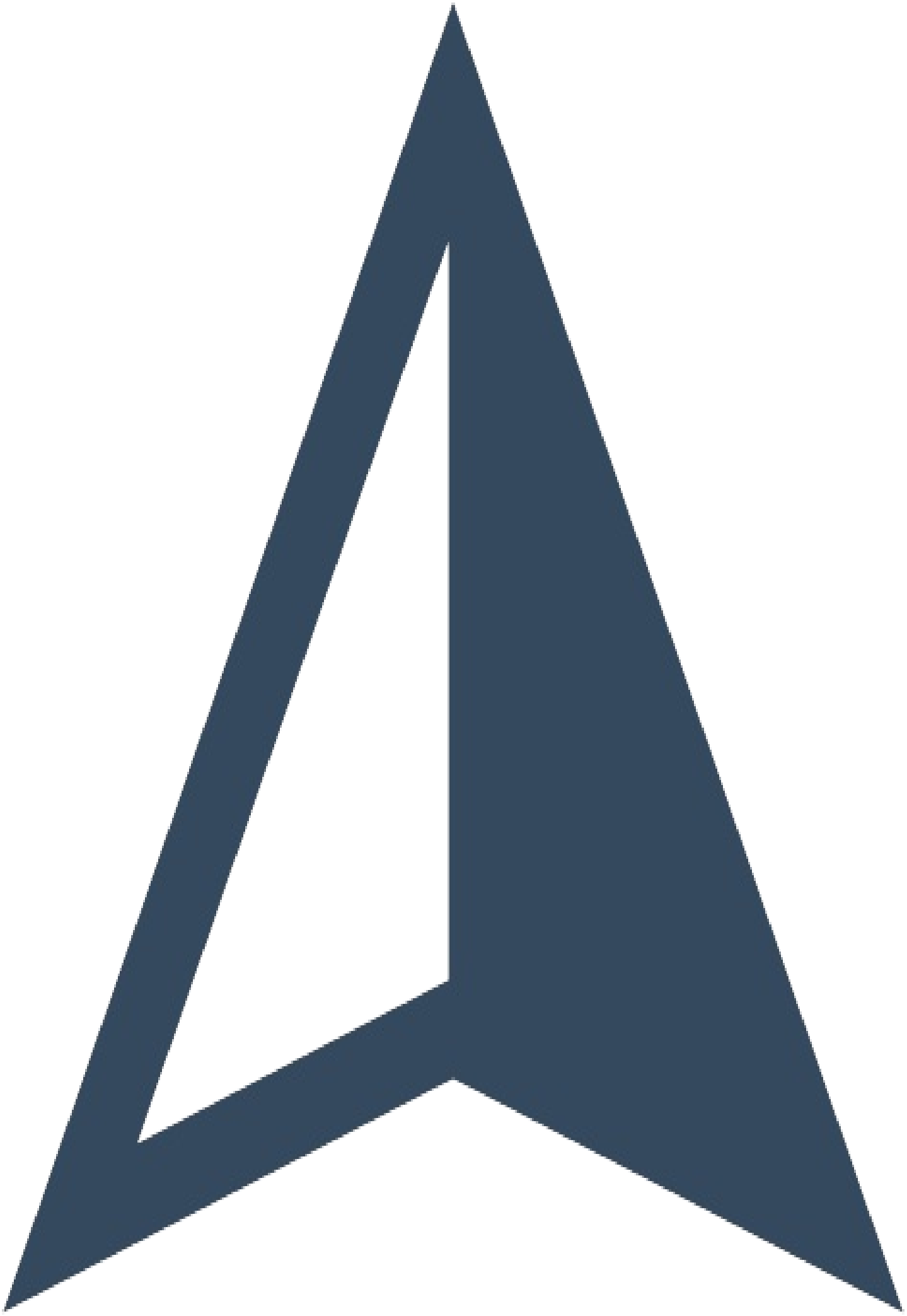


Table of Contents

关于Vux 1.1

- 开始使用之前 1.1.1
- 常见问题 1.1.2
- 兼容 1.1.3
- 组件规范 1.1.4
- PR规范 1.1.5
- playground 1.1.6
- 版本变更 1.1.7
- 谁在使用 1.1.8
- 最佳实践 1.1.9

安装 1.2

- vue-cli项目引用.vue组件(推荐) 1.2.1
- vue-cli项目引用umd组件 1.2.2
- 使用script全局引入 1.2.3

布局组件 1.3

- Flexbox 1.3.1
- Scroller 1.3.2
- Tab 选项卡 1.3.3
- Popup 弹出层 1.3.4
- Sticky 自动固定在顶部 1.3.5
- Tabbar 底部导航 1.3.6
- ButtonTab 1.3.7
- Timeline 1.3.8
- XHeader 1.3.9
- Step 1.3.10

UI组件 1.4

- divider 横向分隔线 1.4.1

badge 1.4.2
icon 1.4.3
Search 搜索 1.4.4
常用样式 1.5
1px解决方案 1.5.1
reddot红点提示 1.5.2
表单组件 1.6
Group表单分组组件 1.6.1
Cell 1.6.2
Address 1.6.3
x-button 按钮 1.6.4
x-input 单行文本输入 1.6.5
x-textarea 多行输入框 1.6.6
x-number 数字输入 1.6.7
checklist 多选 1.6.8
selector 下拉选择 1.6.9
switch 开关 1.6.10
rater 评分 1.6.11
Inline-Calendar 内联日历 1.6.12
Calendar 日历 1.6.13
Radio 单选 1.6.14
Checker 灵活的选择组件 1.6.15
Datetime 时间 1.6.16
Range 1.6.17
picker 1.6.18
popup-picker 1.6.19
表单验证(v0.1.1) 1.6.20
时间组件 1.7
日期倒计时 1.7.1
秒数倒计时 1.7.2

轻量时间格式化 1.7.3
相对时间 1.7.4
数字组件 1.8
number-roller 数字滚动 1.8.1
提示组件 1.9
Loading 1.9.1
Alert 1.9.2
Confirm 1.9.3
Toast 1.9.4
Actionsheet 1.9.5
Dialog 1.9.6
Spinner 1.9.7
图片 1.10
背景模糊 1.10.1
半透明遮罩 1.10.2
轮播 1.10.3
表情图片 1.10.4
上传(未支持) 1.10.5
点击全屏预览图片(未支持) 1.10.6
动态检测 1.11
屏幕翻转 1.11.1
手机摇一摇 1.11.2
图表组件 1.12
Circle 圆圈 1.12.1
http请求 1.13
router 路由 1.14
手势操作 1.15

Be Cool with Vue and WeUI.

Vux = Vue + WeUI + A Bunch of Components

Vux

Vux 基于 `Vue` 和 `WeUI` 的组件库。

Vux的目标是：

- 尽量满足手机端页面大部分组件需求。
- 尽量保持小体积(甚至不依赖 `Zepto`)。
- 尽量简洁优雅。

Demo

https://vux.li

文档修订

```
# clone the doc repo
git clone https://github.com/vuxjs/vux-doc.git

# 安装gitbook-cli
npm install gitbook-cli -g

# cd到文档目录
gitbook install
gitbook serve
```

开始使用vux之前

我们假设你应该已经写过 `vue` 页面或者看过文档，否则在使用过程中会遇到很多疑惑。这种疑惑的锅vux不背 :doge:。

如果以下概念有不清楚的地方，需要再次查看文档，聪明如你，肯定花不了多少时间：

- `slot` 内容插槽
- `:prop` `:prop.sync` `:prop.once`
- `@click` `events` `$emit`

文档链接

- vue官方文档
- webpack
- vue-loader
- es6
- vue-router

相关开源项目

除了Vue 及 WeUI，组件部分代码及设计来自于以下开源项目或网站：

- FrozenUI
- Ant Design
- Ant UI
- XScroll
- Ionic
- SUI Mobile
- Spmjs.io (已下线)

常见问题

其他格式的文档有么?

- PDF
- ePub
- Mobi

不懂Vue，怎么用

去学

Vux是微信官方项目吗

不是，但是Vux依赖于微信官方开发的 `WeUI`

Vux和Vuex名字太像了

真不是故意的

可以在template中使用其他标签吗

当然可以，局部注册组件的时候指定名字。

```
{
  components: {
    xalert: Alert
  }
}
```

如果使用全局注册，则

```
Vue.component('xalert', vuxAlert)
```

为什么部分组件要加x-前缀

若不更名，可能在开发时与标准html标签相同而导致冲突或者bug。

有QQ交流群吗

没有。 但是你可以付费加入Bearychat。 Bearychat vux@bearychat

为什么要付费：

> 目前处理起来很费时间
>
> 设定门槛
>
> 时间有限，无法用同样的标准来对待每一位用户的每一个问题，并且不同的用户对技术支持的需求和紧急程度不同。

- 大部分人加入后没有任何问题建议，这不是我希望看到的，对项目开发无益，而我还需要去审核这些无用的申请
- 一部分同学不熟悉Vue提的跟Vux没任何关系，回答无非帖个文档链接
- 一部分同学不会正确提问题，过滤这些问题浪费时间，即使我有空，一步一步沟通也浪费时间
- 一部分同学问的是Web相关的其他问题
- 可能有惊喜

不付费就提不了问题吗

不是，你依然可以到Github上提交Vux相关issue。付费只是提供了有门槛但更为快速的沟通方式，同时你也可以咨询其他问题。 但是Bearychat不是即时沟通，不能保证随问随回。

假如公司项目已经使用Vux，那么可以理解为支持开源项目以及获取更加快速的技术支持。

如何付费

扫下面支付码支付 ￥**188**，然后把支付单号写在加入申请信息中。

满足任一条目的请不要加入

- 觉得特别贵
- 还没写过Vue
- 觉得是在坑钱
- 要求秒回
- 纯粹观望
- 先加入再退款

我想给你发个红包，怎么发

Alipay

Wechat

兼容

- Vux兼容 `Android 4.0+` 、微信webview及iOS 主流版本的 `Safari` 。
- Vux没有对Android上的火狐做兼容

> 2016-04-19 微信官方发布消息，Android `x5` 内核 已升级到 `Blink` ，可以愉快地写 `flex` 了。

开发规范

0. 代码规范

主要原则

- 使用最新es6语法
- 通过项目的eslint

具体细则

- 用 `const` 或者 `let` 代替 `var`
- 依赖包括工具函数分离成单独的js文件 `import`

1. 组件

组件命名

- 尽量简单、标准。如果与原生标签一样，在前面加上 `x` ，如 `x-input` , `x-textarea` , `x-img`
- export 出来的对象命名为 `驼峰式` ，如 `Radio` , `XInput` , `XTextarea`
- 假设存子组件，子组件命名在父组件后面加上 `-item` ，
  如 `flexbox` 及 `flexbox-item` , `tab` 及 `tab-item`

组件目录

- 每个组件为单独的目录，位于 `src/components/` 下，目录名全小写，入口文件为 `index.vue`
- 若项目包含子组件，入口文件同样为 `index.vue` ，同时目录下包含与组件名同名的文件，如 `tab.vue` 、 `tab-item.vue`

组件属性

- 必须规定 `type` 或者 `validator` 进行类型验证
- 不要加 `coerce`

2. 事件

命名

- 统一以on前缀，如on-change。这样规范的原因：
  - 避免与原生事件(如 `change`) 同名冲突重复收到DOM事件且参数错误
  - 容易读写，与其他类型属性区分

3. 模板

- 不能为 `片断` 模板(不会暴露至外部使用的公用小组件除外)
- `class` 和 `style` 超过两个属性要写到 `computed` 里

4. 版本发布

- 修改 `package.json` 的version
- 添加 `git tag`
- `npm publish`
- `git push origin master`
- `git push origin tag`
- 登录 `Github` 修改最新release的发布信息

PR 规范

- 提交前确认已经 `rebase` 了开发分支代码
- 只修改组件源代码，不需要进行 `xbuild` 操作
- 遵从代码缩进规范，确认 `npm run dev` 时没有 `eslint` 错误
- 如果添加了新的 `feature` 或者修复了某个bug, 需要在相应的demo文件增加example

Commit Message

提交消息如果是修改了特定组件，请使用组件名+冒号+英文空格+英文提交信息的形式

栗子: `Cell: Add some prop`

特定组件依赖

带有子组件并且有selected状态的需要使用multi-mixin, 工具库位于 `src/mixins/multi-items.js` ，可以参考 `button-tab` 、 `tabbar` 等组件。

playground

页面接口

https://vux.li/api/v1/demo.html

> 简易接口，用于把代码拼凑成完整的vue页面，文档中的例子都是使用该接口以iframe嵌入。
>
> 注意 `GET` 方式url长度有限制，所以这种方式用于相对简单的例子。

url参数

- `components` 使用的组件，英文逗号分隔，如 `Group,Radio`
- `javascript` js代码，以 `export default {` 开头， `}` 结束
- `template` 模板，普通 `html` 代码
- `style` 样式,只支持一般css，不支持指定 `lang` 及 `scoped`

在文档中的例子：

写法同 `vue` 文件，但是只支持普通css写法，不支持less等。不支持 `import` 外部组件。

> 通过components来注册使用到的组件， `必需`
>
> 宽度width默认为100%
>
> 高度height默认为 100

```
``` vux width=100% height=100 components=Circle
<template>
  <div style='width:100px;height:100px;'>
    <circle :percent='percent2' :stroke-width=6 :trail-width=6 :
stroke-color='strokeColor2' trail-color="#ececec">
      <span :style="{color: strokeColor2}">{{percent2}}%</span>
    </circle>
```
```

```
  </div>
</template>

<script>
export default {
  ready () {
    setInterval(this.update, 2000)
  },
  data () {
    return {
      percent2: 30,
      strokeColor2: '#3FC7FA'
    }
  },
  methods: {
    update: function () {
      const colorMap = ['#3FC7FA', '#85D262', '#FE8C6A']
      this.percent2 = parseInt(Math.random() * 100, 10)
      this.strokeColor2 = colorMap[parseInt(Math.random() * 3, 10
)]
    }
  }
}
</script>

<style>
.hello {


}
</style>
```

渲染结果

{{percent2}}%

template

```
<div style='width:100px;height:100px;'>
  <circle :percent='percent2' :stroke-width=6 :trail-width=6 :stro
ke-color='strokeColor2' trail-color="#ececec">
    <span :style="{color: strokeColor2}">%</span>
  </circle>
</div>
```

script

```
export default {
  ready () {
    setInterval(this.update, 2000)
  },
  data () {
    return {
      percent2: 30,
      strokeColor2: '#3FC7FA'
    }
  },
  methods: {
    update: function () {
      const colorMap = ['#3FC7FA', '#85D262', '#FE8C6A']
      this.percent2 = parseInt(Math.random() * 100, 10)
      this.strokeColor2 = colorMap[parseInt(Math.random() * 3, 10)
]
    }
  }
}
```

style

```
.hello {
}
```

vue项目

> `vue-cli` 的使用请查看[文档][https://github.com/vuejs/vue-cli]
>
> `vue-loader` 文档

```
# 安装 vue-cli
npm install -g vue-cli

# 初始化 webpack 项目
vue init webpack my-project
cd my-project
# npm可能出现访问速度极慢的情况，推荐使用cnpm
npm install
#安装 vux 发版请使用 npm install vux@next
npm install vux
#安装less-loader, vuejs-templates模板默认不安装less less-loader
npm install less less-loader --save-dev
# 调试
npm run dev
```

在webpack.base.conf.js添加loader

```
{
  test: /vux.src.*?js$/,
  loader: 'babel'
}
```

```
<template>
  <div>
    <group>
      <cell title="vue" value="cool"></cell>
    </group>
  </div>
</template>

<script>
import Group from 'vux/src/components/group'
import Cell from 'vux/src/components/cell'

export default {
  components: {
    Group,
    Cell
  }
}
</script>

<style>
@import '~vux/dist/vux.css';
</style>
```

> 可以配置alias使用更简短的引用路径

```
resolve: {
  alias: {
    'vux-components': 'vux/src/components/'
  }
}
```

> 那么就可以这样引用

```
import Group from 'vux-components/group'
import Cell from 'vux-components/cell'
```

vue项目

> `vue-cli` 的使用请查看[文档][https://github.com/vuejs/vue-cli]
>
> `vue-loader` 文档

```
# 安装 vue-cli
npm install -g vue-cli

# 初始化 webpack 项目
vue init webpack my-project
cd my-project
# npm可能出现访问速度极慢的情况，推荐使用cnpm
npm install
#安装 vux
npm install vux
# 开发版安装请使用 npm install vux@next
# 调试
npm run dev
```

```
<template>
  <div>
    <group>
      <cell title="vue" value="cool"></cell>
    </group>
  </div>
</template>

<script>
// 不推荐的方式，会打包所有vux模块
import { Group, Cell } from 'vux'

// 推荐的方式，按需加载需要的组件
import Group from 'vux/dist/components/group'
import Cell from 'vux/dist/components/cell'

export default {
  components: {
    Group,
    Cell
  }
}
</script>

<style>
@import '~vux/dist/vux.css';
</style>
```

使用script引入

不推荐script的方式引入，在组件多的时候会存在比较多冗余的代码导致体积比较大。

但是非新项目确实存在非.vue组件的调用形式。vux提供了所有组件的压缩包以及各个组件分别打包的文件。

下载

- 从Github Release下载
- `bower install vux`

使用

```
<!--include Vux style-->
<link rel="stylesheet" href="vux/dist/vux.css">
<!--include Vue yourself-->
<script src="vue.js"></script>

<div id="demo">
  <group>
    <cell title="vue" value="cool"></cell>
  </group>
</div>

<!--include the components you need-->
<script src="vux/dist/components/group/index.js"></script>
<script src="vux/dist/components/cell/index.js"></script>

<script>
// register components
Vue.component('group', vuxGroup)
Vue.component('cell', vuxCell)

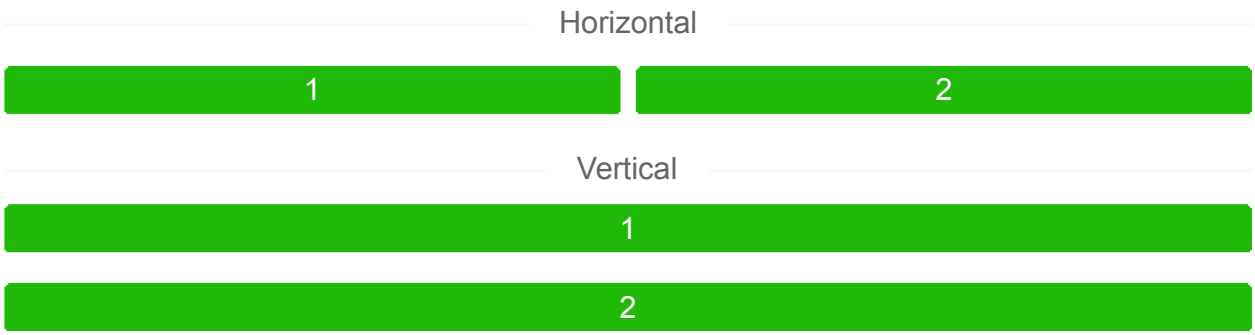
new Vue({
  el: '#demo'
})
</script>
```

Flexbox

> Flexbox功能由 `Flexbox` 及 `FlexboxItem` 子组件组成。

简单平分

新窗口打开



template

```
<divider>Horizontal</divider>
<flexbox style="height:40px;">
  <flexbox-item><div class="flex-demo">1</div></flexbox-item>
  <flexbox-item><div class="flex-demo">2</div></flexbox-item>
</flexbox>
<divider>Vertical</divider>
<flexbox orient="vertical" :margin-left=0>
  <flexbox-item><div class="flex-demo" style="margin-left:0">1</div
></flexbox-item>
  <flexbox-item><div class="flex-demo" style="margin-left:0">2</div
></flexbox-item>
</flexbox>
```

style

```
.flex-demo {
  text-align: center;
  color: #fff;
  background-color: #20b907;
  margin-bottom: 8px;
  border-radius: 4px;
  -webkit-background-clip: padding-box;
}
```

嵌套布局

> 国内电商网站和支付应用最常见
>
> 1像素边框实现请参考1px解决方案

新窗口打开

template

```
<flexbox :margin-left=0 style="height: 200px; background-color: #fff;" class="vux-1px-tb vux-1px-l vux-1px-r">
  <flexbox-item class="vux-1px-r" style="height:200px;"></flexbox-item>
  <flexbox-item>
    <flexbox orient="vertical" :margin-left=0>
      <flexbox-item class="vux-1px-b"></flexbox-item>
      <flexbox-item style="height: 100px;"><!--height: 100% doesnot work here-->
        <flexbox :margin-left=0>
          <flexbox-item class="vux-1px-r"></flexbox-item>
          <flexbox-item></flexbox-item>
        </flexbox>
      </flexbox-item>
    </flexbox>
  </flexbox-item>
</flexbox>
```

Scroller

> Scroller 依赖于 `xscroll`

Props

名字	类型	默认	描述
height	String	Viewport 高度	高度
lockX	Boolean	false	锁定X方向
lockY	Boolean	false	锁定Y方向
scrollbarX	Boolean	false	横向滚动条
scrollbarY	Boolean	false	垂直方向滚动条
bounce	Boolean	true	是否有边缘回弹
use-pulldown	Boolean	false	是否使用下拉组件
use-pullup	Boolean	false	是否使用上拉组件
pulldown-config	Object	见下面	下拉组件配置
pullup-config	Object	见下面	上拉组件配置
pulldown-status	String	无	双向绑定，当需要自定义下拉刷新行为时配置
pullup-status	String	无	双向绑定，当需要自定义上拉行为时配置

pulldown-config默认：

```
{
  content: 'Pull Down To Refresh',
  height: 60,
  autoRefresh: false,
  downContent: 'Pull Down To Refresh',
  upContent: 'Release To Refresh',
  loadingContent: 'Loading...',
  clsPrefix: 'xs-plugin-pulldown-'
}
```

pullup-config默认

```
{
  content: 'Pull Up To Refresh',
  pullUpHeight: 60,
  height: 40,
  autoRefresh: false,
  downContent: 'Release To Refresh',
  upContent: 'Pull Up To Refresh',
  loadingContent: 'Loading...',
  clsPrefix: 'xs-plugin-pullup-'
}
```

Methods

名字	参数	描述
reset	无	重新渲染，因为scroller并不知道内部内容是否变化，因此需要手动取得 `ref` 进行reset, 并且需要在 `$nextTick` 中执行。

示例：

```
this.$nextTick(() => {
  this.$refs.scroller.reset()
})
```

Events

名字	参数	描述
pullup:loading	(scroller的uuid)	上拉加载时触发的事件，需要在获取数据后使用 `$broadcast` 触发状态更新，`this.$broadcast('pullup:reset', uuid)`
pulldown:loading	(scroller的uuid)	下拉加载时触发的事件，需要在获取数据后使用 `$broadcast` 触发状态更新，`this.$broadcast('pulldown:reset', uuid)`
pullup:disable	(scroller的uuid)	禁用上拉加载，当没有更多数据需要禁用时使用 `$broadcast` 触发禁用， `this.$broadcast('pullup:disable', uuid)`
pullup:enable	(scroller的uuid)	启用上拉加载，禁用插件后，当又重新需要时使用 `$broadcast` 触发重新启用， `this.$broadcast('pullup:enable', uuid)`

Slots

> 注意，默认slot根元素必须有且只有一个, 如 `<scroller><div>content</div></scroller>`

名字	描述
默认slot	无

Demos

> 更多的demo请手机访问 `https://vux.li` 进行查看。

- scroller
- pullup
- pulldown
- pulldown & pullup

```
<scroller lock-y :scrollbar-x="false">
  <div class="box1">
    many many content
  </div>
</scroller>
```

Tab 选项卡

Props

tab

参数	说明	类型	默认值
line-width	可选，边框大小	Number	3
active-color	可选，高亮文字的颜色和线条颜色	String	#04be02
default-color	可选，默认文字的颜色	String	#666
animate	可选，是否使用动画	Boolean	true

tab-item

参数	说明	类型	默认值
selected	是否高亮	Boolean	false

Demo

新窗口打开

已发货 未发货 全部订单

template

```
<tab>
  <tab-item :selected="demo1 === '已发货'" @click="demo1 = '已发货'">已发货</tab-item>
  <tab-item :selected="demo1 === '未发货'" @click="demo1 = '未发货'">未发货</tab-item>
  <tab-item :selected="demo1 === '全部订单'" @click="demo1 = '全部订单'">全部订单</tab-item>
</tab>
```

script

```
export default {
  data () {
    return {
      demo1: '未发货'
    }
  }
}
```

更简洁的栗子

新窗口打开

{{item}}

template

```
<tab :line-width="2" active-color="#fc378c">
  <tab-item :selected="demo2 === item" v-for="item in list2" @click
="demo2 = item"></tab-item>
</tab>
```

script

```
export default {
  data () {
    return {
      demo2: '美食',
      list2: ['精选', '美食', '电影', '酒店', '外卖']
    }
  }
}
```

禁用滑动动画

新窗口打开

{{item}}

template

```
<tab :line-width="1" :animate="false">
  <tab-item :selected="demo2 === item" v-for="item in list2" @click
="demo2 = item"></tab-item>
</tab>
```

script

```
export default {
  data () {
    return {
      demo2: '美食',
      list2: ['精选', '美食', '电影', '酒店', '外卖']
    }
  }
}
```

Popup弹出层

Props

参数	说明	类型	默认值
show	是否显示Popup，需要双向绑定	Boolean	false
height	弹出层高度	String	auto

> 如果希望弹出层铺满整个屏幕，则可设置 `height=100%`

Events

名字	参数	描述
on-first-show	-	第一次出现时，用于需要在第一次进行异步数据获取或者必要的UI渲染（如Popup内有Scroller）

Demo

template

```
<div>
  <group>
    <switch title="Default popup" :value.sync="show"></switch>
    <switch title="Full popup" :value.sync="show1"></switch>
  </group>
  <popup :show.sync="show">
    <div class="popup0">
      <group>
        <switch title="Another Switcher" :value.sync="show"></swit
ch>
      </group>
    </div>
  </popup>
  <popup :show.sync="show1" height="100%">
    <div class="popup1">
      <group>
        <switch title="Another Switcher" :value.sync="show1"></swi
tch>
      </group>
    </div>
  </popup>
</div></code></pre>scriptexport default {
    data () {
        return {
            show: false,
            show1: false
        }
    }
}style.popup0 {
  padding-bottom:15px;
  height:200px;
}
.popup1 {
  width:100%;
  height:100%;
}
```

Sticky 自动固定在顶部

> 如果你在使用Chrome开发过程中发现并没有按预期起到作用，不要慌，这是正常现象！具体请参考#246

Demo

> 请打开新窗口查看效果

新窗口打开

正在正映 即将上映

template

```
<br v-for="i in 8">
<sticky>
    <tab :line-width=1>
      <tab-item selected>正在正映</tab-item>
      <tab-item>即将上映</tab-item>
    </tab>
</sticky>
<br v-for="i in 50">
```

Tabbar

Props

tabbar-item

属性	类型	默认	说明
selected	Boolean	false	是否选中
show-dot	Boolean	false	是否显示红色提示点
link	String or Object	-	普通链接或者 `v-link` 参数值，`0.1.2-rc1` 开始支持。 `<tabbar-item link="/somepath"></tabbar-item>`

Slots

名字	说明
icon	图标
label	文字

Demo

template

```
<div>
  <tabbar>
    <tabbar-item>
      <img slot="icon" src="https://o84lhz5xo.qnssl.com/master/src
/assets/demo/icon_nav_button.png">
      <span slot="label">Wechat</span>
    </tabbar-item>
    <tabbar-item show-dot>
      <img slot="icon" src="https://o84lhz5xo.qnssl.com/master/src
/assets/demo/icon_nav_msg.png">
      <span slot="label">Message</span>
    </tabbar-item>
    <tabbar-item selected>
      <img slot="icon" src="https://o84lhz5xo.qnssl.com/master/src
/assets/demo/icon_nav_article.png">
      <span slot="label">Explore</span>
    </tabbar-item>
    <tabbar-item>
      <img slot="icon" src="https://o84lhz5xo.qnssl.com/master/src
/assets/demo/icon_nav_cell.png">
      <span slot="label">News</span>
    </tabbar-item>
  </tabbar>
</div>
```

button-tab

Props

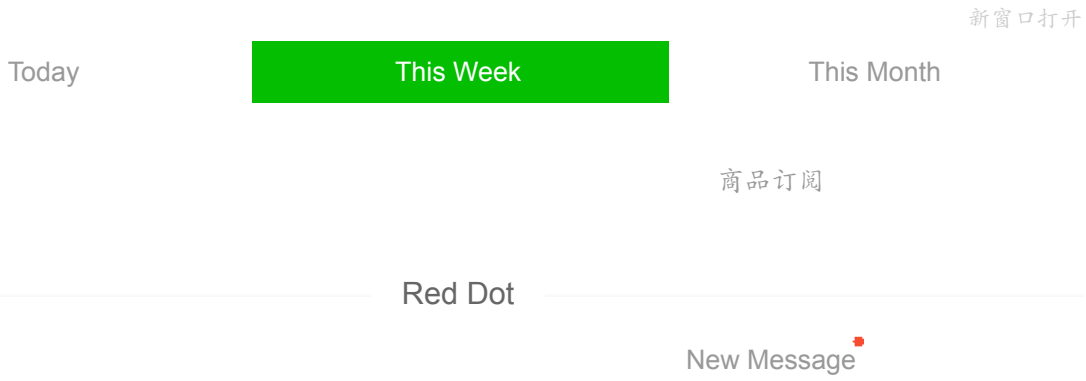
button-tab

属性	类型	默认	说明
height	Boolean	无	高度

button-tab-item

属性	类型	默认	说明
selected	Boolean	false	是否选中

Demos



template

```
<div>
  <button-tab>
    <button-tab-item>Today</button-tab-item>
    <button-tab-item selected>This Week</button-tab-item>
    <button-tab-item>This Month</button-tab-item>
  </button-tab>
  <br>
  <button-tab>
    <button-tab-item selected>文章订阅</button-tab-item>
    <button-tab-item>商品订阅</button-tab-item>
  </button-tab>
  <br>
  <divider>Red Dot</divider>
  <button-tab>
    <button-tab-item selected>All Message</button-tab-item>
    <button-tab-item><span class="vux-reddot-s">New Message</span>
</button-tab-item>
  </button-tab>
</div>
```

Timeline 时间轴

Props

timeline

参数	说明	类型	默认值
color	可选，线条颜色	String	#04BE02

Demo

新窗口打开

【广东】 广州市 已发出

2016-04-17 12:00:00

申通快递员 广东广州 收件员 **xxx** 已揽件

2016-04-16 10:23:00

商家正在通知快递公司揽件

2016-04-15 9:00:00

template

```
<div class="timeline-demo">
  <timeline :color="color">
    <timeline-item>
      <h4 class="recent">【广东】 广州市 已发出</h4>
      <p class="recent">2016-04-17 12:00:00</p>
    </timeline-item>
    <timeline-item>
      <h4> 申通快递员 广东广州 收件员 xxx 已揽件</h4>
      <p>2016-04-16 10:23:00</p>
    </timeline-item>
    <timeline-item>
      <h4> 商家正在通知快递公司揽件</h4>
      <p>2016-04-15 9:00:00</p>
    </timeline-item>
  </timeline>
</div>
```

动态改变条目数量

Timeline Node {{i + 1}}

index {{i + 1}}

Set to 6 nodes Set to 3 nodes

template

```
<div class="timeline-demo">
  <timeline>
    <timeline-item v-for="i in count">
      <h4 :class="[i === 0 ? 'recent' : '']">Timeline Node NaN</h4>

      <p :class="[i === 0 ? 'recent' : '']">index NaN</p>
    </timeline-item>
  </timeline>
  <x-button type="primary" @click="count = 6"> Set to 6 nodes</x-b
utton>
  <x-button type="primary" @click="count = 3"> Set to 3 nodes</x-b
utton>
</div>
```

script

```
export default {
  data () {
    return {
      count: 3
    }
  }
}
```

style

```
.timeline-demo p {
    color: #888;
    font-size: 0.8rem;
  }
  .timeline-demo h4 {
    color: #666;
    font-weight: normal;
  }
  .timeline-demo .recent {
    color: rgb(4, 190, 2);
  }</code></pre>
```

x-header

> 目前x-header并不处理定位，请手动用class或者style设置。

Props

名字	类型	默认	描述
leftOptions.showBack	Boolean	true	是否显示返回箭头
leftOptions.backText	String	Back	返回文字，可以为空
leftOptions.preventGoBack	Boolean	false	是否阻止点击Back时的后退，默认会调用 `history.back()`
rightOptions.showMore	Boolean	false	是否显示更多图标

Events

名字	参数	描述
on-click-back	-	点击后退按钮或者文字时触发, 并且只有leftOptions.preventGoBack设为true时才触发
on-click-more	-	点击更多图标时触发

Slots

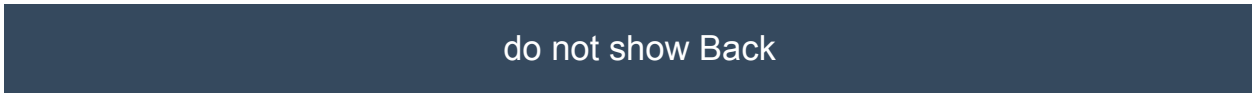
名字	描述
默认slot	标题文字
left	位于Back之后的内容
right	位于showMore之后的内容

Demos

新窗口打开

默认

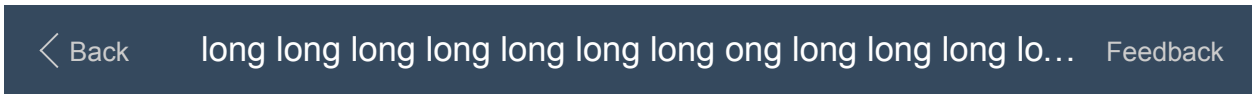
不显示后退



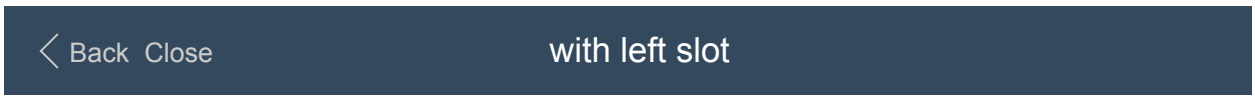
显示右侧更多

使用 right 插槽

自动文字截断



使用 left 插槽



更改背景颜色

template

```
<divider>默认</divider>
<x-header>This is the page title.</x-header>
<divider>不显示后退</divider>
<x-header :left-options="{showBack: false}">do not show Back</x-header>
<divider>显示右侧更多</divider>
<x-header :right-options="{showMore: true}">with more menu</x-header>
<divider>使用 right 插槽</divider>
<x-header>with right link<a slot="right">Feedback</a></x-header>
<divider>自动文字截断</divider>
<x-header>long long long long long long long ong long long long long long long text<a slot="right">Feedback</a></x-header>
<divider>使用 left 插槽</divider>
<x-header>with left slot<a slot="left">Close</a></x-header>
<divider>更改背景颜色</divider>
<x-header style="background-color:#000;">custom background color</x-header>
```

Divider

> 目前线条颜色不可配置。

Demo

新窗口打开

template

```
<divider>华丽分割线</divider>
<divider><span style="color:green;">华丽分割线</span></divider>
```

badge

Props

名字	类型	默认	描述
text	String	无	文字

Demo

新窗口打开

1

123

Used in a Cell New Message 8

template

```
<div>
  <badge text="1"></badge>
  <br>
  <badge text="123"></badge>
  <br>
  <group>
    <cell title="Used in a Cell">
      <div class="badge-value">
        New Message  
        <badge text="8"></badge>
      </div>
    </cell>
  </group>
</div>
```

icon

Props

名字	类型	默认	描述
type	String	无	类型, 可选值见demo

Demos

新窗口打开

```
<div>
  <icon type="success"></icon>
  <icon type="info"></icon>
  <icon type="info_circle"></icon>
  <icon type="warn"></icon>
  <icon type="waiting"></icon>
  <icon type="waiting_circle"></icon>
  <icon type="safe_success"></icon>
  <icon type="safe_warn"></icon>
  <icon type="success_circle"></icon>
  <icon type="success_no_circle"></icon>
  <icon type="circle"></icon>
  <icon type="download"></icon>
  <icon type="cancel"></icon>
  <icon type="search"></icon>
  <icon type="clear"></icon>
  <br/>
  <icon type="success" class="icon_big"></icon>
  <icon type="info" class="icon_big"></icon>
  <icon type="safe_success" class="icon_big"></icon>
  <icon type="safe_warn" class="icon_big"></icon>
</div>
```

1px边框解决方案

应用场景

1px边框问题来源于在Retina屏幕下边框宽度大于1像素，略丑，解决方案来源于 `FronzenUI`

一般用于 `九宫格` 等布局，在vux中配合 `Flexbox` 使用，常见于电商首页，支付工具服务列表页

使用

```
<style lang="less">
@import '~vux/src/styles/1px.less';
</style>
```

根据位置可用类名为：

- `vux-1px-t`
- `vux-1px-b`
- `vux-1px-tb`
- `vux-1px-l`
- `vux-1px-r`

示例

新窗口打开

template

```
<flexbox class="vux-1px-tb" :margin-left=0 style="height:50px;">
    <flexbox-item class="vux-1px-r test"><div>北京</div></flexbox-
item>
    <flexbox-item class="vux-1px-r test"><div>上海</div></flexbox-
item>
    <flexbox-item class="vux-1px-r test"><div>广州</div></flexbox-
item>
    <flexbox-item class="vux-1px-r test"><div>深圳</div></flexbox-
item>
    <flexbox-item class="test"><div>其他</div></flexbox-item>
  </flexbox>
  <flexbox class="vux-1px-b" :margin-left=0 style="height:50px;">
    <flexbox-item class="vux-1px-r test"><div>天津</div></flexbox-
item>
    <flexbox-item class="vux-1px-r test"><div>西安</div></flexbox-
item>
    <flexbox-item class="vux-1px-r test"><div>重庆</div></flexbox-
item>
    <flexbox-item class="vux-1px-r test"><div>杭州</div></flexbox-
item>
    <flexbox-item class="test"><div>其他</div></flexbox-item>
  </flexbox>
  <flexbox class="vux-1px-b" :margin-left=0 style="height:50px;">
    <flexbox-item class="vux-1px-r test"><div>南京</div></flexbox-
item>
    <flexbox-item class="vux-1px-r test"><div>武汉</div></flexbox-
item>
    <flexbox-item class="vux-1px-r test"><div>成都</div></flexbox-
item>
    <flexbox-item class="vux-1px-r test"><div></div></flexbox-item>

    <flexbox-item class="test"><div>其他</div></flexbox-item>
  </flexbox>
```

style

```
.test{
  height:50px;
  text-align:center;
  line-height: 50px;
}
```

Reddot

一般用于新消息提示。

Demo

新窗口打开

template

```
<div>
  <br>
  <div class="reddot-demo vux-reddot">Hi, you got a new message </
div>
  <br>
  <div class="reddot-demo vux-reddot-border">中文文字</div>
  <br>
  <div class="reddot-demo vux-reddot-s">small dot</div>
</div>
```

style

```
.reddot-demo {
  display: inline-block
}
```

Group

`Group` 是一个特殊的表单 `wrapper` 组件，主要用于将表单分组，单个表单元素也算一组。所以常见的 `行内` 组件都 `必须` 作为 `Group` 的子组件。

包括：

- Cell
- XInput
- XTextarea
- Switch
- Calendar
- XNumber
- Radio
- Address
- Datetime
- Selector

Props

参数	类型	默认	说明
title	String	无	分组标题
title-color	String	默认颜色	--

Demo

I'm group title

cell hello world

x-input

switch

x-textarea

Calendar

I'm another group title

cell hello world

x-input

x-number - 0 +

template

```
<group title="I'm group title">
  <cell title="cell" value="hello world" is-link></cell>
  <x-input title="x-input"></x-input>
  <switch title="switch"></switch>
  <x-textarea placeholder="x-textarea"></x-textarea>
  <calendar title="Calendar"></calendar>
</group>
<group title="I'm another group title" title-color="green">
  <cell title="cell" value="hello world" is-link></cell>
  <x-input title="x-input"></x-input>
  <x-number title="x-number"></x-number>
</group></code></pre>
```

Cell

Props

参数	说明	类型	默认值
title	可选，label文字	String	无
value	可选，右边文字	String	无
inline-desc	可选，label第二行文字	String	无
link	可选,支持http绝对路径及 `v-link` 配置	String or Object	无
is-link	可选，是否为链接，如果为true，样式上会出现箭头。当link存在时，is-link会自动设置为 `true`	Boolean	false
primary	可选，可选值为title和content, 对应的div会加上weui_cell_primary类名实现内容宽度自适应	String	title

Slots

名字	说明
默认slot	value区域
icon	title前，用于定义icon区域
after-title	title 后面区域
value	value区域, 同默认slot, 保留仅出于兼容考虑，不建议再使用

Demo

新窗口打开

template

```
<group>
  <cell title="My Account" value="Protected" @click="click"></cell>

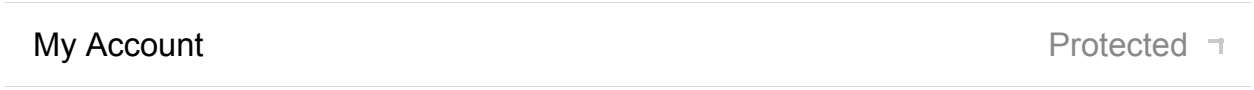
</group>
```

script

```
export default {
  methods: {
    click: function () {
      alert('click')
    }
  }
}
```

链接

新窗口打开



template

```
<group>
  <cell title="My Account" value="Protected" is-link></cell>
</group>
```

Address

> Address 依赖于 `Popup` , `Picker` , `Cell`

Props

名字	类型	默认	描述
title	String	无	标题
value	Array	[]	表单值， 双向绑定
list	Array	无	地址库, address源码目录下有 `list.json`
inline-desc	String	无	cell的子标题

Demos

```
<template>
  <div>
    <group>
      <address :title="title" :value.sync="value" :list="address
Data"></address>
      <cell title="上面value值" :value="value | json"></cell>
      <address :title="title2" :value.sync="value2" raw-value :l
ist="addressData"></address>
      <address title="二级省市" :value.sync="value3" raw-value :l
ist="addressData"></address>
    </group>
    <br/>
    <x-button type="primary" @click="changeData">改变数据</x-butt
on>
  </div>
</template>

<script>
import { Group, Address, AddressChinaData, XButton, Cell } from
'../components'
```

```
export default {
  components: {
    Group,
    Address,
    XButton,
    Cell
  },
  data () {
    return {
      title: '默认为北京',
      value: [],
      title2: '手动设定',
      value2: ['广东省', '深圳市', '南山区'],
      value3: ['海南省', '儋州市', '--'],
      addressData: AddressChinaData
    }
  },
  methods: {
    changeData () {
      this.value2 = ['430000', '430400', '430407']
    }
  }
}
</script>
```

x-button 按钮

Props

> 按钮文字可以用text属性也可以用直接用默认slot

参数	说明	类型	默认值
type	可选，可选值为default,primary,warn	String	default
disabled	可选，是否disabled	Boolean	false
text	可选，按钮文字	String	无
mini	可选，是否为小尺寸	Boolean	false
plain	可选，是否为plain样式(没有背景色)	Boolan	false

Slots

名字	说明
默认slot	按钮文字

一般使用

新窗口打开

submit

primary

Delete

template

```
<x-button>submit</x-button>
<x-button type="primary">primary</x-button>
<x-button type="warn">Delete</x-button>
```

不可点击

新窗口打开

submit

primary

Delete

template

```
<x-button disabled>submit</x-button>
<x-button type="primary" disabled>primary</x-button>
<x-button type="warn" disabled>Delete</x-button>
```

直接用:text指定按钮文字

新窗口打开

template

```
<x-button type="primary" :text="btnText" :disabled="isDisabled" @click="click"></x-button>
```

script

```
export default {
  data: {
    btnText: 'click me',
    isDisabled: false
  },
  methods: {
    click: function () {
      const _this = this
      this.isDisabled = true
      this.btnText = 'Processing'
      setTimeout(function () {
        _this.btnText = 'Done'
      }, 2000)
    }
  }
}
```

x-input 单行文本输入

> 命名为 `x-input` 避免与原生 `input` 标签渲染冲突
>
> 注意不要混淆： `x-input` 不是原生 `input` ,不能使用 `v-model` ,数据绑定语法为 `:value.sync` ,也不支持大多数 `input` 上的事件如 `focus` 等，如果实在需要处理input事件，可以直接参照WeUI文档直接用html标签。
>
> `x-input` 需要与 `group` 配合使用

Props

名字	类型	默认	描述
title	String	无	标题
value	String	无	表单值， 双向绑定
inline-desc	String	无	标题下文字
keyboard	String	无	只支持 `number` ,用于激活数字键盘
placeholder	String	无	输入提示
show-clear	Boolean	true	是否显示清除按钮
type	String	text	设置组件内input的type
is-type	String	无	内置验证类型，支持 `email` , `china-mobile` , `china-name`
readonly	Boolean	false	只读，不要修改
min	Number	无	最小字符数
max	Number	无	最大字符数
equal-with	String	无	当前input需要与某input输入完全一致，用于确认填写
text-align	String	left	input的对齐样式

Events

名字	参数	描述
on-change	(value)	当值改变时触发

Demos

```
<!-- 必须输入6-10位的电子邮件地址 -->
<x-input :min=6 :max=10 is-type=email :value.sync="value"></x-input>
<!-- 手机号码验证 -->
<x-input
    title="手机号"
    is-type="china-mobile"
    :show-clear=true
    placeholder="请输入手机号"></x-input>
<!-- 必须输入123456 -->
<x-input
    title="等值判断"
    type="text"
    equal-with="123456"></x-input>
```

获取验证状态

新窗口打开

template

```
<group title="check if value is valid when required===true">
  <x-input :value.sync="value" title="message" placeholder="I'm pl
aceholder" v-ref:input></x-input>
  <cell title="get valid value" :value="'$valid value:' + $refs.in
put.valid"></cell>
</group>
```

script

```
export default {
  data () {
    value: ''
  }
}
```

x-textarea 多行文本输入框

> 需要与 `group` 搭配使用

Props

参数	说明	类型	默认值	双向绑定
required	是否必填	Boolean	true	false
show-counter	是否显示字数统计（需要设置 max）	Boolean	true	false
max	最大字符数，超出字符数	Number		false
value	绑定的值字段	String	无	true
placeholder	placeholder	String	无	false
rows	行数	Number	3	--
cols	列数	30	Number	--
height	高度	Number	无	--

Demo

- 普通的多行输入框

新窗口打开

template

```
<group title="textarea">
  <x-textarea :max="200" placeholder="请输入详细描述"></x-textar
ea>
</group>
```

- 不显示字数统计

新窗口打开

template

```
<group title="不显示字符统计">
  <x-textarea :max="200" placeholder="请输入详细描述" :show-coun
ter="false"></x-textarea>
</group>
```

x-number 数字输入组件

> `x-number` 需要与 `group` 配合使用。

Props

参数	说明	类型	默认值
title	可选，标题	String	无
value	当前输入框值	Number	0
min	可选，数字范围最小值	Number	无
max	可选，数字范围最大值	Number	无
step	可选，步长	Number	1
fillable	可选，是否可以输入	Boolean	true
width	可选，输入框宽度	Number	50

Events

名字	参数	描述
on-change	value	值变化时触发

示例

基本使用

新窗口打开

Number - 0 +

template

```
<group>
  <x-number title="Number" :value=0 :min=0 :max=10 @on-change="cha
nge"></x-number>
</group>
```

script

```
export default {
  methods: {
    change: function (val) {
      console.log('change', val)
    }
  }
}
```

自定义步长

新窗口打开

template

```
<group title="set step=0.5">
  <x-number title="Number" :step=0.5></x-number>
</group>
```

禁止键盘输入

新窗口打开

fillable = false

Number - 10 +

template

```
<group title="fillable = false">
  <x-number :value=10 title="Number" :fillable=false></x-number>
</group>
```

和其他组件共用

新窗口打开

template

```
<group title='with other element'>
  <x-number title="Number" :min=-5 :max=8 :value=1 type="inline"></
x-number>
  <x-number title="Number" :min=-5 :max=8 :value=1 type="inline"></
x-number>
  <switch title="Other element" :value=true></switch>
</group>
```

checklist 多选

Props

参数	说明	类型	默认值
title	选项标题	String	无
required	可选，是否为必选项	Boolean	true
options	选项数组	Array	无
value	可选，已选择条目值	Array	无
max	可选，至多选择条目个数	Number	无
min	可选，至少选择条目个数	Number	无
random-order	可选，是否打乱条目排序	Boolean	false

> 当设置 `required=false` 时， `min` 设置将无效，即最少选定个数为0

Demo

基本使用

新窗口打开

template

```
<checklist title="请选择你的爱好" :options="hobbies" :value.sync="ho
bby" @change="change"></checklist>
```

script

```
export default {
  data () {
      return {
          hobbies: ['篮球', '足球', '羽毛球', '打飞机'],
          hobby: ['打飞机']
      }
  },
  methods: {
      change (val) {
          console.log('change', val)
      }
  }
}
```

设定选择条目个数

新窗口打开

template

```
<checklist title="至多选择两项" :options="items" :value.sync="selectedItems" :max=2 :required=false @change="change"></checklist>
```

script

```
export default {
  data () {
    return {
      items: ['篮球', '足球', '羽毛球', '台球'],
      selectedItems: []
    }
  },
  methods: {
    change (val) {
      console.log("change", val)
    }
  }
}
```

key-value类型数组展示

> 每个条目的 `key` 必须为字符串

新窗口打开

template

```
<checklist title="Please select" :options="objectList" :value.sync=
"objectListValue"></checklist>
```

script

```
export default {
    data () {
        return {
            objectList: [{key: '1', value: '001 value'}, {key: '2'
, value: '002 value'}, {key: '3', value: '003 value'}],
            objectListValue: ['1', '2']
        }
    }
}
```

打乱展示顺序

新窗口打开

template

```
<checklist title="随机顺序显示" :options="items" :value.sync="select
edItems" random-order></checklist>
```

script

```
export default {
  data () {
    return {
      items: ['篮球', '足球', '羽毛球', '台球'],
      selectedItems: []
    }
  }
}
```

selector

> 需要配合 `group` 使用

Props

名字	类型	默认	描述
title	String	无	标题
placeholder	String	无	placeholder
readonly	Boolean	false	只读
value	String	无	表单值， 双向绑定
options	Array	无	选项，支持简单数组及key=>value键值对

Events

名字	参数	描述
on-change	(value)	值变化时触发

Demos

template

```
<div>
  <group :title="'no placeholder, the current value is : ' + defau
ltValue">
    <selector title="省份" :options="list" :value.sync="defaultVal
ue"></selector>
  </group>
  <group title="with placeholder">
    <selector placeholder="请选择省份" title="省份" :options="list" @
on-change="onChange"></selector>
  </group>
  <group title="without title">
    <selector placeholder="请选择省份" :options="list"></selector>
  </group>
  <group title="set value=广西">
    <selector :value.sync="value1" title="省份" :options="plainLis
t" @on-change="onChange"></selector>
  </group>
  <group title="readonly, displays just like a cell">
```

```
    <selector value="gd" readonly title="省份" :options="list"></s
elector>
  </group>
  <group title="use plain options">
    <selector value="C" title="Selector" :options="list1" @on-chan
ge="onChange"></selector>
  </group>
  <group title='multi selector'>
    <selector placeholder="请选择省份" title="省份" :options="list">
</selector>
    <selector :value.sync="value2" title="省份" :options="list"></
selector>
  </group>
</div></code></pre>scriptexport default {
  data () {
    return {
      defaultValue: '',
      plainList: ['广东', '广西'],
      list: [{key: 'gd', value: '广东'}, {key: 'gx', value: '广西'}
],
      value1: '广西',
      value2: 'gd',
      list1: ['A', 'B', 'C']
    }
  },
  methods: {
    onChange (val) {
      console.log(val)
    }
  }
}
```

Switch

> 作为行内表单组件，`Switch` 必须和 `Group` 一起使用。

Props

名字	类型	默认	描述
value	Boolean	false	表单值, 双向绑定
title	String	无	cell标题
disabled	Boolean	false	是否禁止操作
inline-desc	String	无	副标题

Events

名字	参数	描述
on-change	(value)	值变化时触发

直接值

新窗口打开

template

```
<group>
  <switch title="Switch" :value=true></switch>
  <switch title="Switch" :value=false></switch>
</group>
```

双向绑定

新窗口打开

template

```
<group>
  <switch :title="'双向绑定:值为' + value1" :value.sync="value1"></s
witch>
  <switch :title="'双向绑定:值为' + value1" :value.sync="value1"></s
witch>
</group>
```

script

```
export default {
  data: {
    value1: true
  }
}
```

disabled 设置不可更改

新窗口打开

template

```
<group>
  <switch title="不可更改" :value=true disabled></switch>
  <switch title="不可更改" :value=false disabled></switch>
</group>
```

title支持html

新窗口打开

template

```
<group>
  <switch title="<span style='color:red'>红色文字</span>" :value=tr
ue></switch>
</group>
```

on-change 事件

新窗口打开

template

```
<group>
  <switch title="监听事件" :value=true @on-change="change"></switch>

</group>
```

script

```
export default {
  methods: {
    change: function (val) {
      console.log('change', val)
    }
  }
}
```

Rater 评分组件

Props

参数	说明	类型	默认值
max	可选，最高评分值	Number	5
value	可选，评分值，双向绑定	Number	0
disabled	可选，不可点击	Boolean	false
star	可选，评分图标样式	String	★
active-color	可选，激活颜色	String	#fc6
margin	可选，图标间距离	Number	2
fontSize	可选，图标大小	Number	25

一般使用

新窗口打开

template

```
<div>
    <rater :value=3></rater>
</div>
<div>
    <rater :value=2 :margin="15" active-color="#04BE02"></rater>
</div>
<div>
    <rater :value=5 :max=6 :font-size=15 active-color="#FF9900"></
rater>
</div>
```

style

```
div {
    margin:5px 0;
}
```

不可点击

新窗口打开

template

```
<div>
    <rater :value=4 disabled></rater>
</div>
<div>
    <rater :value=3.5  active-color="#04BE02" disabled></rater>
</div>
```

style

```
div {
    margin:5px 0;
}
```

自定义评分图标

新窗口打开

☼☼☼☼☼

囧囧囧囧囧

template

```
<div>
    <rater :value=5 star="☼" active-color="#FF9900"></rater>
</div>
<div>
    <rater :value=4 star="囧" active-color="#FF9900"></rater>
</div>
```

style

```
div {
    margin:5px 0;
}
```

双向绑定

新窗口打开

★★★☆☆
评分:3

template

```
<rater :value.sync="rate"></rater>
<div v-text="'评分:' + rate"></div>
```

script

```
export default {
    data: {
        rate:3
    }
}
```

Inline Calendar

Props

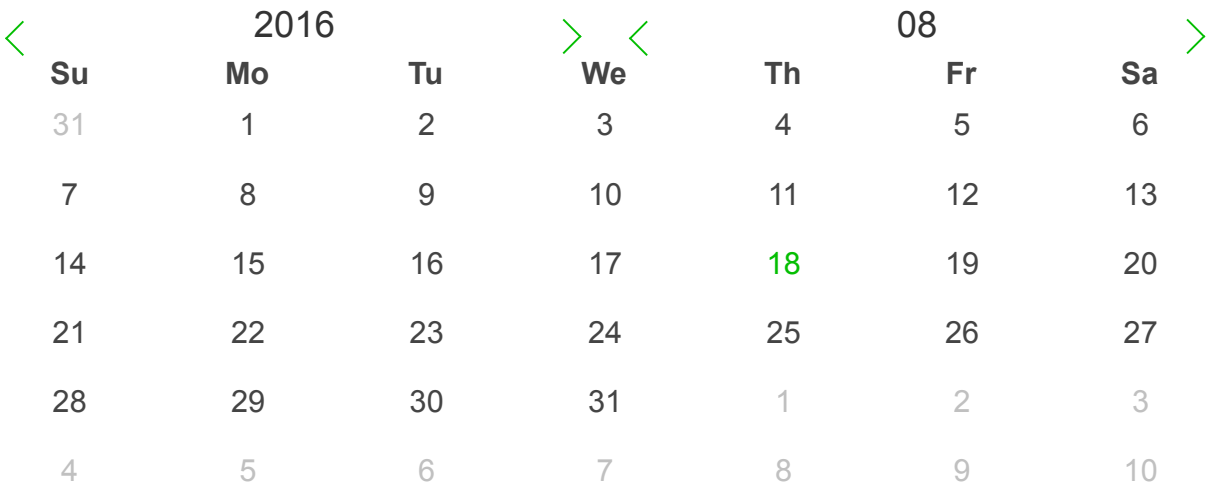
参数	类型	默认值	说明
value	String	无	当前选中日期，双向绑定，默认为空，即选中当天日期
start-date	String	无	可选起始日期，格式为'YYYY-MM-dd'
end-date	String	无	可选结束日期，格式为'YYYY-MM-dd'
render-month	Array	无	可选，指定渲染日期，如 [2018, 8]
show-last-month	Boolean	true	是否显示上个月的日期
show-next-month	Boolean	true	是否显示下个月的日期
highlight-weekend	Boolean	false	是否高亮周末
return-six-rows	Boolean	true	是否总是渲染6行日期
hideHeader	Boolean	false	是否隐藏日历头部
hiddeWeekList	Boolean	false	是否隐藏星期列表
replace-text-list	Object	{}	替换列表，可以将默认的日期换成文字，比如今天的日期替换成 `今` ，{'TODAY':'今'}
weeks-list	Array	['Su', 'Mo', 'Tu', 'We', 'Th', 'Fr', 'Sa']	导航星期列表，从周日开始
custom-slot-fn	Function	无	用于为特定日期添加额外的html内容，参数为(行index,列index,日期详细属性)
render-on-value-chhange	Boolean	true	当日期变化时是否重新渲染日历，如果是渲染了多个日历的话需要设为false
disable-past	Boolean	false	禁止选择过去的日期，该选项可以与start-date同时使用

demos

> PC文档样式会有偏差，后面修复。

设置开始和结束时间

新窗口打开

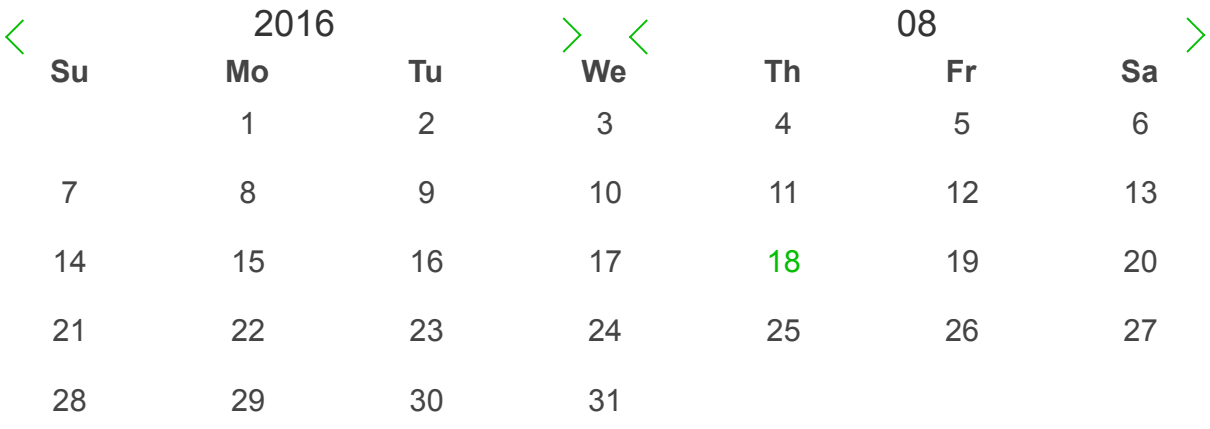


template

```
<inline-calendar start-date="2016-06-04" end-date="2017-06-18"></inline-calendar>
```

不显示上个月和下个月的日期

新窗口打开



template

```
<inline-calendar :show-next-month="false" :show-last-month="false">
</inline-calendar>
```

取值

新窗口打开

template

```
<inline-calendar :value.sync="value"></inline-calendar>
<group>
  <cell title="current value" :value="value"></cell>
</group>
```

script

```
export default {
  data () {
    return {
      value: 'TODAY'
    }
  }
}
```

Calendar 日历

> `Calendar` 组件直接依赖于 `InlineCalendar` + `Popup`

与Datetime的使用场景差别

- `calendar` 不支持具体钟点选择，因此需要选择具体时间时需要使用 `datetime`
- `calendar` 支持范围选择，如果使用 `datetime` 需要使用两次来分别选择开始时间和结束时间
- `calendar` 更适合于需要更好展现整个月信息及星期信息的情况，如跨月的日程安排， `datetime` 更适合于选择距离今天不远的日期场景，如近期任务安排

Props

日历相关属性同 `inline-calendar`

名字	类型	默认	描述
title	String	无	cell标题

Demo

```
<group>
  <calendar :value.sync="value" title="Date Picker"></calendar>
</group>
```

Radio 单选

> `Radio` 必须和 `Group` 一起使用。

Props

名字	类型	默认	描述
options	Array	无	选项列表, 支持简单数组及key=>value键值对，使用键值对时表单值为key
value	String	无	表单值，必选，双向绑定
fill-mode	Boolean	false	是否增加自定义输入框
fill-placeholder	String	无	自定义输入框的提示
fill-lable	String	无	自定义输入框标签

Events

名字	参数	描述
on-change	value	值变化时触发

预设值

新窗口打开

template

```
<group>
    <radio :options="options" value="China"></radio>
</group>
```

script

```
export default {
    data () {
        return {
            options:['China','Japan']
        }
    }
}
```

自定义输入框

新窗口打开

template

```
<group>
    <radio :options="options" fill-mode fill-label="Other" fill-placeholder="填写其他的哦"></radio>
</group>
```

script

```
export default {
    data () {
        return {
            options:['China','Japan']
        }
    }
}
```

数据双向绑定

新窗口打开

template

```
<group title="radio1">
    <radio :options="options" :value.sync="value"></radio>
</group>
<group title="radio2">
    <radio :options="options" :value.sync="value"></radio>
</group>
```

script

```
export default {
    data () {
        return {
            options:['China','Japan'],
            value:'China'
        }
    }
}
```

change事件

template

```
<group>
    <radio :options="options" @change="change"></radio>
</group>
```

script

```
export default {
    data () {
        return {
            options:['China','Japan']
        }
    },
    methods: {
        change (val) {
            console.log('change',val)
        }
    }
}
```

checker 灵活的单选组件

虽然有 `Radio` 组件提供单选功能，但是有很多情况下我们需要定义选项的样式布局，比如电商网站的商品属性选择。因此提供了 `checker` ，它提供了底层选择的抽象封装，使用者只需要考虑不同状态的样式。

> 当前组件由 `checker` 和 `checker-item` 组成
>
> 未来版本(可能在v0.2.0)将支持多选

checker 属性

参数	说明	类型
default-item-class	可选，应用于所有子选项	String
selected-item-class	必选，选中时的class	String
disabled-item-class	当有 `不可点击` 的选项时必选	String
value	必选，双向绑定	String
on-change	选中值变化时触发	参数为 `(value)`
on-item-click	当点击任何选项时触发	参数为 `(itemValue, isDisabled)`

checker-item 属性

参数	说明	类型	默认值
value	必选，选项值	String	
disabled	可选，是否不可点击	Boolean	false

使用

```
<checker :value.sync="somevalue"
         default-item-class="demo-item-defalut"
         selected-item-class="demo-item-selected">
  <checker-item value="1">1</checker-item>
  <checker-item value="2">2</checker-item>
  <checker-item value="3" disabled>3</checker-item>
</checker>
```

例子

新窗口打开

template

```
<checker :value.sync="demo1" default-item-class="demo1-item" selec
ted-item-class="demo1-item-selected">
  <checker-item value="1">1</checker-item>
  <checker-item value="2">2</checker-item>
  <checker-item value="3">3</checker-item>
</checker>
<br/>
<span>current value is: </span>
```

script

```
export default {
  data () {
    return {
      demo1: ''
    }
  }
}
```

style

```
.demo1-item {
  display: inline-block;
  border: 1px solid #ececec;
  padding: 5px 15px;
}
.demo1-item-selected {
  border: 1px solid green;
}
```

默认值

新窗口打开

template

```
<checker :value.sync="demo2" default-item-class="demo2-item" selec
ted-item-class="demo2-item-selected">
  <checker-item value="1">1</checker-item>
  <checker-item value="2">2</checker-item>
  <checker-item value="3">3</checker-item>
</checker>
<br/>
<span>current value is: </span>
```

script

```
export default {
  data () {
    return {
      demo2: '2'
    }
  }
}
```

style

```
.demo2-item {
  width: 40px;
  height: 40px;
  border: 1px solid #ccc;
  display: inline-block;
  border-radius: 50%;
  line-height: 40px;
  text-align: center;
}
.demo2-item-selected {
  border-color: green;
}
```

disabled

新窗口打开

template

```
<group>
  <cell title="select color" :value="demo4" is-link @click="showPo
pup=true"></cell>
</group>
<popup :show.sync="showPopup" class="checker-popup">
  <div style="padding:10px 10px 40px 10px;">
    <p style="padding: 5px 5px 5px 2px;color:#888;">Colors</p>
    <checker
    :value.sync="demo4"
    default-item-class="demo4-item"
    selected-item-class="demo4-item-selected"
    disabled-item-class="demo4-item-disabled"
    @on-item-click="showPopup=false">
      <checker-item value="花跟叶">花跟叶</checker-item>
      <checker-item value="鸟与树">鸟与树</checker-item>
      <checker-item value="我和你">我和你</checker-item>
      <checker-item value="全套礼品装" disabled>全套礼品装</checker-i
tem>
    </checker>
  </div>
</popup>
```

script

```
export default {
  data () {
    return {
      demo4: '花跟叶',
      showPopup: false
    }
  }
}
```

style

```
.demo4-item {
  display: inline-block;
  background-color: #ddd;
  color: #222;
  font-size: 14px;
  padding: 5px 10px;
  margin-right: 10px;
  line-height: 18px;
  border-radius: 15px;
}
.demo4-item-selected {
  background-color: #FF3B3B;
  color: #fff;
}
.demo4-item-disabled {
  color: #999;
}
```

datetime 时间

> 需要与 `group` 一起使用。

Props

名字	类型	默认	描述
format	String	'YYYY-MM-DD'	显示格式
title	String	无	显示标题
value	String	默认为当前日期	日期值
inline-desc	String	无	副标题
placeholder	String	无	提示文字
min-year	Number	无	最小可选年
max-year	Number	无	最大可选年
confirm-text	String	ok	确认按钮文字
cancel-text	String	cancel	取消按钮文字
year-row	String	{value}	年份显示格式
month-row	String	{value}	月份显示格式
day-row	String	{value}	日期显示格式
hour-row	String	{value}	小时显示格式
minute-row	String	{value}	分钟显示格式

demos

```
<group>
  <datetime title="select a date"></datetime>
</group>

<!-- 可选分钟 -->
<group>
```

```
  <datetime title="select minutes" format="YYYY-MM-DD HH:mm"></d
atetime>
</group>

<!-- 设置最小和最大可选年 -->
<group>
  <datetime
    title="specify min-year and max-year"
    :min-year=2000
    :max-year=2016
    value="please select"></datetime>
</group>

<!-- 自定义确认按钮和取消按钮文字 -->
<gropu>
  <datetime
    title="自定义按钮文字"
    confirm-text="确认"
    cancel-text="取消"></datetime>
</gropu>

<!-- 自定义时间显示文字 -->
<group>
  <datetime
    title="xx年xx月xx日xx点xx分"
    year-row="{value}年"
    month-row="{value}月"
    day-row="{value}日"
    hour-row="{value}点"
    minute-row="{value}分"></datetime>
</group>

<!-- change事件 -->
<group>
  <datetime
    title="change事件"
    @change="change"></datetime>
</group>

<script>
```

```
export default {
    methods:{
        change (val) {
            console.log('change',val)
        }
    }
}
</script>
```

Range 组件

Props

参数	说明	类型	默认值
decimal	可选，是否开启小数支持	Boolean	false
value	可选，当前选择值	Number	0
min	可选，取值范围最小值	Number	0
max	可选，取值范围最大值	Number	100
min-html	可选，最小值定制内容	String	无
max-html	可选，最大值定制内容	String	无
step	可选，滑动步长	Number	0
disabled	可选，是否为禁止状态	Boolean	false
disabled-opacity	可选，禁止状态下控件的透明度	Number	0.75
range-bar-height	可选，状态条的高度	Number	1
range-handle-height	可选，滑柄位置	Number	30

Demo

基本使用

新窗口打开

template

```
<group>
  <cell title="Default" :inline-desc="'value: '+value1" primary="c
ontent">
    <range slot="value" :value.sync="value1"></range>
  </cell>
  <cell title="allow decimals" :inline-desc="'value is: '+value2"
primary="content">
    <range slot="value" :value.sync="value2" decimal></range>
  </cell>
  <cell title="value=20" :inline-desc="'value is: '+value3" primary
="content">
    <range slot="value" :value.sync="value3"></range>
  </cell>
</group>
```

script

```
export default {
  data () {
    return {
      value1: 0,
      value2: 0,
      value3: 20
    }
  }
}
```

设置起始值和最大值

新窗口打开

template

```
<group>
  <cell title="min=8" :inline-desc="'value is: '+value1" primary="
content">
    <range slot="value" :value.sync="value1" :min=8></range>
  </cell>
  <cell title="max=88" :inline-desc="'value is: '+value2" primary=
"content">
    <range slot="value" :value.sync="value2" :max=88></range>
  </cell>
  <cell title="min and max" :inline-desc="'value is: '+value3" pri
mary="content">
    <range slot="value" :value.sync="value3" :min=7 :max=77></range
>
  </cell>
</group>
```

script

```
export default {
  data () {
    return {
      value1: 25,
      value2: 25,
      value3: 25
    }
  }
}
```

自定义步长

新窗口打开

template

```
<group>
  <cell title="step=10" :inline-desc="'valus is: '+value" primary=
"content">
    <range slot="value" :value.sync="value" :min=7 :max=77 :step=10
></range>
  </cell>
</group>
```

script

```
export default {
  data () {
    return {
      value: 25
    }
  }
}
```

禁用组件

新窗口打开

template

```
<group>
    <cell title="disabled=true" :inline-desc="'valus is: '+value1"
primary="content">
      <range slot="value" :value.sync="value1" disabled></range>
    </cell>
    <cell title="Opacity" :inline-desc="'valus is: '+value2" prima
ry="content">
      <range slot="value" :value.sync="value2" disabled :disabled-
opacity=0.1></range>
    </cell>
  </group>
```

script

```
export default {
  data () {
    return {
      value1: 25,
      value2: 25
    }
  }
}
```

设置bar的高度和手柄的位置

新窗口打开

template

```
<group>
  <cell title="Bar height" :inline-desc="'value is: '+value" prima
ry="content">
    <range slot="value" :value.sync="value" :range-bar-height=4></
range>
  </cell>
  <cell title="Handle position" :inline-desc="'value is: '+value"
primary="content">
    <range slot="value" :value.sync="value" :range-handle-height=5>
</range>
  </cell>
</group>
```

script

```
export default {
  data () {
    return {
      value: 25
    }
  }
}
```

自定义初始值和最大值

新窗口打开

template

```
<group>
  <cell title="文字大小" :inline-desc="'font size: ' + value1" prim
ary="content">
    <range slot="value" :value.sync="value1" :min="12" :max="22" m
in-HTML="<span style='font-size:12px;'>小</span>" max-HTML="<span
style='font-size:22px;'>大</span>"></range>
  </cell>
  <cell title="bcontentness" :inline-desc="'value is: ' + value2 +
 '%'" primary="content">
    <range slot="value" :value.sync="value2" min-HTML="<span style
='font-size:16px;color:#F90;'>☼</span>" max-HTML="<span style='fon
t-size:30px;color:#F90;'>☼</span>"></range>
  </cell>
</group>
```

script

```
export default {
  data () {
    return {
      value1: 14,
      value2: 25
    }
  }
}
```

双向绑定

新窗口打开

template

```
<group>
  <cell title="Default" :inline-desc="'value: '+value" primary="co
ntent">
    <range slot="value" :value.sync="value"></range>
  </cell>
  <cell title="Default" :inline-desc="'value: '+value" primary="co
ntent">
    <range slot="value" :value.sync="value"></range>
  </cell>
</group>
```

script

```
export default {
  data () {
    return {
      value: 0
    }
  }
}
```

倒计时

常用场景

- 活动报名、商品折扣时间倒计时
- 有效支付时间倒计时

选项

示例

一般使用

新窗口打开

距离2017-04-01还有
225 天 10 小时 03 分 49 秒

template

```
<p style="padding:15px;">
  <span>距离2017-04-01还有</span>
  <clocker time="2017-04-01"></clocker>
</p>
```

在Cell中使用

新窗口打开

20170401	225 天 10 小时 03 分 49 秒

template

```
<group>
  <cell title="20170401">
    <clocker time="2017-04-01" slot="value"></clocker>
  </cell>
</group>
```

自定义html模板

> 分割数字只支持两位日期，如果有两位以上，可以考虑使用 `on-tick` 事件来处理

新窗口打开

Date:0501	00 天 00 小时 00 分 00 秒
分割两位数字	71天10时03分49秒

template

```
<group>
  <cell title="Date:0501">
    <clocker time="2016-05-01" slot="value">
      <span style="color:red">%D 天</span>
      <span style="color:green">%H 小时</span>
      <span style="color:blue">%M 分 %S 秒</span>
    </clocker>
  </cell>
  <cell title="分割两位数字">
    <clocker time="2018-08-08" slot="value">
      <span class="day">%_D1</span>
      <span class="day">%_D2</span>天
      <span class="day">%_H1</span>
      <span class="day">%_H2</span>时
      <span class="day">%_M1</span>
      <span class="day">%_M2</span>分
      <span class="day">%_S1</span>
      <span class="day">%_S2</span>秒
    </clocker>
  </cell>
</group>
```

style

```
.day {
  background-color:#000;
  color:#fff;
  text-align:center;
  display:inline-block;
  padding:0 3px;
  border-radius:3px;
}
```

秒数倒计时

使用场景

> 用于发送短信等耗费操作时，不要依赖前端验证，`必须一定要`同时在后端进行ratelimit验证。

- 短信验证重发的场景中。一般为60s, 120s

API

参数	说明	类型	默认值
time	可选，计时秒数	Number	60
start	可选，默认自动倒计时，设为false时可以手动开始计时	Boolean	true

Demo

新窗口打开

template

```
<group>
  <cell title="15s" :value="value">
    <countdown slot="value" :time="15" @on-finish="finish" v-show="show"></countdown>
  </cell>
</group>
```

script

```
export default {
  methods: {
    finish: function (index) {
      this.show = false
      this.value = 'completed'
      console.log('current index', index)
    }
  },
  data () {
    return {
      show: true,
      value: ''
    }
  }
}
```

新窗口打开

template

```
<group>
  <switch title="start" :value.sync="start"></switch>
  <cell title="15s">
    <countdown slot="value" :time="time" :start="start" @on-finish=
"finish"></countdown>
  </cell>
</group>
```

script

```
export default {
  methods: {
    finish: function (index) {
      this.start = false
      this.time = 20
    }
  },
  data () {
    return {
      time: 15,
      start: false
    }
  }
}
```

轻量日期格式化

轻量简单的日期格式化工具函数，毕竟很多时候我们并不想引入 `moment` 这个大块头。

可以引入 `DateFormatter` 组件作为 `filter`，参数为日期格式和时间及字符串格式。

新窗口打开

template

```
<div>
  <span v-text="time1 | date-formatter 'YYYY-MM-DD'"></span>
  <br/>
  <span v-text="time2 | date-formatter 'YYYY-MM-DD HH:mm:ss'"></sp
an>
</div>
```

script

```
export default {
  data: {
    time1: new Date('2016/01/03 19:19:19'),
    time2: new Date('2016-01-03 09:09:09')
  }
}
```

number-roller

属性

参数	说明	类型	默认值
number	可选(异步赋值)，选项值	Number	0
width	必选，数字位数	Number	--

demo

新窗口打开

template

```
<number-roller :number="number" :width="6"></number-roller>
```

script

```
export default {
  ready () {
    setInterval(() => {
      this.number = 100000 + Math.round(Math.random() * 899999)
    }, 3000)
  },
  data () {
    return {
      number: 123765
    }
  }
}
```

Loading

Props

参数	类型	默认	说明
show	Boolean	false	是否显示，`双向绑定`
text	String or DOM	Loading	提示文字，与默认slot作用一致

Slots

名字	说明
默认slot	提示文字，因和text属性功能一致，因此可以二选一

Demo

新窗口打开

template

```
<div>
  <group>
    <switch title="Toggle" :value.sync="show" @on-change="delayHid
e"></switch>
  </group>
  <loading :show="show" :text="text1"></loading>
</div>
```

script

```
export default {
  methods: {
    delayHide () {
      setTimeout(() => {
        this.show = false
      }, 5000)
    }
  },
  data () {
    return {
      show: false,
      text1: 'Hello world'
    }
  }
}
```

Alert

Props

参数	说明	类型	默认值
button-text	可选，按钮文字	String	OK
默认slot	可选，提示消息内容	DOM	无
title	必选，提示标题	String	无
show	必选，是否显示， 双向绑定	Boolean	false
on-show	显示时事件	事件	无
on-hide	关闭时事件	事件	无

Demo

默认按钮文字

新窗口打开

中大奖了！99999元只要转4000元手续费

template

```
<group>
  <switch title="Toggle" :value.sync="show"></switch>
</group>
<alert :show.sync="show" title="恭喜您" button-text="好棒，去ATM转账">

  <p style="text-align:center;">中大奖了！99999元只要转4000元手续费</p>

</alert>
```

script

```
export default {
  data () {
    return {
      show: true
    }
  }
}
```

Confirm

使用场景

confirm用于需要用户确认操作的情况。

Props

参数	说明	类型	默认值
cancel-text	可选，取消按钮文字	String	cancel
confirm-text	可选，确认按钮文字	String	confirm
默认slot	可选，提示消息内容	DOM	无
title	必选，提示标题	String	无
show	必选，是否显示，`双向绑定`	Boolean	false
on-confirm	确认事件	事件	无
on-cancel	取消事件	事件	无

> 事件名字从 `0.0.105` 后从 `confirm` 和 `cancel` 重命名为 `on-confirm` 和 `on-cancel`

Demo

默认按钮文字

新窗口打开

template

```
<group>
  <switch title="Toggle" :value.sync="show"></switch>
</group>
<confirm :show.sync="show" title="confirm deleting the item"><p st
yle="text-align:center;">Are you sure?</p></confirm>
```

script

```
export default {
  data () {
    return {
      show: false
    }
  }
}
```

监听事件

新窗口打开

template

```
<group>
  <switch title="Toggle" :value.sync="show"></switch>
</group>
<confirm :show.sync="show" confirm-text="确定" cancel-text="取消" t
itle="操作提示" @on-confirm="onAction('确认')" @on-cancel="onAction(
'取消')">
  <p style="text-align:center;">操作不可撤消哦?</p>
</confirm>
```

script

```
export default {
  methods: {
    onAction: function (type) {
      alert(type)
    }
  },
  data () {
    return {
      show: false
    }
  }
}
```

Toast

Props

参数	类型	默认	说明
show	Boolean	false	是否显示，`双向绑定`
time	Nummber	2000	显示时间
type	String	success	图标类型，可选为 `success` , `text` , `cancel` , `warn`
transition	String	vux-fade	动画
width	String	`7.6em`	Toast宽度

> 尽管Toast组件提供了 `cancel` 和 `warn` 类型，但不推荐使用，需要用户关注的通知推荐使用 `Alert` 或者 `Confirm`

Slot

名字	说明
默认slot	提示文字

Demo

默认按钮文字

默认提示 处理成功 取消操作 禁止操作 1s关闭 Talk is cheap, show me the code.

template

```
<div>
  <group>
    <switch title="默认提示" :value.sync="show1"></switch>
    <switch title="文字提示" :value.sync="show2"></switch>
    <switch title="提示取消" :value.sync="show3"></switch>
    <switch title="提示禁止" :value.sync="show4"></switch>
    <switch title="设置出现时间1s" :value.sync="show5"></switch>
    <switch title="long text" :value.sync="show6"></switch>
  </group>
  <toast :show.sync="show1" >默认提示</toast>
  <toast :show.sync="show2" type="text">处理成功</toast>
  <toast :show.sync="show3" type="cancel">取消操作</toast>
  <toast :show.sync="show4" type="warn">禁止操作</toast>
  <toast :show.sync="show5" :time="1000">1s关闭</toast>
  <toast :show.sync="show6" type="text" width="20em">Talk is cheap
, show me the code.</toast>
</div>
```

script

```
export default {
  data () {
    return {
      show1: false,
      show2: false,
      show3: false,
      show4: false,
      show5: false,
      show6: false
    }
  }
}
```

Actionsheet

Props

参数	类型	默认值	说明
show	Boolean	false	显示绑定值，双向绑定
show-cancel	Boolean	false	是否显示取消按钮
menus	Object	{}	菜单列表，格式见下
cancelText	String	cancel	取消按钮文字

> menus格式如下

```
{
  menu1: 'Take Photo',
  menu2: 'Choose from photos'
}
```

Events

事件名	参数	说明
on-menu-click	(menuKey, menuValue)	点击菜单时触发
on-cancel	--	点击取消时触发

Demo

template

```
<div>
  <group>
    <switch title="show actionsheet1" :value.sync="show1"></switch>

    <switch title="show actionsheet2" :value.sync="show2"></switch>

    <switch title="show actionsheet3" :value.sync="show3"></switch>

  </group>
  <actionsheet :show.sync="show1" :menus="menus1"></actionsheet>
  <!-- 显示取消按钮 -->
  <actionsheet :show.sync="show2" :menus="menus2" show-cancel></ac
tionsheet>
  <!-- 菜单响应 -->
  <actionsheet :show.sync="show3" :menus="menus3" @on-click-menu="
click" show-cancel @cancel-text="取消"></actionsheet>
</div>
```

script

```
export default {
  data () {
    return {
      show1:false,
      menus1:{
        menu1: 'Share to friends',
        menu2: 'Share to timeline'
      },
      show2:false,
      menus2: {
        menu1: 'Take Photo',
        menu2: 'Choose from photos'
      },
      show3:false,
      menus3: {
        menu1: 'friends comment'
      }
    }
  },
  methods: {
    click (key) {
      console.log(key)
    }
  }
}
```

Dialog

`Alert` 及 `Confirm` 依赖于 `Dialog` 。

Props

参数	类型	默认值	说明
show	Boolean	false	是否显示，`双向绑定`
mask-transition	String	vux-fade	遮罩动画
dialog-transition	String	vux-dialog	窗口动画

Events

事件名	参数	说明
on-show	无	显示时触发
on-hide	无	关闭时触发

Slots

名字	说明
默认slot	弹窗主体内容

Demo

```
<template>
  <div>
    <dialog :show.sync="show" class="dialog-demo">
      <p class="dialog-title">I'm a Dialog.</p>
    </dialog>
  </div>
</template>
```

Spinner

Props

文档模板

Props

名字	类型	默认	描述
type	String	ios	类型,可选值有 'android', 'ios', 'ios-small', 'bubbles', 'circles', 'crescent', 'dots', 'lines', 'ripple', 'spiral'

Demos

template

```
<div>
  <group>
    <cell v-for="type in types" :title="type">
      <spinner :type="type"></spinner>
    </cell>
  </group>
</div>
```

script

```
export default {
  data () {
    return {
      types: ['android', 'ios', 'ios-small', 'bubbles', 'circles',
'crescent', 'dots', 'lines', 'ripple', 'spiral']
    }
  }
}
```

背景模糊

场景

背景模糊常用于个人中心的头像背景, 音乐播放界面的背景。

Props

名字	类型	默认	描述
url	String	无	图片地址
height	Number	200	高度
blur-amount	Number	10	模糊程度

Slots

名字	描述
默认slot	图片上面内容

示例

纯背景

新窗口打开

template

```
<blur :blur-amount=40 url="https://o3e85j0cv.qnssl.com/tulips-1083572__340.jpg"></blur>
```

背景上添加内容

> 通过默认slot支持

新窗口打开

template

```
<blur :blur-amount=40 url="https://o3e85j0cv.qnssl.com/hot-chocolate-1068703__340.jpg">
  <p class="center">
    <img src="https://o3e85j0cv.qnssl.com/hot-chocolate-1068703__340.jpg">
  </p>
</blur>
```

style

```
.center {
  text-align: center;
  padding-top: 20px;
  color: #fff;
  font-size: 18px;
}
.center img {
  width: 100px;
  height: 100px;
  border-radius: 50%;
  border: 2px solid #ececec;
}
```

半透明遮罩

Props

名字	类型	默认	描述
color	String	0,0,0	颜色
opacity	Number	0.5	透明度

Slots

名字	描述
默认slot	遮罩层下面的内容
content	遮罩层上面的内容

场景

常见于电商网站，摄影网站的列表展示。

template

```
<div>
  <div style="margin: 10px;overflow: hidden;" v-for="item in list">

    <masker style="border-radius: 2px;" :opacity="0.5">
      <div class="m-img" :style="{backgroundImage: 'url(' + item.i
mg + ')'}"></div>
      <div slot="content" class="m-title"><br/>
        <span class="m-time">2016-03-18</span>
      </div>
    </masker>
  </div>
  <div style="margin: 10px;overflow: hidden;">
    <masker style="border-radius: 2px;" color="F9C90C" :opacity="0
.6">
      <div class="m-img" style="background-image:url(https://cdn.x
iaotaojiang.com/uploads/56/4b3601364b86fdfd234ef11d8712ad/_.jpg)">
</div>
      <div slot="content" class="m-title">
        VUX
        <br/>
        <span class="m-time">2016-03-18</span>
      </div>
    </masker>
  </div>
</div>
```

script

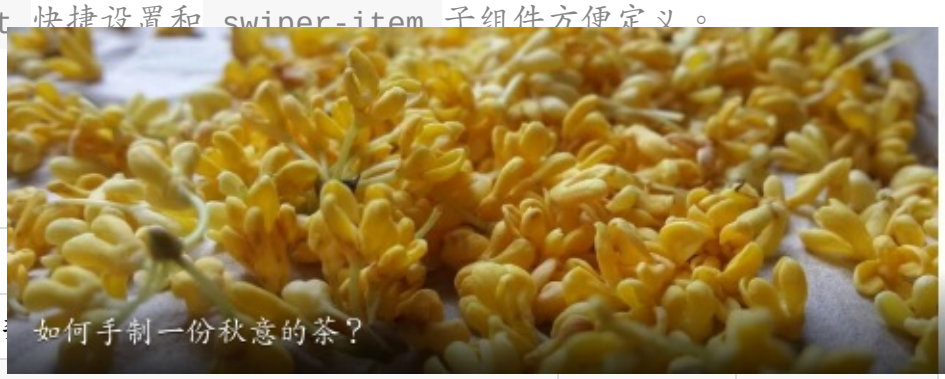
```
export default {
  data () {
    return {
      list: [{
        title: '洗颜新潮流！人气洁面皂排行榜',
        img: 'https://cdn.xiaotaojiang.com/uploads/82/1572ec37969e
e263735262dc017975/_.jpg'
      }, {
        title: '美容用品 & 日用品（上）',
        img: 'https://cdn.xiaotaojiang.com/uploads/59/b22e0e62363a
4a652f28630b3233b9/_.jpg'
      }, {
        title: '远离车内毒气，日本车载空气净化器精选',
        img: 'https://cdn.xiaotaojiang.com/uploads/56/4b3601364b86
fdfd234ef11d8712ad/_.jpg'
      }]
    }
  }
}
```

style

```
.m-img {
  padding-bottom: 33%;
  display: block;
  position: relative;
  max-width: 100%;
  background-size: cover;
  background-position: center center;
  cursor: pointer;
  border-radius: 2px;
}
.m-time {
  font-size: 12px;
  padding-top: 4px;
  border-top: 1px solid #f0f0f0;
  display: inline-block;
  margin-top: 5px;
}
.m-title {
  color: #fff;
  text-align: center;
  text-shadow: 0 0 2px rgba(0,0,0,.5);
  font-weight: 500;
  font-size: 16px;
  position: absolute;
  left: 0;
  right: 0;
  width: 100%;
  text-align: center;
  top: 50%;
  transform: translateY(-50%);
}</code></pre>
```

swiper

> swiper提供了 `list` 快捷设置和 `swiper-item` 子组件方便定义。



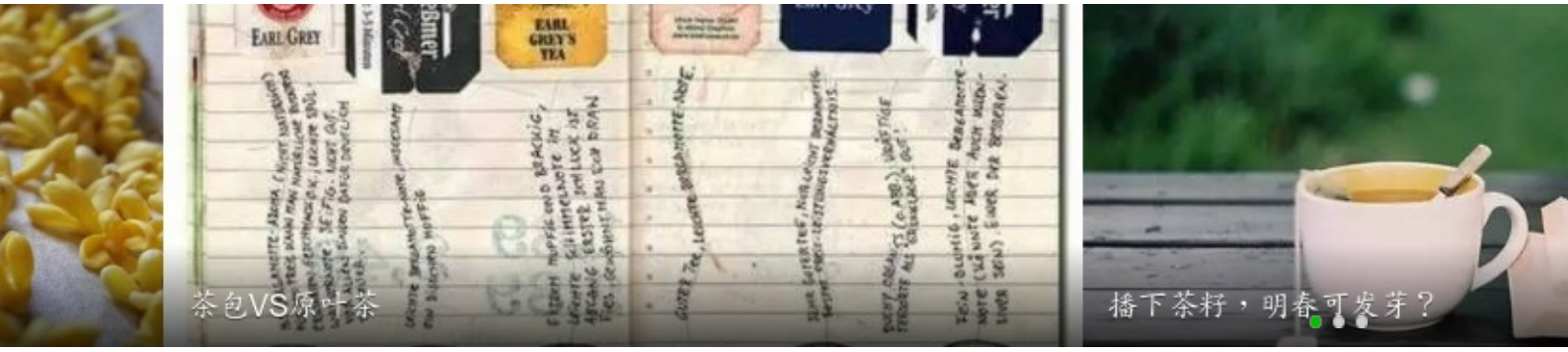
Props

参数			
list	列表		
direction	方向	String	horizontal
show-dots	是否显示indicator,只在list快捷形式下显示	Boolean	true
show-desc-mask	是否显示描述梦曾	Boolean	true
dots-position	indicator位置	String	right
dots-class	indicator的附加样式类	String	无
auto	是否自动播放	Boolean	false
loop	是否循环播放	Boolean	无
aspect-ratio	纵横比,设置则自动根据宽度计算高度	Number	无
min-moving-distance	最小滑动距离	Number	0
index	指定显示item的	Number	0
interval	轮播时间间隔	Number	3000
threshold	滑动距离阀值，当按住屏幕滑动超过此距离，松开手时，自动滑，否则不滑动	Number	50
duration	滑屏动画时间，单位ms，数值越小，滑动越快	Number	300
height	容器高度	String	auto

示例

基本使用

新窗口打开



template

```
<div>
  <swiper :list="list" auto></swiper>
<div>
```

script


```
export default {
  data: function () {
    return {
      list: [{
        url: 'http://mp.weixin.qq.com/s?__biz=MzAxNjU0MDYxMg==&amp
                                        6b8d99715384bdcc7746596d88359&amps

                                        .z0.glb.qiniucdn.com/1.jpg',
                                        ？'

                                        q.com/s?__biz=MzAxNjU0MDYxMg==&amp
                                        f02af25793a11a3f6aec92bfb46c1&amps
cene=19#wechat_redirect',
        img: 'http://7xqzw4.com2.z0.glb.qiniucdn.com/2.jpg',
        title: '茶包VS原叶茶'
      }, {
        url: 'http://mp.weixin.qq.com/s?__biz=MzAxNjU0MDYxMg==&amp
mid=400094682&ampidx=1&ampsn=8231a2053b772b2108784fccc254d28c&amps
cene=19#wechat_redirect',
        img: 'http://7xqzw4.com2.z0.glb.qiniucdn.com/3.jpg',
        title: '播下茶籽，明春可发芽？'
      }]
    }
  }
}
```

表情图片

选项

- `is-gif` 是否显示动态图，默认为 `false` 显示静态图

Demo

新窗口打开

{{txt}}
{{txt}}

template

```
<div>
    <wechat-emotion v-for="txt in list" track-by="$index"></wechat
-emotion>
    <br>
    <wechat-emotion v-for="txt in list" track-by="$index" is-gif></
wechat-emotion>
  </div>
```

script

```
export default {
  data () {
    return {
      list: ['微笑', '撇嘴', '色', '发呆', '得意', '流泪', '害羞', '闭嘴', '睡', '大哭', '尴尬', '发怒', '调皮', '呲牙', '惊讶', '难过', '酷', '冷汗', '抓狂', '吐', '偷笑', '可爱', '白眼', '傲慢', '饥饿', '困', '惊恐', '流汗', '憨笑', '大兵', '奋斗', '咒骂', '疑问', '嘘', '晕', '折磨', '衰', '骷髅', '敲打', '再见', '擦汗', '抠鼻', '鼓掌', '糗大了', '坏笑', '左哼哼', '右哼哼', '哈欠', '鄙视', '委屈', '快哭了', '阴险', '亲亲', '吓', '可怜', '菜刀', '西瓜', '啤酒', '篮球', '乒乓', '咖啡', '饭', '猪头', '玫瑰', '凋谢', '示爱', '爱心', '心碎', '蛋糕', '闪电', '炸弹', '刀', '足球', '瓢虫', '便便', '月亮', '太阳', '礼物', '拥抱', '强', '弱', '握手', '胜利', '抱拳', '勾引', '拳头', '差劲', '爱你', 'NO', 'OK', '爱情', '飞吻', '跳跳', '发抖', '怄火', '转圈', '磕头', '回头', '跳绳', '挥手', '激动', '街舞', '献吻', '左太极', '右太极']
    }
  }
}
```

横竖屏切换提示

此组件以指令的形式控制DOM在横竖屏下的显示，支持的选项有

- landscape - 当横屏时显示
- portrait - 当竖屏时显示

查看在线DEMO

```
<template>
  <div>
    <div v-orientation="landscape" class="landscape"><div>竖屏观看更合适哦</div></div>
    <div v-orientation="portrait" class="portrait"><div>翻转手机看效果</div></div>
  </div>
</template>
<script>
export default {
  directives: {
    Orientation
  }
}
</script>
```

Shake 摇一摇

API

参数	说明	类型	默认值
stop	是否停止派发摇动事件	Boolean	false
threshold	可选，摇动力度阈值	Number	15
timeout	可选，事件收集频率	Number	1000

Demo

新窗口打开

shake your phone

template

```
<div>
  <p style="text-align:center">shake your phone</p>
  <p><x-button text="停止接收摇动事件" type="primary" @click="stopSh
ake = false"></x-button></p>
  <shake @shake="shake" :threshold=5></shake>
</div>
```

script

```
export default {
  data () {
    return {
      stopShake: false
    }
  },
  methods: {
    'on-shake': function () {
      alert('shaked')
    }
  }
}
```

圆圈

应用场景

圆圈组件一般用于显示进度，展示百分比数据。

示例

简单示例

新窗口打开

template

```
<div style='width:100px;height:100px;'>
  <circle :percent=30 :stroke-width=5 stroke-color=#04BE02>
    <span></span>
  </circle>
</div>
```

添加中间内容

> 通过默认slot支持

新窗口打开

{{percent2}}%

template

```
<div style='width:100px;height:100px;'>
  <circle :percent='percent2' :stroke-width=6 :trail-width=6 :stro
ke-color='strokeColor2' trail-color="#ececec">
    <span :style="{color: strokeColor2}">%</span>
  </circle>
</div>
```

script

```
export default {
  ready () {
    setInterval(this.update, 2000)
  },
  data () {
    return {
      percent2: 30,
      strokeColor2: '#3FC7FA'
    }
  },
  methods: {
    update: function () {
      const colorMap = ['#3FC7FA', '#85D262', '#FE8C6A']
      this.percent2 = parseInt(Math.random() * 100, 10)
      this.strokeColor2 = colorMap[parseInt(Math.random() * 3, 10)
]
    }
  }
}
```

http请求

vue有作者开发的 vue-resource 可以直接使用，当然也可以选择 `Zepto` 或者 `jQuery` ， `fetch` 的polyfill，甚至自己封装的库，具体考虑实际的开发情况，用于移动端在理想情况下体积越小越好。

路由

请参照 `vue-router` 文档