

# ZanUI 微信小程序 中文文档

书栈(BookStack.CN)

# 目 录

致谢

基础

Icon 图标

Button 按钮

Helper 基础样式

Loading 加载

布局

Layout 布局

Cell 单元格

Card 卡片

Panel 面板组件

表单

Field 输入框

Switch 开关

Select 选择

Stepper 计数器

展示

Tag 标签

Badge 徽章

Capsule 胶囊

Noticebar 通告栏

Steps 步骤条

交互

Actionsheet 行动按钮

DatetimePicker 时间选择

Dialog 弹出框

Popup 弹出层

Tab 标签

Toast 轻提示

TopTips 顶部提示

Loadmore 加载

## 致谢

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

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

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

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

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

文档地址：<http://www.bookstack.cn/books/ZanUI>

书栈官网：<http://www.bookstack.cn>

书栈开源：<https://github.com/TruthHun>

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

## 基础

- [Icon 图标](#)
- [Button 按钮](#)
- [Helper 基础样式](#)
- [Loading 加载](#)

## Icon 图标

## Icon 图标

---

### 使用指南

在 index.json 中引入组件

```
1.  {  
2.    "usingComponents": {  
3.      "zan-icon": "path/to/zanui-weapp/dist/icon/index"  
4.    }  
5.  }
```

### 代码演示

可以在任意位置上使用 zan-icon 标签。通过 type 可以控制 icon 显示的图标

```
1. <zan-icon type="success"></zan-icon>
```

支持的 icon 和 名称 见下图



zan-icon-close



zan-icon-location



zan-icon-clock



zan-icon-gold-coin



zan-icon-chat



zan-icon-exchange



zan-icon-upgrade



zan-icon-edit



zan-icon-contact



zan-icon-passed



zan-icon-points



zan-icon-delete



zan-icon-records



zan-icon-logistics



zan-icon-check



zan-icon-checked



zan-icon-gift



zan-icon-like-o



zan-icon-wap-nav



zan-icon-password-not-view



zan-icon-wap-home



zan-icon-ecard-pay



zan-icon-balance-pay



zan-icon-peer-pay



zan-icon-credit-pay



zan-icon-debit-pay



zan-icon-other-pay



zan-icon-shopping-cart



zan-icon-browsing-history



zan-icon-goods-collect



zan-icon-shop-collect



zan-icon-receive-gift



zan-icon-send-gift



zan-icon-setting



zan-icon-coupon



zan-icon-free-postage



zan-icon-wap-nav



zan-icon-password-not-view



zan-icon-wap-home



zan-icon-ecard-pay



zan-icon-balance-pay



zan-icon-peer-pay



zan-icon-credit-pay



zan-icon-debit-pay



zan-icon-other-pay



zan-icon-shopping-cart



zan-icon-browsing-history



zan-icon-goods-collect



zan-icon-shop-collect



zan-icon-receive-gift



zan-icon-send-gift



zan-icon-setting



zan-icon-coupon



zan-icon-free-postage





zan-icon-discount



zan-icon-birthday-privilege



zan-icon-member-day-privilege



zan-icon-balance-details



zan-icon-cash-back-record



zan-icon-points-mall



zan-icon-exchange-record



zan-icon-pending-payment



zan-icon-pending-orders



zan-icon-pending-deliver



zan-icon-pending-evaluate



zan-icon-cash-on-deliver



zan-icon-gift-card-pay



zan-icon-underway



zan-icon-point-gift



zan-icon-after-sale



zan-icon-edit-data



zan-icon-question



zan-icon-description



zan-icon-card



zan-icon-gift-card



zan-icon-completed



zan-icon-value-card



zan-icon-certificate



zan-icon-to-send



zan-icon-sign



zan-icon-home



zan-icon-phone



zan-icon-play



zan-icon-pause



zan-icon-stop



zan-icon-hot



zan-icon-new



zan-icon-new-arrival



zan-icon-hot-sale



Icon 图标

微信扫一扫

原文：

<https://www.youzanyun.com/zanui/weapp#/zanui/base/icon>

# Button 按钮

## Button 按钮

### 使用指南

在 json 文件中配置button组件

```
1. "usingComponents": {  
2.   "zan-button": "/dist/btn/index"  
3. }
```

### 代码演示

#### 基础用法

```
1. <zan-button>取消订单</zan-button>
```

#### 按钮类型

按钮支持额外的三种类型 primary, danger, warn

```
1. <zan-button type="primary">确认付款</zan-button>  
2. <zan-button type="danger">确认付款</zan-button>  
3. <zan-button type="warn">确认付款</zan-button>
```

#### 按钮大小

按钮支持额外三种大小 large, small, mini

```
1. <zan-button size="large">确认付款</zan-button>  
2. <zan-button size="small">取消订单</zan-button>  
3. <zan-button size="mini">确认付款</zan-button>
```

#### 其他

##### 按钮镂空状态

```
1. <zan-button plain>确认付款</zan-button>
```

## 按钮加载状态

```
1. <zan-button loading>确认付款</zan-button>
```

## 按钮禁用状态

```
1. <zan-button disabled>确认付款</zan-button>
```

## 配合 button-group 使用

通过配合 `zan-button-group` 使用，可以让按钮之间自动有合适的间距出现，使用方式如下

### 1. 在 json 文件中配置 button-group 组件

```
1. "usingComponents": {  
2.   "zan-button": "/dist/btn/index",  
3.   "zan-button-group": "/dist/btn-group/index"  
4. }
```

### 2. 在 wxml 中直接引入

```
1. <zan-button-group>  
2.   <zan-button>确认付款</zan-button>  
3.   <zan-button>再考虑下</zan-button>  
4. </zan-button-group>
```



属性

名称	类型	是否必须	默认	描述
type	String	否	空	按钮类型，值有primary、warn、danger
size	String	否	空	按钮大小，值有large、small、mini
plain	Boolean	否	false	按钮是否镂空，默认为false
disabled	Boolean	否	false	按钮是否禁用，默认为false
loading	Boolean	否	false	按钮加载状态，默认为false
openType	String	否	-	微信开放能力
appParameter	String	否	-	打开 APP 时，向 APP 传递的参数
hoverStartTime	Number	否	20	按住后多久出现点击态，单位毫秒
hoverStayTime	Number	否	70	手指松开后点击态保留时间，单位毫秒
lang	String	否	en	指定返回用户信息的语言，zh_CN 简体中文，zh_TW 繁体中文，en 英文
sessionFrom	String	否	-	会话来源
sendMessageTitle	String	否	当前标	会话内消息卡片标题

sendMessageTitle	String	否	题	会话内消息卡片标题
sendMessagePath	String	否	当前分 享路径	会话内消息卡片点击跳转小程序路径
sendMessageImg	String	否	截图	会话内消息卡片图片
showMessageCard	String	否	false	显示会话内消息卡片

## 事件

事件名称	说明	回调 参数
btnclick	按钮在可用状态被点击时触发	
disabledclick	在传入的 disabled 为 true 时，点击按钮会触发此事件	
getuserinfo	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wx.getUserInfo	
contact	客服消息回调	
getphonenumber	获取用户手机号回调	
error	当使用开放能力时，发生错误的回调	



微信扫一扫

原文：

<https://www.youzanyun.com/zanui/weapp#/zanui/base/btn>

# Helper 基础样式

## Helper 基础样式

### 使用指南

在 `app.wxss` 中引入组件库所有样式

```
1. @import "path/to/zanui-weapp/dist/index.wxss";
```

### 代码演示

直接在元素上增加指定 `class` 即可

```
1. <view class="zan-pull-right">zan-pull-right: 往右靠</view>
```

具体可用类名如下：

#### 字体大小

- |                          |             |
|--------------------------|-------------|
| <code>zan-font-8</code>  | 文字以8像素大小展示  |
| <code>zan-font-10</code> | 文字以10像素大小展示 |
| <code>zan-font-12</code> | 文字以12像素大小展示 |
| <code>zan-font-14</code> | 文字以14像素大小展示 |
| <code>zan-font-16</code> | 文字以16像素大小展示 |
| <code>zan-font-18</code> | 文字以18像素大小展示 |
| <code>zan-font-20</code> | 文字以20像素大小展示 |
| <code>zan-font-22</code> | 文字以22像素大小展示 |
| <code>zan-font-24</code> | 文字以24像素大小展示 |
| <code>zan-font-30</code> | 文字以30像素大小展示 |

#### 字体颜色

- |                        |         |
|------------------------|---------|
| <code>zan-c-red</code> | 文字以红色展示 |
|------------------------|---------|



`zan-c-gray` 文字以浅灰色展示

`zan-c-gray-dark` 文字以灰色展示

`zan-c-gray-darker` 文字以深灰色展示

`zan-c-black` 文字以黑色展示

`zan-c-blue` 文字以蓝色展示

`zan-c-green` 文字以绿色展示

## 字体样式

`zan-pull-right` 文字往右靠

`zan-text-deleted` 文字显示删除效果

`zan-font-bold` 文字加粗显示

## 其他

`zan-arrow` 展示向右侧箭头，以 `absolute` 布局，需要在外层加上 `relative` 来定位

`zan-ellipsis` 文字过长点点点显示

`zan-ellipsis-12` 文字过长点点点显示，最多显示两行

`zan-ellipsis-13` 文字过长点点点显示，最多显示三行



微信扫一扫

原文：

<https://www.youzanyun.com/zanui/weapp#/zanui/base/helper>

# Loading 加载

## Loading 加载

### 使用指南

在 index.json 中引入组件

```
1.  {
2.    ...
3.    "usingComponents": {
4.      "zan-loading": "../../dist/loading/index"
5.    }
6.    ...
7.  }
```

在页面上直接使用 zan-loading 标签即可

```
1.  <zan-loading type="circle"></zan-loading>
2.  <zan-loading type="spinner" color="black"></zan-loading>
3.  <zan-loading type="dot"></zan-loading>
```

### 具体参数

名称	类型	是否必须	默认	描述
type	String	否	circle	loading 类型, 可支持 circle, spinner, dot
color	String	否	无	可选值 black
use	String, Number	否	1	选择每种 Loading 类型的样式



微信扫一扫

原文：

<https://www.youzanyun.com/zanui/weapp#/zanui/base/loading>

# 布局

- [Layout](#) 布局
- [Cell](#) 单元格
- [Card](#) 卡片
- [Panel](#) 面板组件

# Layout 布局

## Layout 布局

### 使用指南

在 index.json 中引入组件

```
1.  {
2.    "usingComponents": {
3.      "zan-row": "path/to/zanui-weapp/dist/row/index",
4.      "zan-col": "path/to/zanui-weapp/dist/col/index"
5.    }
6.  }
```

### 代码演示

Layout 组件提供了24列栅格，设置 col 属性可以设置元素所占宽度

```
1. <zan-row>
2.   <zan-col col="8" col-class="custom-zan-col">span: 8</zan-col>
3.   <zan-col col="8" col-class="custom-zan-col">span: 8</zan-col>
4.   <zan-col col="8" col-class="custom-zan-col">span: 8</zan-col>
5. </zan-row>
```

Layout 提供了 offset 功能。设置 offset 属性可以设置列的偏移宽度，计算方式与 span 相同

```
1. <zan-row row-class="custom-zan-row">
2.   <zan-col col="4" col-class="custom-zan-col">span: 4</zan-col>
3.   <zan-col col="10" offset="4" col-class="custom-zan-col">offset: 4, span:
4.     10</zan-col>
5. </zan-row>
6. <zan-row>
7.   <zan-col col="12" offset="12" col-class="custom-zan-col">offset: 12, span:
8.     12</zan-col>
9. </zan-row>
```

### API

Row

参数	说明	类型	默认值
row-class	自定义row class	String	-

Col

参数	说明	类型	默认值
col-class	自定义col class	String	-
col	元素所占宽度	Number	<input type="text" value="0"/>
offset	元素偏移宽度	Number	<input type="text" value="0"/>



微信扫一扫

原文：

<https://www.youzanyun.com/zanui/weapp#/zanui/layout/row>

# Cell 单元格

## Cell 单元格

### 使用指南

#### 单个 cell 使用示例

```
1.  {
2.    ...
3.    "usingComponents": {
4.      "zan-cell": "../..../dist/cell/index",
5.    }
6.    ...
7.  }
```

```
1.  <zan-cell title="单行列表" label="附加描述" value="详细信息"></zan-cell>
```

#### cell 组使用示例

多个 cell 组件必须作为 `cell-group` 组件的子组件，否则可能出现显示问题。

```
1.  {
2.    ...
3.    "usingComponents": {
4.      "zan-cell": "../..../dist/cell/index",
5.      "zan-cell-group": "../..../dist/cell-group/index"
6.    }
7.    ...
8.  }
```

```
1.  <zan-cell-group>
2.    <zan-cell title="只显示箭头" is-link></zan-cell>
3.    <zan-cell title="跳转到首页" is-link url="/pages/dashboard/index"></zan-
  cell>
4.    <zan-cell title="单行列表" label="附加描述" value="详细信息"></zan-cell>
5.    <zan-cell title="表单">
6.      <input slot="footer" type="digit" placeholder="带小数点的数字键盘"/>
7.    </zan-cell>
```



```
8.      <zan-cell title="开关">
9.          <switch slot="footer" checked/>
10.     </zan-cell>
11. </zan-cell-group>
```

属性与事件

名称	类型	是否必须	默认	描述
title	String	否	无	左侧标题
label	Boolean	否	false	标题下方的描述信息
value	String	否	取消	右侧内容
isLink	Boolean	否	false	是否展示右侧箭头并开启尝试以 url 跳转
url	String	否	-	当 isLink 设置为 true 时，点击 cell 会尝试路径
linkType	String	否	navigateTo	链接跳转类型，可选值为 <code>navigateTo</code> ， <code>redirectTo</code> ， <code>switchTab</code> ，
onlyTapFooter	Boolean	否	false	只有点击 footer 区域才触发 tab 事件
bindtap	EventHandle	否	无	点击 cell 时触发， <code>onlyTapFooter</code> 为 <code>true</code> 时，仅在 footer 区域触发

可用的 slot

名称	是否必须	默认	描述
默认	否	无	左侧除了 <code>title</code> ， <code>label</code> 外的自定义 wxml 内容
icon	否	无	标题前自定义的 icon，可使用 <code>icon</code> 自定义组件，具体使用参考 icon 组件
footer	否	无	右侧自定义 wxml 内容，如果设置了 <code>value</code> 属性，则不生效

原文：

```
https://www.youzanyun.com/zanui/weapp#/zanui/layout/cell
```

# Card 卡片

## Card 卡片

### 使用指南

在 `index.json` 中引入组件

```
1.  {
2.    "usingComponents": {
3.      "zan-card": "path/to/zanui-weapp/dist/card/index"
4.    }
5.  }
```

### 代码演示

#### 基础用法

卡片可以用于左侧图片，右侧描述信息的展示。默认是商品相关内容的展示，需要展示其他内容可以使用自定义slot。

```
1. <zan-card
2.   card-class="test-card"
3.
4.   thumb="https://img.yzcdn.cn/upload_files/2016/11/25/FpqPXlrMRjKwJs8VdTU3ZDJCj4j5
   imageView2/2/w/200/h/200/q/90/format/jpeg"
5.   price="999.99"
6.   title="红烧牛肉【虚拟商品】【有库存】【有sku】 "
7.   num="2"
8.   desc="3000克 50%"
9.   status="已发货"
10. >
11. </zan-card>
```

#### 使用slot

`zan-card` 由 `zan-cardthumb` 和 `zan-card detail` 组成。分别负责左侧图片展示和右侧内容区域展示。两部分内容可以使用slot进行替换。

```
1. <zan-card
2.   card-class="test-card"
3.   useDetailSlot="{{ true }}"
4. >
5.   <!-- 右侧详情 -->
6.   <view slot="detail-slot" class="zan-card__detail">
7.     我是标题
8.   </view>
9. </zan-card>
```

API

参数	说明	类型	默认值
card-class	自定义最外层class	String	-
thumb	左侧缩略图	String	-
price	商品价格	String	-
title	商品标题	String	-
desc	商品描述	String	-
num	商品数量	Number	-
status	商品状态	String	-
useDetailSlot	是否使用detail-slot ( true时需要添加对应slot )	Boolean	false



微信扫一扫

原文：

<https://www.youzanyun.com/zanui/weapp#/zanui/layout/card>

# Panel 面板组件

## Panel 面板组件

### 使用指南

在 index.json 中引入组件

```
1.  {
2.    "usingComponents": {
3.      "zan-panel": "/packages/panel/index"
4.    }
5.  }
```

### 代码演示

Panel 提供了一块白色的展示区域，使用方式如下

```
1. <zan-panel title='我是标题'>
2.   <view>内容</view>
3. </zan-panel>
```

参数	说明	类型	默认值	必须
title	panel的标题	String	-	
hide-border	内容区隐藏边框	Boolean	-	



微信扫一扫

原文：

<https://www.youzanyun.com/zanui/weapp#/zanui/layout/panel>

## 表单

- [Field](#) 输入框
- [Switch](#) 开关
- [Select](#) 选择
- [Stepper](#) 计数器

# Field 输入框

## Field 输入框

### 使用指南

在 index.json 中引入组件

```
1.  {
2.    "usingComponents": {
3.      "zan-field": "path/to/zanui-weapp/dist/field/index"
4.    }
5.  }
```

### 代码演示

#### 基础用法

field 支持多种展示方式，在 `data` 中传入对应的设置即可。

```
1. <zan-field
2.   title="{{ field.title }}"
3.   placeholder="{{ field.placeholder }}"
4.   focus="{{ field.focus }}"
5.   value="{{ field.value }}"
6.   bind:change="handleFieldChange"
7. >
8. </zan-field>
```

```
1. Page({
2.   data: {
3.     field: {
4.       focus: true,
5.       title: '收货人',
6.       placeholder: '名字',
7.       value: 'test'
8.     }
9.   }
10. });
```

## 监听事件

field会触发一些事件，当你需要监听这些事件时，可以绑定对应的事件。

```
1. <zan-field
2.   title="{{ field.title }}"
3.   placeholder="{{ field.placeholder }}"
4.   focus="{{ field.focus }}"
5.   value="{{ field.value }}"
6.   bind:change="handleFieldChange"
7.   bind:focus="handleFieldFocus"
8.   bind:blur="handleFieldBlur"
9. >
10. </zan-field>
```

```
1. Page.extend({}, {
2.   data: {
3.     field: {
4.       focus: true,
5.       title: '收货人',
6.       placeholder: '名字',
7.       value: 'test'
8.     }
9.   },
10.
11.   methods: {
12.     handleFieldChange(event) {
13.       console.log(event);
14.     },
15.
16.     handleFieldFocus(event) {
17.       console.log(event);
18.     },
19.
20.     handleFieldBlur(event) {
21.       console.log(event);
22.     }
23.   }
24. }));
```

## API



参数	说明	类型	默认值	必须
title	输入框左侧标题，若传入为空，则不显示标题	String	-	
name	输入框的名字，作为 form 表单提交时数据的 key	String	componentId 指定的值	
value	输入框的内容	String	-	
type	输入框的类型，可选值为 input, textarea	String	input	
inputType	输入框为 input 情况下，输入框的类型，例如：number, text, password	String	text	
placeholder	输入框为空时占位符	String		
maxlength	最大输入长度，设置为 -1 的时候不限制最大长度	Number	140	
focus	自动聚焦，拉起键盘	Boolean	false	
disabled	输入框是否禁用	Boolean	false	
mode	输入框展示样式，可选值为 wrapped, normal	String	normal	
right	输入框内容是否居右显示	Boolean	false	
error	是否显示为输入框错误情况下的样式	Boolean	false	
componentId	用于区分输入框之间的唯一名称	String	-	

## Event

事件名称	说明	回调参数
change	当绑定值变化时触发的事件	event对象
focus	输入框focus	event对象
blur	输入框blur	event对象



微信扫一扫

原文：

<https://www.youzanyun.com/zanui/weapp#/zanui/form/field>

# Switch 开关

## Switch 开关

---

### 使用指南

在 index.json 中引入组件

```
1.  {
2.    "usingComponents": {
3.      "zan-switch": "/packages/switch/index"
4.    }
5.  }
```

### 代码演示

=====

```
1. <zan-switch
2.   disabled="{{ disabled }}"
3.   checked="{{ checked }}"
4.   loading="{{ loading }}"
5.   bind:change="handleFieldChange"
6. ></zan-switch>
```

```
1. Page({
2.   data: {
3.     disabled: false,
4.     checked: false,
5.     loading: false
6.   },
7.
8.   methods: {
9.     handleFieldChange(event, data) {
10.       console.log(event, data);
11.     }
12.   }
13. });
```

## API

参数	说明	类型	默认值	必须
loading	switch 是否是 loading 状态	Boolean	false	
disabled	是否不可用	Boolean	false	
checked	是否打开状态	Boolean	false	必须

## Event

事件名称	说明	回调参数
change	当绑定值变化时触发的事件	event对象和数据对象（包含loading和checked）



微信扫一扫

原文：

<https://www.youzanyun.com/zanui/weapp#/zanui/form/switch>

# Select 选择

## Select 选择

---

### 使用指南

在 index.json 中引入组件

```
1.  {
2.    "usingComponents": {
3.      "zan-select": "/packages/select/index"
4.    }
5.  }
```

### 代码演示

```
1. <zan-select
2.   items="{{ items }}"
3.   checkedValue="{{ checkedValue }}"
4.   activeColor="{{ activeColor }}"
5.   bind:change="handleSelectChange"
6. />
```

```
1. Page({
2.   data: {
3.     items: [
4.       {
5.         value: '1',
6.         // 选项文案
7.         name: '选项一',
8.       },
9.       {
10.        value: '2',
11.        name: '选项二',
12.      },
13.    ],
14.    checkedValue: '选项一',
15.    activeColor: '#ff4443'
16.  },
```

```
17.  
18.   methods: {  
19.     handleSelectChange({ detail }) {  
20.       console.log(detail);  
21.     }  
22.   }  
23. });
```

## 具体参数

参数	说明	类型	默认值	必须
items	select 显示各个项的配置	Array	-	
checkedValue	高亮的 item 的 value 值	String	-	
activeColor	Select 高亮颜色	String	#ff4444	

items 具体格式如下

```
1.  {  
2.    items: [  
3.      {  
4.        // 选项选中时，代表的选中值。会以此作为唯一值，判断是否选中  
5.        value: '1',  
6.        // 选项的文字描述  
7.        name: '选项一',  
8.      },  
9.      {  
10.       value: '2',  
11.       name: '选项二',  
12.     },  
13.   ]  
14. }
```



微信扫一扫

原文：

<https://www.youzanyun.com/zanui/weapp#/zanui/form/select>

# Stepper 计数器

## Stepper 计数器

### 使用指南

在 `index.json` 中引入组件

```
1.  {
2.    "usingComponents": {
3.      "zan-stepper": "path/to/zanui-weapp/dist/stepper/index"
4.    }
5.  }
```

### 代码演示

#### 基础用法

**Stepper** 组件通过传入的 `stepper` 对象控制，内部数据格式如下：

```
1.  Page({
2.    data: {
3.      stepper: {
4.        // 当前 stepper 数字
5.        stepper: 1,
6.        // 最小可到的数字
7.        min: 1,
8.        // 最大可到的数字
9.        max: 1,
10.       // 小尺寸，默认大尺寸
11.       size: 'small'
12.     }
13.   },
14.
15.   handleZanStepperChange({
16.     // stepper 代表操作后，应该要展示的数字，需要设置到数据对象里，才会更新页面展示
17.     detail: stepper
18.   }) {
19.     this.setData({
20.       'stepper.stepper': stepper
```



```
21.     });  
22.   }  
23. });
```

当一个 `Stepper` 中，min 超过 max，就会导致组件被置灰。

当 stepper 被点击时，需要监听 `change` 事件，处理计数器值的改变。

```
1. <zan-stepper  
2.   stepper="{{ stepper.stepper }}"  
3.   min="{{ stepper.min }}"  
4.   max="{{ stepper.max }}"  
5.   bind:change="handleZanStepperChange"  
6. >  
7. </zan-stepper>
```

## API

参数	说明	类型	默认值	必须
size	计数器尺寸	String	-	
stepper	计数器的值	Number	1	必须
min	计数器最小值	Number	1	
max	计数器最大值	Number	无穷大	
step	步数	Number	1	

## Event

事件名称	说明	回调参数
change	当绑定值变化时触发的事件	{ index, stepper }
minus	点击减少按钮时触发	-
plus	点击增加按钮时触发	-



微信扫一扫

原文：

<https://www.youzanyun.com/zanui/weapp#/zanui/form/stepper>

## 展示

- [Tag](#) 标签
- [Badge](#) 徽章
- [Capsule](#) 胶囊
- [Noticebar](#) 通告栏
- [Steps](#) 步骤条

# Tag 标签

## Tag 标签

### 使用指南

在 json 文件中配置tag组件

```
1.  "usingComponents": {
2.    "zan-tag": "/dist/tag/index"
3.  }
```

### 属性

名称	类型	是否必须	默认	描述
type	String	否	空	tag类型，值有primary、warn、danger
plain	Boolean	否	false	tag是否镂空，默认为false
disabled	Boolean	否	false	tag是否禁用，默认为false

### 代码演示

#### 基础用法

```
1.  <zan-tag>取消订单</zan-tag>
```

#### 类型

tag支持额外的三种类型 primary, danger, warn

```
1.  <zan-tag type="primary">会员折扣</zan-tag>
2.  <zan-tag type="danger">返现</zan-tag>
3.  <zan-tag type="warn">返现</zan-tag>
```

#### 其他

tag镂空状态

```
1.  <zan-tag plain>返现</zan-tag>
```

## tag禁用状态

1. `<zan-tag disabled>不可用</zan-tag>`



微信扫一扫

原文：

<https://www.youzanyun.com/zanui/weapp#/zanui/view/tag>

# Badge 徽章

## Badge 徽章

---

### 使用指南

在 index.json 中引入组件

```
1.  {
2.    "usingComponents": {
3.      "zan-badge": "path/to/zanui-weapp/dist/badge/index"
4.    }
5.  }
```

### 代码演示

#### 基础用法

```
1. <view class="badge-container">
2.   <zan-badge>10</zan-badge>
3. </view>
```

#### 自定义参数

```
1. <view class="badge-container">
2.   <zan-badge
3.     color="{{ color }}"
4.     background-color="{{ backgroundColor }}"
5.     font-size="{{ fontSize }}"
6.     box-shadow="{{ boxShadow }}"
7.   >10</zan-badge>
8. </view>
```

```
1. .badge-container {
2.   width: 100px;
3.   height: 100px;
4. }
```

# API

参数	说明	类型	默认值
color	字体颜色	String	#fff
background-color	背景颜色	String	#f44
font-size	字体大小	Number	10
box-shadow	为了更好的控制宽度，使用了box-shadow来实现badge的边框，可以根据box-shadow的语法自行修改颜色和宽度	String	0 0 0 2px #fff



微信扫一扫

原文：

<https://www.youzanyun.com/zanui/weapp#/zanui/view/badge>

# Capsule 胶囊

## Capsule 胶囊

### 使用指南

在 index.json 中引入组件

```
1.  {
2.    "usingComponents": {
3.      "zan-capsule": "/packages/capsule/index"
4.    }
5.  }
```

### 代码演示

Panel 提供了一块白色的展示区域，使用方式如下

```
1. <zan-capsule color="#38f" leftText="1折扣" rightText="限购一份" />
```

参数	说明	类型	默认值	必须
type	capsule的主体颜色	String	''（有danger这个主题色）	
color	自定义capsule颜色	String	-	
leftText	左侧文案	String	-	
rightText	右侧文案	String	-	





Capsule 胶囊

微信扫一扫

原文：

<https://www.youzanyun.com/zanui/weapp#/zanui/view/capsule>

# Noticebar 通告栏

## Noticebar 通告栏

### 使用指南

在 `index.json` 中引入组件

```
1.  {
2.    "usingComponents": {
3.      "zan-noticebar": "path/to/zanui-weapp/dist/noticebar/index"
4.    }
5.  }
```

在 `index.js` 中声明组件数据

```
1.  // 在 Page 中声明 Noticebar 依赖的展示数据
2.  Page({
3.    data: {
4.      text: 'xxx',
5.      scrollable: 'xxx',
6.      ...
7.    }
8.  })
```

### 代码演示

**Noticebar** 组件支持滚动和静止两种展示方式，通过 `text` 传入展示文案

### 静止公告栏

```
1. <zan-noticebar
2.   text="{{ text }}"
3. />
```

### 滚动通告栏

```
1. <zan-noticebar
```

```

2.   text="{{ text }}"
3.   scrollable="true"
4. />

```

## 延时滚动通告栏

```

1. <zan-noticebar
2.   text="{{ text }}"
3.   scrollable="true"
4.   delay="{{ delay }}"
5. />

```

## 改变滚动通告栏滚动速度

```

1. <zan-noticebar
2.   text="{{ text }}"
3.   scrollable="true"
4.   speed="{{ speed }}"
5. />

```

## 自定义通告栏字体颜色和背景色

```

1. <zan-noticebar
2.   text="{{ text }}"
3.   color="{{ color }}"
4.   background-color="{{ backgroundColor }}"
5. />

```

## 添加左侧icon通告栏

```

1. <zan-noticebar
2.   text="{{ text }}"
3.   left-
      icon="https://img.yzcdn.cn/public_files/2017/8/10/6af5b7168eed548100d9041f07b7c6c"
4. />

```

## 可关闭通告栏

```
1. <zan-noticebar
2.   text="{{ text }}"
3.   mode="closeable"
4. />
```

参数	说明	类型	默认值	可选值
text	通告栏展示文案	String	-	
mode	通告栏模式	String	' '	closeable
delay	滚动延时时间	Number	0	
speed	滚动速度	Number	40	
scrollable	是否可滚动	Boolean	false	
leftIcon	左侧图标	String	-	
color	通告栏字体颜色	String	#f60	
backgroundColor	通告栏背景色	String	#fff7cc	



微信扫一扫

原文：

<https://www.youzanyun.com/zanui/weapp#/zanui/view/noticebar>

# Steps 步骤条

## Steps 步骤条

### 使用指南

在 app.wxss 中引入组件库所有样式

```
1. @import "path/to/zanui-weapp/dist/index.wxss";
```

在 index.json 中引入组件

```
1. {
2.   "usingComponents": {
3.     "zan-steps": "path/to/zanui-weapp/dist/steps/index"
4.   }
5. }
6.
7. ### 代码演示
8. 在模板中使用 zan-steps 模板，并传入相应数据
9. ```html
10. <zan-steps type="horizon" steps="{steps}"></zan-steps>
```

参数	说明	类型	默认值	必须
type	steps 的展示状态，可选值为 'horizon', 'vertical'	String	horizon	
hasDesc	是否展示描述	Boolean	false	
steps	步骤条展示数据	Array		必须
className	自定义类目，方便自定义显示	String		

steps 数据格式如下：

```
1. [
2.   {
3.     // 此步骤是否当前完成状态
4.     current: false,
5.     // 此步骤是否已经完成
6.     done: true,
```

```
7.      // 此步骤显示文案
8.      text: '步骤一',
9.      // 此步骤描述语
10.     desc: '10.01'
11.   },
12.   {
13.     done: true,
14.     current: false,
15.     text: '步骤二',
16.     desc: '10.02'
17.   },
18.   {
19.     done: true,
20.     current: true,
21.     text: '步骤三',
22.     desc: '10.03'
23.   }
24. ]
```



微信扫一扫

原文：

<https://www.youzanyun.com/zanui/weapp#/zanui/view/steps>

## 交互

- [Actionsheet](#) 行动按钮
- [DatetimePicker](#) 时间选择
- [Dialog](#) 弹出框
- [Popup](#) 弹出层
- [Tab](#) 标签
- [Toast](#) 轻提示
- [TopTips](#) 顶部提示
- [Loadmore](#) 加载

# Actionsheet 行动按钮

## Actionsheet 行动按钮

### 使用指南

在 index.json 中引入组件

```
1.  {
2.    "usingComponents": {
3.      "zan-actionsheet": "path/to/zanui-weapp/dist/actionsheet/index"
4.    }
5.  }
```

### 使用指南

```
1.  <button bindtap="openActionSheet">Open ActionSheet</button>
2.  <view class="actionsheet-container">
3.    <!-- 监听自定义事件 cancel 和 actionclick, 绑定回调函数 -->
4.    <zan-actionsheet
5.      show="{{ show }}"
6.      actions="{{ actions }}"
7.      cancel-text="{{ cancelText }}"
8.      cancel-with-mask="{{ cancelWithMask }}"
9.      bind:cancel="closeActionSheet"
10.     bind:actionclick="handleActionClick"
11.    >
12.  </zan-actionsheet>
13. </view>
```

```
1.  // 在 Page 中混入 Actionsheet 里面声明的方法
2.  Page({
3.    data: {
4.      show: false,
5.      cancelWithMask: true,
6.      actions: [{
7.        name: '选项1',
8.        subname: '选项描述语1',
9.        loading: false
```



```

10.     }, {
11.       name: '选项2',
12.       subname: '选项描述语2',
13.       loading: false
14.     }, {
15.       name: '去分享',
16.       openType: 'share'
17.     }],
18.     cancelText: '关闭 Action'
19.   },
20.   openActionSheet() {
21.     this.setData({
22.       'show': true
23.     });
24.   },
25.   closeActionSheet() {
26.     this.setData({
27.       'show': false
28.     });
29.   },
30.   handleActionClick({ detail }) {
31.     // 获取被点击的按钮 index
32.     const { index } = detail;
33.   }
34. });

```

Actionsheet 支持的具体参数如下（ 传入时使用分隔线写法 ）

参数	说明	类型	默认值	必须
show	用来表示是否展示行动按钮	Boolean	false	
actions	指定弹层里的按钮	Array	[]	
cancelText	行动按钮底部取消按钮的文案，不传则不显示取消按钮	String		
cancelWithMask	是否在点击背景时，关闭行动按钮	Boolean	false	
mask-class	用于控制蒙层样式的外部类	String		
container-class	用于控制容器样式的外部类	String		

actions 的具体数据结构

```

1.  // actions 为数组结构传入
2.  [{
3.    // 按钮文案

```

```

4.   name: '选项1',
5.   // 按钮描述文案，不传就不显示
6.   subname: '选项描述语1',
7.   // 按钮是否显示为 loading
8.   loading: false,
9.   // 按钮的微信开放能力
10.  // 具体支持可参考微信官方文档：
    https://mp.weixin.qq.com/debug/wxadoc/dev/component/button.html
11.  openType: 'share'
12.  }]

```



微信扫一扫

原文：

<https://www.youzanyun.com/zanui/weapp#/zanui/interactive/actionsheet>

# DatetimePicker 时间选择

## DatetimePicker 时间选择

使用 picker 组件开发的时间日期组件，弥补小程序 picker 自身对于快速时间选择的不支持

### 示例代码

```
1. {
2.   "usingComponents": {
3.     "zan-date-picker": "../dist/datetime-picker/index"
4.   }
5. }
```

```
1. <zan-date-picker
2.   bindchange="change"
3.   bindcancel="cancel"
4.   placeholder="请选择一个时间"
5.   placeholder-class="my-customer-class-name"
6.   format="你选择了YYYY年MM月DD日HH点mm分ss秒"
7. />
```

### 属性与事件

名称	类型	是否必须	默认	描述
value	null	<div>否</div>	当前时间	初始化时间，传入的值会被 Date 构造函数转换为一个 Date 对象，不合法的值将抛出一个错误
placeholder-class	<div>String</div>	<div>否</div>	无	自定义类，可改变 placeholder 样式，其他类无效，picker-view 为 true 时不支持
placeholder	<div>String</div>	<div>否</div>	请选择时间	设置 picker 的 placeholder
not-use	<div>Array</div>	<div>否</div>	无	不需要显示的列，可选择 years, months, days, hours, minutes, seconds 中的多个
native	<div>Boolean</div>	<div>否</div>	无	使用原生 picker，还是自定义的 picker（自定义 picker 滚动不如原生）
picker-view	<div>Boolean</div>	<div>否</div>	无	如果为 true，相当于 picker-view 组件
format	<div>String</div>	<div>否</div>	YYYY-MM-DD HH:mm:ss	设置选中的时间显示的格式，支持 YYYY, yyyy, YY, yy, MM, M, DD, dd, D, d, HH, hh, H, h, mm, m, ss, s

bindchange	String	是	无	用户点击 <b>确认</b> 触发该事件，返回值为按“年，月，日，时，分，秒”顺序的数组，可以通过 <b>detail.value</b> 获取
bindcancel	String	否	无	用户点击 <b>取消</b> 触发该事件

方法

名称	参数	描述		
getFormatStr	无	返回	<b>format</b>	格式的字符串，在 <b>picker-view</b> 为 true 时比较实用

增强优化

- 支持可选择时间区域限制
- 滚动优化



微信扫一扫

原文：

[https://www.youzanyun.com/zanui/weapp#/zanui/interactive/datetime\\_picker](https://www.youzanyun.com/zanui/weapp#/zanui/interactive/datetime_picker)

# Dialog 弹出框

## Dialog 弹出框

### 使用指南

在 index.json 中引入组件

```
1.  {
2.    "usingComponents": {
3.      "zan-dialog": "path/to/zanui-weapp/dist/dialog/index"
4.    }
5.  }
```

在需要使用的页面里引入组件库模板和脚本

```
1. <zan-dialog id="zan-dialog-test"></zan-dialog>
```

```
1. const Dialog = require('path/to/zanui-weapp/dist/dialog/dialog');
2.
3. Page({
4.   // ...
5.   // 可以在任意方法里直接调用，即可唤起
6.   handleClick() {
7.     Dialog({
8.       title: '',
9.       message: '',
10.      selector: '#zan-dialog-test'
11.    }).then((res) => {
12.      console.log(res);
13.    })
14.  }
15. });
```

### 代码演示

#### 按钮展示方式

按钮可以通过设置 `buttonsShowVertical` 来切换按钮纵向展示或者横向并排展示，方便各种场景

下使用。

```
1. Dialog({
2.   message: '这是一个模态弹窗',
3.   buttonsShowVertical: true,
4.   showCancelButton: true
5. });
```

### 自定义展示按钮

`dialog` 支持自定义展示按钮。设置 `buttons` 数组即可实现。自定义按钮的点击后，都会在 `resolve` 状态中监听到。

```
1. Dialog({
2.   message: '这是一个模态弹窗',
3.   buttons: [{
4.     // 按钮文案
5.     text: '现金支付',
6.     // 按钮文字颜色
7.     color: 'red',
8.     // 按钮类型，用于在 then 中接受点击事件时，判断是哪一个按钮被点击
9.     type: 'cash'
10.  }, {
11.    text: '微信支付',
12.    color: '#3CC51F',
13.    type: 'wechat'
14.  }, {
15.    text: '取消',
16.    type: 'cancel'
17.  }]
18. }).then(({ type }) => {
19.   // type 可以用于判断具体是哪一个按钮被点击
20.   console.log('=== dialog with custom buttons ===', `type: - {type}`);
21. });
```

### 具体参数

参数	说明	类型	默认值	必须
message	弹窗内容	String	-	必须
selector	显示弹窗对应组件节点的选择器	String	-	必

selector	显示弹窗对应组件节点的选择器	String	-	须
title	弹窗标题	String	-	
buttonsShowVertical	按钮是否纵向展示	Boolean	false	
showConfirmButton	是否展示确认按钮	Boolean	true	
confirmButtonText	确认按钮文案	String	确定	
confirmButtonColor	确认按钮文字颜色	String	#3CC51F	
showCancelButton	是否展示取消按钮	Boolean	false	
cancelButtonText	取消按钮文案	String	取消	
cancelButtonColor	取消按钮文字颜色	String	#333	
buttons	自定义按钮列表，设置以后，以上关于 确认 和 取消 按钮的设置全部不生效。	Array	-	

## buttons 数据格式

```

1.  [{
2.    // 按钮文案
3.    text: '现金支付',
4.    // 按钮文字颜色
5.    color: 'red',
6.    // 按钮类型，用于在 then 中接受点击事件时，判断是哪一个按钮被点击
7.    type: 'cash'
8.  }, {
9.    // 按钮文案
10.   text: '微信支付',
11.   // 按钮文字颜色
12.   color: '#3CC51F',
13.   // 按钮类型，用于在 then 中接受点击事件时，判断是哪一个按钮被点击
14.   type: 'wechat'
15. }, {
16.   // 按钮文案
17.   text: '取消',
18.   // 按钮类型，用于在 then 中接受点击事件时，判断是哪一个按钮被点击
19.   type: 'cancel'
20. }]

```



微信扫一扫

原文：

<https://www.youzanyun.com/zanui/weapp#/zanui/interactive/dialog>



# Popup 弹出层

## Popup 弹出层

### 使用指南

在 `index.json` 中引入组件

```
1.  {
2.    "usingComponents": {
3.      "zan-popup": "path/to/zanui-weapp/dist/popup/index"
4.    }
5.  }
```

### 代码演示

可以在页面任意位置上使用 `zan-popup` 标签。通过 `show` 可以控制弹窗是否展示

```
1. <zan-popup show="{{ show }}"></zan-popup>
```

### popup 动画

popup 额外支持了 上下左右 四种动画方式，通过增加 `type` 属性即可控制。使用方式如下

```
1. <!-- 从底部弹出的弹层 -->
2. <zan-popup type="bottom" show="{{ show }}"></zan-popup>
3.
4. <!-- 从顶部弹出的弹层 -->
5. <zan-popup type="top" show="{{ show }}"></zan-popup>
6.
7. <!-- 从左侧弹出的弹层 -->
8. <zan-popup type="left" show="{{ show }}"></zan-popup>
9.
10. <!-- 从右侧弹出的弹层 -->
11. <zan-popup type="right" show="{{ show }}"></zan-popup>
```

### 具体参数和事件

#### 参数说明

参数	说明	类型	默认值	必须
show	是否显示弹出层	Boolean	false	
overlay	是否显示遮罩层	Boolean	false	
closeOnClickOverlay	遮罩层点击时，是否触发关闭事件	Boolean	true	
type	弹出层动画方式，可选 <code>center</code> ， <code>left</code> ， <code>right</code> ， <code>top</code> ， <code>bottom</code>	String	center	

事件说明

事件名	说明	参数
click-overlay	遮罩层点击触发	
close	遮罩层关闭时触发	



微信扫一扫

原文：

<https://www.youzanyun.com/zanui/weapp#/zanui/interactive/popup>

# Tab 标签

## Tab 标签

### 使用指南

在 `index.json` 中引入组件

```
1.  {
2.    "usingComponents": {
3.      "zan-tab": "path/to/zanui-weapp/dist/tab/index"
4.    }
5.  }
```

在 `index.js` 中声明组件数据

```
1.  // 在 Page 中声明 Tab 依赖的展示数据
2.  Page({
3.    data: {
4.      list: [{
5.        id: 'xxx',
6.        title: 'xxx'
7.      }],
8.      selectedId: 'xxx',
9.      ...
10.   }
11. })
```

### 代码演示

可以在任意位置上使用 `zan-tab` 标签。传入对应的数据即可。

```
1.  <zan-tab
2.    scroll="{{ scroll }}"
3.    list="{{ list }}"
4.    selected-id="{{ selectedId }}"
5.    height="{{ height }}"
6.    fixed="{{ fixed }}"
7.    bindtabchange="handleTabChange"
```

```
8. />
```

参数	说明	类型	默认值	必须
scroll	是否开启 tab 左右滑动模式	Boolean	-	
list	可选项列表	Array	-	
selectedId	选中id	-	-	
height	tab高度	Number	-	
fixed	是否固定位置	Boolean	-	

tab 组件中, list 数据格式如下

```
1. [{
2.   // tab 项 id
3.   id: 'all',
4.   // tab 项展示文案
5.   title: '全部'
6. }, {
7.   id: 'topay',
8.   title: '待付款'
9. }, {
10.  id: 'tosend',
11.  title: '待发货'
12. }, {
13.  id: 'send',
14.  title: '待收货'
15. }, {
16.  id: 'sign',
17.  title: '已完成'
18. }]
```

可以监听 bindtabchange 事件回调, 在页面注册回调函数

```
1. Page({
2.   customCallback(selectedId) {
3.     // selectId 表示被选中 tab 项的 id
4.   }
5. });
```



微信扫一扫

原文：

<https://www.youzanyun.com/zanui/weapp#/zanui/interactive/tab>

# Toast 轻提示

## Toast 轻提示

### 使用指南

在 json 文件中配置 toast 组件

```
1. "usingComponents": {  
2.   "zan-toast": "/path/to/zanui-weapp/dist/toast/index"  
3. }
```

在需要使用的页面里引入组件库模板和脚本

```
1. <zan-toast id="zan-toast-test"></zan-toast>
```

```
1. const Toast = require('path/to/zanui-weapp/dist/toast/toast');  
2.  
3. Page({  
4.   // ...  
5.   // 可以在任意方法里直接调用，即可唤起  
6.   handleClick() {  
7.     Toast({  
8.       message: 'toast me',  
9.       selector: '#zan-toast-test'  
10.    });  
11.  }  
12. });
```

### 加载提示

```
1. Toast.loading({  
2.   selector: '#zan-toast-test'  
3. });
```

### 参数说明

### 方法

方法名	参数	返回值	介绍
Toast	<code>options</code> , <code>timeout</code>	-	展示提示
Toast.loading	<code>options</code>	-	展示加载提示
Toast.clear	-	-	关闭提示
Toast.setDefaultOptions	<code>options</code> 格式同 Toast 函数可以传入的参数, <code>type</code> 可选 global/page, 分别指定对整个小程序生效/对当前页面生效	-	修改默认配置, 对所有 Toast 生效
Toast.resetDefaultOptions	<code>type</code> 可选 global/page	-	重置默认配置, 对所有 Toast 生效

options 具体参数如下

参数	说明	类型	默认值	可选值
message	toast 显示文案	String	-	
type	提示类型	String	-	loading success fail
icon	toast 显示图标, 可以用 icon 里面支持的所有图标	String	-	-
image	toast 显示图标, 为图片的链接, 传入此值后会覆盖 icon 值	String	-	
timeout	toast 显示时间, 小于0则会一直显示, 需要手动调用 clearZanToast 清除	Number	-	



微信扫一扫

原文：

<https://www.youzanyun.com/zanui/weapp#/zanui/interactive/toast>



# TopTips 顶部提示

## TopTips 顶部提示

### 使用指南

在 index.json 中引入组件

```
1.  {
2.    "usingComponents": {
3.      "zan-toptips": "path/to/zanui-weapp/dist/toptips/index"
4.    }
5.  }
```

在 index.js 中声明组件数据

**toptips**提供了声明式和命令式2种调用方式，但是由于小程序本身限制，会有一定使用的要求

```
1.  // 使用声明式调用的方式， 必须在Page中声明 - zanui对象， 结构如下
2.  // 同时在其他触发toptips显示的函数中，需要手动改变对应的数值
3.  Page({
4.    data: {
5.      duration: 1000,
6.      content: 'xxx',
7.      - zanui: {
8.        toptips: {
9.          show: false
10.        }
11.      }
12.    }
13.  })
14.
15.  // 使用命令式调用的方式， 必须在 wxml 模板中声明组件id，
16.  // 默认我们使用了 zan-toptips， 如果使用者要更换，可以手动传入
17.
18.  const Toptips = require('path/to/zanui-weapp/dist/toptips/index');
19.  Page({
20.    customCallback() {
21.      Toptips('只穿文案展示');
22.    }
23.  })
```

```
23. })  
24.
```

## 代码演示

### 声明式调用

#### 使用声明式调用

```
1.  Page({  
2.    data: {  
3.      duration: 1000,  
4.      content: 'xxx',  
5.      - zanui: {  
6.        toptips: {  
7.          show: false  
8.        }  
9.      }  
10.   },  
11.  
12.   customCallback() {  
13.     this.setData({  
14.       - zanui: {  
15.         toptips: {  
16.           show: true  
17.         }  
18.       }  
19.     });  
20.  
21.     setTimeout(() => {  
22.       this.setData({  
23.         - zanui: {  
24.           toptips: {  
25.             show: false  
26.           }  
27.         }  
28.       })  
29.     }, this.data.duration);  
30.   }  
31. })
```

```
1. <zan-toptips
2.   content="tip内容"
3.   duration="{{ duration }}"
4.   is-show="{{ - zanui.toptips.show }}"
5. />
```

## 命令式调用

```
1. Page({
2.   customCallback() {
3.     Toptips('我只改文案')
4.   }
5. })
```

```
1. <zan-toptips
2.   id="zan-toptips"
3.   contetn="{{ content }}"
4. />
```

## 修改组件id

```
1. Page({
2.   customCallback() {
3.     Toptips({
4.       content: '传入其他参数',
5.       selector: '#other-id',
6.       duration: 5000
7.     })
8.   }
9. })
```

```
1. <zan-toptips
2.   id="other-id"
3.   contetn="{{ content }}"
4. />
```

参数	说明	类型	默认值	可选值
content	展示文案	String	-	
duration	弹层持续时间	Number	3000	

isShow	弹层是否展示	Boolean	false
color	字体颜色	String	#fff
backgroundColor	提示背景色	String	#e64340



微信扫一扫

原文：

<https://www.youzanyun.com/zanui/weapp#/zanui/interactive/toptips>

# Loadmore 加载

## Loadmore 加载

### 使用指南

在 app.wxss 中引入组件库所有样式

```
1. @import "path/to/zanui-weapp/dist/index.wxss";
```

在需要使用的页面里引入组件库模板

```
1. <import src="path/to/zanui-weapp/dist/loadmore/index.wxml" />
2.
3. <!-- 直接使用 zan-loadmore 模板，并且直接传入设置值 -->
4. <template is="zan-loadmore" data="{{ loading: true }}"></template>
```

### 代码演示

**loadmore** 支持三种状态，loading，nodata，nomore。传入指定的值即可显示

```
1. <!-- 加载更多 -->
2. <template is="zan-loadmore" data="{{ loading: true }}"></template>
3. <!-- 没有可以显示的数据 -->
4. <template is="zan-loadmore" data="{{ nodata: true }}"></template>
5. <!-- 没有更多的数据了 -->
6. <template is="zan-loadmore" data="{{ nomore: true }}"></template>
```



微信扫一扫

原文：

<https://www.youzanyun.com/zanui/weapp#/zanui/interactive/loadmore>