// 高精度进制转换
SetonclickListener（）
Setimgdrawable（）
if((username.equals("andro")&&password.equals("20"))){
imgbtn1.setImageDrawable(getResources().getDrawable(R.drawable.state1));
user.setVisibility(View.GONE); // 按键消失
Toast.makeText(getApplicationContext(), "默认 Toast 样式",Toast.LENGTH_SHORT).show();

dylayout = (LinearLayout)findViewById(R.id.x); // 动态按钮相关
temp2 = new TextView(context);
temp2.setText("动态添加的按钮！");
temp2.setTextColor(Color.rgb(255, 255, 255));
dylayout.addView(temp2); // 添加
dylayout.removeAllViews(); // 回收

// activity 跳转相关
Intent intent = new new Intent (MainActivity.this, AAA.class);
startActivity(intent); // （1）无参数跳转
startActivityForResult (Intent intent, requestCode) //（2）请求结果跳转
final Bundle bundle = new Bundle(); bundle 用 key-value 存储数据
bundle.putString("txt", name[position]);
intent.putExtras(bundle);
setResult(int resultCode, Intent data) AAA 中
然后 AAA.this.finish(); 或直接 finish();
接收返回的数据/结果的处理函数
onActivityResult(int requestCode, int resultCode, Intent data)
if resultCode==result_ok
String name = idata.getExtras().getString("txt");
（A 向 B 传递数据跳转）
bundle.putString("txt", name[position]);
intent.putExtras(bundle);
startActivity(intent);
然后在 B 的 onCreate 中获取
***新建 Activity 之后需要将该 Activity 注册 Manifest.xml 文件中

```
<activity
    android:name="com.example.hello.Fruit"
    android:label="@string/app_name">
</activity>
```

-------ListView 相关----------
（1）Item.xml 正常线性布局即可，每一项的布局里面加上 textview imageView

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal" >
```

android:gravity="center" 居中
（2）listView，直接新建一个 listView 即可，可以放在别的布局里面

```
<?xml version="1.0" encoding="utf-8"?>
<ListView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id = "@+id/fruits"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >
</ListView>
```

（3）存放数据 先用 setData 放好数据
private List<Map<String, Object>> mData = new ArrayList<Map<String, Object>>();
SimpleAdapter adapter = new SimpleAdapter(this, mData, 数据源
R.layout.item, item 布局文件，下面是 list 中数据的 key，item 中的 id
new String[] { "txt", "img" }, new int[] { R.id.txt, R.id.img });
List.setAdapter(adapter);

```
OnItemClickListener listener = new OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> arg0, View view, int position,
            long id) {
        // TODO Auto-generated method stub
        final Bundle bundle = new Bundle();
        bundle.putString("txt", name[position]);
        // bundle.putString("","Jack");
        Intent intent = new Intent(Fruit.this, MainActivity.class);
        intent.putExtras(bundle);
        setResult(android.app.Activity.RESULT_OK, intent);
        finish();
    }
};
```

List.setOnItemClickListener(listener); //但是如果这行内容中包含 Button，ImgButton 等控件时就不会被调用
**可修改 android:focusable="false" android:longClickable="true"实现
setData(){ // 利用 map 把数据存进去
Map<String, Object> map = new HashMap<String, Object>();
for (int i = 0; i < name.length; i++) {
map = new HashMap<String, Object>();
map.put("txt", name[i]);
map.put("img", imgId[i]);
mData.add(map);
}
（4）更新数据 simpleAdapter.notifyDataSetChanged();

---app widget 相关---
（1）创建 AppWidget 布局文件 widget_layout.xml，布局中有一个 ImageView 和一个 TextView
（2）在/res/xml/下创建类型 AppWidget Provider 的 Android XML 文件
（3）设置 provider 的长 宽及布局 layout 为上面的 widget.xml
(4) 创建 AppWidgetProvider 的子类， 重写 onUpdate 方法， 添加 widget 上 ImageView 的点击事件响应。

```
public void onUpdate(Context context,AppWidgetManager appWidgetManager,int[] appWidgetIds){

    Intent clickIntent = new Intent(context,Fruit.class);
    PendingIntent pi = PendingIntent.getActivity(context, 0, clickIntent, 0);
    RemoteViews remoteView = new RemoteViews(context.getPackageName(), R.layout.widget_layout);
    remoteView.setOnClickPendingIntent(R.id.widget_image, pi);
    appWidgetManager.updateAppWidget(appWidgetIds, remoteView);
    super.onUpdate(context, appWidgetManager, appWidgetIds);
}
```

(5) 在 Manifest 文件中注册 Widget：

```
<receiver
    android:name = "com.example.ex1.Widget"
    android:label="@string/app_name">
    <meta-data
        android:name="android.appwidget.provider"
        android:resource="@xml/provider"/>
    <intent-filter >
        <action android:name="android.appwidget.action.APPWIDGET_UPDATE"/>
        <action android:name ="BROADCAST"/>
    </intent-filter>
</receiver>
```

----对话框---

```
findViewById(R.id.button1).setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        new AlertDialog.Builder(DialogDemo.this)
                .setIcon(R.drawable.gong1)
                .setTitle("提示文字Dialog")
                .setPositiveButton("确定",
                        new DialogInterface.OnClickListener() {
                            @Override
                            public void onClick(DialogInterface dialog,
                                    int which) {
                                // TODO Auto-generated method stub

                            }
                        }).setNegativeButton("取消", null).create()
                .show();
    }
});
```

如果有 3 个选项，则中间的为 setNeutralButton

```
lv.setOnItemClickListener(new OnItemClickListener(){
    @Override
    public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,
            long arg3) {
        Builder b = new AlertDialog.Builder(MainActivity.this);
        b.setTitle("详情");
        LinearLayout l = (LinearLayout)getLayoutInflater().inflate(
                R.layout.details, null);
        final TextView name = (TextView) l.findViewById(R.id.textView1);
        name.setText("hi:"+cursor.getString(cursor.getColumnIndex("name")));
        b.setView(l);
        b.create().show();
    }});
```

点击出现详情界面。

--多线程—
class DownLoadThread implements Runnable
new Thread(new DownLoadThread(code.getText().toString())).start();
final ProgressDialog dialog = new ProgressDialog(MainActivity.this);
dialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
dialog.setTitle("正在生成验证码");
dialog.setMessage("请稍后");
dialog.setIndeterminate(false);
dialog.setCancelable(true);
dialog.show();
handler = new Handler(){
public void handleMessage(Message msg){
im.setImageBitmap((Bitmap)msg.getData().getParcelable("im"));
//im.setImageBitmap((Bitmap) msg.obj);
dialog.dismiss();
发数据：
Bundle bundle = new Bundle(); bundle.putParcelable("im", bm);
Message msg = MainActivity.handler.obtainMessage();
msg.setData(bundle);
MainActivity.handler.sendMessage(msg);

-- SharedPreferences—

（1）创建的存储文件保存在/data/data/<package name>/shares_prefs

```
/** 存储的文件名 */
public static final String DATABASE = "Database";
/** 存储的文件路径：/data/data/<package name>/shares_prefs + 文件名.xml */
public static final String PATH = "/data/data/com.example.test/shared_prefs/Database.xml";
```

（2）通过 Context.getSharedPreferences 方法获取当前 SharedPreferences 对象

```
//获取当前应用的sharepreferences
sp = getSharedPreferences(DATABASE,Activity.MODE_PRIVATE);
```

（3）读取数据 count = sp.getInt("autoCount", 0); //后面是默认值

通过 editor 来编辑 ed = sp.edit();

ed.putInt("autoCount", nameStr.size());

插入数据：调用 Editor.putxxxx 方法，两个参数分别为键和值。

获取数据：调用 sp.getxxxx 方法，两个参数分别为键和默认值。

删除数据：调用 Editor.remove 方法，参数为指定的键。

编辑完之后 ed.commit（） Clear（）删除所有数据

***获取所有数据

```
Map<String, ?> allEntries = prefA.getAll();
for (Map.Entry<String, ?> entry : allEntries.entrySet()) {
    Log.d("map values", entry.getKey() + ": " +
entry.getValue().toString());
}
```

--AutoCompleteTextView—

（1）首先在布局文件中创建 AutoCompleteTextView，然后在 Activity 中找到该控件。

```
name = (AutoCompleteTextView) findViewById(R.id.editname);
```

（2）设置适配器，将提示数据添加到适配器中

```
//设置适配器，将提示数据添加到适配器中
ArrayAdapter<String> autoadapter=new ArrayAdapter<String>(this,
        android.R.layout.simple_dropdown_item_1line,autoStr);
```

```
private ArrayList<String> nameStr = new ArrayList<String>();
ArrayAdapter<String>nameAdapter = new ArrayAdapter<String>(this,
android.R.layout.simple_list_item_1,nameStr);
```

（3）name.setAdapter(nameAdapter);

name.setThreshold(1);

（）每次打开程序都需要通过 SharePreferences 获取本地已保存的文件名，并将这些文件名赋值到提示信息数组中。另外每次保存文件后，记得更新提示信息数组，重新设置适配器和设置控件适配器，这样才能即时更新提示信息。

--buffer reader

```
StringBuffer buff = new StringBuffer();
BufferedReader reader = new BufferedReader(new InputStreamReader(
        new FileInputStream(PATH)));
```

调用 readLine 方法，读取一行。

```
while ((str = reader.readLine()) != null) {
    buff.append(str);//该方法的作用是追加内容到当前StringBuffer对象的末尾，类似于字符串的连接。
```

返回读取的数据内容。

```
return buff.toString();//转换String
```

---数据库---

（1） 创建 MyDB，继承 SQLiteOpenHelper 的子类

```
public class MyDB extends SQLiteOpenHelper {
    public MyDB(Context context, String name,
            CursorFactory factory, int version) {
        super(context, name, factory, version);
        // TODO Auto-generated constructor stub
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        // TODO Auto-generated method stub
        String sql = "Create table finalexam(_id integer primary key autoincrement," +
                "name text);";
        db.execSQL(sql);
    }
    public void onUpgrade(SQLiteDatabase arg0, int arg1, int arg2) {
    public void insertItem(ContentValues values){
        SQLiteDatabase db = getWritableDatabase();
        db.insert("finalexam", null, values);
        db.close();
    }
    public Cursor query(){
        SQLiteDatabase db = getReadableDatabase();
        return db.query("finalexam",null,null,null,null,null,null);
    }
}
```

Values 参数：

完全退出程序 A.getinstance.exit() 或 finish()

在 SDCard 中创建与删除文件权限 MOUNT_UNMOUNT_FILESYSTE

写数据 WRITE_EXTERNAL_STORAGE

Cursor cursor2 = (Cursor) ((ListView) parent).getItemAtPosition(position); int getID1 = cursor2.getInt(cursor2.getColumnIndex("_id")); mydatabase.delete_table(getID1); }

```
findViewById(R.id.button1).setOnClickListener(new OnClickListener(){
    @Override
    public void onClick(View arg0) {
        ContentValues v = new ContentValues();
        v.put("name",
                ((EditText)findViewById(R.id.editText1)).getText().toString());
        MyDB a = new MyDB(getApplicationContext(),"fe.db",null,1);
        a.insertItem(v);
        setResult(RESULT_OK);
        finish();
```

ii. update 方法需要使用 ContentValues 和 Where 语句：

```
SQLiteDatabase db = getWritableDatabase();

ContentValues values = new ContentValues();
values.put("<列1>", "<值1>");
values.put("<列2>", "<值2>");
// ....

String whereClause = "<主键列名> = ?"; // where 语句
String[] whereArgs = { "< ? 的值>" };

// 相当于执行" update <TABLE_NAME> set <列1>=<值1>, ... where <主键列名> = ?"
```

iii. delete 方法需要使用 Where 语句：

```
SQLiteDatabase db = getWritableDatabase();

String whereClause = "<主键列名> = ?"; // where 语句
String[] whereArgs = { "< ? 的值>" };

// 相当于执行" delete from <TABLE_NAME> where <主键列名> = ?"
int row = db.delete(TABLE_NAME, whereClause, whereArgs);
```

```
public class MainActivity extends Activity {
    public Cursor cursor;
    ListView lv;
    CursorAdapter a;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        findViewById(R.id.button1).setOnClickListener(new OnClickListener(){
            @Override
            public void onClick(View arg0) {
                Intent i = new Intent(MainActivity.this,AddActivity.class);
                startActivityForResult(i,1);
            }});
        lv = (ListView)findViewById(R.id.listView1);
        cursor = (new MyDB(this,"fe.db",null,1)).query();
        a = new SimpleCursorAdapter(this,R.layout.item,
                cursor, new String[]{"name"}, new int[]{R.id.textView1});
        lv.setAdapter(a);
        lv.setOnItemClickListener(new OnItemClickListener(){
          public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,
                  long arg3) {
              Builder b = new AlertDialog.Builder(MainActivity.this);
              b.setTitle("详情");
              LinearLayout l = (LinearLayout)getLayoutInflater().inflate(
                      R.layout.details, null);
              final TextView name = (TextView) l.findViewById(R.id.textView1);
              name.setText("hi:"+cursor.getString(cursor.getColumnIndex("name")));
              b.setView(l);
              b.create().show();
          }});
    }
    @Override
    protected void onResume() {
        lv = (ListView)findViewById(R.id.listView1);
        cursor = (new MyDB(this,"fe.db",null,1)).query();
        CursorAdapter a = new SimpleCursorAdapter(this,R.layout.item,
                cursor, new String[]{"name"}, new int[]{R.id.textView1});
        lv.setAdapter(a);
        super.onResume();
    }
    @Override
    protected void onActivityResult(int arg0, int arg1, Intent arg2) {
        cursor = (new MyDB(this,"fe.db",null,1)).query();
        ((CursorAdapter)lv.getAdapter()).notifyDataSetChanged();
        super.onActivityResult(arg0, arg1, arg2);
    }
        new TimeThread().start();
        handler = new Handler(){
            @Override
            public void handleMessage(Message msg){
                long systime = System.currentTimeMillis();
                CharSequence systimestr = DateFormat.format("hh:mm:ss", systime);
                tt.setText(systimestr);
            }
        };
    }
    class TimeThread extends Thread{
        @Override
        public void run(){
            do{try{
                    Thread.sleep(1000);
                    Message msg = new Message();
                    handler.sendMessage(msg);
                }catch(InterruptedException e){e.printStackTrace();}
            }while(true);
```