1、怎样监听Home键。

**boolean** isFinish = **false**;

@Override

**protected** **void** onStop() {

**super**.onStop();

**if** (isFinish == **false**) {

Toast.*makeText*(getApplicationContext(), "you click home", Toast.*LENGTH\_SHORT*).show();

}

}

@Override

**public** **void** finish() {

**super**.finish();

isFinish = **true**;

}

2、怎样实现应用截屏。

button1.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

View vv = getWindow().getDecorView();

vv.setDrawingCacheEnabled(true);

Bitmap bitmap = vv.getDrawingCache();

Rect appRect = new Rect();

getWindow().getDecorView()

.getWindowVisibleDisplayFrame(appRect);

int statusHeight = appRect.top;

DisplayMetrics displayMetrics = new DisplayMetrics();

getWindow().getWindowManager().getDefaultDisplay()

.getMetrics(displayMetrics);

int screenH = displayMetrics.heightPixels;

int screenW = displayMetrics.widthPixels;

Bitmap bitmap2 = Bitmap.createBitmap(bitmap, 0, statusHeight,

screenW, screenH - statusHeight);

vv.destroyDrawingCache();

try {

FileOutputStream fo = new FileOutputStream(File

.createTempFile("capture", ".jpg", new File(

"/sdcard")));

bitmap2.compress(Bitmap.CompressFormat.PNG, 90, fo);

Toast.makeText(getApplicationContext(), "截屏成功", Toast.LENGTH\_SHORT).show();

} catch (FileNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

});

3、利用粘贴板技术在应用之间传递数据。

ClipboardManager clipboardManager = (ClipboardManager) getSystemService(Context.*CLIPBOARD\_SERVICE*);

ClipData clipData = ClipData.*newPlainText*("clipText", "Hello I am John");

clipboardManager.setPrimaryClip(clipData);

Toast.*makeText*(getApplicationContext(), clipboardManager.getPrimaryClip().getItemAt(0).getText().toString(), Toast.*LENGTH\_LONG*).show();

4、利用全局对象传递数据。

s1:

**package** com.example.counttest;

**import** android.app.Application;

**public** **class** MyApp **extends** Application {

**public** String name;

}

s2: <application

android:name=*"com.example.counttest.MyApp">*

s3: MyApp app = (MyApp) getApplicationContext();

app.name = "gogo";

s4:Toast.*makeText*(getApplicationContext(),app.name, Toast.*LENGTH\_LONG*).show();

5、怎样制作这个窗口



s1:

制作一个主题包含：

<item name="android:windowIsTranslucent">true</item>

并在activity中加载这个主题

s2:

强行设置activity的宽高以及位置

WindowManager.LayoutParams layoutParams = getWindow().getAttributes();

layoutParams.gravity = Gravity.***TOP***;

layoutParams.x = 0;

layoutParams.y = 0;

layoutParams.width = 200;

layoutParams.height = 300;

getWindow().setAttributes(layoutParams);

6、在程序中模拟按home键的效果。

Intent intent = **new** Intent("android.intent.action.MAIN");

intent.addCategory("android.intent.category.HOME");

startActivity(intent);

7、通过classname直接调用其他app的activity

Context ct = createPackageContext("com.android.cpp.browser", Context.***CONTEXT\_IGNORE\_SECURITY***|Context.***CONTEXT\_INCLUDE\_CODE***);

Class c = ct.getClassLoader().loadClass("com.android.cpp.browser.MainActivity");

intent.setClass(ct, c);

intent.setData(Uri.*parse*("http://www.taobao.com"));

startActivity(intent);

8、在手机方向改变的时候不调用oncreate ondestory重绘窗口的方法。

在activity里面声明：android:configChanges=*"orientation|screenSize"*

*而且配置了之后在activity才会调用onConfigurationChanged（Configuration c）这个方法*

9、说说Activity启动有哪几种模式分别说说它们的区别。

standard、singleTop、singleTask、singleInstance

10、说说启动Intent启动Activity的几个Flag

s1:FLAG\_ACTIVITY\_SINGLE\_TOP

**singleTop用法是一样的**

s2:FLAG\_ACTIVITY\_CLEAR\_TOP

用法有两种

a:**若要显示的窗口是standard模式，而且没有与其他flag连用，那么则将压在这个Activity上的Activity出栈，包括自身然后再新建一个自身的Activity。** b：**若这个将要启动的Activity是其他三种模式，或者与FLAG\_ACTIVITY\_SINGLE\_TOP连用的话那么则将其上面的Activity出栈，自身不出栈，调用newIntent**

s3:FLAG\_ACTIVITY\_NEW \_TASK

**用法相当于以singleTask模式调用另一个程序的窗口**

s4:FLAG\_ACTIVITY\_CLEAR\_TASK

**清除回退栈中的所有Activity ，然后重新建立一个对象压入栈，注意此标志必须和FLAG\_ACTIVITY\_NEW \_TASK连用才会起作用。**

s5:FLAG\_ACTIVITY\_REORDER\_TO\_FRONT

**作用只是将这个在栈中的元素调到最前面来，其他元素依然保持相对顺序。**

**11、android:taskAffinity=*"com.example.aproject "***

***这个参的作用是：***

**例如有A、B两个APP在A中调用了B,并使用了FLAG\_ACTIVITY\_CLEAR\_TASK标志，若此时A隐藏了，并打开A，若没有这个参数的话，打开的会是A（因为它在不同的回退栈里面），如果有这个参数的话那么打开的是B界面（A、B界面在同一个回退栈里面）。注意这个参数是在B activity中声明的 com.example.aproject是指向目标APP的package name.**

**12 activity-alias 的用法**

<activity-alias

android:icon=*"@drawable/box"*

android:name=*"sss"*

android:label=*"BOX"*

android:targetActivity=*"BoxActivity"* >

<intent-filter>

<action android:name=*"android.intent.action.MAIN"* />

<category android:name=*"android.intent.category.LAUNCHER"* />

</intent-filter>

</activity-alias>

name、label、targetActivity、icon这四个属性都要设置，如果<intent-filter>

**不设置为MAIN和LAUCHER的话应用程序是不会设置入口的**

**判断是从不是从别名过来的方法**

**getComponentName().getShortClassName().equals("xxx别名");**

**13 RTL layout above api 17 4.2**

**s1:** <application

android:supportsRtl=*"true"*

*s2:* android:layoutDirection="rtl"

**14 什么是onCreateContextMenu，怎么用？一般在长按的情况下弹出菜单**

**好处是在简单的情况下可以代替popopwindow、alertdialog**

Button bt;

@Override

**protected** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_main*);

bt = (Button) findViewById(R.id.*button1*);

//这一行很重要，它实现了单击弹出选择菜单

bt.setOnClickListener(**new** OnClickListener() {

@Override

**public** **void** onClick(View v) {

bt.showContextMenu();

}

});

//为某个按钮注册ContextMenu

registerForContextMenu(bt);

}

@Override

**public** **void** onCreateContextMenu(ContextMenu menu, View v,

ContextMenuInfo menuInfo) {

**super**.onCreateContextMenu(menu, v, menuInfo);

menu.add("man");

menu.add("woman");

}

//选择

@Override

**public** **boolean** onContextItemSelected(MenuItem item) {

Toast.*makeText*(**this**, item.getTitle(), Toast.*LENGTH\_SHORT*).show();

**return** **super**.onContextItemSelected(item);

}

15、通过反射通过名字在res 里面获取ID

**public** **int** getResourceIdByName(String name){

//获取内部类

Class [] classes = R.**class**.getClasses();

**for** (Class class1 : classes) {

**try** {

//获取常量

Field field = class1.getField(name);

//获取常量值

**return** field.getInt(**null**);

} **catch** (NoSuchFieldException e) {

e.printStackTrace();

} **catch** (IllegalAccessException e) {

e.printStackTrace();

} **catch** (IllegalArgumentException e) {

e.printStackTrace();

}

}

**return** -1;

}

**16 字符串的使用**

**s1:复数 getResources().getQuantityString(R.plurals.*song*, 0)；**

<plurals name=*"song"*>

<item quantity=*"zero"*>there is no song</item>

<item quantity=*"one"*>1 song</item>

<item quantity=*"other"*>%d songs are found</item>

</plurals>

**s2：占位符**

<string name="welcome">you are %1$s! group, you have %2$d people</string>

Toast.*makeText*(getApplicationContext(),String.*format*(getResources().getString(R.string.*welcome*), "ppp",100), Toast.*LENGTH\_LONG*).show();

%1$s! %1代表第一个， $s!代表多个字符串

**S3：用html 格式化**

<b> 加粗

<i> 斜体

<u> 下划线

**17、思考什么是资源数组资源，怎么用？**

<array name=*"pic"*>

<item>@drawable/ic\_launcher</item>

<item>@drawable/box</item>

<item>@drawable/sweep</item>

</array>

TypedArray pic = getResources().obtainTypedArray(R.array.*pic*);

Drawable d = pic.getDrawable(1);

**int** color = pic.getColor(0, 0);

**18 transition 动画的用法（淡入淡出动画）**

**s1:**

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

<transition xmlns:android=*"http://schemas.android.com/apk/res/android"* >

<item android:drawable=*"@drawable/lamp\_off"*/>

<item android:drawable=*"@drawable/lamp\_on"*/>

</transition>

**s2:**

<ImageView

android:id=*"@+id/imageView1"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignParentTop=*"true"*

android:layout\_centerHorizontal=*"true"*

**android:src=*"@drawable/tran"*** />

**s3:**

TransitionDrawable drawable = (TransitionDrawable) imageView

.getDrawable();

**if** (isOn) {

drawable.reverseTransition(1000);

isOn = **false**;

button1.setText("开灯");

}**else** {

drawable.startTransition(1000);

isOn = **true**;

button1.setText("关灯");

}

**18 clip用法（剪裁）**

**s1:**

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

<clip xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:clipOrientation=*"horizontal"*

android:drawable=*"@drawable/tidjiao"*

android:gravity=*"left"* >

</clip>

**s2:**

<ImageView

android:id=*"@+id/imageView1"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

**android:src=*"@drawable/load"*** />

**s3:**

**public** **class** LoadingActivity **extends** Activity {

ImageView imageView;

ClipDrawable clipDrawable;

**int** count;

Handler handler = **new** Handler() {

@Override

**public** **void** handleMessage(Message msg) {

**super**.handleMessage(msg);

clipDrawable.setLevel(count);

}

};

@Override

**protected** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_loading*);

imageView = (ImageView) findViewById(R.id.*imageView1*);

clipDrawable = (ClipDrawable) imageView.getDrawable();

**new** Timer().schedule(**new** TimerTask() {

@Override

**public** **void** run() {

**if** (count < 9900) {

count += 100;

handler.sendEmptyMessage(0x123);

}

}

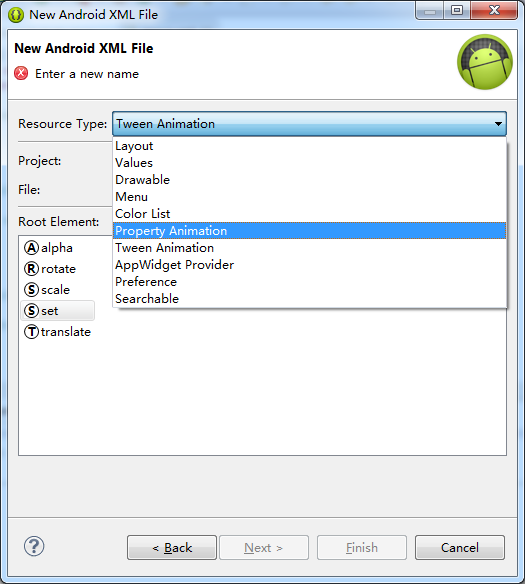
}, 0, 100);

}

}

**19、属性动画 Property**

**s1:在anim文件夹下新建一个**

****

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

<set xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:ordering=*"sequentially"* >

<objectAnimator

android:duration=*"3000"*

android:propertyName=*"x"*

android:valueTo=*"700"*

android:valueType=*"intType"* >

</objectAnimator>

<objectAnimator

android:duration=*"3000"*

android:propertyName=*"y"*

android:valueTo=*"700"*

android:valueType=*"intType"* >

</objectAnimator>

</set>

**S2：**

**public class PropertyAnimActivity extends Activity {**

**private TextView tv;**

**@Override**

**protected void onCreate(Bundle savedInstanceState) {**

**super.onCreate(savedInstanceState);**

**setContentView(R.layout.activity\_property\_anim);**

**tv = (TextView) findViewById(R.id.tv);**

**{**

**AnimatorSet set = (AnimatorSet) AnimatorInflater.loadAnimator(this, R.anim.move);**

**set.setTarget(new Dog());**

**set.start();**

**}**

**}**

**class Dog {**

**private int x;**

**private int y;**

**public int getX() {**

**return x;**

**}**

**public void setX(int x) {**

**this.x = x;**

**tv.layout(x, tv.getTop(), x+tv.getMeasuredWidth(), tv.getBottom());**

**}**

**public int getY() {**

**return y;**

**}**

**public void setY(int y) {**

**this.y = y;**

**tv.layout(tv.getLeft(), y, tv.getRight(), y+tv.getMeasuredHeight());**

**}**

**}**

**}**

**20、帧动画Frame**

s1:<?xml version=*"1.0"* encoding=*"utf-8"*?>

<animation-list xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:oneshot=*"false"* >

<item

android:drawable=*"@drawable/loading\_01"*

android:duration=*"150"*>

</item>

<item

android:drawable=*"@drawable/loading\_02"*

android:duration=*"150"*>

</item>

</animation-list>

s2:

<ImageView

android:background=*"@anim/loading"*

android:id=*"@+id/imageView1"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_centerInParent=*"true"* />

s3:

AnimationDrawable drawable = (AnimationDrawable) imageView.getBackground();

// or imageView.setBackgroundResource(R.anim.loading);

drawable.stop();

drawable.start();

OR:

AnimationDrawable drawable2 = (AnimationDrawable) getResources().getDrawable(R.anim.*loading*);

imageView.setImageDrawable(drawable2);

drawable2.stop();

drawable2.start();

**21、TextView**

String html = "<big><a href=\"http://www.baidu.com/s?wd=北大青鸟\">学电脑到北大青鸟</a></big>";

tv.setText(Html.*fromHtml*(html));

//设置后才能点击后用浏览器打开

tv.setMovementMethod(LinkMovementMethod.*getInstance*());

**小技巧：利用Html标签向TextView里面插入图像**

textView1 = (TextView) findViewById(R.id.*textView1*);

String html2 = "<img src='box'>";

CharSequence c = Html.*fromHtml*(html2, **new** ImageGetter() {

@Override

**public** Drawable getDrawable(String source) {

**try** {

**Drawable drawable = getResources().getDrawable(R.drawable.class.getField(source).getInt(null));**

**drawable.setBounds(0, 0, drawable.getIntrinsicWidth(), drawable.getIntrinsicHeight());**

**return** drawable;

} **catch** (IllegalAccessException | IllegalArgumentException

| NoSuchFieldException e) {

e.printStackTrace();

}

**return** **null**;

}

}, **null**);

textView1.setText(c);

使用Span设置可扩展文本

textView2 = (TextView) findViewById(R.id.*textView2*);

String a = "123123";

SpannableStringBuilder spannableStringBuilder = **new** SpannableStringBuilder(a);

Drawable drawable = **null**;

**try** {

drawable = getResources().getDrawable(R.drawable.**class**.getField("sweep").getInt(**null**));

drawable.setBounds(0, 0, drawable.getIntrinsicWidth(), drawable.getIntrinsicHeight());

spannableStringBuilder.setSpan(new ImageSpan(drawable), 0, a.length(), Spannable.*SPAN\_EXCLUSIVE\_EXCLUSIVE*);

textView2.setText(spannableStringBuilder);

} **catch** (NotFoundException e) {

e.printStackTrace();

} **catch** (IllegalAccessException e) {

e.printStackTrace();

} **catch** (IllegalArgumentException e) {

e.printStackTrace();

} **catch** (NoSuchFieldException e) {

e.printStackTrace();

}

**example2:**

textView3 = (TextView) findViewById(R.id.*textView3*);

String ab = "打电话给我老大";

SpannableStringBuilder spannableStringBuilder2 = **new** SpannableStringBuilder(ab);

spannableStringBuilder2.setSpan(**new** ClickableSpan() {

@Override

**public** **void** onClick(View widget) {

Intent intent = **new** Intent(Intent.*ACTION\_CALL*,Uri.*parse*("tel:18883285243"));

startActivity(intent);

}

}, 0,ab.length(), Spannable.*SPAN\_EXCLUSIVE\_EXCLUSIVE*);

textView3.setText(spannableStringBuilder2);

textView3.setMovementMethod(LinkMovementMethod.*getInstance*())

**22、小技巧：当edittextview获得焦点之后阻止屏幕上移的方法。**

**getWindow().setSoftInputMode(WindowManager.LayoutParams.*SOFT\_INPUT\_ADJUST\_RESIZE*);**

**23、加载图片的一种策略**

imageView = (ImageView) findViewById(R.id.*imageView1*);

Options options = **new** Options();

//加载为原图的五分之一

**options.inSampleSize = 5;**

Bitmap bitmap = BitmapFactory.*decodeResource*(getResources(), R.drawable.*heart*, options);

imageView.setImageBitmap(bitmap);

24、旋转bitmap

Matrix m = **new** Matrix();

m.setRotate(45);

Bitmap bitmap2 = bitmap.*createBitmap*(bitmap, 0, 0, bitmap.getWidth(), bitmap.getHeight(),m , **true**);

**25、惰性装载 <ViewStub>**

**s1:**

<ViewStub

android:id=*"@+id/stubView"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:inflatedId=*"@+id/bt\_layout"*

android:layout=*"@layout/custom"* />

**s2:**

**public** **class** StubViewActivity **extends** Activity {

View vv;

@Override

**protected** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_stub\_view*);

vv = findViewById(R.id.*stubView*);

}

**public** **void** loadView(View v) {

//在需要的时候加载并显示

**if (vv != null) {**

**//(( ViewStub)vv).inflate(); OR vv.setVisibility(View.*VISIBLE*);**

**}**

}

}

**25、AlertDialog**



实现代码

alertDialog = **new** AlertDialog.Builder(**this**,

//实现白色

AlertDialog.*THEME\_HOLO\_LIGHT*)

.setTitle("请选择城市")

.setSingleChoiceItems(**new** String[] { "四川", "重庆" }, 0,

**new** OnClickListener() {

@Override

**public** **void** onClick(DialogInterface dialog,

**int** which) {

}

}).setPositiveButton("完成", **new** OnClickListener() {

@Override

**public** **void** onClick(DialogInterface dialog, **int** which) {

}

**//实现点击外部不消失**

}).setCancelable(**false**).create();

alertDialog.show();

**26、Popupwindow点击外部消失的方法**

[**android：点击popupwindow以外区域 popupwindow自动消失(转载)**](http://www.cnblogs.com/joey-home/archive/2012/07/02/2573855.html)

方法一（这种方法可以处理popupwindows dimiss的时候一些其他的操作，比如让其他控件的隐藏，消失等）:

代码如下popupWindow.setFocusable(false);//focusable要为false(不设置默认的就是False)；  
//这是Activity 的OnTouchEvent。OnTouchEvent代表的是Activity 获得事件（即为PopupWindow之外）

@Override

public boolean onTouchEvent(MotionEvent event) {

// TODO Auto-generated method stub

if (popupWindow != null && popupWindow.isShowing()) {

popupWindow.dismiss();

popupWindow = null;

}

return super.onTouchEvent(event);

}

方法二：设置popupWindow参数（这种方法只能让自身消失，不能够提供其他伴随操作，比如让其他控件的隐藏，消失等）

pop = new PopupWindow(view, ViewGroup.LayoutParams.FILL\_PARENT, ViewGroup.LayoutParams.WRAP\_CONTENT);  
pop.setBackgroundDrawable(new BitmapDrawable());  
pop.setOutsideTouchable(true);

}

方法三：响应popup里面view的触摸事件

pop = **new** PopupWindow(LayoutInflater.*from*(**this**).inflate(R.layout.*tv*,

**null**), LayoutParams.*MATCH\_PARENT*, LayoutParams.*WRAP\_CONTENT*);

**pop.setFocusable(true);**

**pop.setBackgroundDrawable(new ~~BitmapDrawable~~());**

pop.showAtLocation(

LayoutInflater.*from*(**this**).inflate(

R.layout.*activity\_alert\_dialog*, **null**), Gravity.*CENTER*,0, 0);

27 fragment 添加到回退栈

**public** **void** nextFragment(Fragment fragmentCurrent,Fragment fragmentAdd, **boolean** isAddToBack) {

FragmentTransaction transaction = getFragmentManager()

.beginTransaction();

**if** (isAddToBack) {

transaction.addToBackStack(**null**);

}

transaction.commit();

}

@Override

**public** **void** onBackStackChanged() {

Toast.*makeText*(**this**,

getFragmentManager().getBackStackEntryCount() + "",

Toast.*LENGTH\_SHORT*).show();

}