小程序架构和配置

讲师：王红元
微博：coderwhy

小程序交流群

加小撩拿资料

配置小程序

- 小程序的很多开发需求被规定在了配置文件中。
- **为什么这样做呢?**
  - 这样做可以更有利于我们的开发效率;
  - 并且可以保证开发出来的小程序的某些风格是比较一致的;
  - 比如导航栏 – 顶部TabBar，以及页面路由等等。
- 常见的配置文件有哪些呢?
  - project.config.json：项目配置文件, 比如项目名称、appid等;
    - https://developers.weixin.qq.com/miniprogram/dev/devtools/projectconfig.html
  - sitemap.json：小程序搜索相关的;
    - https://developers.weixin.qq.com/miniprogram/dev/framework/sitemap.html
  - app.json：全局配置;
  - page.json：页面配置;

全局配置

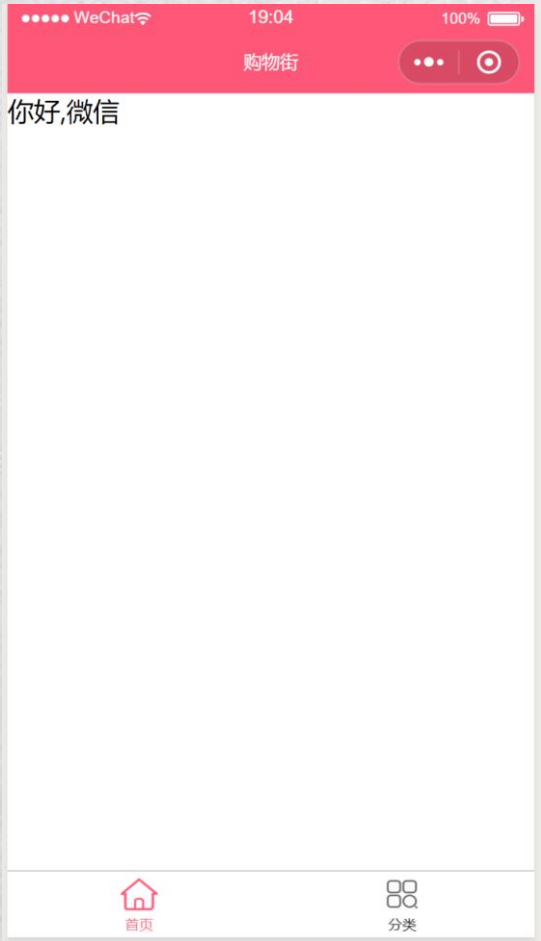
- 全局配置比较多, 我们这里将几个比较重要的. 完整的查看官方文档.
  - https://developers.weixin.qq.com/miniprogram/dev/reference/configuration/app.html

属性	类型	必填	描述
pages	string[]	是	页面路径列表
window	Object	否	全局的默认窗口表现
tabBar	Object	否	底部 tab 栏的表现

- **pages: 页面路径列表**
  - 用于指定小程序由哪些页面组成，每一项都对应一个页面的 路径（含文件名） 信息。
  - 小程序中所有的页面都是必须在pages中进行注册的。
- **window: 全局的默认窗口展示**
  - 用户指定窗口如何展示, 其中还包含了很多其他的属性
- **tabBar: 顶部tab栏的展示**
  - 具体属性稍后我们进行演示

案例实现

- 我们来做如下的效果:

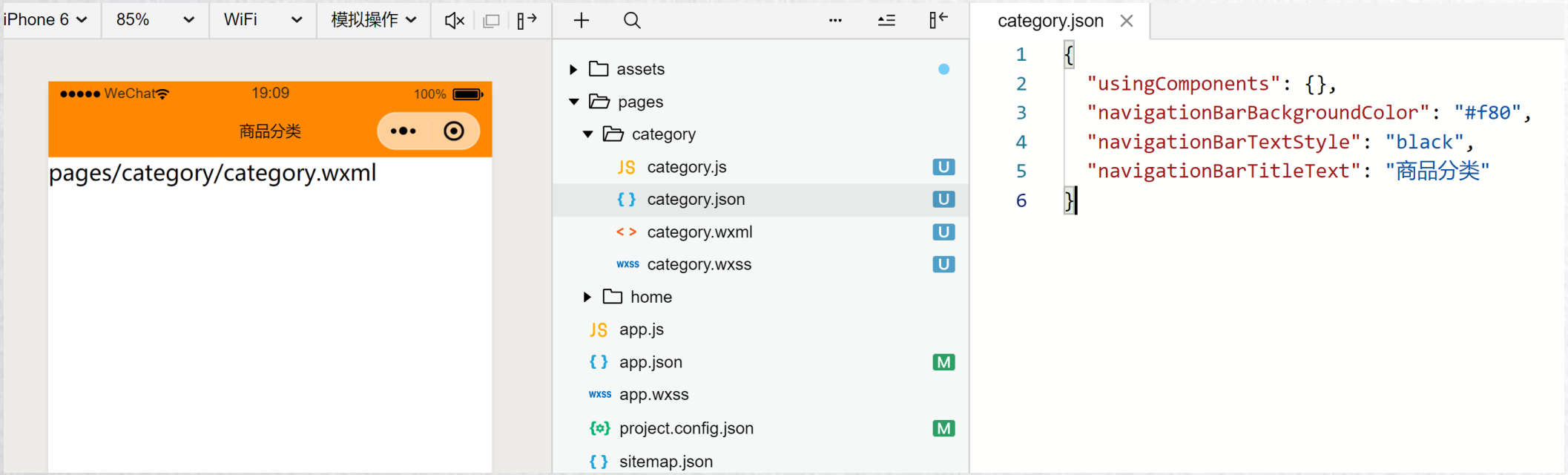


app.json

```
{
  "pages": [
    "pages/home/home",
    "pages/category/category"
  ],
  "window": {
    "navigationBarBackgroundColor": "#ff5777",
    "navigationBarTextStyle": "white",
    "navigationBarTitleText": "购物街",
    "backgroundColor": "#f00",
    "backgroundTextStyle": "dark",
    "enablePullDownRefresh": true
  },
  "tabBar": {
    "selectedColor": "#ff5777",
    "list": [
      {
        "pagePath": "pages/home/home",
        "text": "首页",
        "iconPath": "/assets/images/tabbar/home.png",
        "selectedIconPath": "/assets/images/tabbar/home_active.png"
      },
      {
        "pagePath": "pages/category/category",
        "text": "分类",
        "iconPath": "/assets/images/tabbar/category.png",
        "selectedIconPath": "/assets/images/tabbar/category_active.png"
      }
    ]
  }
```

页面配置

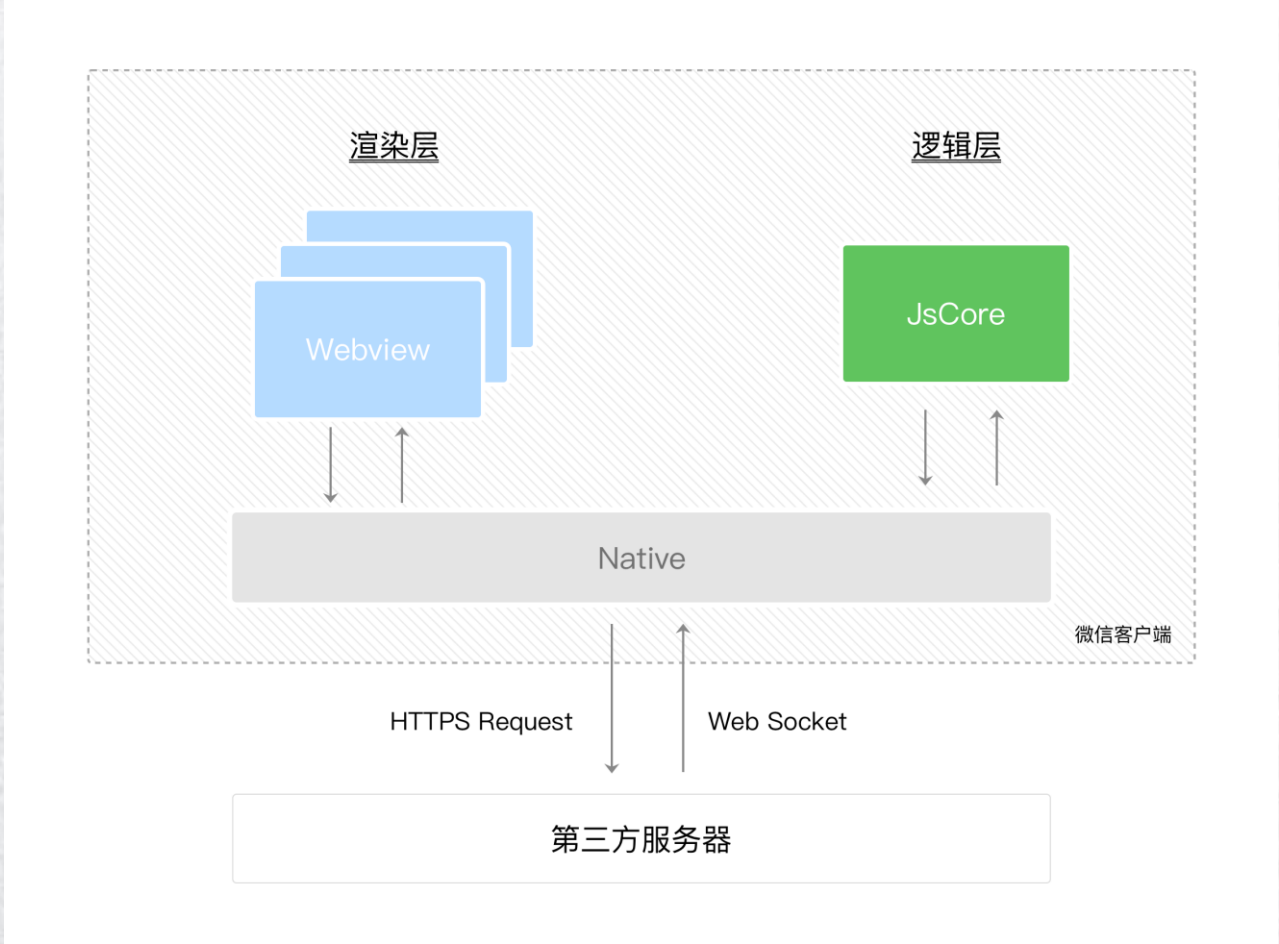
- 每一个小程序页面也可以使用 .json 文件来对本页面的窗口表现进行配置。
  - 页面中配置项在当前页面会覆盖 app.json 的 window 中相同的配置项。



- 关于usingComponents选项, 讲到自定义组建时, 再详细讲解

小程序的双线程模型

- 谁是小程序的宿主环境呢？**微信客户端**
  - 宿主环境为了执行小程序的各种文件：wxml文件、wxss文件、js文件
  - 提供了小程序的双线程模型
- **双线程模型：**
  - WXML模块和WXSS样式运行于 渲染层，渲染层使用WebView线程渲染（一个程序有多个页面，会使用多个WebView的线程）。
  - JS脚本（app.js/home.js等）运行于 逻辑层，逻辑层使用JsCore运行JS脚本。
  - 这两个线程都会经由微信客户端（Native）进行中转交互。
- 下面我们探讨一下如何通过这两个线程渲染出了界面。

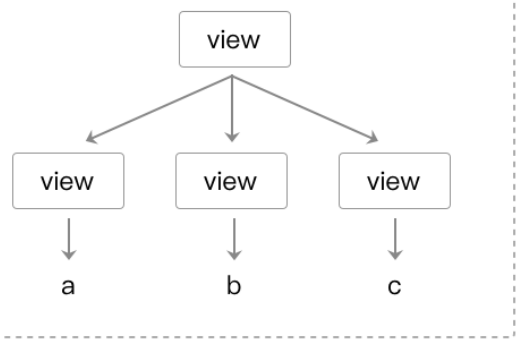


界面渲染过程 – wxml和DOM树

- 首先，我们需要知道，wxml等价于一棵DOM树，也可以使用一个JS对象来模拟（虚拟DOM）

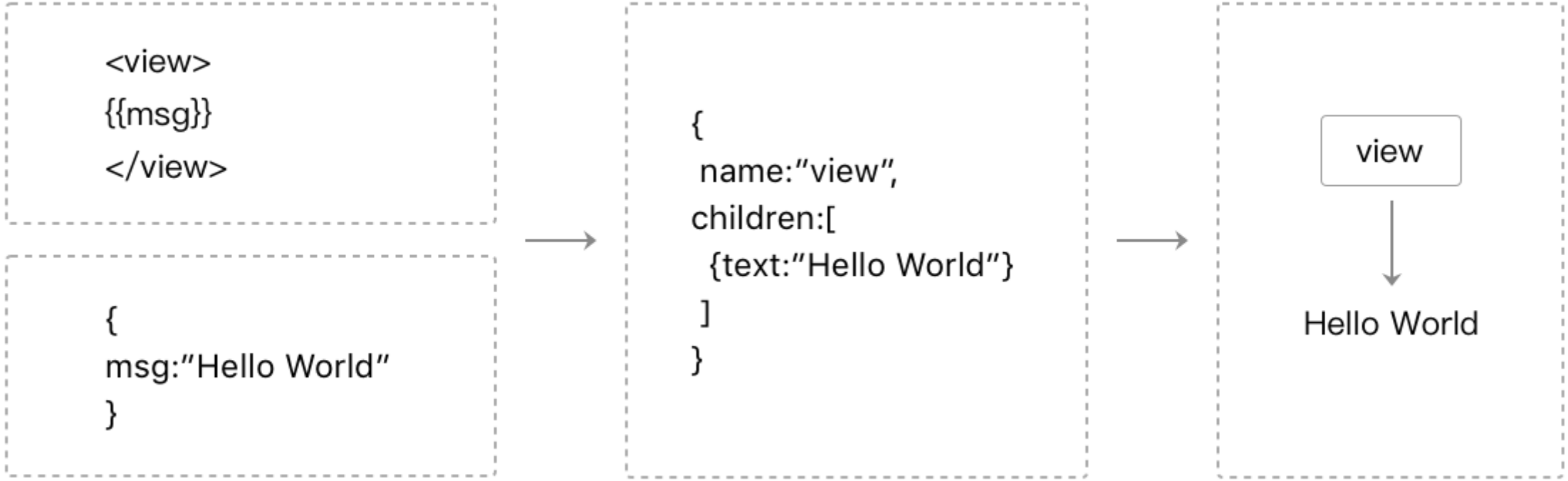
```
<view>
  <view>a</view>
  <view>b</view>
  <view>c</view>
</view>
```

```
{
name:"view",
children:[
  {name:"view",children:[{text:"a"}]},
  {name:"view",children:[{text:"b"}]},
  {name:"view",children:[{text:"c"}]},
  ]
}
```



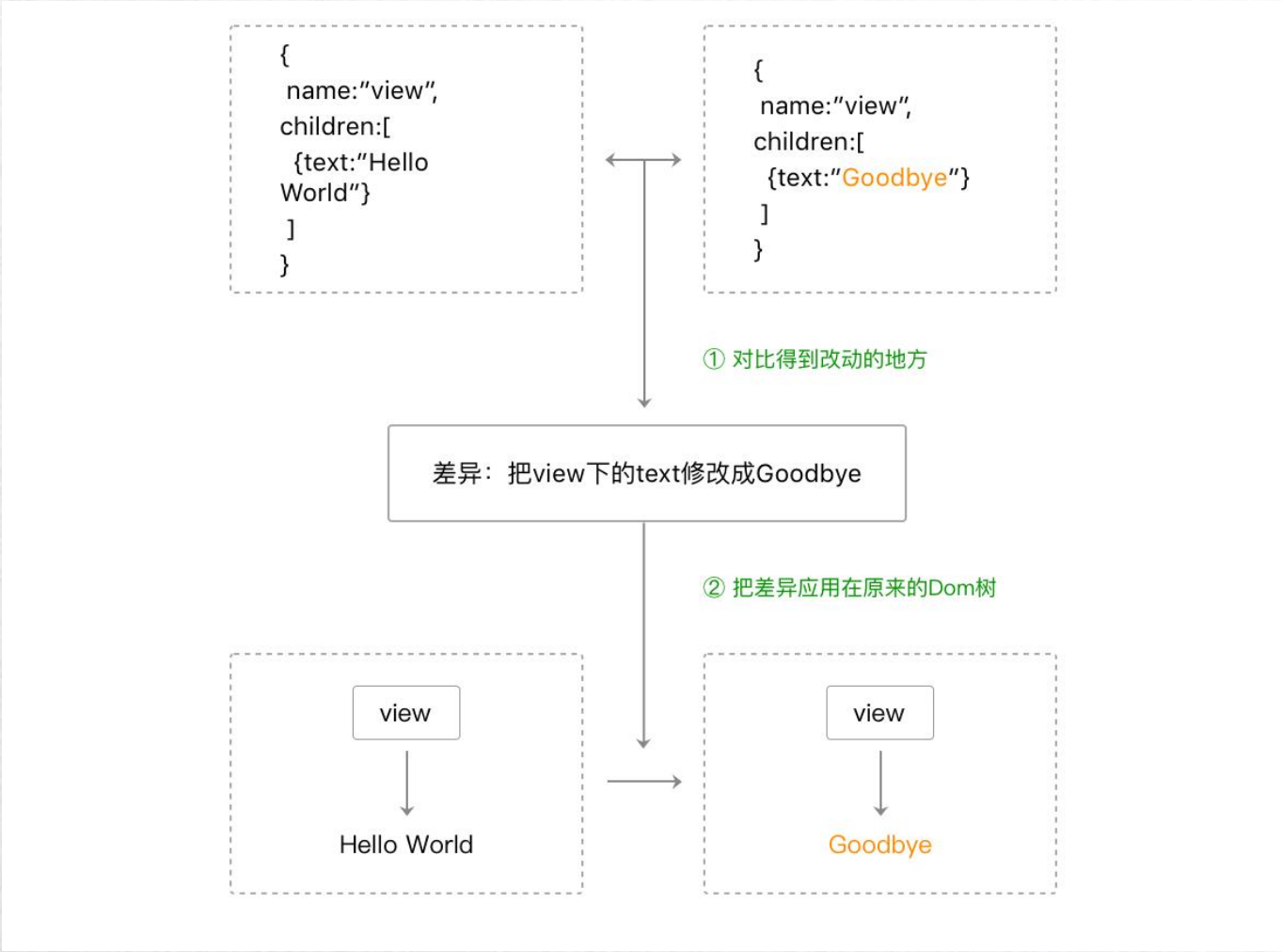
界面渲染过程– 初始化渲染

- 那么，WXML可以先转成JS对象，再渲染出真正的DOM树



界面渲染过程 – 数据发生变化

- 通过setData把msg数据从“Hello World”变成“Goodbye”
  - 产生的JS对象对应的节点就会发生变化
  - 此时可以对比前后两个JS对象得到变化的部分
  - 然后把这个差异应用到原来的Dom树上
  - 从而达到更新UI的目的，这就是“数据驱动”的原理

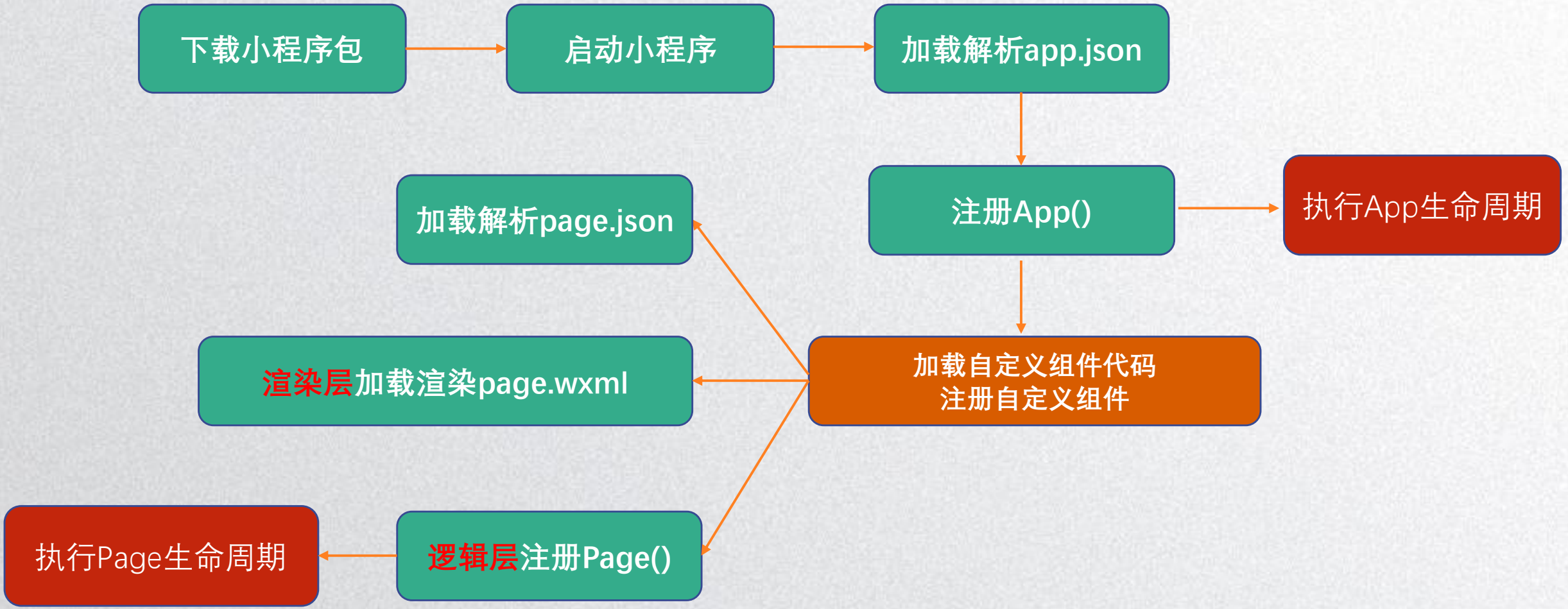


界面渲染的整体流程

- **界面渲染整体流程：**
  - 1.在渲染层，宿主环境会把WXML转化成对应的JS对象；
  - 2.将JS对象再次转成真实DOM树，交由渲染层线程渲染；
  - 3.数据变化时，逻辑层提供最新的变化数据，JS对象发生变化比较进行diff算法对比；
  - 4.将最新变化的内容反映到真实的DOM树中，更新UI；

小程序的启动流程

- 我们一起来看一下一个小程序的启动流程，通过了解小程序的启动流程，我们就知道了自己代码的执行顺序：



注册小程序 – 参数解析

- 每个小程序都需要在 app.js 中调用 App 方法注册小程序示例
  - 在注册时, 可以绑定对应的生命周期函数, 在生命周期函数中, 执行对应的代码.
  - https://developers.weixin.qq.com/miniprogram/dev/reference/api/App.html

属性	类型	默认值	必填	说明
onLaunch	function		否	生命周期回调——监听小程序初始化。
onShow	function		否	生命周期回调——监听小程序启动或切前台。
onHide	function		否	生命周期回调——监听小程序切后台。
onError	function		否	错误监听函数。
onPageNotFound	function		否	页面不存在监听函数。
其他	any		否	开发者可以添加任意的函数或数据变量到 `Object` 参数中，用 `this` 可以访问

注册App时做什么呢?

- **我们来思考**：注册App时，我们一般会做什么呢?
  - 1. 判断小程序的进入场景
  - 2. 监听生命周期函数，在生命周期中执行对应的业务逻辑，比如在某个生命周期函数中获取微信用户的信息。
  - 3. 因为App()实例只有一个，并且是全局共享的（单例对象），所以我们可以将一些共享数据放在这里。

- **小程序的打开场景较多：**
  - 常见的打开场景：群聊会话中打开、小程序列表中打开、微信扫一扫打开、另一个小程序打开
  - https://developers.weixin.qq.com/miniprogram/dev/reference/scene-list.html
- **如何确定场景?**
  - 在onLaunch和onShow生命周期回调函数中,会有options参数，其中有scene值

获取用户信息 – 保存全局变量

- 获取微信用户的基本信息的方式：
  - 1.wx.getUserInfo – 即将废弃的接口；
  - 2.button组件 – 将open-type改成getUserInfo，并且绑定bindgetuserinfo事件去获取；
  - 3.使用open-data组件展示用户信息；


```
<button open-type='getUserInfo' bindgetuserinfo='onGetUserInfo'>授权</button>
<open-data type="userNickName"></open-data>
<open-data type="userGender"></open-data>
<open-data type="userAvatarUrl"></open-data>
```

- 保存全局变量：

```
25    globalData: {
26      name: 'coderwhy',
27      age: 18
28    }
```

注册小程序 - 代码演示

```
App({
  // 当小程序初始化完成时，会触发 onLaunch（全局只触发一次）
  onLaunch: function () {
    console.log('小程序初始化完成：onLaunch')
  },

  // 当小程序启动，或从后台进入前台显示，会触发 onShow
  onShow: function (options) {
    console.log('小程序第一次启动，或者从后台进入前台，onShow')
  },

  // 当小程序从前台进入后台，会触发 onHide
  onHide: function () {
    console.log('小程序从前台进入后台，onHide')
  },

  // 当小程序发生脚本错误，或者 api 调用失败时，会触发 onError 并带上错误信息
  onError: function (msg) {
    console.log('小程序发生错误，onError', msg)
  }
})
```

```
  },
  globalData: {
    title: '全局的title',
    name: 'coderwhy',
    age: 18,
    height: 1.88
  }
```

app.js | home.js | home.wxml

```
1  // home.js
2
3  // 1.获取全局的app对象
4  const app = getApp()
5
6  Page({
7    data: {
8      message: "微信",
9      name: 'kobe'
10   },
11   onClick() {
12     // 2.通过app.globalData.属性的方式获取
13     this.setData({
14       message: app.globalData.title,
15       name: app.globalData.name
16     })
17   }
18 })
19
```

注册页面

- 小程序中的每个页面, 都有一个对应的js文件, 其中调用Page方法注册页面示例
  - 在注册时, 可以绑定初始化数据、生命周期回调、事件处理函数等。
  - https://developers.weixin.qq.com/miniprogram/dev/reference/api/Page.html

属性	类型	默认值	必填	说明
data	Object			页面的初始数据
onLoad	function			生命周期回调—监听页面加载
onShow	function			生命周期回调—监听页面显示
onReady	function			生命周期回调—监听页面初次渲染完成
onHide	function			生命周期回调—监听页面隐藏
onUnload	function			生命周期回调—监听页面卸载
onPullDownRefresh	function			监听用户下拉动作
onReachBottom	function			页面上拉触底事件的处理函数
onShareAppMessage	function			用户点击右上角转发
onPageScroll	function			页面滚动触发事件的处理函数
onResize	function			页面尺寸改变时触发，详见 响应显示区域变化
onTabItemTap	function			当前是 tab 页时，点击 tab 时触发
其他	any			开发者可以添加任意的函数或数据到 `Object` 参数中，在页面的函数中用 `this` 可以访问

注册Page时做什么呢?

- **我们来思考：**注册一个Page页面时，我们一般需要做什么呢?
  - 1.在生命周期函数中发送网络请求，从服务器获取数据;
  - 2.初始化一些数据，以方便被wxml引用展示;
  - 3.监听wxml中的事件，绑定对应的事件函数;
  - 4.其他一些监听（比如页面滚动、上拉刷新、下拉加载更多等）;
- 网络请求和其他一些事件的监听，放在后续再来使用

```
Page({
  // ---------- 初始化数据 ----------
  data: {
    message: '你好啊,李银河'
  },

  // ---------- 声明周期函数 ----------
  onLoad() {
    console.log('页面加载: onLoad')
  },
  onShow() {
    console.log('页面展示: onShow')
  },
  onReady() {
    console.log('页面渲染: onReady')
  },
  onHide() {
    console.log('页面隐藏: onHide')
  },
  onUnload() {
    console.log('页面卸载: onUnload')
  },

  // ---------- 事件监听 ----------
  onClick(e) {
    console.log('按钮被点击')
  }
})
```

Page实例生命周期