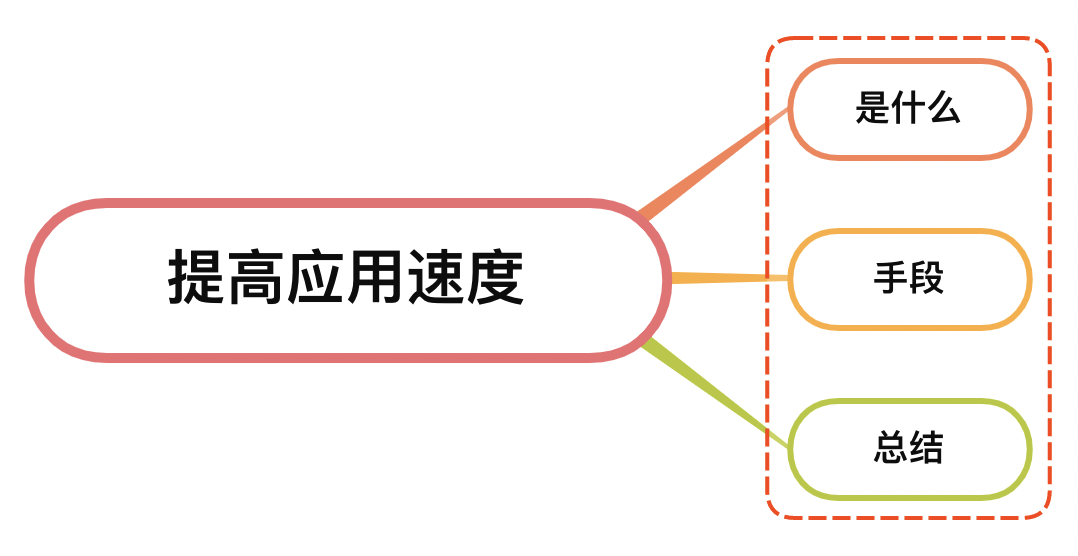
# 面试官：说说提高微信小程序的应用速度的手段有哪些？



## 一、是什么

小程序启动会常常遇到如下图场景：



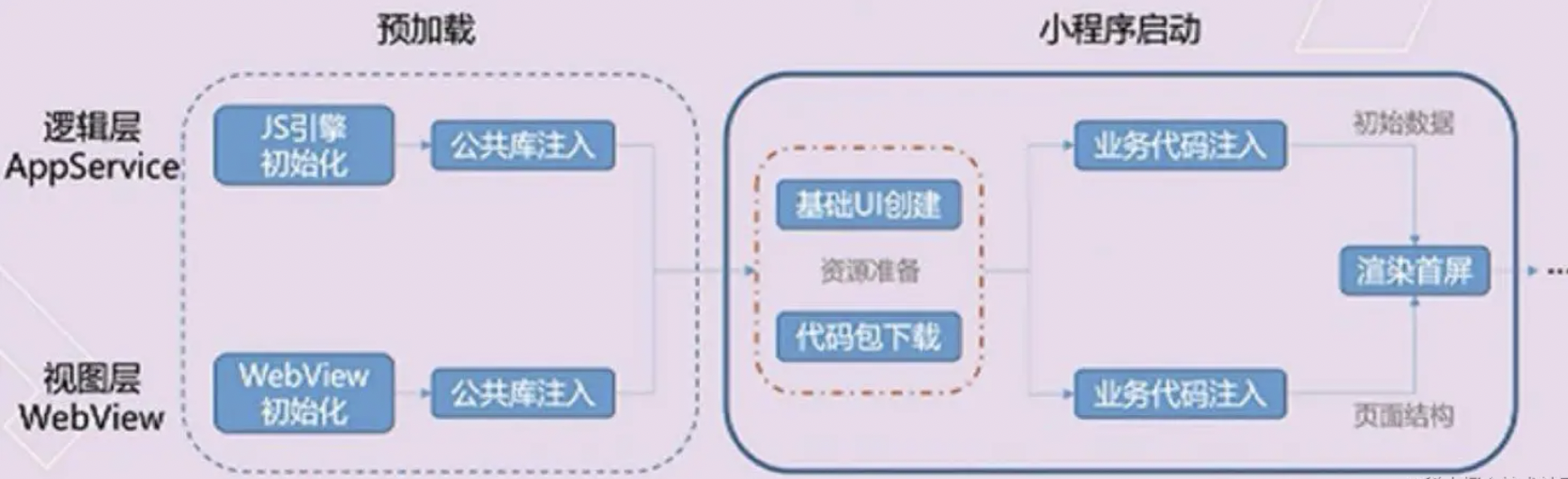
这是因为，小程序首次启动前，微信会在小程序启动前为小程序准备好通用的运行环境，如运行中的县城和一些基础库的初始化

然后才开始进入启动状态，展示一个固定的启动界面，界面内包含小程序的图标、名称和加载提示图标。此时，微信会在背后完成几项工作：

* 下载小程序代码包
* 加载小程序代码包
* 初始化小程序首页

下载到的小程序代码包不是小程序的源代码，而是编译、压缩、打包之后的代码包

整体流程如下图：



## 二、手段

围绕上图小程序的启动流程， 我们可以从加载、渲染两个纬度进行切入：

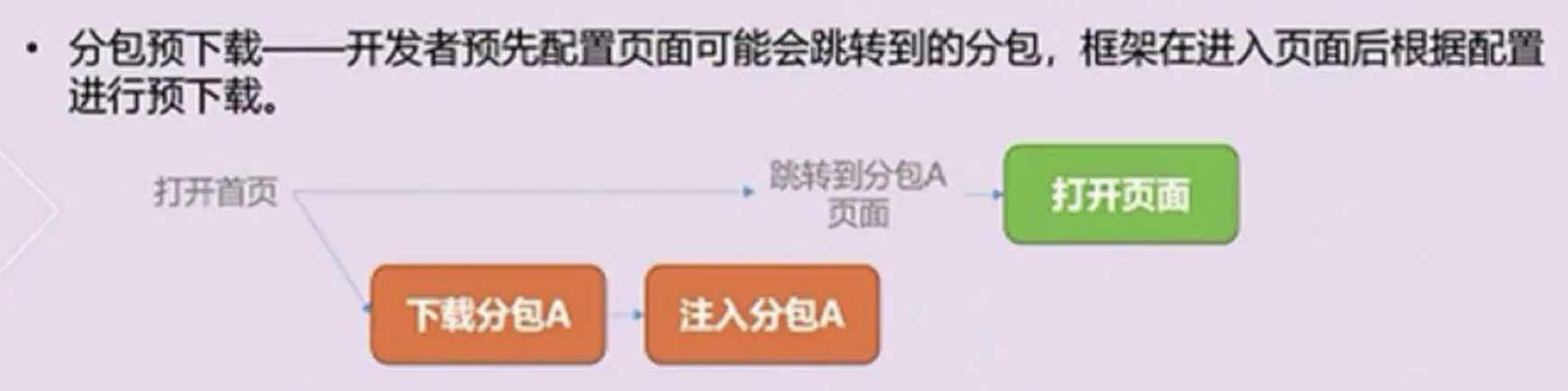
### 加载

提升体验最直接的方法是控制小程序包的大小，常见手段有如下：

* 代码包的体积压缩可以通过勾选开发者工具中“上传代码时，压缩代码”选项
* 及时清理无用的代码和资源文件
* 减少资源包中的图片等资源的数量和大小（理论上除了小icon，其他图片资源从网络下载），图片资源压缩率有限

并且可以采取分包加载的操作，将用户访问率高的页面放在主包里，将访问率低的页面放入子包里，按需加载

当用户点击到子包的目录时，还是有一个代码包下载的过程，这会感觉到明显的卡顿，所以子包也不建议拆的太大，当然我们可以采用子包预加载技术，并不需要等到用户点击到子包页面后在下载子包



### 渲染

关于微信小程序首屏渲染优化的手段如下：

* 请求可以在页面onLoad就加载，不需要等页面ready后在异步请求数据
* 尽量减少不必要的https请求，可使用 getStorageSync() 及 setStorageSync() 方法将数据存储在本地
* 可以在前置页面将一些有用的字段带到当前页，进行首次渲染（列表页的某些数据--> 详情页），没有数据的模块可以进行骨架屏的占位

在微信小程序中，提高页面的多次渲染效率主要在于正确使用setData：

* 不要过于频繁调用setData，应考虑将多次setData合并成一次setData调用
* 数据通信的性能与数据量正相关，因而如果有一些数据字段不在界面中展示且数据结构比较复杂或包含长字符串，则不应使用setData来设置这些数据
* 与界面渲染无关的数据最好不要设置在data中，可以考虑设置在page对象的其他字段下

除此之外，对于一些独立的模块我们尽可能抽离出来，这是因为自定义组件的更新并不会影响页面上其他元素的更新

各个组件也将具有各自独立的逻辑空间。每个组件都分别拥有自己的独立的数据、setData调用

## 三、总结

**小程序启动加载性能**：

* 控制代码包的大小
* 分包加载
* 首屏体验（预请求，利用缓存，避免白屏，及时反馈

**小程序渲染性能**：

* 避免不当的使用setData
* 使用自定义组件

### 参考文献

* https://juejin.cn/post/6969779451177484296
* https://segmentfault.com/a/1190000008925450
* https://juejin.cn/post/6844903638226173965
* https://juejin.cn/post/6844903726939897869