1. JSON文件都是被包裹在一个大括号中 {}，通过key-value的方式来表达数据。JSON的Key必须包裹在一个双引号中，在实践中，编写 JSON 的时候，忘了给 Key 值加双引号或者是把双引号写成单引号是常见错误。另外不能为undefined

Page({

**data**: {

text: "This is page data."

},

**onLoad**: function(options) {

// 页面创建时执行

},

**onShow**: function() {

// 页面出现在前台时执行

},

**onReady**: function() {

// 页面首次渲染完毕时执行

},

**onHide**: function() {

// 页面从前台变为后台时执行

},

**onUnload**: function() {

// 页面销毁时执行

},

**onPullDownRefresh**: function() {

// 触发下拉刷新时执行

},

**onReachBottom**: function() {

// 页面触底时执行

},

**onShareAppMessage**: function () {

// 页面被用户分享时执行

},

**onPageScroll**: function() {

// 页面滚动时执行

},

**onResize**: function() {

// 页面尺寸变化时执行

},

**onTabItemTap**(item) {

// tab 点击时执行

console.log(item.index)

console.log(item.pagePath)

console.log(item.text)

},

// 事件响应函数

viewTap: function() {

this.setData({

text: 'Set some data for updating view.'

}, function() {

// this is setData callback

})

},

// 自由数据

customData: {

hi: 'MINA'

}

})

* navigateTo, redirectTo 只能打开非 tabBar 页面。
* switchTab 只能打开 tabBar 页面。
* reLaunch 可以打开任意页面。
* 页面底部的 tabBar 由页面决定，即只要是定义为 tabBar 的页面，底部都有 tabBar。
* 调用页面路由带的参数可以在目标页面的onLoad中获取。

我们约定，以 Sync 结尾的 API 都是同步 API， 如 [wx.setStorageSync](https://developers.weixin.qq.com/miniprogram/dev/api/storage/wx.setStorageSync.html)，[wx.getSystemInfoSync](https://developers.weixin.qq.com/miniprogram/dev/api/base/system/system-info/wx.getSystemInfoSync.html) 等。此外，也有一些其他的同步 API，如 [wx.createWorker](https://developers.weixin.qq.com/miniprogram/dev/api/worker/wx.createWorker.html)，[wx.getBackgroundAudioManager](https://developers.weixin.qq.com/miniprogram/dev/api/media/background-audio/wx.getBackgroundAudioManager.html) 等，详情参见 API 文档中的说明。

同步 API 的执行结果可以通过函数返回值直接获取，如果执行出错会抛出异常。

<!--wxml-->

<!-- 下面的 getMax 函数，接受一个数组，且返回数组中最大的元素的值 -->

<wxs module="m1">

var getMax = function(array) {

var max = undefined;

for (var i = 0; i < array.length; ++i) {

max = max === undefined ?

array[i] :

(max >= array[i] ? max : array[i]);

}

return max;

}

module.exports.getMax = getMax;

</wxs>

<!-- 调用 wxs 里面的 getMax 函数，参数为 page.js 里面的 array -->

<view> {{m1.getMax(array)}} </view>

// /pages/tools.wxs

var foo = "'hello world' from tools.wxs";

var bar = function (d) {

return d;

}

module.exports = {

FOO: foo,

bar: bar,

};

module.exports.msg = "some msg";

<!-- page/index/index.wxml -->

<wxs src="./../tools.wxs" module="tools" />

<view> {{tools.msg}} </view>

<view> {{tools.bar(tools.FOO)}} </view>

组件：

|  |  |
| --- | --- |
| movable-view必须在movable-area中，而且必须是子节点；  direction：all、vertical、horizontal、none；  inertia：false是否具有惯性；  out-of-bounds:false超过后可视区：是否可移动;  x、y:3;定义x/y轴可以动的范围；  damping:20阻尼系数，过界回弹，值越大，回弹越快；  friction:3;摩擦系数；必须大于0；  disable:false;是否禁用；  scale：false;是否支持缩放，默认在movable内生效；  scale-min:0.5最小缩放  scale-max:3,；最大缩放  scale-value:0.5-10;取值范围；  animationtrue;:是否使用动画；  bindchange:eventhandle;拖动过程中的事件；event.detail = {x, y, source}原因，滑动，超出，回弹  bindscale:envent;缩放过程中的事件，缩放过程中的事件  htouchmove：even;初次横向；  vtouchmove:初次纵向  cover-image:可以嵌套在cover-view内  src:暂不支持base64;  bindload=’event’ 成功回调  binderror=’event’失败回调  cover-view:可覆盖的对象map/video/canvas/camera/live-plaayer；只能嵌套cover-view/cover-image/button;  movable-area:设置scale-area:true整个area缩放  scroll-view:必须设置一个固定高度；  scroll-x=’false’;  scroll-y=’false’;  upper-threshold:50;距离顶部/左边多远时触发scrolltoupper事件  lower-threshold:50;  scroll-top:120;设置竖向滚动条位置  scroll-left:120;  scroll-into-view=’dataid’值为某个元素的id;  scroll-with-animation=’false’ 是否动画  enable-back-to-top=’false’;滚动条回顶部；  enble-flex=’false’;容器为flex  bindscrolltoupper=’event’ 滚动到顶部右边时触发；  bindscrolltolower=’event’  bindscroll=’event’滚动时触发  swiper  indicator-dots=’false’是否显示点  indeicator-color=’#eee’  indicator-active-color=’red’  autoplay=’false’  current=’index’ 当前所在滑块的index;  interval=’5000’自动切换事件  duration=’500’滑动动画时长  circular=’false’是否衔接动画；  vertical=’false’是否纵向；  previous-magin=’10px’前边距，露出前一项的一小部分；  next-magin=’10px’后边距，露出后一项的一小部分；  display-multiple-items=’1’同时显示滑块的数量；；  skip-hidden-item-layout=’false’是否跳过未显示滑块布局  easing-function=’default’动画滑动类型‘  bindchange=’event’;current改变时触发event.detail={current,source}; autoplay/autoplay  bindtransition=’event’位置改变时触发；event.detail  bindanimationfinish=’event’动画结束时触发；  swiper-itme 宽高自动100%  view  hover-class=’名字’按下时的颜色  hover-stop-propagation=’true’阻止冒泡 hover-start-time=’50’按住多久出现点击状态  hover-stay-time=‘1000’手指松开保留多久  基础容器  icon图标  type=’success,success\_no\_circle,info,warn,waiting,cancel,download,search,clear;    size=’20’icon大小；color=’icon颜色’  progress:进度条  percent=’num 右侧显示0-100  show-info=false右侧显示百分比  border-radius=num/str 圆角大小  font-size=16 右侧字体大小  stroke-width=6进度条宽度  color=red底色  activeColor=blue 已选颜色  backgroundColor=yellow 未选进度条颜色  active=false 进度条从左往右的动画  active-mode=backwards从头播放；forwards接着上一次  bindactiveend=fn 动画完成时间  text文本  selectable=false 文本是否可选  space 显示空格和法制nbsp/emsp中文空格大小/ensp中文空格一般  decod=false 可以解析&nbsp;&lt;&gt;  表单bindinput=’’e.detail.value  button  size=default/mini  type=primary/default/warn  open-type = share/ getUserInfo/ getPhoneNumber/ openSetting  plain=false镂空的按钮  disabled=false 是否禁用  loading=false 名称前是否带有加载图标  form-type 会触发from的submit/reset  open-type 微信开放能力  hover-class=‘pressBtn’ 按下去的样式类  hover-stop-propagation=false 阻止冒泡  hover-start-time=20  hover-stay-time=70;  lang=en,返回信息的语言  session-from =str 回话来源 open-type = contact时有效  send-message-title 回话标题  send-message-path 会话消息卡片点击跳转小程序的路径  send-message-img 会话内消息卡片的图片  app-parameter=’’打开APP时传递的参数；open-type=contact时有效  bindgetuserinfo=fn open=getUserInfo时有效；==wx.getUserInfo()  bindgetphonenumber=fn open-type = getPhoneNumber时有效  binderror=fn 使用开放能时，发生错误时有效 open-type = launchApp;  bindopensetting =fnopen-type=openSetting时有效  bindlaunchapp = fn; 打开app成功的回调；open-type =launchApp;  checkbox  value=str 选中时触发checkebox-group 的change事件  disabled=false  checked=false  color=str  editor  form表单中需要增加name= key;  report-submit=false 是否返回formID  report-submit-timeout=’18000’检车formID是否有效；如失败requestFormId:fail;  bindsubmit=fn e.detail={value:{‘name’:value;},formId:’’}  bindreset=fn  input 文本框  value=str 输入框出事内容  type = text/password  placeholder=str 输入框为空时的占位符  placeholder-style =str placeholder的样式  placeholder-class = str placeholder的样式类  disabled=false  maxlength =num -1不限制长度  cursor-spacing 取光标与键盘的距离和input与键盘距离的最小值  focus=false 获取焦点  confirm-type=str 设置右下角的文字  confirm-hold=false 点击有下家是否键盘不受气  adjust-position =true键盘弹起是否自动上推页面  bindinput=fn e.detail.value  bindfocus=fn  bindblur=fn  bindkeyboardheightchange=fn 键盘高度发生变化是触发  type 合法值text/number/idcard身份证/digit数字键盘  confirm-type合法值send/search/next/go//done  label可以用for找到id也可以直接在标签内  <label> buttom/checked/input/switch</label>目前只此四个  picker底部弹起选择器  mode= selector/multiSelector多列/time/data/gegion省市  bindchange触发  radio 选中时会触发radio-group的change事件携带value  value=str radio的标识  checked =false 是否选中  disabled=false 是否禁用  color =str 颜色  slider滑动选择器  min  max  step 步长  disabled  value当前取值  color  selected-color=srt  activeColor  backgroundColor  block-size =范围12-28  block-color=str 滑块颜色  show-value 是否显示当前值  bindchange =fn完成拖动  bindchanging=fn 拖动触发  switch 开关  checked  disabled  type=switch/checkbox  color  bindchange=fn  textarea 同input 如果实在fixed区域 fixed=true  naviter  camera相机或者二维码  mode=str normal相机/scanCode扫码模式  device-position=back 摄像头朝向；front前置  flash=auto 闪光灯 on/off  frame-size =small/medium/large  bindstop=fn 非正常终止，如退出后台  binderror=fn 不予许使用摄像头时触发  bindinitdone =fn 相机初始化时触发  bindscancode=fn识别二维码成功时触发  image 图片  src=str  mode=str图片剪裁缩放模式  lazy-load=false 图片懒加载  show-menu-by-longpress  binderror=fn  bindload=fn |  |

cover-view