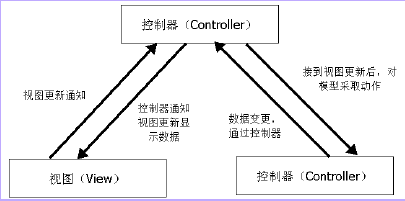
**3Android高薪之路**

1. Android的认识
2. 为什么看好Android？
3. Android是开源的移动操作系统，开发成本低；
4. Android开发商Google实力雄厚；
5. Android是目前市场占有率最高的移动操作系统；
6. Android的支持者众多，由全球34家知名企业组成开放手机联盟。
7. Android的特点：
8. 开源操作系统，开发成本低；
9. 完善的应用程序框架，支持4大组件（activity、service、content provider和broadcast receiver）；
10. 程序在基于linux的Dalvik虚拟机上运行，速度更快；
11. 内置轻量级的SQLite和WebKit核心浏览器（支持HTML5）；
12. 众多硬件传感器和OpenGL绘图功能。
13. MVC模式：使程序中的数据和界面分离。

原理：controller连接view和model。View中数据变化时会通知controller，controller收到通知后让model采取相应操作。如果model中的数据发生变化会通知controller，然后controller会通知view重新显示数据。

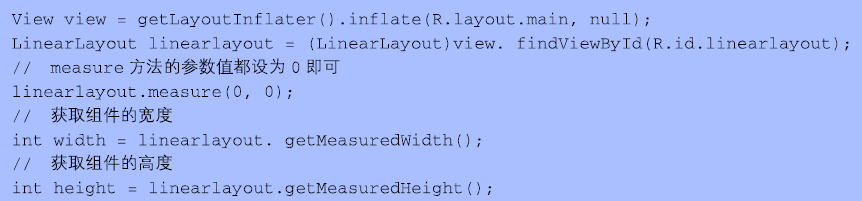
Android中的listview就是MVC模式，V代表组件，M代表数据源及操作数据库，C代表adapter类。



1. Android系统框架：Linux内核层、系统运行库（Android运行时）、应用框架层、框架层。
2. 搭建Android开发环境需要哪些东西？（答：JDK、SDK、eclipse和ADT）
3. 与Android版本相关的内容保存在<Android SDK安装目录>/platforms中。
4. adb服务重启的方法：
5. 单击DDMS的Devices视图右上角的“Reset adb”按钮；
6. 重启模拟器；
7. 在任务管理器中结束adb.exe进程，并重启模拟器；
8. 重启eclipse或重启操作系统。
9. Android布局
10. Android的五种布局：FrameLayout（堆栈布局）、LinearLayout（线性布局）、RelativeLayout（相对布局）、TableLayout（表格布局）和AbsoluteLayout（绝对布局）。
11. <LinearLayout xmlns:android=[*http://schemas.android.com/apk/res/android*](http://schemas.android.com/apk/res/android)*>*

xmlns:android是xml的命名空间，防止属性冲突

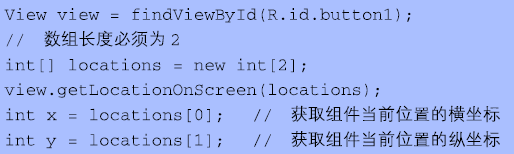
1. 自定义组件时，需要使用<declare-styleable>标签定义属性名。
2. FrameLayout主要用于层次结构布局。类似于PhotoShop中的图层。
3. LinearLayout布局中如果android: orientation为horizontal（vertical），则android: layout\_gravity属性只有在垂直（水平）方向起作用。
4. 测量LinearLayout的高度和宽度：



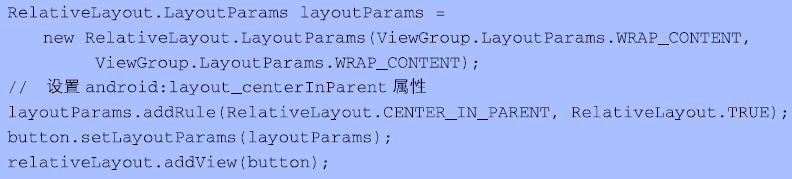
1. 设置分隔线的方法android:showDeviders和android:divider属性。
2. android:gravity用于设置View中内容相对于View组件的对齐方式；

android:layout\_gravity用于设置View相对于Container（父view）的对齐方式。

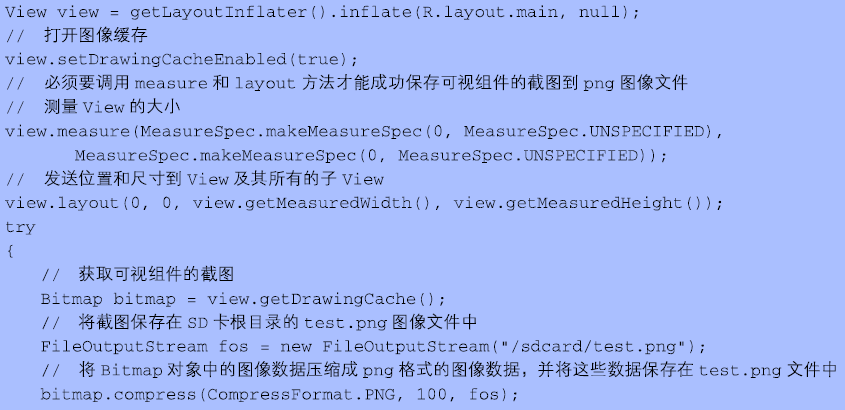
1. 获取当前view相对于屏幕的坐标：



1. 用java代码设置android属性：

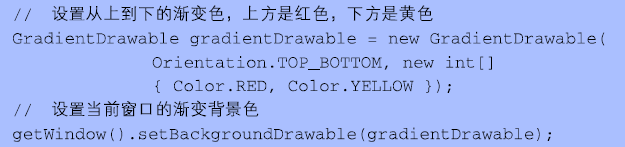


1. TableLayout的用法：<TableLayout>中嵌套<TableRow>标签，<TableRow>标签中有若干个view。
2. <TableLayout>标签中的stretchCloumns主要用于拉伸的列的索引（从0开始）。
3. 将当前的可视组件以同样的相对位置和大小保存在png图像文件中：





1. 将Android应用程序窗口的背景色设置成渐变色：



1. android:padding是view中的内容四个方向距离边缘的距离；

android:margin是view距离其他view的距离。

1. **px**：像素； **pt**：1pt＝1/72英寸；

**dp**：与密度无关的像素； **sp**：与比例无关的像素。

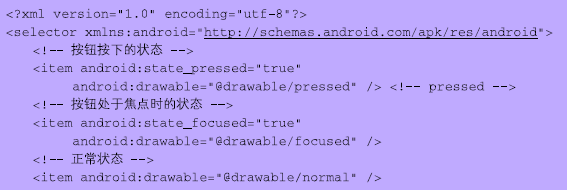
1. 用<include>标签引用其他的布局文件，用android:id覆盖被引用文件顶层节点的android:id的属性值。



1. 通过对apk文件进行反编译处理（如AXMLPrinter2工具）后变成可以阅读的文件。
2. 组件
3. @+id/属性值：当前属性引用的属性没有事先在R类中定义；

@id/属性值：当前属性引用的属性已经在R类中定义。

1. View类的layout方法可以改变组件的大小和位置。Layout的四个参数分别表示vi组件的左上角顶点和右下角顶点。
2. 让一个组件在不同状态显示不同的图像：

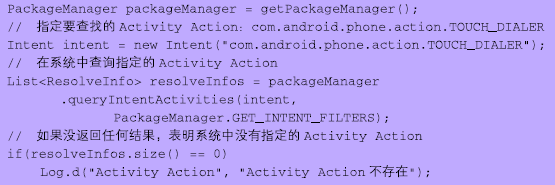




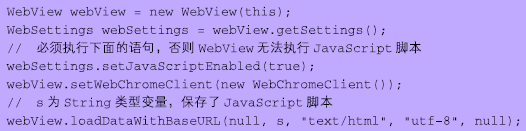
1. BaseAdapter类有四个抽象方法：getItem()、getItemId()、getCount()、getView()
2. Android SDK支持的容器组件：FrameLayout、LinearLayout、RelativeLayout、TableLayout、AbsoluteLayout和GridView、Gallery（用于循环显示）、ListView。
3. 静态引用：将组件同主程序一起封装在编译后的目标文件中，如VC中的lib文件。

动态引用：主程序和组件分离，组件可单独升级和卸载，如Windows的dll和Linux的.so文件。

1. 四大应用程序组件使不同应用程序之间进行交互，需要在AndroidManifest.xml文件中进行注册。
2. Android应用程序可以动态引用classes.dex（Dalivk虚拟机格式）的jar文件，但是不能引用.class（JVM格式）的jar文件。
3. 判断包含某个Activity Action的Android应用程序是否安装：（适用于四大组件）

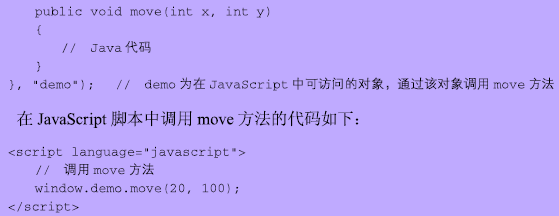


1. Java代码执行JavaScript脚本：

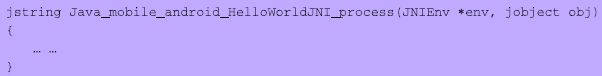


JavaScript中调用Java方法：





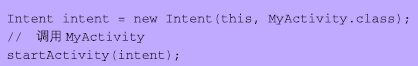
1. 如果调用NDK方法的Java类是mobile.android.HelloWordJNI，在该类中实际访问的本地方法名是process，返回值类型为String，没有参数，则NDK方法的定义为：



NDK方法必须以Java开头，加上调用NDK方法的Java全类名，再加实际方法名。各部分之间用下划线分隔，NDK方法的前两个参数类型必须是JNIEnv\*和jobject。JNIEnv\*表示当前NDK环境的对象指针，可通过该参数方位NDK中的内置成员；jobject表示调用当前NDK方法的JAVA对象，用该参数可以访问Java对象的成员。

1. Activity生命周期：onCreate()、onStart()、onResume()、onPause()、onStop()、onDestory()、onRestart()。
2. 启动Activity的方法：显式调用和隐式调用。

显示调用：

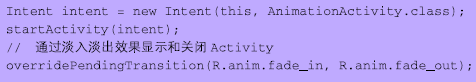


隐式调用：（通过Activity Action来调用）



1. Activity之间传递数据：Intent对象、静态变量、全局对象（Application）和剪切板。
2. 将activity变成对话框：在AndroidMainfest.xml中注册activity时，指定android:theme=”@android:style/Theme.Dialog”
3. 设置activity显示和关闭时的动画效果:

先在res/anim中建立动画资源文件，然后通过overridePendingTransition方法设置。



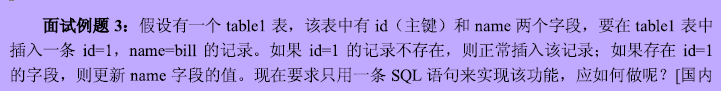
1. 在广播接收器中启动activity要设置flag为Intent.FLAG\_ACTVITY\_NEW\_TASK。
2. 开启service的方法
3. startService：onCreate() –> onStart()，stopService：onDestroy();
4. bindService：onCreate() –> onBind()，unbindService：onUnbind() –> onDestory()。
5. 对话框、信息提示和菜单
6. 标准对话框AlertDialog最多有3个按钮。AlertDialog类显示3个按钮的方法分别为setButton、setButton2、setButton3。AlertDialog.Builder类提供更直观的3个显示按钮方法：setPositiveButton、setNeutralButton、setNegativeButton。
7. Android中显示对话框的方式：
8. AlertDialog显示对话框；
9. 将activity的主体设置为Theme.Dialog



1. 使用Activity.showDialog方法
2. 对话框是独占式的，只有在关闭后才能做其他操作，而toast和notification不会影响用户进行其他操作。
3. 在状态栏显示Notification的实现步骤：



1. 文件存储
2. Android支持的数据存储：SharedPreferences、SQLite数据库、Content Provider、文件存储、网络存储。
3. 获取SharedPreferences对象的三种方法：
4. Context类中的getSharedPreferences()方法；
5. Activity类中的getPreferences()方法；
6. PreferenceManager类中的getDefaultSharedPreferences()方法。
7. SharedPreferences实现文件存储的三步骤：
8. 调用SharedPreferences对象的edit()方法来获取SharedPreferences.Editor对象。
9. 向SharedPreferences.Editor对象中添加数据，比如添加一个布尔型数据就使用putBoolean方法，添加一个字符串则使用putString()方法，以此类推。
10. 调用commit()方法将添加的数据提交，从而完成数据存储操作。
11. SharedPreferences文件保存在/data/data/包名/shared\_prefs中。
12. 解析XML格式数据的方法：Pull解析、SAX解析和DOM解析。
13. 数据库



replace into table(id, name) value(1, ‘bill’);

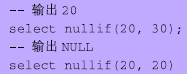
1. 将table1表的表结构和数据复制到table2中

create table table2 as select \* from table1;

1. 获取最近一次执行SQL语句影响的记录行数：



1. 返回第一个不为null的字段值，可以用coalesce()。
2. 



1. typeof()判断当前字段的数据类型。



1. stringObject.substring(start,end)用于提取字符串中介于start和end之间的字符；

stringObject.substr(start [,length])用于返回从start开始的指定长度的子字符串。

1. 数据库框架类SQLiteOpenHelper类。
2. 在绑定数据时，Cursor对象返回的记录集必须有一个“\_id”字段。
3. Android打开或创建数据库的三种方式:SQLiteOpenHelper、Context.openOrCreateDatabase和SQLiteDatabase.openOrCreateDatabase。
4. 网络与加密
5. Socket对象至少要指定IP（域名）和端口号。

客户端：

服务端：

1. 引起网络连接关闭的4中情况：
2. Socket.close()；
3. InputStream和OutputStream中有一个关闭；
4. 程序退出时网络连接关闭；（不会立马关闭）
5. Socket为null或Socket未释放被新的Socket对象覆盖。（不会立马关闭）
6. 判断Socket对象是否处于关闭状态：Socket.isClosed()；

判断Socket对象是否曾今成功连接：Socket.isConnected()；

判断Socket对象是否正处于连接状态：isClosed为false，isConnected为true。

1. 使用Socket.connect方法连接服务器：



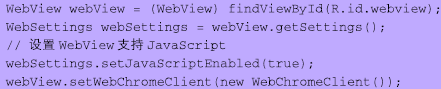
1. 客户端Socket超时：连接超时和读取数据超时。
2. 确保调用Socket.close方法后，将所有的数据都放松到服务器端：



1. 阻塞IO：在IO通信中只要某一步的通信为结束，就无法进行其他的通信。

非阻塞IO：暂时无法获取数据时，非阻塞IO立即返回，一遍执行其他代码，然后系统会不断侦测未完成的读取操作，直到可继续读取数据再完成此操作。

1. WebView是一个使用WebKit引擎的浏览器控件，支持HTML、CSS、JavaScript及在JavaScript调用Java代码。
2. Java中执行JavaScript脚本



JavaScript调用Java方法：





**注**：JavaScript是通过异步方式调用Java方法，所以在Java中不能直接方位UI线程。

1. Android两种直接访问HTTP资源的方式：DefaultHttpClient和HttpURLConnection。也可以用socket发送Http请求。

DefaultHttpClient通过HttpGet/HttpPost分别封装GET和POST。

HttpURLConnection通过URL的构造方法指定Http资源地址，然后通过URL.openConnection方法获取HttpURLConnection对象。

1. HTTPS应用了安全套接字（SSL）作为HTTP应用层的子层。
2. 对称加密：加密和解密都是使用同一个key；

非对称加密：使用public key对数据加密，使用private key对数据解密。

1. Android支持三种不可逆的加密算法：MD5、SHA、HMAC，支持一种可逆加密算法：AES。
2. 蓝牙、WiFi、电话、短信、联系人、多媒体
3. 开启蓝牙的两种方式：Activity Action和使用BluetoothAdapter(要在AndroidMainfest.xml中注册)。
4. 蓝牙搜索过程是异步，通过蓝牙socket连接，蓝牙的两种状态： BluetoothDevice.ACTION\_FOUND和BluetoothDevice.ACTION\_DISCOVERY\_FINISHED。
5. 始终呼叫转移：呼叫转移（\*\*21\*电话号码#），取消呼叫转移（##21#）；

占线时呼叫转移：呼叫转移（\*\*67\*电话号码#），取消呼叫转移（##67#）；

无应答时呼叫转移：呼叫转移（\*\*61\*电话号码#），取消呼叫转移（##61#）；

无法接通时呼叫转移：呼叫转移（\*\*62\*电话号码#），取消呼叫转移（##62#）。

1. Android系统监听来电的两种方式：广播接收器和TelephoneManager对象。
2. Android系统支持3种电话状态：振铃、接听和空闲状态。
3. ListView控件显示数据库获取数据是，记录集必须有一个字段名为“\_id”。
4. Android支持的两种视频播放方式：VideoView和SurfaceView。
5. 2D绘图、动画技术和OpenGL
6. Android支持的3种基本的动画技术：帧动画、补间动画和属性动画。
7. 画图：实现view类中的onDraw方法。
8. 绘制位图的两种方法：Canvas.drawBitmap和Drawable.draw方法。
9. 控制帧动画循环播放：<animation-list>标签的android：oneshot=”false”或者用AnimationDrawable.setOneShot(false)。
10. Android支持的四种补间动画：移动、旋转、缩放、透明度补间动画。
11. OpenGL ES是专为嵌入和移动设备设计的一个2D/3D轻量图形库，基于OpenGL API设计的，是OpenGL三维图形API的子集。
12. OpenGL ES需要在onDrawFrame中完成绘图工作。
13. OpenGL ES支持两种光源：定向光和定位光。
14. Android系统资源
15. res目录和assets目录都可以存放资源文件，两者的区别：
16. 获取res中的资源要用R类中的资源ID，而获取assets中的资源可直接用文件名。
17. res目录只能建立指定的目录（不能建立其他目录或多级目录），assets可建立任意多级子目录。
18. Draw 9-patch工具，上边缘和左边缘表示水平和垂直可拉伸的范围，下边缘和右边缘表示内容显示的区域。
19. res/drawable中3中可控制图像状态：图像状态资源、图像级别资源和淡入淡出资源。
20. Android两种常用的XML图像资源：剪切资源和外形资源。
21. 其他
22. Android NDK是一套允许开发人员将本地代码嵌入Android应用程序的开发包。这些本地代码以动态链接库（.so文件）的形式存在。