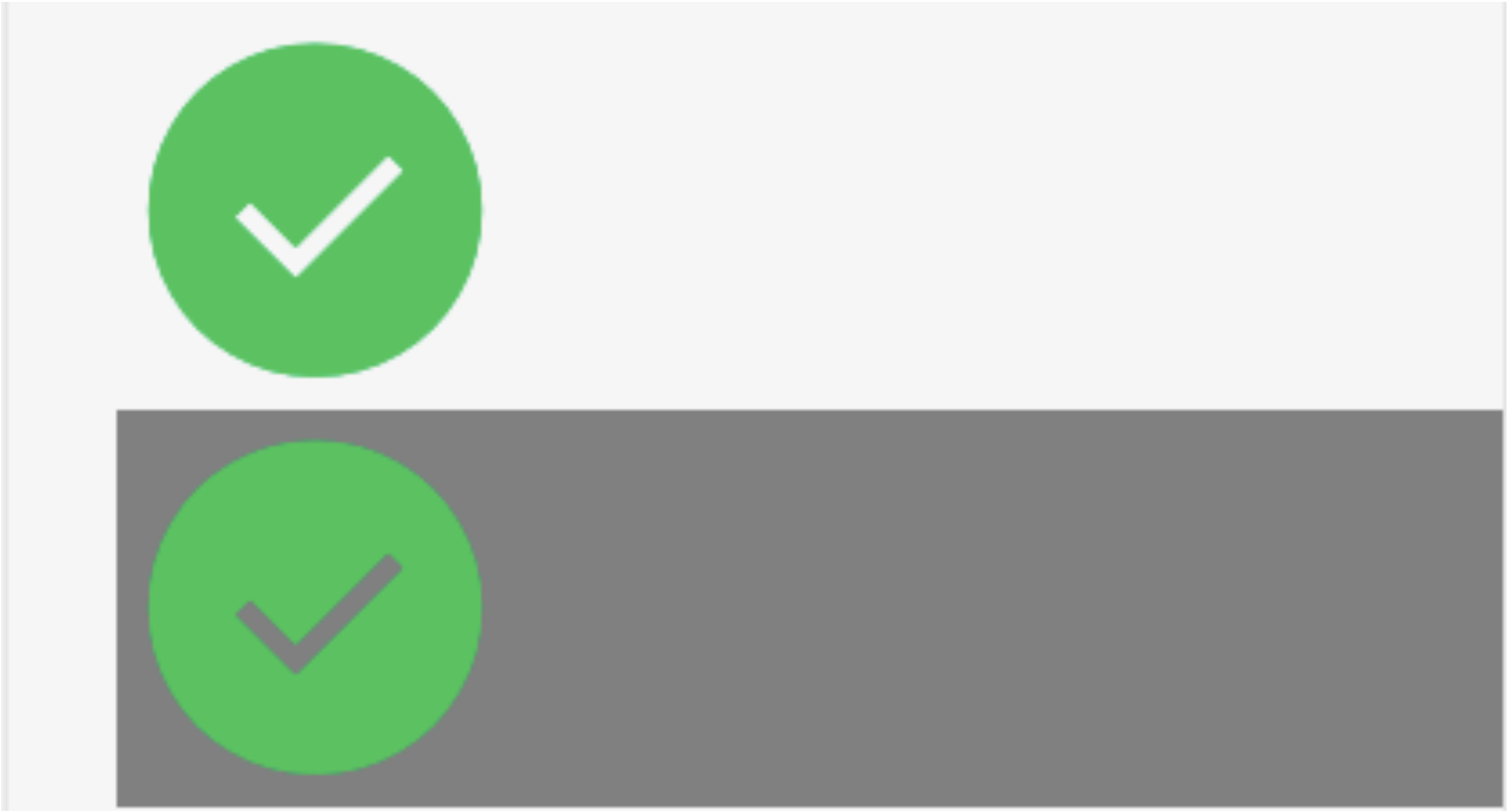
2.1 icon 组件介绍及如何自定义实现图标?

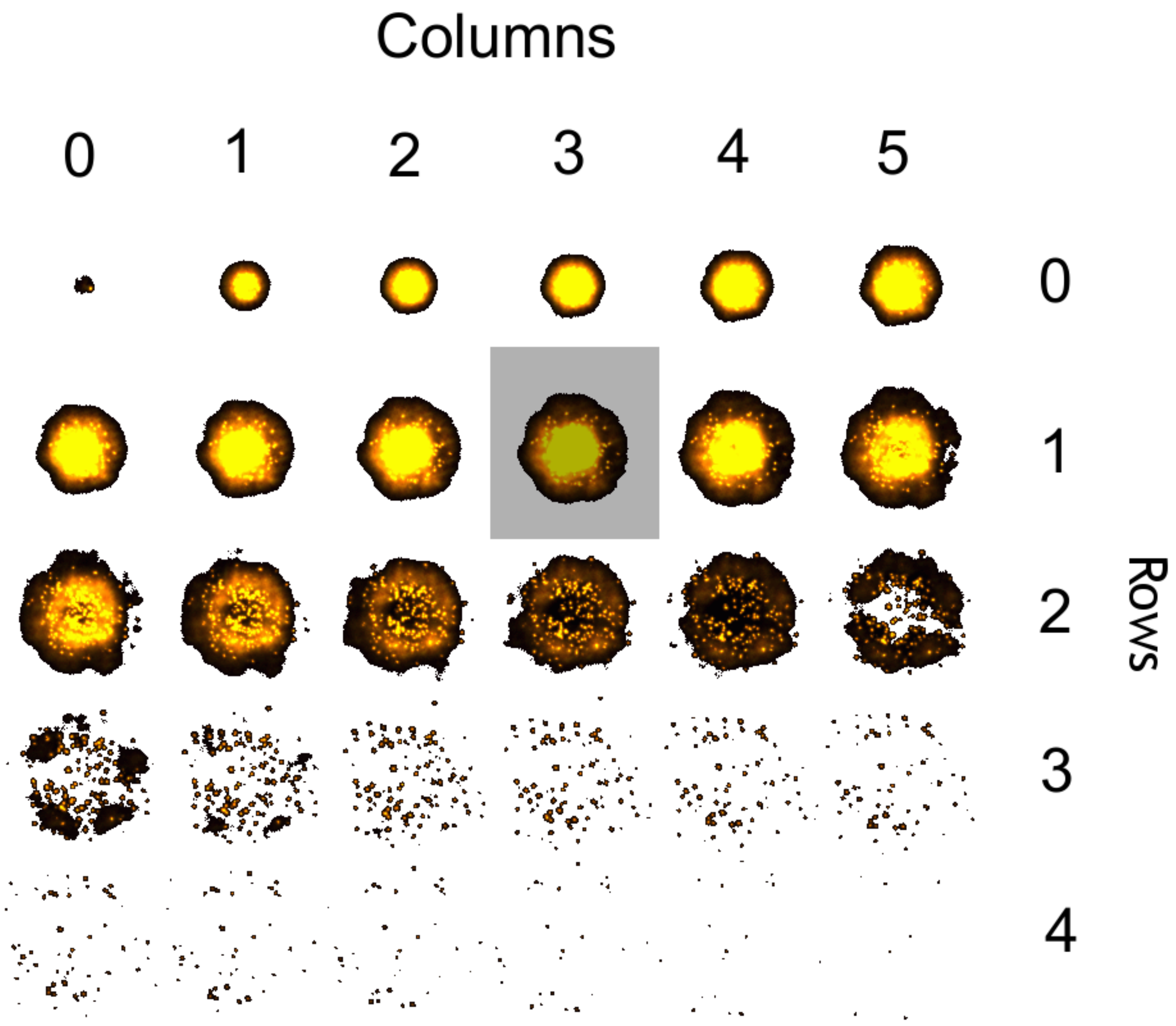
```
//developers.weixin.qq.com/miniprogram/dev/component/icon.html

<icon type="success" size="30px" color="green" />

type类型范围： 'success', 'success_no_circle', 'info', 'warn',
'waiting', 'cancel', 'download', 'search', 'clear'
```

```
<icon type="success" size="100rpx"/>
<icon style="background:grey;" type="success" size="100rpx"/>
```

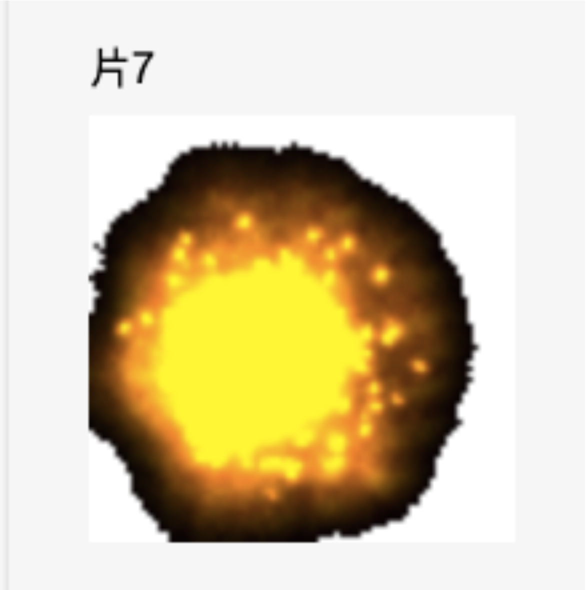


图标能否与文本同行，放在段落中？

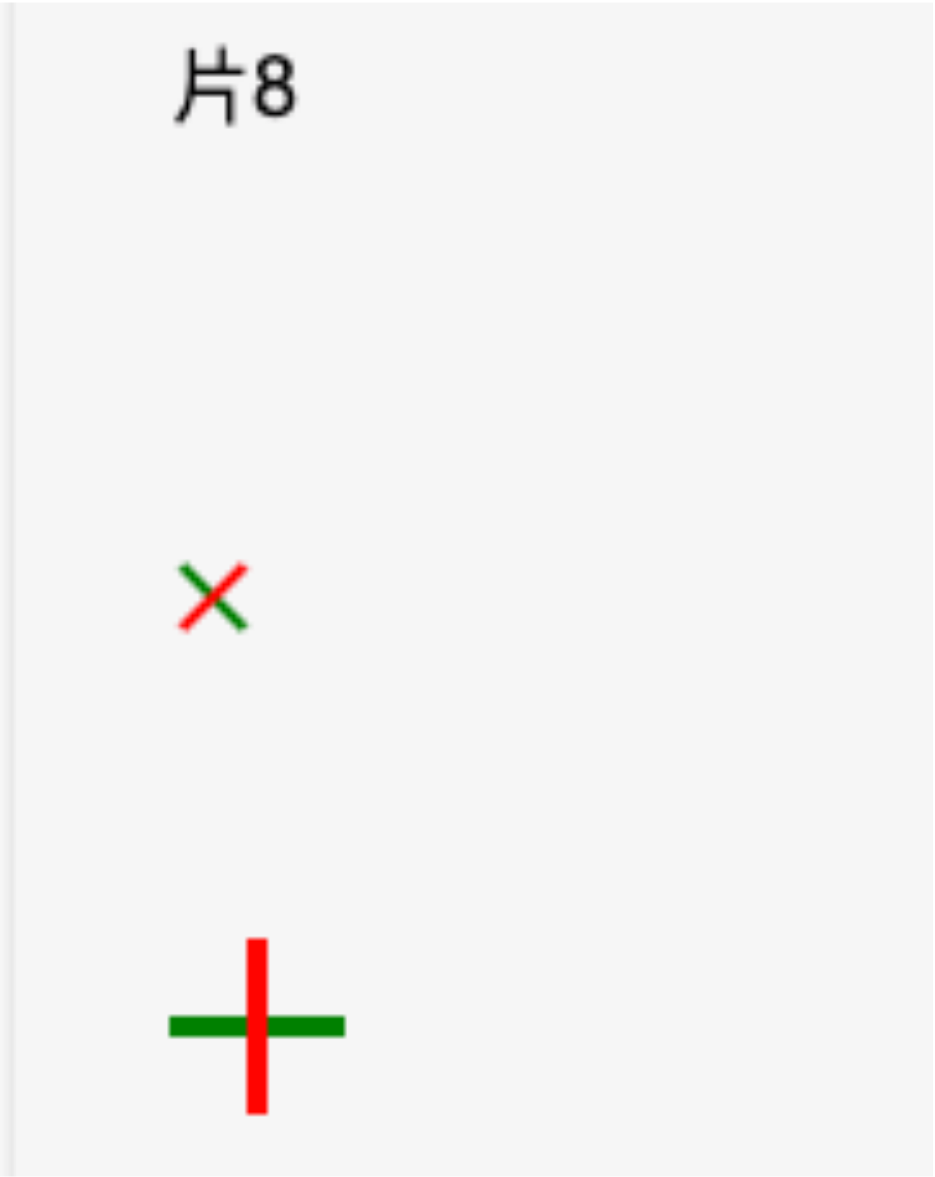
实现 icon 图标的方案有哪些，原理是什么？


Columns
0
1
2
3
4
5
Rows
0
1
2
3
4

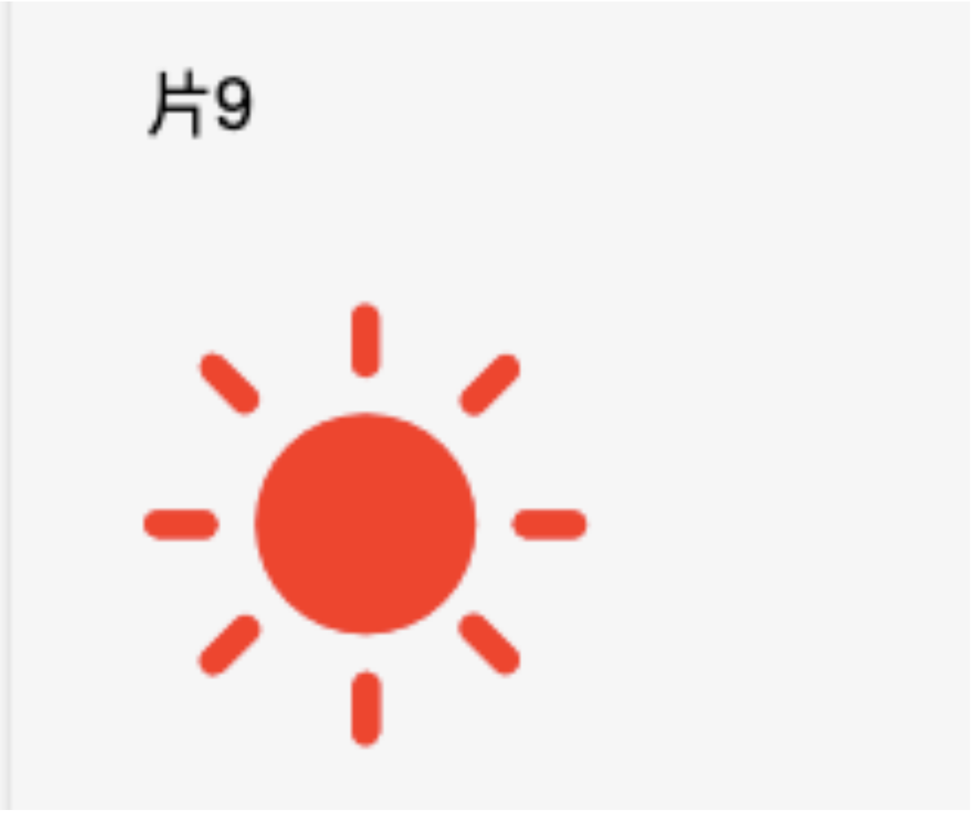
```
.sprite_icon {
  display: block;
  width: 80px;
  height: 80px;
  /* 此处在wxss中，可以使用网络图片，不能使用本地图片 */
  background: url("https://cdn.nlark.com/..1bd0.png") -180px -310px;
}
```



```css
/* 使用css3绘制图标 */
.icon-close {
  display: inline-block;
  width: 17px;
  height: 2px;
  background: red;
  transform: rotate(45deg);
}
.icon-close::after {
  content: '';
  display: block;
  width: 17px;
  height: 2px;
  background: red;
  transform: rotate(-90deg);
}
```

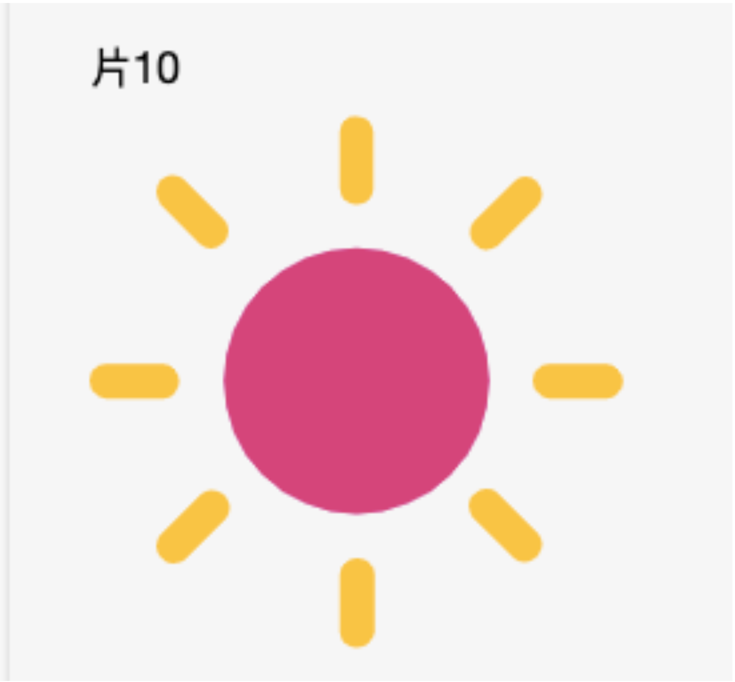


```
@font-face {
  font-family: 'iconfont';
  src: url('//at.alicdn.com/t/font_1716930_3m30jvz589y.eot');
  src: url('//at.alicdn.com/t/font_1716930_3m30jvz589y.eot?#iefix')
    format('embedded-opentype'),
  url('//at.alicdn.com/t/font_1716930_3m30jvz589y.woff2') format('woff2'),
  url('//at.alicdn.com/t/font_1716930_3m30jvz589y.woff') format('woff'),
  url('//at.alicdn.com/t/font_1716930_3m30jvz589y.ttf') format('truetype'),
url('//at.alicdn.com/t/font_1716930_3m30jvz589y.svg#iconfont') format('svg');
}
.iconfont {
  font-family: "iconfont" !important;
  font-size: 16px;
  font-style: normal;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}
.icon-sun:before {
  content: "\e603";
  color: red;
  font-size: 20px;
}
<icon class="iconfont icon-sun"></icon>
```



www.sojson.com/image2base64.html

```
.svg-icon{
  display: block;
  width: 200px;
  height: 200px;
  background-repeat: no-repeat;
  background: url("data:image/
svg+xml;base64,PHN...Zz4=");
}

<icon class="svg-icon"></icon>
```



https://github.com/Tencent/omi/tree/master/packages/cax/cax

https://developers.weixin.qq.com/community/develop/article/doc/000ca493bc09c0d03a8827b9b5b013

有时候真机上显示 icon 空白，不正常显示的问题。

weui 组件库里的 icon 组件的图标
如何取出来，保存到本地？

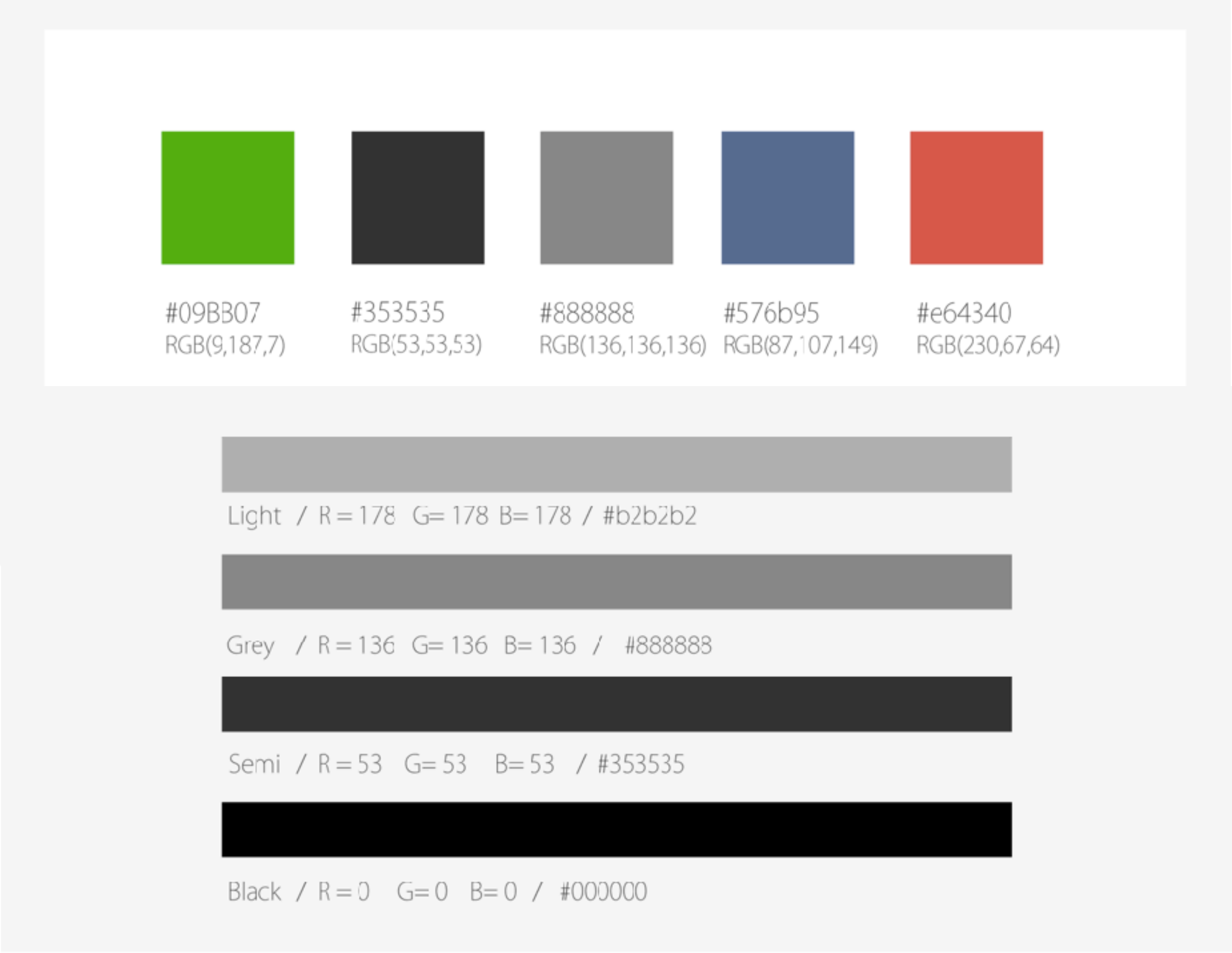
https://developers.weixin.qq.com/miniprogram/design/#设计

源码：https://git.weixin.qq.com/rxyk/weapp-practice/repository/archive.zip?ref=2.1-icon-514

2.2 progress 组件简介：如何实现一个环形进度条？

```
<!-- 2 代码示例-->
<view class="gap">代码示例,单击模拟网络异步</view>
<progress show-info bindtap="onTapProgressBar"
stroke-width="2" percent="{{percentValue}}"
backgroundColor="#f2f2f2" active-mode="forwards"
active bindactiveend="onProgressActiveEnd"/>
```

- https://developers.weixin.qq.com/miniprogram/design/# 字体



#09B307
RGB(9,187,7)

#353535
RGB(53,53,53)

#888888
RGB(136,136,136)

#576b95
RGB(87,107,149)

#e64340
RGB(230,67,64)

Light / R = 178 G= 178 B= 178 / #b2b2b2

Grey / R = 136 G= 136 B= 136 / #888888

Semi / R = 53 G= 53 B= 53 / #353535

Black / R = 0 G= 0 B= 0 / #000000

如何实现一个下载文件并显示动态进度条的功能?

progress 已产生的进度条如何设置圆角？

```
<progress border-radius="5" percent="20" show-info />
```

本地组件样式：~/Library/Application\ Support/微信开发者工具/WeappCode/package.nw/js/vendor/dev/wx-components.css

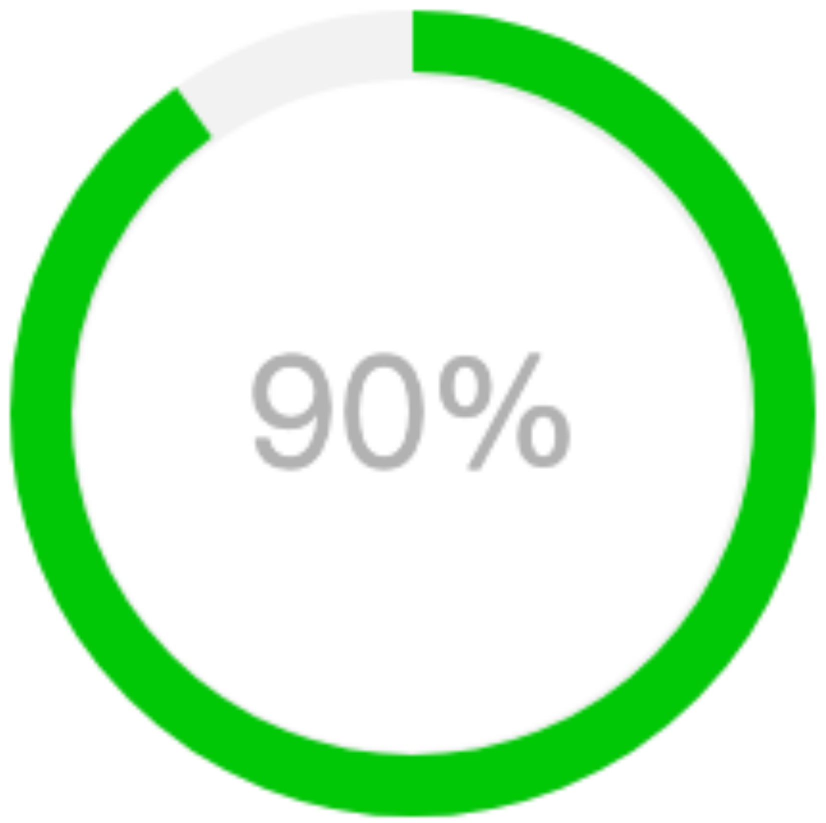
```
.wx-progress-inner-bar {
  width: 0;
  height: 100%;
 }
 .wx-progress-inner-bar {
  border-radius: 5px;
 }
```

已经加载完的进度条 progress，
如何点击某个按钮让它重新加载呢？

```
this.setData({ percentValue: 0 });
if (wx.canIUse('nextTick')) {
  wx.nextTick(() => {
    this.setData({ percentValue: 100 });
  });
} else {
  setTimeout(() => {
    this.setData({ percentValue: 100 });
  }, 17)
}

onTapReloadBtn(e){
  this.setData({percentValue:0})
  this.setData({percentValue:50})
}
```

能否实现一个圆环形进度条呢？

```
<view class='canvasBox'>
  <view class='bigCircle'></view>
  <view class='littleCircle'></view>
  <canvas canvas-id="runCanvas" id="runCanvas"
class='canvas'></canvas>
</view>

properties: {
  percent: {
    type: Number,
    value: 50,
    observer: function (newVal, oldVal) {
      this.draw(newVal);
    }
  },
}

var num = (2 * Math.PI / 100 * c) - 0.5 * Math.PI;//c是进度值percent
that.ctx2.arc(w, h, w - 8, -0.5 * Math.PI, num)
```



```
<!-- 环形进度条 -->
<circle-progress id="progress1" percent="{{percentValue}}" />

drawProgress(){
 if (this.data.percentValue >= 100){
  this.setData({
   percentValue:0
  })
 }
 this.setData({
  percentValue:this.data.percentValue+10
 })
}

const ctx2 = wx.createCanvasContext(canvasId, this)
const query = wx.createSelectorQuery().in(this)
query.select('#'+id).boundingClientRect((res)=>{
 ...
}).exec()
```

progress 右边进度的百分比数字，
它的颜色怎么设置？

```
<progress percent="40" stroke-width="5" show-info
style="color:red"/>
.wx-progress-info {
  color: red;
}
```

progress 组件右侧的百分比文字，
与左边离得太近了，可否增加一个边距？

```
.wx-progress-info {
  color: red;
  margin-left: 5px;
}
```

源码：https://git.weixin.qq.com/rxyk/weapp-practice/repository/archive.zip?ref=2.2-progress-515

实践：从 iconfront.cn 搜索两个图标，以自定义的方式用在自己的小程序项目中

2.3 富文本组件 rich-text 简介：如何单击预览节点图片并保存？

```
<rich-text space="emsp" nodes="{{nodes}}" bindtap=“tap"></rich-text>

nodes: [{
  name: 'div',
  attrs: {
    class: 'div_class',
    style: 'line-height: 20px;padding:20px;'
  },
  children: [
    {
      type: 'text',
      text: '小程序实践'
    },{
      name:'img',
      attrs:{
        src:'https://www.yishulun.com/favicon.ico',
        style:'width:100px'
      }
    },{
      name:'img',
      attrs:{
        src:'https://www.yishulun.com/image/篆刻-茹意.png',
        style:'width:100%'
      }
    },{
      name:'img',
      attrs:{
        src:'https://www.yishulun.com/image/篆刻-茹意.png',
        style:'width:100%'
      }
    }
  ]
}],
{
  type: 'text',
  text: 'message'
}
{
  name:'img',
  attrs:{
    src:'https://www.yishulun.com/favicon.ico',
    style:'width:100%'
  }
}
```

```
<img src="https://www.yishulun.com/favicon.ico" style=‘width:30px'/>
```

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

如何预览、保存 rich-text 富文本组件中的图片?

```
// 取出 urls
function findUrl(nodes){
  let urls = []
  nodes.forEach(item=>{
    if (item.attrs){
      for (const key in item.attrs) {
        if (key == 'src') {
          urls.push(item.attrs[key])
        }
      }
    }
    if (item.children){
      urls = urls.concat( findUrl(item.children) )
    }
  })
  return urls
}
this.data.urls = findUrl(this.data.nodes)

tap(e) {
  let urls = this.data.urls
  wx.previewImage({
    current: urls[0],
    urls: urls
  })
}
```

在富文本 rich-text 中
如何解决图片之间的间距问题？

```
{
  name:'img',
  attrs:{
    src:'https://www.yishulun.com/images/篆刻-茹意.png',
    style:'width:100%;font-size:0;display:block;'//修改样式
  }
}
.img{
  font-size:0;
  display:block;
}
{
  name:'img',
  attrs:{
    src:'https://www.yishulun.com/images/篆刻-茹意.png',
    class: 'img'
  }
}
```

在富文本 rich-text 里面怎么插入 ad 广告标签?
如何将 HTML 文本直接解析呈现?

```
https://github.com/jin-yufeng/Parser

{
  "usingComponents": {
    "parser":"../parser/parser"
  }
}

tagStyle:{
  img: 'font-size:0;display:block;',//样式
},
html:"<div>小程序实践<span>message</span><img src='https://www.yishulun.com/
image/篆刻-茹意.png' /><img src='https://www.yishulun.com/image/篆刻-茹意.png' /
></div>"

<parser html="{{html}}" tag-style="{{tagStyle}}" />
```

```
miniprogram/pages/2.1/parser/libs/MpHtmlParser.js:
Comment() {
  var key;
  if (this.data.substring(this.i + 2, this.i + 4) == '--') key = '-->';
  else if (this.data.substring(this.i + 2, this.i + 9) == '[CDATA[') key = ']]>';
  ...
}

// 处理属性
matchAttr(node) {
  ...
  switch (node.name) {
    case 'a':
    case 'ad':
      this.bubble();
      break;
    case 'font':
    ...
```

```
miniprogram/pages/2.1/parser/trees/trees.wxml:
<!--trees 递归子组件-->
<wxs module="handler" src="./handler.wxs" />
<block wx:for="{{nodes}}" wx:key="index" wx:for-item="n">
  <rich-text wx:if="{{n.en||n.svg||n.err}}" class="_svg" nodes="{{[n]}}" />
<!--图片-->
<image wx:elif="{{n.name=='img'}}" class="_img" ...
  <!--文本-->
  <text wx:elif="{{n.type=='text'}}" decode>{{n.text}}</text>
  <text wx:elif="{{n.name=='br'}}">\n</text>
  <!--链接-->
  <view wx:elif="{{n.name=='a'}}" ...
  </view>
  <!--视频-->
  <block wx:elif="{{n.name=='video'}}">
  ...
  <!--广告-->
  <ad wx:elif="{{n.name=='ad'}}" unit-id="{{n.attrs['unit-id']}}".../>
  ...
```

miniprogram/pages/2.1/parser/trees/trees.wxml:

```
<!--图片-->
<image wx:elif="{{n.name=='img'}}" ... bindtap="imgtap" bindload="{{canIUse?
handler.load:'loadImg'}}" binderror="error" />
```

miniprogram/pages/2.1/parser/trees/trees.js:

```
// 图片点击事件
imgtap(e) {
  ...
    this.top.triggerEvent('imgtap', {
      id: e.target.id,
      src: attrs.src,
      ignore: () => preview = false
    })
    if (preview) {
      ...
      wx.previewImage({
        current,
        urls
      })
    }
  }
}
```

```
<parser bindimgtap="onTapImage" html="{{html}}" tag-
style="{{tagStyle}}" />

onTapImage(e){
  console.log('iamge url', e.detail.src)
}
output:
iamge url https://www.yishulun.com/image/篆刻-茹意.png
iamge url https://www.yishulun.com/favicon.i

github.com/icindy/wxParse
```

本课源码：https://git.weixin.qq.com/rxyk/weapp-practice
/repository/archive.zip?ref=2.3-richtext-515-2

2.4 view 及 Flex 布局简介：
如何使用 view 实现常见的UI布局？（一）

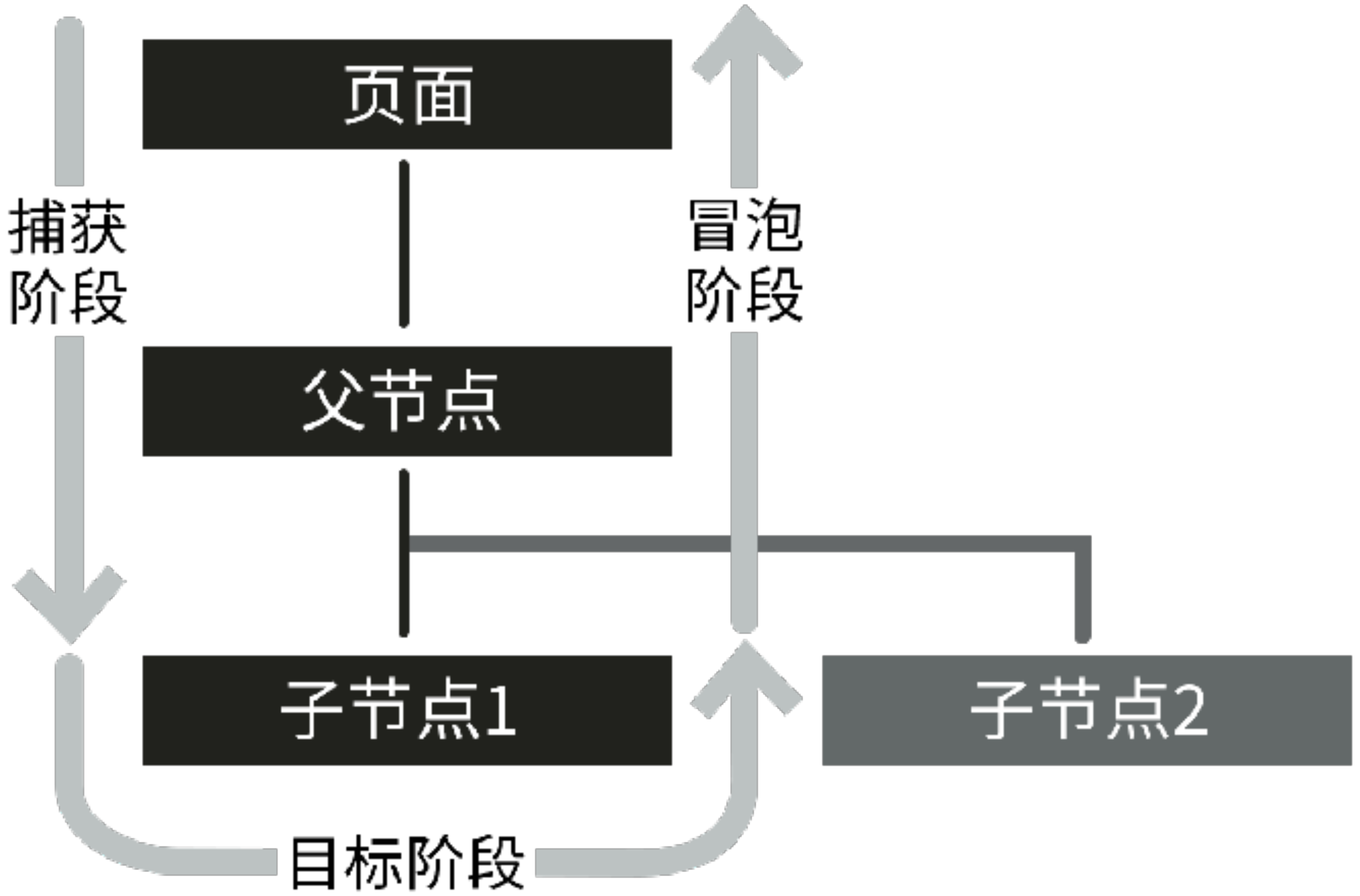
hover-class

```
<view hover-class="bc_red" class="section__title">content</view>
```

hover-stop-propagation

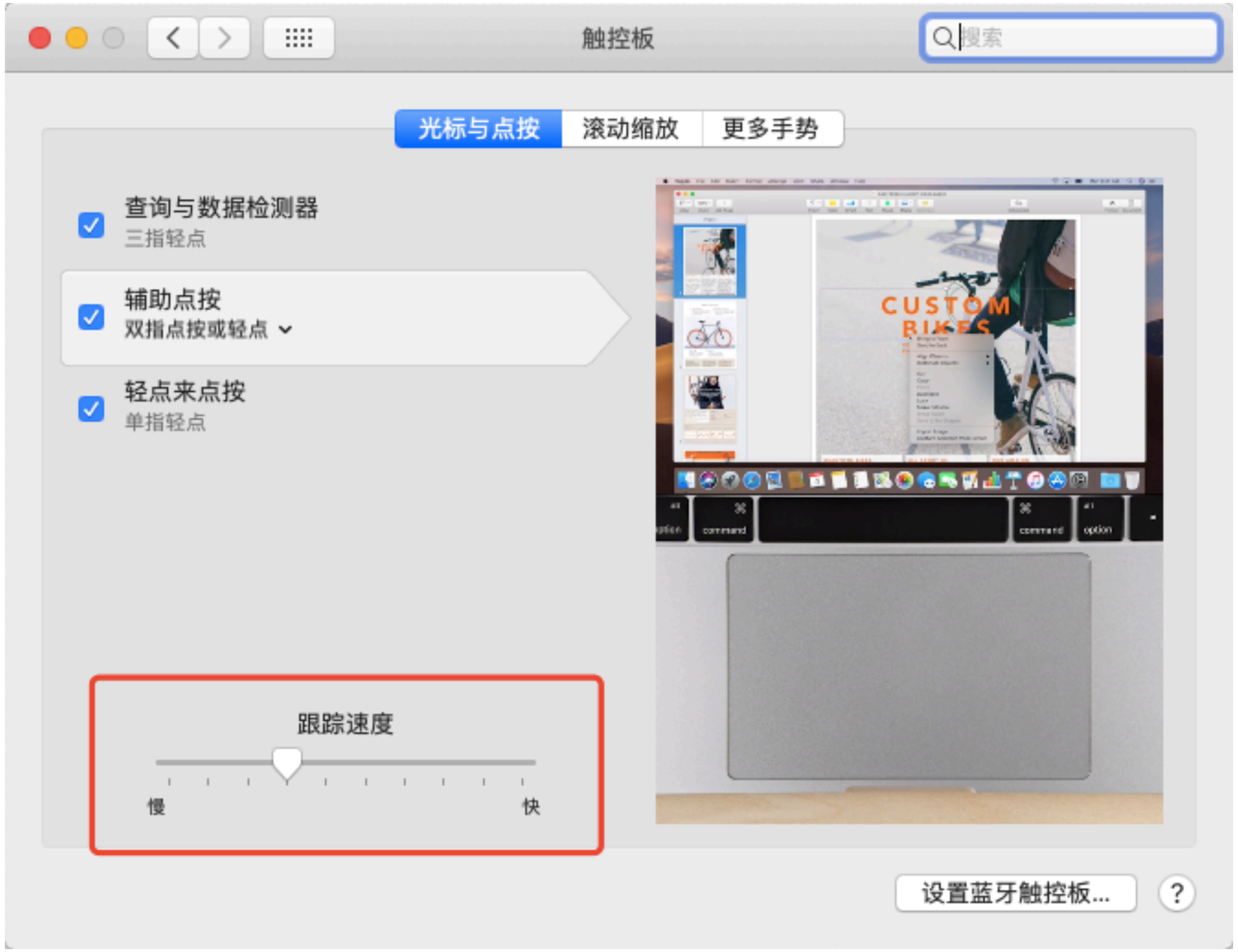
```
<!-- 阻止父节点出现 hover 状态 -->
<view hover-class="bc_red" class="section__title">
parent
<view hover-stop-propagation hover-class="bc_green" class="section__title">
child view
</view>
</view>
```

```
<!-- 阻止父节点出现 hover 状态 -->
<view id="parentView" bindtap="onTap" hover-class="bc_red" class="section__title">
parent
<view id="childView" bindtap="onTap" hover-stop-propagation hover-class="bc_green" class="section__title">
child view
</view>
</view>
onTap(e){
console.log(e.target)
}
output:
{id: "parentView", offsetLeft: 20, offsetTop: 460, dataset: {…}}
{id: "childView", offsetLeft: 20, offsetTop: 485, dataset: {…}}
{id: "childView", offsetLeft: 20, offsetTop: 485, dataset: {…}}
```



```
<!-- 阻止父节点出现 hover 状态，阻止冒泡 -->
<view id="parentView" bindtap="onTap" hover-class="bc_red" class="section__title">
parent
<view id="childView" catchtap="onTap" hover-stop-propagation hover-class="bc_green" class="section__title">
child view
</view>
</view>
```

hover-start-time、hover-stay-time



触控板

光标与点按 | 滚动缩放 | 更多手势

☑ 查询与数据检测器
三指轻点

☑ 辅助点按
双指点按或轻点

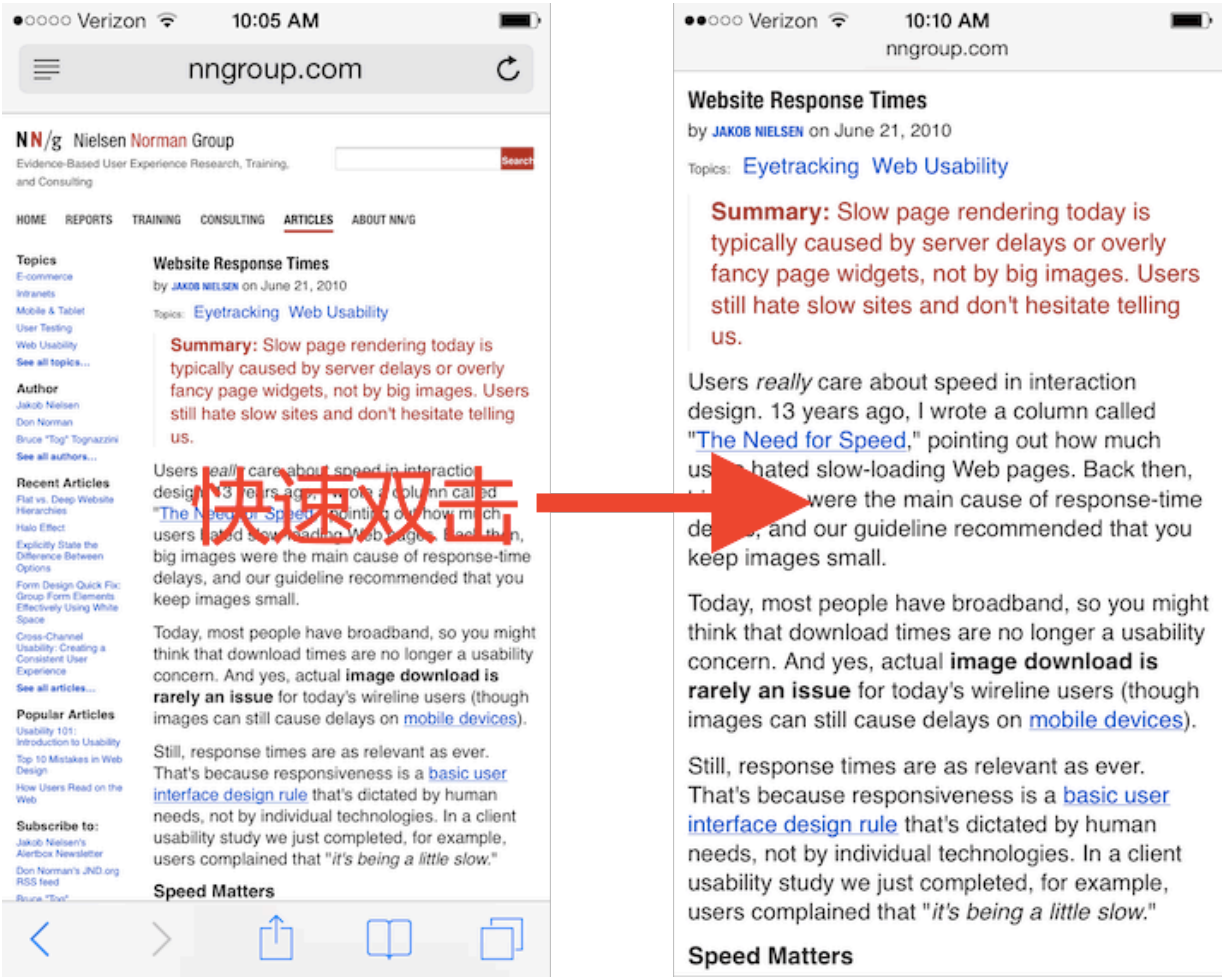
☑ 轻点来点按
单指轻点

跟踪速度

慢 快

设置蓝牙触控板...

拒绝 300 毫秒延迟



●○○○○ Verizon 10:05 AM
nngroup.com
NN/g Nielsen Norman Group
Evidence-Based User Experience Research, Training, and Consulting
Search
HOME REPORTS TRAINING CONSULTING ARTICLES ABOUT NN/G
快速双击
●●○○○ Verizon 10:10 AM
nngroup.com

Topics
E-commerce
Intranets
Mobile & Tablet
User Testing
Web Usability
See all topics...

Author
Jakob Nielsen
Don Norman
Bruce "Tog" Tognazzini
See all authors...

Recent Articles
Flat vs. Deep Website Hierarchies
Halo Effect
Explicitly State the Difference Between Options
Form Design Quick Fix: Group Form Elements Effectively Using White Space
Cross-Channel Usability: Creating a Consistent User Experience
See all articles...

Popular Articles
Usability 101: Introduction to Usability
Top 10 Mistakes in Web Design
How Users Read on the Web

Subscribe to:
Jakob Nielsen's Alertbox Newsletter
Don Norman's JND.org RSS feed
Bruce "Tog"

Website Response Times

by JAKOB NIELSEN on June 21, 2010

Topics: Eyetracking Web Usability

Summary: Slow page rendering today is typically caused by server delays or overly fancy page widgets, not by big images. Users still hate slow sites and don't hesitate telling us.

Users *really* care about speed in interaction design. 13 years ago, I wrote a column called "The Need for Speed," pointing out how much users hated slow-loading Web pages. Back then, big images were the main cause of response-time delays, and our guideline recommended that you keep images small.

Today, most people have broadband, so you might think that download times are no longer a usability concern. And yes, actual **image download is rarely an issue** for today's wireline users (though images can still cause delays on mobile devices).

Still, response times are as relevant as ever. That's because responsiveness is a basic user interface design rule that's dictated by human needs, not by individual technologies. In a client usability study we just completed, for example, users complained that "*it's being a little slow.*"

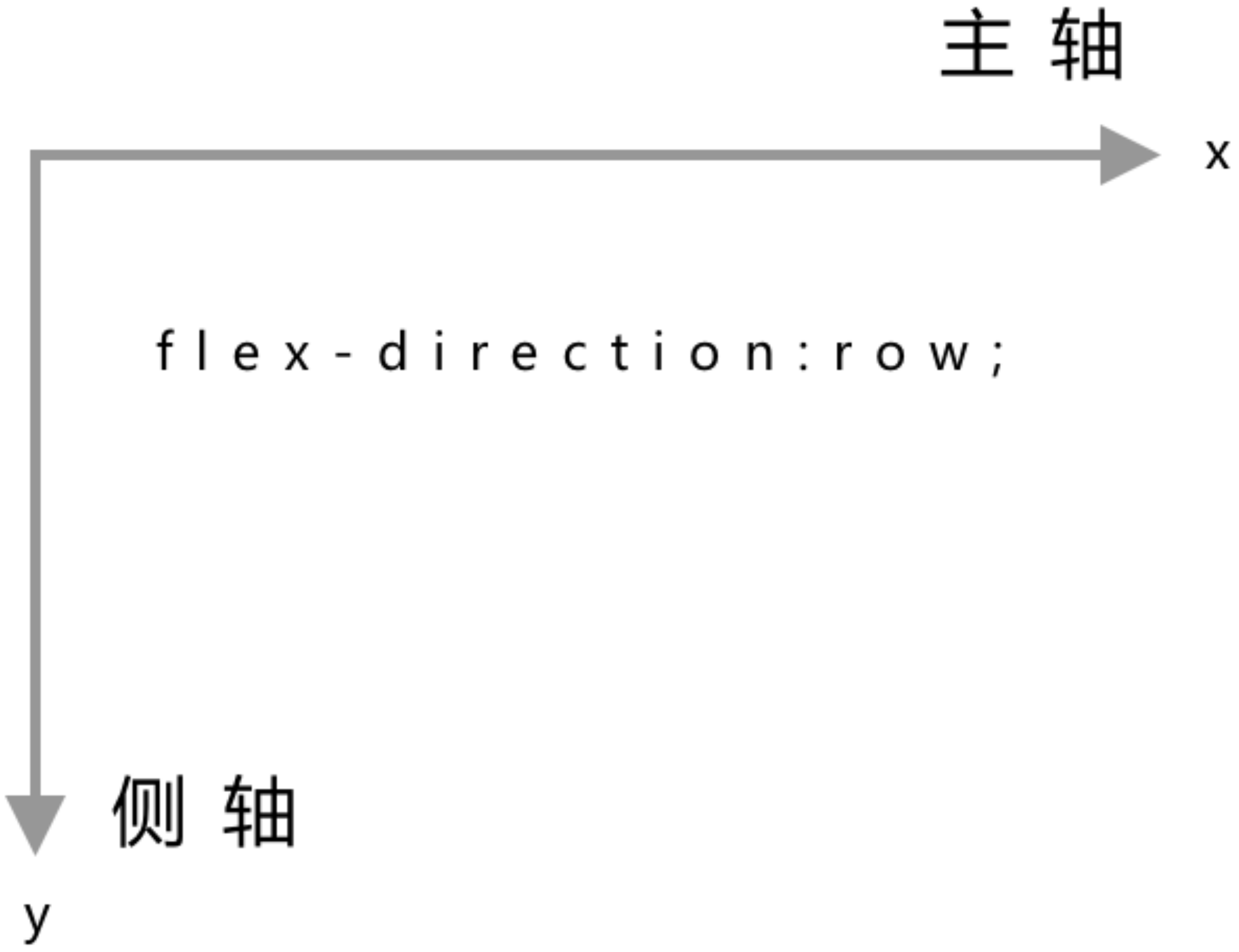
Speed Matters

使用 hover-class 定义按钮状态

```
<!-- 普通按钮 -->
<view class="section">
<button hover-class="rect-btn__hover_btn" type="primary">完成</button>
</view>
<!-- 圆形按钮 -->
<view class="section">
<button hover-class="circle-btn__hover_btn">
<icon type="success" size="80px"></icon>
</button>
</view>
<!-- 距形按钮 -->
<view class="section">
<button type="default" class="btn" plain hover-class="rect-btn__hover_btn">
<icon type="success_no_circle" size="26px"></icon>完成
</button>
</view>
```

```
.btn{
  display: flex;
  align-items: middle;
  padding: 8px 50px 8px;
  border: 1px solid #b2b2b2;
  background-color: #f2f2f2;
  width:auto;
}
/* 圆角按钮 */
.circle-btn__hover_btn {
  opacity: 0.8;
  transform: scale(0.95, 0.95);
}
/* 方框按钮 */
.rect-btn__hover_btn {
  position: relative;
  top: 3rpx;
  left: 3rpx;
  box-shadow: 0px 0px 8px rgba(175, 175, 175, .2) inset;
}
```

2.5 view 及 Flex 布局简介：如何使用 view 实现常见的UI布局？（二）



主轴

x

flex-direction:row;

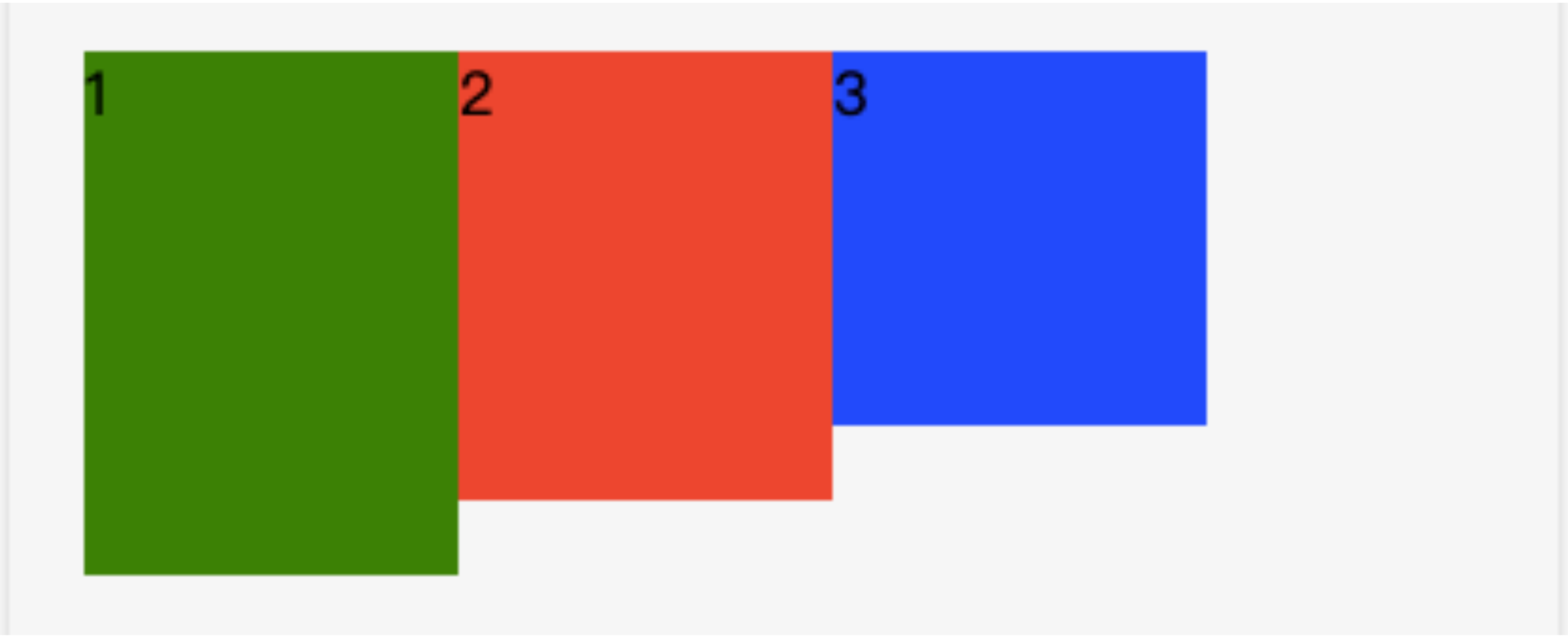
侧轴

y

justify-content

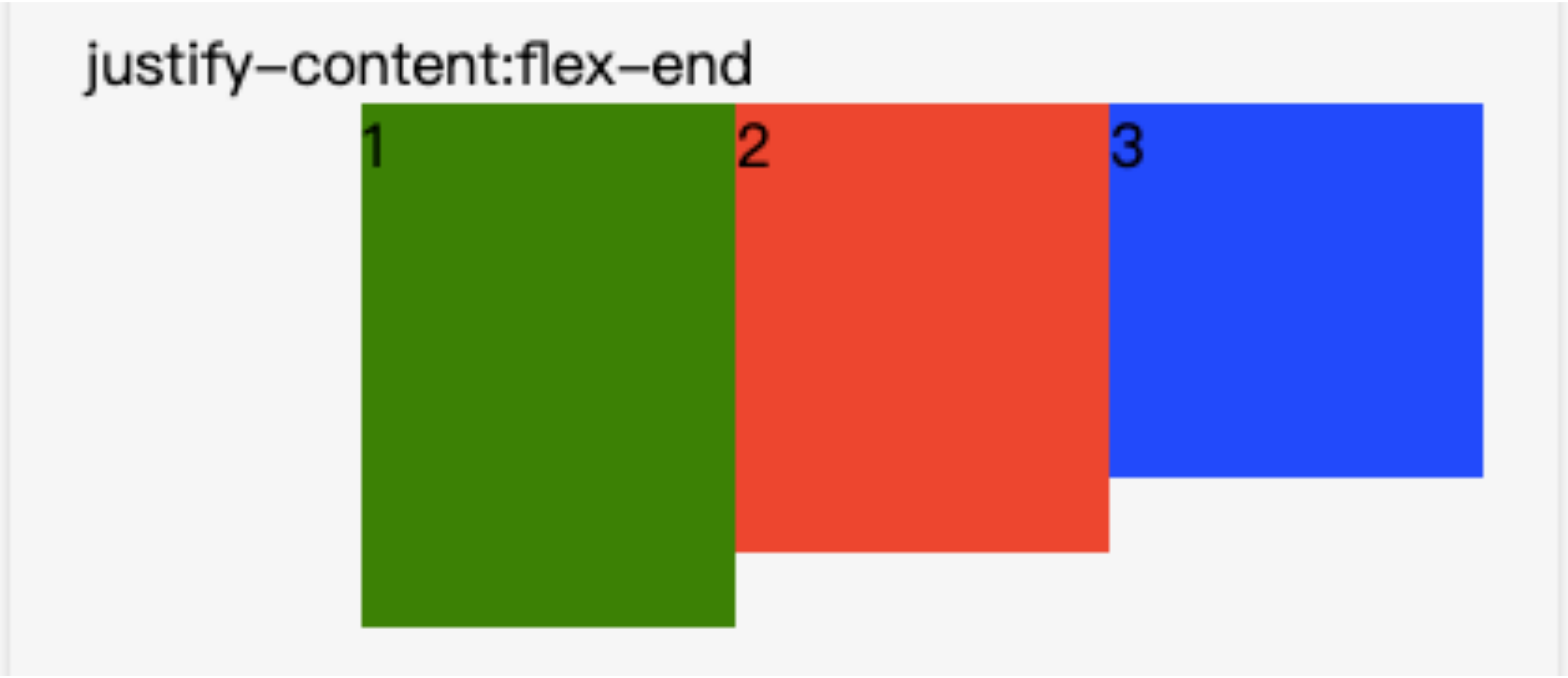
第一个值：flex-start

```
<view class="section">
<view class="section__title">4 justify-content:flex-start</view>
<view class="flex-wrp" style="flex-direction:row;;justify-content:flex-start">
<view class="flex-item bc_green">1</view>
<view class="flex-item bc_red">2</view>
<view class="flex-item bc_blue">3</view>
</view>
</view>
```



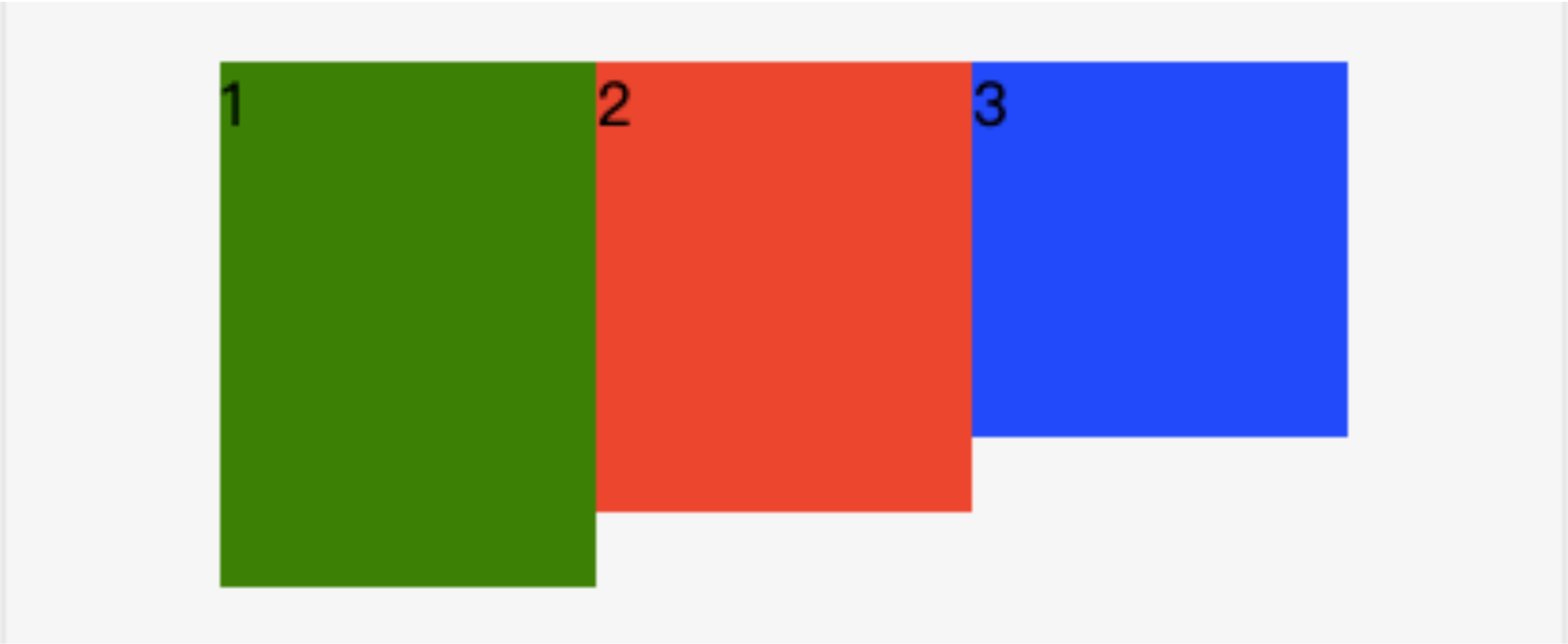
第二个值：flex-end

```
<view class="section">
<view class="section__title">5 justify-content:flex-end</view>
<view class="flex-wrp" style="flex-direction:row;justify-content:flex-end">
<view class="flex-item bc_green">1</view>
<view class="flex-item bc_red">2</view>
<view class="flex-item bc_blue">3</view>
</view>
</view>
```


justify-content:flex-end

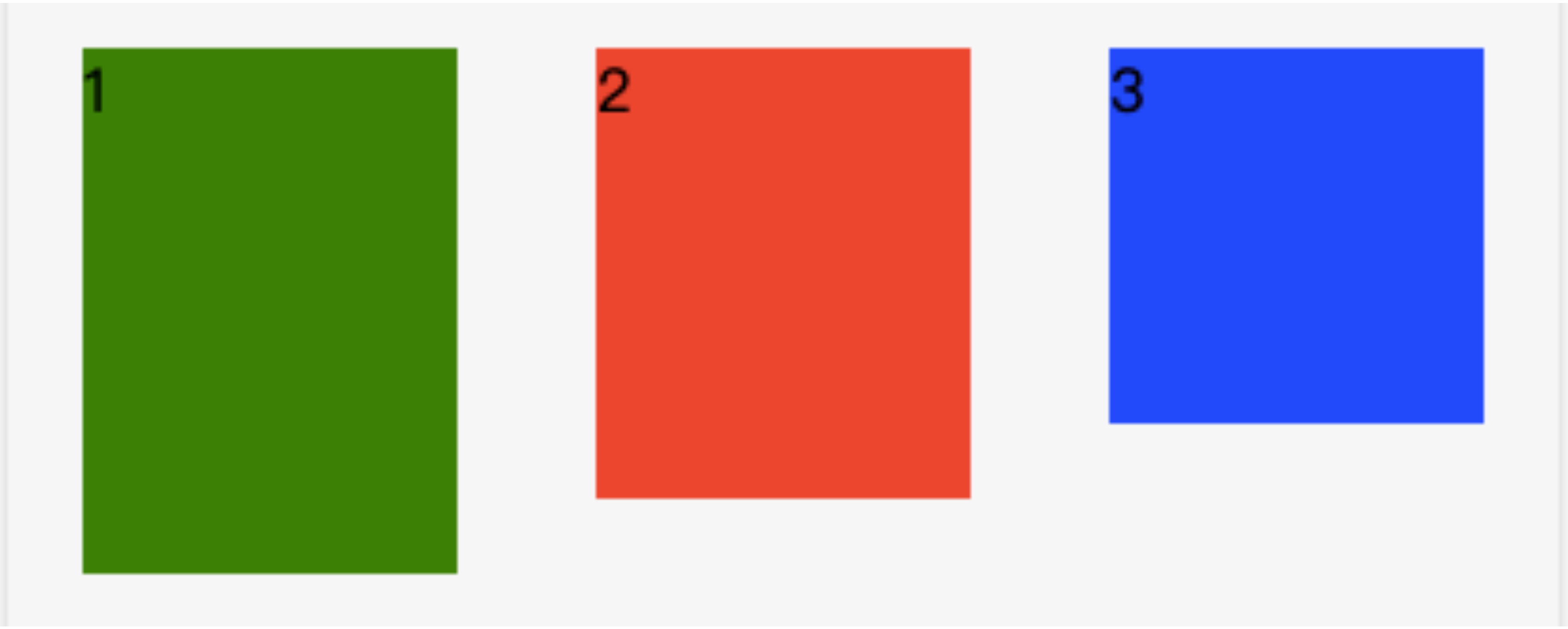
第三个值：center

```
<view class="section">
<view class="section__title">6 justify-content:center</view>
<view class="flex-wrp" style="flex-direction:row;justify-content:center">
<view class="flex-item bc_green">1</view>
<view class="flex-item bc_red">2</view>
<view class="flex-item bc_blue">3</view>
</view>
</view>
```



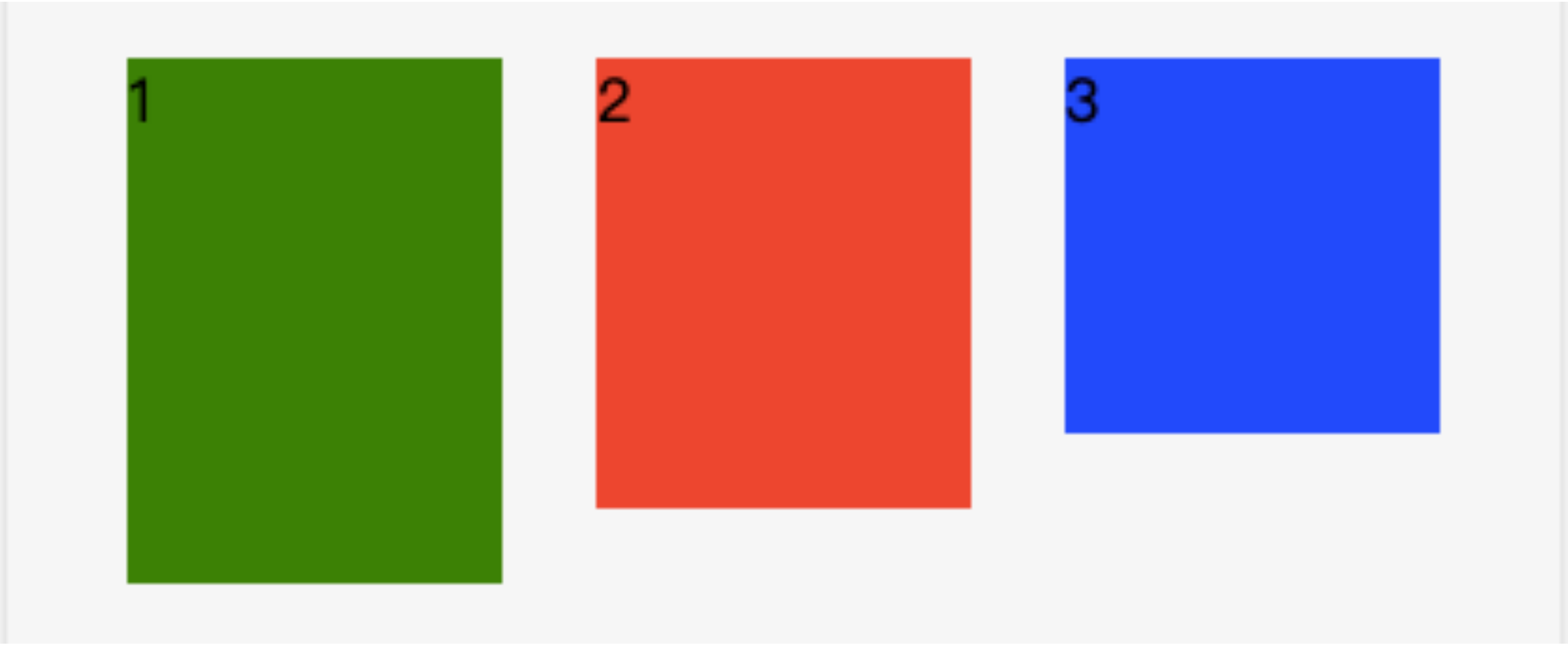
第四个值：space-between

```
<view class="section">
<view class="section__title">7 justify-content:space-between</view>
<view class="flex-wrp" style="flex-direction:row;justify-content:space-between">
<view class="flex-item bc_green">1</view>
<view class="flex-item bc_red">2</view>
<view class="flex-item bc_blue">3</view>
</view>
</view>
```



第五个值：space-around

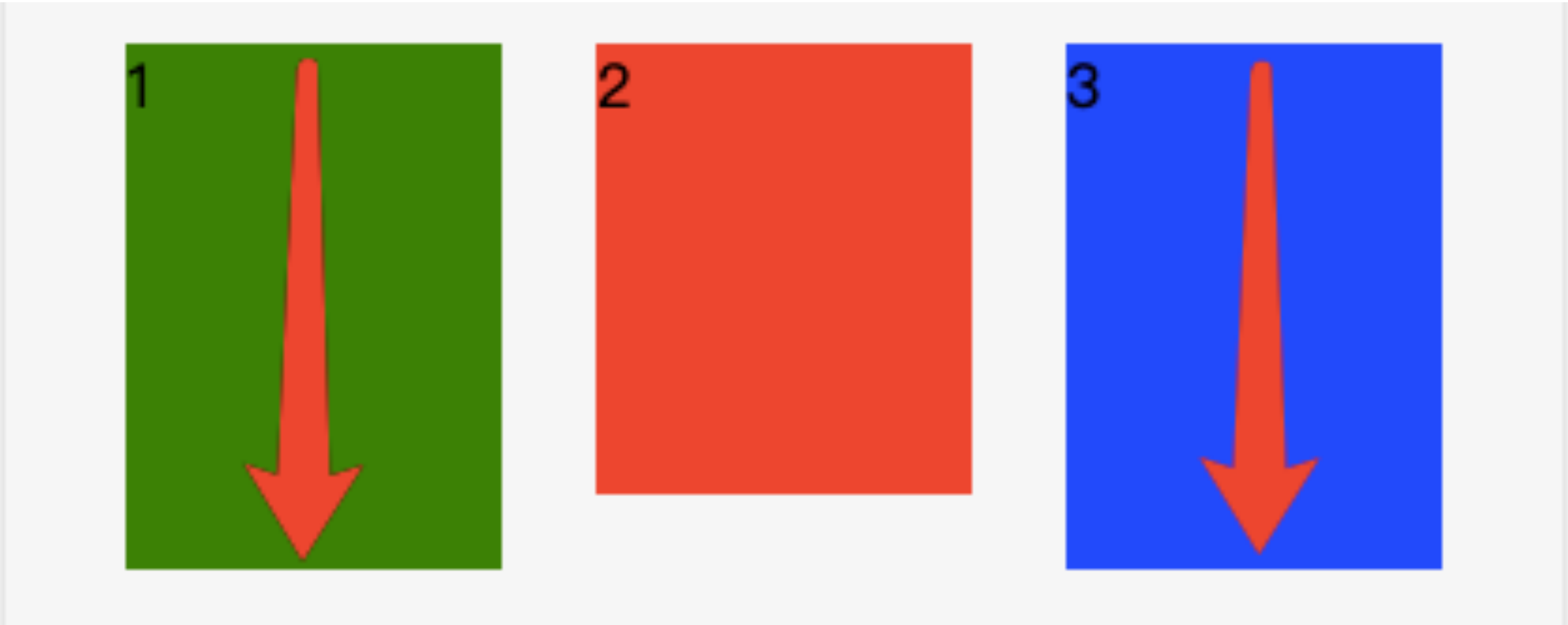
```
<view class="section">
<view class="section__title">8 justify-content:space-around</view>
<view class="flex-wrp" style="flex-direction:row;justify-content:space-around">
<view class="flex-item bc_green">1</view>
<view class="flex-item bc_red">2</view>
<view class="flex-item bc_blue">3</view>
</view>
</view>
```



align-items

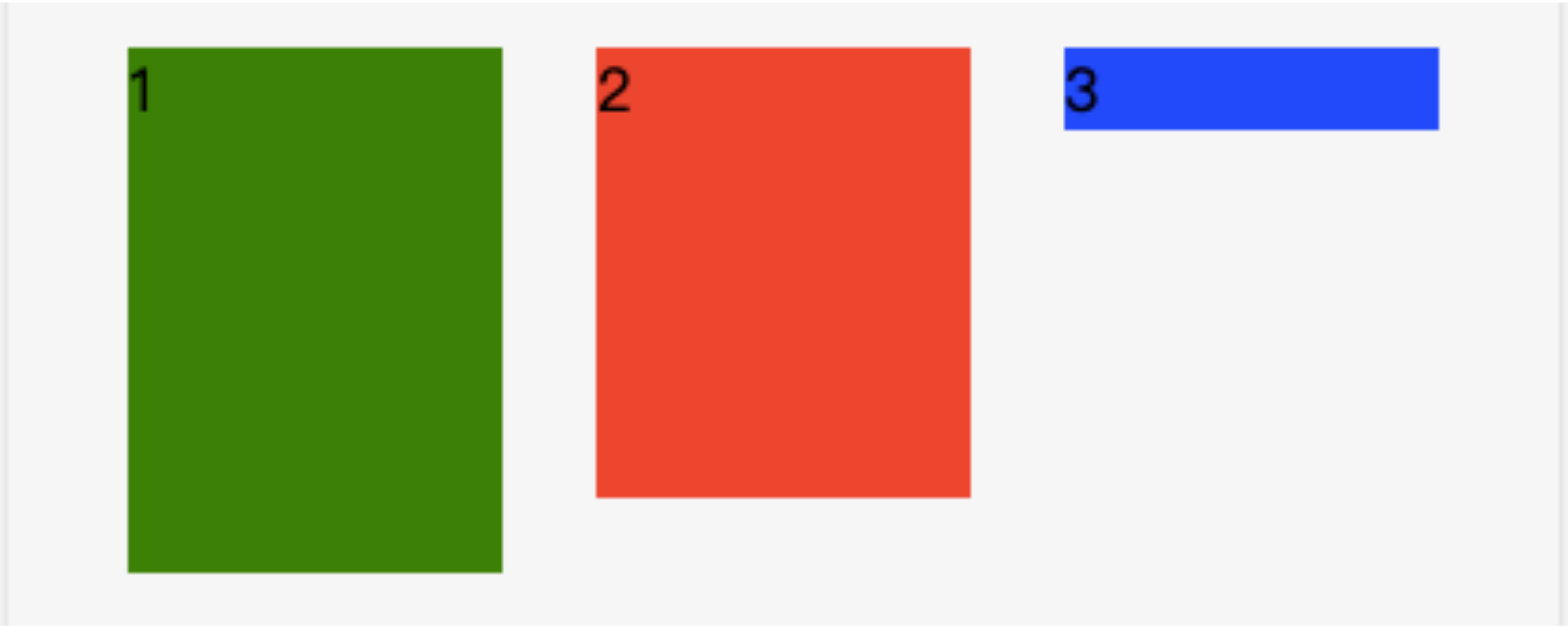
第一个值：stretch

```
<view class="section">
<view class="section__title">10 align-items:stretch</view>
<view class="flex-wrp" style="flex-direction:row;justify-content:space-around;align-items:stretch;">
<view class="flex-item bc_green">1</view>
<view class="flex-item bc_red">2</view>
<view style="height:auto;" class="flex-item bc_blue">3</view>
</view>
</view>
```



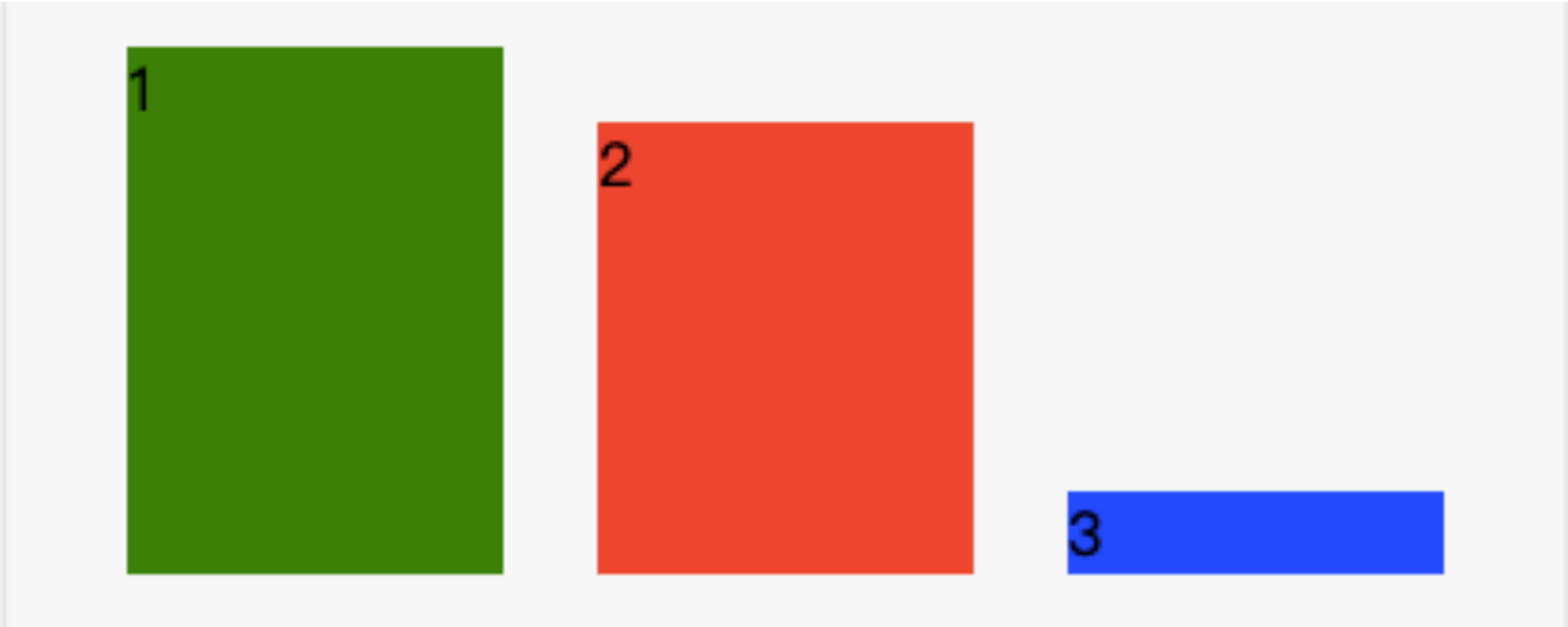
第二个值：flex-start

```
<view class="section">
<view class="section__title">11 align-items:flex-start</view>
<view class="flex-wrp" style="flex-direction:row;justify-content:space-around;align-items:flex-start;">
<view class="flex-item bc_green">1</view>
<view class="flex-item bc_red">2</view>
<view style="height:auto;" class="flex-item bc_blue">3</view>
</view>
</view>
```



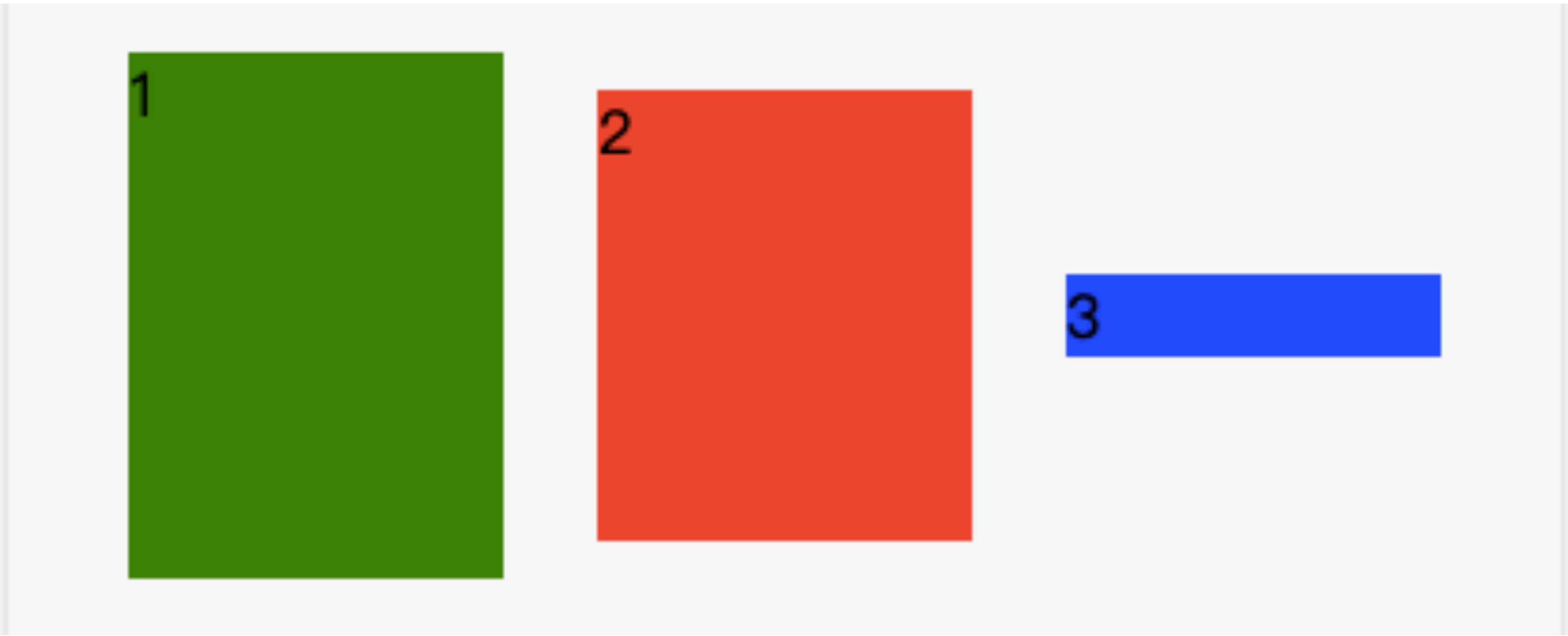
第三个值：flex-end

```
<view class="section">
<view class="section__title">12 align-items:end</view>
<view class="flex-wrp" style="flex-direction:row;justify-content:space-around;align-items:flex-end;">
<view class="flex-item bc_green">1</view>
<view class="flex-item bc_red">2</view>
<view style="height:auto;" class="flex-item bc_blue">3</view>
</view>
</view>
```



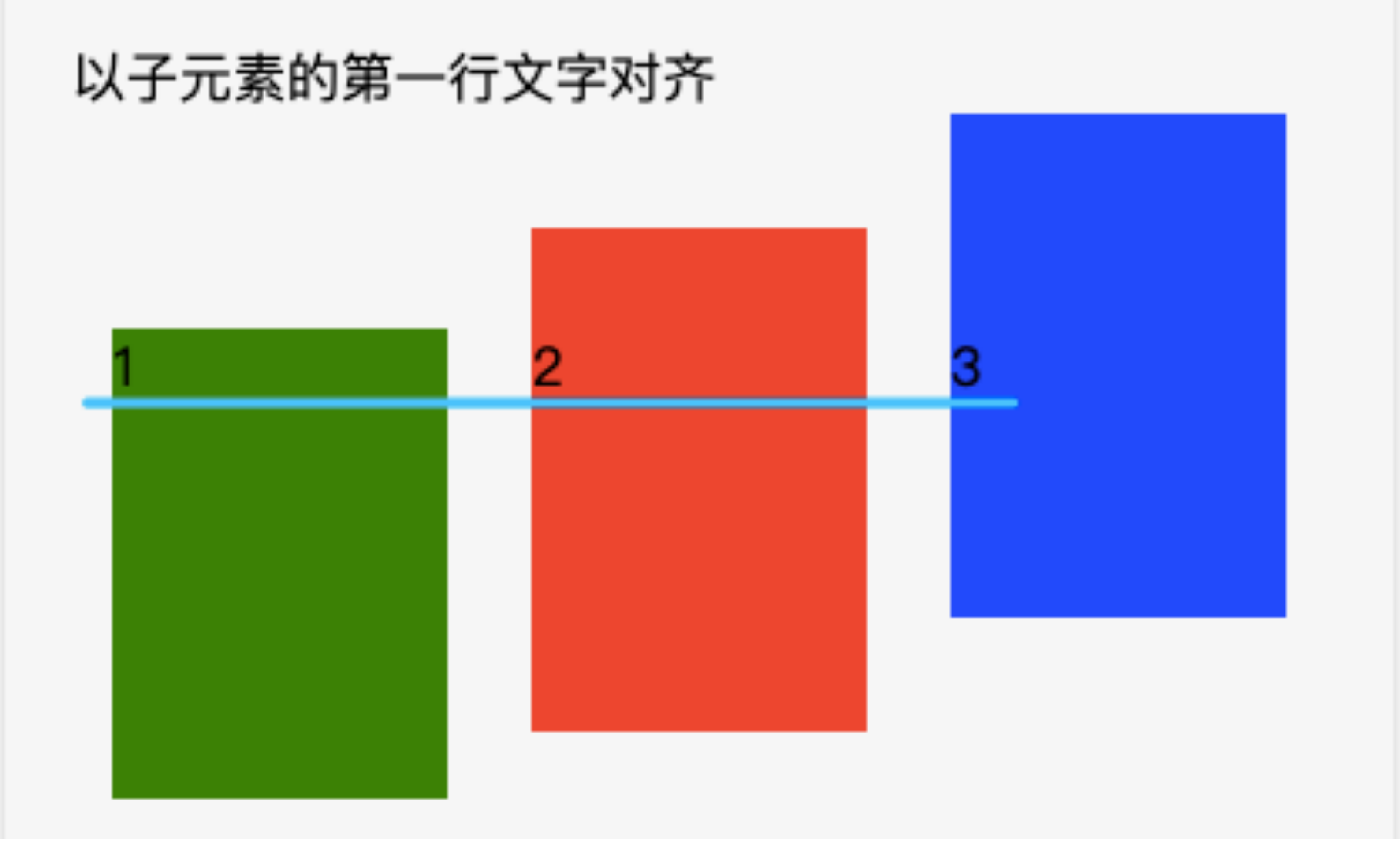
第四个值：center

```
<view class="section">
<view class="section__title">12 align-items:center</view>
<view class="flex-wrp" style="flex-direction:row;justify-content:space-around;align-items:center;">
<view class="flex-item bc_green">1</view>
<view class="flex-item bc_red">2</view>
<view style="height:auto;" class="flex-item bc_blue">3</view>
</view>
</view>
```



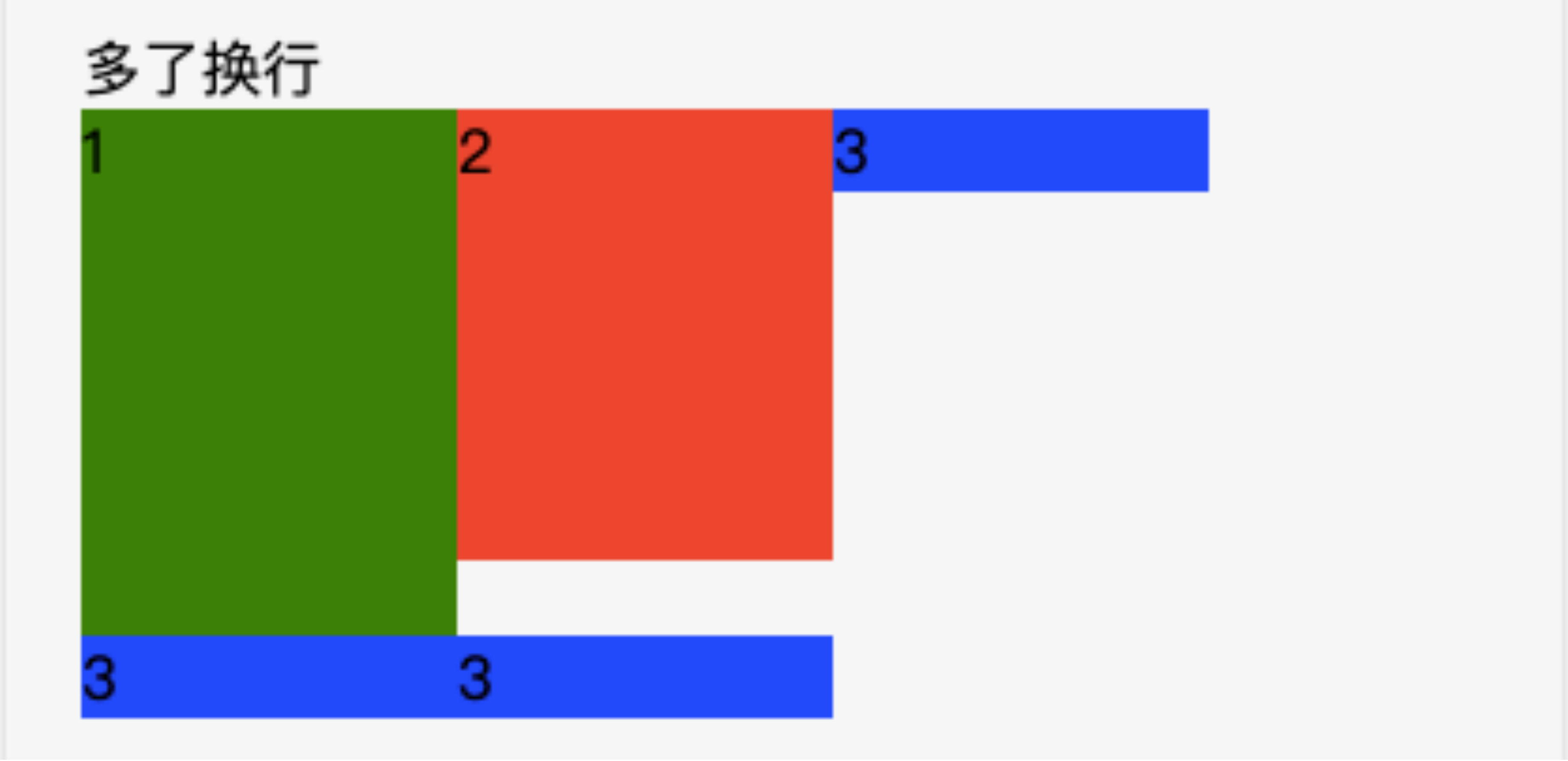
第五个值：baseline

```
<view class="section">
<view class="section__title">14 以子元素的第一行文字对齐</view>
<view class="flex-wrp" style="flex-direction:row;justify-content:space-around;align-
items:baseline;">
<view class="flex-item bc_green">1</view>
<view style="padding-top:30px;" class="flex-item bc_red">2</view>
<view style="height:auto;line-height:150px;" class="flex-item bc_blue"><text>3</text></
view>
</view>
</view>
```


以子元素的第一行文字对齐

flex-wrap 的值

```
<view class="section">
<view class="section__title">元素多了，换行</view>
<view class="flex-wrp" style="flex-direction:row;justify-content:flex-start;align-items:baseline;flex-wrap:wrap;">
<view class="flex-item bc_green">1</view>
<view class="flex-item bc_red">2</view>
<view style="height:auto;" class="flex-item bc_blue">3</view>
<view style="height:auto;" class="flex-item bc_blue">3</view>
<view style="height:auto;" class="flex-item bc_blue">3</view>
</view>
</view>
```

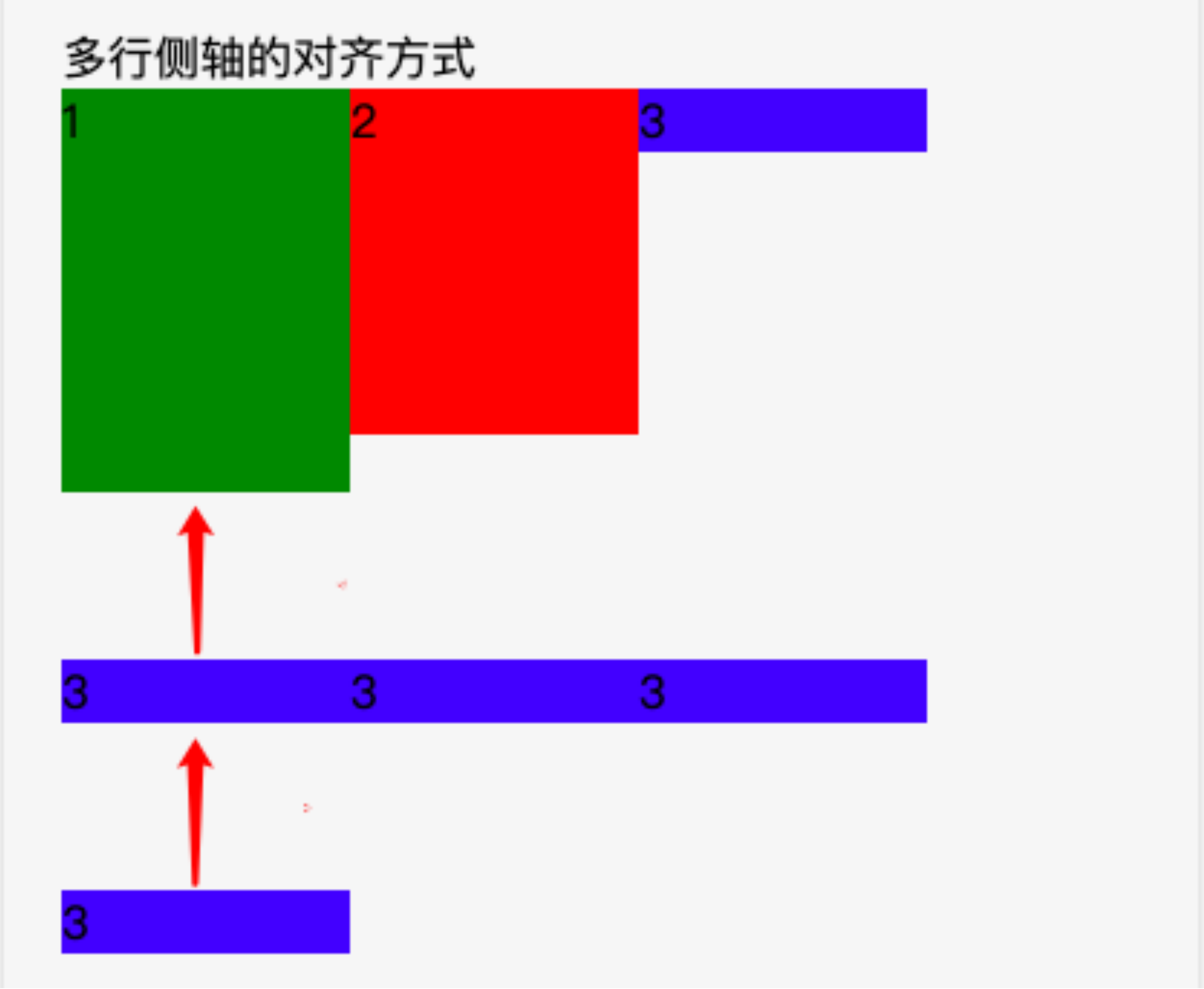


多了换行

align-content的值

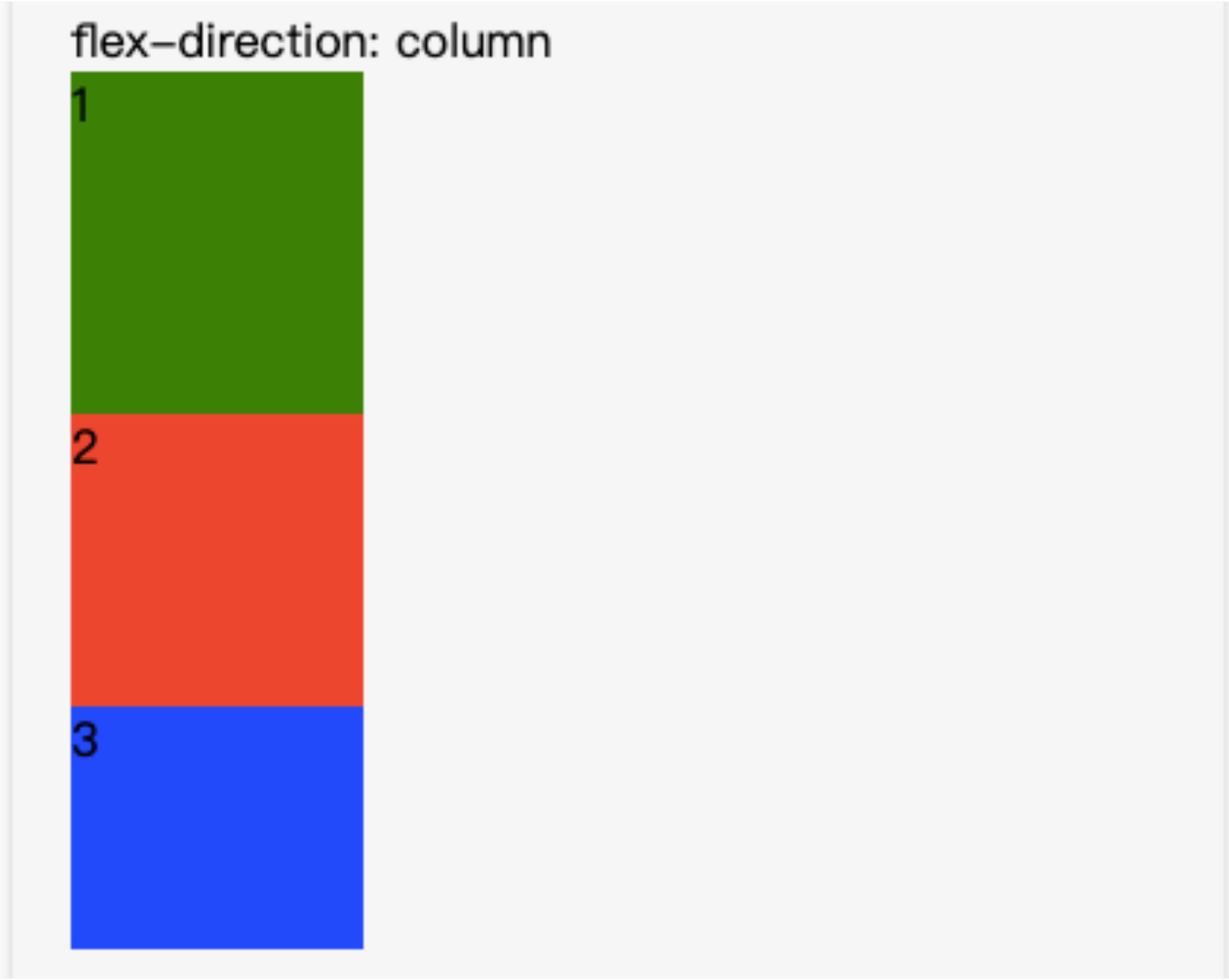
stretch、center、flex-start、flex-end、space-between、space-around

```
<view class="section">
<view class="section__title">18 多行侧轴的对齐方式</view>
<view class="flex-wrp" style="flex-direction:row;justify-content:flex-start;align-items:baseline;flex-wrap:wrap;align-content:space-between;height:300px;">
<view class="flex-item bc_green">1</view>
<view class="flex-item bc_red">2</view>
<view style="height:auto;" class="flex-item bc_blue">3</view>
<view style="height:auto;" class="flex-item bc_blue">3</view>
<view style="height:auto;" class="flex-item bc_blue">3</view>
<view style="height:auto;" class="flex-item bc_blue">3</view>
<view style="height:auto;" class="flex-item bc_blue">3</view>
</view>
</view>
```



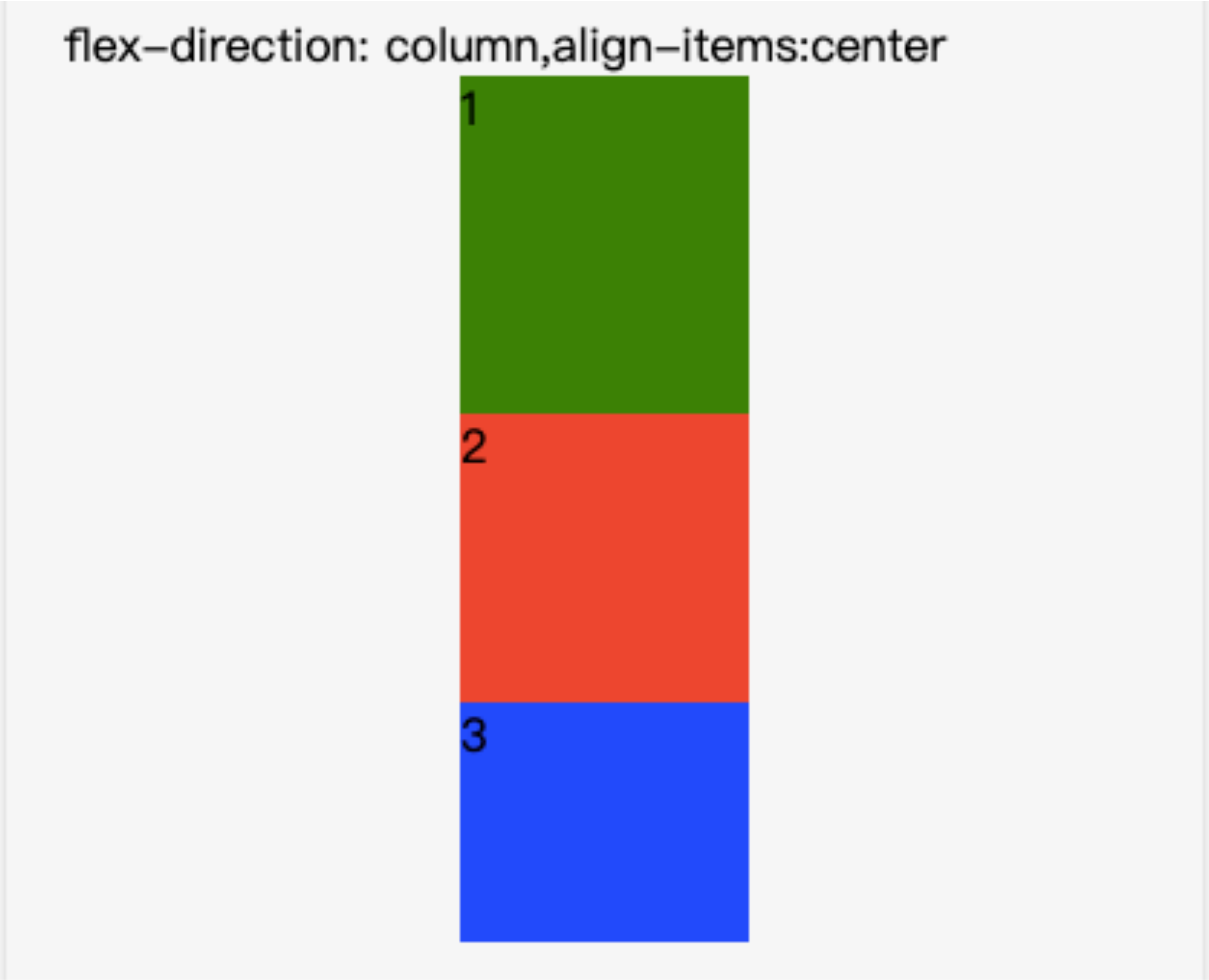
多行侧轴的对齐方式

flex-direction 的值



flex-direction: column

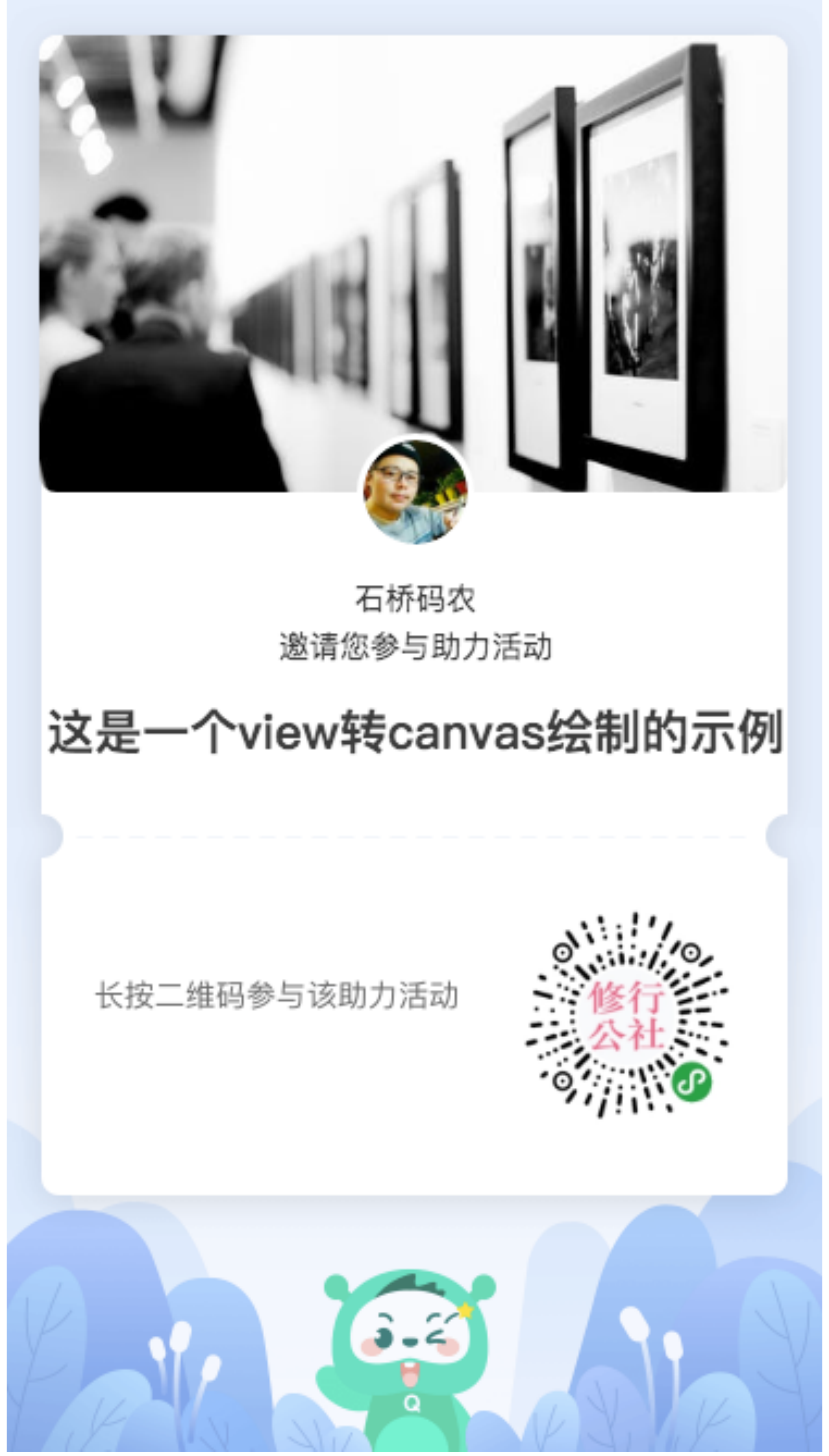
```
<view class="section">
<view class="section__title">21flex-direction: column,align-items:center</view>
<view class="flex-wrp" style="height: 300px;flex-direction:column;align-items:center;">
<view class="flex-item bc_green">1</view>
<view class="flex-item bc_red">2</view>
<view class="flex-item bc_blue">3</view>
</view>
</view>
```



flex–direction: column,align–items:center

如何把 view 上的内容绘制在画布上，生成一张海报？

https://github.com/Kujiale-Mobile/Painter



石桥码农
邀请您参与助力活动

这是一个view转canvas绘制的示例

长按二维码参与该助力活动

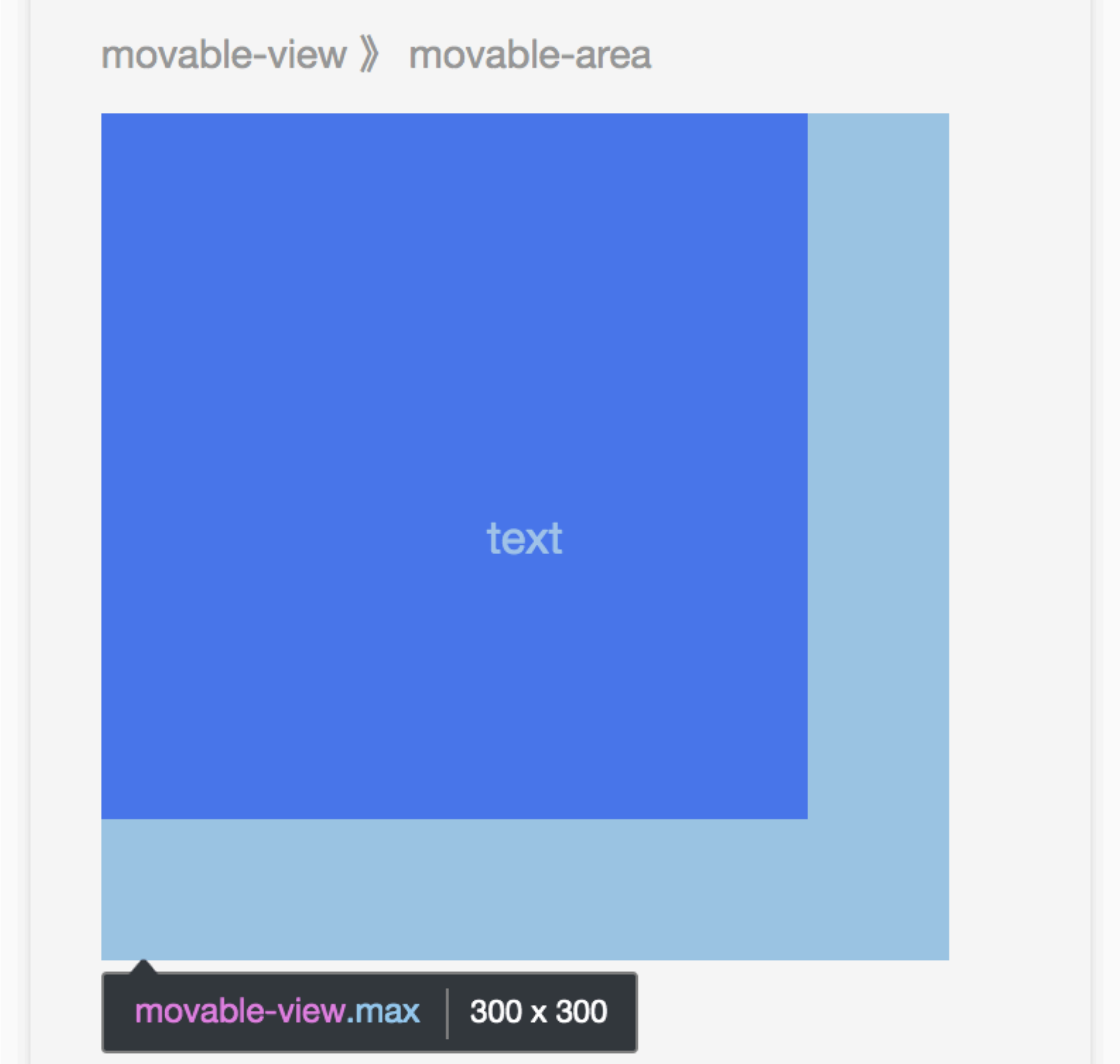
2.6 可移动容器及可移动区域介绍：
如何实现单条消息左滑删除功能？（一）

关于元素的定位

```css
#word { position: sticky; top: 10px; }
```

关于sticky： https://developer.mozilla.org/zh-CN/docs/Web/CSS/position

三种拖拽情况

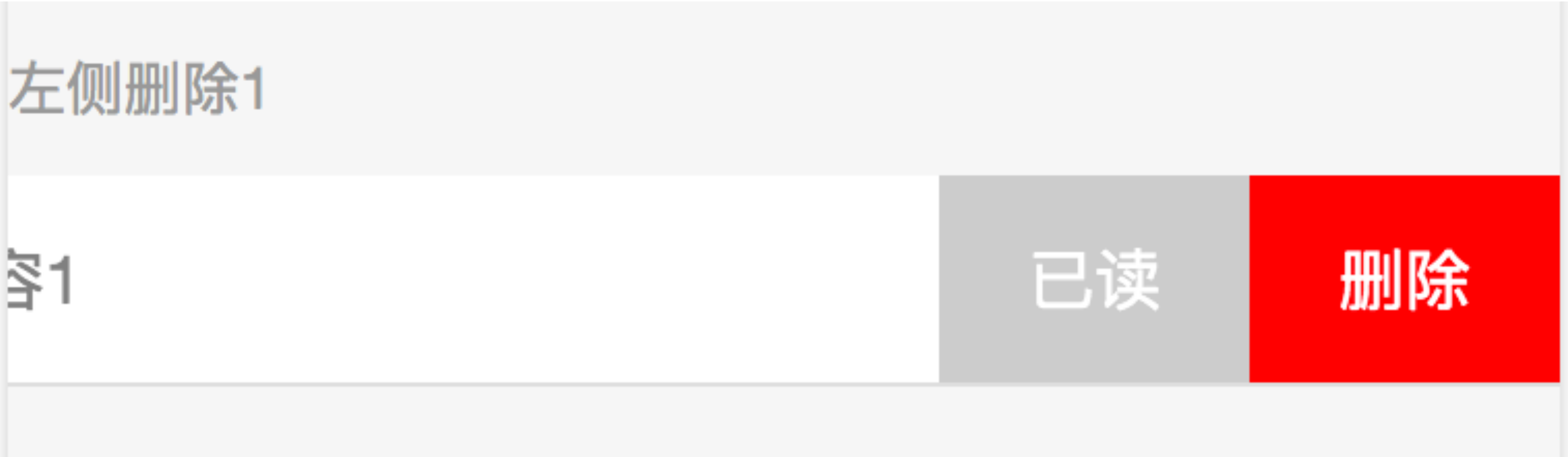

movable-view 》 movable-area
text
movable-view.max
300 x 300

实现动画

```
<movable-area>
  <movable-view inertia damping="20" friction="1"
bindchange="onMovableViewChange" animation="{{true}}" x="{{x}}" y="{{y}}"
direction="all">text</movable-view>
</movable-area>
```

自实现左滑删除效果

```
<movable-area style="width:750rpx;height:100rpx;">
<movable-view style="width:1050rpx;height:100rpx;" direction="horizontal" class="max" direction="all">
<view class="left">这里是插入到组内容 1</view>
<view class="right">
<view class="read">已读</view>
<view class="delete">删除</view>
</view>
</movable-view>
</movable-area>
```

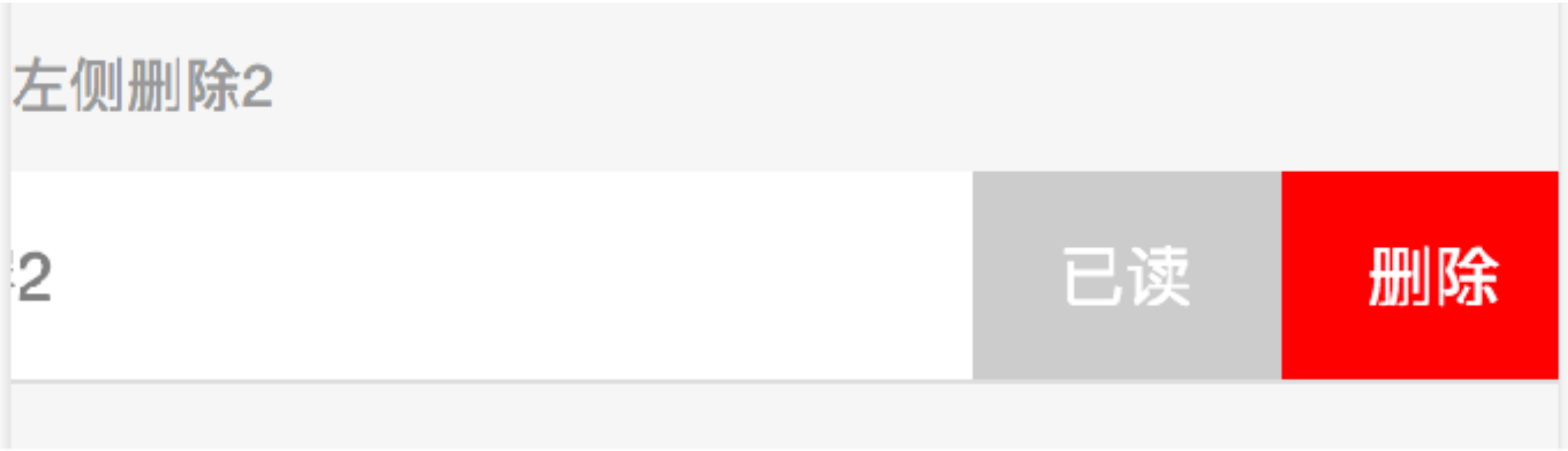


2.7 可移动容器及可移动区域介绍：
如何实现单条消息左滑删除功能？（二）

使用 npm 安装项目模块

https://github.com/wechat-miniprogram/slide-view

```
cd miniprogram
npm init -y
npm install --save miniprogram-slide-view
```



```
{
"usingComponents": {
"slide-view": "miniprogram-slide-view"
}
}
```

```
<slide-view class="slide" width="750" height="100" slideWidth="300">
<view class="left" slot="left">这里是插入到组内容 2</view>
<view class="right" slot="right">
<view class="read">已读</view>
<view class="delete">删除</view>
</view>
</slide-view>
```

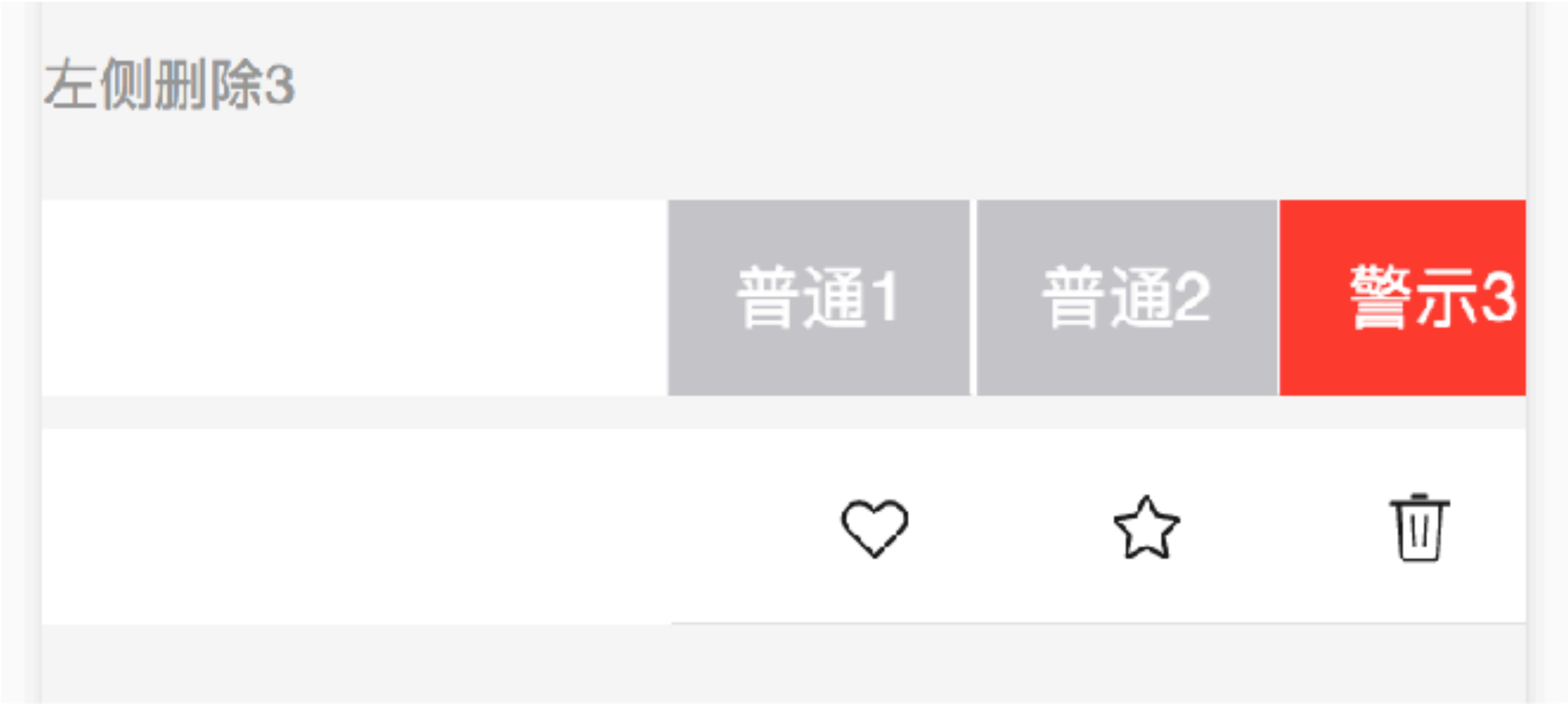
“Component is not found in path”

以扩展声明的方式，使用 weui 组件库

https://github.com/wechat-miniprogram/weui-miniprogram

- kbone: 多端开发框架
- weui: WeUI 组件库

```
"useExtendedLib": {
"weui": true
}
```



```
/* @import '/miniprogram_npm/weui-miniprogram/weui-wxss/dist/style/weui.wxss'; */
<mp-slideview buttons="{{slideButtons}}" icon="{{true}}" bindbuttontap="slideButtonTap">
<view class="weui-slidecell">
左滑可以删除（图标 Button）
</view>
</mp-slideview>
```

weui组件slideview源码：weui-miniprogram/src/slideview

```
<wxs module="handler" src="./slideview.wxs"></wxs>
<view ....>
<view ...class="weui-slideview__left left" style="width:100%;">
<slot></slot>
</view>
<view class="weui-slideview__right right">
...
</view>
</view>
```

关于页面未注册错误

"xxx" has not been registered yet.

课后作业

https://developers.weixin.qq.com/miniprogram/dev/extended/weui/

```
npm install --save wxml-to-canvas
```

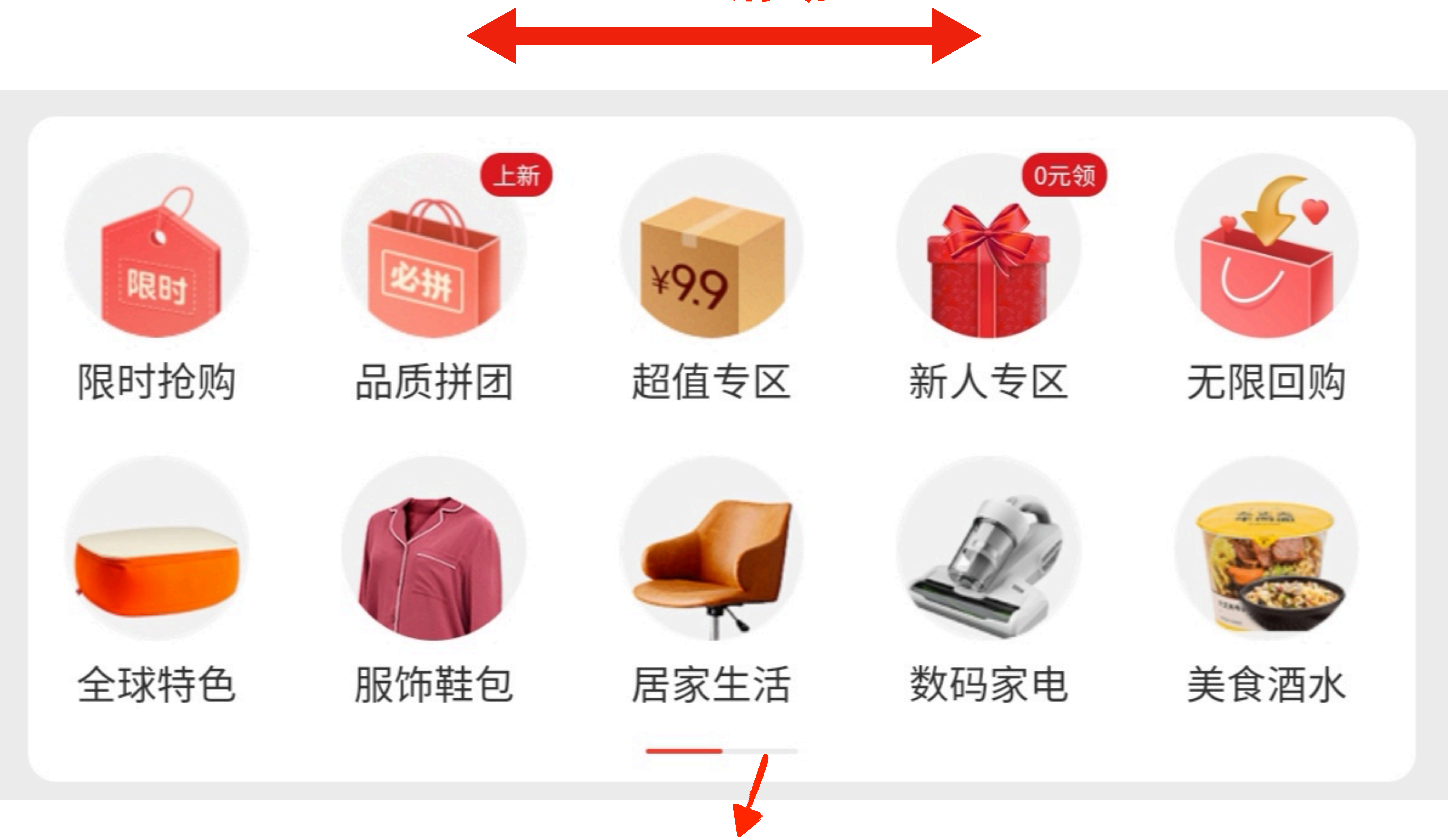
关于 Canvas Api 错误：
createImage fail: http://tmp/xxx....png

错误： createImage fail: http://tmp/xxx....png

```
data: {
use2dCanvas: false
}

const use2dCanvas = false //compareVersion(SDKVersion, '2.9.2') >= 0

img = ` https://cdn.nlark.com/yuque/0/2020/png/1252071/1590050624644-dd5948db-22fe-48d9-af37-8a2a9f099715.png `

const isNetworkFile = /^https?:\/\//.test(img)
const isTempFile = /^wxfile:\/\//.test(img)
const isTempFile = true
```

2.8 scroll-view 介绍：在小程序中如何实现滚动锚定？

左右滑动


上新
0元领
限时
必拼
¥9.9
限时抢购
品质拼团
超值专区
新人专区
无限回购
全球特色
服饰鞋包
居家生活
数码家电
美食酒水

滑动指示

了解 scroll-view 的滚动属性

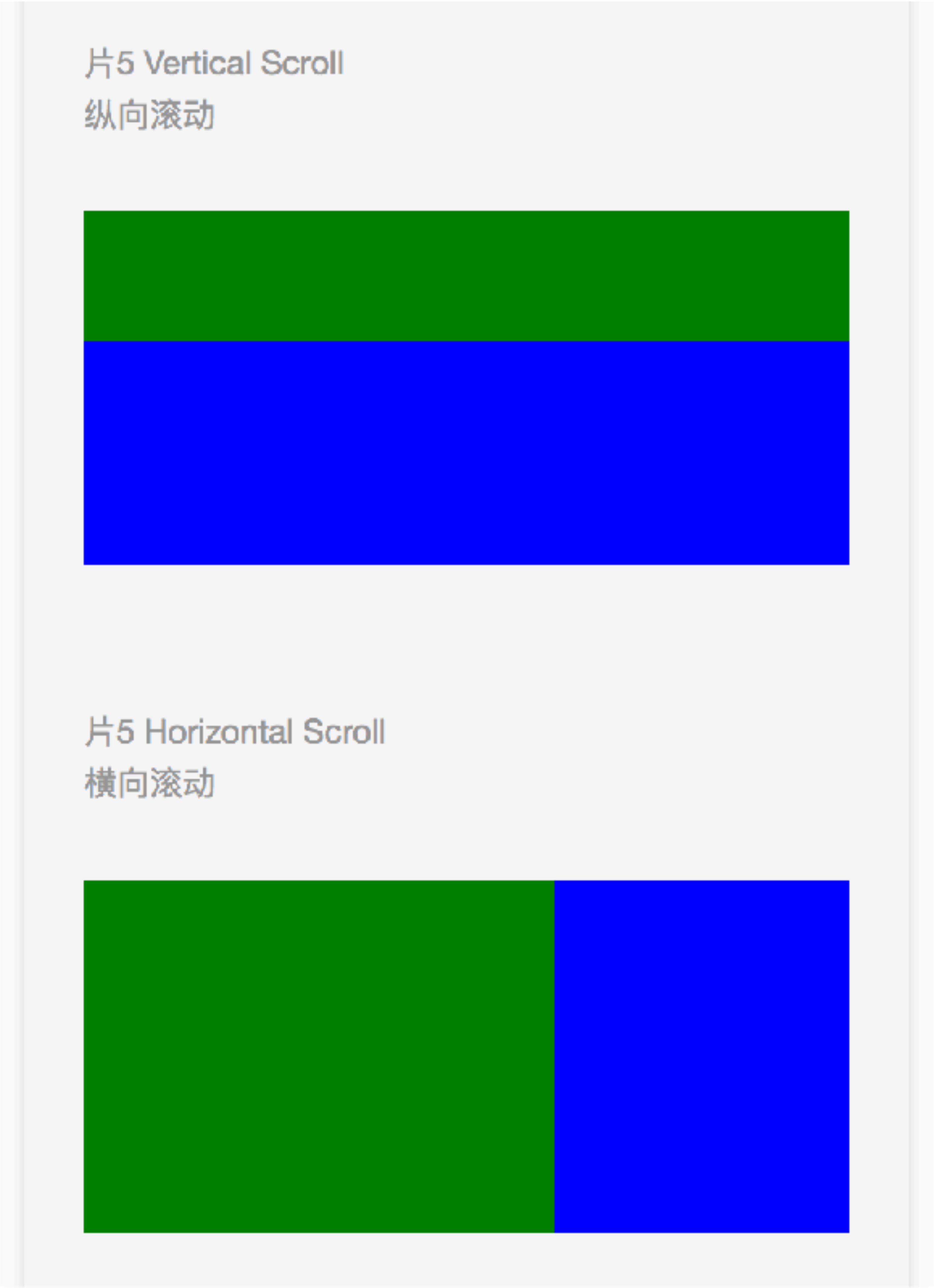
scroll-x

scroll-y

scroll-top

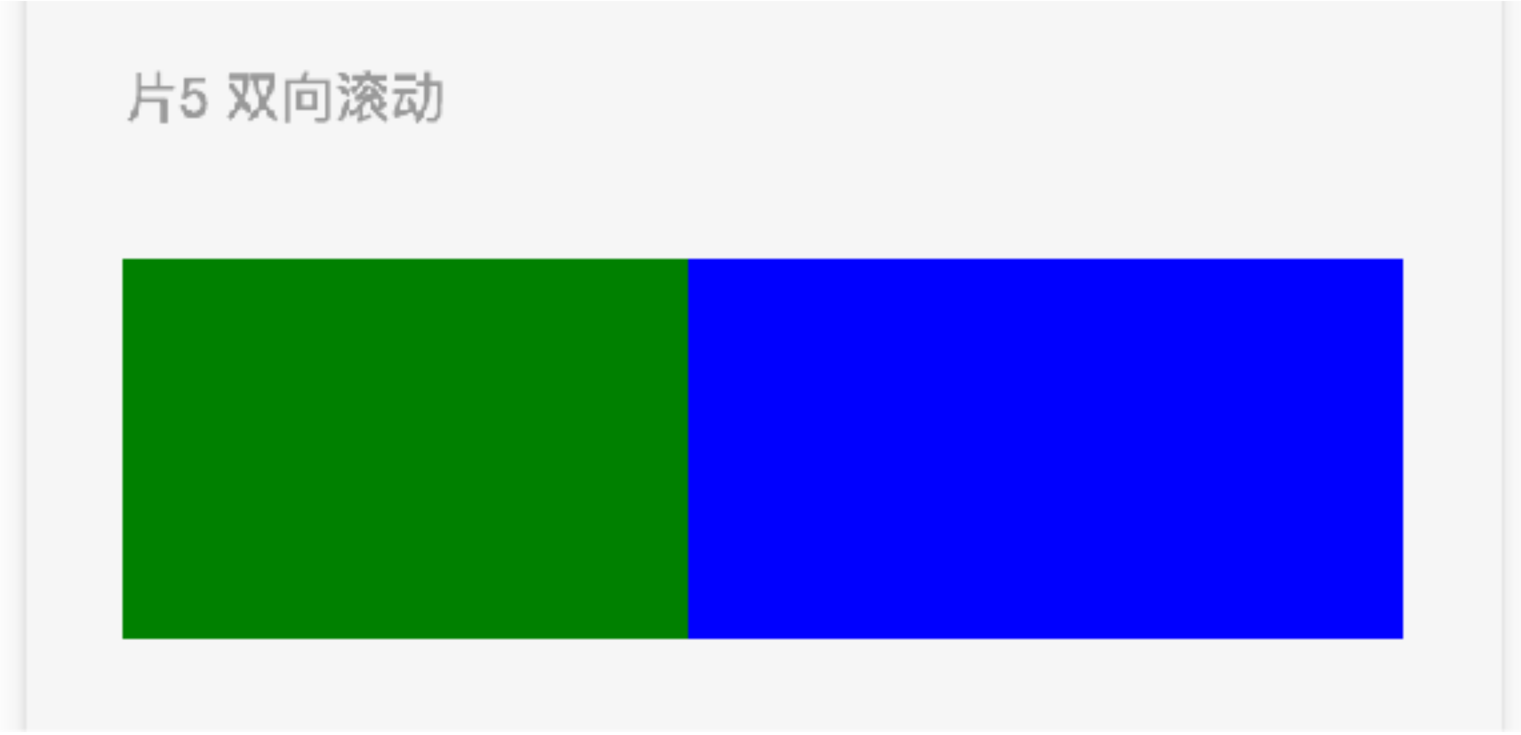
scroll-left

scroll-into-view



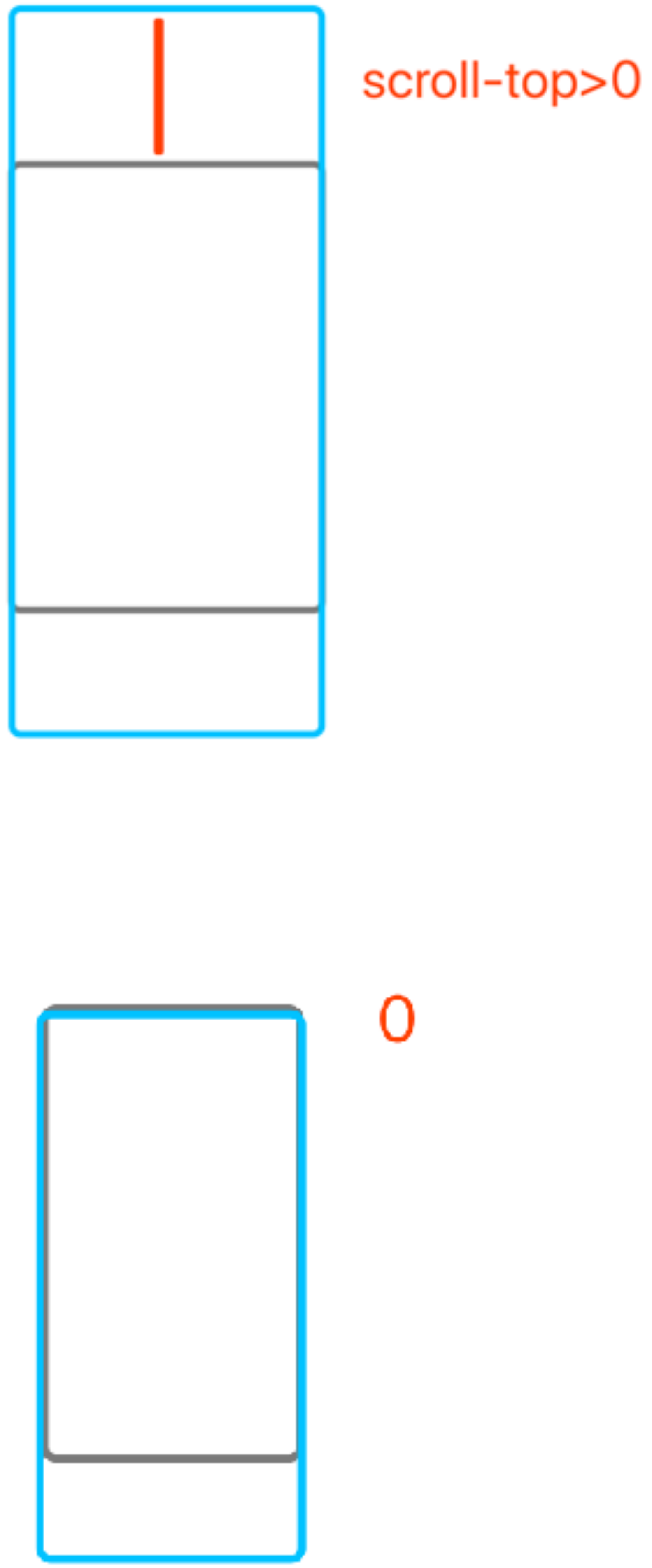
片5 Vertical Scroll
纵向滚动

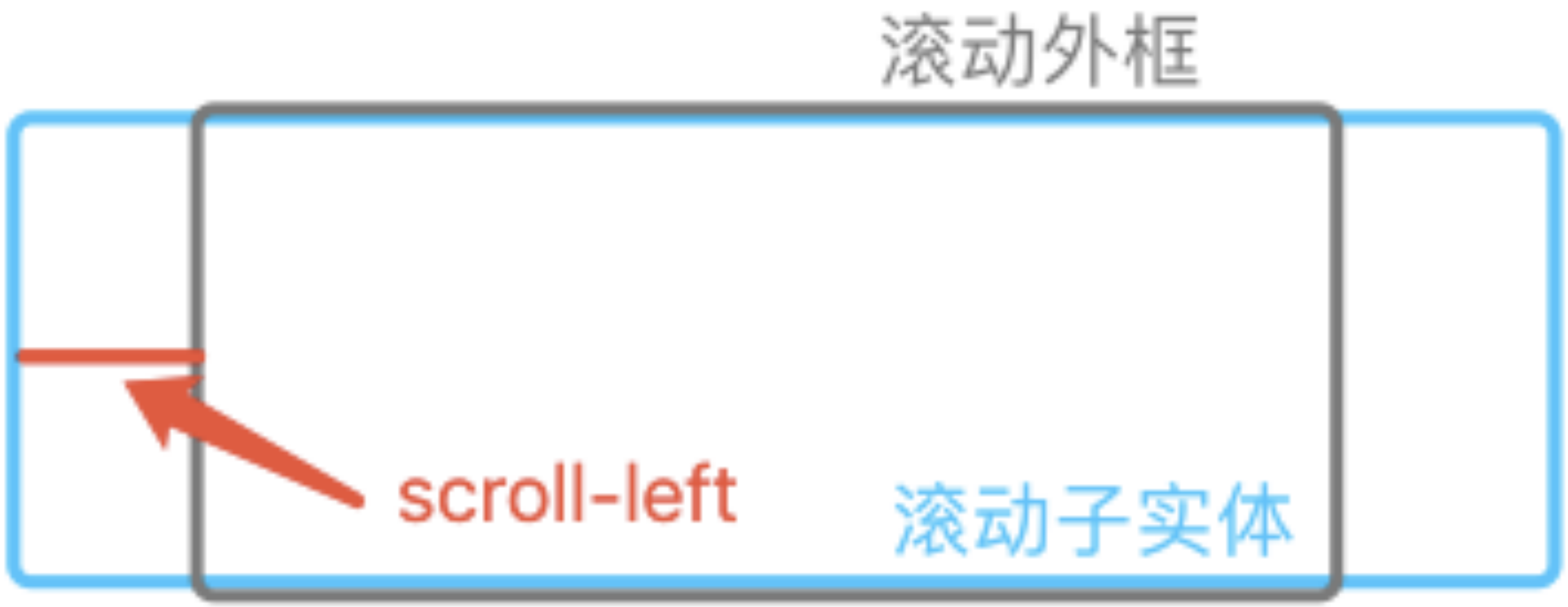
片5 Horizontal Scroll
横向滚动



片5 双向滚动

如何理解 scroll-top 属性?


scroll-top>0
0


滚动外框
scroll-left
滚动子实体

关于绑定更新

scroll-into-view 属性

滚动锚定：scroll-anchoring

overflow-anchor: none

overflow-anchor: auto

upper-threshold

lower-threshold

bindscrolltoupper

bindscrolltolower

bindscroll

```
.slideViewClass .weui-cell{
 padding: 0;
}
```

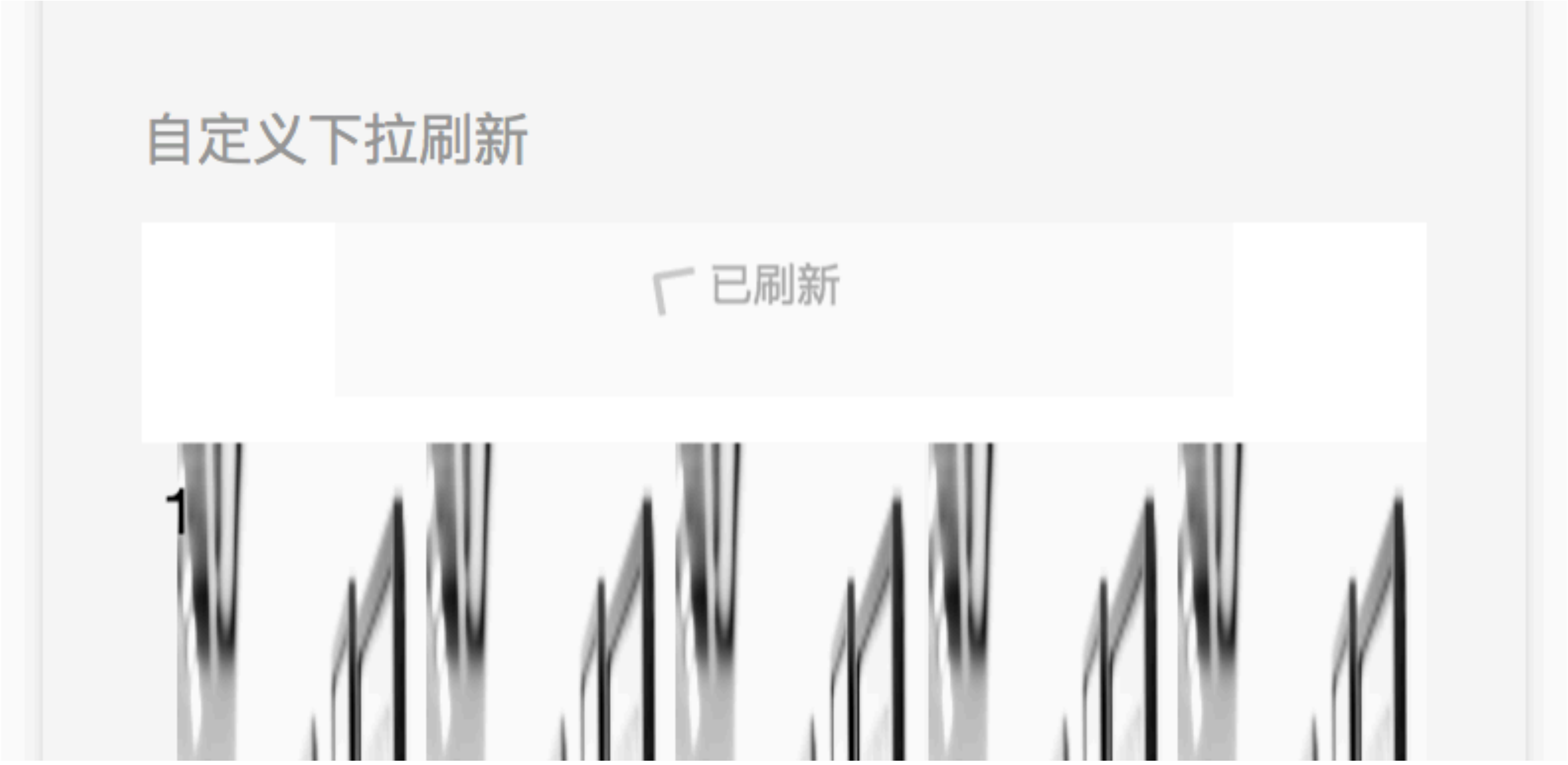
2.9 scroll-view 介绍：如果渲染一个滚动的长列表？

与下拉更新相关的属性：

- refresher-enabled
- refresher-threshold
- refresher-triggered
- bindrefresherpulling
- bindrefresherrefresh
- bindrefresherrestore
- bindrefresherabort

使用 wxs 自定义实现下拉刷新

```
<wxs module="refresh">
    ...
    onPulling: function (e, instance) {
      var p = Math.min (e.detail.dy/ 80, 1)
      var icon = instance.selectComponent ('#refresherIcon')
      icon.setStyle ({
        opacity: p,
        transform: "rotate (" + (90 + p * 180) + "deg)"
      })
      var view = instance.selectComponent ('.refresh-container')
      view.setStyle ({
        opacity: p,
        transform: "scale (" + p + ")"
      })
      if (e.detail.dy >= 80) {
        if (pullingMessage == "下拉刷新") {
          pullingMessage = "释放更新"
          instance.callMethod ("setData", {
            pullingMessage
          })
        }
      }
    }
  }
</wxs>
```



自定义下拉刷新

已刷新

```
onRefresh: function (e, instance) {
  // 此时手拉开了，进入了加载中的状态
  pullingMessage = "更新中"
  instance.callMethod ("setData", {
    pullingMessage: pullingMessage,
    refresherTriggered: true
  })
  instance.callMethod ("willCompleteRefresh", {})
}
```

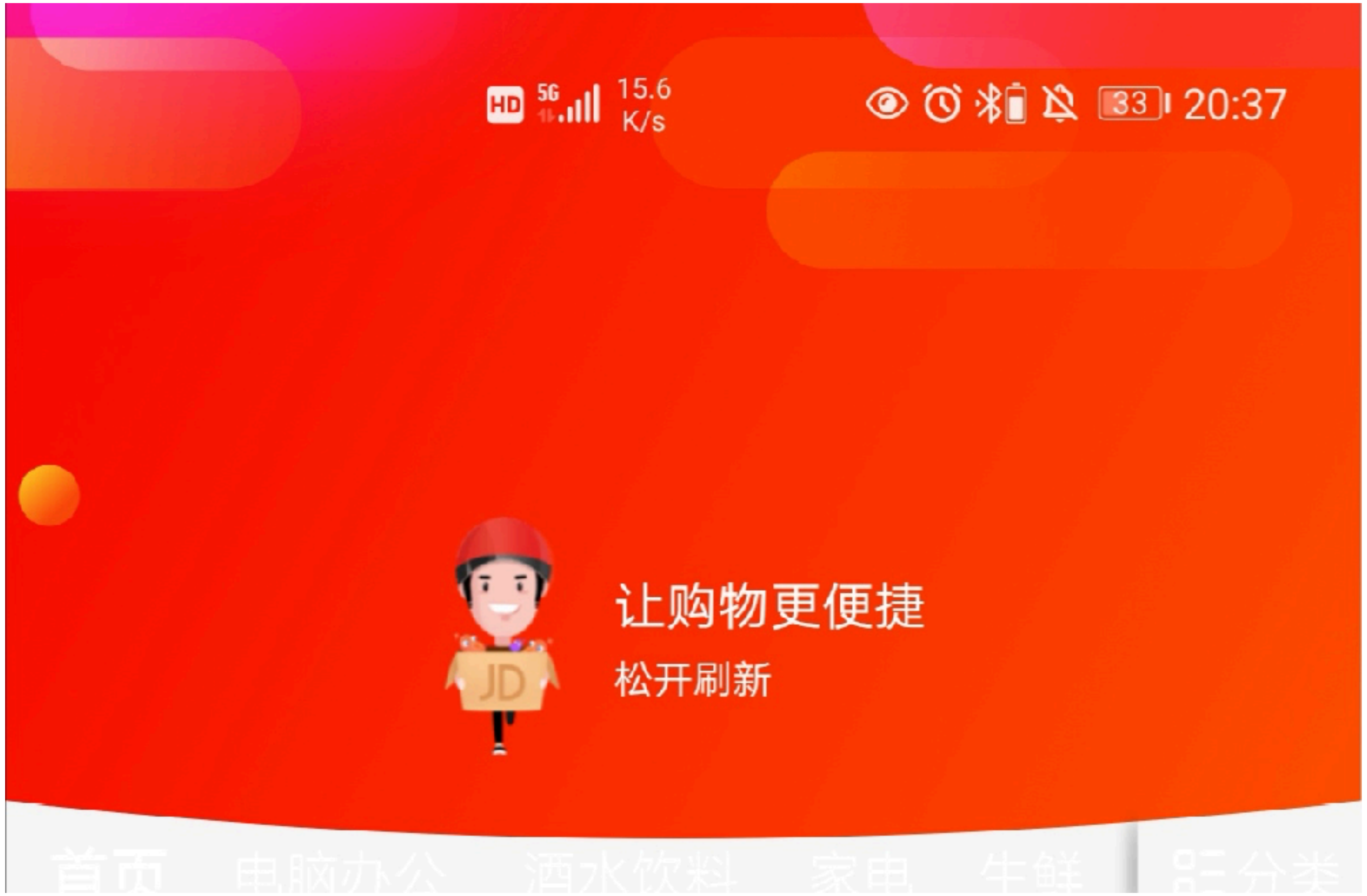
```
willCompleteRefresh (){
  let intervalId = setInterval ((()=>{
    let pullingMessage = this.data.pullingMessage
    console.log (pullingMessage,pullingMessage == ' 更新中 ')
    if (pullingMessage.length < 7){
      pullingMessage += '.'
    } else {
      pullingMessage = ' 更新中 '
    }
    this.setData ({
      pullingMessage
    })
  },500)
  setTimeout ((()=>{
    clearInterval (intervalId)
    this.setData ({
      pullingMessage:"已刷新",
      refresherTriggered:false,
    })
  },2000)
}
```

mescroll：https://github.com/mescroll/mescroll

minirefresh：https://github.com/minirefresh/minirefresh

最佳实践

```
white-space:nowrap;
display:inline-block
```



让购物更便捷

松开刷新

使用 scroll-view 时，如何优化使用 setData 向其传递大数据、渲染长列表？

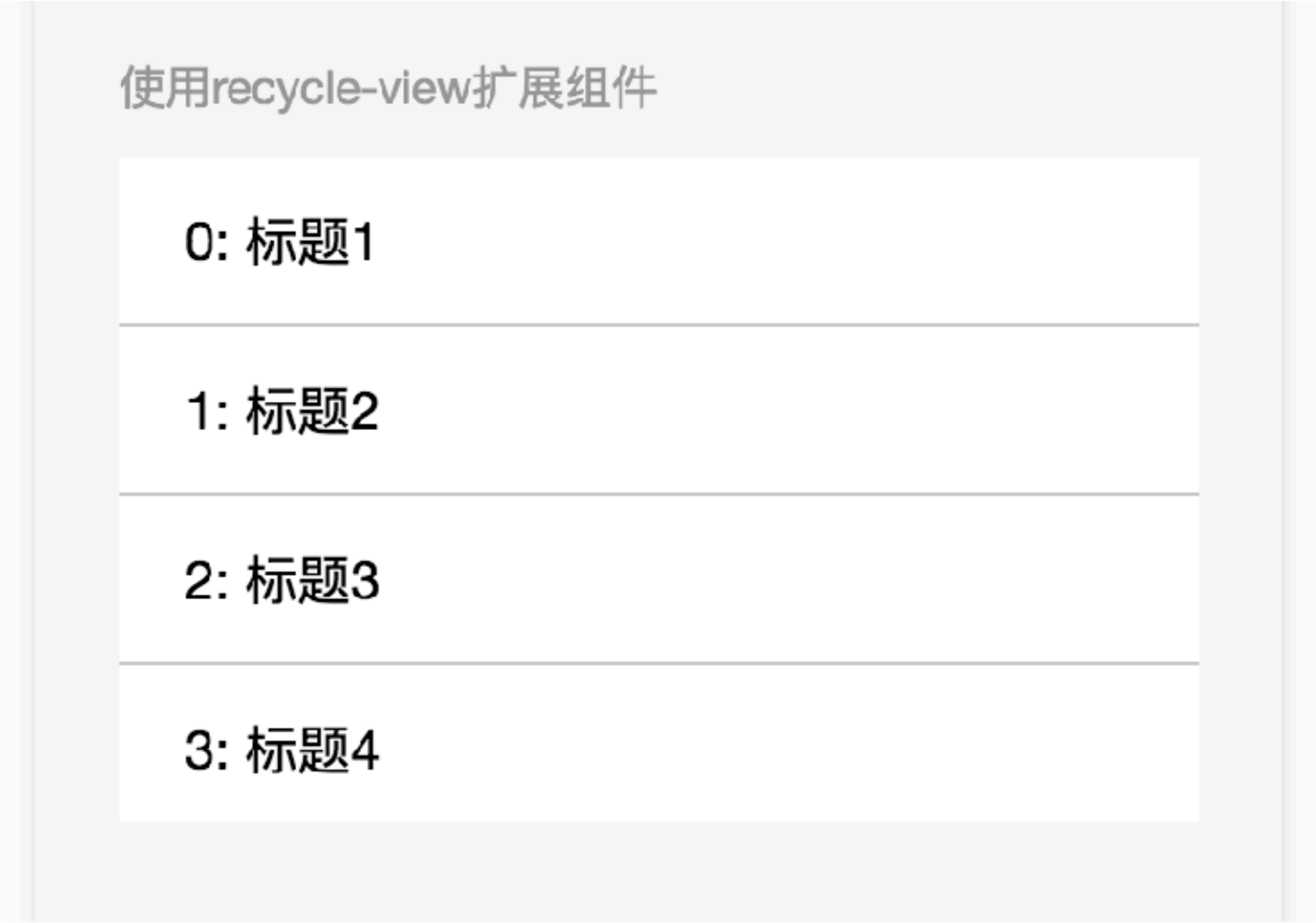
```
// 更新二维数组
const updateList = `tabs [${activeTab}].list [${page}]`
const updatePage = `tabs [${activeTab}].page`
this.setData ({
[updateList]: res.data,
[updatePage]: page + 1
})
<view wx:for="{{gameListWrap}}" wx:for-item="gameList">
...
</view>
```

```
let tabData = this.data.tabs [activeTab]
tabData.list.push (res.data)
tabData.page = page+1
let key = `tabs [${activeTab}]`
this.setData ({
[key]: tabData
})
```

```
const updateListStr = `gameListData [${activeTab}][${page}]`
const updatePageStr = `pages [${activeTab}]`
this.setData ({
[updateListStr]: res,
[updatePageStr]: page + 1
})
```

使用recycle-view扩展组件：
https://developers.weixin.qq.com/miniprogram/dev/extended/component-plus/recycle-view.html

```
<recycle-view height="200" batch="{{batchSetRecycleData}}" id="recycleId" batch-
key="batchSetRecycleData" style="background:white;">
  <recycle-item wx:for="{{recycleList}}" wx:key="index" class='item'>
    <view>
      {{item.id}}: {{item.name}}
    </view>
  </recycle-item>
</recycle-view>
var ctx = createRecycleContext({
  id: 'recycleId',
  dataKey: 'recycleList',
  page: this,
  itemSize: {
    width: rpx2px(650),
    height: rpx2px(100)
  }
})
let newList = []
...
ctx.append(newList)
```



使用recycle-view扩展组件

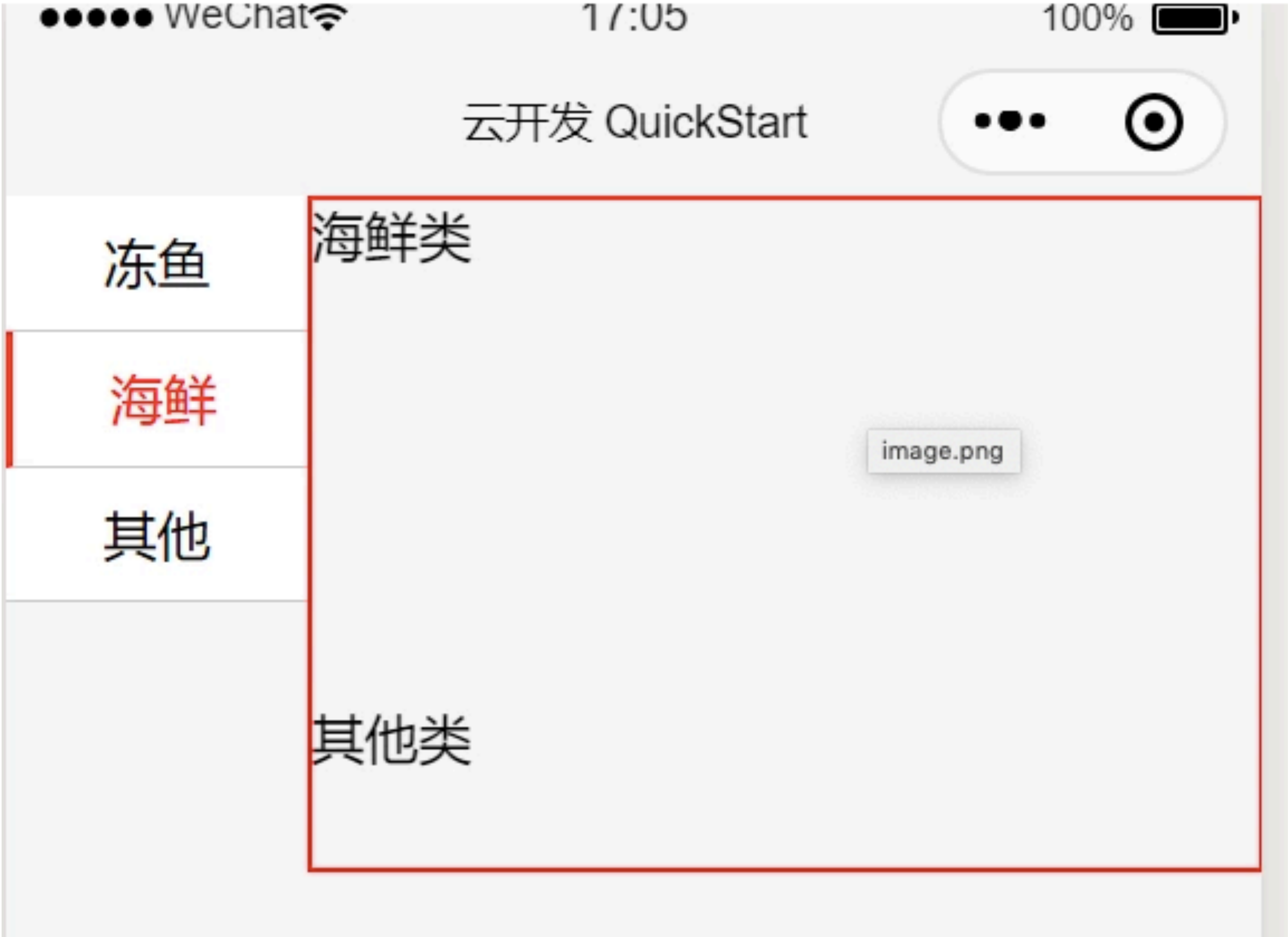
0: 标题1

1: 标题2

2: 标题3

3: 标题4

如何实现购物类小程序分类选择物品的页面？


●●●●● WeChat
17:05
100%
云开发 QuickStart
冻鱼
海鲜
其他
海鲜类
image.png
其他类

```
<!-- 左侧菜单 -->
<scroll-view scroll-y='true' class='nav'>
<view wx:for='{{list}}' wx:key='{{item.id}}' id='{{item.id}}'
class='navList {{currentIndex==index?"active":""}}' bindtap="menuListOnClick" data-
index='{{index}}'>{{item.name}}</view>
</scroll-view>
<!-- 右侧内容 -->
<scroll-view scroll-y='true' scroll-into-view='{{activeViewId}}' bindscroll='scrollFunc'>
<view class="fishList" wx:for='{{content}}' id='{{item.id}}' wx:key='{{item.id}}'>
<p>{{item.name}}</p>
</view>
</scroll-view>
```

```
// 单击左侧菜单
menuListOnClick:function (e){
  let me=this;
  me.setData ({
    activeViewId:e.target.id,
    currentIndex:e.target.dataset.index
  })
}
// 滚动时触发，计算当前滚动到的位置对应的菜单是哪个
scrollFunc:function (e){
  this.setData ({
    scrollTop:e.detail.scrollTop
  })
  for (let i = 0; i < this.data.heightList.length; i++) {
    let height1 = this.data.heightList [i];
    let height2 = this.data.heightList [i + 1];
    if (!height2 || (e.detail.scrollTop >= height1 && e.detail.scrollTop < height2)) {
      this.setData ({
        currentIndex: i
      })
      return;
    }
  }
  this.setData ({
    currentIndex: 0
  })
}
```

作业

2.10 滚动选择器：如何自定义省市区多级联动选择器？（一）

object array 这个属性值是怎么回事？如何使用它？

```
array: ['美国', '中国', '巴西', '日本'],
objectArray: [
{
id: 0,
name: '美国'
},
{
id: 1,
name: '中国'
},
{
id: 2,
name: '巴西'
},
{
id: 3,
name: '日本'
}
],
```

如何取用户选择的值？

如何使用多项选择器，如何从多项选择器里面取到值，并使用取到的值？

```
<picker mode="multiSelector" bindchange="bindMultiPickerChange"
bindcolumnchange="bindMultiPickerColumnChange" value="{{multiIndex}}"
range="{{multiArray}}">
<view class="picker">
当前选择：{{multiArray[0][multiIndex[0]]}}，{{multiArray[1][multiIndex[1]]}}，
{{multiArray[2][multiIndex[2]]}}
</view>
</picker>
```

如何理解选择器的 mode 属性？
它都有哪些可以用的值？

1. selector 普通选择器

2. multiSelector 多列选择器

3. time 时间选择器

4. date 日期选择器

5. region 省市区选择器

如何在日期选择器中直接选择月？

如何处理选择器的粒度？

1. year 选择器粒度为年

2. month 选择器粒度为月份

3. day 选择器粒度为天

微信小程序的编码规范是什么?
事件名如何写、函数名如何写、
变量如何写、文件名如何写？等等

一. 四种常用的命名方法

二. WXSS 命名规范

三. WXML 命名规范

四. JavaScript 命名规范

五. 文件与目录的命名规范

一. 四种常用的命名方法

1.小驼峰命名法

2.大驼峰命名法

3.连字符命名法

4.匈牙利命名法

WXSS 命名规范

模块__模块元素-描述符

- weui-cell__radio
- weui-cell__radio-selected

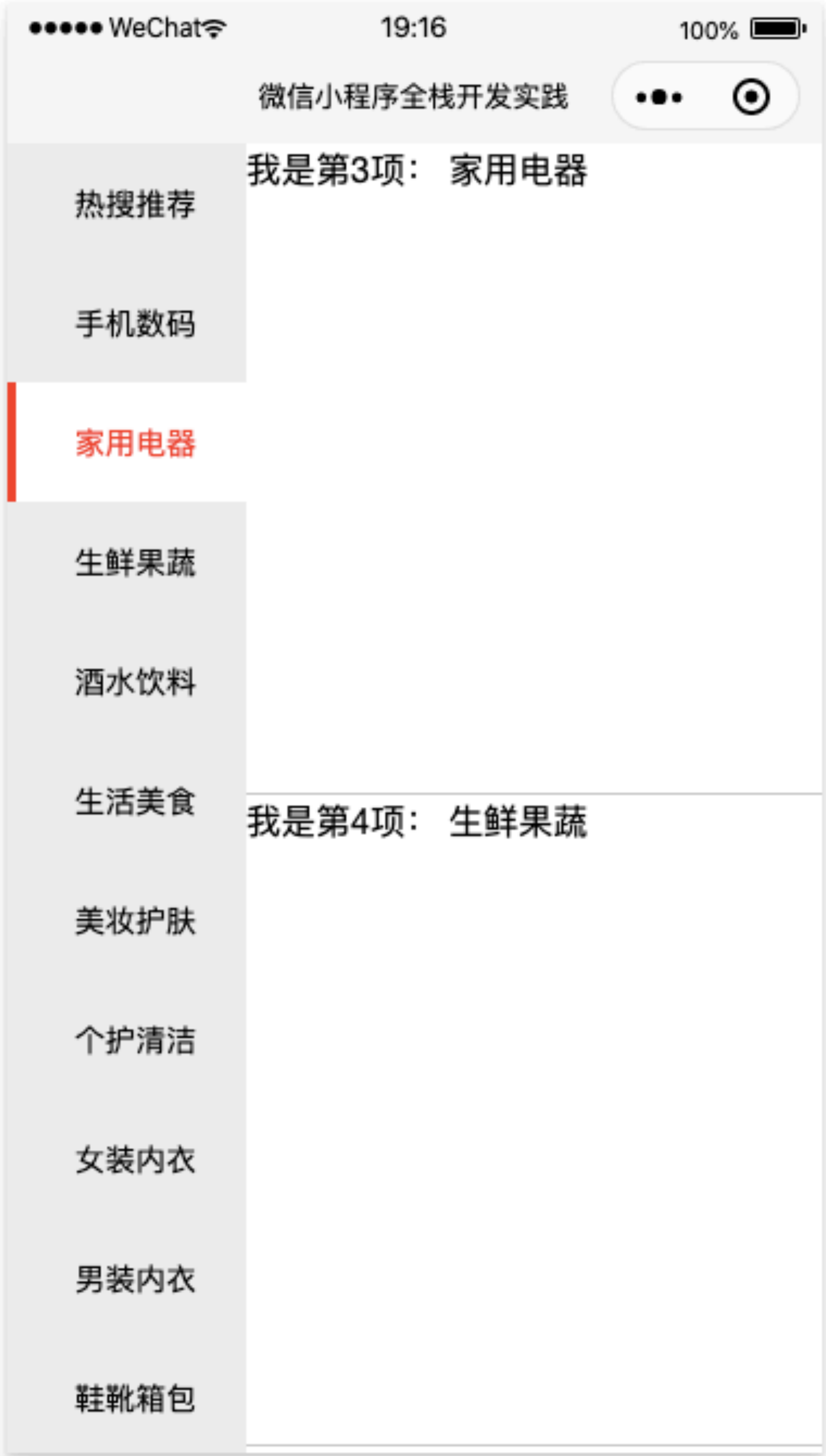
WXML 命名规范

JavaScript 命名规范

文件与目录的命名规范

作业：如何使用 vtabs

```
npm i @miniprogram-component-plus/vtabs --save
npm i @miniprogram-component-plus/vtabs-content --save
"usingComponents": {
"mp-vtabs": "@miniprogram-component-plus/vtabs/index",
"mp-vtabs-content": "@miniprogram-component-plus/vtabs-content/index"
}
```



我是第3项： 家用电器

我是第4项： 生鲜果蔬

```
relations: {
  '../vtabs/index': {
      type: 'parent'
  }
},
methods: {
  calcHeight: function calcHeight(callback) {
      var query = this.createSelectorQuery();
      query.select('.weui-vtabs-content__item').boundingClientRect(function
(rect) {
          callback && callback(rect);
      }).exec();
  }
}
```

```
relations: {
  '../vtabs-content/index': {
      type: 'child',
      linked: function linked(target) {
          var _this = this;
          target.calcHeight(function (rect) {
              _this.data._contentHeight[target.data.tabIndex] = rect.height;
              if (_this._calcHeightTimer) {
                  clearTimeout(_this._calcHeightTimer);
              }
              _this._calcHeightTimer = setTimeout(function () {
                  _this.calcHeight();
              }, 100);
          });
      },
      unlinked: function unlinked(target) {
          delete this.data._contentHeight[target.data.tabIndex];
      }
  }
},
```

2.11 滚动选择器：如何自定义省市区多级联动选择器？（二）

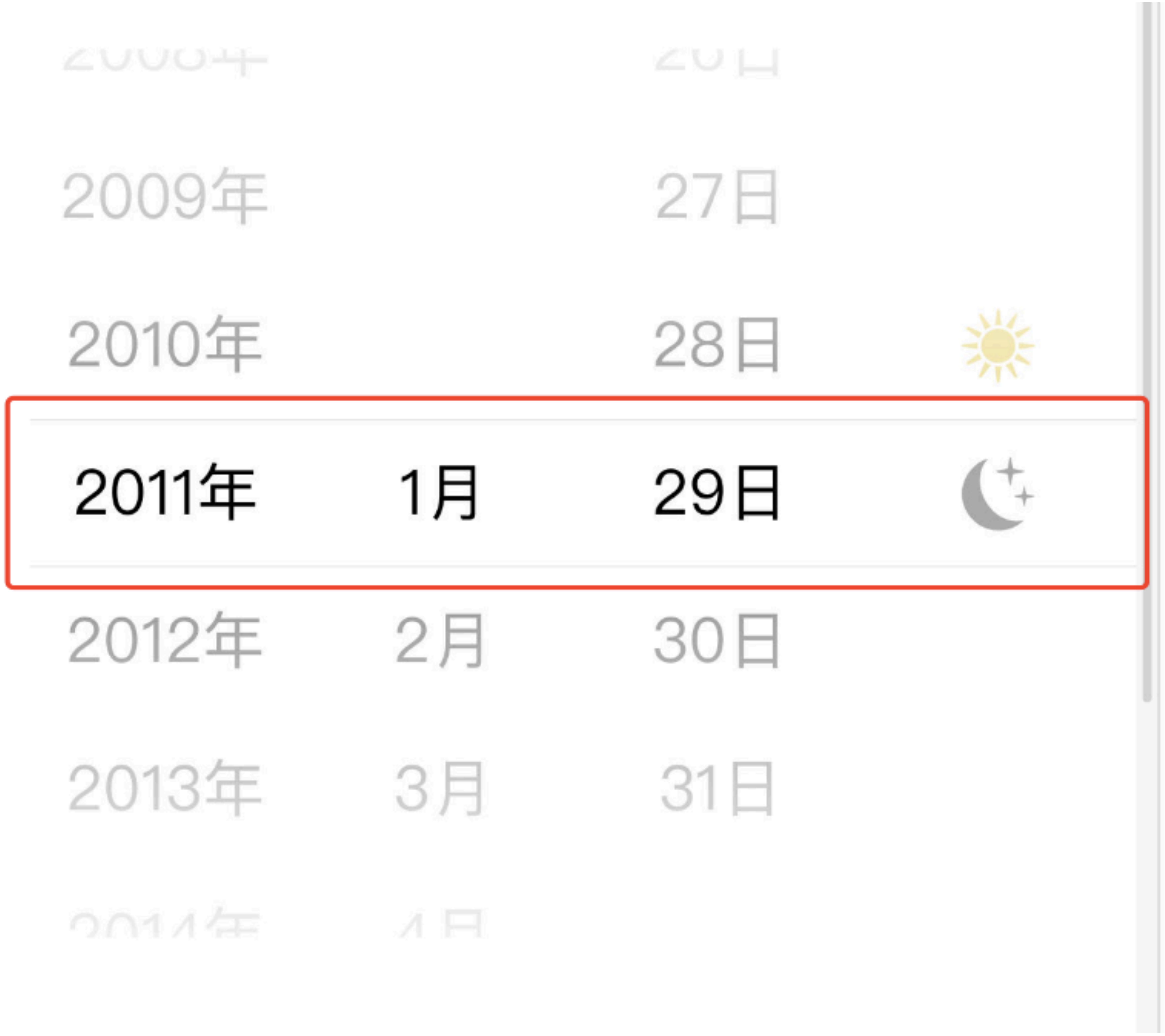
picker 与 picker-view 有什么区别？

我们已经选择了值，

为什么还要给选择器的 value 属性绑定值呢？

```
<picker bindchange="bindPickerChange" value="{{index}}" range="{{array}}">
<view class="picker">
当前选择：{{array[index]}}
</view>

</picker>
<picker model:value="{{index1}}" range="{{array}}">
<view class="picker">
当前选择：{{array[index1]}}
</view>
</picker>
```

在 picker-view 中，indicator-style
与 indicator-class 这两个属性是做什么用的？



2008年 26日

2009年 27日

2010年 28日 ☀

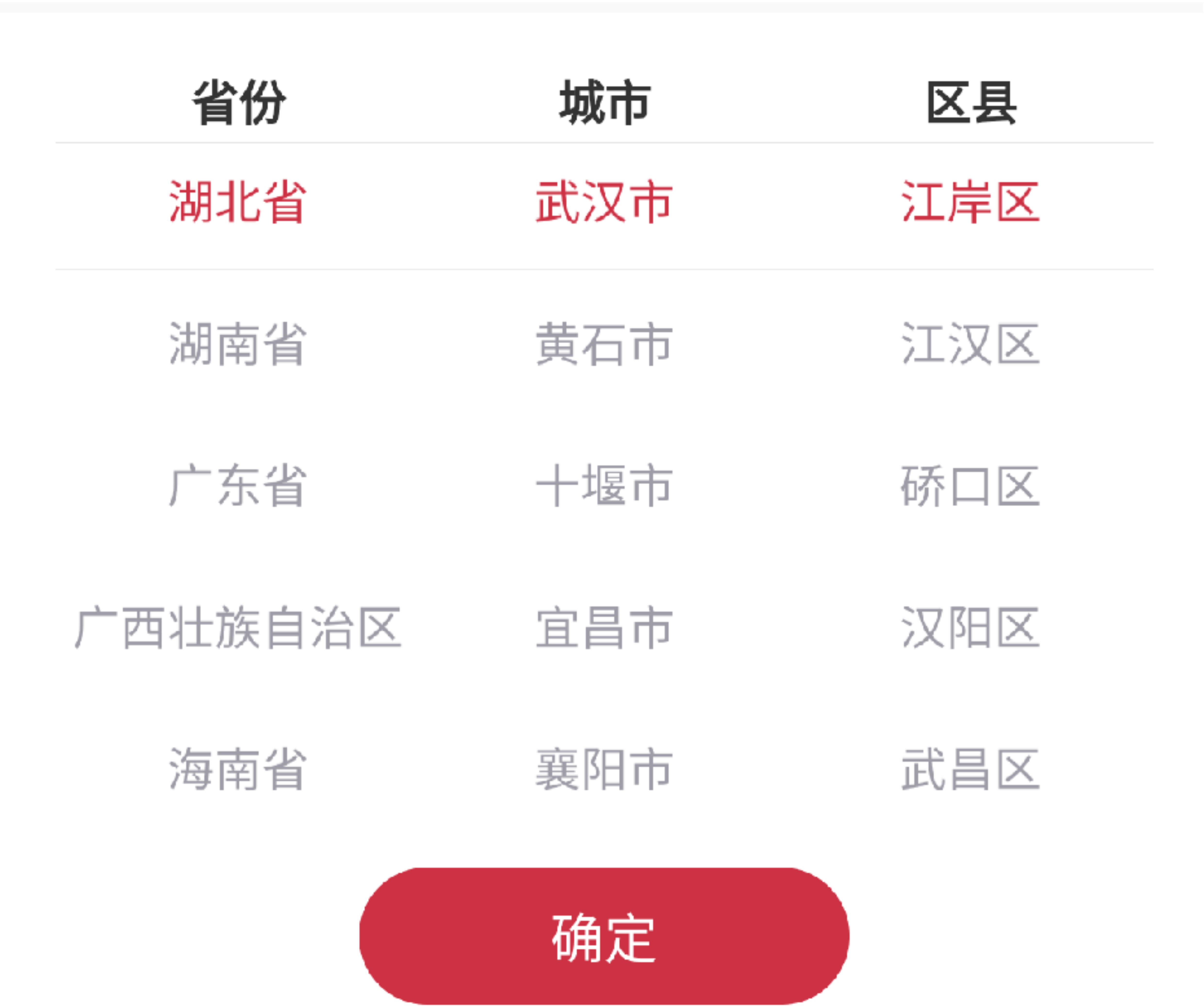
2011年 1月 29日 ☾

2012年 2月 30日

2013年 3月 31日

2014年 4月

如何用 picker-view 实现一个多列选择器？
例如三级联动的省市区选择器。



省份	城市	区县
湖北省	武汉市	江岸区
湖南省	黄石市	江汉区
广东省	十堰市	硚口区
广西壮族自治区	宜昌市	汉阳区
海南省	襄阳市	武昌区

miniprogram/components/region-picker-view/index.wxml:

```
<picker-view class="pick-view__group" bindchange="cityChange"
value="{{value}}" wx:key="*this">
  <picker-view-column indicator-class="item_active">
    <view wx:for="{{provinces}}" class="picker-item"
wx:key="index">{{item.name}}</view>
  </picker-view-column>
  <picker-view-column>
    <view wx:for="{{citys}}" class="picker-item"
wx:key="index">{{item.name}}</view>
  </picker-view-column>
  <picker-view-column>
    <view wx:for="{{areas}}" class="picker-item"
wx:key="index">{{item.name}}</view>
  </picker-view-column>
</picker-view>
```

miniprogram/components/region-picker-view/index.js:

```
cityChange(e) {
  var value = e.detail.value
  ...
  var provinceNum = value[0]
  var cityNum = value[1]
  var areaNum = value[2]
  if (this.data.value[0] !== provinceNum) {
    var id = provinces[provinceNum].id
    this.setData({
      value: [provinceNum, 0, 0],
      citys: address.citys[id],
      areas: address.areas[address.citys[id][0].id]
    })
  } else if (this.data.value[1] !== cityNum) {
    var id = citys[cityNum].id
    this.setData({
      value: [provinceNum, cityNum, 0],
      areas: address.areas[citys[cityNum].id]
    })
  } else {
    this.setData({
      value: [provinceNum, cityNum, areaNum]
    })
  }
}
```

miniprogram/components/region-picker-view/city.js:

```
var provinces = [{
  "name": "北京市",
  "id": "110000"
},...]
var citys = {
  "110000": [
    {
      "province": "北京市",
      "name": "市辖区",
      "id": "110100"
    }
  ],...]
var areas = {
  "110100": [
    {
      "city": "市辖区",
      "name": "东城区",
      "id": "110101"
    },...]
module.exports = {
  citys,//cities
  provinces,
  areas
}
```

miniprogram/components/region-picker-view2/index.js：

```
// var address = require('./city')
Component({
  options: {
    multipleSlots: false
  },
  properties: {},
  data: {
    value: [0, 0, 0], // 地址选择器省市区 暂存 currentIndex
    regionText: '', //所在地区
    provinces: null, // 一级地址
    citys: null, // 二级地址
    areas: null, // 三级地址
    visible: false
  },
  ready(){},
  methods: {}
})
```

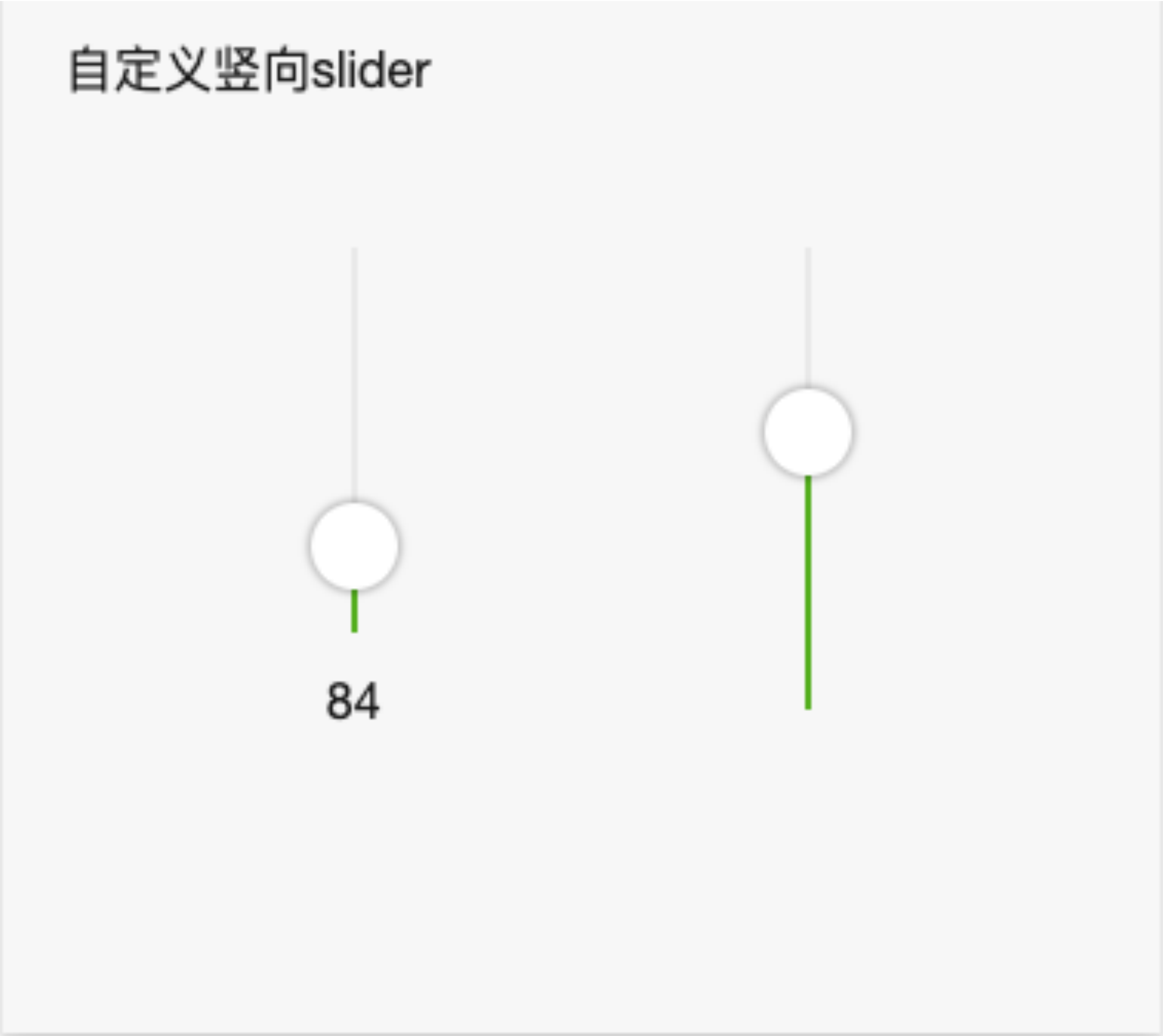
Cannot use wxs function to handle custom event "ready" miniprogram/components/region-picker-view2/index.wxml:

```
<picker-view class="pick-view__group"
bindpickstart="{{region.onPickStart}}" bindpickend="{{region.onPickEnd}}"
bindchange="{{region.onChange}}" value="{{value}}" wx:key="*this">
  <picker-view-column indicator-class="item_active">
    <view wx:for="{{provinces || region.provinces}}" class="picker-item"
wx:key="index">{{item.name}}</view>
  </picker-view-column>
  <picker-view-column>
    <view wx:for="{{citys || region.citys}}" class="picker-item"
wx:key="index">{{item.name}}</view>
  </picker-view-column>
  <picker-view-column>
    <view wx:for="{{areas || region.areas}}" class="picker-item"
wx:key="index">{{item.name}}</view>
  </picker-view-column>
</picker-view>
```

```
<view change:class="{{region.onPropSigned}}" class="address-item" ....>
...
</view>
region.onPropSigned = function(newValue, oldValue, ownerInstance, instance){
  ownerInstance.callMethod("setData", {
    provinces: region.provinces
    , citys: region.citys
    , areas: region.areas
    , value: [0, 0, 0] // 地址选择器省市区 暂存 currentIndex
    , regionText: ''
  })
}
<picker-view-column indicator-class="item_active">
  <view wx:for="{{provinces}}" class="picker-item" wx:key="index">{{item.name}}</view>
</picker-view-column>
```

2.12 滑动选择器：如何基于 wxs 自定义实现一个竖向的 slider?

https://developers.weixin.qq.com/community/develop/article/doc/0002c8ac9603d0600e09b003b56413



```
<wxs module="eventHandle" src=“./index.wxs"></wxs>

<view class="slider-container" change:prop="{{eventHandle.propsChange}}"
prop="{{ {max,min,step,value,totalTop,totalHeight,disabled} }}" >

propsChange: function(newValue, oldValue, ownerIns, ins) {
  var state = ownerIns.getState()
  var step = newValue.step;
  var min = newValue.min;
  var max = newValue.max;
  ...
  state.totalTop = newValue.totalTop
  state.totalHeight = newValue.totalHeight
  if (newValue.totalTop !== null && newValue.totalHeight !== null) {
    calculate(ownerIns, state, function(currentValue){
      ownerIns.callMethod("setCurrent", state.current + state.min)
    })
  }
}
```

- selectComponent
- selectAllComponents
- setStyle
- addClass/removeClass/ hasClass
- getDataset
- callMethod(funcName:string, args:object)
- requestAnimationFrame
- getState
- triggerEvent(eventName, detail)

```
<view class="slider-container"...>
<view class="slider-upper" id="upper" catchtap="{{eventHandle.tap}}">
  <view class="slider-upper-line" style="background-color:
{{backgroundColor}}"></view>
</view>
<view class="slider-middle">
  <view
    class="slider-block"
    style="background-color:{{blockColor}};box-shadow:
{{blockColor=='#ffffff'?'0 0 2px 2px rgba(0,0,0,0.2)':'none'}};width:
{{blockSize}}px;height:{{blockSize}}px"
    catchtouchstart="{{eventHandle.start}}"
    catchtouchmove="{{eventHandle.move}}"
    catchtouchend="{{eventHandle.end}}"
    ></view>
</view>
<view class="slider-lower" id="lower" catchtap="{{eventHandle.tap}}">
```

```css
.slider-container {
  flex: 1;
  margin: 0 20px;
  width: 0;
  display: flex;
  flex-direction: column;
  align-items: center;
}
```

```
var calculate = function(instance,state,changeCallback){
  ...
  instance.selectComponent("#upper").setStyle({
    height: (100 - percent) + "%"
  })
  ....
}
```

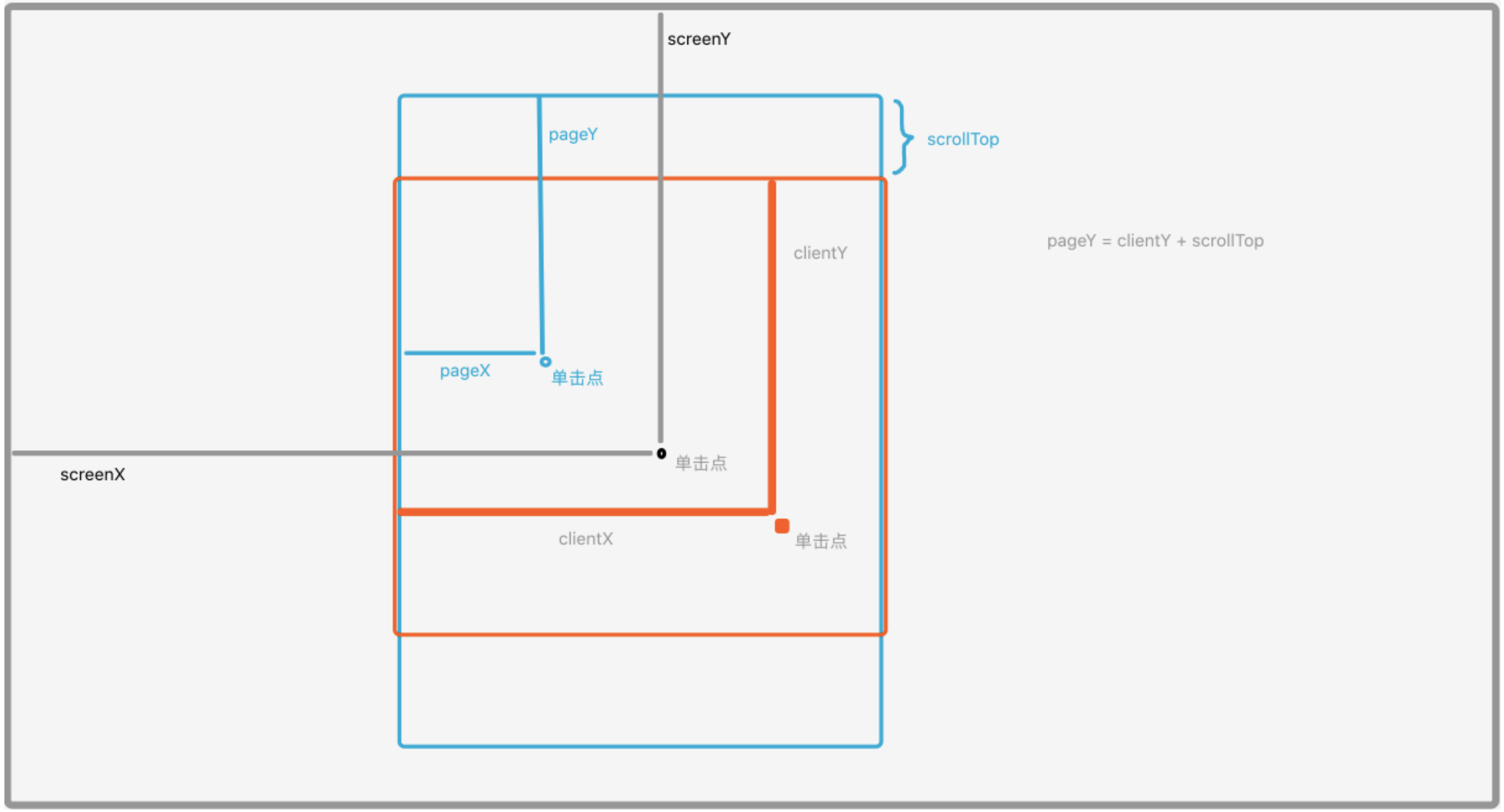
```
<view class="slider-append" data-percent="1"
bindtap="{{eventHandle.tapEndPoint}}"></view>

<view class="slider-append" data-percent="0"
bindtap="{{eventHandle.tapEndPoint}}"></view>

tapEndPoint: function(e, ins){
  var state = ins.getState()
  var percent = e.currentTarget.dataset.percent
  state.offset = (state.max - state.min) * percent
  calculate(ins, state, function (currentValue) {
    ins.triggerEvent("change", {
      value: currentValue
    })
    ins.callMethod("setCurrent", currentValue)
  })
}
```

```
tap: function(e, ins) {
  var state = ins.getState()
  var percent = (state.totalTop + state.totalHeight -
e.changedTouches[0].pageY) / state.totalHeight
  state.offset = (state.max - state.min) * percent
  calculate(ins, state, function(currentValue){
    ins.callMethod("setCurrent", currentValue)
    ins.triggerEvent("change", {
      value: currentValue
    })
  })
}
```

1. identifier

2. pageX, pageY

3. clientX, clientY



```
start: function(e, ins) {
  var state = ins.getState()
  state.startPoint = e.changedTouches[0]
  var currentPx = state.current / (state.max - state.min) * state.totalHeight
  state.currentPx = currentPx
},
move: function(e, ins) {
  var state = ins.getState()
  var startPoint = state.startPoint
  var endPoint = e.changedTouches[0]
  var currentPx = state.currentPx + startPoint.pageY - endPoint.pageY
  var percent = currentPx / state.totalHeight
  ...
  calculate(ins, state, function(currentValue){
    ins.triggerEvent("changing", {
      value: currentValue
    })
    ins.callMethod("setCurrent", currentValue)
  })

。
},
end: function(e, ins) {
  var state = ins.getState()
  ins.triggerEvent("change", {
    value: state.current + state.min
  })
}
```

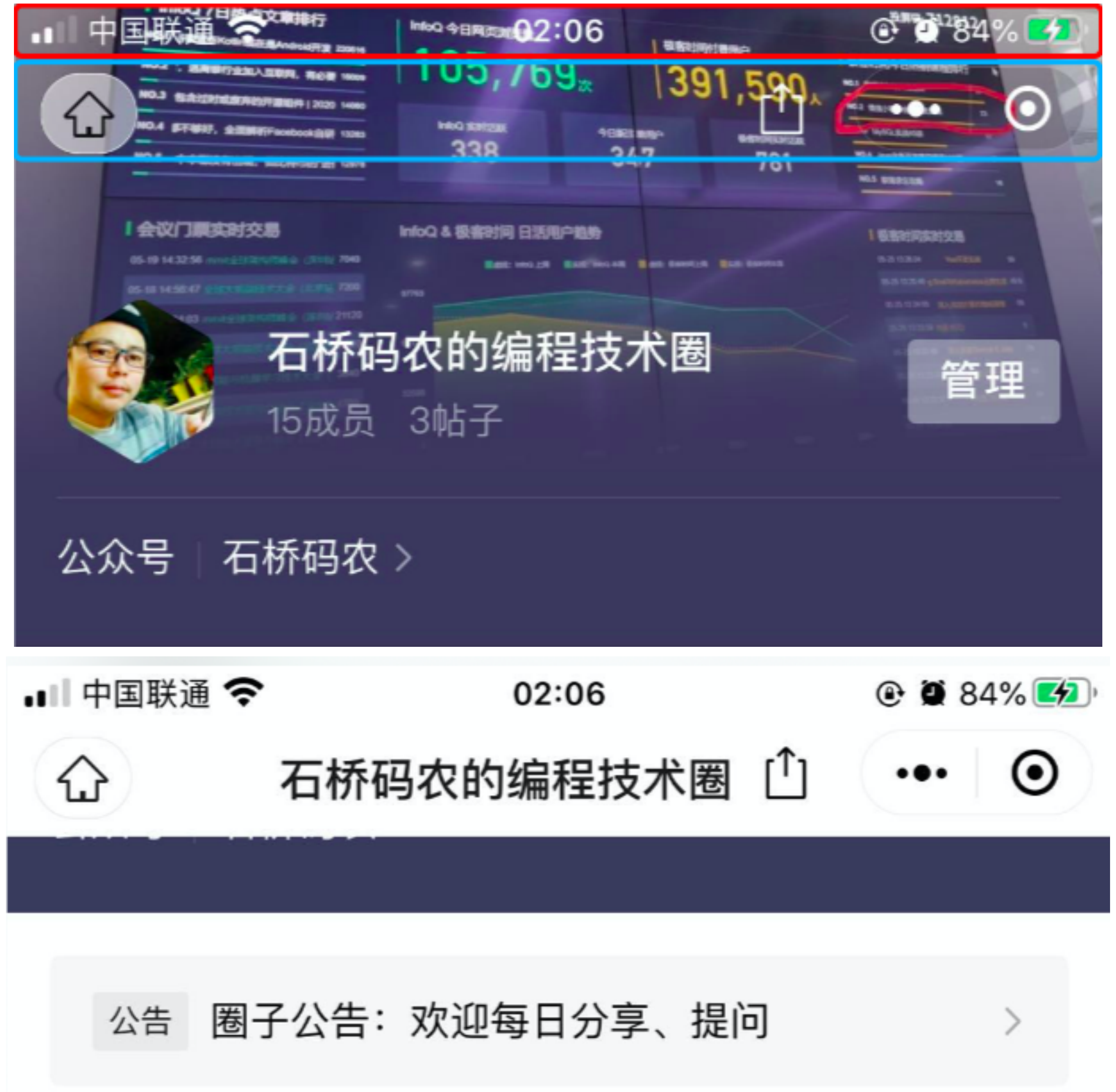
```css
.slider-middle {
  flex-shrink: 1;
  width: 0;
  height: 0;
  display: flex;
  align-items: center;
  justify-content: center;
}
```

压缩计算公式：单个组件压缩量 = 总压缩量 x (单个flex-shrink值 / 总flex-shrink值)

2.13 页面链接：如何自定义一个导航栏？

1. functional-page-navigator
2. navigator

- navigate
- redirect
- switchTab
- reLaunch
- navigateBack
- exit



中国联通 02:06 84%

石桥码农的编程技术圈

15成员 3帖子

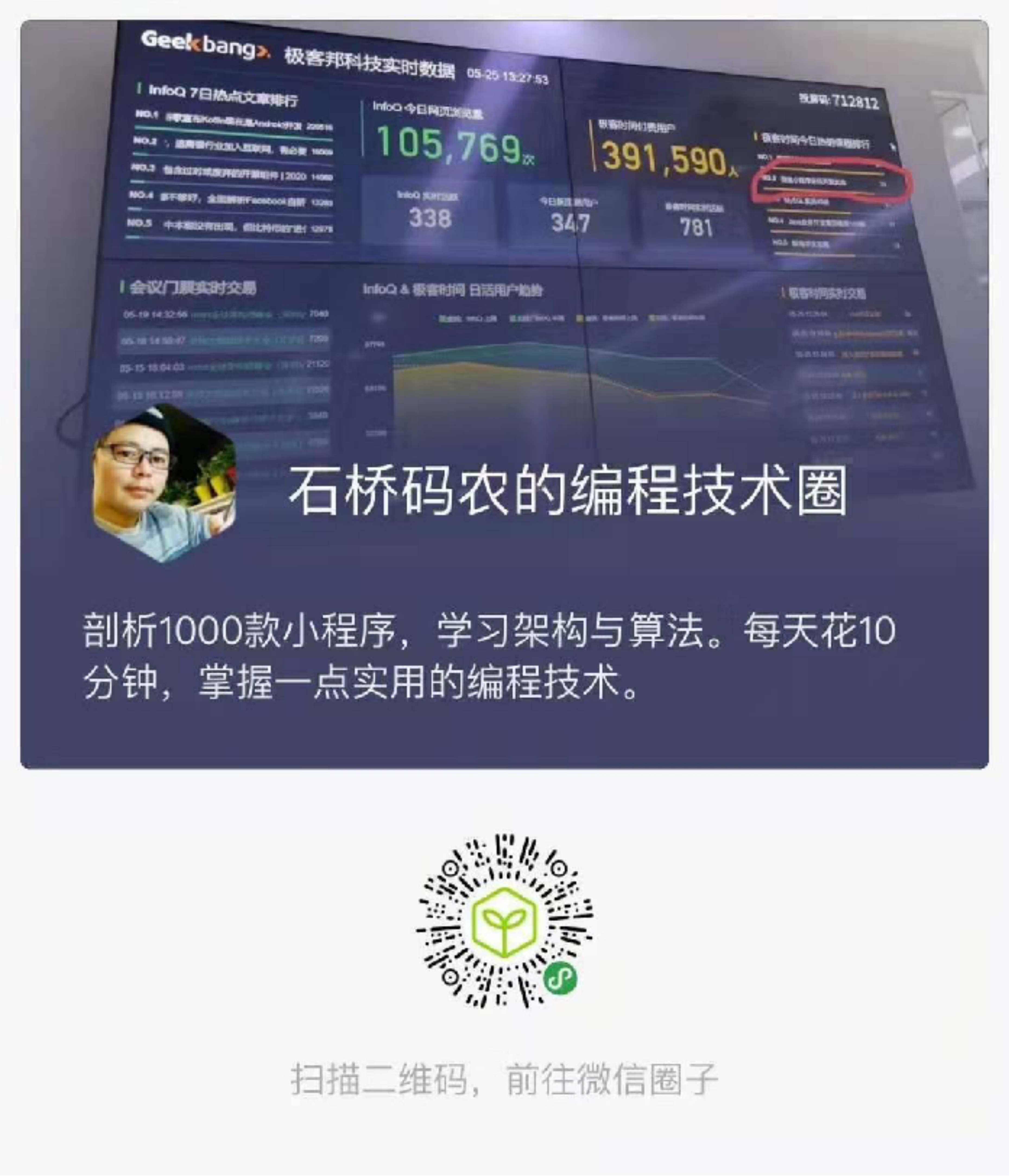
管理

公众号 | 石桥码农 >

中国联通 02:06 84%

石桥码农的编程技术圈

公告 圈子公告：欢迎每日分享、提问 >

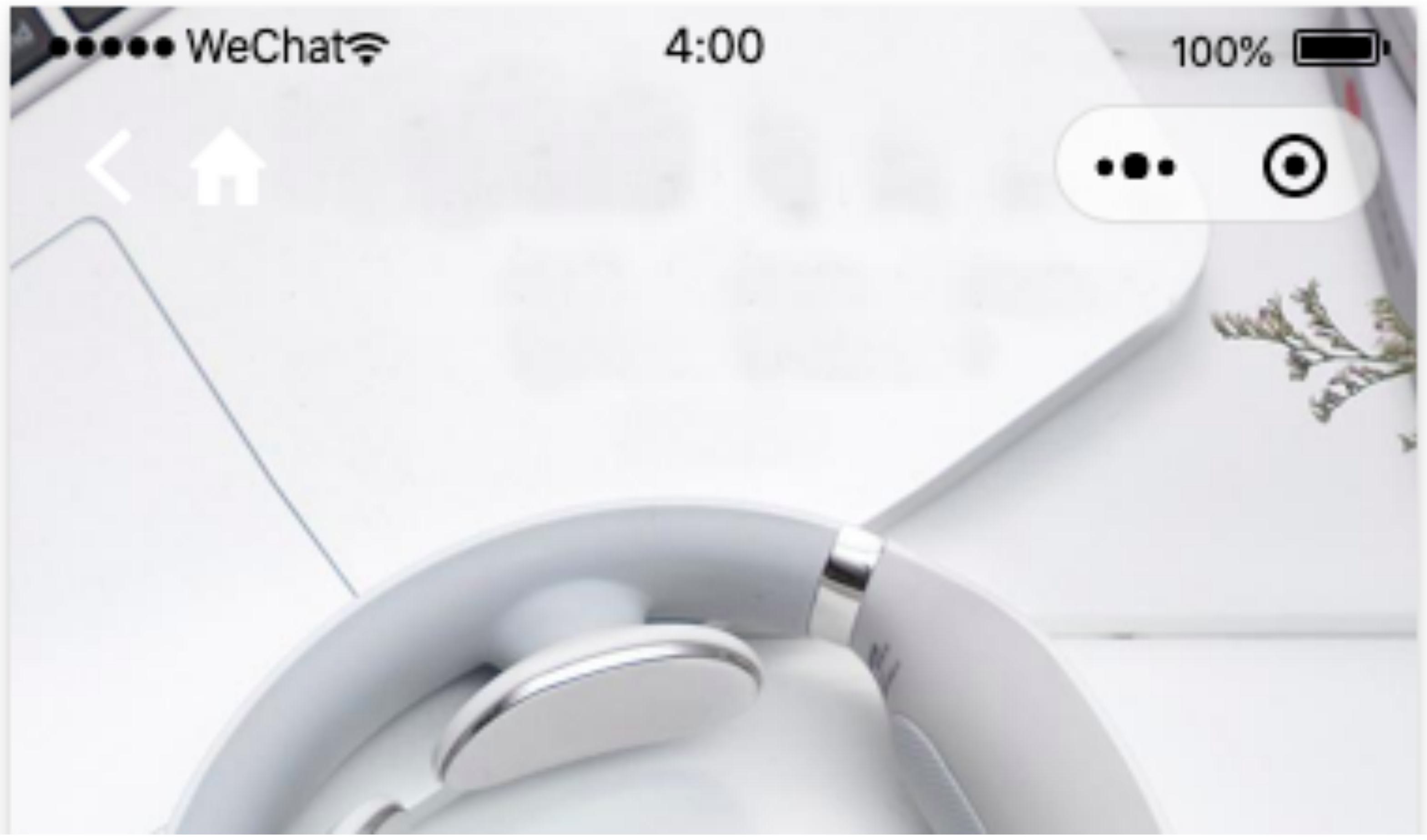


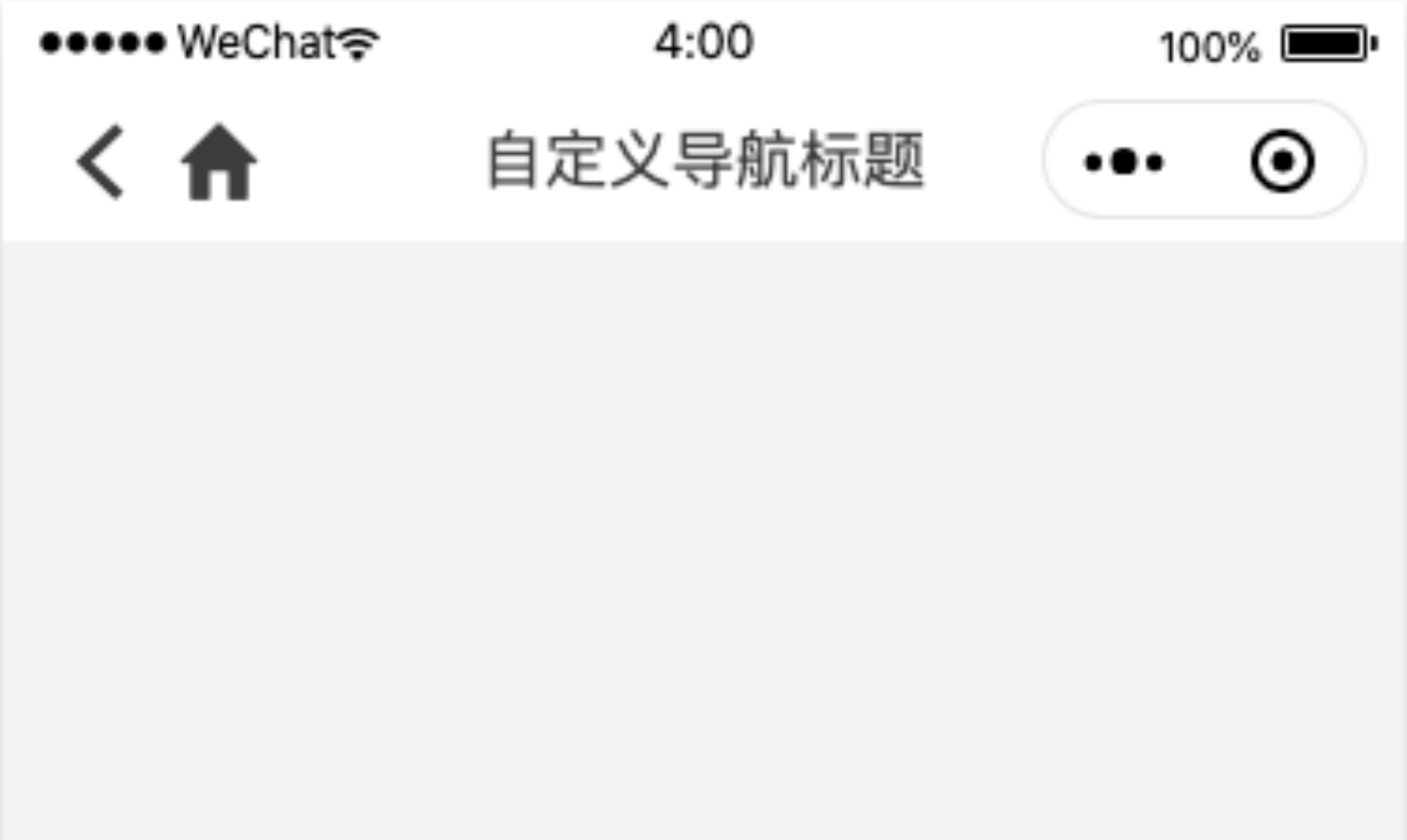
石桥码农的编程技术圈

剖析1000款小程序，学习架构与算法。每天花10分钟，掌握一点实用的编程技术。

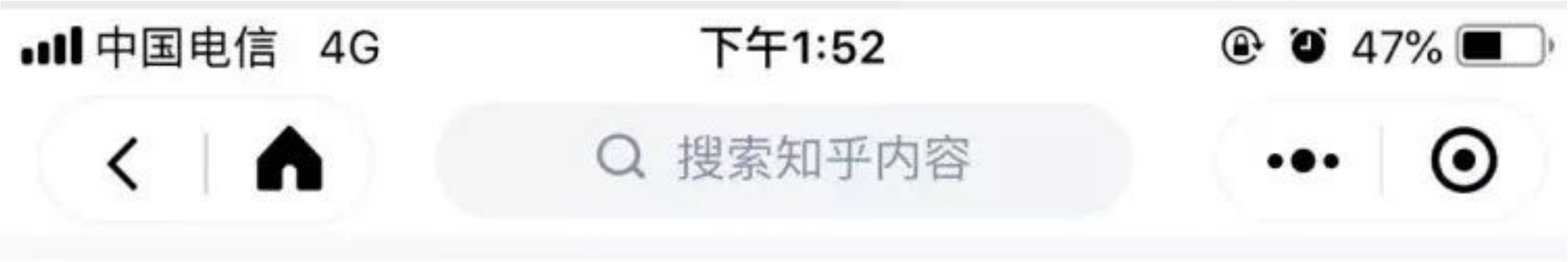
扫描二维码，前往微信圈子

```
{
"navigationStyle": "custom"
}
```


●●●●● WeChat
4:00
100%


●●●●● WeChat
4:00
100%
自定义导航标题

```
<navigation-bar
ext-class="page-navigator-bar"
active="{{active}}"
loading="{{loading}}">
<view class="left" slot="left">
<icon bindtap="goBack" class="iconfont icon-back"></icon>
<icon bindtap="goHome" class="iconfont icon-home"></icon>
</view>
<view slot="center">
<view>自定义导航标题</view>
</view>
</navigation-bar>
```



```
miniprogram/pages/2.13/index.js:
onPageScroll(res) {
  if (res.scrollTop > 400) {
    if (!this.data.active) {
      this.setData({
        active: true
      })
    }
  } else {
    if (this.data.active) {
      this.setData({
        active: false
      })
    }
  }
}
miniprogram/components/navigation-bar/index.wxml:
<view class="{{active ? 'navigator-active' : 'navigator-normal'}} ...>
```

```
properties: {
  'ext-class': {
    type: String,
    value: ''
  },
  ...
}
<view class="weui-navigation-bar {{ext-class}}"
bindtouchstart="doubleClick">
miniprogram/components/navigation-bar/index.json:
{
  "styleIsolation": "apply-shared"
}
```

1. isolated

2. apply-shared

3. shared

miniprogram/pages/2.13/index.wxml:

```
<navigation-bar
  ext-class="page-navigator-bar".../>
```

miniprogram/pages/2.13/index.wxss:

```
.page-navigator-bar .navigator-normal .icon-back{
  color: white;
}
.page-navigator-bar .navigator-active .icon-back{
  color: black;
}
```

```
miniprogram/components/navigation-bar/index.wxss
page {
  --height: 44px;
  --right: 190rpx;
}
.weui-navigation-bar .android {
  --height: 48px;
  --right: 222rpx
}
.weui-navigation-bar__inner {
  height: var(--height);
  padding-right: var(--right);
  width: calc(100% - var(--right))
}
```

```
page{
  --base-size-200: 200px;
  --base-size-10: 10rpx;
}
div{
  inline-size: calc(var(--base-size-200) + 10px);
  inline-size: calc(var(--base-size-200) - var(--base-size-10));
  inline-size: calc(var(--base-size-200) * 2);
  inline-size: calc(var(--base-size-200) / 2);
}
```

```
attached: function attached() {
 wx.getSystemInfo({
   success: function success(res) {
    var ios = !!(res.system.toLowerCase().search('ios') + 1);
    var statusBarHeight=res.statusBarHeight;
    var topBarHeight=ios ? (44 + statusBarHeight) : (48 +
statusBarHeight);
    ...
   }
 })
<view class="weui-navigation-bar__inner {{ios ? 'ios' : 'android'}}" ...></
view>
```