一、前言

1、开发Android工具：

Eclipse+ADT插件、Android Studio

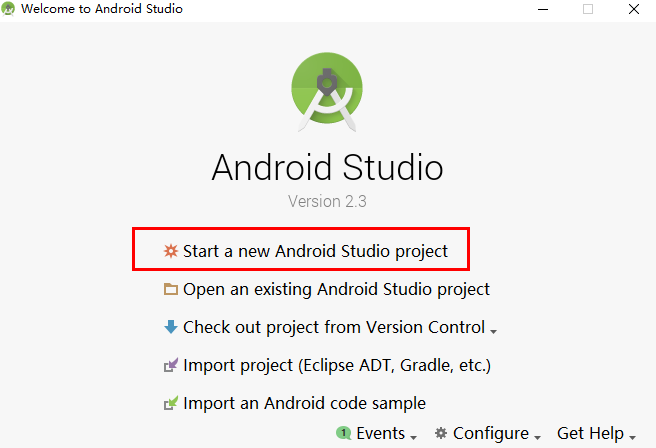
2、Android Studio工作环境：

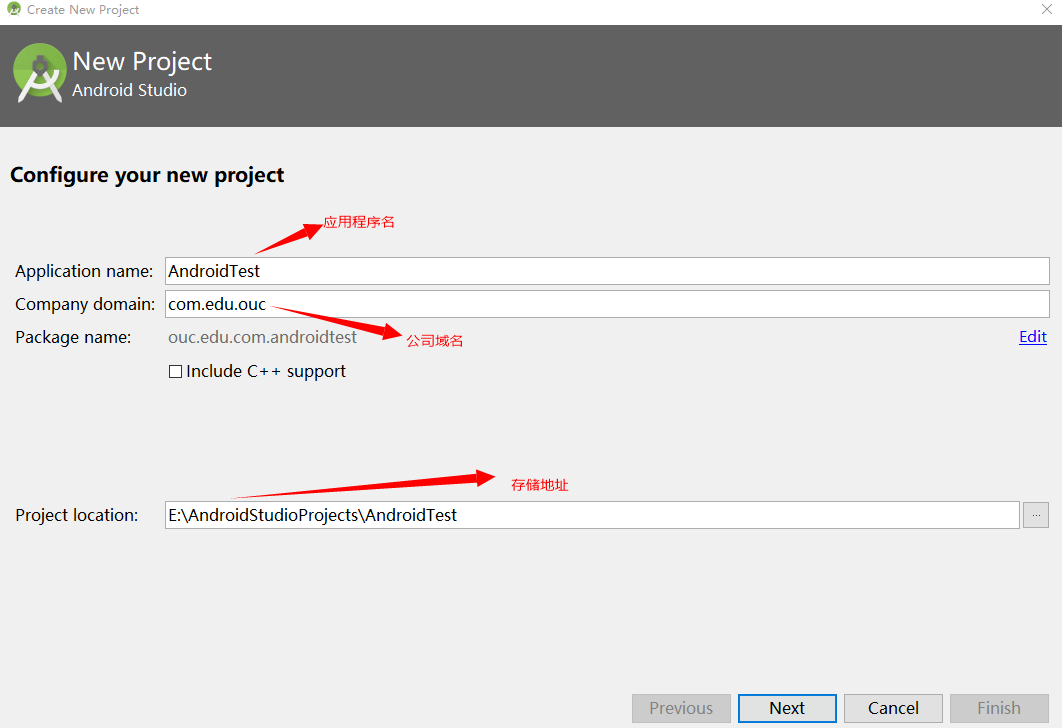
JDK

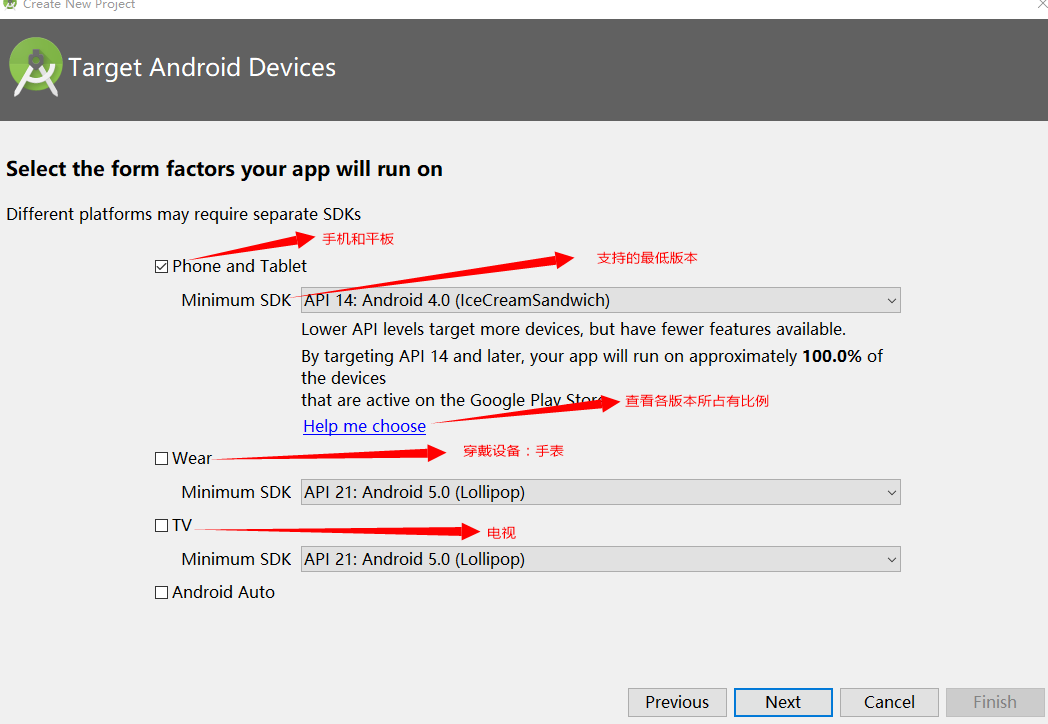
3、Android测试：

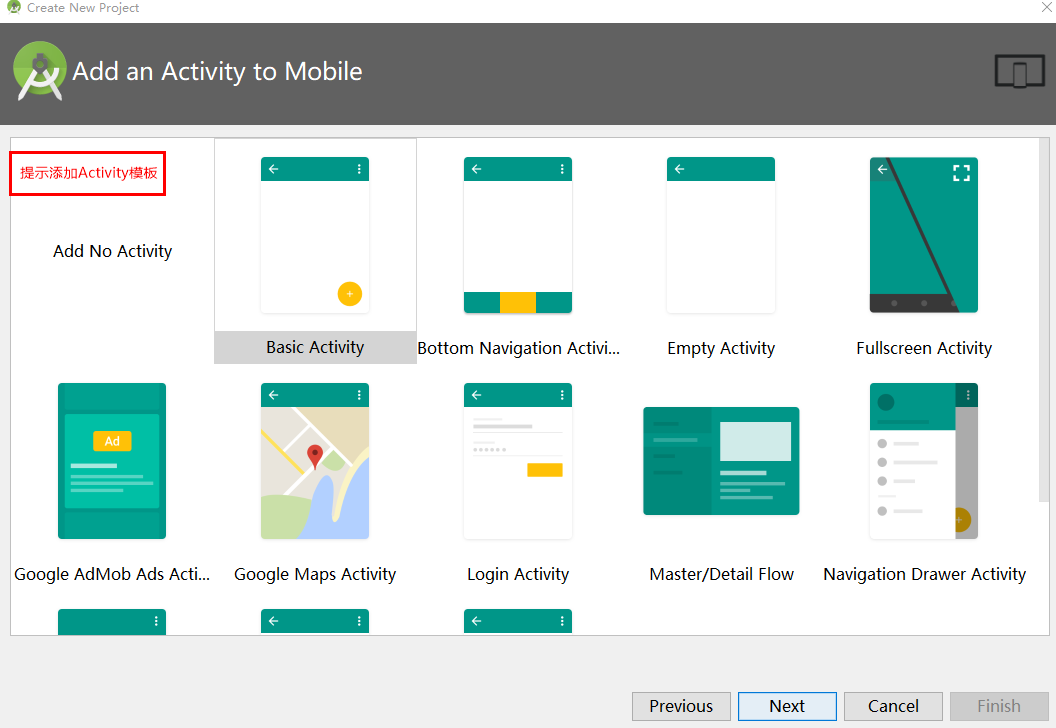
至少有一台安卓设备、Google Nexus模拟器

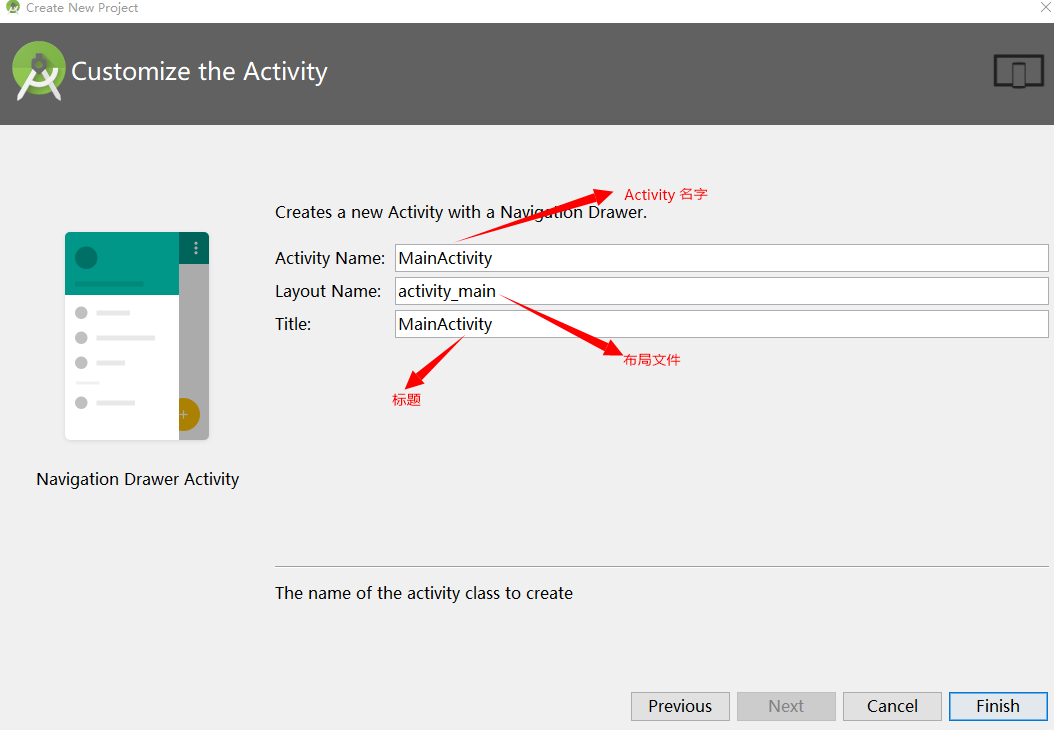
二、创建Android项目

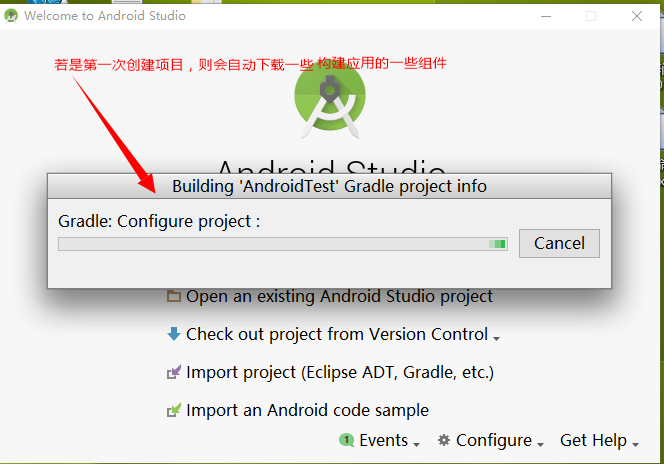


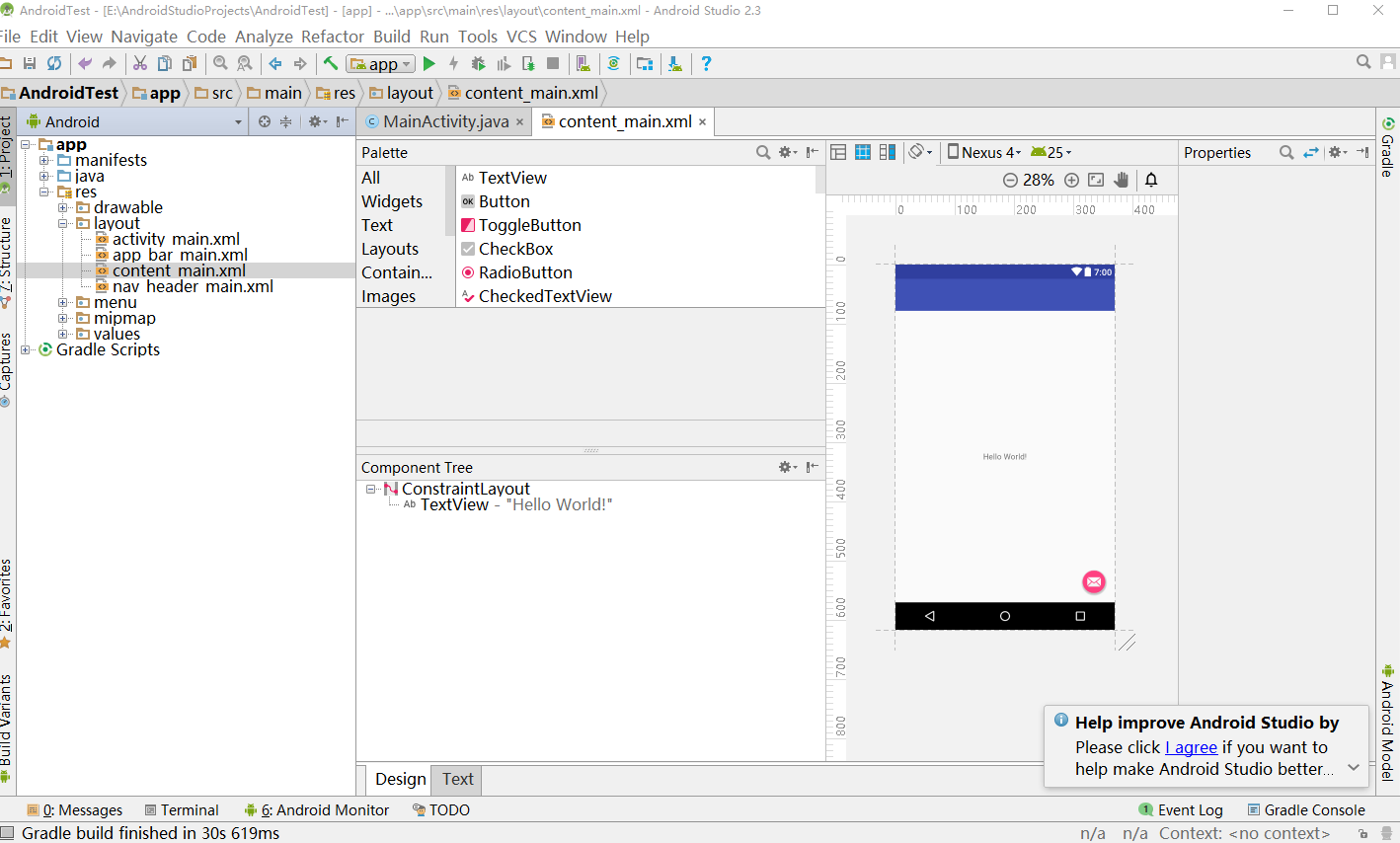




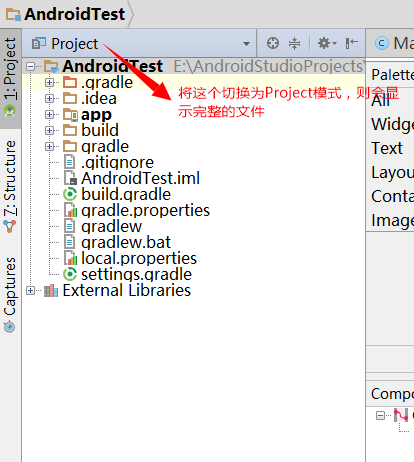


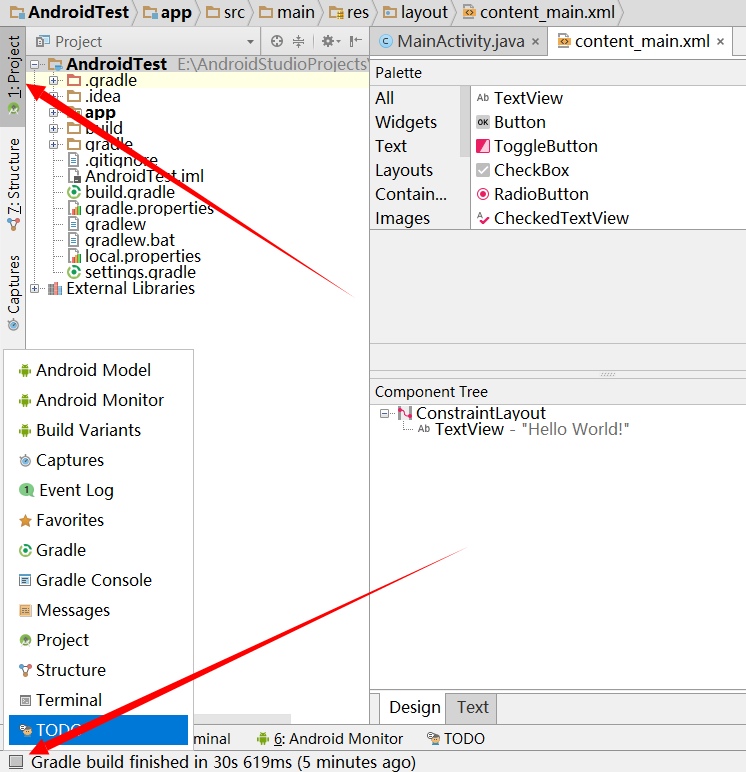






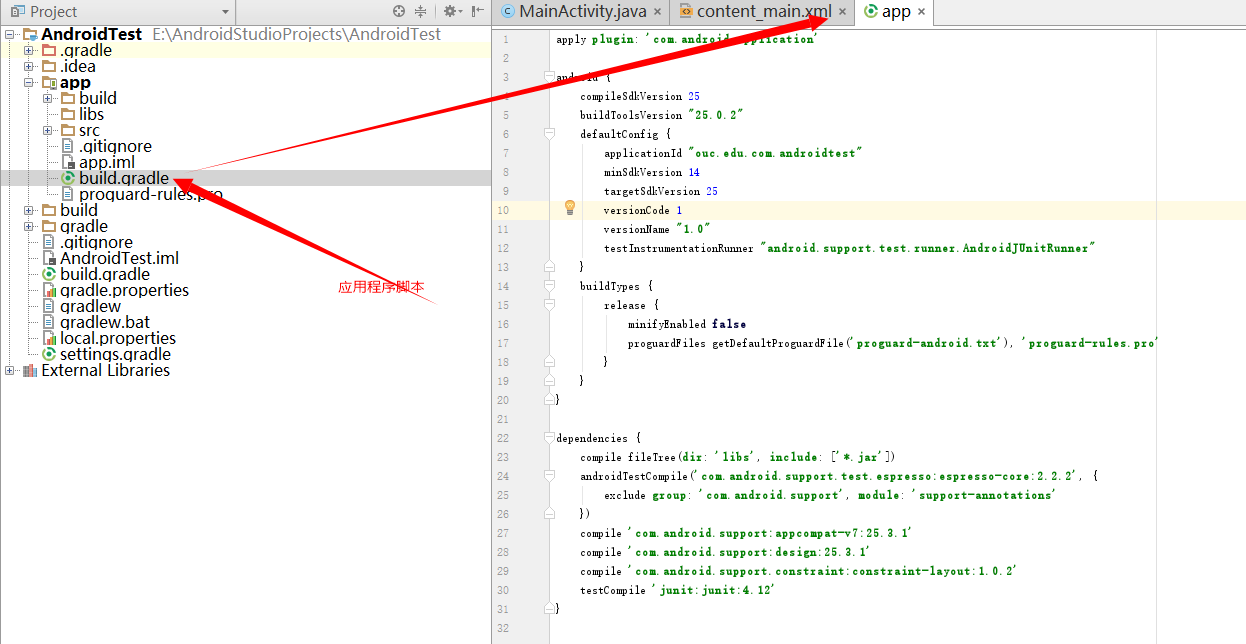
三、熟悉Android Studio运行环境

若在左侧没有Project窗口，可以点击左下角，选择Project



1、App下的Build.gradle:构建应用程序的脚本

通过这个脚本，告诉系统构建那个版本的应用程序



2、Android外观设置

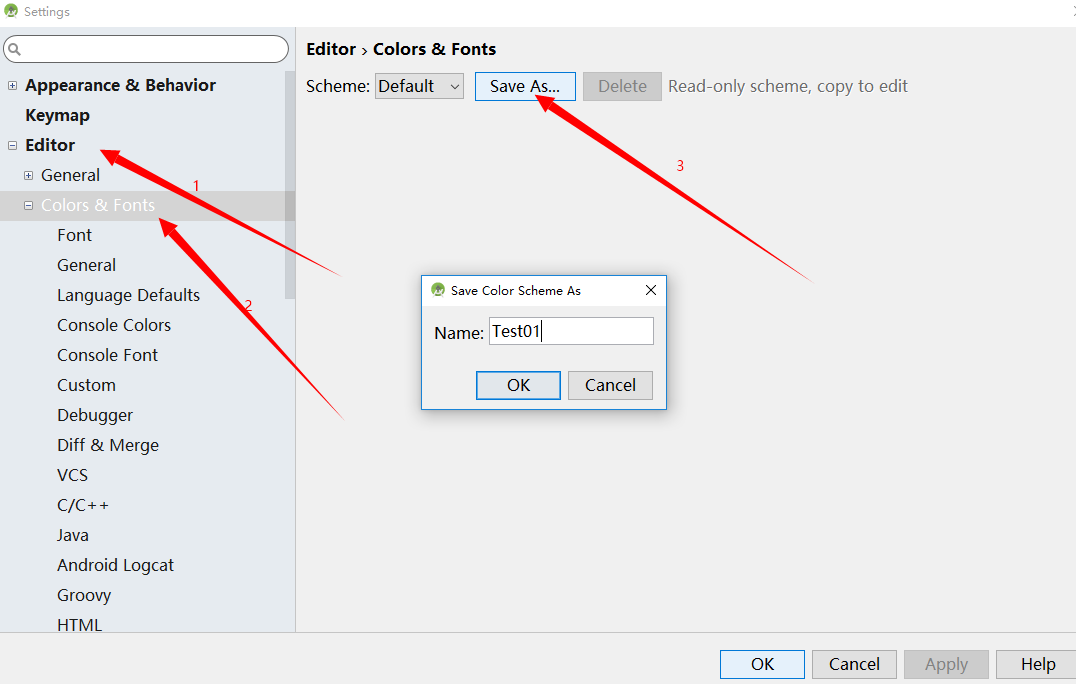
File->Setting

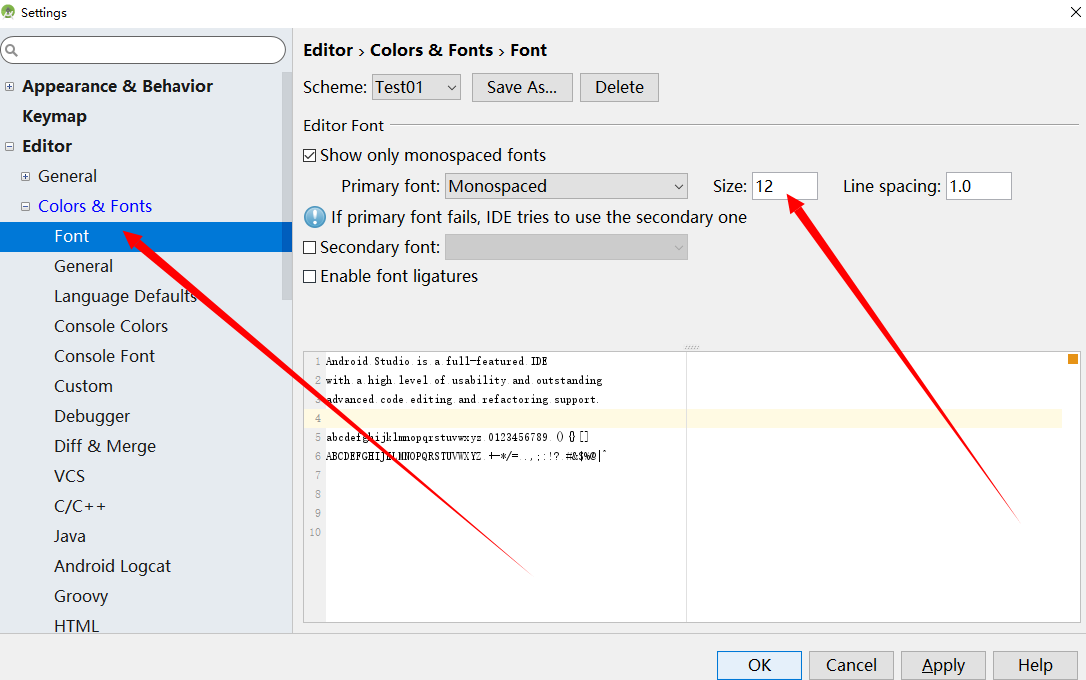
3、显示行号

File->Setting->Editor->General->Appearance->选中Show line numbers

4、自定义字体

File->Setting->Editor->Colors&Fonts->Save As输入标识名称，->ok->Font->输入字体大小即可





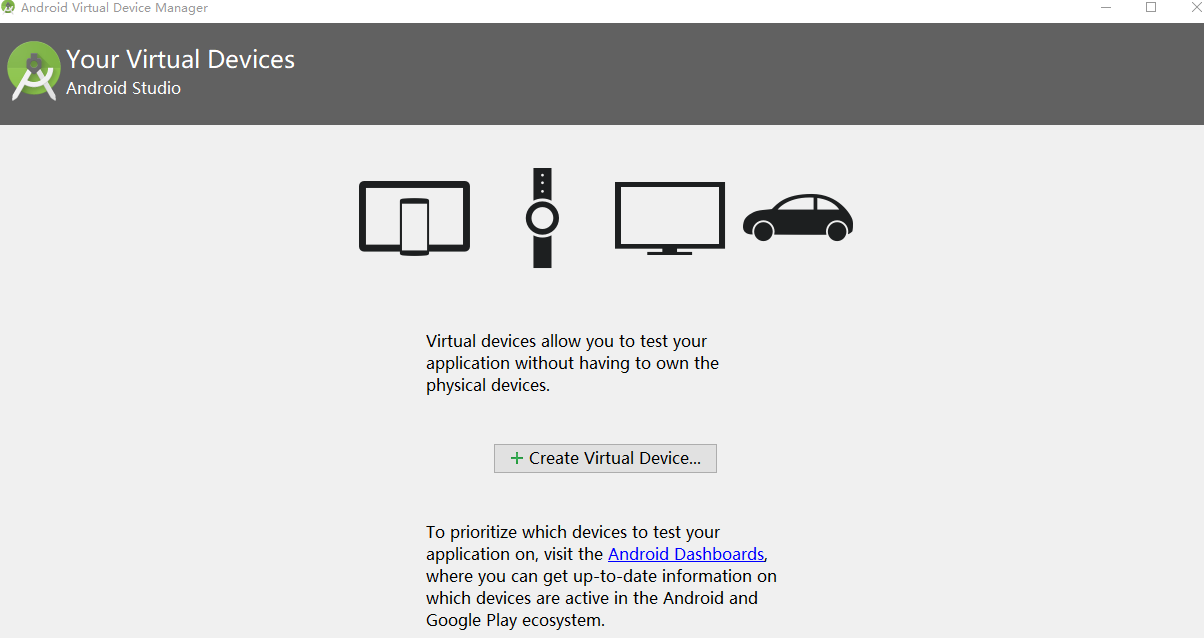
5、查看日志

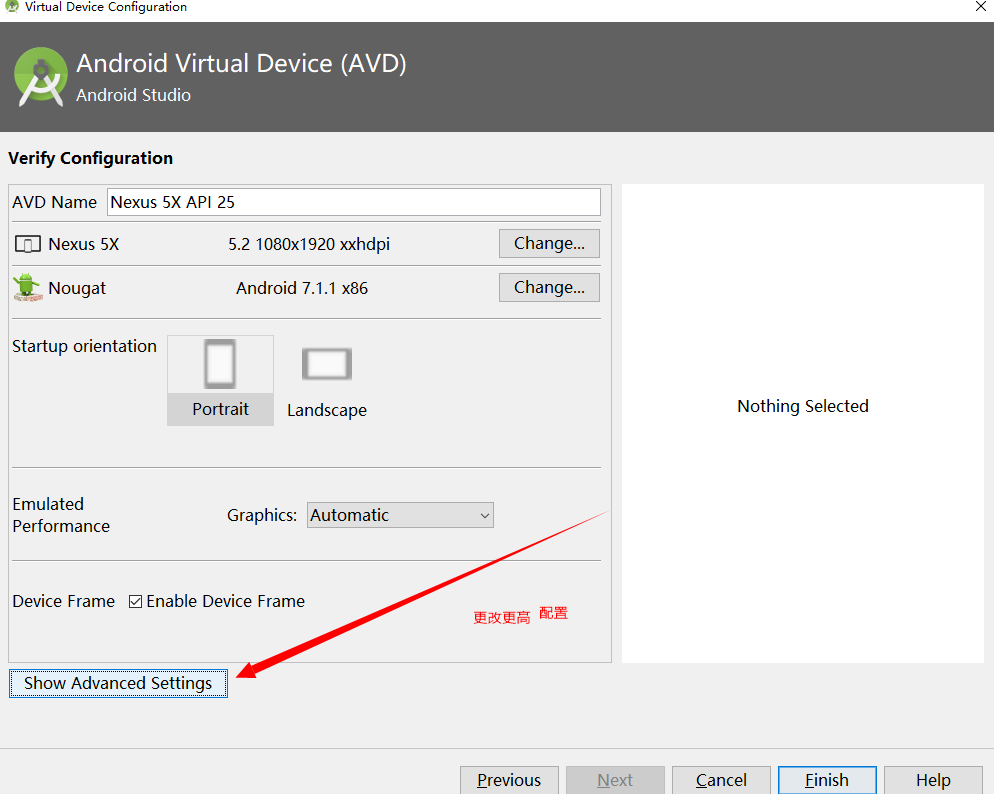
Android monitor->logcat

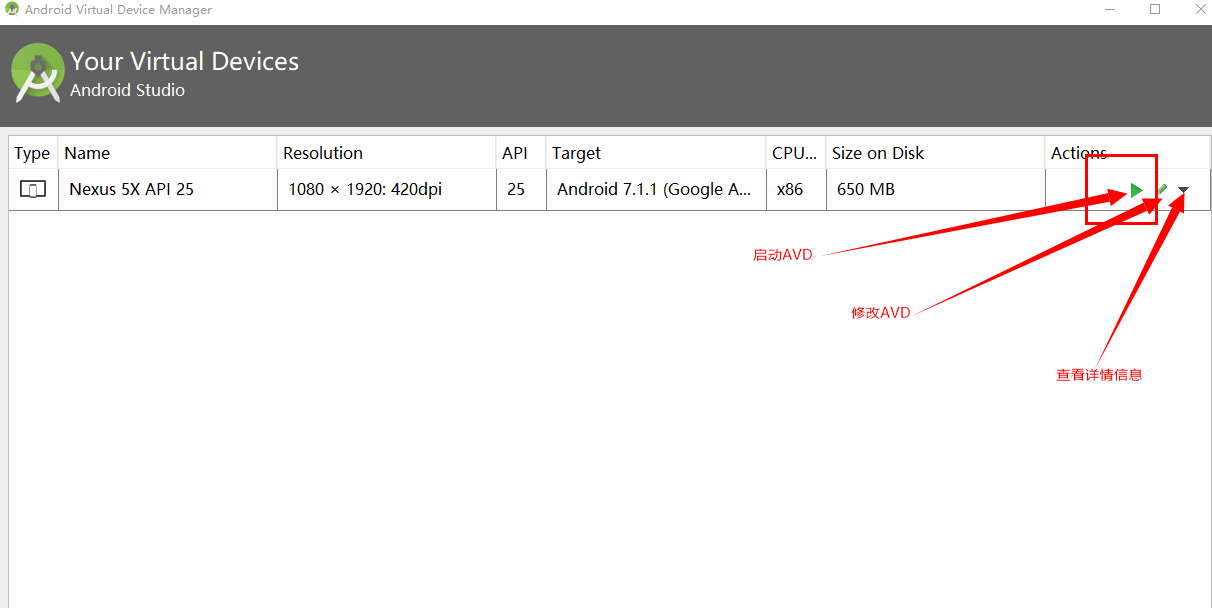
五、测试应用程序

1、android虚拟设备

Tools->Android->AVD Manager

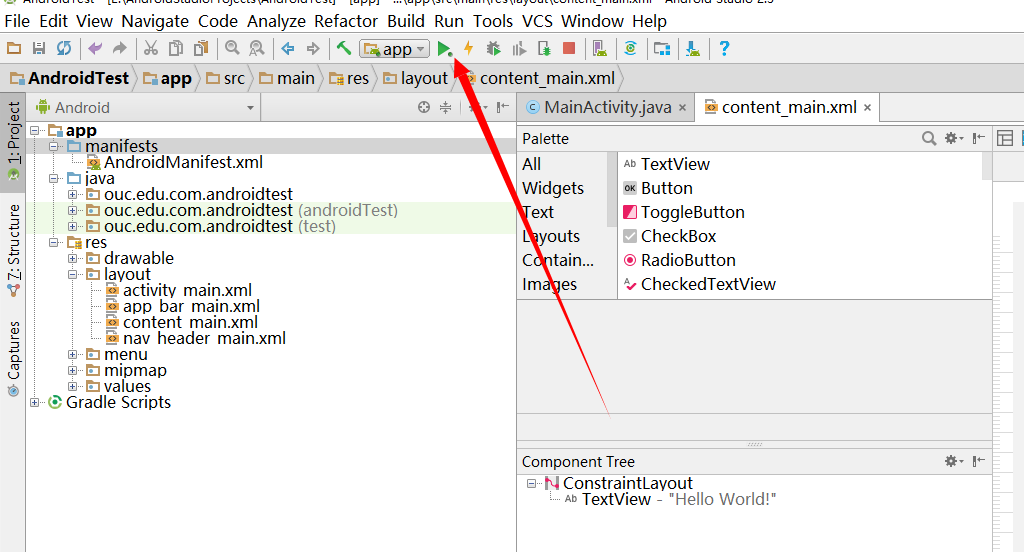




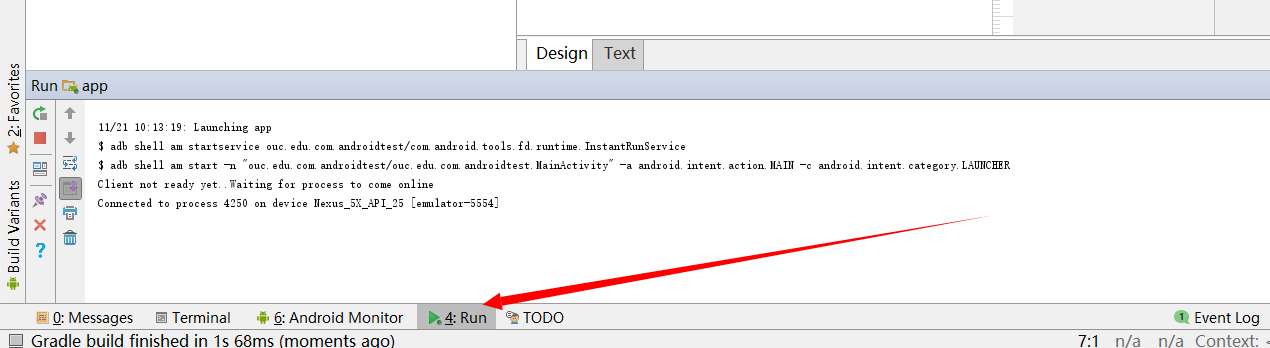


2、在虚拟设备下运行Android项目

（1）直接点击运行按钮



（2）查看日志信息

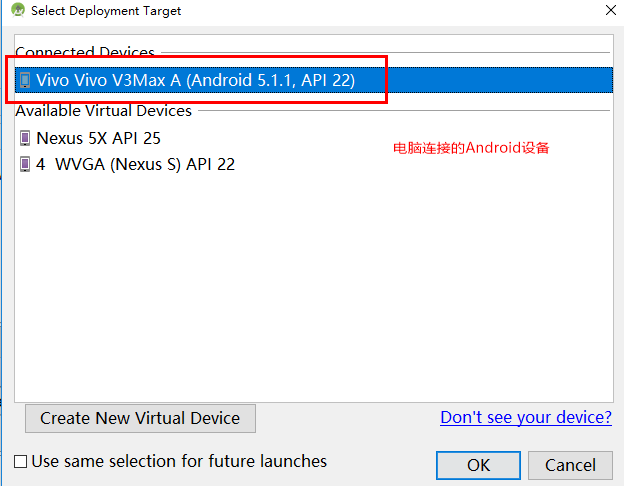


3、在真机上运行Android项目

（1）Tools->Android->SDK Manager,安装Google USB Driver

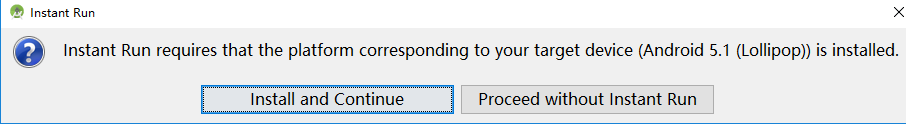
（2）将Android手机通过USB连接电脑，并打开手机的USB调试

（3）直接点击运行项目，我的手机是Vivo V3 MAXA Android5.1.1



出现一个提示框：即时运行需要安装与目标设备（Android 5.1（Lollipop））相对应的平台。

此时选择安装并继续或选择不需要安装也可以，首次安装会在手机弹出是否信任和安装界面，直接安装即可。



六、Android文件目录介绍

将模式切换到Project模式下

1、src->main->java——java源代码存放目录

2、src->main->res——存放资源文件（图片、xml文件等）

3、AndroidManifest.xml——功能清单文件。这个文件列出了应用程序所提供的功能

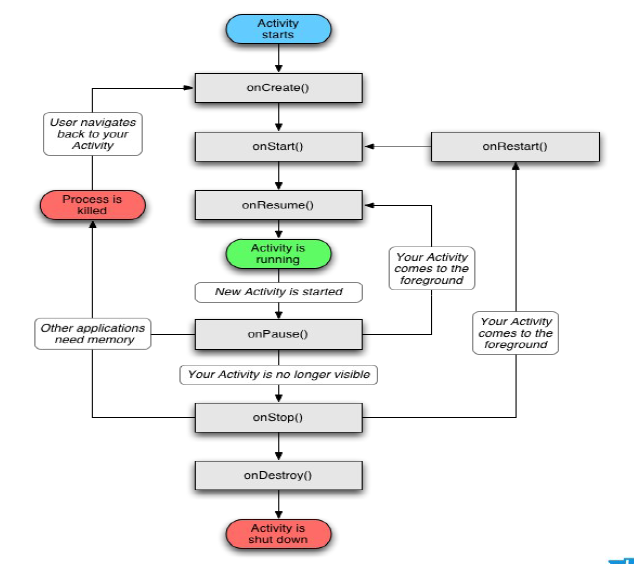
七、Activity

1、简介：

* Activity是Android四大基本组件之一
* 每个Activity对应一个软件界面
* 一个Android程序可以由一个或多个Activity组成
* 对于有界面的程序来说，程序的入口就是一个Activity

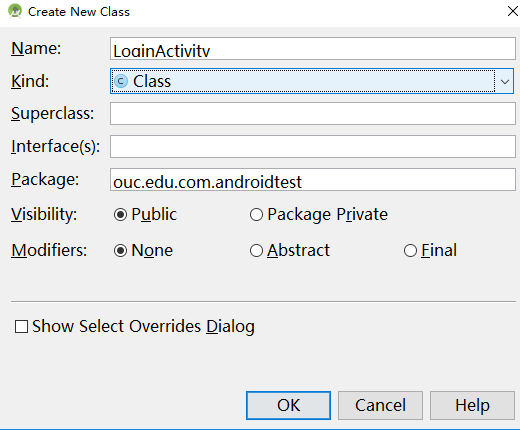
2、Activity生命周期

* 运行——当它在屏幕前台时（位于当前任务堆栈的顶部），它是激活或运行状态。它就是响应用户操作的Activity。方法：Oncreate,Onstart,Onresume
* 暂停——当它失去焦点但仍然对用户可见时，它处于暂停状态。方法：Onresume,Onpuse 若再次重新运行，会调用Onresume方法
* 停止——完全被另一个Activity覆盖时则处于停止状态。它仍然保留所有的状态和成员信息。然而对用户是不可见的，所以它的窗口将被隐藏，如果其它地方需要内存，则系统经常会杀死这个Activity。方法：Onpuse，OnStop,Ondestroy若再次重新运行，会调用Onrestart，Onstart，Onresume方法



3、创建一个Activity

* 在src->main->java文件下新建一个java class文件，并继承Activity，实现其Oncreate方法



* 每个Activity都要加载一个视图，即xml文件

在src->main->res->layout下新建一个布局文件为activity\_login.xml文件，并在LoginActivity中添加此视图setContentView(R.layout.***activity\_login***);

* 在AndroidManifest.xml中注册此Activity（必须在Application标签内注册才有效）

在src->main下的AndroidManifest.xml的Application标签内添加中

<activity android:name=".LoginActivity"></activity>

注：

<intent-filter> <!—标明应用程序的入口程序，此标签在那个Activity下，应用程序就会先启动哪个Activity-->  
 <action android:name="android.intent.action.MAIN" />  
 <category android:name="android.intent.category.LAUNCHER" />  
</intent-filter>

八、添加控件方法

1、通过代码添加

在src->main->res->layout下的activity\_main.xml的Test模式下添加

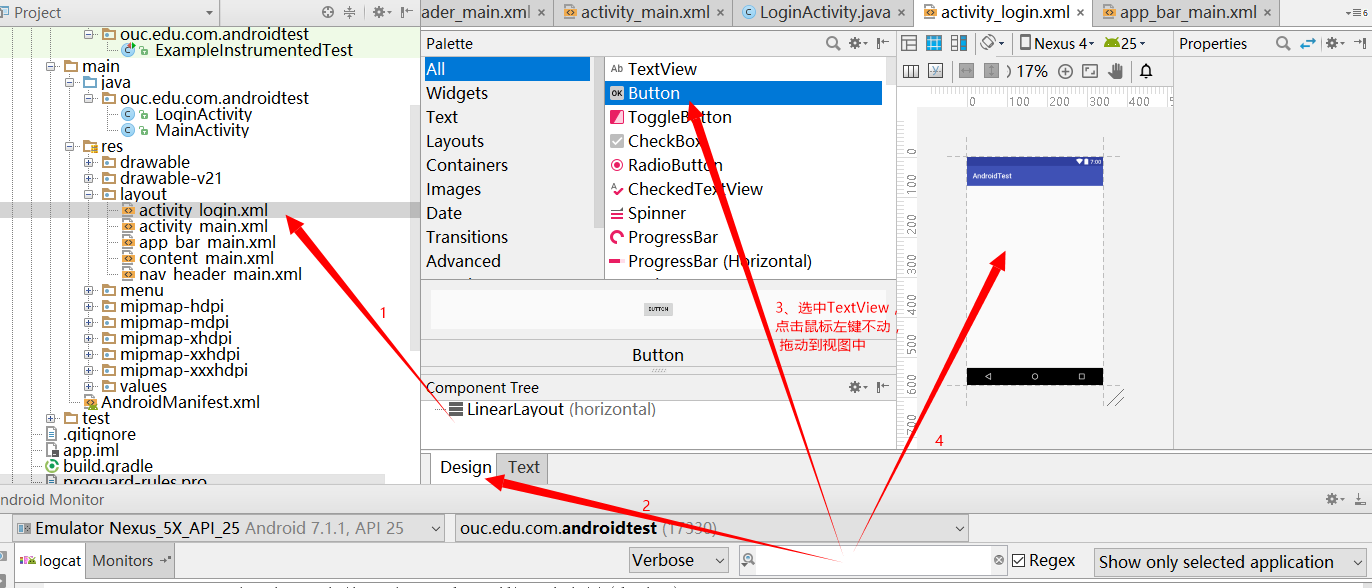
<TextView android:id="@+id/ tv\_test"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="TextView"/>

注：**wrap\_content**：自适应

fill\_content:充满父控件

2、通过拖动控件添加

在src->main->res->layout下的activity\_main.xml的Design模式下，将要添加的控件拖动到视图上



九、添加Button的Click事件

1、为视图添加一个button按钮

2、在Activity中添加代码：

//声明一个Button对象  
private Button button;  
@Override  
protected void onCreate(@Nullable Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 //这里可以使用setContentView(R.layout.xxx) 显示某个视图....  
 setContentView(R.layout.activity\_login);  
 //将通过id找到button实例并强制转换成Button类型  
 button=(Button)findViewById(R.id.btn\_test);  
 //添加button的点击事件  
 button.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 }  
 });  
}

注：若有多个点击事件，可将其进行归类

在Activity中实现OnClickListener接口

Button1.setOnClickListener(this);

Button2.setOnClickListener(this);

Public void onClick(View v){

Switch(v.getId()){

Case R.id.start:

Break;

Default:

Break;

}

}

九、控件的使用

1、RatingBar是很重要的评星控件，在很多软件中都有这评星的功能。在Android中系统也提供了一个专门评星的控件： RatingBar

• 相关属性：

– Android:numStars：指定评星数量

– Android:rating ：指定当前分数

– Android:stepSize：指定分数的增量单位

* 获取星星分数

实现 OnRatingBarChangeListener接口，并重写其星星改变函数

@Override  
public void onRatingChanged(RatingBar ratingBar, float rating, boolean fromUser) {  
 //得到当前星星的数量  
 ratingBar.setRating(rating);  
 //如果当前的星星iD和用户操作的ID是同一个  
 if(ratingBar.getId()==R.id.ratingBar2){  
 tv.setText("分数："+rating);  
 }  
}

2、Gallery一般用于显示图像列表，也可以称为相册控件。这里重点，他需要一个适配器，那么我们需要提供自己的适配器（adapter）。

九、Intent

1、概念：

Intent:意图，负责对应用中一次操作的动作、动作涉及数据、附加数据进行描述 。所以Intent在这里主要是一个媒介的作用。

2、作用

* Intent可以显示Activity，可以发送广播和启动服务。
* 在同一个应用程序中往往会使用Intent对象来指定一个Activity，并通过startActivity或startActivityForResult方法启动一个Activity。
* 通过Intent在不同的Activity中进行携带数据

3、通过Intent实现Activity之间的跳转

* 创建两个Activity（包含两个layout文件）

4、通过Intent传递数据

Intent传递数据方法：putExtra

* 在第一个Activity中的添加传递数据方法

Intent intent=new Intent(LoginActivity.this,MainActivity.class);  
intent.putExtra("name","通过Intent传递过来的数据");  
startActivity(intent);

* 在第二个Activity中添加获取数据方法

String val=getIntent().getStringExtra("name");

十、Service

1、概念

* 是一个应用程序组件，没有图形界面，通常用来处理一些耗时比较长的操作（闹钟）。

2、生命周期

OnCreat（）

OnStartCommand（）

OnDestroy（）

3、创建一个Service

* 创建一个Service类，并实现Serice
* 在一个Activity中启动Service

Intent i=new Intent(LoginActivity.this,MyService.class);  
startService(i);

* 在Manifest.xml中注册Service(application标签内)

<service android:name=”MyService”></service>

十一、BroadcastRecevicer

1、简介

2、创建一个广播接收器

* 发送广播消息

Intent intent =new Intent(LoginActivity.this,MyReceive.class);

intent.putExtra("name","通过Intent传递过来的数据");  
sendBroadcast(intent);

* 接收广播消息

创建一个Broadcast类，并实现BroadcastRecevicer，重写Onreceive方法

Intent.getStrubgExtra(“ name”);

* 在Manifest.xml中注册BroadcastRecevicer (application标签内)

<receiver android:name=”MyReceiver”></ receiver >

十二、Handler

1、简介

2、创建一个Handler

* 通过handler的post方法启动runnable

handler.post(runnable);

* 移除runnable

handler.removeCallbacks(runnable)

* 实例化Handler

Handler handler=new Handler();

Runnable runnable=new Runnable(){

Public void run(){

Systemt.out.println(“runnable”);

//handler是一个消息队列，每执行一次就会传递一个runnable

handler.postDelayed(runnable,3000);//每隔3000毫秒执行一次

}

};

十三、UI组件与布局

1、线性布局：线性布局又分为水平布局和垂直布局，可通过orientation属性进行设置。标签中有一个非常重要的属性：Gravity属性。

* android:layout\_weight**="1"**通过设置控件的layout\_weight属性以控制各个控件在布局中的**相对大小**,线性布局会根据该控件layout\_weight值与其所处布局中所有控件layout\_weight值之和的比值为该控件分配占用的区域。
* android:gravity　属性是对该view中内容的限定．比如一个button 上面的text. 你可以设置该text 相对于view的靠左，靠右等位置．
* 这两个属性可选的值有：top、bottom、left、right、center\_vertical、fill\_vertical、center\_horizontal、fill\_horizontal、center、fill、clip\_vertical。一个属性可以包含多个值，需用“|”分开。其含义如下：

|  |  |
| --- | --- |
| top | 将对象放在其容器的顶部，不改变其大小. |
| bottom | 将对象放在其容器的底部，不改变其大小. |
| left | 将对象放在其容器的左侧，不改变其大小. |
| right | 将对象放在其容器的右侧，不改变其大小. |
| center\_vertical | 将对象纵向居中，不改变其大小.  垂直对齐方式：垂直方向上居中对齐。 |
| fill\_vertical | 必要的时候增加对象的纵向大小，以完全充满其容器.  垂直方向填充 |
| center\_horizontal | 将对象横向居中，不改变其大小.  水平对齐方式：水平方向上居中对齐 |
| fill\_horizontal | 必要的时候增加对象的横向大小，以完全充满其容器.  水平方向填充 |
| center | 将对象横纵居中，不改变其大小. |
| fill | 必要的时候增加对象的横纵向大小，以完全充满其容器. |
| clip\_vertical | 附加选项，用于按照容器的边来剪切对象的顶部和/或底部的内容. 剪切基于其纵向对齐设置：顶部对齐时，剪切底部；底部对齐时剪切顶部；除此之外剪切顶部和底部.  垂直方向裁剪 |
| clip\_horizontal | 附加选项，用于按照容器的边来剪切对象的左侧和/或右侧的内容. 剪切基于其横向对齐设置：左侧对齐时，剪切右侧；右侧对齐时剪切左侧；除此之外剪切左侧和右侧.  水平方向裁剪 |

注：

* center\_vertical和center\_horizontal两个属性值，center\_vertical是指将对象在垂直方向上居中对齐，即在从上到下的方向上选择中间的位置放好；center\_horizontal是指将对象水平方向上居中对齐，即在从左到右的方向上选择中间的位置放好。
* 当我们采用LinearLayout布局时，有以下特殊情况需要我们注意：  
  (1)当 android:orientation="vertical"  时， android:layout\_gravity只有水平方向的设置才起作用，垂直方向的设置不起作用。即：left，right，center\_horizontal 是生效的。

(2)当 android:orientation="horizontal" 时， android:layout\_gravity只有垂直方向的设置才起作用，水平方向的设置不起作用。即：top，bottom，center\_vertical 是生效的。

2、相对布局

* 相对布局可以设置某一个视图相对于其他视图的位置，这些位置包括上、下、左、右。以及可以设置其相对于父级控件的位置。
* 相对布局属性

android:layout\_marginTop="25dip" //顶部距离

android:gravity="left" //空间布局位置

android:layout\_marginLeft="15dip //距离左边距

// 相对于给定ID控件

android:layout\_above 将该控件的底部置于给定ID的控件之上;

android:layout\_below 将该控件的底部置于给定ID的控件之下;

android:layout\_toLeftOf 将该控件的右边缘与给定ID的控件左边缘对齐;

android:layout\_toRightOf 将该控件的左边缘与给定ID的控件右边缘对齐;

android:layout\_alignBaseline 将该控件的baseline与给定ID的baseline对齐;

android:layout\_alignTop 将该控件的顶部边缘与给定ID的顶部边缘对齐;

android:layout\_alignBottom 将该控件的底部边缘与给定ID的底部边缘对齐;

android:layout\_alignLeft 将该控件的左边缘与给定ID的左边缘对齐;

android:layout\_alignRight 将该控件的右边缘与给定ID的右边缘对齐;

// 相对于父组件

android:layout\_alignParentTop 如果为true,将该控件的顶部与其父控件的顶部对齐;

android:layout\_alignParentBottom 如果为true,将该控件的底部与其父控件的底部对齐;

android:layout\_alignParentLeft 如果为true,将该控件的左部与其父控件的左部对齐;

android:layout\_alignParentRight 如果为true,将该控件的右部与其父控件的右部对齐;

// 居中

android:layout\_centerHorizontal 如果为true,将该控件的置于水平居中;

android:layout\_centerVertical 如果为true,将该控件的置于垂直居中;

android:layout\_centerInParent 如果为true,将该控件的置于父控件的中央;

// 指定移动像素

android:layout\_marginTop 上偏移的值;

android:layout\_marginBottom 下偏移的值;

android:layout\_marginLeft 　　左偏移的值;

android:layout\_marginRight 　 右偏移的值;

十四、SQLite数据库存储

1、特点

* 轻量级

SQLite和C\S模式的数据库软件不同，它是进程内的数据库引擎，因此不存在数据库的客户端和服务器。使用SQLite一般只需要带上它的一个动态库，就可以享受它的全部功能。而且那个动态库的尺寸也相当小

* 独立性

SQLite数据库的核心引擎本身不依赖第三方软件，使用它也不需要“安装”，所以在使用的时候能够省去不少麻烦

* 隔离性

SQLite数据库中的所有信息（比如表、视图、触发器）都包含在一个文件内，方便管理和维护

* 跨平台

SQLite数据库支持大部分操作系统，除了我们在电脑上使用的操作系统之外，很多手机操作系统同样可以运行，比如Android、WindowsMobile、Symbian、Palm等

2、简介

* 多语言接口

SQLite数据库支持很多语言编程接口，比如C\C++、Java、Python、dotNet、Ruby、Perl等，得到更多开发者的喜爱。

* 安全性

SQLite数据库通过数据库级上的独占性和共享锁来实现独立事务处理。这意味着多个进程可以在同一时间从同一数据库读取数据，但只有一个可以写入数据。在某个进程或线程向数据库执行写操作之前，必须获得独占锁定。在发出独占锁定后，其他的读或写操作将不会再发生

3、使用SqlLite

十五、SharedPreference储存

1、简介

是Android提供的一种轻量级的数据存储方式，主要用来存储一些简单的配置信息，例如，默认欢迎语，登录用户名和密码等。其以键值对的方式存储,使得我们能很方便进行读取和存入