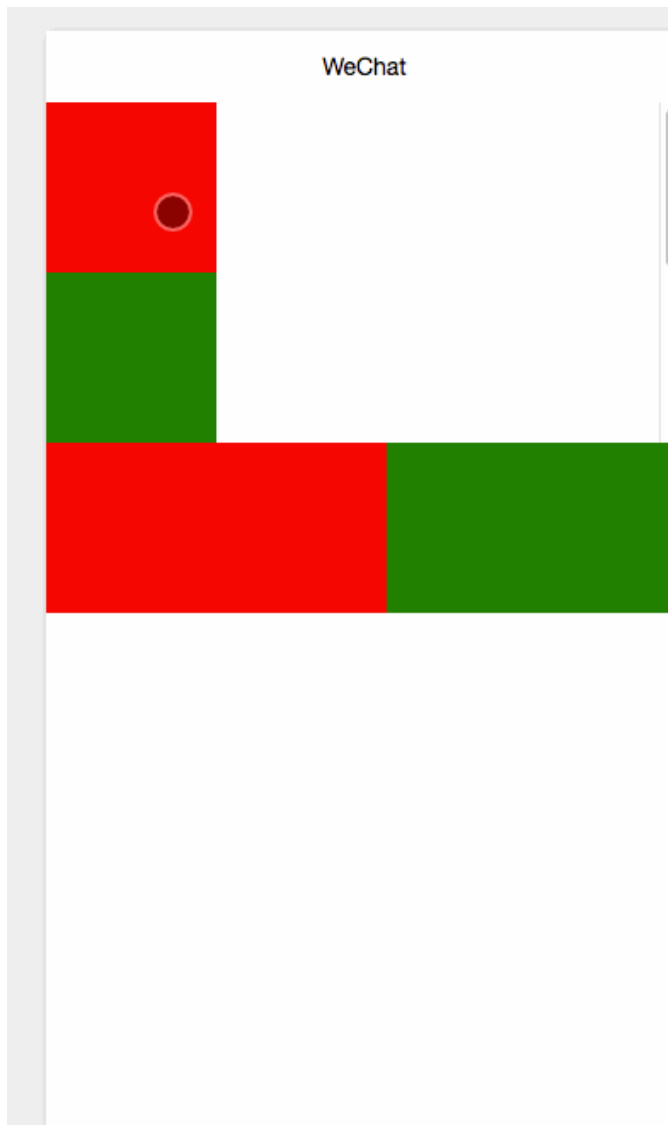


微信小程序把玩《三》：**scroll-view** 组件，**swiper** 组件，**icon** 组件

一：**scroll-view** 组件



scroll-view 为滚动视图，分为水平滚动和垂直滚动。注意滚动视图垂直滚动时一定要设置高度否则的话 **scroll-view** 不会生效。滚动视图常用的地方

一般都是 **Item** 项比较多的界面，比如我的模块

主要属性：

属性名	类型	默认值	说明
scroll-x	Boolean	false	允许横向滚动
scroll-y	Boolean	false	允许纵向滚动
upper-threshold	Number	50	距顶部/左边多远时（单位px），触发scrolltoupper事件
lower-threshold	Number	50	距底部/右边多远时（单位px），触发scrolltolower事件
scroll-top	Number		设置竖向滚动条位置
scroll-left	Number		设置横向滚动条位置
scroll-into-view	String		值应为某子元素id，则滚动到该元素，元素顶部对齐滚动区域顶部
bindscrolltoupper	EventHandle		滚动到顶部/左边，会触发scrolltoupper事件
bindscrolltolower	EventHandle		滚动到底部/右边，会触发scrolltolower事件
bindscroll	EventHandle		滚动时触发，event.detail总携带 {scrollLeft,scrollTop,scrollHeight,scrollWidth,deltaX,deltaY}

使用演示:

wxml

<!--垂直滚动, 这里必须设置高度-->

```
<scroll-view scroll-y="true" style="height: 200px">
  <view style="background: red; width: 100px; height: 100px" ></view>
  <view style="background: green; width: 100px; height: 100px"></view>
  <view style="background: blue; width: 100px; height: 100px"></view>
  <view style="background: yellow; width: 100px; height: 100px"></view>
</scroll-view>
```

<!-- white-space

normal: 正常无变化(默认处理方式.文本自动处理换行.假如抵达容器边界内容会转到下一行)

pre: 保持 HTML 源代码的空格与换行,等同与 pre 标签

nowrap: 强制文本在一行,除非遇到 br 换行标签

pre-wrap: 同 pre 属性,但是遇到超出容器范围的时候会自动换行

pre-line: 同 pre 属性,但是遇到连续空格会被看作一个空格

inherit: 继承

-->

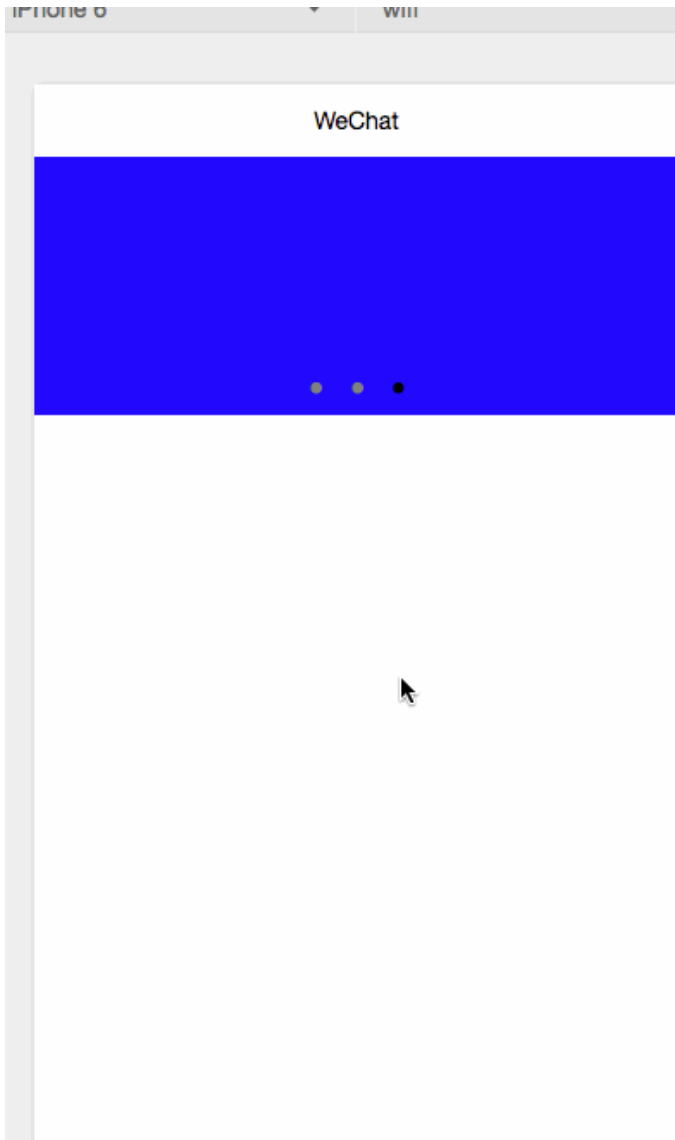
<!--水平滚动-->

```
<scroll-view scroll-x="true" style="white-space: nowrap; display: flex" >
```

<!-- display: inline-block-->

```
  <view style="background: red; width: 200px; height: 100px; display: inline-block" ></view>
  <view style="background: green; width: 200px; height: 100px; display: inline-block"></view>
  <view style="background: blue; width: 200px; height: 100px; display: inline-block"></view>
  <view style="background: yellow; width: 200px; height: 100px; display: inline-block"></view>
</scroll-view>
```

二: swiper 组件



Android 写过轮播图的痛楚只有写过的知道，相对还是比较麻烦的，并没有一个轮播图组件，有个 `ViewPager` 也需要自己定制，**iOS** 则多用 `UIScrollerView` 去实现，这个 `swiper` 封装的相对还是方便的，使用方式也相对那俩容易些。

主要属性：

属性名	类型	默认值	说明
indicator-dots	Boolean	false	是否显示面板指示点
autoplay	Boolean	false	是否自动切换
current	Number	0	当前所在页面的index
interval	Number	5000	自动切换时间间隔
duration	Number	1000	滑动动画时长
bindchange	EventHandle		current改变时会触发change事件，event.detail={current:current}

属性只需要设置就行了 也可以抽到 js 文件的 data 中进行数据绑定，监听使用 bindchange，在 js 中做业务处理。

wxml

```
<!--是否显示圆点， 自动播放， 间隔时间， 监听滚动和点击事件-->
<swiper indicator-dots="true" autoplay="true" duration="1000" bindchange="listenSwiper" >
<!--swiper-item 只能包含一个节点再多会自动删除-->
  <swiper-item>
    <view style="background: red; height: 150px"></view>
  </swiper-item>
  <swiper-item>
    <view style="background: green; height: 150px"></view>
  </swiper-item>
  <swiper-item>
    <view style="background: blue; height: 150px"></view>
```

```
</swiper-item>
```

```
</swiper>
```

```
js:
```

```
Page({
```

```
  data:{
```

```
    // text:"这是一个页面"
```

```
  },
```

```
/**
```

```
 * 这里处理滚动事件处理
```

```
*/
```

```
  listenSwiper:function(e) {
```

```
    //打印信息
```

```
    console.log(e)
```

```
  },
```

```
  onLoad:function(options){
```

```
    // 页面初始化 options 为页面跳转所带来的参数
```

```
  },
```

```
  onReady:function(){
```

```
    // 页面渲染完成
```

```
  },
```

```
  onShow:function(){
```

```
    // 页面显示
```

```
  },
```

```
  onHide:function(){
```

```
    // 页面隐藏
```

```
  },
```

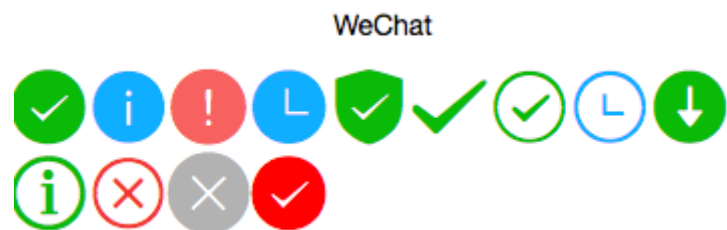
```
  onUnload:function(){
```

```
    // 页面关闭
```

```
  }
```

```
}}
```

三: icon 组件



这些是提供的所支持的图标样式，根据需求在此基础上去修改大小和颜色。

主要属性:

属性名	类型	默认值	说明
type	String		icon的类型，有效值： success,success_no_circle,info,warn,waiting,cancel,download,search,clear
size	Number	23	icon的大小，单位px
color	Color		icon的颜色，同css的color

使用方式：

wxml

<!--成功图标-->

<icon type="success" size="40"/>

<!--安全成功标志图标-->

<icon type="safe_success" size="40"/>

<!--提示信息图标-->

<icon type="info" size="40"/>

<!--带圆的信息提示图标-->

<icon type="info_circle" size="40"/>

<!--不带圆的成功图标-->

<icon type="success_no_circle" size="40"/>

<!--带圆的成功图标-->

<icon type="success_circle" size="40"/>

<!--警告图标-->

<icon type="warn" size="40"/>

<!--带圆的等待图标-->

<icon type="waiting_circle" size="40"/>

<!--等待图标-->

<icon type="waiting" size="40"/>

<!--下载图标-->

<icon type="download" size="40"/>

<!--取消图标-->

<icon type="cancel" size="40"/>

<!--清除图标-->

<icon type="clear" size="40"/>

<!--改变颜色的 success-->

<icon type="success" size="40" color="red"/>