue页面上，它解决了plus.nativeObj.view的不足，提供了强大的层级问题解决方案。subNVue的详细介绍见：<https://ask.dcloud.net.cn/article/35948>

注意App的非自定义组件编译模式不支持subNVue。

* [弹出部分区域透明的nvue页面](https://ext.dcloud.net.cn/plugin?id=953)

uni-app支持在App端弹出半透明的nvue窗体。即看起来是在本窗体弹出一个元素，但实际上是弹出了一个部分区域蒙灰透明的新窗体。这样的窗体比subnvue有个优势就是可以全局复用。具体参考这个[插件](https://ext.dcloud.net.cn/plugin?id=953)

subNVue或弹出部分区域透明的nvue页面，会比plus.nativeObj.view多占用一些内存。所以如果你要覆盖的内容很简单，cover-view或plus.nativeObj.view可以简单实现的话，就没必要用subNVue或nvue。

所以如果你的层级覆盖问题比较简单，不用嵌套，且有跨端需求，就使用cover-view。

如果App端cover-view无法满足需求，且需要覆盖的原生界面比较简单，可以用plus.nativeObj.view。否则，就使用subnvue或部分区域透明的nvue吧。

**关于subNVue和Webview的层级问题** subNVue的层级高于前端元素，但多个subNVue以及Webview，它们之间的也存在层级关系。

默认规则是，先创建的subNVue或webview在底部，后创建的会盖住之前的。

当然每个subNVue和webview，都支持Style参数配置，其中有一个zindex属性，可以调节它们的层级。

#### [App的nvue页面层级问题](https://uniapp.dcloud.io/component/native-component?id=app%e7%9a%84nvue%e9%a1%b5%e9%9d%a2%e5%b1%82%e7%ba%a7%e9%97%ae%e9%a2%98)

nvue页面全部都是原生组件，互相之间没有层级问题。

但如果在pages.json里注册了原生导航栏和tabbar，nvue里的界面元素默认也无法覆盖这些，也需要plus.nativeObj.view或subNVue。

如果仅开发App，不跨端，不愿涉及层级问题，也可以不使用pages.json里的原生导航栏和tabbar，nvue页面不需要这些来强化性能。

#### [Android系统主题字体对原生组件渲染的影响](https://uniapp.dcloud.io/component/native-component?id=android%e7%b3%bb%e7%bb%9f%e4%b8%bb%e9%a2%98%e5%ad%97%e4%bd%93%e5%af%b9%e5%8e%9f%e7%94%9f%e7%bb%84%e4%bb%b6%e6%b8%b2%e6%9f%93%e7%9a%84%e5%bd%b1%e5%93%8d)

在Android手机上，调整系统主题字体，所有原生渲染的控件的字体都会变化，而webview渲染的字体则不会变化。

如果原生渲染和webview渲染出现在同一页面，就会发现字体不一致。

部分小程序通过修改了自带的webview内核，实现了webview也可以使用rom主题字体，比如微信、qq、支付宝；其他小程序及app-vue下，webview仍然无法渲染为rom主题字体。

不管Android字体问题、还是同层渲染问题，微信小程序都是依靠自带一个几十M的定制webview实现的，这对于App而言增加了太大的体积，不现实。

app端若在意字体不一致的问题，有2种解决建议：

1. 直接使用nvue。nvue是纯原生渲染，不存在webview渲染和原生字体不一致的问题。
2. app端不使用系统webview，而是使用x5浏览器内核。详见：<https://ask.dcloud.net.cn/article/36806>