

## 2.1 icon 组件介绍及如何自定义实现图标？



扫码试看/订阅

《微信小程序全栈开发实战》视频课程

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

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

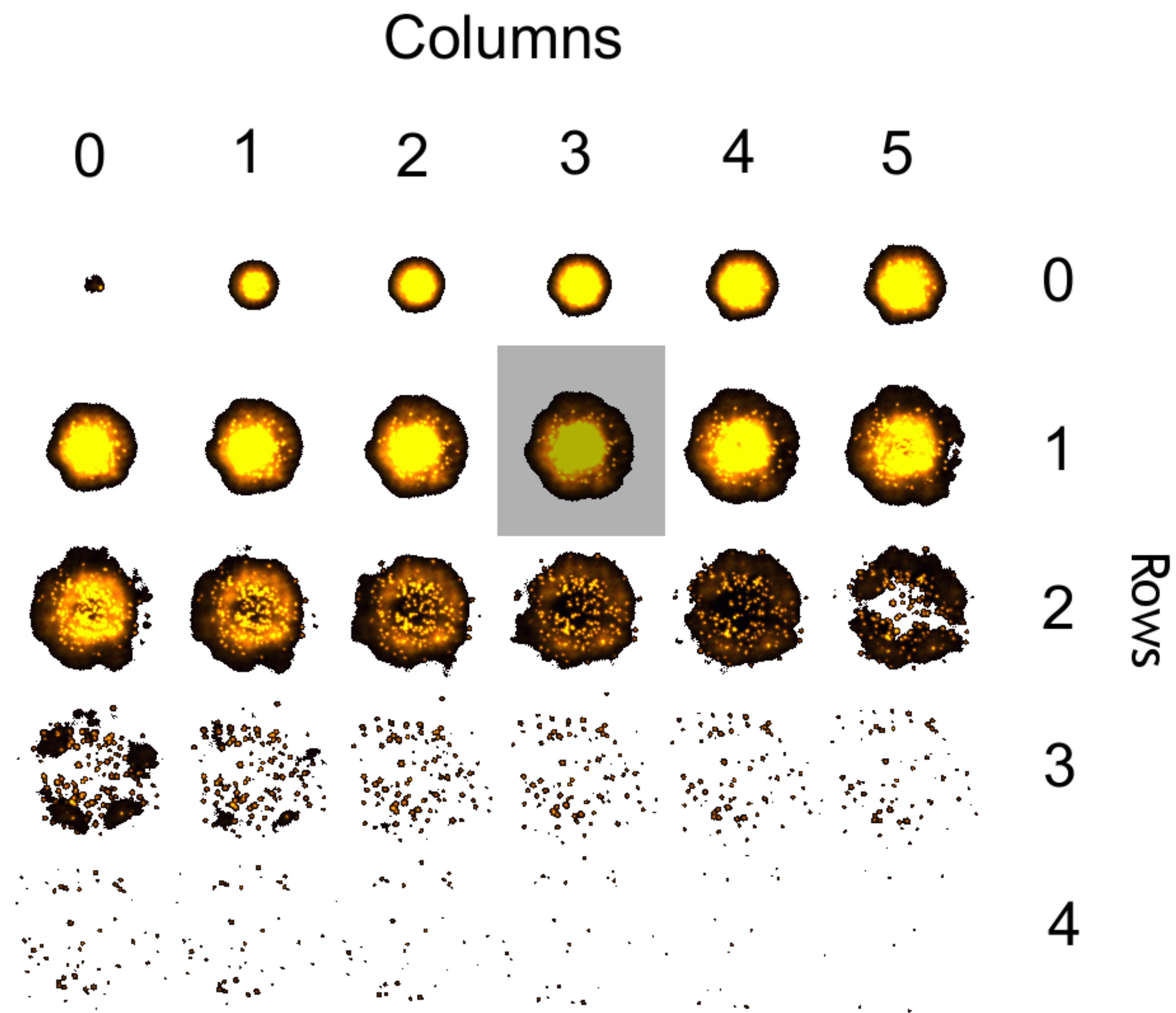
type类型范围: 'success', 'success\_no\_circle', 'info', 'warn',  
'waiting', 'cancel', 'download', 'search', 'clear'

```
<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 图标有哪些方案，原理是什么？

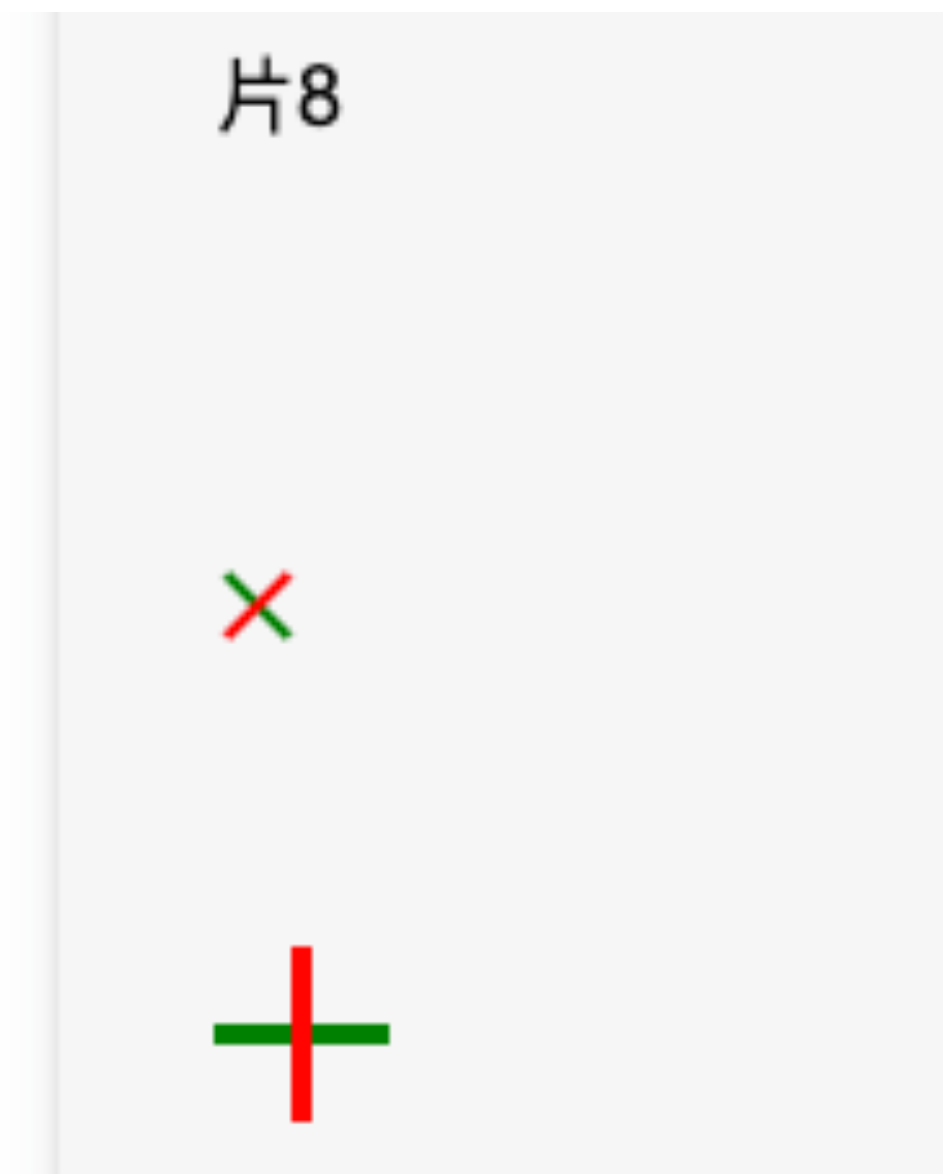


```
.sprite_icon {  
  display: block;  
  width: 80px;  
  height: 80px;  
  /* 此处.wxss中，可以使用网络图片，不能使用本地图片 */  
  background: url("https://cdn.nlark.com/..1bd0.png") -180px -310px;  
}
```





```
/* 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](http://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](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](https://developers.weixin.qq.com/miniprogram/design/#字体)



#09BB07  
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})  
}
```

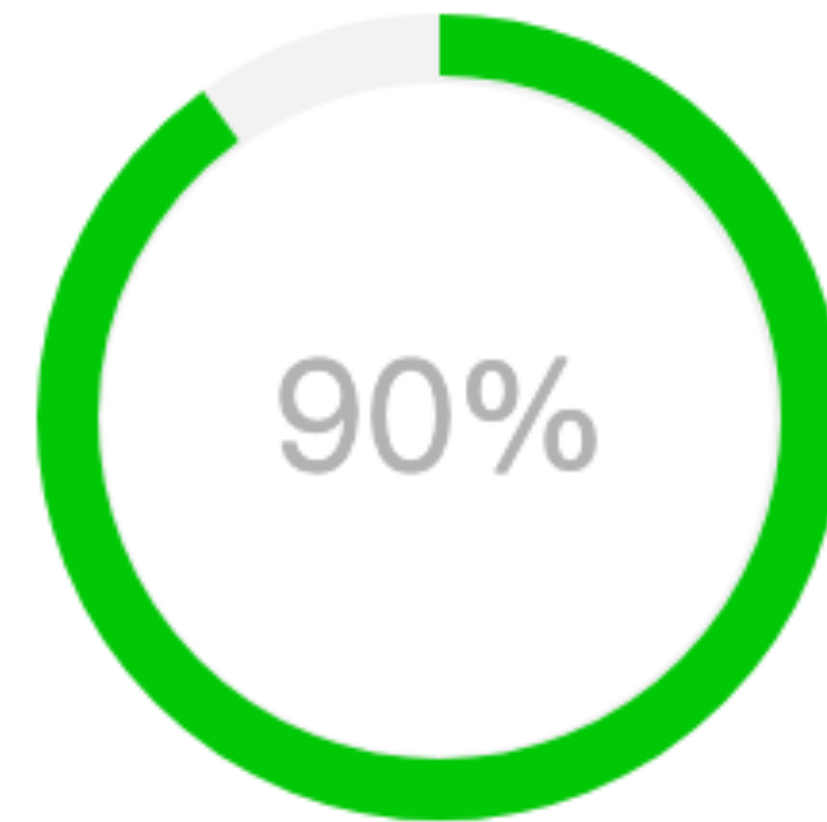
能否实现一个圆环形进度条呢？



```
<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](https://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%'
  }
}
```

```

```

<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-如意.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-如意.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']}}".../>
  ...
</block>
```

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 bindingtap="onTapImage" html="{{html}}" tag-  
style="{{tagStyle}}" />
```

```
onTapImage(e){  
  console.log('image url', e.detail.src)  
}
```

output:

image url <https://www.yishulun.com/image/篆刻-如意.png>

image url <https://www.yishulun.com/favicon.i>

[github.com/icindy/wxParse](https://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)
```

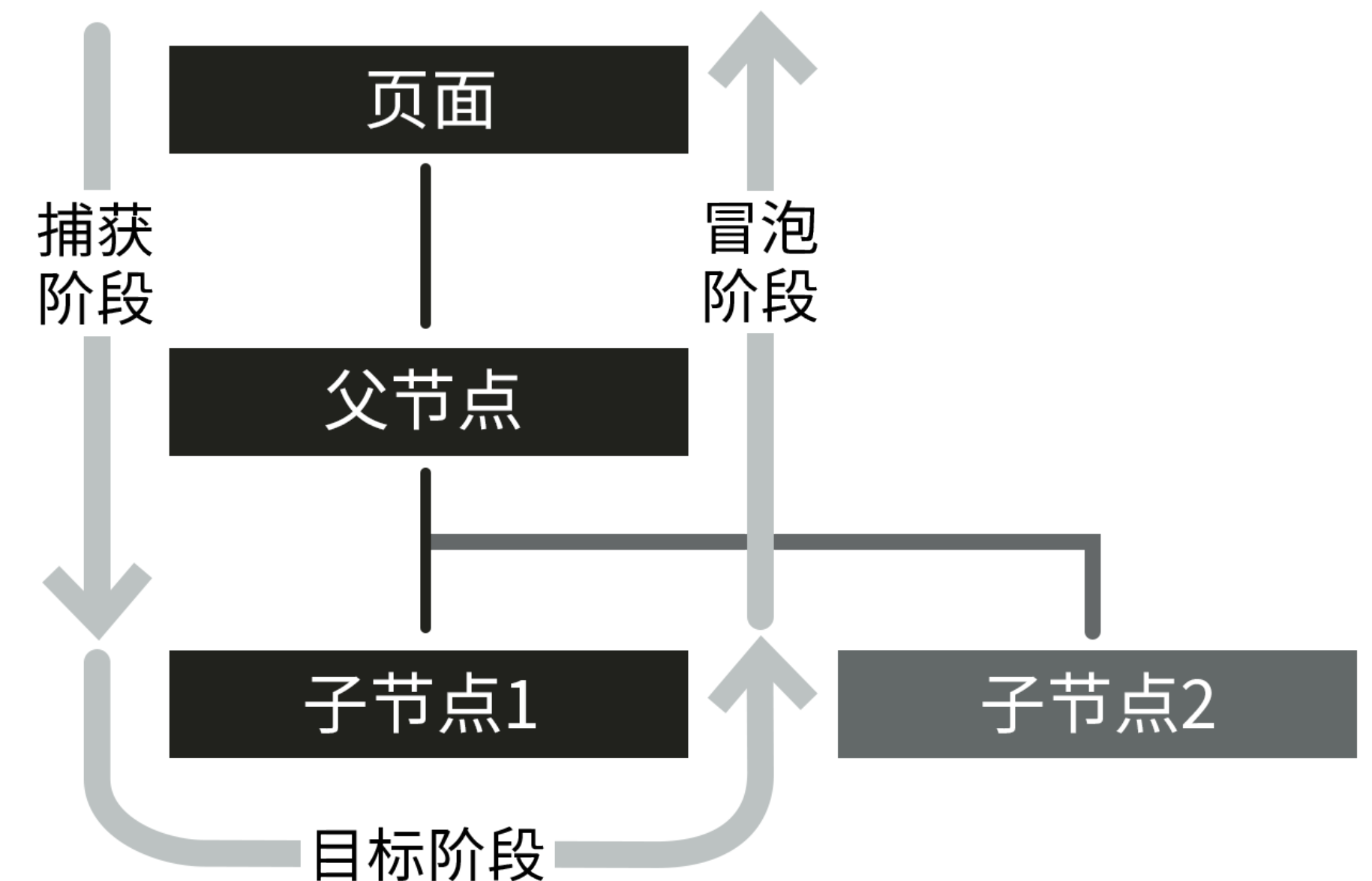
```
}
```

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

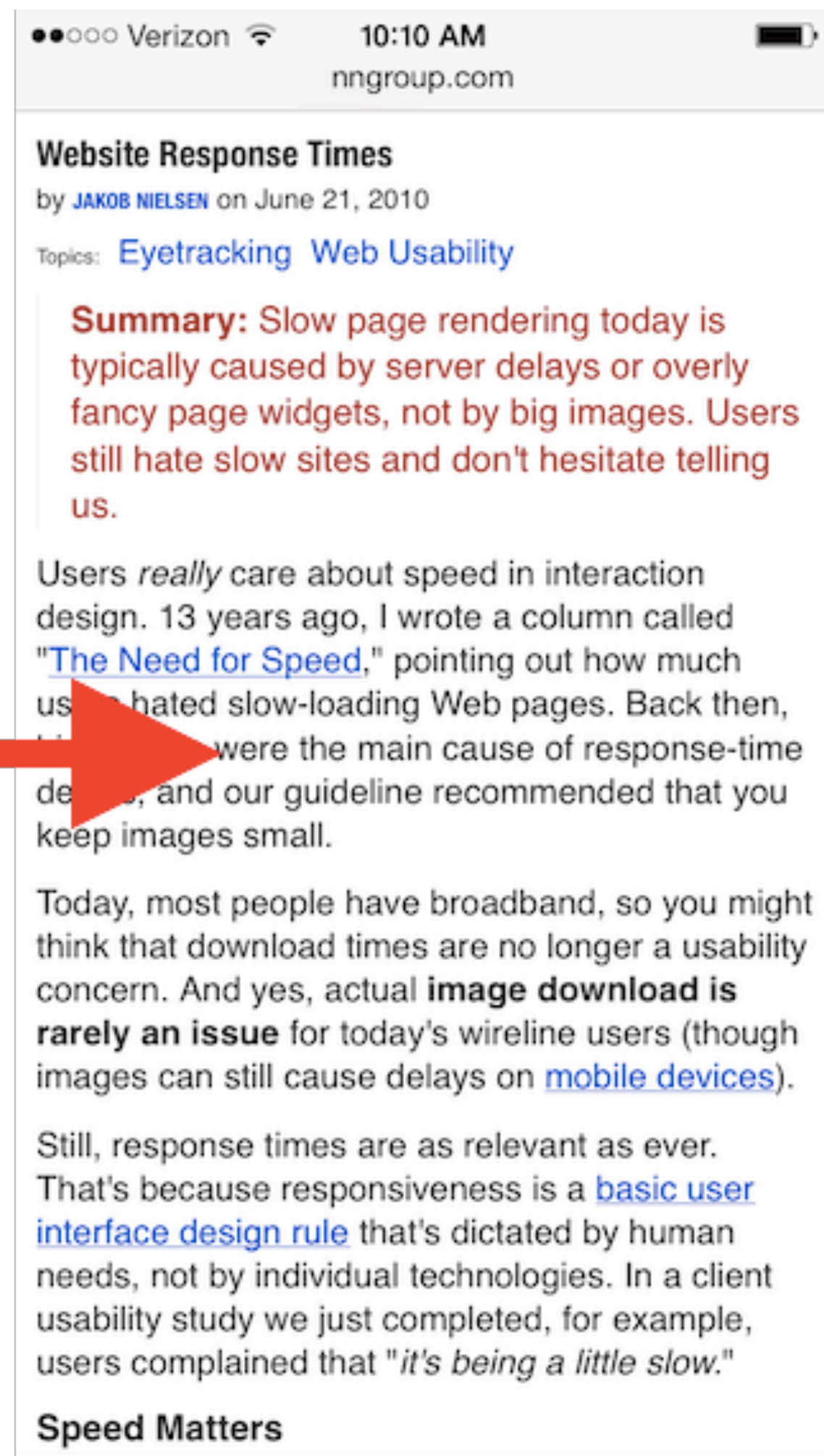
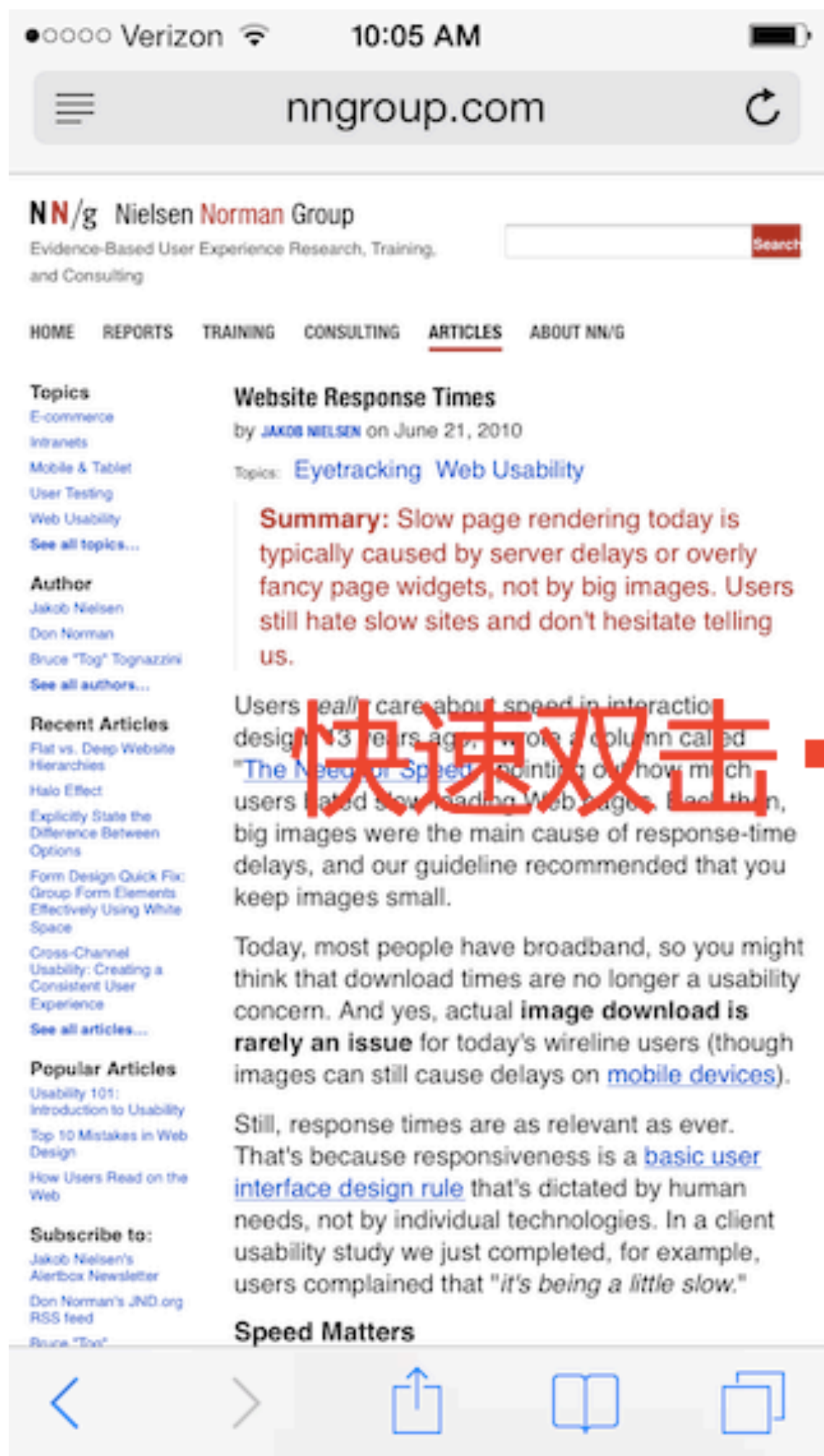
</view>

</view>

# hover-start-time、hover-stay-time



# 拒绝 300 毫秒延迟





# 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>
```

<!-- 矩形按钮 -->

```
<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>
```

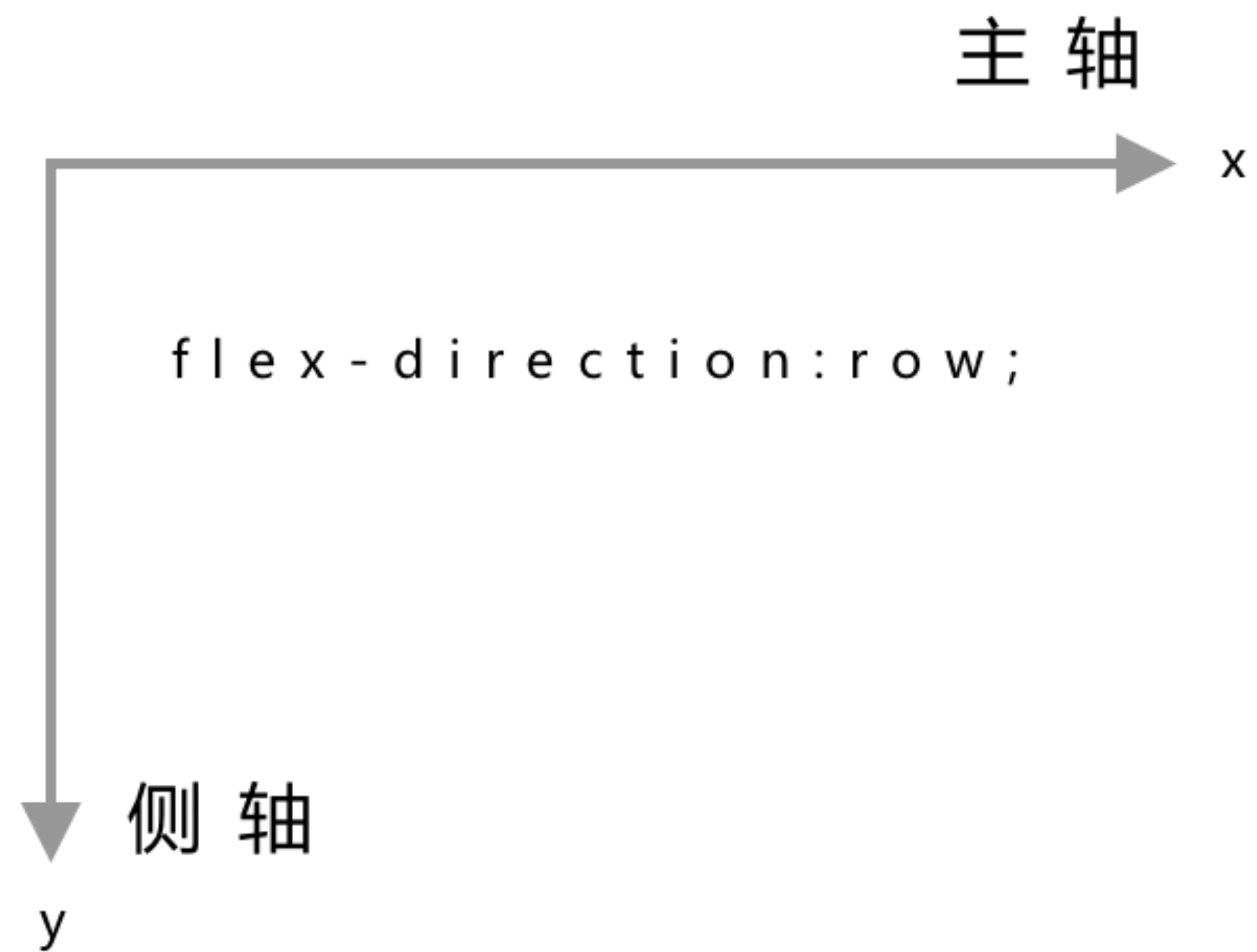
```
</view>
```

```
.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）

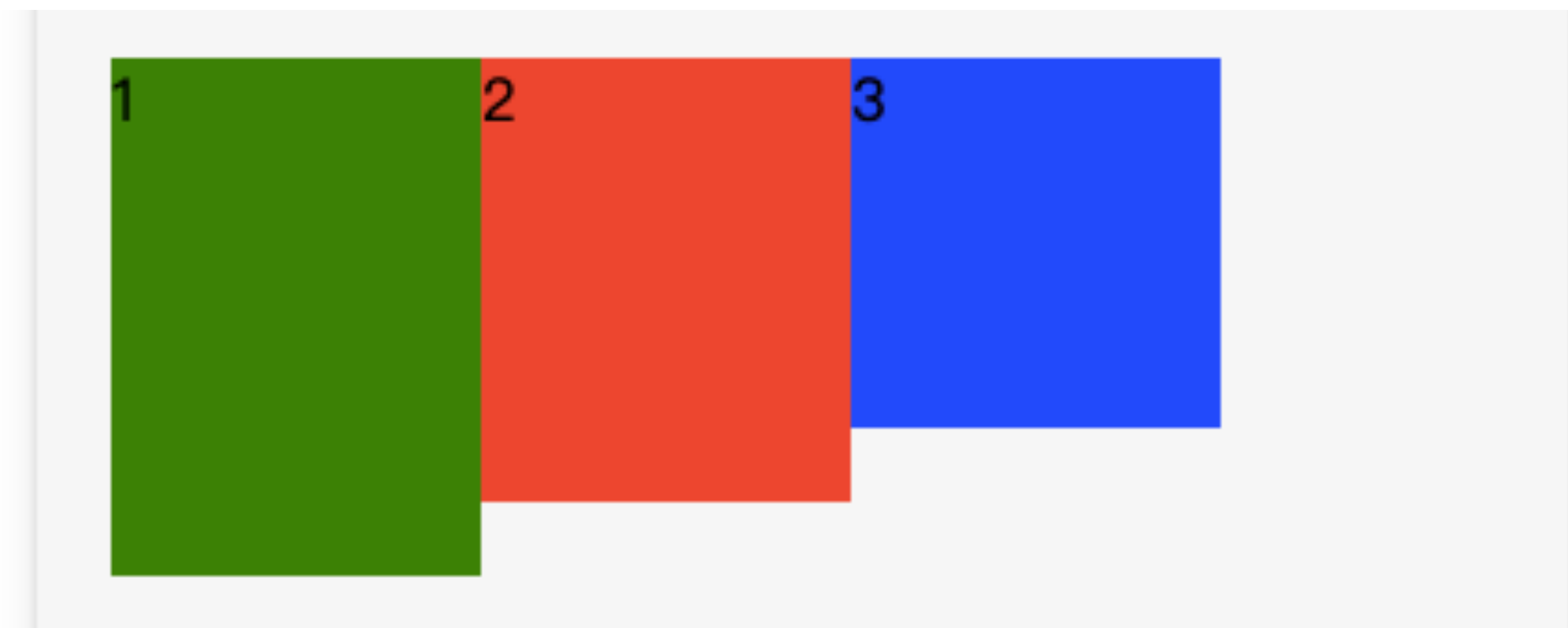


justify-content



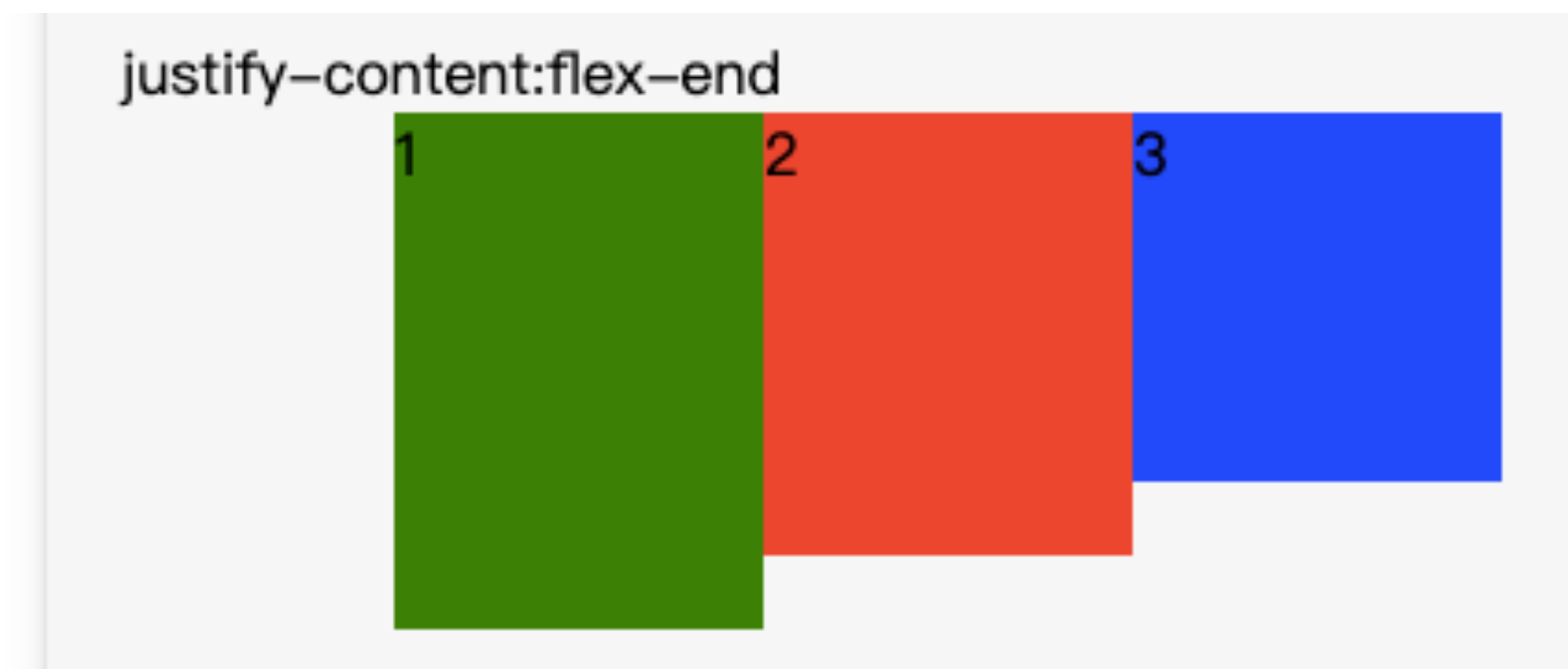
# 第一个值：flex-start

```
<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>
```



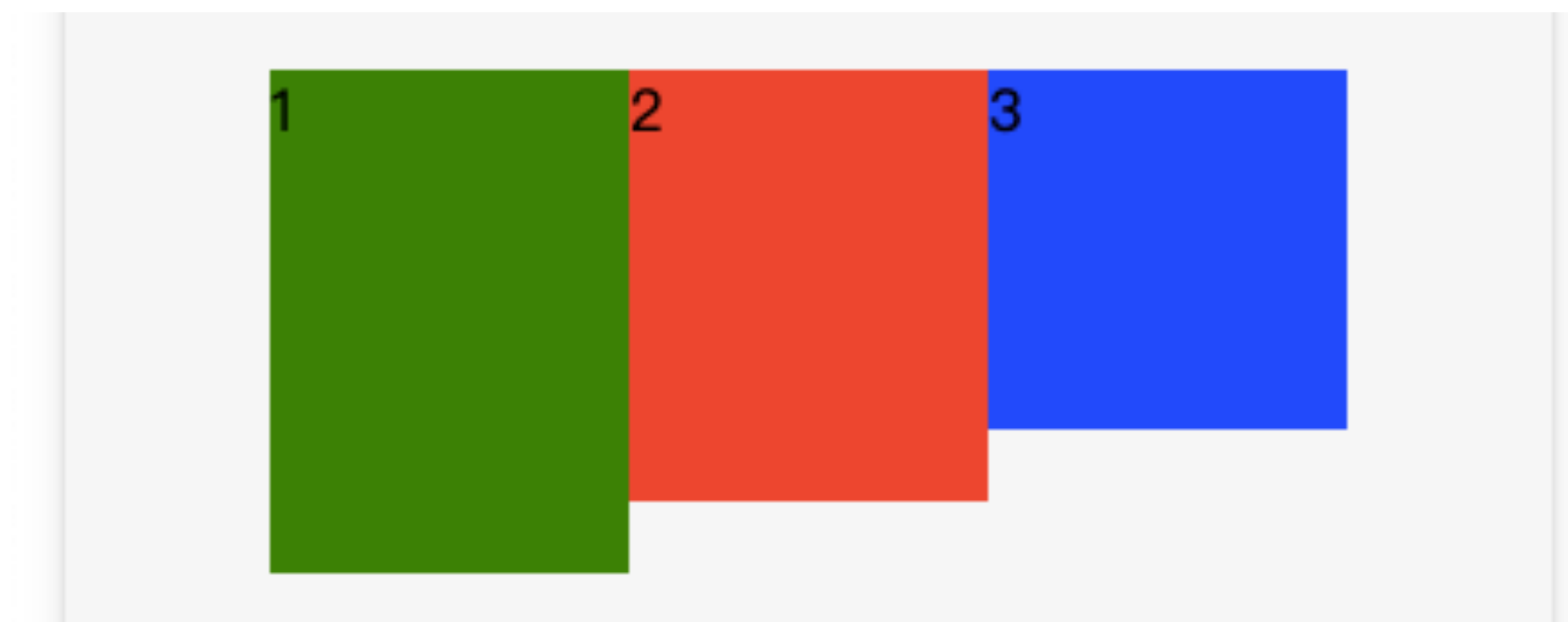
## 第二个值：flex-end

```
<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>
```



## 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>
```



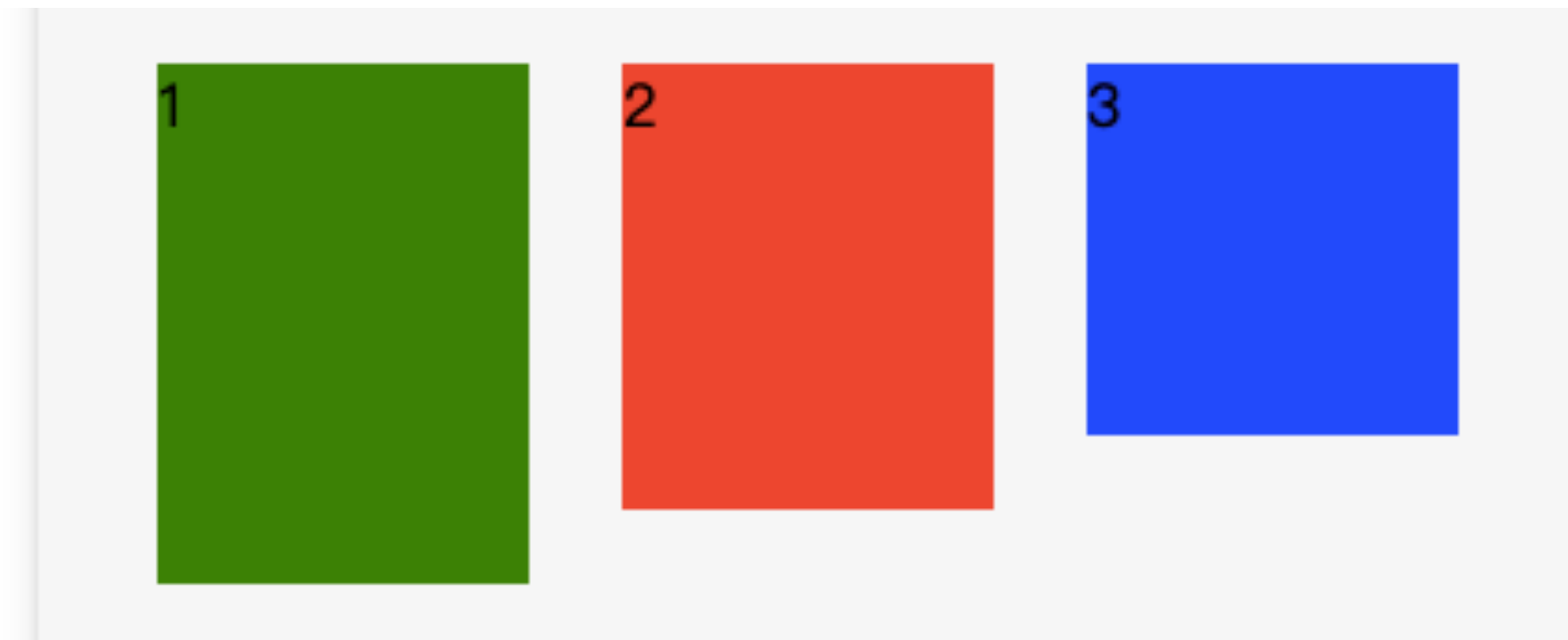
## 第四个值： space-between

```
<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

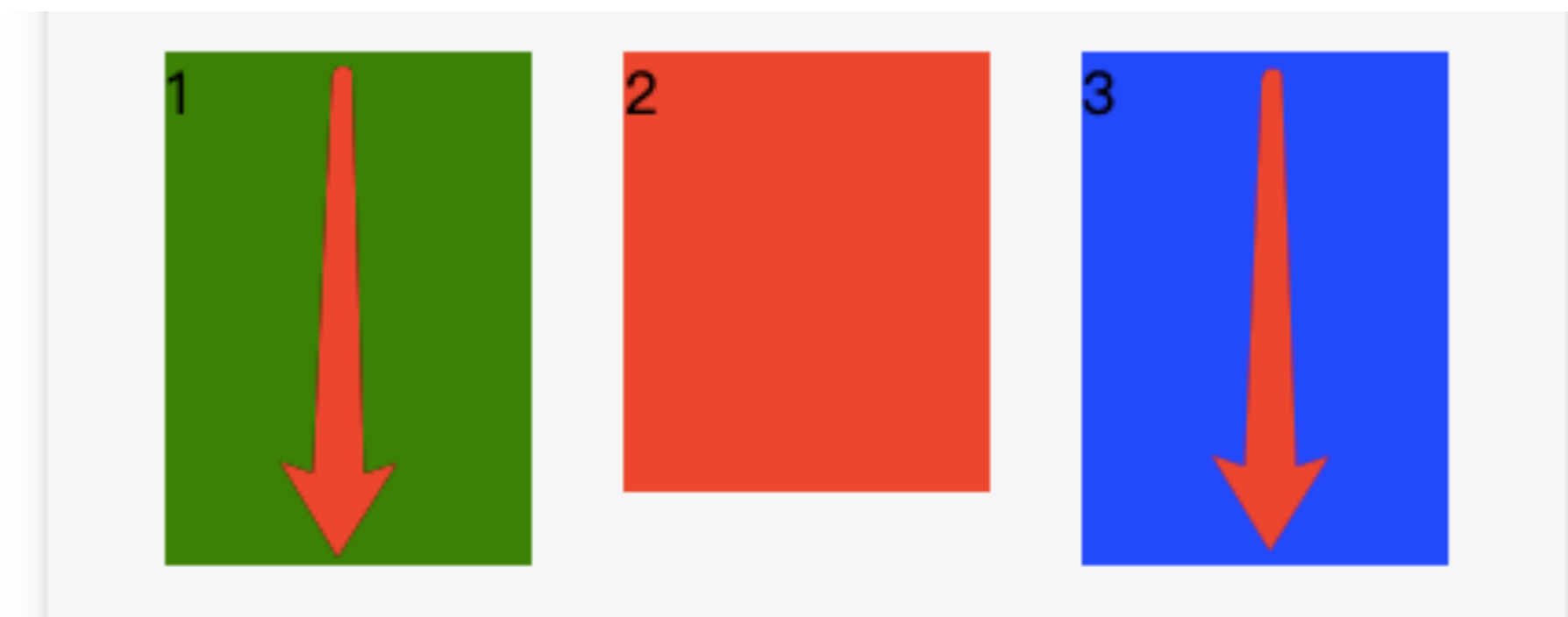
```
<view class="section">  
<view class="section__title">8 justify-content:space-around</view>  
<view class="flex-wrp" style="flex-direction:row;justify-content:space-around">  
<view class="flex-item bc_green">1</view>  
<view class="flex-item bc_red">2</view>  
<view class="flex-item bc_blue">3</view>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



align-items

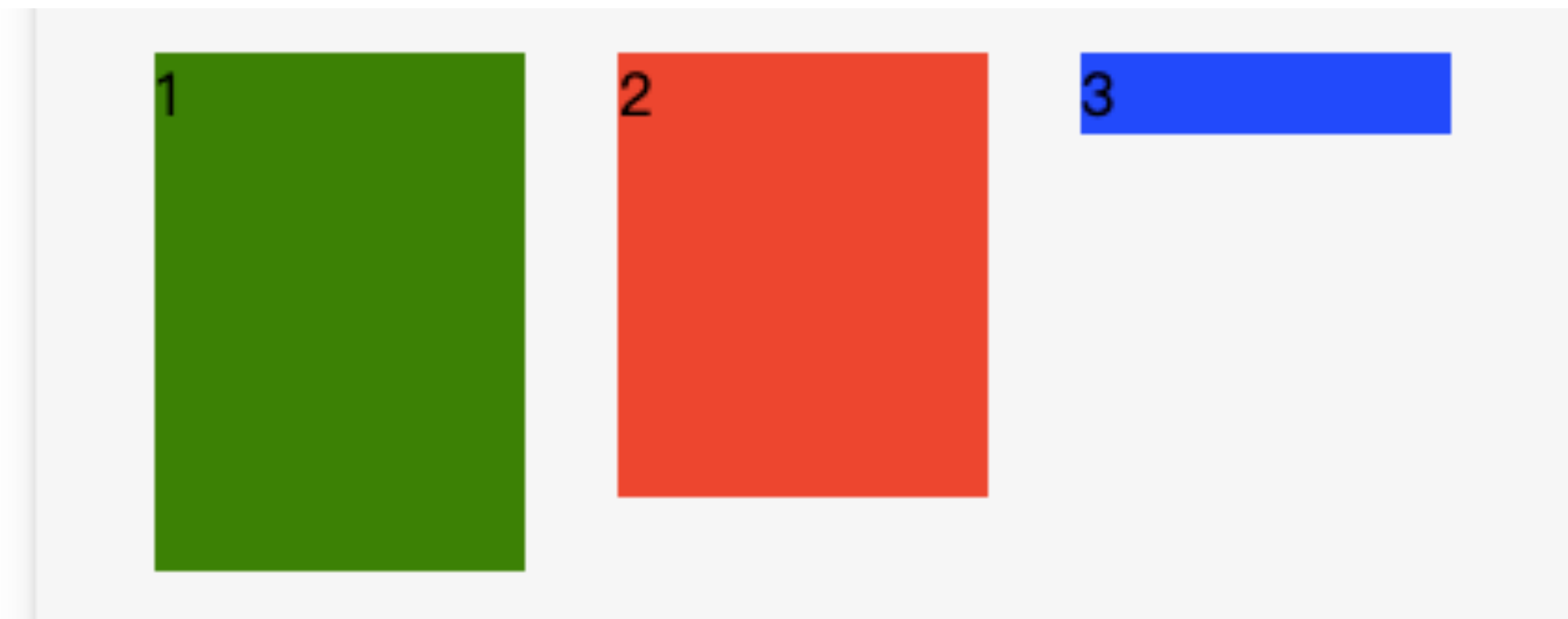
# 第一个值：stretch

```
<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

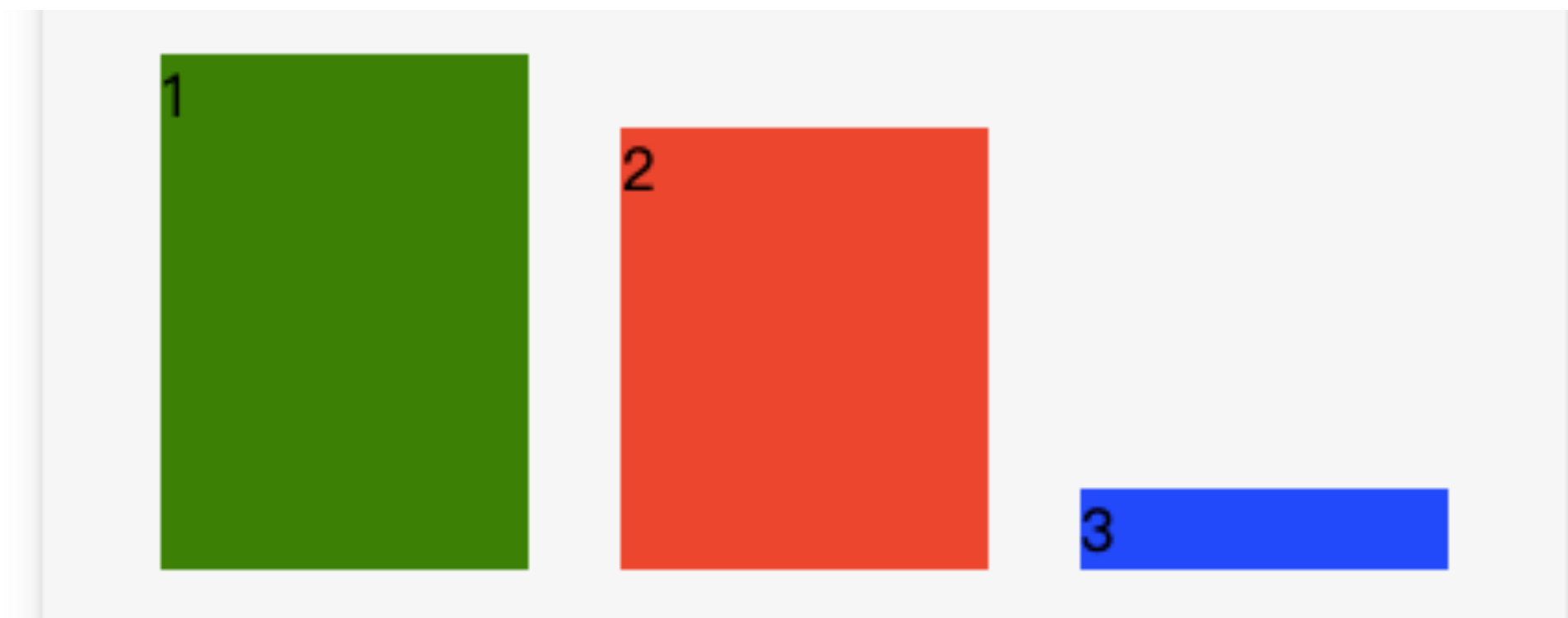
```
<view class="section">
<view class="section__title">11 align-items:flex-start</view>
<view class="flex-wrp" style="flex-direction:row;justify-content:space-around;align-
items:flex-start;">
<view class="flex-item bc_green">1</view>
<view class="flex-item bc_red">2</view>
<view style="height:auto;"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





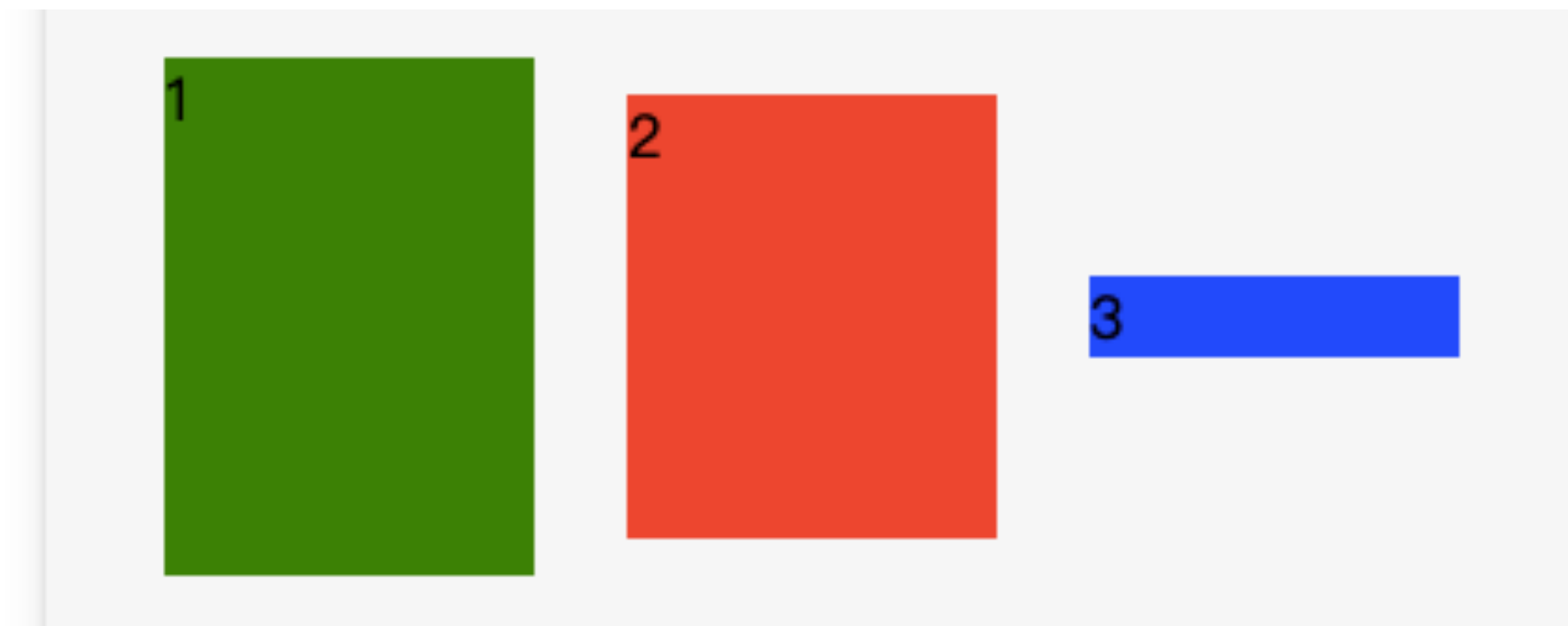
## 第三个值：flex-end

```
<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>
```



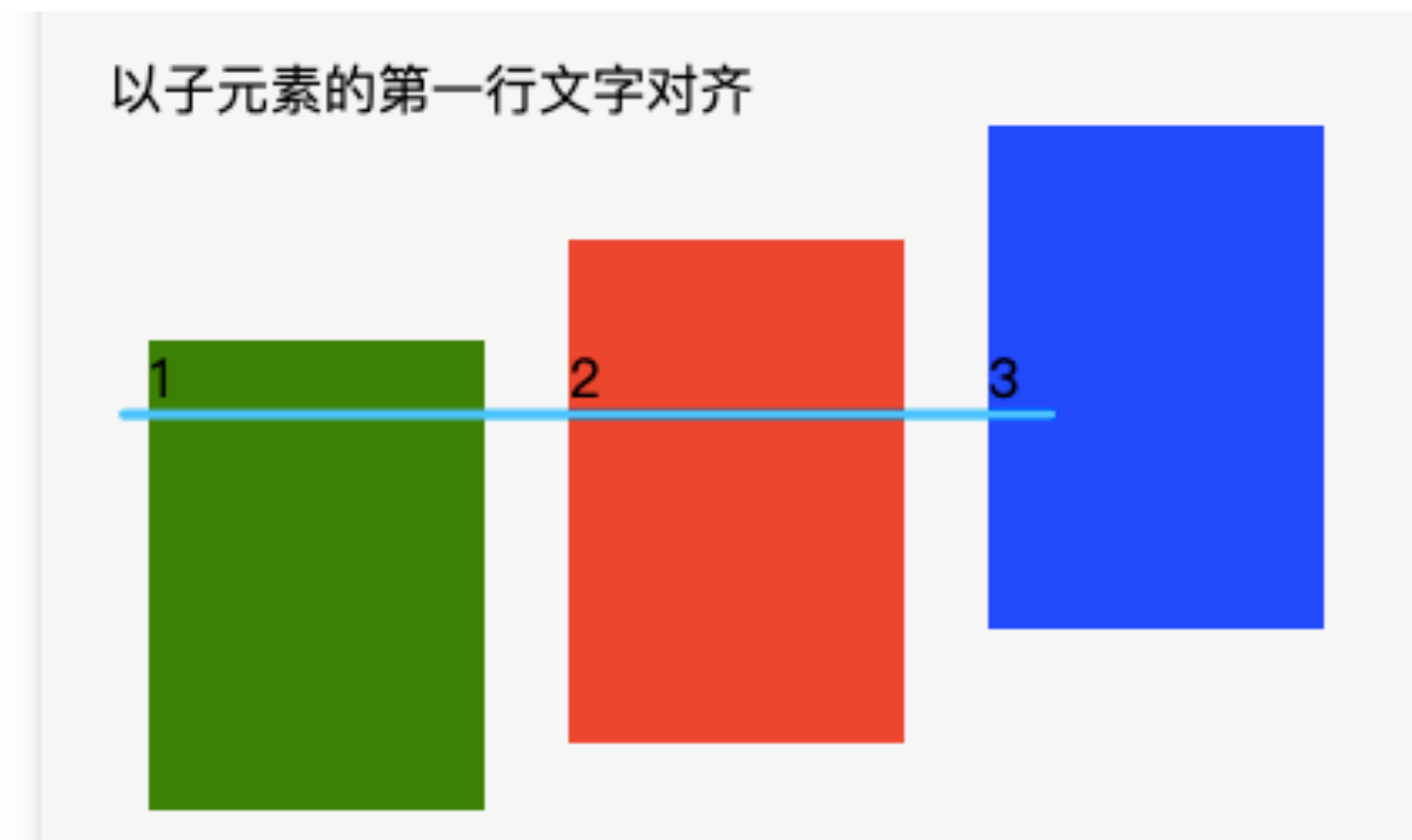
## 第四个值：center

```
<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>
```



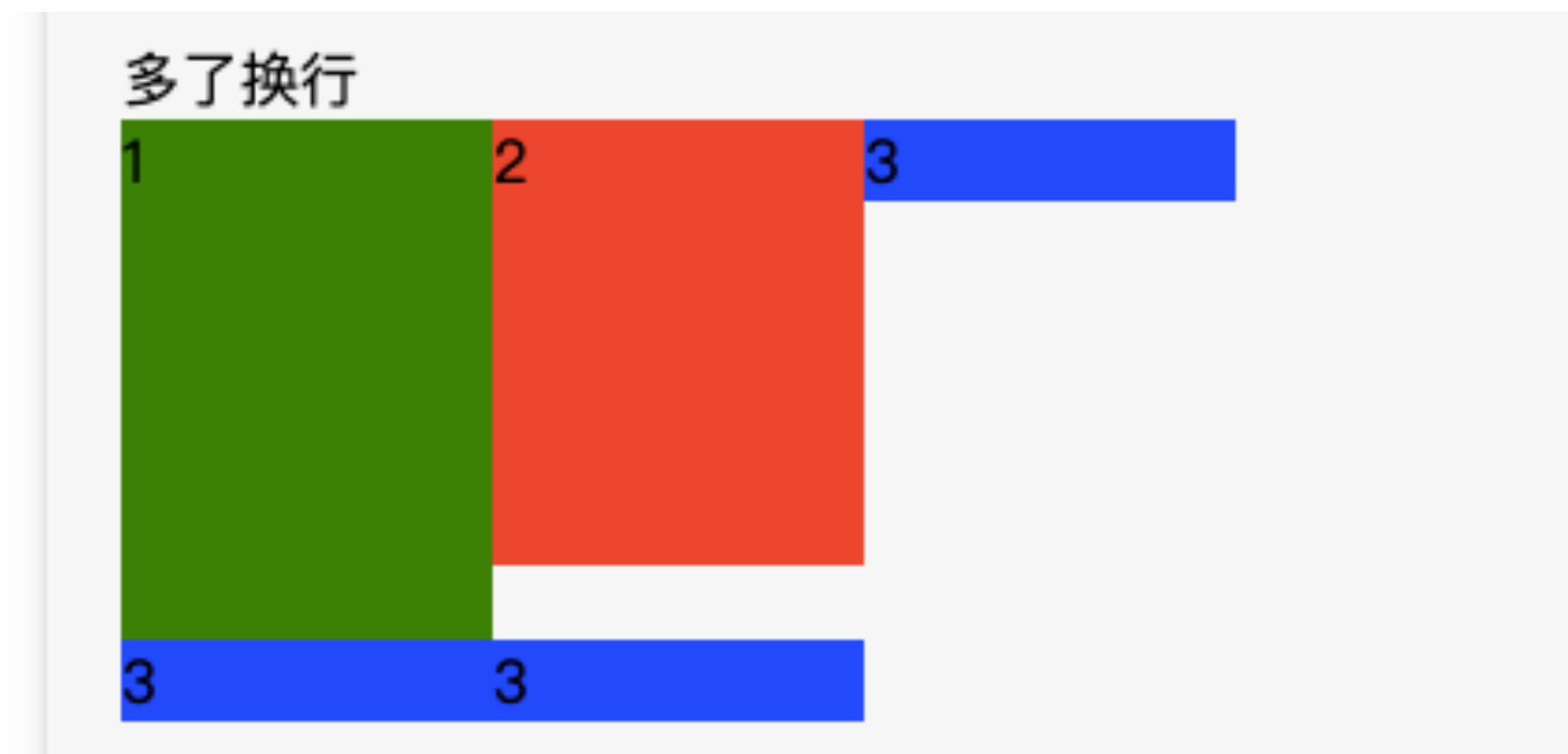
## 第五个值：baseline

```
<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>
```



flex-wrap 的值

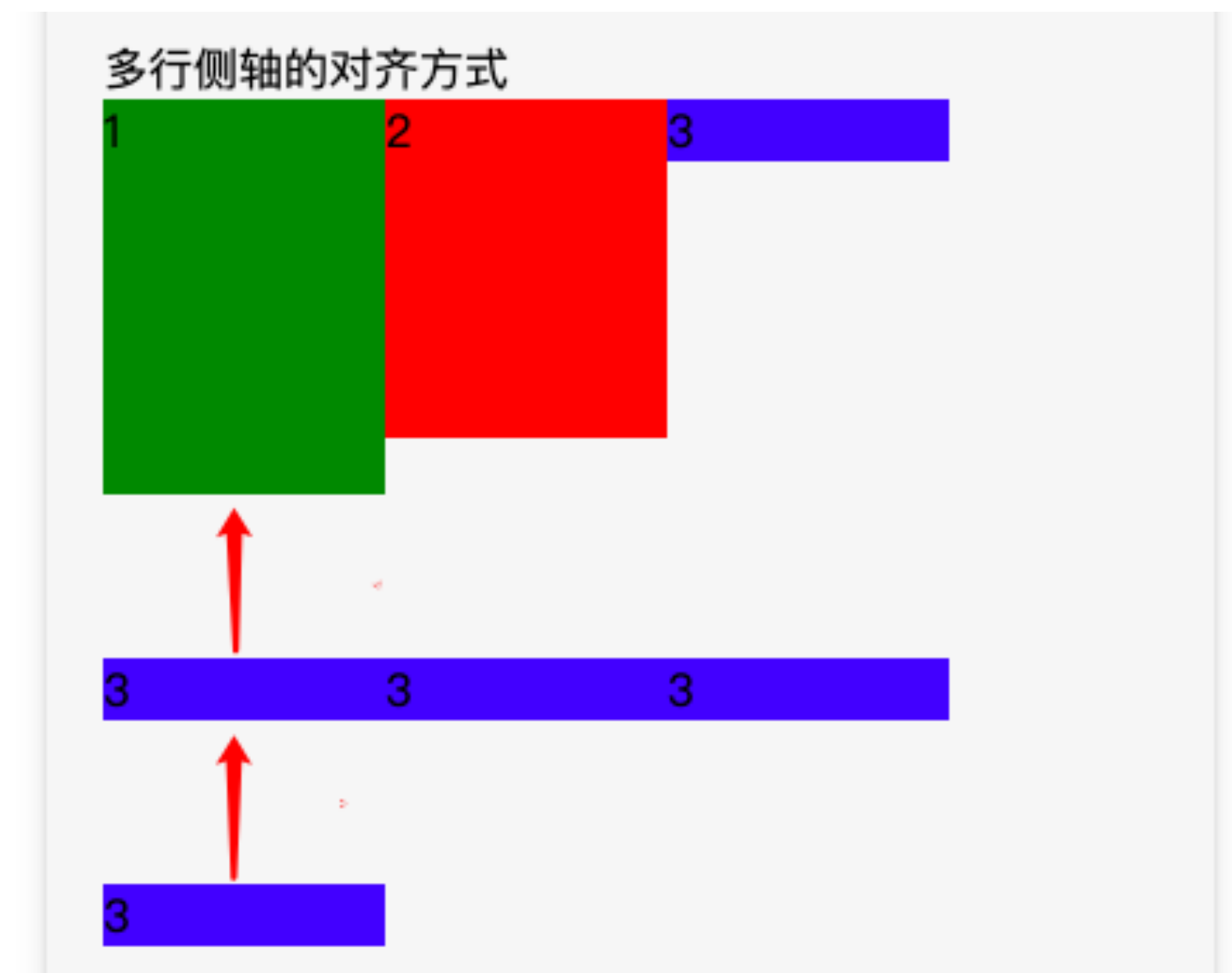
```
<view class="section">
<view class="section__title">元素多了，换行</view>
<view class="flex-wrp" style="flex-direction:row;justify-content:flex-start;align-items:baseline;flex-wrap:wrap;">
<view class="flex-item bc_green">1</view>
<view class="flex-item bc_red">2</view>
<view style="height:auto;" class="flex-item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



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>
```



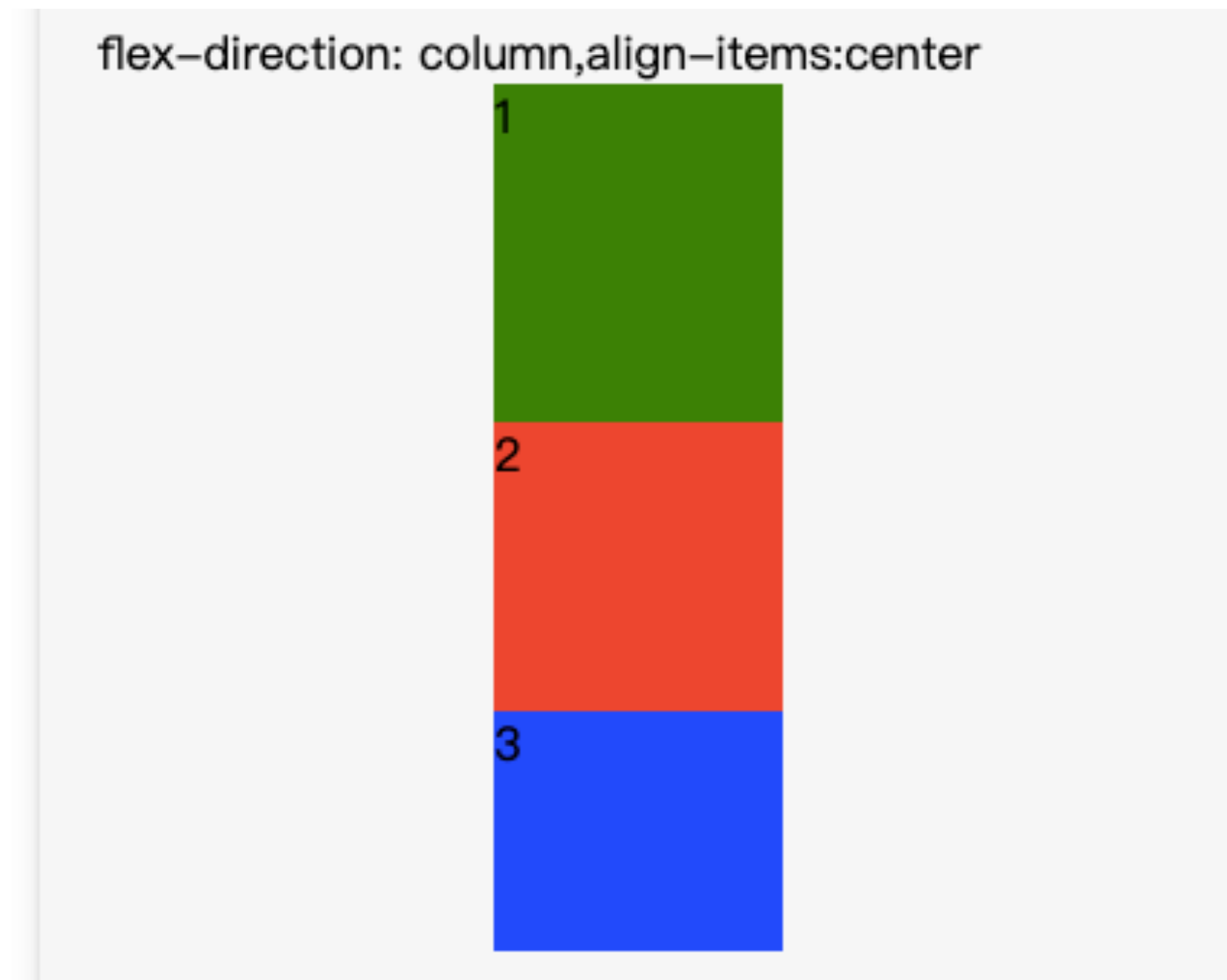
flex-direction 的值



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>
```



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>





扫码试看/订阅

《微信小程序全栈开发实战》视频课程