# 《Android互联网架构师课程》

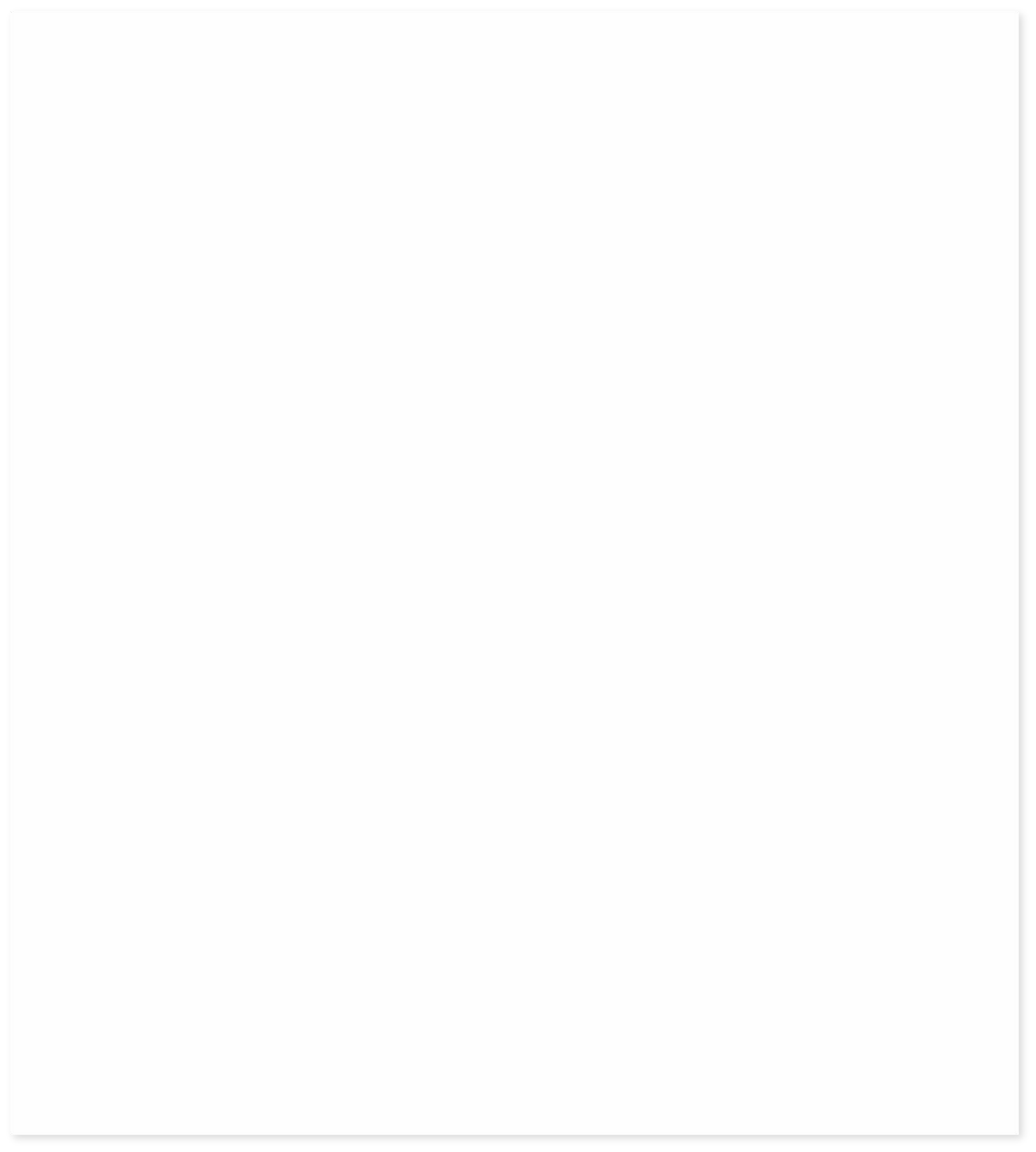
直播VIP课

做一家受人尊敬的企业，做一位受人尊敬的老师

# APK极限压缩与webp应用

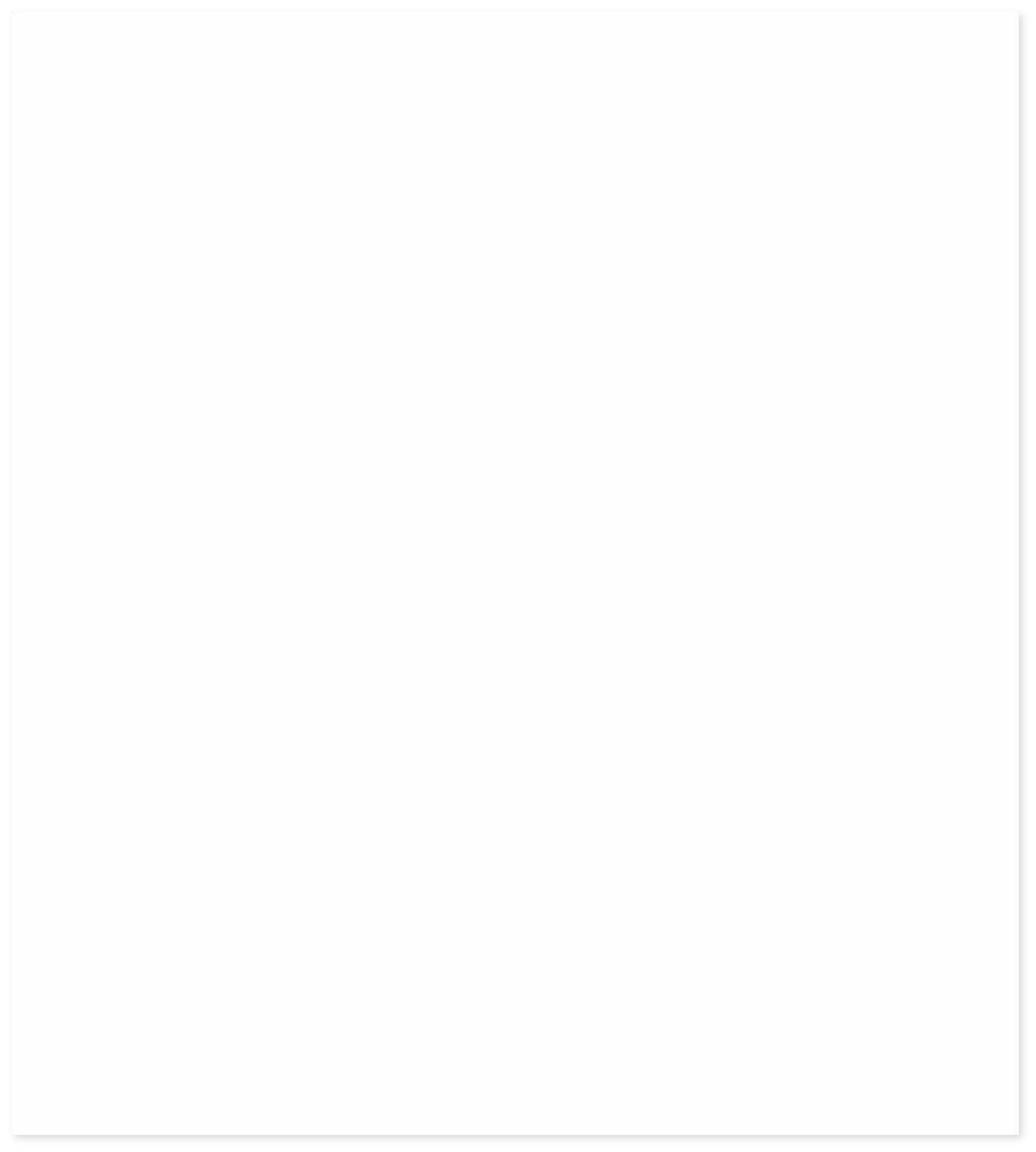
* 1 webp对jpg png的优势 京东为什么会放弃传统图片
* 2 webp编码原理 跟哈夫曼压缩的不同之处
* 3 帧内预测技术详解
* 4 APK极限压缩7大步

**课程安排**



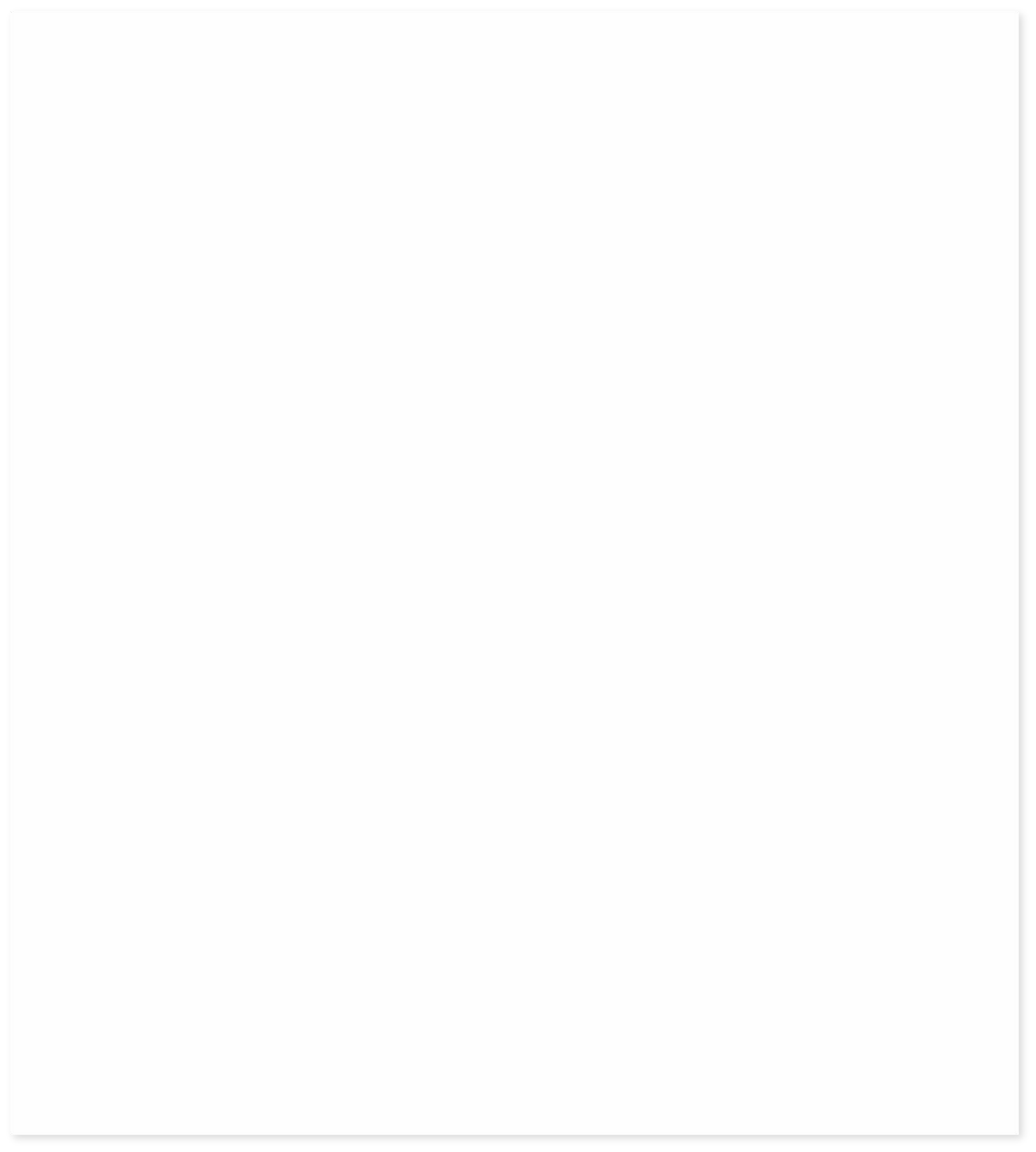
***01***

Webp优势



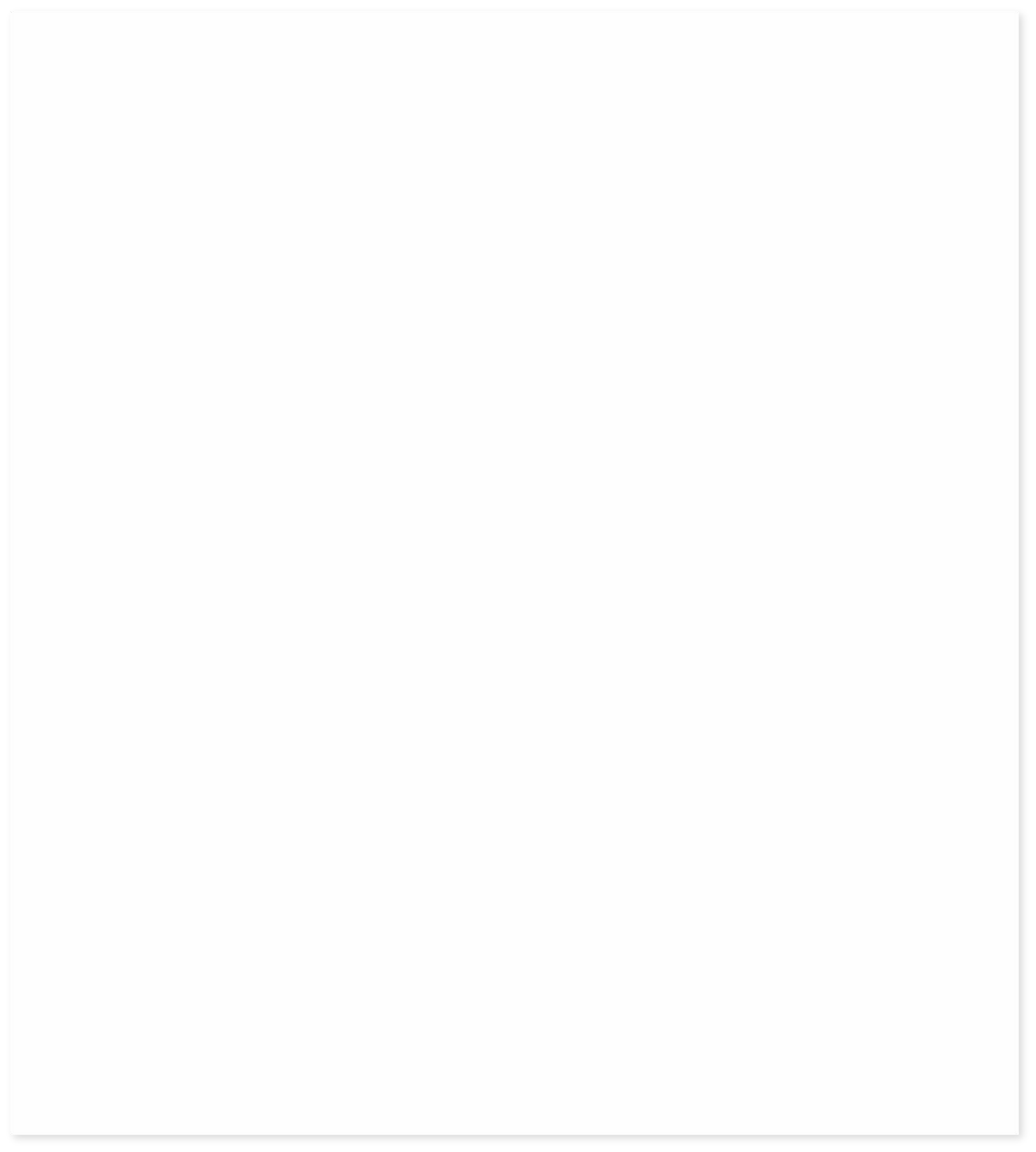
***02***

Webp编码原理



***03***

APK极限压缩



***04***

WebP在Apk的应用

优势体现在哪里



webp编码对于哈夫曼压缩性能更优异些。

哈夫曼与webp本质上都是从编码来解决图像压缩哈夫曼是对rgb的元数据进行变频压缩

webp编码是通过预测技术对图片压缩

WebP 压缩使用的图像编码方式与 VP8 视频编码对关键帧压缩方式相同换句话解释：Google将视频编码技术搬到了图片上 形成了Webp编码格式

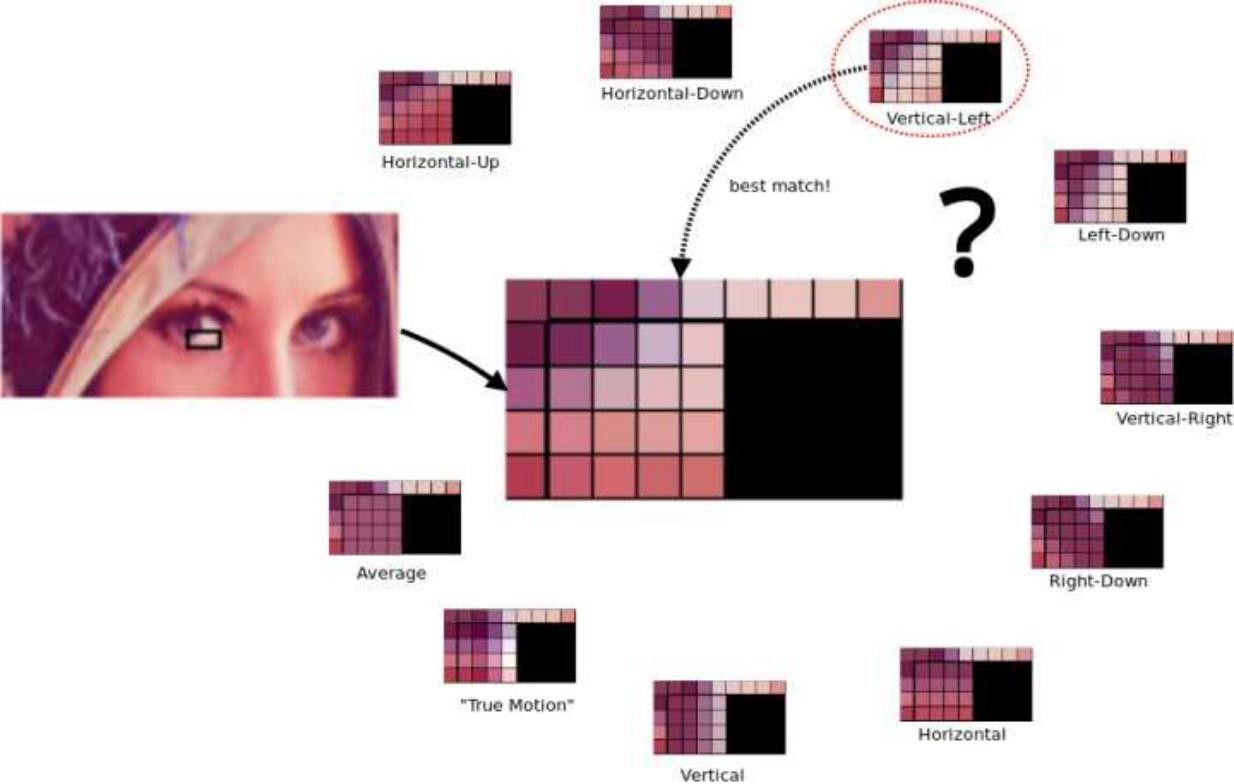
WebP编码流程

WebP 会将图片划分为两个 8x8 色度像素宏块和一个 16x16 亮度像素宏块。在每个宏块内， 编码器基于之前处理的宏块来预测冗余动作和颜色信息。

通过图像关键帧运算，使用宏块中已解码的像素来绘制图像中未知部分 通过预测模式 去除冗余数据，实现更高效的压缩。

* H\_PRED（水平预测）： 用宏块左边的列 L 的填充块的每一列；
* V\_PRED（垂直预测）： 用宏块上边的行 A 的填充宏块的每一行；
* DC\_PRED（DC预测）： 用行 A 和列 L 的像素的平均值作为宏块唯一的值来填充宏块；
* TM\_PRED（TrueMotion预测）： 除了行 A 和列 L 之外，用宏块上方和左侧的像素P、A

（从P开始）中像素块之间的水平差异以列 L 为基准拓展每一行。



我们都知道播放视频需要下载播放器，播放器的作用是视频文件进行解码。所以每一个播放器都会包含解码器

WebP使用的是视频编码 难道需要**下载视频编码器**

其实是不需要的，很多厂商已经支持了webp的解码，可以说webp就像png 一样普遍。



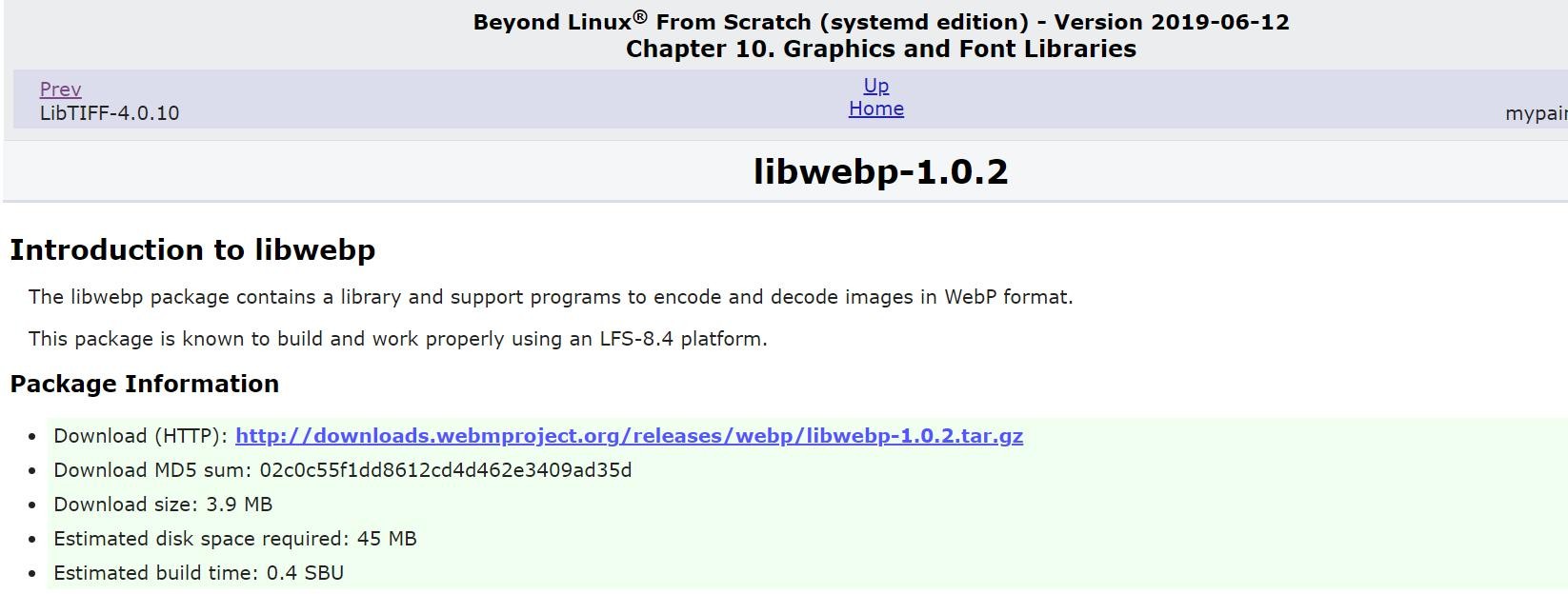
**| Android高手课 | 性能优化**

**可是？**

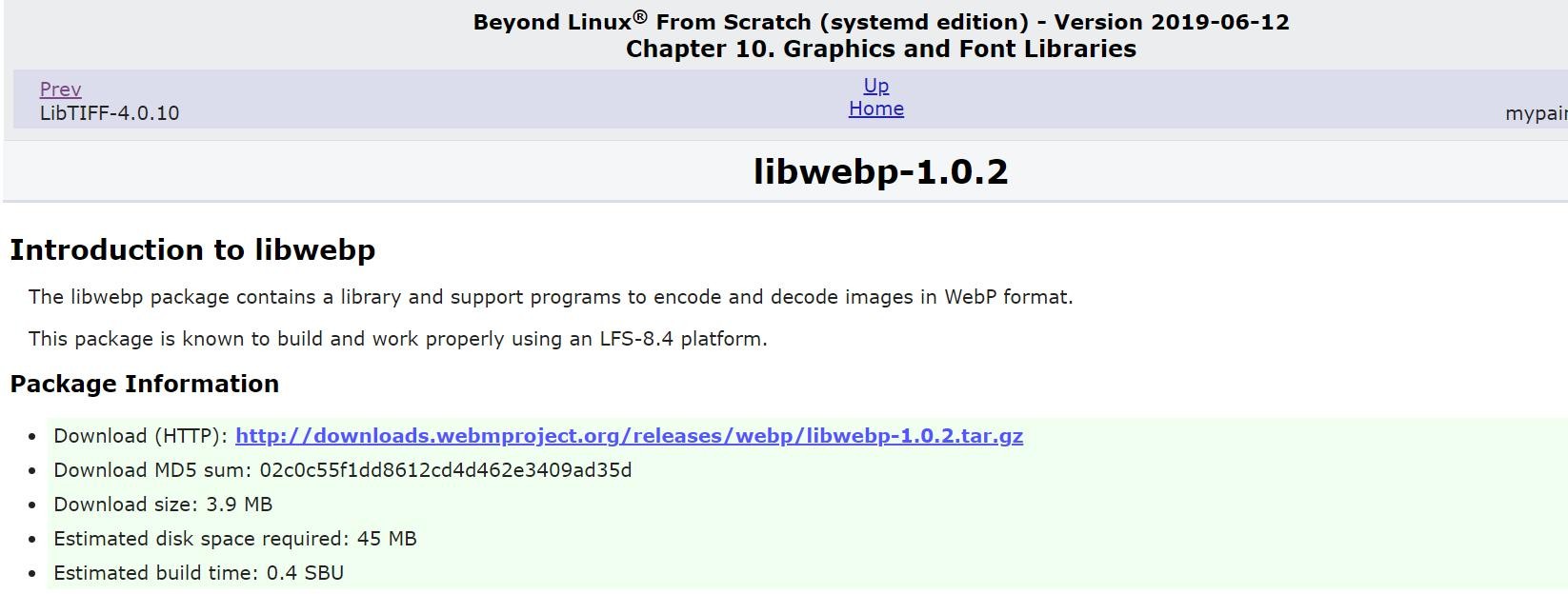
**在早些年代 那个时候还没有webp，**

**比如Android4.3之前 他们是不支持webp的**

百度搜索libwebp 下载libwebp

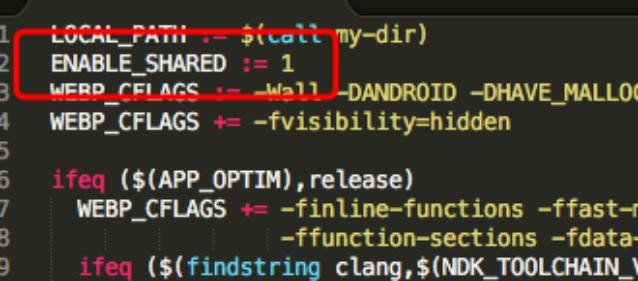


百度搜索libwebp 下载libwebp

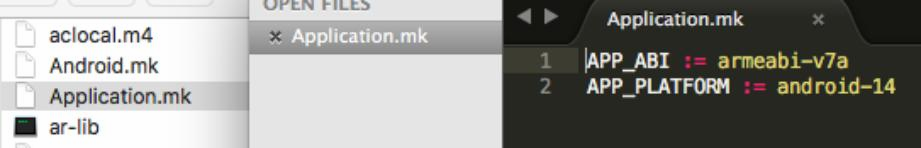


[下载地址:http://downloads.webmproject.org/releases/webp/libwebp-1.0.2.tar.gz](http://downloads.webmproject.org/releases/webp/libwebp-1.0.2.tar.gz)

1. Android.mk增加ENABLE\_SHARED := 1
2. 添加(jar包调用需要 用到jni接口) swig/libwebp\_java\_wrap.c \

1. 创建Application.mk



我们都知道播放视频需要下载播放器，播放器的作用是视频文件进行解码。所以每一个播放器都会包含解码器

WebP使用的是视频编码 难道需要**下载视频编码器**

其实是不需要的，很多厂商已经支持了webp的解码，可以说webp就像png 一样普遍。



**可是？**

**在早些年代 那个时候还没有webp，**

**比如Android4.2之前 他们是不支持webp的**

**| Android高手课 | 性能优化**

* + 减少webp解码时间 可以超过 jpg 和png解码的时间
  + 降低编码的时间，比原生的编码小号时间更短 30ms
  + 增加兼容性，兼容所有型号的手机，4.3以下也是兼容的 播放器

**| Android高手课 | 性能优化**



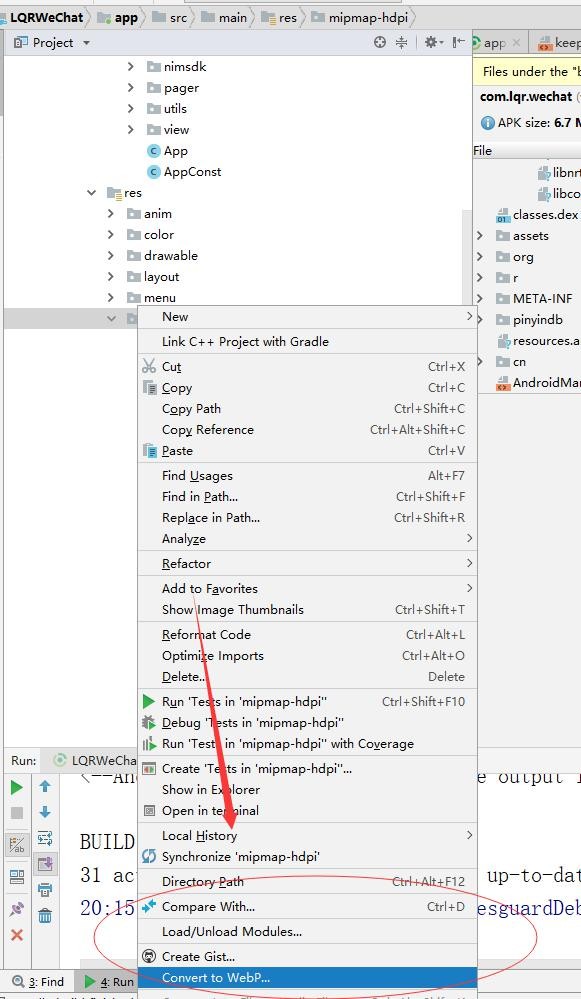
# APK瘦身七大步

**| Android架构师**课程

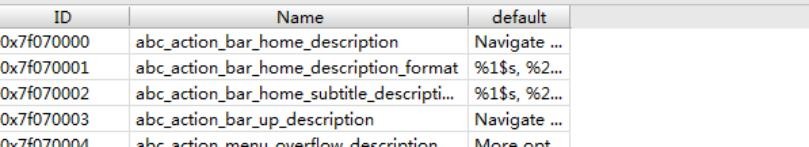
**| Android高手课 | 性能优化**



1、点击文件夹 进行全部转换



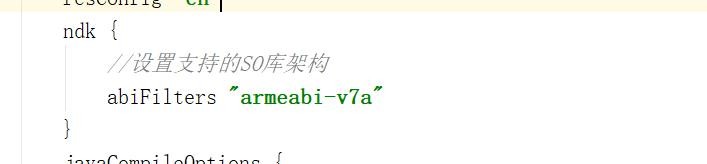
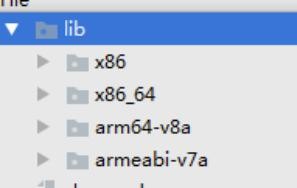
1、由于第三方库，如appcompat-v7的引入，库中包含了大量的国际化资源， 根据情况通过配置删除。



如果项目中包含第三方SDK或者自己使用了ndk，

如果不进行配置会打包全cpu架构的动态库进入apk。

对于真机，只需要保留一个armeabi(armeabi-v7a)就可以了。



进入源码LayoutInflater源码分析：

**LayoutInflater.inflate(xml);**

**void** rInflate(XmlPullParser parser, ……){

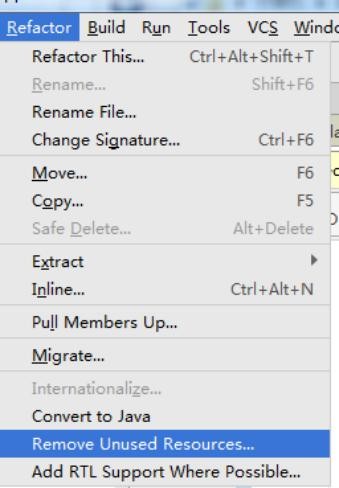
…… **else** {

**final** View view = createViewFromTag(parent, name, context, attrs); **final** ViewGroup viewGroup = (ViewGroup) parent;

**final** ViewGroup.LayoutParams params = viewGroup.generateLayoutParams(attrs); rInflateChildren(parser, view, attrs, **true**);

viewGroup.addView(view, params);

}}

1、一键移除,如果出现使用动态id使用资源会出现问题(不建议)

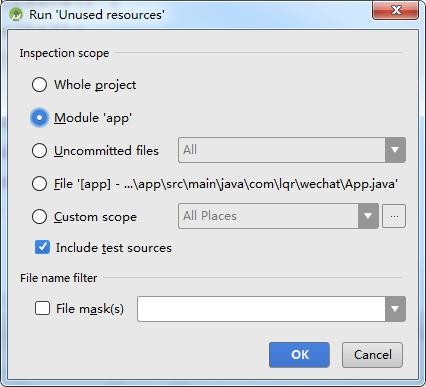
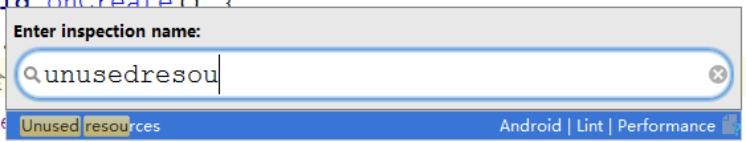
int indetifier =getResources().getIdentifier("img\_bubble\_receive", "drawable", getPackageName()); getResources().getDrawable(indetifier);

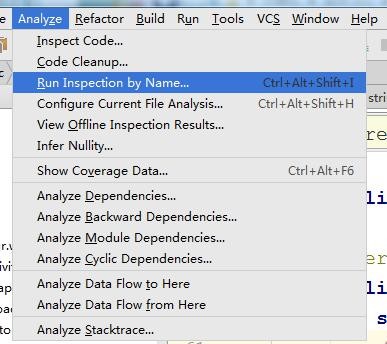
动态获取资源id,未直接使用R.xx.xx ，则这个id代表的资源会被认为没有使用过(类似不能混淆反射类)

* 什么是 Lint

Lint 是Android Studio 提供的 代码扫描分析工具，它可以帮助我们发现代码结构/质量问题，同时提供一些解决方案，而且这个过程不需要我们手写测试用例。

代码迭代版本一多，很容易会遗留一些无用的代码、资源文件，我们可以使用 Lint 进行清除。





int indetifier =getResources().getIdentifier("img\_bubble\_receive", "drawable", getPackageName()); getResources().getDrawable(indetifier);

动态获取资源id,未直接使用R.xx.xx ，则这个id代表的资源会被认为没有使用过(类似不能混淆反射类)

* 什么是 混淆

Android代码混淆，又称Android混淆，是一种Android APP保护技术，用于保护APP不被破解和逆向分析。



ProGuard的三大作用

压缩

移除未被使用的类、属性、方法等，并且会在优化动作执行之后再次执行（因为优化后可能会再 次暴露一些未被使用的类和成员。

优化

优化字节码，并删除未使用的结构。混淆

将类名、属性名、方法名混淆为难以读懂的字母

**| Android高手课 | 性能优化**

* shrinkResources = true

shrinkResources 用来开启压缩无用资源，也就是没有被引用的文件（经过实测是

drawable,layout，实际并不是彻底删除，而是保留文件名，但是没有内容，等等），但是因为需要知道是否被引用所以需要配合mififyEnable使用，只有当两者都为true的时候才会起到真正的删 除无效代码和无引用资源的目的

与去除无用资源不同的是，比如 某个java类没有用到，被混淆时删除了，而该类引入了layout资源 。此时会将这个资源也压缩掉

* 如何开启严格模式

**<?xml version="1.0" encoding="utf-8"?>**

**<resources xmlns:tools="**[**http://schemas.android.com/tools**](http://schemas.android.com/tools)**" tools:shrinkMode="strict" />**

* 如何手动保留xml

## <?xml version="1.0" encoding="utf-8"?>

**<resources xmlns:tools="**[**http://schemas.android.com/tools**](http://schemas.android.com/tools)**" tools:keep="@layout/base\_\*" tools:discard="@layout/unused2" />**

* 什么是 AndResGuard

 AndResGuard是一个缩小APK大小的工具，它的原理类似Java Proguard，但是只针对资源。它会将原本冗长的资源路径变短，例如将res/drawable/wechat变为r/d/a。

* 为什么 使用AndResGuard

在以往的开发中，我们通常只混淆了代码，资源文件却暴露在他人面前，res文件夹下所有文件 名的可读性过强

  微信的开源库AndResGuard 7zip压缩正好解决这种问题，对资源进行混淆，保护res 资源文件的可读性，同时，可以减少APP的大小

此时，我们根本无法知道哪个文件夹是存放布局文件的，哪些文件夹是存放图片的，

即使找到存放xml文件的文件夹，我们也难以知道这些xml是干嘛用的

**| Android高手课 | 性能优化**



**谢谢观看**