一、前言

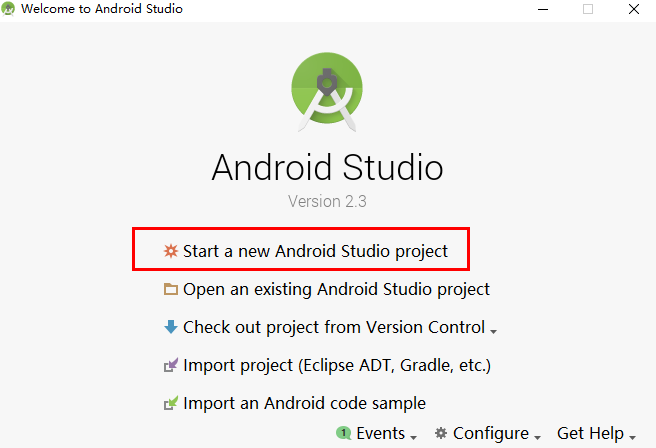
1、开发Android工具：Eclipse+ADT插件、Android Studio

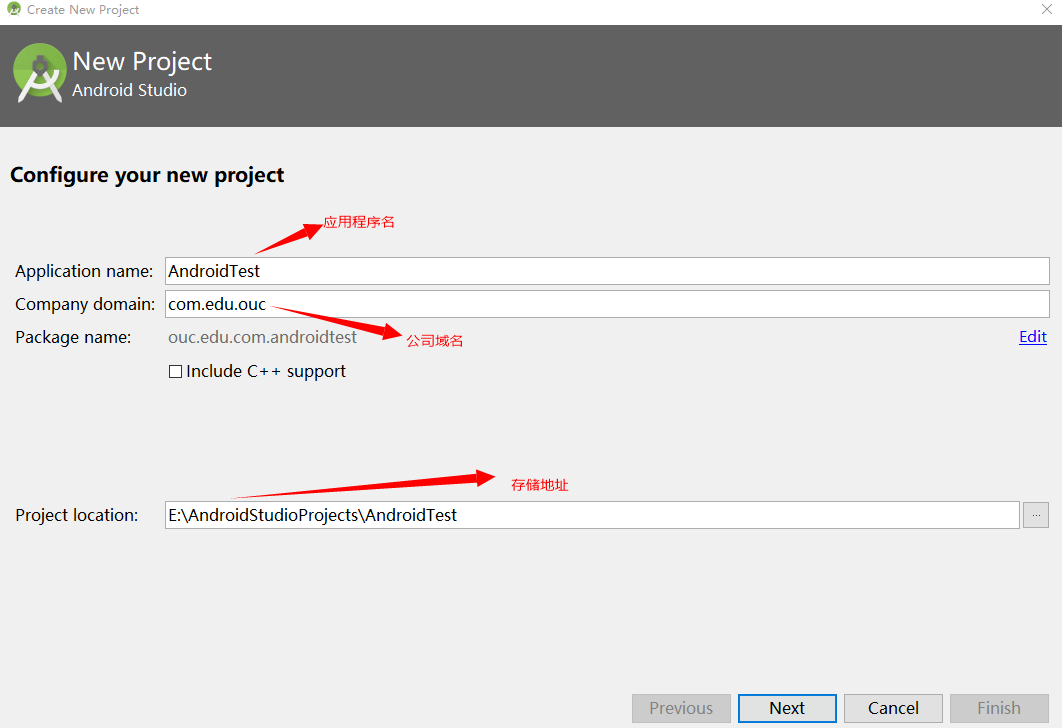
2、Android Studio工作环境：JDK

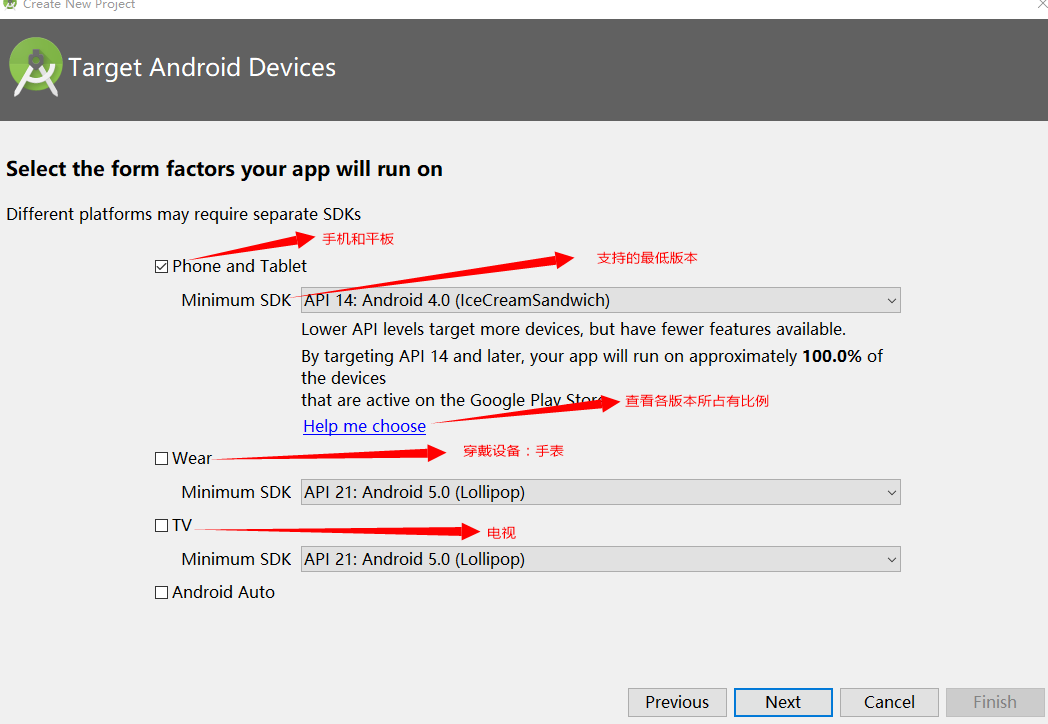
3、Android测试：至少有一台安卓设备、Google Nexus模拟器

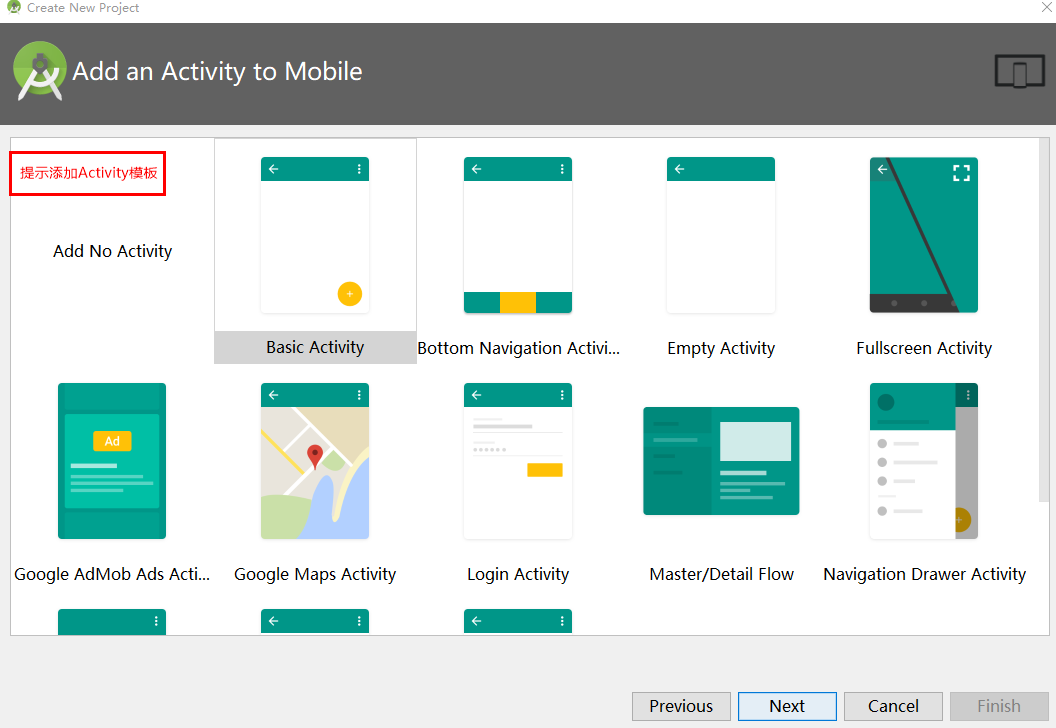
二、下载安装JDK，配置其环境变量，下载安装Android Studio，更新SDK

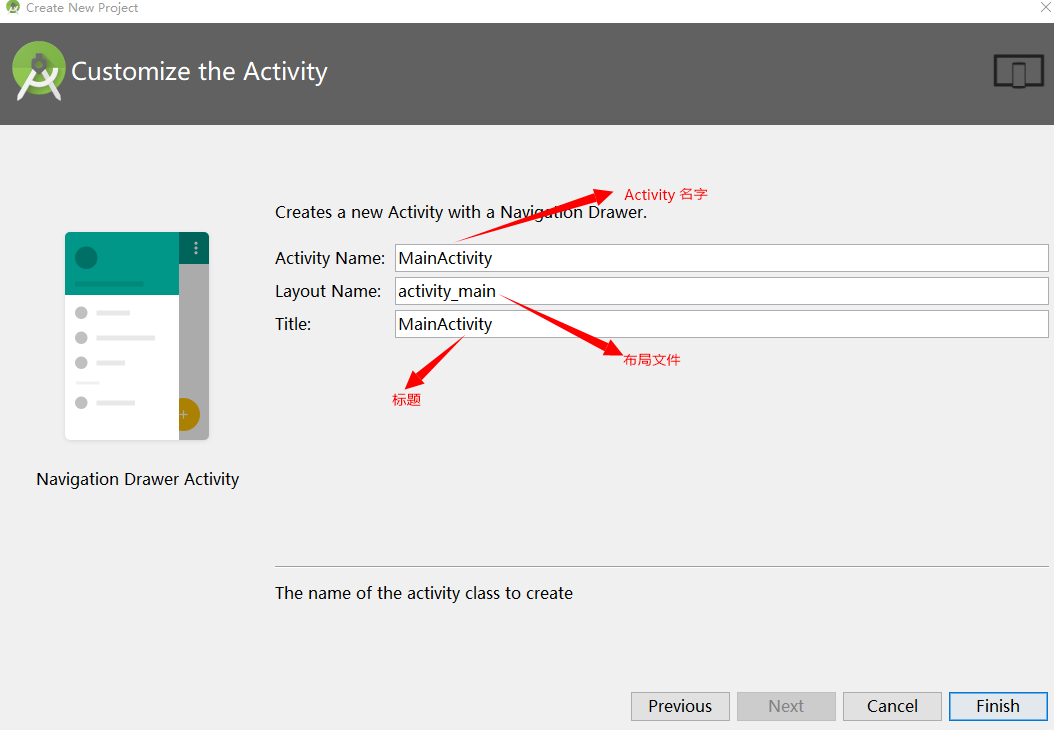
三、创建Android项目

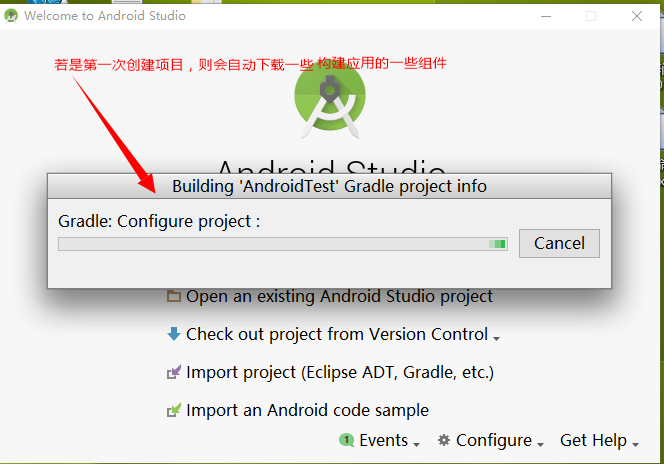


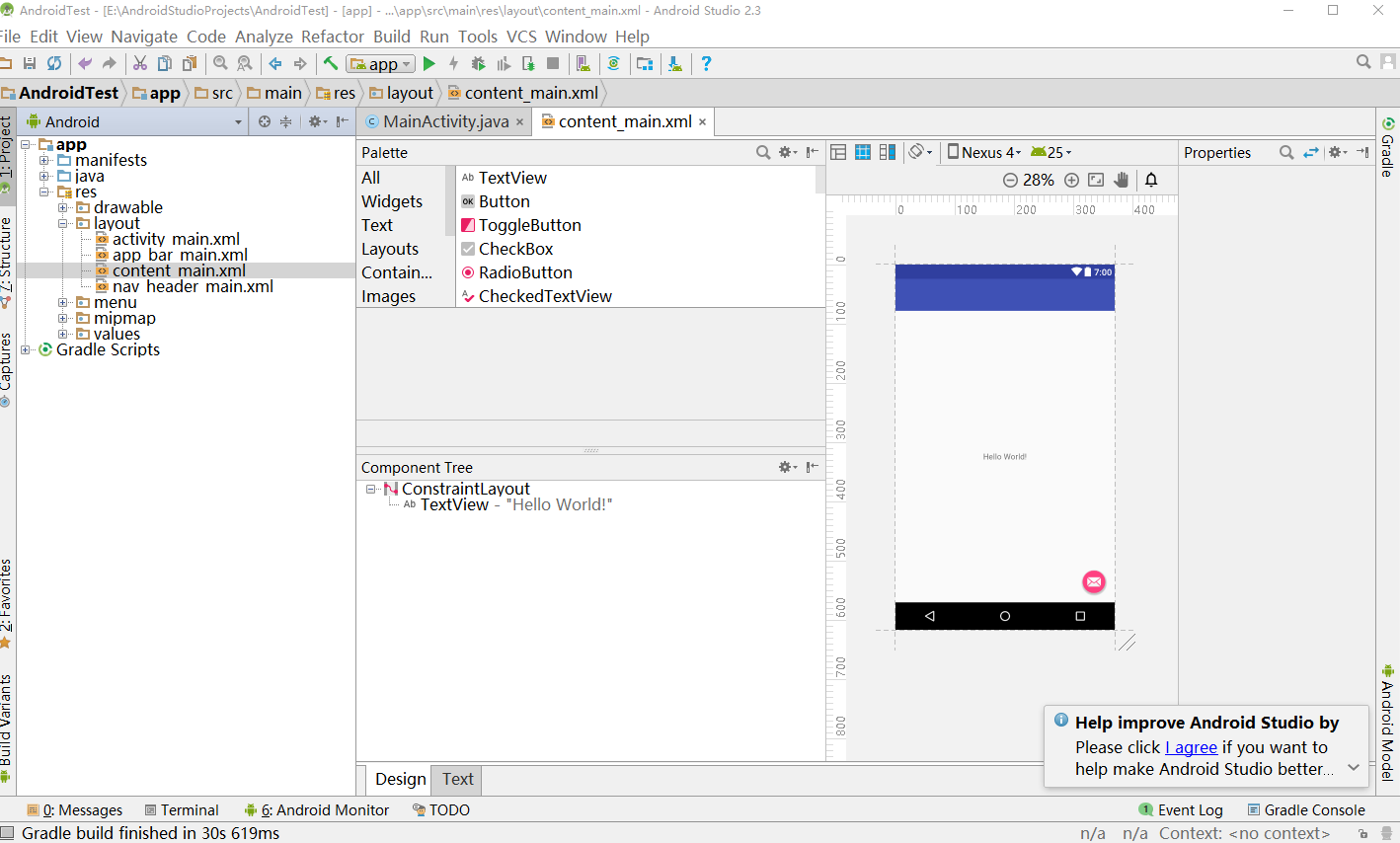




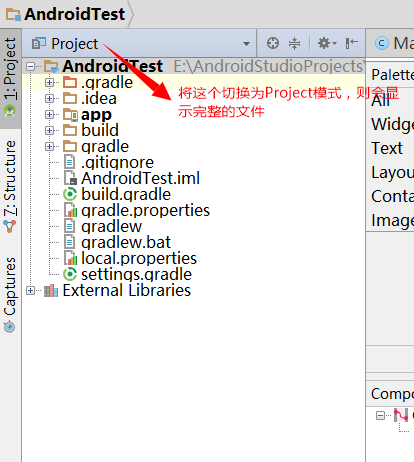


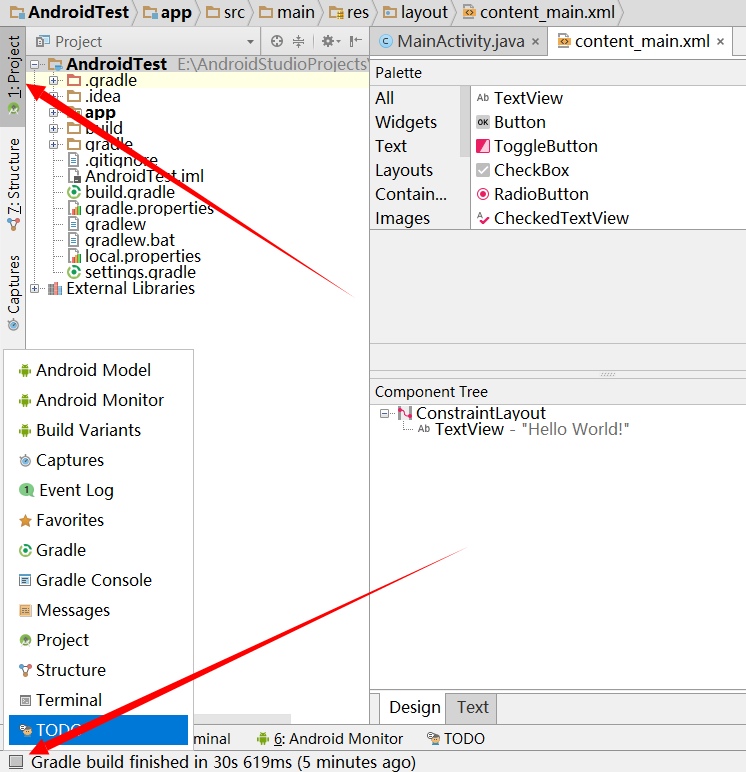




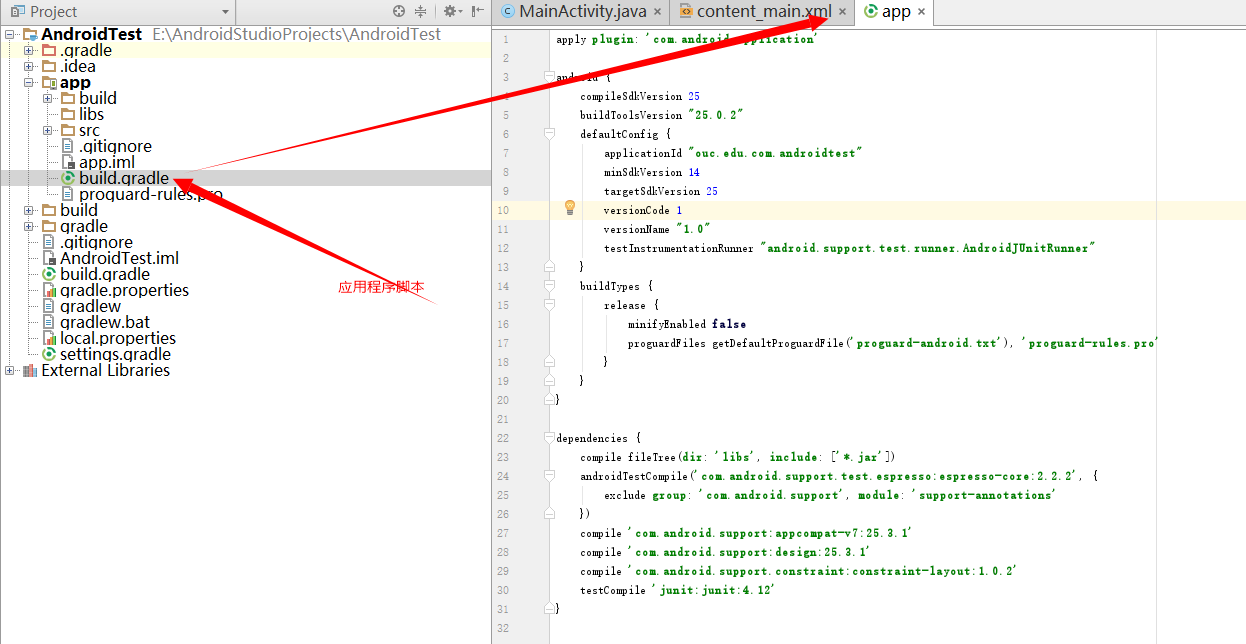


四、熟悉Android Studio运行环境

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

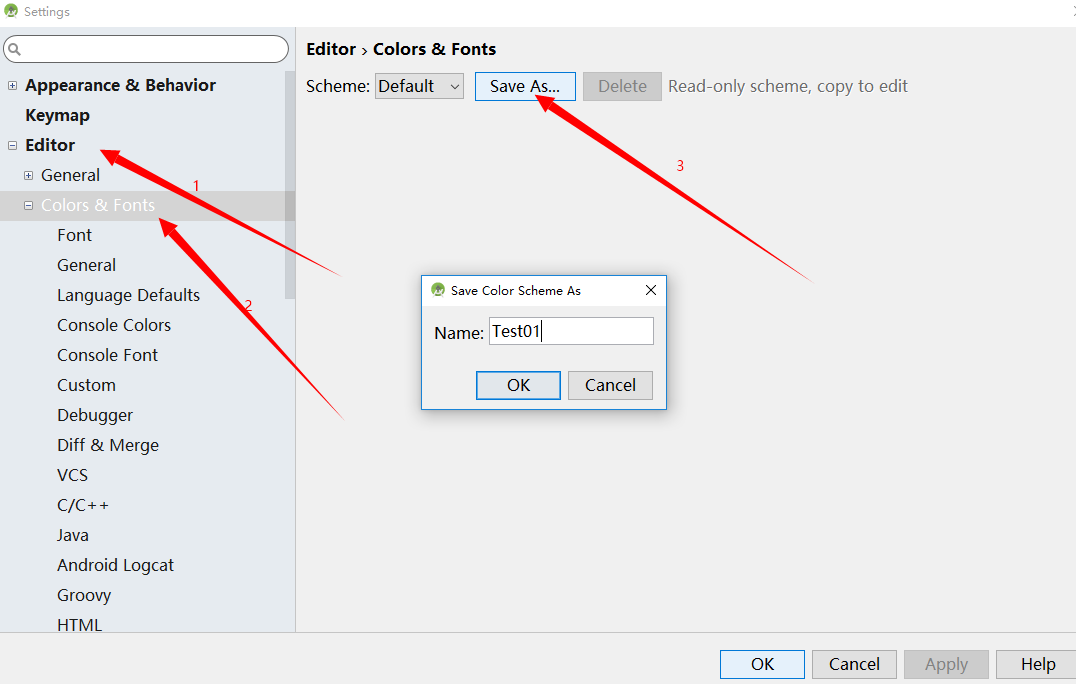
File->Setting

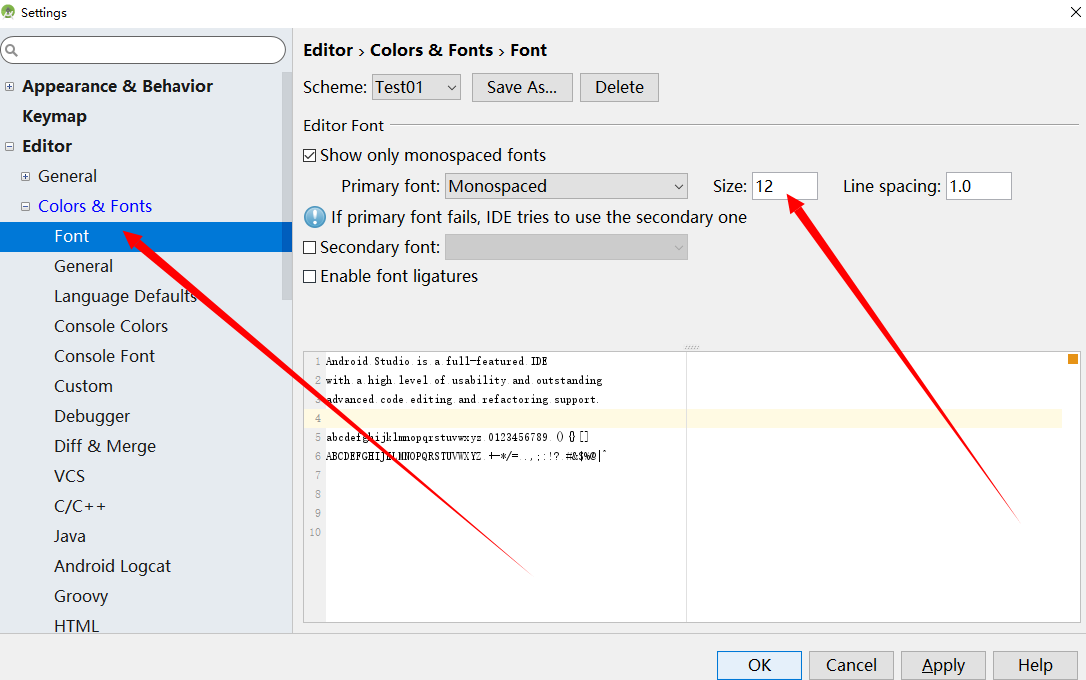
3、显示行号

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



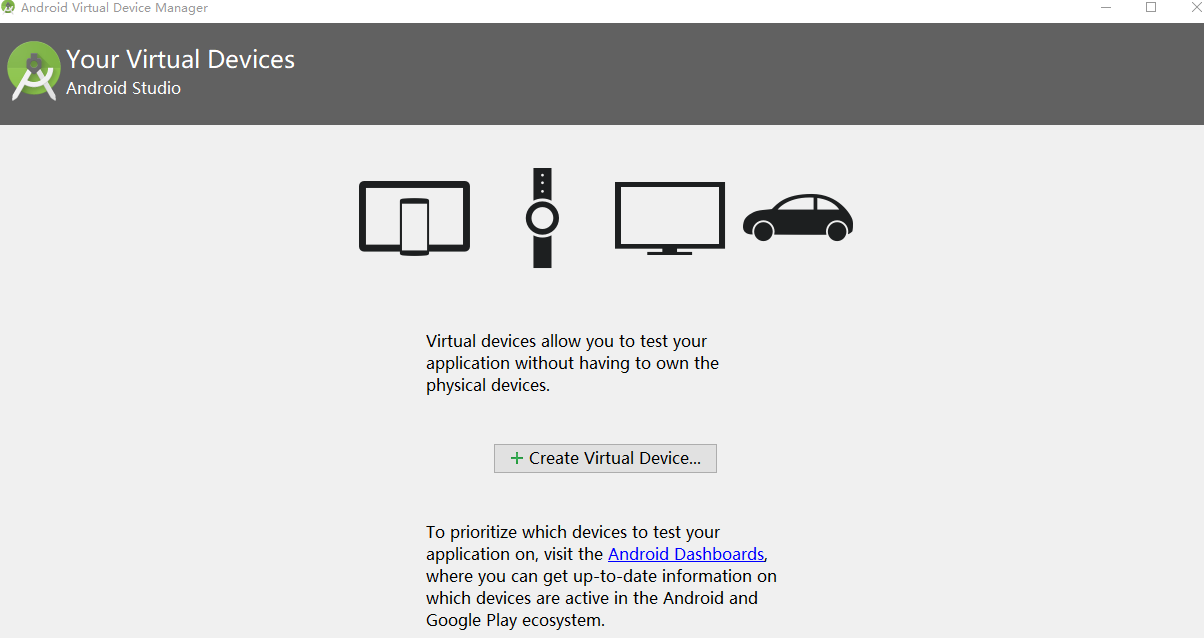


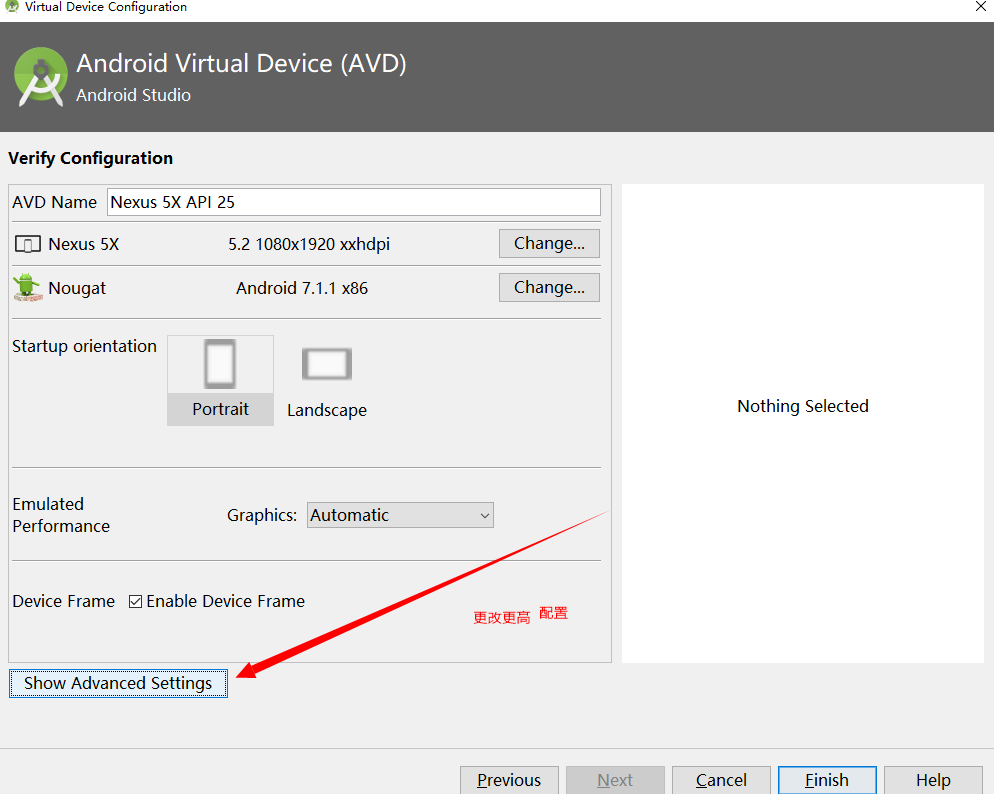
6、查看日志

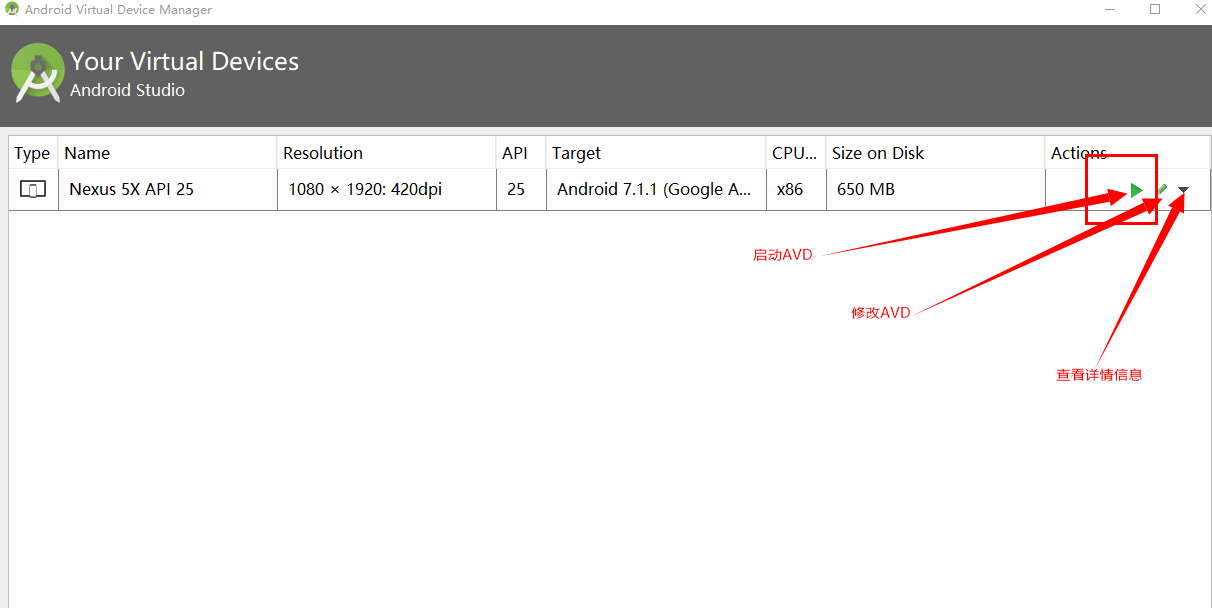
Android monitor->logcat

五、测试应用程序

1、android虚拟设备：Tools->Android->AVD Manager

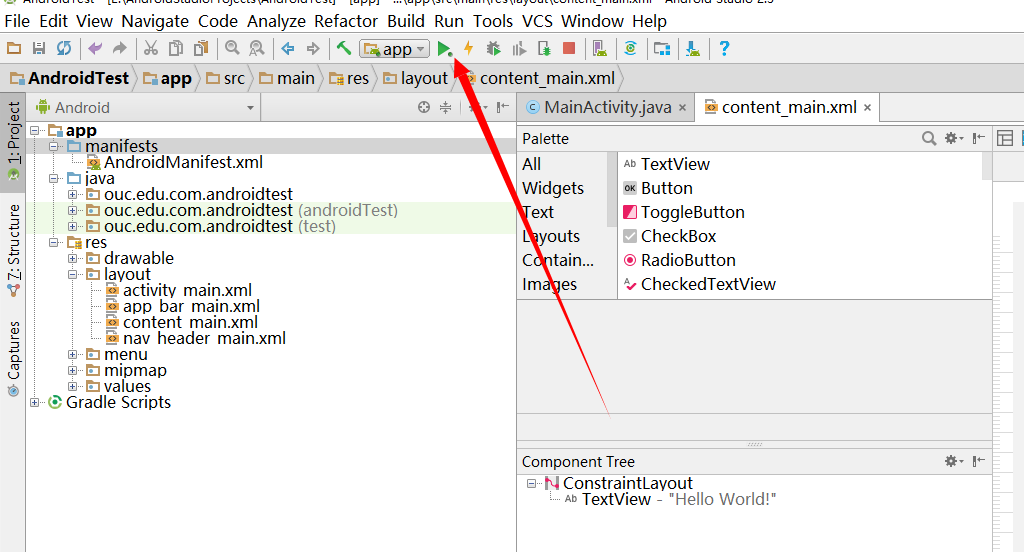




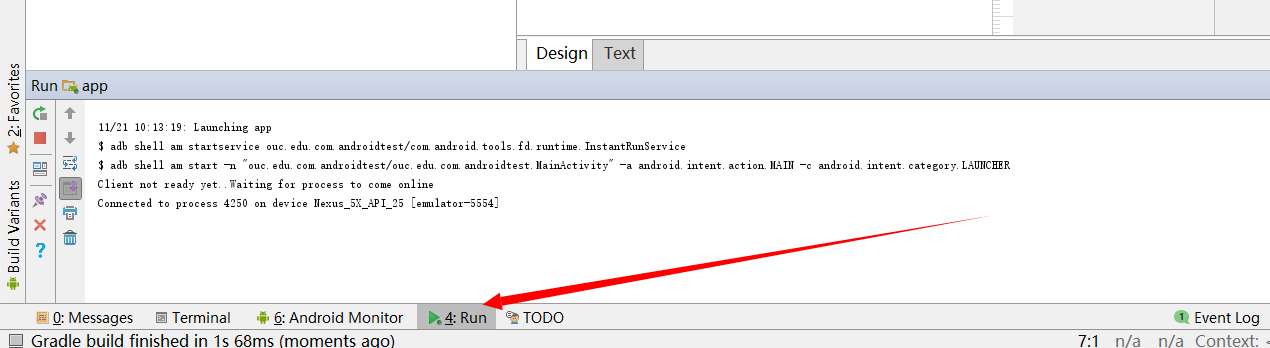


2、在虚拟设备下运行Android项目

（1）直接点击运行按钮



（2）查看日志信息

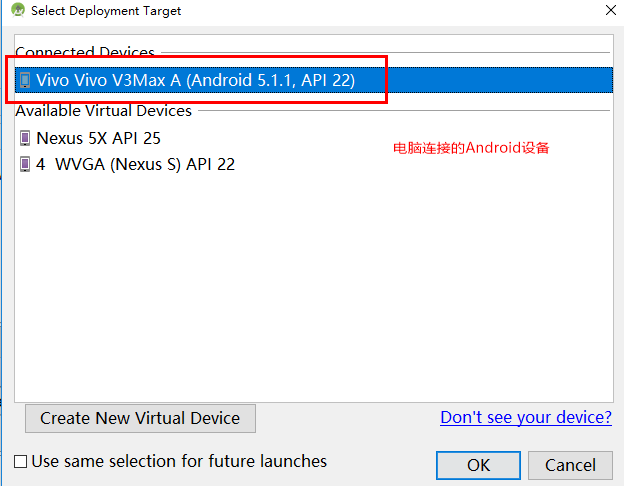


3、在真机上运行Android项目

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

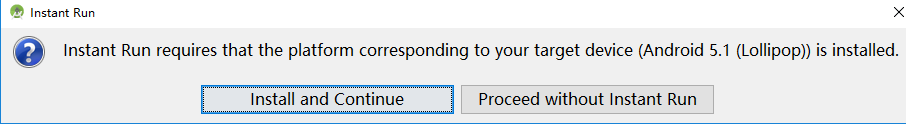
（2）将Android手机通过USB连接电脑，并打开手机的USB调试

（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

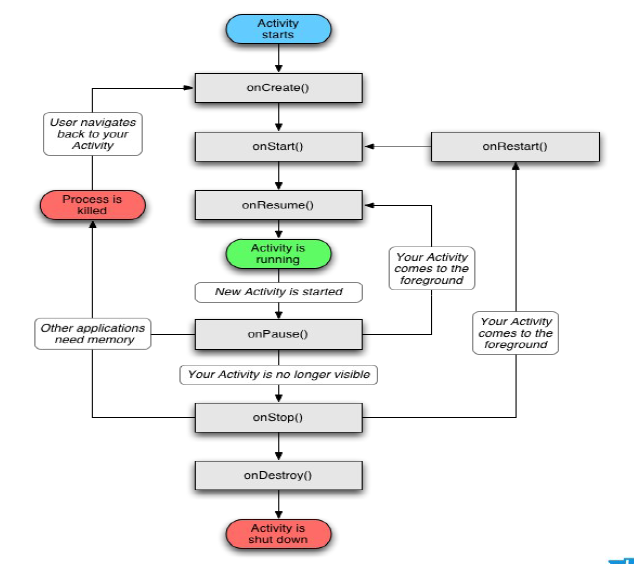
七、Activity

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

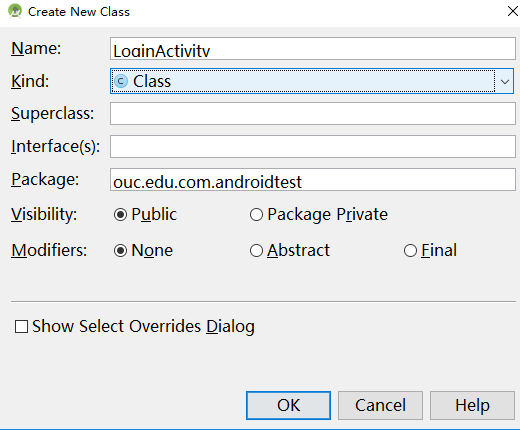
2、Activity生命周期

* 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

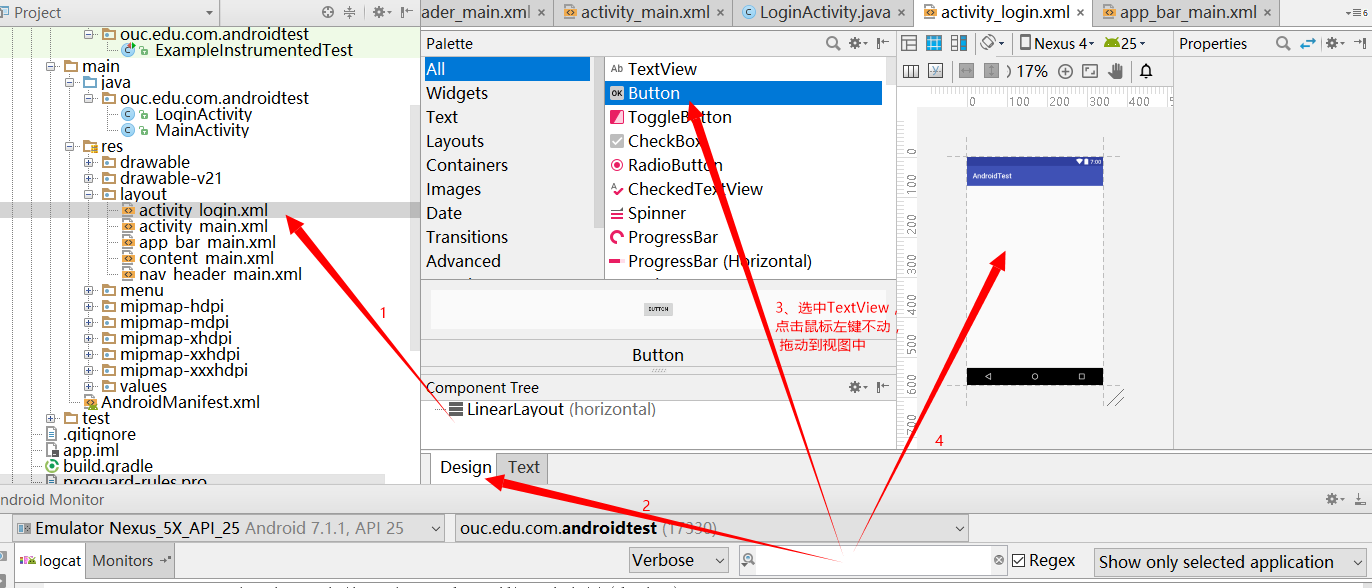
<**TextView android:id="@+id/ tv\_test"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="TextView"**/>

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、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

}

}

开机自启动

* 新建一个接收类，并实现BroadcastRecevicer，重写Onreceive方法

Intent intent =**new** Intent(context,MyReceive.**class**);

intent.setFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK);

context.startActivity(intent);

* 在Manifest.xml注册此类，并添加

<**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.BOOT\_COMPLETED"** />  
 <**category android:name="android.intent.category.HOME"** />  
</**intent-filter**>

）

视频连接：http://study.163.com/course/courseLearn.htm?courseId=270014#/learn/video?lessonId=377177&courseId=270014

Handler控制PorgressBar

* 启动Handler

handler.post(runnable);

* 创建Handler

Handler handler=new Handler(){

//接收消息

Public void handlerMessage(Message msg){

Pb.setProgess(msg.arg1);//其中pb为ProgresBar的对象

handler.post(runnable);

if(msg.arg1==100){

handler.removeCallbacks(runnable);

}

}

}

Runnable runnable=new Runnable(){

Int i=0;

Public void run(){

I+=10;

//Message是Android系统所提供的的消息对象

Message message=handler.obtainMessage();

//message.arg1是系统提供的，并且会节省系统资源消耗

message.arg1=i;

Try{

Thread.sleep(1000); //由于线程执行太快，故而可使线程睡眠1秒

}catach(InterruptedException e){

e.printStackTrace();

}

//发送消息

handler.sendMessage(message);

}

}