
REACT NATIVE

概要

- ▶ 起源
- ▶ 官方首页介绍
- ▶ 基础语法
- ▶ 优点
- ▶ 聊天窗口demo



起源

- ▶ Native App
- ▶ HTML5
- ▶ 2012.09 Facebook放弃移动H5
- ▶ 2015.03 React Native iOS版发布
- ▶ 2015.09 React Native Android版发布

官方首页介绍

- ▶ 原生的iOS组件
- ▶ 异步执行
- ▶ 触摸处理
- ▶ 弹性框和样式
- ▶ 可扩展性

官方首页介绍 - 原生的iOS组件

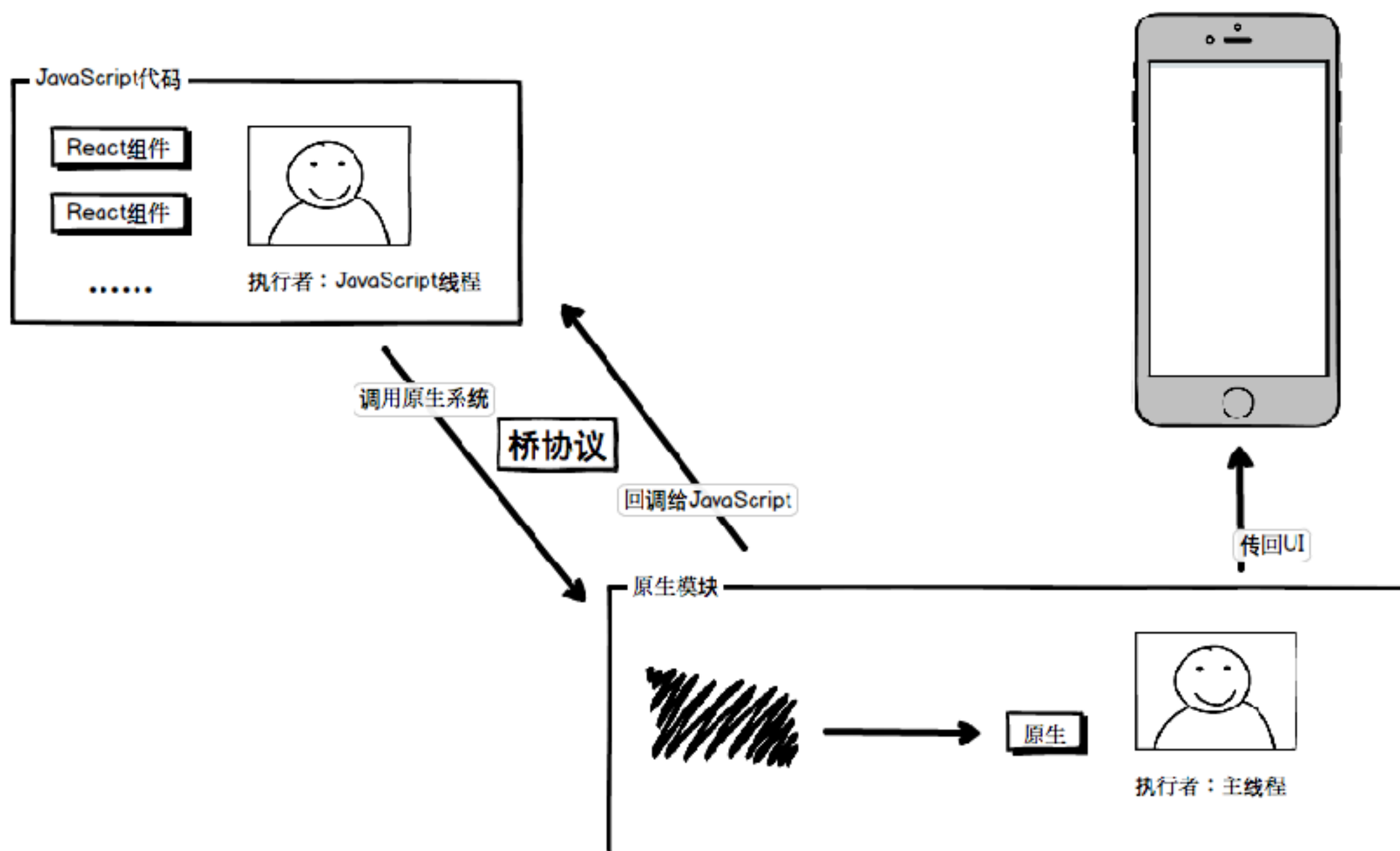
► 如 UITabBar、UINavigationController



官方首页介绍 - 原生的iOS组件

```
var React = require('react-native');
var { TabBarIOS, NavigatorIOS } = React;
var App = React.createClass({
  render: function() {
    return (
      <TabBarIOS>
        <TabBarIOS.Item title="React Native" selected={true}>
          <NavigatorIOS initialRoute={{ title: 'React Native' }} />
        </TabBarIOS.Item>
      </TabBarIOS>
    );
  },
});
```

官方首页介绍 - 异步执行



官方首页介绍 - 触摸处理

- ▶ 原生iOS : Responder Chain
- ▶ React Native : TouchableHiglight

官方首页介绍 - 样式

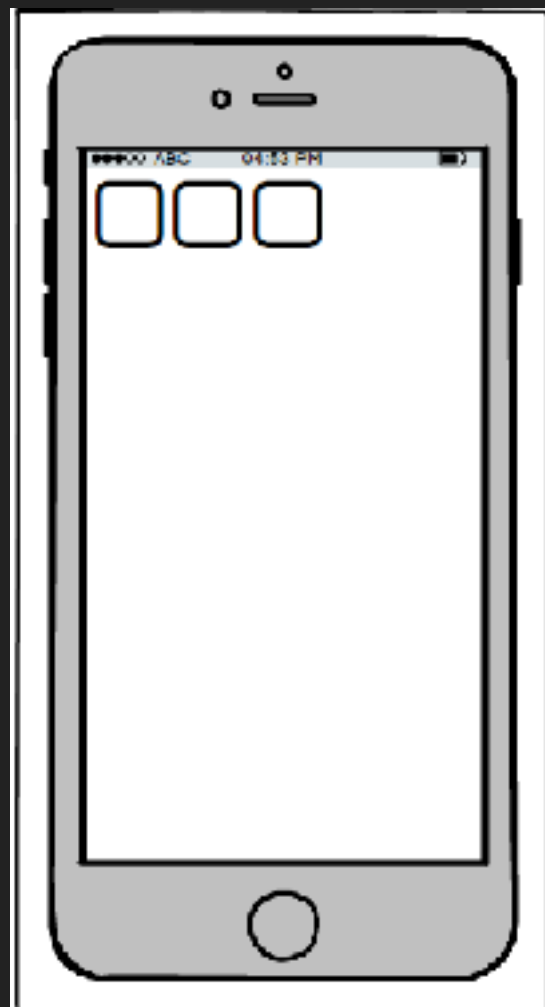
```
const styles = StyleSheet.create({
  nav: {
    marginTop: 25
  },
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#F5FCFF',
  }
});
```

```
<View style={styles.container}>hello</View>
```

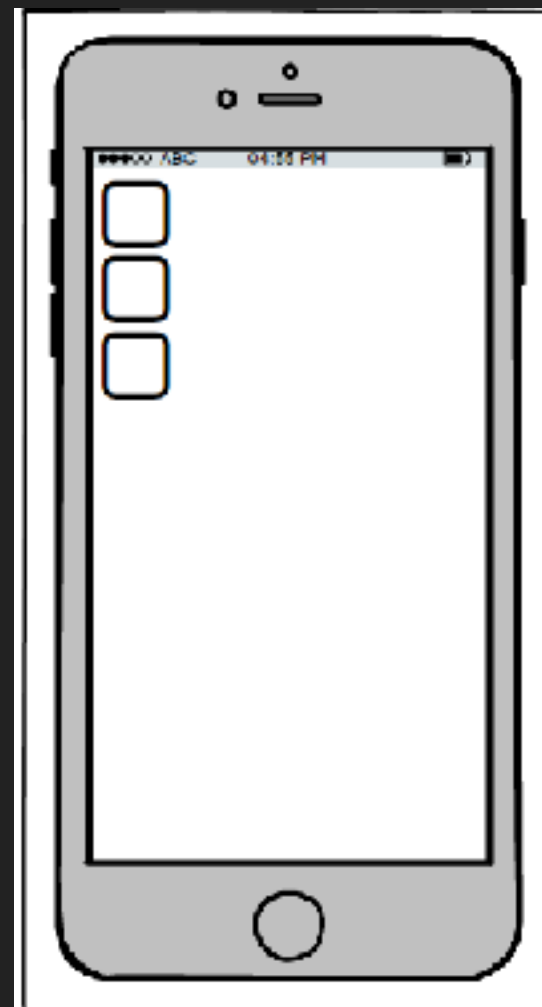
官方首页介绍 - 弹性框

- ▶ flexDirection : 布局的主轴
- ▶ alignItems : 子元素沿着次轴的排列方式
- ▶ justifyContent : 子元素沿着主轴的排列方式

官方首页介绍 - 弹性框

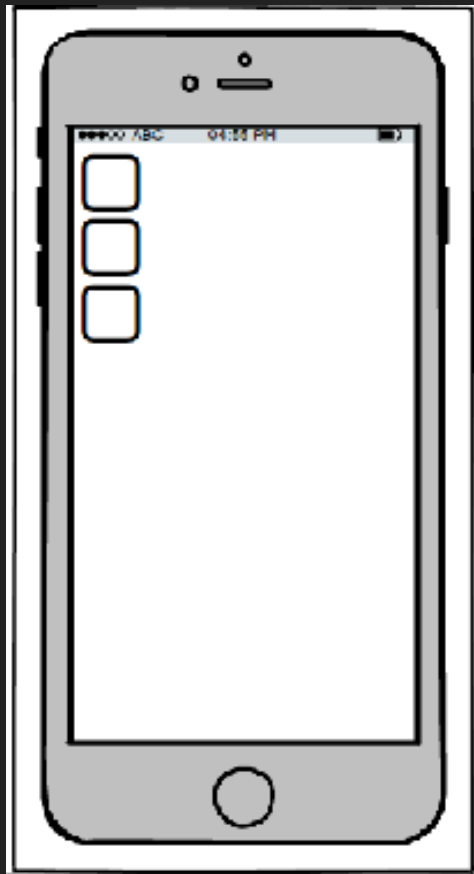


`flexDirection:row`

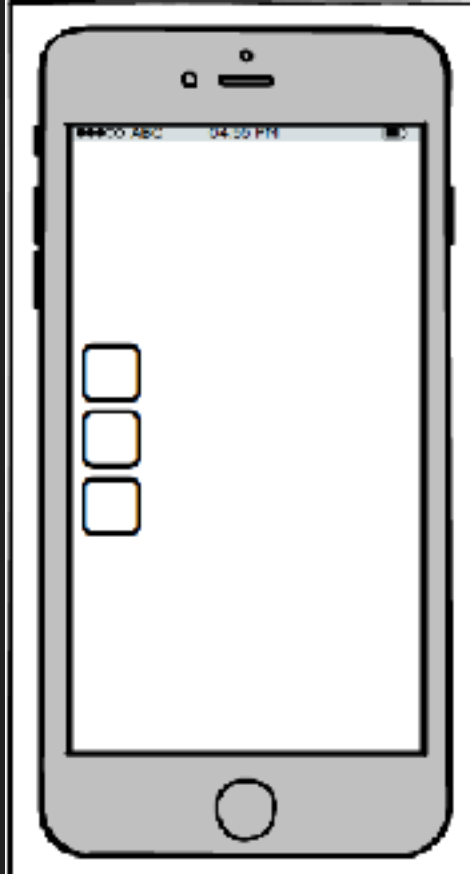


`flexDirection:column`

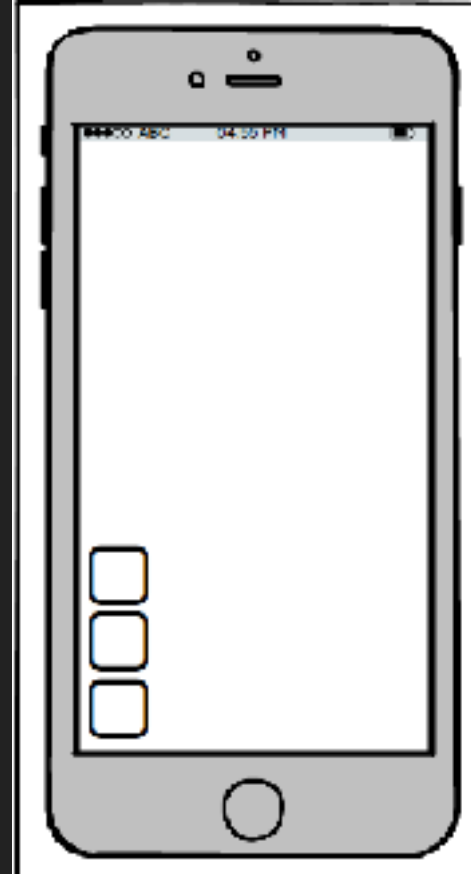
官方首页介绍 - 弹性框



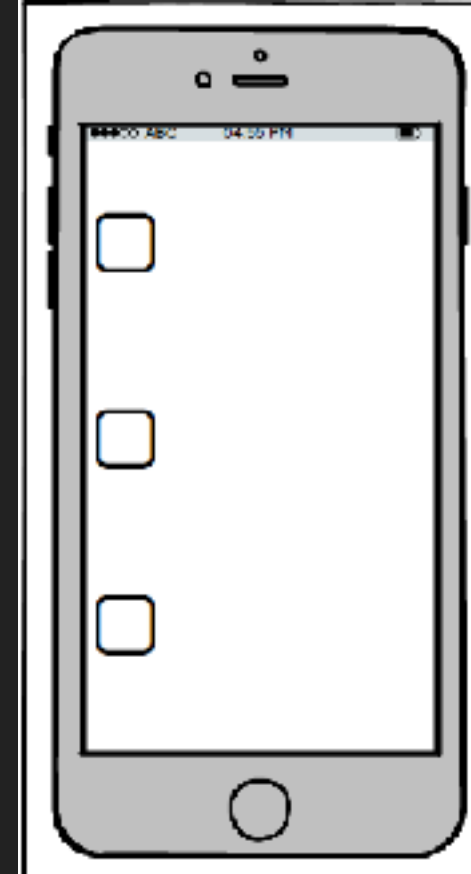
justifyContent:
flex-start



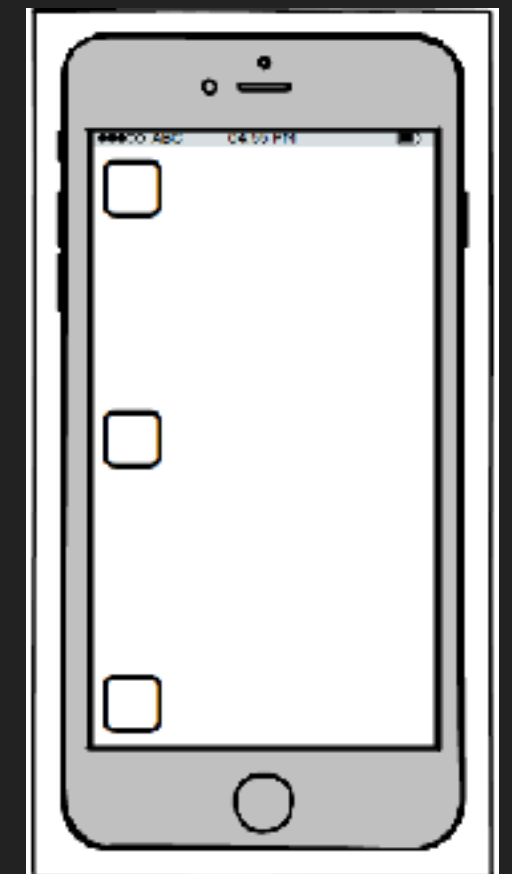
justifyContent:
center



justifyContent:
flex-end

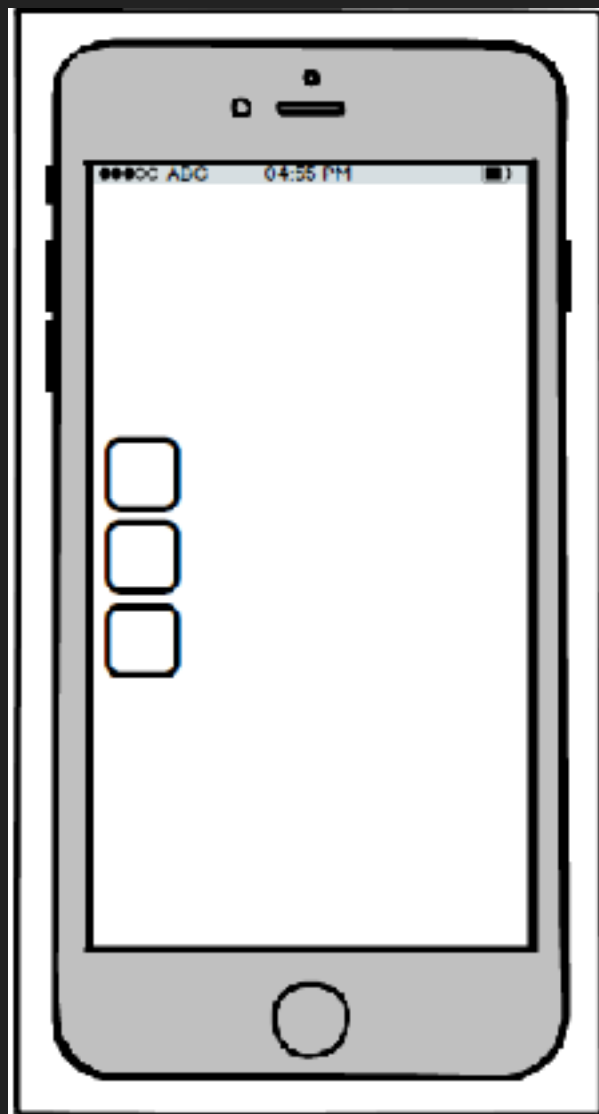


justifyContent:
space-around

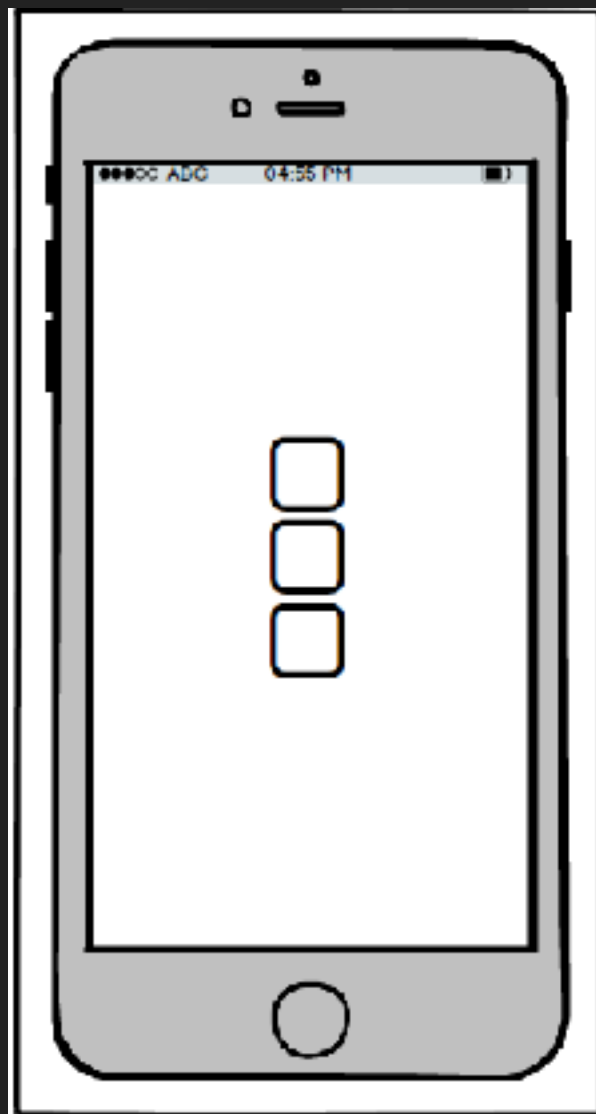


justifyContent:
space-between

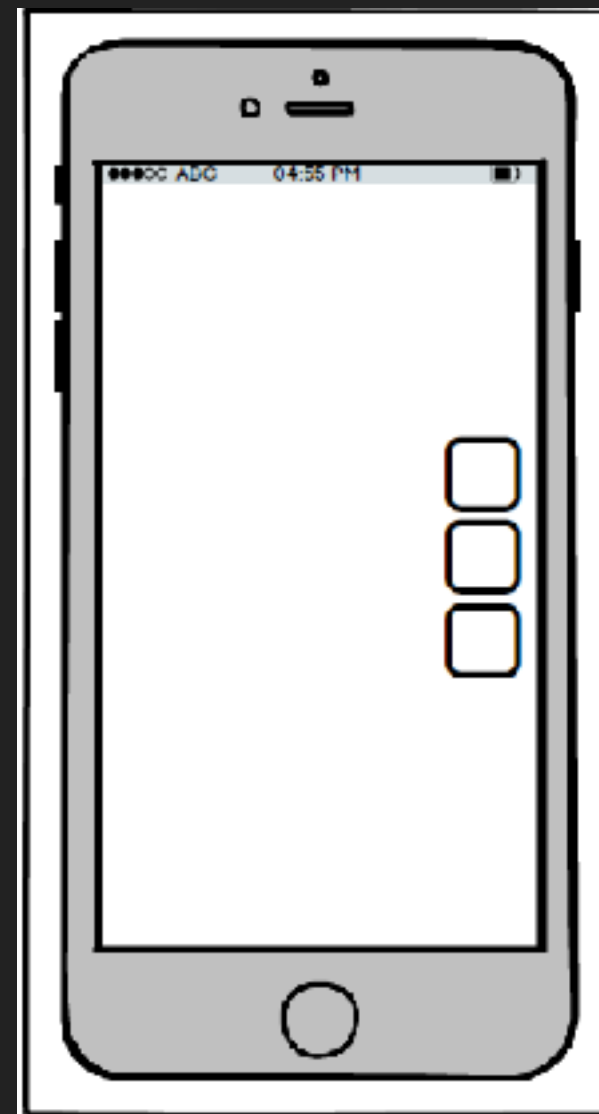
官方首页介绍 - 弹性框



alignItems:
flex-start



alignItems:
center



alignItems:
flex-end

官方首页介绍 - 可扩展性

- ▶ RCTBridgeModule

基础语法

- ▶ Props 属性
- ▶ State 状态

基础语法 - PROPS 属性

```
class Bananas extends Component {  
  render() {  
    let pic = {  
      uri: 'https://XXX.jpg'  
    };  
    return (  
      <Image source={pic} style={{width: 193, height: 110}} />  
    );  
  }  
}
```



基础语法 - STATE 状态

```
class Blink extends Component {
  constructor(props) {
    super(props);
    this.state = { showText: true };

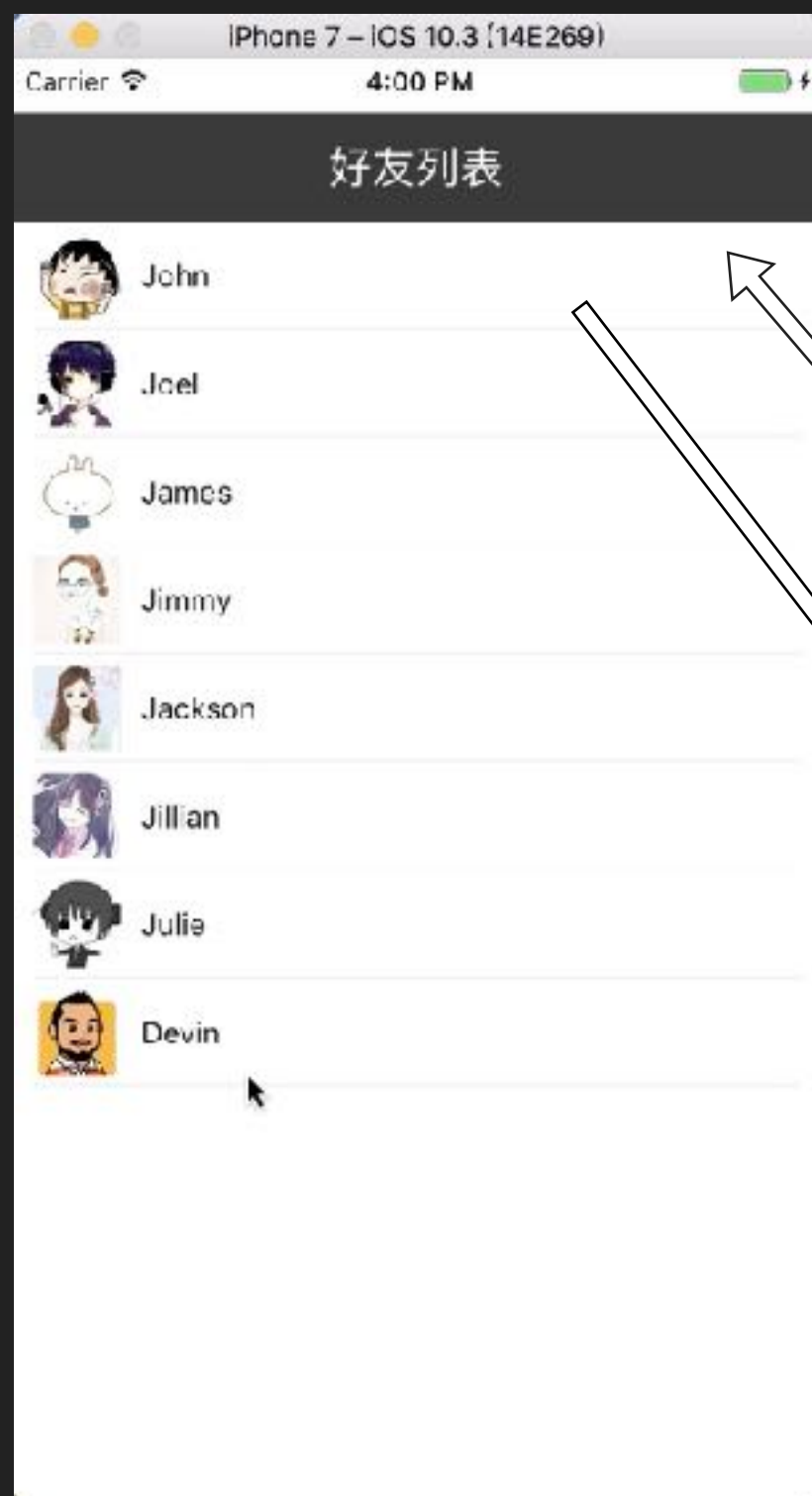
    // 每1000毫秒对showText状态做一次取反操作
    setInterval(() => {
      this.setState(previousState => {
        return { showText: !previousState.showText };
      });
    }, 1000);
  }

  render() {
    // 根据当前showText的值决定是否显示text内容
    let display = this.state.showText ? this.props.text : ' ';
    return (
      <Text>{display}</Text>
    );
  }
}
```

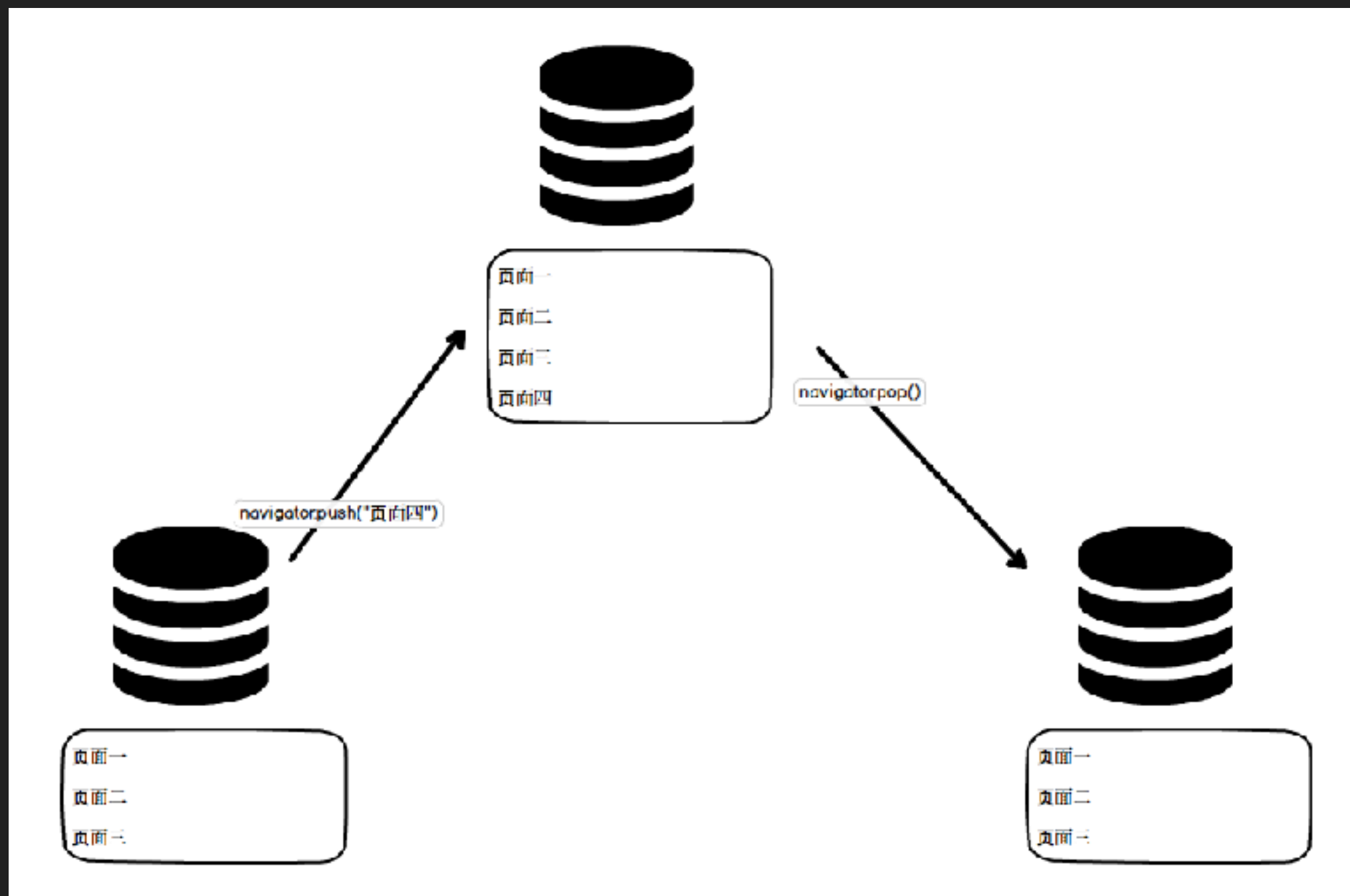
优点

- ▶ 性能媲美原生系统
- ▶ 开发效率高
- ▶ 实时远程调试

聊天窗口DEMO



聊天窗口DEMO



聊天窗口DEMO

- ▶ Demo源码: <https://github.com/abigaleypc/react-native-demo>

THANKS