复习：

WebView方案：Vue.js+MintUI

混编方案：Angular+Ionic

JSBridge方案：React+ReactNative

1.ReactNative概述

RN本质是一种JSBridge方案，代码使用JS/React来编写，RN会将这些代码转换为Android/Java或者iOS/OC，最后运行在手机端

优势：一套代码到处运行；运行效率高；功能丰富；

不足：原生Android和iOS默认组件效果不一致，非要相同只能深度定制

**提示：语法是React语法，采用组件化编程；可以使用的组件要么是用户自定义的组件，要么是RN预定义的组件，如<View>、<Text >、<Image>等，不能使用任何HTML标签组件——底层没有浏览器！**

官网：www.reactnative.com

中文网：reactnative.cn

搭建RN开发运行环境：

前提：

Windows系统(服务器) + Android手机或模拟器(客户端)

MacOS系统(服务器) + iOS手机或模拟器(客户端)

①下载并安装全局脚手架工具

npm i -g react-native-cli

②运行脚手架工具创建空白项目

react-native init 项目名

③在PC上下载并安装Android或iOS开发环境，编译当前项目得到APP安装程序包——详情参见手册

此步骤会生成：项目目录\android\app\build\outputs\apk\debug\app-debug.apk

④进入空白项目并运行其中的开发服务器

cd 项目名

npm start

⑤在手机上安装APP安装包，远程访问PC服务器

|  |
| --- |
| 注意：开发服务器命令行界面千万不要用鼠标选中任何内容！如果选中了客户端手机APP无法再得到更新后的内容！！ |

2.RN开发常见的错误

①点击APP图标，立即弹出对话框“此应用程序已关闭”

原因：APP安装失败，或与当前手机不兼容；

可以卸载并重新安装；

不行就重启一下模拟器再试试；

再不行换一个其它版本的模拟器

②点击APP图标，一片空白

原因：APP启动失败了，或者无法从服务器获取更新后的内容；

保证服务器端处于运行状态，并且控制台没有用鼠标选中任何内容

关闭手机APP程序，重新启动；

③点APP图标，显示黑色底红色错误信息

原因：APP设置有问题，或者服务器端代码执行出错；

仔细查看红色错误信息，从中找到解决方案

常见APP红色错误信息：

|  |
| --- |
| Unable to load Script ...  原因：APP无法从服务器加载更新后的内容  解决办法：给APP重新设置服务器的主机名和端口号  摇一摇 > Dev Settings > Debug Server host & port for device > 输入完成后记得重启APP即可 |
| Could not connect to development server  原因：APP不能连接到开发服务器；可能原因：①开发服务器没有启动 ②没有正确查找到服务器的IP地址（正确的地址应该形如 172.x.x.x或者192.168.x.x） ③APP没有正确的设置服务器IP和端口 |

3.ReactNative中的组件样式编写

提示：RN应用中没有浏览器，没有CSS解释器！RN样式有些名称有点类似CSS样式名称，但是本质完全不一样！！而且名称细节也不一样，例如不存在float、存在borderColor但是不存在border等。

①RN中的尺寸都是数字类型，不能赋值为字符串！也没有单位！如fontSize: 14

②RN中父元素的样式，不会继承给子元素——每个元素的样式都由自己的style完全控制，不需要叠加计算父元素的样式

**方法1：行内样式**

<Any style={ {color:'red', padding: 20} }>

**方法2：外部样式**

let ss = StyleSheet.create({

success: { color: 'red' }

})

<Any style={ ss.success }>

也可以给一个组件指定多个样式对象——多个对象组成的数组：

<Any style={ [ ss.success, ss.textRight, {fontSize:30} ] }>

|  |
| --- |
| CSS：Cascading StyleSheet，层叠/级联样式表，子元素可以继承父元素的样式 |
| RN SS： StyleSheet，样式表，子元素不会继承父元素的样式 |

4.ReactNative中的组件布局

提示：RN中的组件没有“块级元素”和“行内元素”之分；

所有元素的尺寸由width和height两个属性控制；

若没有指定宽和高，默认有布局系统来控制尺寸——一套很类似于CSS重点FlexBox布局系统。常用属性：

**①flex：指定当前元素在主轴上的尺寸占比**

number

**②flexDirection：子元素排列方向**

column：默认值，纵向排列

column-reverse：纵向排列（反向）

row：横向排列

row-reverse：横向排列（反向）

**③alignItems：子元素在交叉轴方向上的对齐方式**

flex-start：子元素对齐到容器的开始位置

flex-end：子元素对齐到容器的结束位置

center：子元素对齐到容器的中部

stretch：默认值，子元素在交叉轴方向上拉伸撑满容器

**④justifyContent：调整所有的子元素（即内容）在主轴上的分布方式**

flex-start：子元素码放到容器的主轴开始位置

flex-end：子元素码放到容器的主轴结束位置

center：子元素码放到容器的主轴中央

space-between：第一个孩子在开始，最后一个孩子在结尾，其它孩子之间平均分配空白空间

space-around：每个孩子上下都包裹着相同的空白空间——相邻两个子元素间的空白是开头和结尾空白的2倍

space-evenly：所有的空白在孩子之间以及上下顶部平均分配

练习：使用RN的布局系统实现如下内容：

flex、flexDirection、alignItems、justifyContent



分析：三行两列

<View grid>

<View row>

<View col>Text x 3</View>

<View col>Text x 3</View>

</View> x3

</View>

5.ReactNative提供的常用组件

**①View：**最简单的容器，默认没有高度，要靠内容撑起来，内容溢出后将不显示

<View style={ }>子组件</View>

注意：View中不能直接放置文本，文本只能放置在Text中

**②ScrollView：**可以滚动的容器，默认没有高度，要靠内容撑起来，内容溢出后将出现滚动条

<ScrollView style={ }>子组件</ScrollView>

**③Text：**显示单行或多行文本

<Text style={} onPress={} onLongPress={} numberOfLines={显示出来的行数} ellipsizeMode='省略号显示的位置'>文本{'\n'}内容</Text>

注意：RN允许Text中嵌套Text，而且子Text可以继承父Text的样式！

练习：点击一段文本，实现单击计数器

**④Button**：按钮

<Button title="按钮上的文字" color="按钮背景色" onPress={ } disabled={true/false} />

**⑤TextInput：**文本输入框(单行/多行)

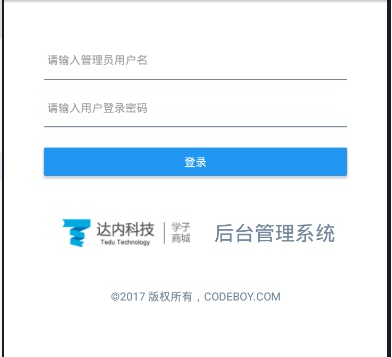
<TextInput placeholder="提示文字" secureTextEntry={true} multiline={true} numberOfLines={3} value={this.state.uname} onChangeText={this.doChange}/>

**⑥Image**：图片

课后任务：

①读RN手册，掌握Image组件的两种用法

②完成“RN阶段项目”管理员登录页的页面布局效果



提示：LOGO图片可以下载后保存在当前项目目录下，使用<Image>元素进行渲染