Android 应用开发培训

王　雪明　.
mwongxming@gmail.com

www.3gdci.com

UI的观念与变革

- 区别于桌面系统的窗体
- 屏幕尺寸的限制
- 适合手指触摸的操作体验
  - 菜单(menu)
  - 对话框(Dialog)
  - 提示信息(Toast & Notification)

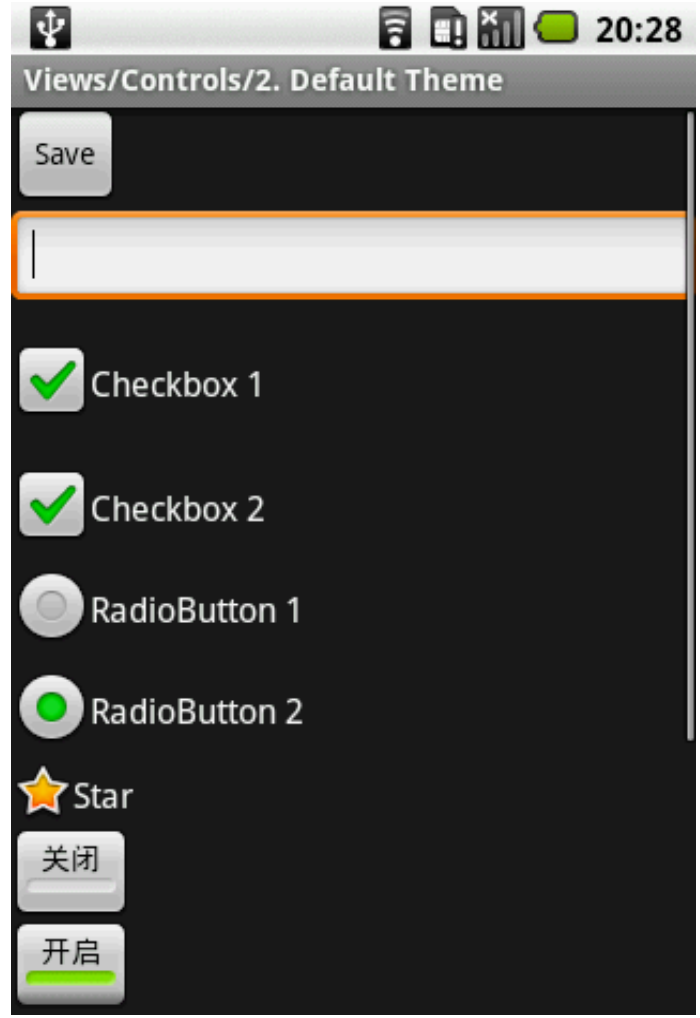
UI 简单就是好

- 简化的界面和人机交互流程
- 独特的UI组件设计
- 使用少量的组件，创建良好的操作体验

常用组

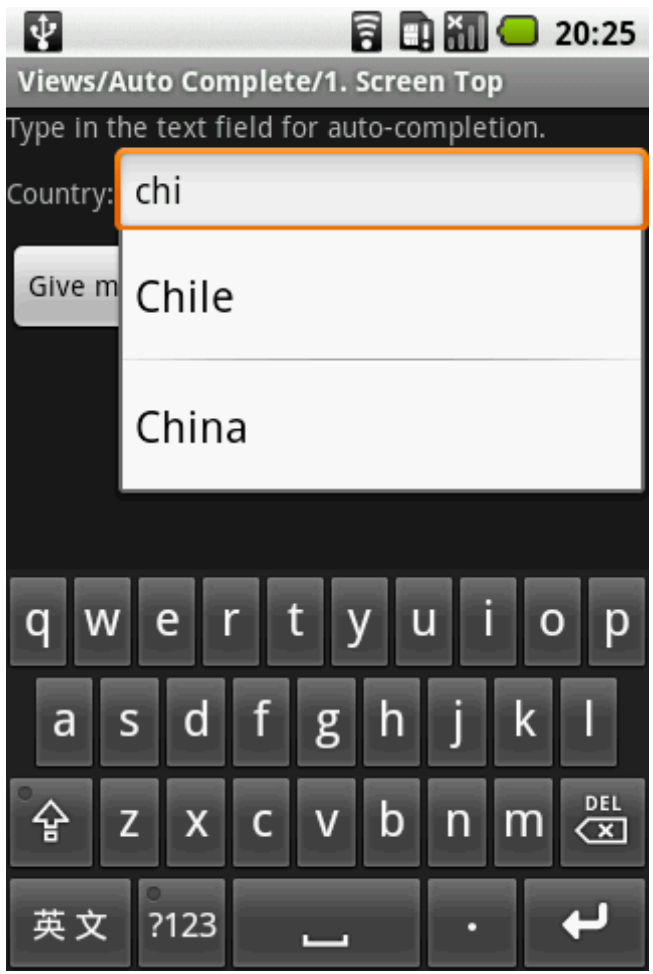
android.widget.TextView
android.widget.Button
android.widget.EditText
android.widget.CheckBox
android.widget.RadioButton
android.widget.RadioGroup
android.widget.ToggleButton

UI组件 可以自定义风格



Android 基础UI组件

android.widget.
AutoCompleteTextView



android.widget.Spinner

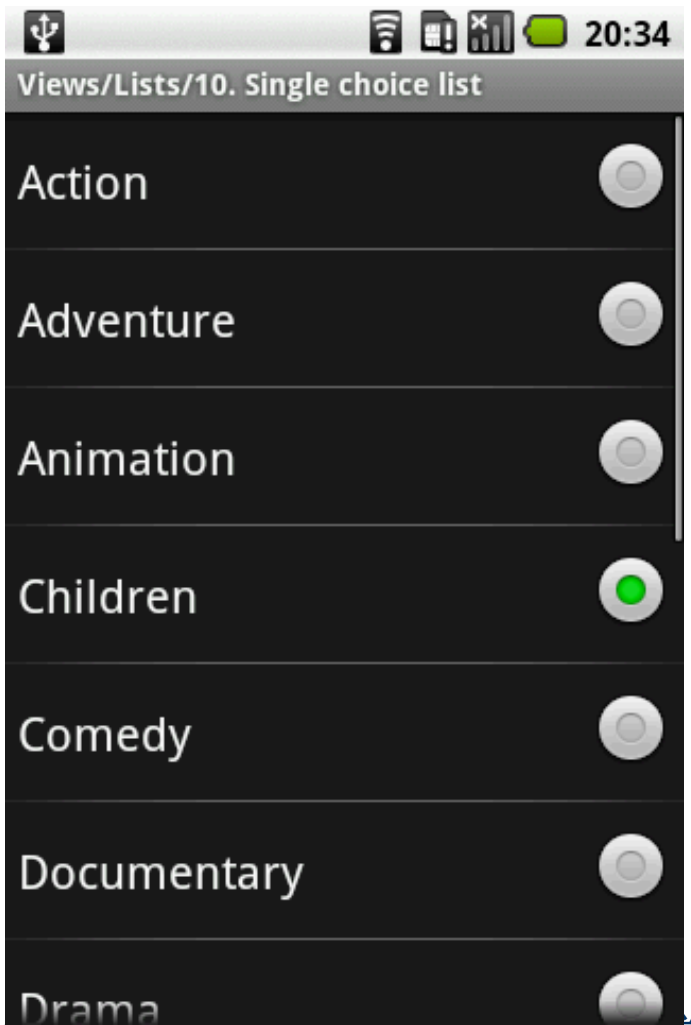
Android 基础UI组件

android.widget.DatePicker

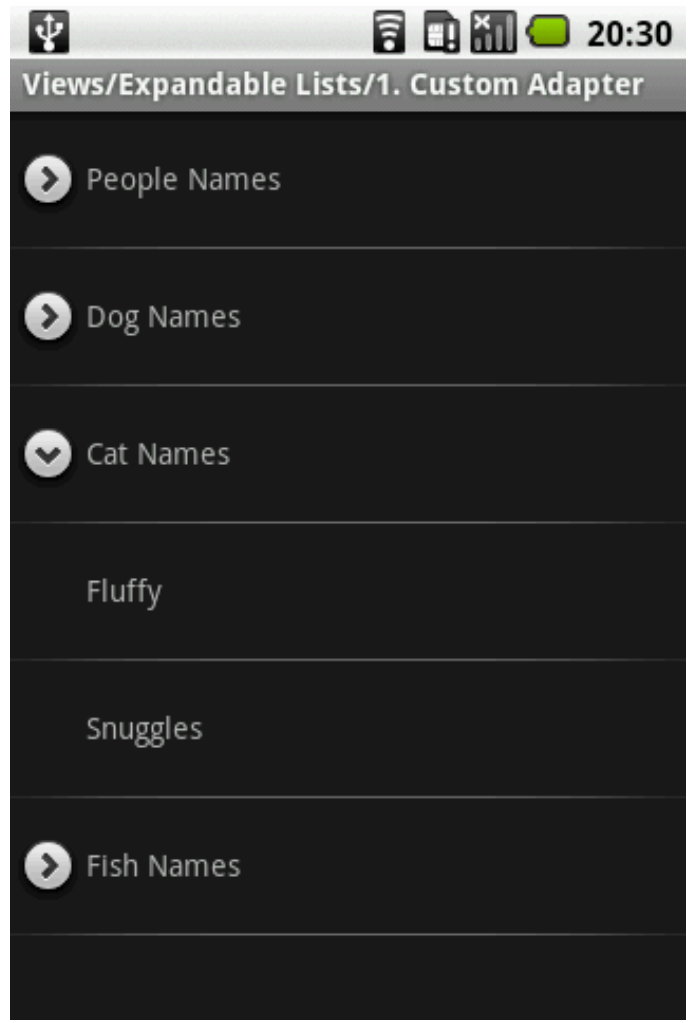
android.widget.TimePicker

Android 基础UI组件

android.widget.ListView



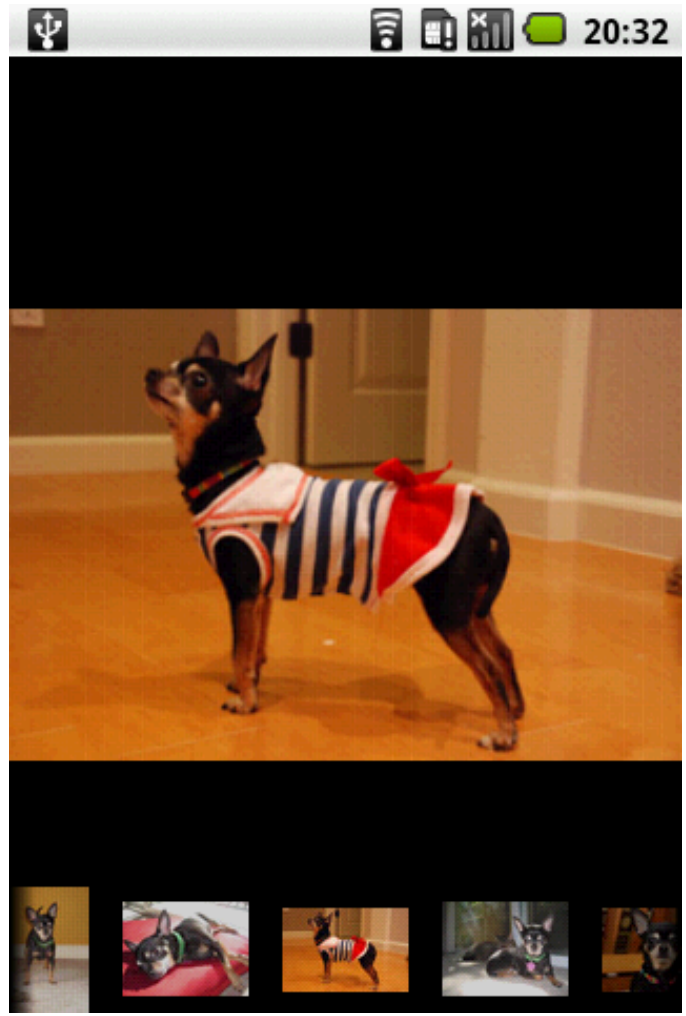
android.widget.
ExpandableListView



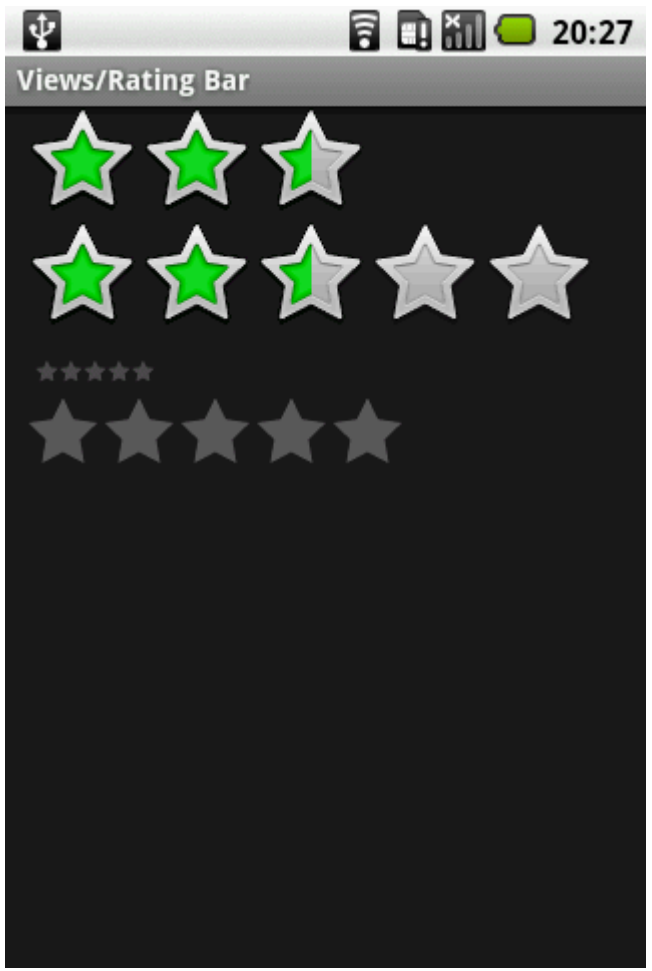
Android 基础UI组件

android.widget.GridView

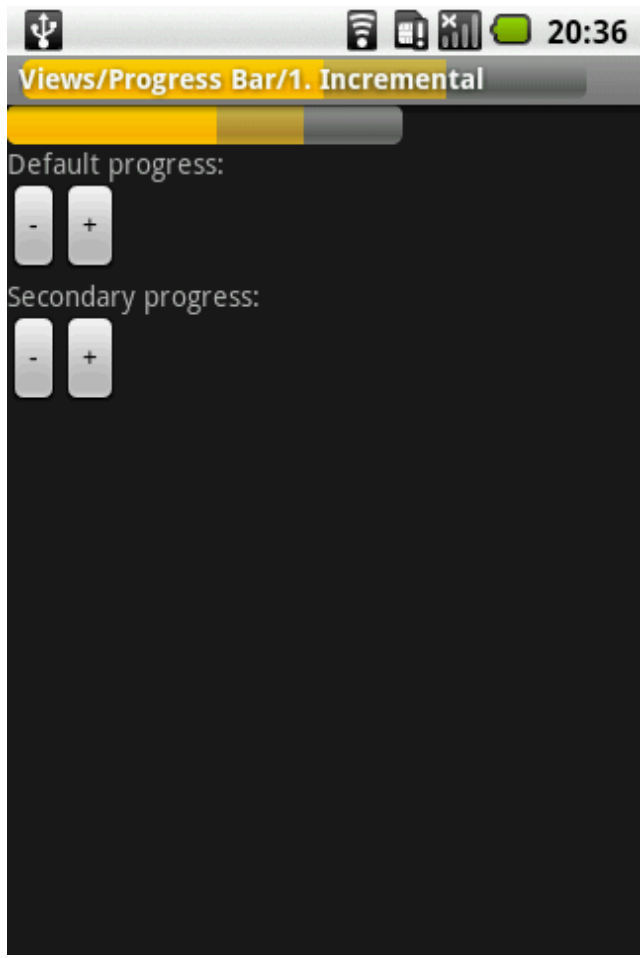
android.widget.Gallery
android.widget.ImageSwitcher



android.widget.RatingBar

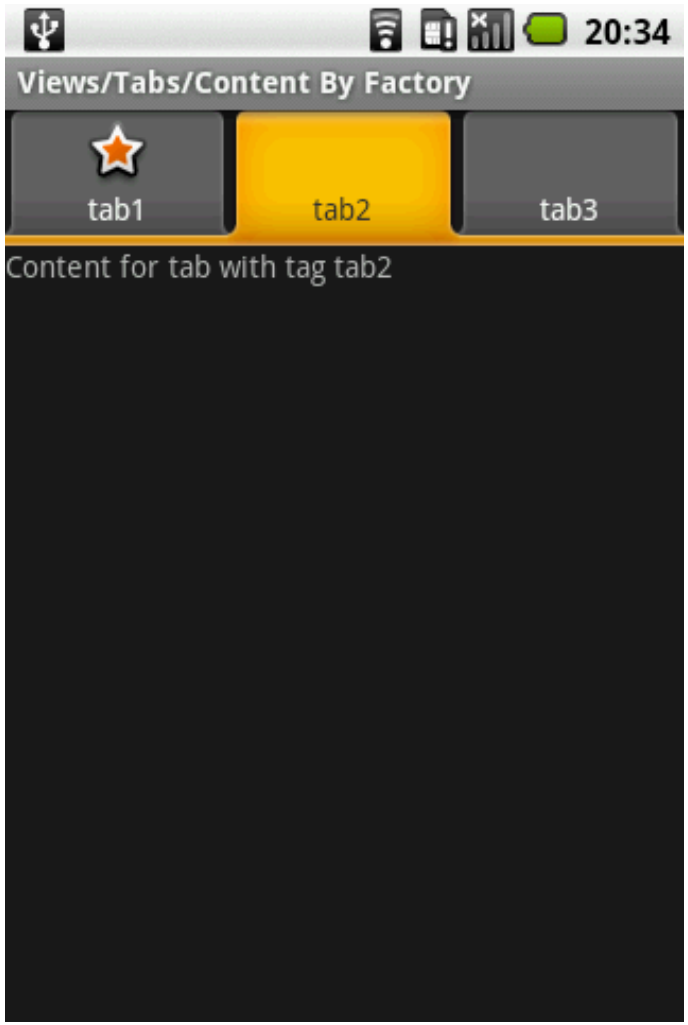


android.widget.ProgressBar

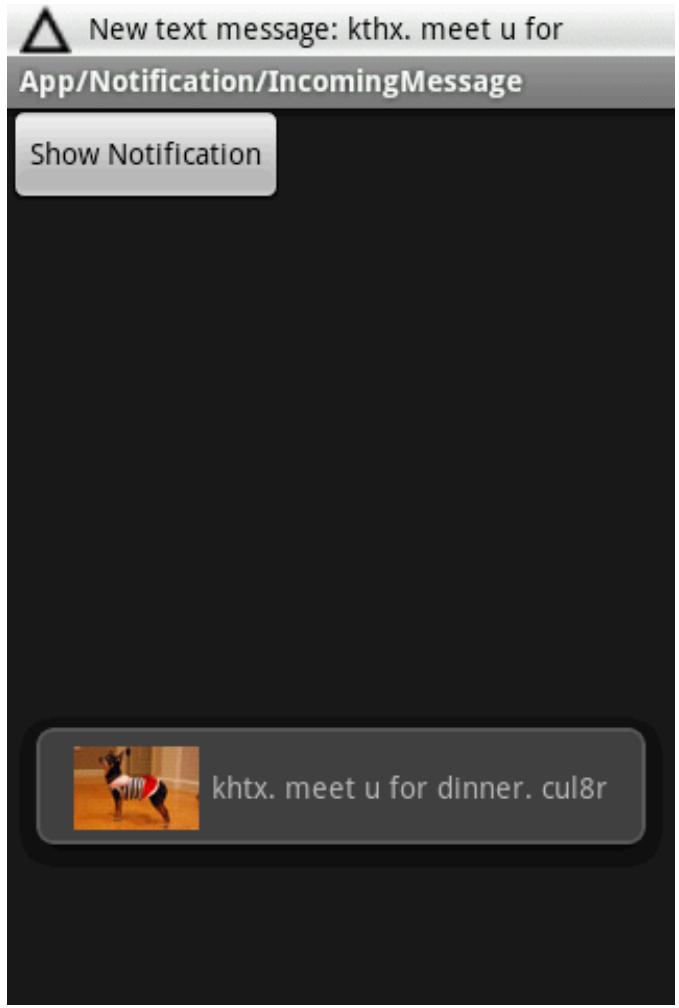


Android 基础UI组件

android.widget.TabWidget



android.widget.Toast



Android 基础UI组件

- menu:如何打造友好的菜单
- ListView:用好列表，做好程序
- Dialog:人机友好互动交流
- Toast和Notification:温馨的提醒

Android的菜单Menu

- menu
  - 负责管理MenuItem
  - 添加一个menuItem
    - add (int groupId, int itemId, int order, CharSequence title)
  - 删除所有的menuItem
    - clear()
- MenuItem
  - 一个菜单的条目
  - 常用的方法:
    - setTitle
    - setIcon
    - getItemId()

1.) 通过代码创建Menu

```
public boolean onCreateOptionsMenu(Menu menu) {
//  分组id，Item的id，顺序，名字
menu.add(0, M_FBACK, 0, "反馈")
.setAlphabeticShortcut('F');
menu.add(0, M_HELP, 1, "帮助")
.setAlphabeticShortcut('H')
.setIcon(android.R.drawable.ic_menu_help);
return true;
}
```

2.) 通过xml创建Menu

- 在Android工程的res/目录下新增一个**menu/**子目录，然后建立option_menu.xml文件
- onCreateOptionsMenu()方法里通过**MenuInflater**类引入定义好的菜单文件

option_menu.xml文件

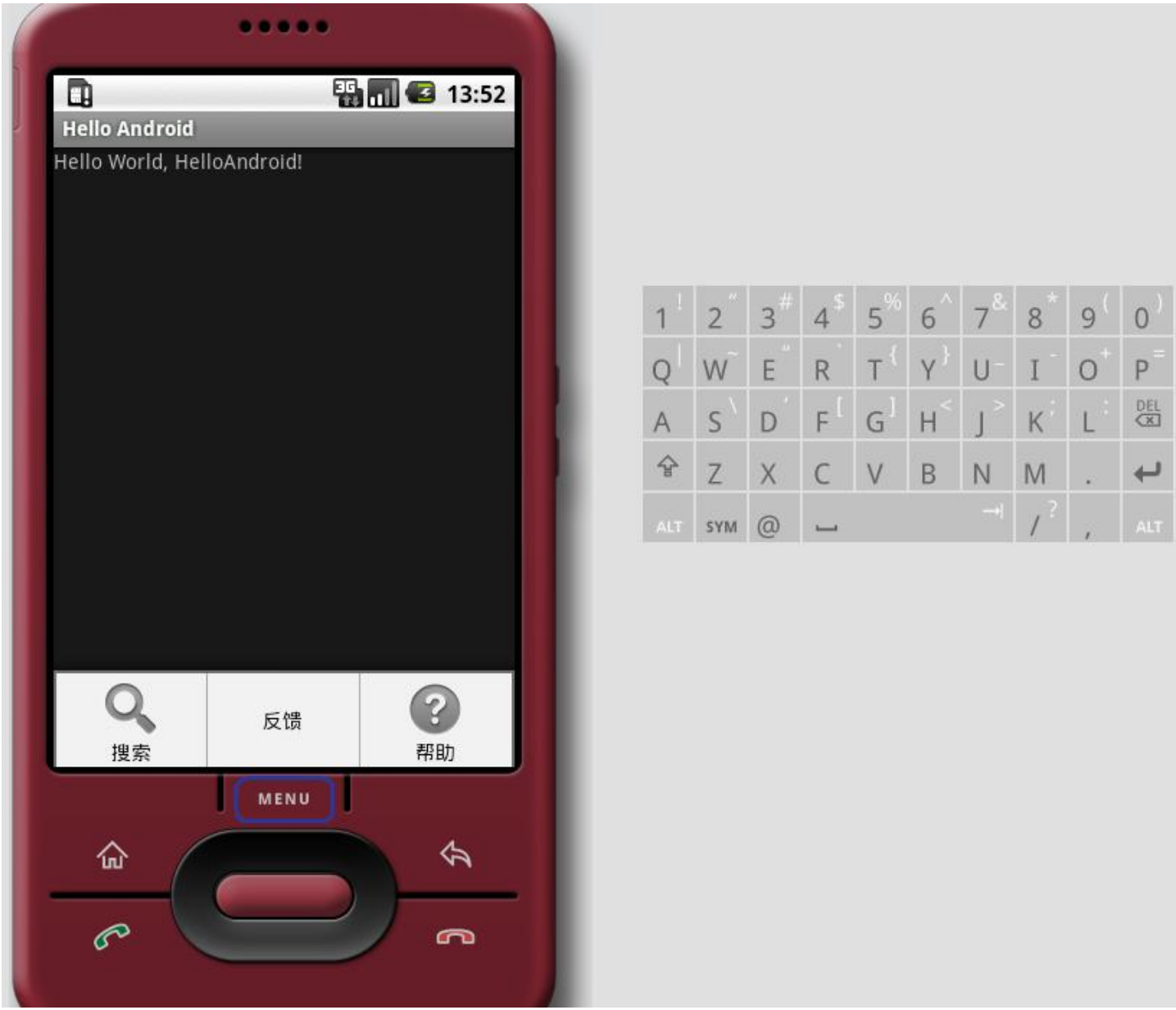
```
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/search_menu" android:title="搜索″
android:icon="@android:drawable/ic_menu_search"></item>
    <item android:id="@+id/open_menu" android:title="反馈"></item>
    <item android:id="@+id/help_menu" android:title="帮助″
android:icon="@android:drawable/ic_menu_help"></item>
</menu>
```

onCreateOptionsMenu()实现menu

```
public boolean onCreateOptionsMenu(Menu menu) {

        MenuInflater inflater = getMenuInflater();
   inflater.inflate(R.menu.option_menu, menu);
return true;
}
```

Android 菜单Menu （6）



3.)处理Menu响应事件

- 我们定义的菜单项的id会保存到R类文件里
- 当我们点击菜单时，框架会捕获被点击的MenuItem组件，传递给相应的处理方法
- onOptionsItemSelected()方法用来处理事件

onOptionsItemSelected()添加响 应事件

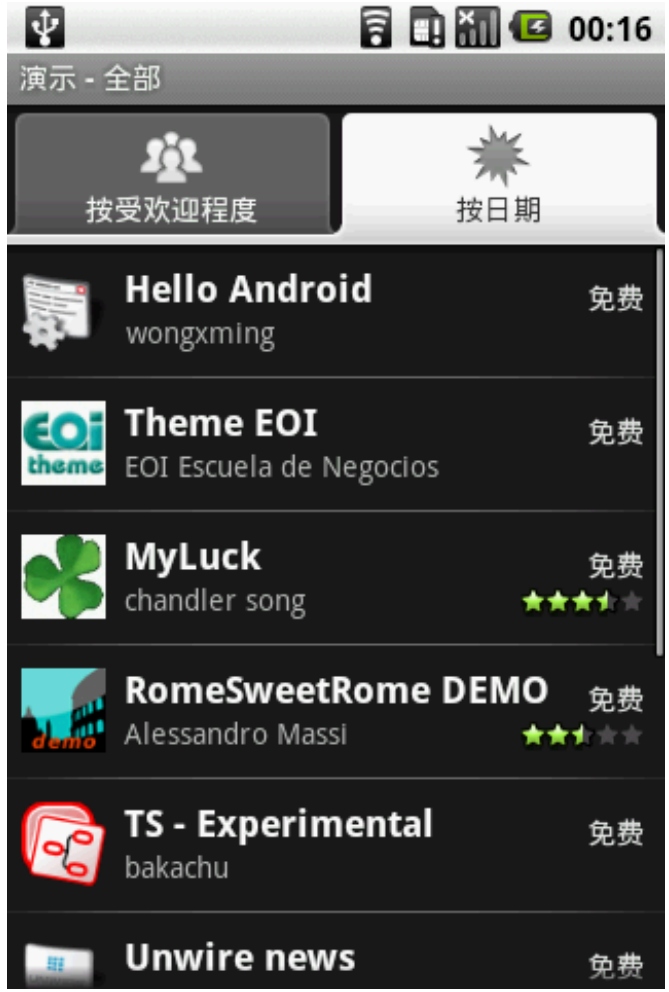
```
public boolean onOptionsItemSelected(MenuItem item) {
      switch ( item.getItemId() ) {
           case R.id.help_menu:
Toast.makeText(this, "你选中的是 ‘帮助’ 菜单！ ",
                                        Toast.LENGTH_SHORT).show();
break;
case R.id.open_menu:
Toast.makeText(this, "你选中的是 ‘反馈’ 菜单！ ",
                                        Toast.LENGTH_SHORT).show();
break;
}
return true;
}
```

Android 菜单Menu (9)

Android的列表ListView

什么是ListView?

ListView可以按设定的规则自动填充并展示一组数据列表



Android 列表ListView

创建一个最简单列表

- xml布局方式

```
<ListView android:id="@+id/myList"
        android:layout_width="fill_parent"
      android:layout_height="fill_parent"/>
```

- 获得引用

```
listView = new ListView(Context context);
listView= (ListView )findViewById(myList);
```

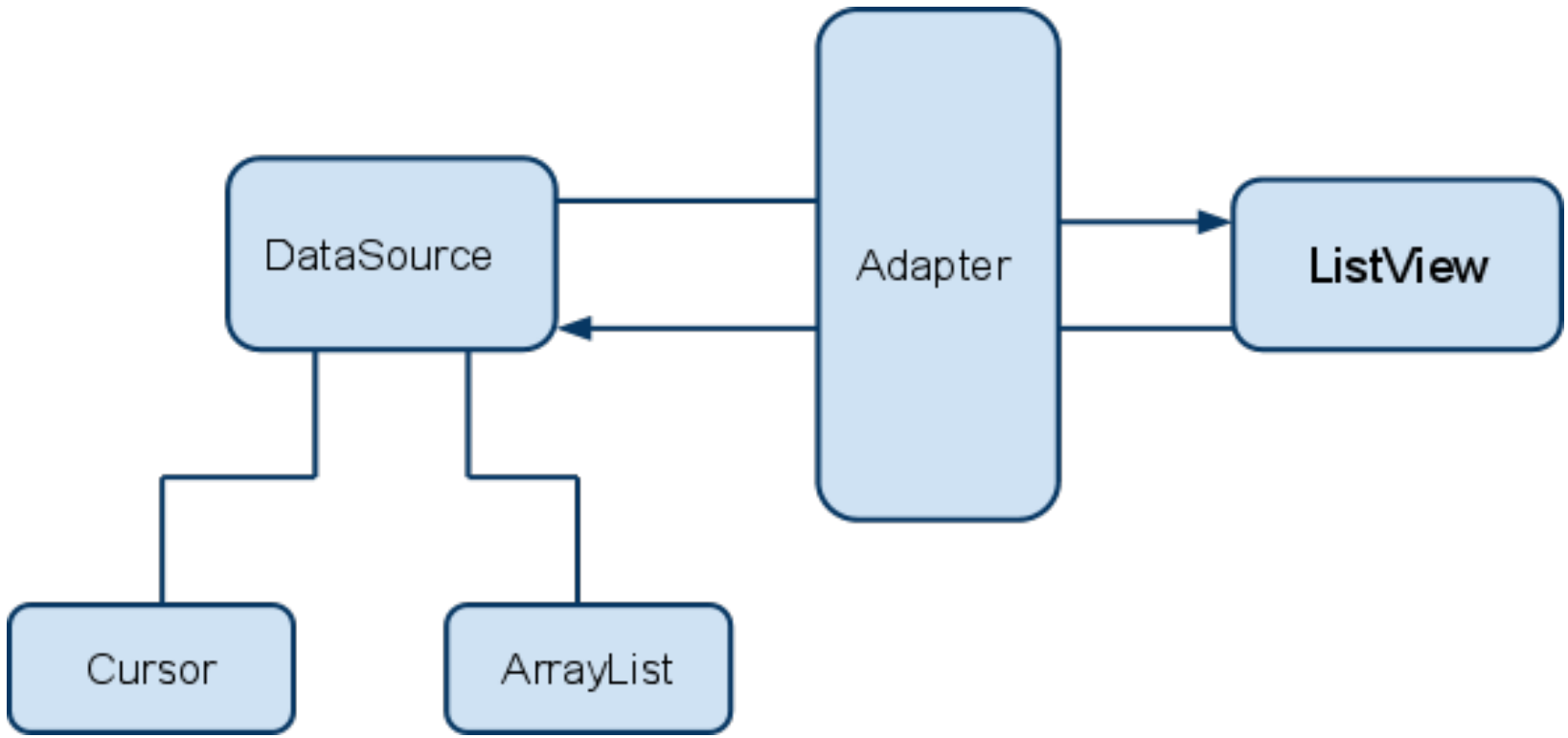
- 设置一个适配器ListAdapter

```
listView.setAdapter(ListAdapter adapter);
```

- 显示ListView

```
setContentView(listView);
```

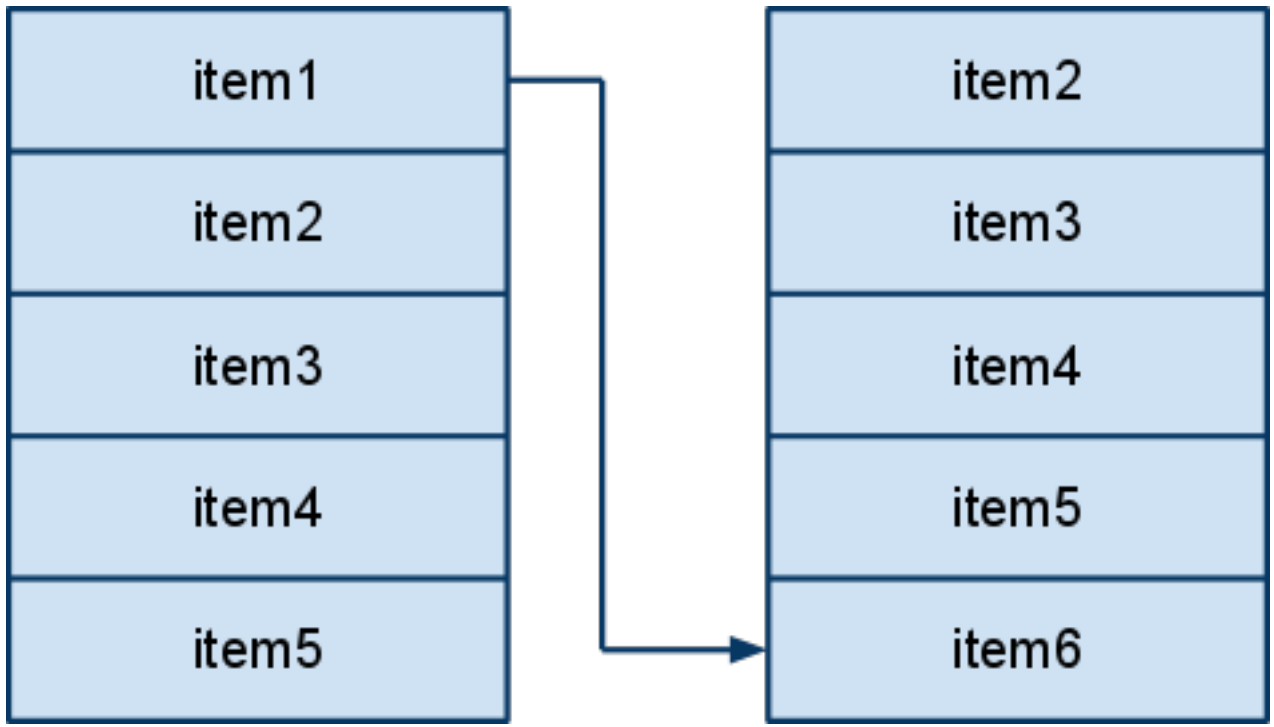
Adapter 是ListView和数据源的中间人



Adpter深入分析

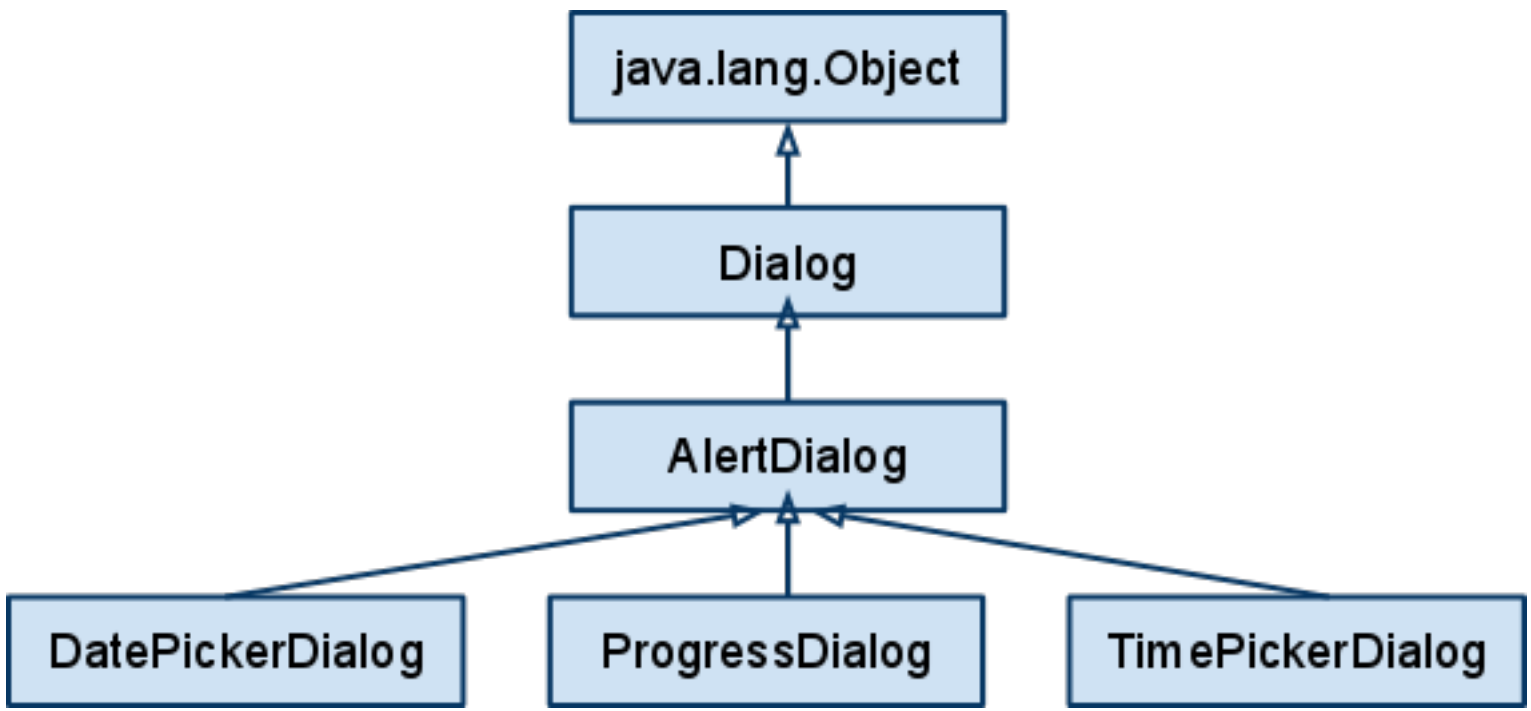
- 当每条数据进入看见区域时
  - Adapter的getView()会被调用
  - 返回展示数据的View视图
- 但滚动屏幕时，会频繁调用上面步骤
- 可以支持上千条数据的显示

Adpter深入分析



Android 的对话框 Dialog

什么是Dialog？



实现一个AlertDialog

生成AlertDialog三步走

- 生成一个AlertDialog的构造者AlertDialog.Builder

```
AlertDialog.Builder builder = new AlertDialog.Builder(context);
```

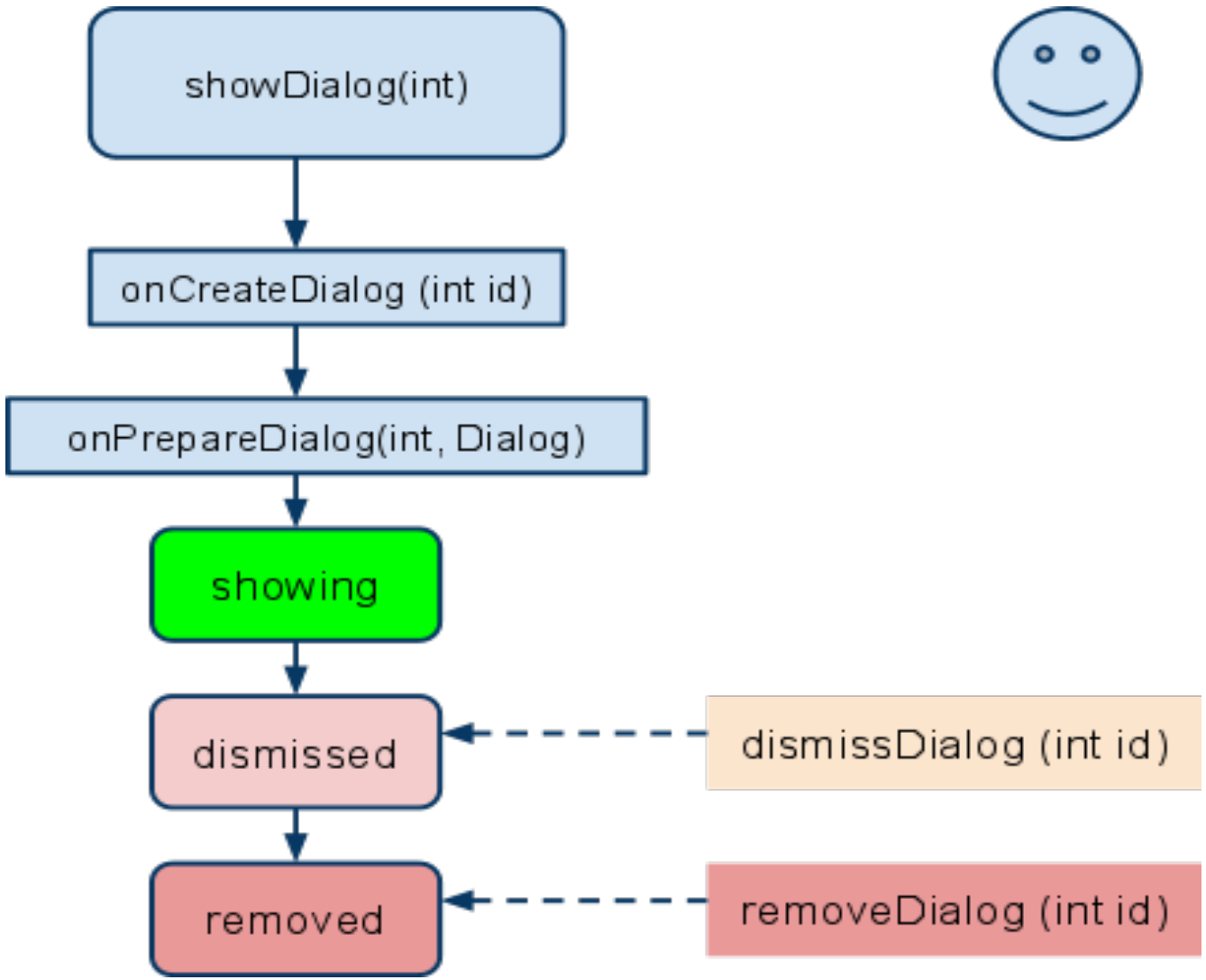
- 设置属性，包括标题、按钮和图标

```
builder.setIcon();
builder.setTitle();
builder.setPositiveButton();
builder.setNegativeButton();
```

- 最后生成AlertDialog

```
builder.create();
```

Dialog的生命周期



创建个性化对话框

- 继承Dialog
- 重写onCreate()方法。
  - setTitle():设置标题
  - setContentView():设置内容

Android 温馨提醒 Toast

Toast是什么？

- Toast是Android提供的
  轻量级的提醒机制
- Toast永远不会获得聚焦
- 不会打断用户当前的操作
- 信息在floating view呈现，
  然后会自动消失

Android 温馨提醒 – Toast

如何创建Toast

1.)简单文字信息

- 通过make()方法创建Toast信息
- 调用show()方法来显示Toast提示信息

2.)复杂Toast信息

- Toast支持通过setView(view)添加view组件

Android 温馨提醒 – Toast (2)

Android 温馨提醒 Notification

- Notification是Android提供的在状态栏的提醒机制
- Notification同样不会打断用户当前的操作
- Notification支持更复杂的点击事件响应
- NotificationManager来管理

创建Notification

创建Notification的四大步骤

1.)得到一个NotificationManager的引用:

```
String ns = Context.NOTIFICATION_SERVICE;
NotificationManager nManager = (NotificationManager) getSystemService(ns);
```

2.)初始化一个Notification

```
int icon = R.drawable.notification_icon;
CharSequence tickerText = "Hello";
long when = System.currentTimeMillis();
Notification notification = new Notification(icon, tickerText, when);
```

如何创建Notification

3.)设置Notification的参数:

```
Context context = getApplicationContext();
CharSequence contentTitle = "My notification";
CharSequence contentText = "Hello World!";
Intent notificationIntent = new Intent(this, MyClass.class);
PendingIntent pIntent = PendingIntent.getActivity(this, 0, notificationIntent,
0);
notification.setLatestEventInfo(context, contentTitle, contentText, pIntent);
```

4.)显示一个Notification

```
private static final int HELLO_ID = 1;
nManager.notify(HELLO_ID, notification);
```