**微信小程序笔记**

**第一课 微信小程序介绍**

1. **什么是微信小程序？**

**微信小程序，简称小程序，英文名Mini Program，是一种不需要下载安装既可使用的应用，它实现了应用”触手可及”的梦想，用户扫一扫或搜一下既可打开应用。**

1. **为什么要使用微信小程序？**
2. **微信有海量用户，而且粘性很高，在微信里开发产品更容易触达用户；**
3. **推广app或公众号的成本太高；**
4. **开发适配成本低；**
5. **容易小规模试错，然后快速迭代；**
6. **跨平台；**

**第二课 微信小程序环境准备**

**一、APPID：wx535a8e3cc7d6f23f**

**第三课 原生框架的目录结构**

1. **小程序框架的目标：**

**是通过尽可能简单、高效的方式让开发者可以在微信中开发具有原生APP体验的服务。**

1. **小程序文件结构和传统web对比**
2. **结构：HTML——wxml**
3. **样式：CSS——wxss**
4. **逻辑：JavaScript——JavaScript**
5. **配置：无——json**

**通过以上对比得出，传统web是三层结构，而微信小程序是四层结构，多个一层配置json文件**

1. **项目的基本目录**



**四、原生框架一般都是mina.html页面**

**第四课 全局配置文件**

**1.1【pages】字段**

1. **如果想要将自己写的页面一个运行，既可以吧自己写的页面放到pages文件的第一个即可。**
2. **只有在微信小程序中的app.json文件中添加页面才能被添加，在其他任何地方添加都不可以添加。**

**1.2【window】字段**

**一、window用于设置小程序的状态栏、导航条、标题、窗口背景色。**

**二、window的属性设置：**

1. **navigationBarBackgroundColor——导航栏背景颜色**
2. **navigationBarTitleText——导航栏中的标题**
3. **navigationBarTextStyle——导航栏中标题的文字颜色(只能为黑色或者白色的文字颜色black或white)**
4. **backgroundTextStyle——下拉刷新的样式，默认为light(白色)，共两个值：light或dark(黑)**
5. **enablePullDownRefresh——是否开启下拉刷新功能，默认为false**
6. **backgroundColor——指下拉刷新那个区域的背景颜色**

**1.3【tabbar】字段**

**一、tabbar也就是一个底部标签效果**

**二、tabbar中list属性设置：**

1. **pagePath：页面跳转路径**
2. **text：底部标题**
3. **iconPath：未被选中的图标路径**
4. **selectedIconPath：已选中的图标路径**
5. **tabbar的属性设置：(不懂翻文档)**
6. **color：设置底部导航文字未选中的颜色**
7. **selectedColor：设置底部导航文字选中的颜色**
8. **backgroundColor：“设置底部导航的背景颜色”**
9. **position：设置底部导航的位置**

**1.4 页面配置文件**

**一、页面配置文件：就是可以将每个页面的导航栏的单独进行设置**

**二、将app.json文件中的window对象中的属性可以单独对页面中的json文件进行设置，只需要进行复制粘贴即可**

**三、当你某个页面不需要进行下拉刷新的时候也可以在页面配置文件中单独设置‘**

**1.5 sitemap配置文件**

**一、sitemap配置文件主要用于配置小程序及其页面是否允许被微信索引**

**第五课 模板语法**

1. **模板语法**

**WXML：是框架设计的一套标签语言，结合基础组件、事件系统，可以构建出页面的结构**

1. **数据绑定：**
2. **普通写法**

**<view>{{message}}</view>**

**Page({**

**Data:{**

**Message:‘Hellow mina!’**

**}**

**})**

1. **组件属性**

**<view id=”item-{{id}}”></view>**

**Page({**

**Data:{**

**Id:0**

**}**

**})**

1. **运算：**
2. **三元运算**

**<view hidden=”{{flag?true:false}}”>hidden</view>**

1. **算数运算**

**<view>{{a+b}}+{{c}}+d</view>**

**Page({**

**Data:{**

**A:1,**

**B:2,**

**C:3**

**}**

**})**

1. **逻辑判断**

**<view wx:if=”{{length>5}}”></view>**

1. **字符串运算**

**<view>{{“hello”+name}}</view>**

**Page({**

**Data:{**

**}**

**})**

1. **数组和对象循环：**
2. **对象循环**
3. **wx:for=”{{数组或者对象}}” wx:for-item=”循环项的名称” wx:for-index=”循环的索引”**
4. **wx:key=”唯一的值” 用来提高列表渲染的性能**

* **wx:key：绑定一个普通的字符串的时候，那么这个字符串名称肯定是循环数组中的对象的唯一属性**
* **wx:key=”\*this”：就表示你的数组是一个普通的数组\*this表示是循环项**

1. **当出现数组的嵌套循环的时候尤其要注意一下绑定的名称不要重名**

* **Wx:for-item=”item” wx:for-index=”index”**

1. **默认情况下我们不写**

* **wx:for-item=”item” wx:for-index=”index”**
* **小程序也会把循环项的名称和索引的名称item和index**
* **只有一层循环的话(wx:for-item=”item” wx:for-index=”index”)可以省略不写**

1. **对象循环**
2. **wx:for=”{{对象}}” wx:for-item=”对象的值” wx:for-index=”对象的属性”**
3. **循环对象的时候最好把item和index的名称都修改一下**

* **wx:for-item=”value” wx:for-index=”key”**

1. **block标签：‘’**
2. **占位符的标签**
3. **写代码的时候 可以看到这个标签存在**
4. **页面渲染小程序会把它移除掉**
5. **条件渲染：**
6. **wx:if**
7. **wx:if={{true/false}}：进行显示隐藏**
8. **wx:if wx:elif wx:else(不需要表达式)**
9. **hidden**
10. **hidden=”{{true}}”：隐藏**
11. **hidden=”{{false}}”：显示**
12. **当在什么情况下使用哪一个进行显示隐藏**
13. **当标签不是频繁的切换显示，优先使用wx:if**

* **直接把标签从页面结构给移除掉**

1. **当标签频繁的切换显示的时候，优先使用hidden**

* **通过添加样式的方式来切换显示**
* **Hidden属性不要和样式display属性一起使用，因为display属性会覆盖hidden样式**

**第六课 事件绑定**

1. **步骤：**
2. **需要给input标签绑定input事件**
3. **绑定关键字：bindinput**
4. **如果获取输入框的值**
5. **通过事件源对象来获取：e.datail.value**
6. **把输入框的值赋值到data当中不能直接赋值**
7. **错误示范：**

**This.data.num=e.detail.value**

**This.num=e.detail.value**

1. **正确示范：**

**This.setData({**

**Num:e.datail.value**

**})**

1. **需要加入一个点击事件**
2. **bindtap**
3. **无法在小程序当中的事件中直接传参**
4. **通过自定义属性的方式来传递参数**
5. **事件源中获取，自定义属性**

**第七课 样式-rpx**

1. **wxss：：是一套样式语言，用于描述wxml的组件样式**
2. **wxss与css相比，wxss的扩展特性有：**
3. **响应式长度单位rpx**
4. **样式导入**
5. **wxss的尺寸单位：**

**rpx(responsive pixel)：可以更具屏幕宽度进行自适应。规定屏幕宽度为750rpx。则750rpx=375px=750物理像素，1rpx=0.5px=1物理像素。**

|  |  |  |
| --- | --- | --- |
| **设备** | **rpx换算px(屏幕宽度/750)** | **px换算rpx(750/屏幕宽度)** |
| **iPhone5** | **1rpx=0.42px** | **1px=2.34rpx** |
| **iPhone6** | **1rpx=0.5px** | **1px=2rpx** |
| **iPhone6 Plus** | **1rpx=0.552px** | **1px=1.81rpx** |

**建议：开发微信小程序时设计师可以用iPhone6作为视觉稿的标准**

**使用步骤：**

1. **确定设计稿宽度pageWidth**
2. **计算比例750rpx=pageWidth px，因此1px=750rpx/pageWidth**
3. **在less文件中，只要把设计稿中的px=>750/pageWidth rpx即可**

**第八课 样式的引入**

1. **引入样式：引入样式的代码是通过@import来引入**

**如：@import “../../style/commin.wxss”**

**第九课 选择器以及less文件**

1. **在微信小程序中不支持：\*通配符，我们通常使用每个标签进行编写**
2. **在小程序中使用less文件**

**原生小程序不支持less，其他基于小程序的框架大体都支持，如wepy，mpvue，taro等。但是仅仅因为一个less功能，而去引入一个框架，肯定是不可取的。因此可以使用一下方式来实现：**

1. **编辑器是vs code**
2. **安装插件easy less**
3. **在vs code的设置中加入如下，配置：**

**“less.compile”:{**

**“outExt”:”.wxss”**

**}**

1. **在要编写样式的地方，新建less文件，如index.less，然后正常编辑即可**
2. **在less文件中编写样式的格式：在wxss文件中会自动生成相对应的样式**
3. **定义变量的方式：**

**@color:yellow;**

**view{**

**color:@color;**

**}**

1. **嵌套样式**

**view{**

**view{**

**text{**

**color:red;**

**}**

**}**

**}**

1. **自动导入**

**@import “../../style/commin.less”**

**view{**

**color:@color;**

**}**

**第十课 常见组件**

1. **view标签**

**相当于html中的div标签，但是其中有新增的属性**

1. **text标签**
2. **文本标签**
3. **只能嵌套自己**
4. **长按文字可以复制(只有text标签有这个功能)**
5. **可以对空格、回车进行编码**

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名** | **类型** | **默认值** | **说明** |
| **selectable** | **Boolean** | **false** | **文本是否可选** |
| **decode** | **Boolean** | **false** | **是否解码** |

1. **image标签**
2. **src指定要加载的图片的路径**
3. **图片村子爱默认的高度和宽度240\*320，原图大小时100\*200**
4. **mode决定图片内容，如何和图片标签宽高做适配**
5. **scaleToFill：默认值，确保纵横比缩放图片，使图片的宽高完全拉伸至填满image元素**
6. **aspectFit：保持宽高比，确保图片的长边，显示出来，页面轮播图，常用**
7. **widthFix：以前web的图片的宽度指定了之后高度会自己按比例来调整 常用**
8. **aspectFill：保持纵横比缩放图片，只保证图片的短边能完全显示出来 少用**
9. **bottom：类似以前的background-position**
10. **小程序放当中的图片，直接就支持，懒加载**

**lazy-load：会自己判断当图片出现在视口上下三屏的高度之内的时候，自己就开始加载图片。**