界面控制

```xml
<TextView android:id="@+id/TextView01"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="TextView01">
</TextView>
<EditView android:id="@+id/EditView01"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="EditView01">
</EditView>
```

```java
TextView textView = (TextView)findViewById(R.id.ViewList01);
EditView editView = (EditView)findViewById(R.id.EditView01);
textView.setText("用户名:");
editView.setText("Rajan");
```

```java
button.setOnClickListener(new View.OnClickListener(){
    public void onClick(View view){
        textView.setText("Button按钮");
    }
});
```

Spinner与ListView

```java
Spinner spinner = (Spinner)findViewById(R.id.spinner01);
List<String> list = new ArrayList<String>(); // 泛型
list.add("Spinner 子项1"); // 这里是 add
list.add("Spinner 子项2");
list.add("Spinner 子项3");
ArrayAdapter<String>adapter = new ArrayaAdapter<String>(this, android.R.layout.simple_spinner_i
spinner.setAdapter(adapter); // 最重要的
```

界面布局

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="..."
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"> <!-- 这里比较重要, vertical是布局方向 -->
</LinearLayout>
```

菜单

```java
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getManuInflater();
    inflater.inflate(R.menu.main menu, menu);
    return true;
}
```

Intent，显式启动Activity

```java
Intent intent = new Intent(IntentDemo.this, ActivityToStart.class);
startActivity(intent);
```

Intent过滤器

```xml
<activity android:name=".ActivityToStart"
    android:label="@string/app_name">
    <intent-filter>
        <action android:name="android.intent.ation.VIEW" />
        <categroy android:name="android.intent.category.DEFAULT" />
        <data android:scheme="schemodemo" android:host="edu.hrbeu" />
    </intent-filter>
</activity>
```

```java
Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse("schemodemo://edu.hrbeu/path"));
startActivity(intent);
```

Intent发送广播消息

```java
String UNIQUE_STRING = "edu.hrbeu.BroadcastReceiverDemo";
Intent intent = new Intent(UNIQUE_STRING);
intent.putExtra("key1", "value1");
intent.putExtra("key2", "value2");
intent.putExtra("key3", "value3"); // putExtra 是重点
sendBroadcast(intent); // 发送广播消息
```

线程使用

```java
ptivate Runnable backgroundWork = new Runnable(){
    @Override
    public void run() {
        // 过程代码
    }
}


private Thread workThread;
workThread = new Thread(null, backgroundWork, "WorkThread");
```

Handle更新用户界面

```java
private static Handler handler = new Handler();
public static void UpdateGUI(double refreshDouble) {
    handler.post(RefreshLabel); // 使用post更新
}
private static Runnable RefreshLabel = new Runnable() {
    @Override
    public void run() {
        // 过程代码
    }
}
```

服务绑定

```java
public class MathService extends Service {
    private final IBinder mBinder = LoaclBinder();

    public class LocalBinder extends Binder {
        MathService getService() {
            return MathService.this;
        }
    }

    @Override
    public IBinder onBind(Intent intent) {
        return mBinder;
    }
}


final Intent serviceIntent = new Intent(this, MathService.class); // class 是待启动的项目
bindService(serviceIntent, mConnection, Context.BIND_AUTO_CREATE); // 这里是重点
```

```java
private ServiceConnection mConnection = new ServiceConnection() {
    @Override
    public void onServiceConnected(ComponentName name, IBinder service) {
        mathService = ((MathSercice.LocalBinder)service).getService(); // 这是重点
    }
    @Override
    public void onServiceDisconnectioned(ComponentName name) {
        mathService = null;
    }
};
```