# 图片加载

## 静态图片资源

RN提供了统一方式管理ios和android应用中图片。

使用静态图片只需要如下引用：

<Image source={require('./state.jpg')}/>

如果需要显示不同分辨率的图片，通过在图片后缀添加@2x @3作为不同的适配图片，需要放在同一个文件夹项目。（默认的那个图片不加任何的后缀）

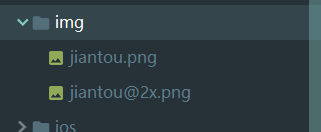


图1-1 适配图片

引用的时候只需要使用无后缀的图片，系统会自动匹配的。

<Image source={require('./img/jiantou.png')}/>

小技巧：在style中使用tintColor可以改变图片主体颜色。

注意：如果导入新的资源，正在运行的应用是收不到的，需要重启一下。

使用resizeMode设置缩放模式：

1、cover模式只求在显示比例不失真的情况下填充整个显示区域。可以对图片进行放大或者缩小，超出显示区域的部分不显示， 也就是说，图片可能部分会显示不了。（默认）

2、contain模式是要求显示整张图片, 可以对它进行等比缩小, 图片会显示完整,可能会露出Image控件的底色。 如果图片宽高都小于控件宽高，则不会对图片进行放大。

3、stretch模式不考虑保持图片原来的宽,高比.填充整个Image定义的显示区域,这种模式显示的图片可能会畸形和失真。

4、center模式contain模式基础上支持等比放大。

5、repeat模式平铺（ios可用）

## 2、网络资源

使用网络资源只需要如下引用：

<Image  
 style={{height: 100, width: 100}}  
 source={{uri:'https://ss2.bdstatic.com/70cFvnSh\_Q1YnxGkpoWK1HF6hhy/it/u=3938488120,2576257904&fm=27&gp=0.jpg'}}  
/>

网络资源无法获得高度宽度 必须自己设置宽高。

## 3、本地图片

访问存储在用户手机中的图片，首先区分android和ios不同的平台文件地址。

**let** imaUrl=Platform.OS==='android'?'file:///'+'图片地址':'图片地址';

在android中访问file需要在url前面添加“file：///”，ios不需要。

使用：

<Image  
 style={{height: 100, width: 100}}  
 source={{uri: imaUrl}}  
/>

使用uri访问就可以显示本机图片资源了。

# 布局介绍

主要参考贾鹏辉的技术博客地址<http://www.devio.org/2016/08/01/Reac-Native布局详细指南/>

我就简单的介绍一下：

# Flex in React Native

RN支持的flex。

## 父类

* flexDirection enum(‘row’, ‘column’,’row-reverse’,’column-reverse’)
* flexWrap enum(‘wrap’, ‘nowrap’)
* justifyContent enum(‘flex-start’, ‘flex-end’, ‘center’, ‘space-between’, ‘space-around’)
* alignItems enum(‘flex-start’, ‘flex-end’, ‘center’, ‘stretch’)

1. **flexDirection**

属性定义了父视图中的子元素沿横轴或侧轴方片的排列方式。

* row: 从左向右依次排列
* row-reverse: 从右向左依次排列
* column(default): 默认的排列方式，从上向下排列
* column-reverse: 从下向上排列

1. flexWrap

属性定义了子元素在父视图内是否允许多行排列，默认为nowrap。

* nowrap flex的元素只排列在一行上，可能导致溢出。
* wrap flex的元素在一行排列不下时，就进行多行排列。

1. ustifyContent

属性定义了浏览器如何分配顺着父容器主轴的弹性（flex）元素之间及其周围的空间，默认为flex-start。

* flex-start(default) 从行首开始排列。每行第一个弹性元素与行首对齐，同时所有后续的弹性元素与前一个对齐。
* flex-end 从行尾开始排列。每行最后一个弹性元素与行尾对齐，其他元素将与后一个对齐。
* center 伸缩元素向每行中点排列。每行第一个元素到行首的距离将与每行最后一个元素到行尾的距离相同。
* space-between 在每行上均匀分配弹性元素。相邻元素间距离相同。每行第一个元素与行首对齐，每行最后一个元素与行尾对齐。
* space-around 在每行上均匀分配弹性元素。相邻元素间距离相同。每行第一个元素到行首的距离和每行最后一个元素到行尾的距离将会是相邻元素之间距离的一半。

1. alignItems

属性以与justify-content相同的方式在侧轴方向上将当前行上的弹性元素对齐，默认为stretch。

* flex-start 元素向侧轴起点对齐。
* flex-end 元素向侧轴终点对齐。
* center 元素在侧轴居中。如果元素在侧轴上的高度高于其容器，那么在两个方向上溢出距离相同。
* stretch 弹性元素被在侧轴方向被拉伸到与容器相同的高度或宽度。

## 子视图

1）alignSelf

属性以属性定义了flex容器内被选中项目的对齐方式。注意：alignSelf 属性可重写灵活容器的 alignItems 属性。

* auto(default) 元素继承了它的父容器的 align-items 属性。如果没有父容器则为 “stretch”。
* stretch 元素被拉伸以适应容器。
* center 元素位于容器的中心。
* flex-start 元素位于容器的开头。
* flex-end 元素位于容器的结尾。

2）flex

属性定义了一个可伸缩元素的能力，默认为0。