微信小程序api 微信api

|  |
| --- |
| 开发微信的小程序的文档 |

小程序反编译

|  |  |
| --- | --- |
| ****一：下载node.js**** | <https://nodejs.org/en/> |
| ****二：下载反编译脚本**** | <https://github.com/qwerty472123/wxappUnpacker> |
| ****三：下载安装夜神模拟器**** | <https://www.yeshen.com/> |
| ****四：下载RE文件管理器**** | 百度 |
| ****五：获取.wxapkg文件**** | 1. 打开里面的微信，并自动安装re文件管理器及打开 2. 设置root权限 3. 模拟器里打开微信，在微信中运行你想要获取的小程序，re文件管理器进入/data/data/com.tencent.mm/MicroMsg/{数字串}/appbrand/pkg/ 4. 你会看到发现里面的一些.wxapkg后缀的文件，可以根据时间来判断那个是你需要小程序.wxapkg文件，右键长按文件，点击右上角压缩所选文件,然后再将压缩好的包通过微信发送给好友或者文件助手 |
| ****六：反编译**** | 1. npm安装依赖 2. node wuWxapkg.js D:\\_163200311\_32.wxapkg |
| 参考博客 | <https://blog.csdn.net/u013373006/article/details/83115965> |
| npm安装问题 | A complete log of this run can be found in:  多次安装失败 |

位置

|  |  |  |
| --- | --- | --- |
| 框架 | 组件 | api |
|  |  |  |
|  | 导航超链接 | 界面弹窗 |
|  |  | 媒体：视频、音频、地图、图片、直播 |

全局配置

|  |  |  |
| --- | --- | --- |
| pages | 页面路劲列表 | "pages": [  "pages/mine/mine",  "pages/home/home",  "pages/logs/logs",  "pages/index/index"  ] |
| window | 全局默认窗口表现 | "window": {  "backgroundTextStyle": "dark",  "navigationBarBackgroundColor": "#f00",  "navigationBarTitleText": "商城",  "navigationBarTextStyle": "white",  "enablePullDownRefresh": true  } |
| tabBar | 底部栏 | "tabBar": {  "color": "#666666",  "selectedColor": "#ff0000",  "list": [  {  "text": "首页",  "pagePath": "pages/home/home",  "iconPath":"img/index.png",  "selectedIconPath":"img/indexHL.png"  },  {  "text": "我的",  "pagePath": "pages/mine/mine",  "iconPath": "img/account.png",  "selectedIconPath": "img/accountHL.png"  }  ]  } |
| 开启下来刷新 | "enablePullDownRefresh": true | |

页面配置

|  |  |  |
| --- | --- | --- |
| 跟全局配置的属性差不多，只能在此页面生效 | | |
| navigationBarTitleText | 导航栏标题文字内容 | {  "navigationBarTitleText": "我的"  } |

WXML语法 HTML

|  |
| --- |
| <text>基础语法</text>  <view wx:for="{{arr\_obj}}">{{index}}---{{item.time}}</view>  <view wx:if="{{bool}}">dfe</view>  <view wx:else>154</view> |
| <text>模板</text>  <template name="bing">  <view>{{name}}----{{age}}</view>  </template>  <template is="bing" data="{{name:'king',age:18}}"></template> |
| <text>引用</text>  <include src="top.wxml"/>  <include src="footer.wxml"/> |
| <text>模块</text>  <wxs src="../../utils/test1.wxs" module="sy" />  <view wx:for='{{sy.list}}'>{{item.name}}</view> |

WXS语法 模块数据

|  |
| --- |
| module.export={  list:[  {name:"小狗"},  { name: "小欧" },  { name: "小雀" },  { name: "小溪" }  ]  }  <text>模块</text>  <wxs src="../../utils/test1.wxs" module="sy" />  <view wx:for='{{sy.list}}'>{{item.name}}</view> |
|  |
| stringify(object): 将 object 对象转换为 JSON 字符串，并返回该字符串。 |
| parse(string): 将 JSON 字符串转化成对象，并返回该对象。 |
|  |

无缝轮廓图 滚动条 弹性布局 滚动视图 日期选择 一些图标 进度条 富文本

|  |
| --- |
| <text>轮廓图</text>  <swiper autoplay="true" interval="1000" indicator-dots="true" circular="true">  <swiper-item wx:for="{{imgUrls}}">  <image src="{{item}}"/>  </swiper-item>  </swiper>  <view style="clear:both"></view> |
| <text>滚动条</text>  <scroll-view style="height:100px" scroll-y scroll-top="20" scroll-with-animation="true">  </scroll-view> |
| <text>滚动视图</text>  <movable-area style="height: 150px; width: 300px; background: red;">  <movable-view style="height: 50px; width: 50px; background: blue;" direction="all" inertia="true" x="50" bindchange="test">  </movable-view>  </movable-area> |
| <text>一些图标</text>  <view class="group">  <block wx:for="{{iconSize}}">  <icon type="success" size="{{item}}"/>  </block>  </view> |
| <text>进度条</text>  <progress percent="45" show-info="true"></progress> |
| <text>富文本</text> |

日期 单选 修改data数据

|  |
| --- |
| <text>性别选择</text>  <radio-group>  <radio color="grey"/>  <radio size="40"/>  <radio/>  </radio-group>  <text>清晰度</text>  <slider></slider>  <text>选择按钮</text>  <switch></switch> |
| <picker mode="date" bindchange="test3">  日期:{{val}}  </picker>  test3(event) {  this.setData({ //修改data数据  val: event.detail.value  })  }, |

点击事件

|  |  |  |
| --- | --- | --- |
| bindchange | 点击触发 |  |
| bindchange | 内容改变触发 |  |
| catchtap | 事件能阻止冒泡  <view bindtap="test2">  <!-- 用catchtap事件不会冒泡 -->  <button size="mini" type="primary" bindtap="test" data-xxx="你好" data-name="tom">dianji</button> | |
| 冒泡事件 | touchstart   touchmove   touchend longtap   tap | |

生命周期

|  |  |
| --- | --- |
| 底部导航栏 | 超链接跳转 |
| 1.页面加载 | 1.页面卸载 |
| 2.页面显示（空白） | 2.页面加载 |
| 3.页面渲染（dom） | 3.页面显示 |
| 4.页面隐藏 | 4.页面渲染 |
| 离开隐藏与进入显示，保存在内存中 |  |

定时器

|  |  |  |
| --- | --- | --- |
| [setTimeout](https://developers.weixin.qq.com/miniprogram/dev/api/base/timer/setTimeout.html) | 循环定时器 |  |
| [setInterval](https://developers.weixin.qq.com/miniprogram/dev/api/base/timer/setInterval.html) | 一次定时器 |  |
| [clearTimeout](https://developers.weixin.qq.com/miniprogram/dev/api/base/timer/clearTimeout.html) | 关闭定时器 |  |
| [clearInterval](https://developers.weixin.qq.com/miniprogram/dev/api/base/timer/clearInterval.html) | 关闭定时器 |  |

弹窗

|  |  |  |
| --- | --- | --- |
| wx.showLoading() | 加载中 |  |
| wx.showModal() | 提示框 |  |
| wx.showToast() | 成功弹框 |  |

案例注意点

|  |
| --- |
| rpx单位以750为标 记混的单位rem |
| 轮廓图图片地址'../../{{item}}' |
| 超出显示y轴 <scroll-view scroll-y></scroll-view> |
| 数组查找一类filter(function(value){return value.a==b})函数 value指每一行 |
| 图片标签为双标记 |
| 数组查找一个find(function(value,index){ return value.a==b})函数 value指每一行index下标 |
| 数量加减中add(),jian(),all()方法的this指wx |
| 屏幕到指定位置显示内容使用onPageScroll(event){}函数event.scrollTop来判断 |
| 设置不可点击 disabled=”false” |

事件委托 target currentTarget的区别

|  |
| --- |
| 长的是当前节点，短的是最终作用的节点 |

案例点

|  |
| --- |
| data数据通过模块赋值  onLoad: function (options) {  //data数据来自模块  this.setData({  products: list.product,  img:list.banner  })  }, |
| 当前属性添加一个class为on 使用事件委托  步骤1 html  <view wx:for="{{classity}}" class="{{num==index?'on':''}}" data-index="{{index}}" >{{item}} {{index}}  步骤2 js  tab(event){  this.setData({  num: event.target.dataset.index  }) |
| 商品显示结合超链接  <scroll-view class="right" scroll-y bindtap="tab">  <view class="box" wx:for="{{products}}">  <navigator>  <image src="../../{{item.img[0]}}"></image>  <view>{{item.tit}}</view>  </navigator>  </view>  </scroll-view> |
|  |

案例点

|  |
| --- |
| 根据类型显示商品列  //在产品中匹配类别相同的并返回一个集合  var arr=list.product.filter(function(value){  return value.class\_name == event.target.dataset.lei;  })  this.setData({  num: event.target.dataset.index,  products:arr.length==0?list.product:arr  }) |
| 随机函数  function choose() {  var num = parseInt(arr.length \* Math.random())  return arr[num].img  } |

详情案例点

|  |  |
| --- | --- |
| 超链接id获得产品信息  //html中  <navigator url="../details/details?id={{item.bid}}"><navigator>  //js中  var pro=list.product.find(function(value){  return value.bid=options.id;  }) | |
| 加减求和  //加  add(){  //当前数量 限制值  var val=this.data.num;  var max=this.data.product.num;  this.setData({  num:val<max?val+1:val  })  this.all()  }  //减  jian(){  //求和  var val=this.data.num;  this.setData({  num:val>1?val-1:val  })  this.all()  }  //求和  all(){  var val = this.data.num  var pay = this.data.food.pay  this.setData({  all: val \* pay  })  } | |
| 屏幕到指定位置显示购物车  onPageScroll(event){  this.setData({  bool:event.scrollTop>253?true:false  })  console.log(event.scrollTop)  } | |
| 点击购物车存储到本地  if(arr.length==0){  arr.push(obj)  }else{  var po = arr.find(function (x, y) {  if(x.id==id){  arr[y].num=arr[y].num+num  return true  }  })  if(!po){  arr.push(obj)  }  } | wx.setStorage({  key: "shop",  data: arr,  success() {  wx.showToast({  title: '添加成功',  icon: 'success',  duration: 2000  })  }  }) |
| 修改购物车里的数量 | |

小程序二次改进