

Android程序设计

列表和布局

2019.4.2

isszym sysu.edu.cn

[官方文档（中文）](#) [官方文档（英文）](#)
[runoob.cnblogs.adroid.widget](#)

总目录

【列表和适配器】

适配器(Adapter)

下拉框(Spinner)

SpinnerStatic

SpinnerCustom(LayoutFlater)

列表框(ListView)

Static

ArrayAdapter

SimpleAdapter

CursorAdapter

网格框(GridView)

动态生成控件

【布局】 45

线性布局(LinearLayout)

相对布局(Relative Layout)

帧布局(FrameLayout)

表布局(TableLayout)

网格布局(GridLayout)

动态创建布局

布局填充器(LayoutInflater)

【附录】 62

附录1、课件所学的控件

* 更详细内容见每个部分目录和附录5

列表和适配器

适配器(Adapter)

下拉框(Spinner)

SpinnerStatic
arrays.xml
entries
spinnerMode
setOnItemSelectedListener()

SpinnerCustom
LayoutInflater

列表框(ListView)

概述

stackFromBottom
transcriptMode
cacheColorHint
divider
scrollbars
fadeScrollbars
setVerticalScrollBarEnabled()

Static(Arrays.xml)

longClickable
setOnItemClickListener()
setOnItemLongClickListener()

ArrayAdapter

setAdapter
simple_list_item_multiple_choice
simple_list_item_single_choice
simple_list_item_1
custom_simple_list_item_1

setChoiceMode
CHOICE_MODE_MULTIPLE
CHOICE_MODE_SINGLE

setOnClickListener

SimpleAdapter

setAdapter(Context,
ArrayList<HashMap>,
item_list,String[]from,int[] to)

CursorAdapter

requiresFadingEdge
fadingEdgeLength
getResource(imageName)
getResourceByReflect(imageName)
SQLiteOpenHelper
setOnItemSelectedListener()

网格框(GridView)

numColumns
columnWidth
stretchMode
horizontalSpacing
verticalSpacing
setOnItemClickListener()

自定义控件(CustomView)

onMeasure()
onLayout()
onDraw()

动态生成控件(DynamicView)

Layout.addView(
view,LayoutParams)
ScrollView

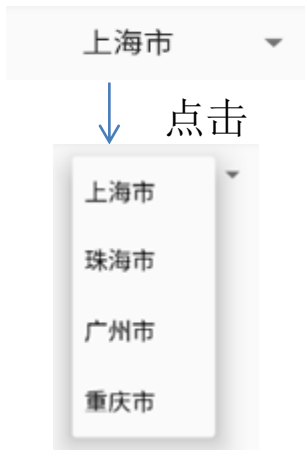
列表类视图和适配器 (ListView and Adapter)

□ 概述

列表类视图可以把相同类型的数据和图像用列表或网格的方式展示出来，主要包括下拉列表Spinner，列表视图ListView和网格列表GridView。它们需要通过适配器Adapter连接到数据集以及结合每行的布局才能显示数据。

适配器主要有 ArrayAdapter， SimpleAdapter， CursorAdapter和自定义适配器。ArrayAdapter用于每行单数据项(单列)的情况(XML数组、Java数组和ArrayList)， SimpleAdapter用于多数据项(多列)的情况(ArrayList<HashMap>)， CursorAdapter用于显示数据库的数据。它们都可以采用自定义方式。

Spinner



ListView

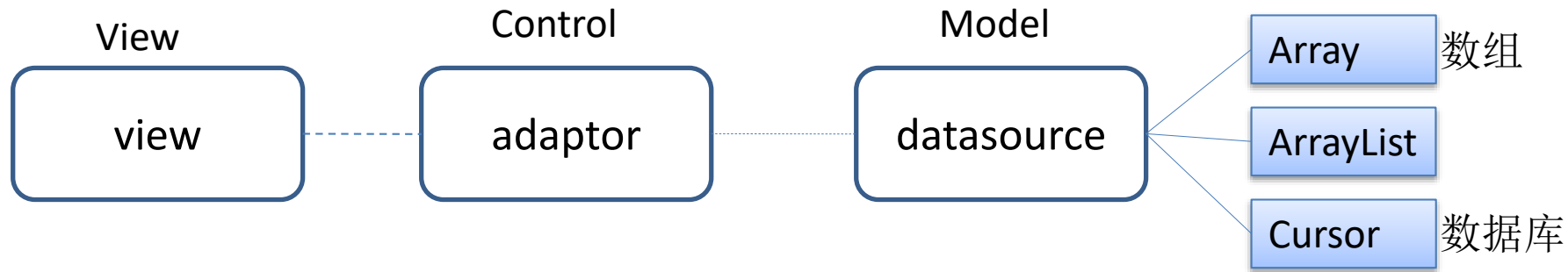


GridView



多列

□ 设计模式：适配器模式



- (1) **Adapter**给出数据来源和显示布局，对于多数据项方式，还要给出数据项和显示项的映射
- (2) 把**View**用它的方法**setAdapter()**绑定到**Adapter**上进行显示

```
String[] cities = {"上海", "广州", "北京"};
ArrayAdapter<String> adapter = new ArrayAdapter<String>(
    MainActivity.this, android.R.layout.simple_list_item_1, cities);
Spinner spinner = (Spinner)findViewById(R.id.spinner);
spinner.setAdapter(adapter);
```

spinner

上海 ▼

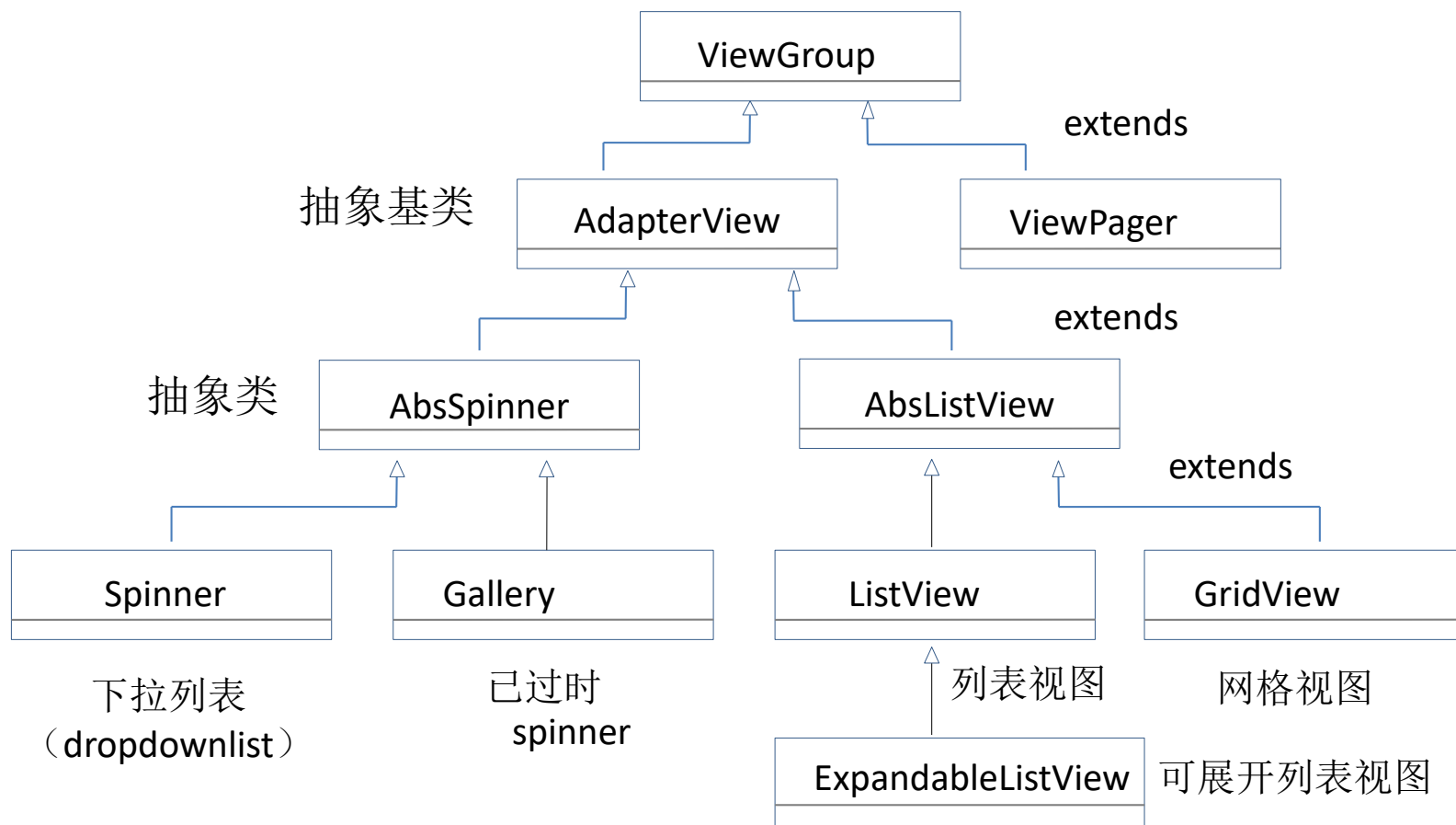
上海
广州
北京

android.R.layout.simple_list_item_1.xml

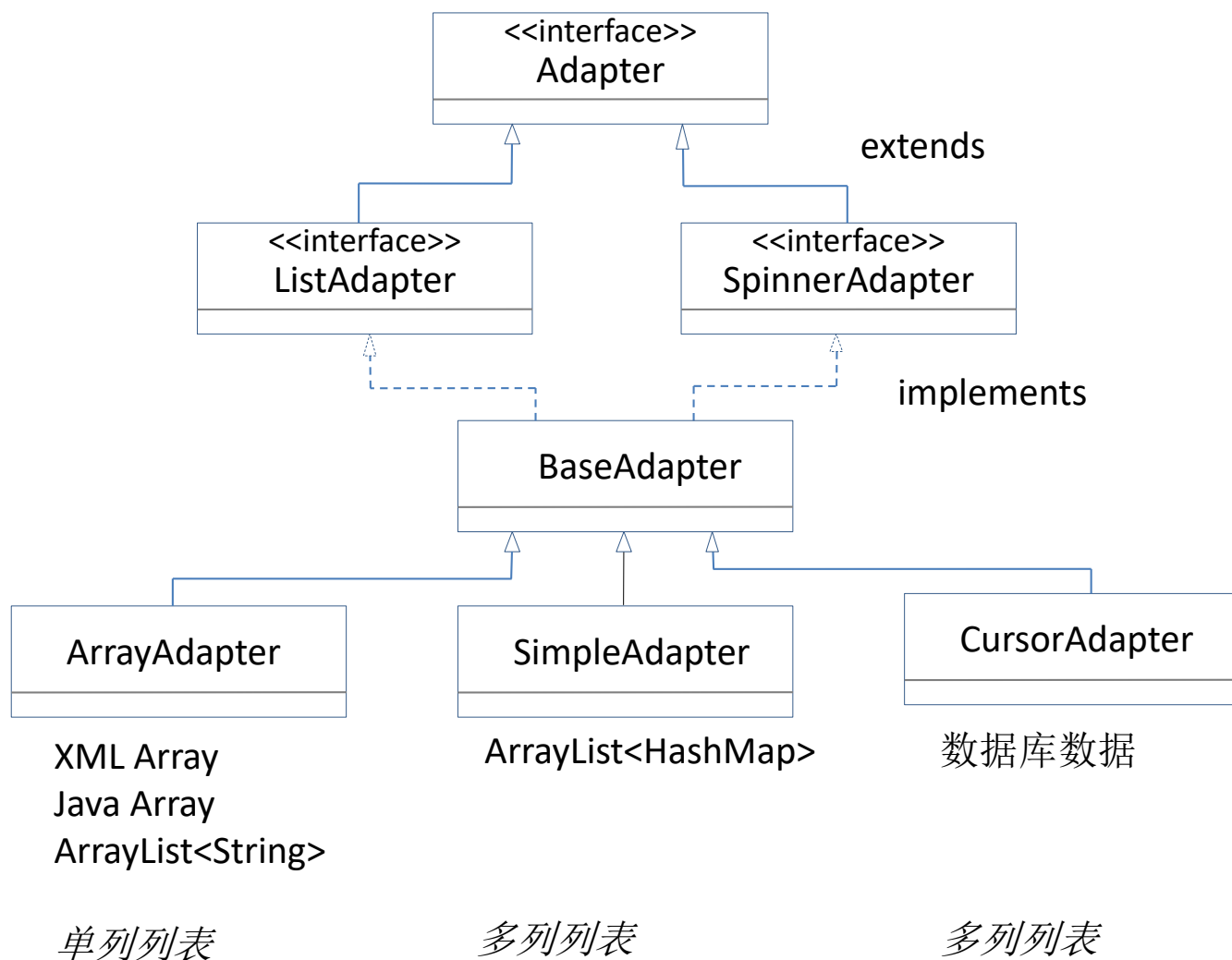
```
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@android:id/text1"
    android:gravity="center_vertical"
    ...
/>
```

系统内置布局

□ 列表类视图



□ 列表类适配器



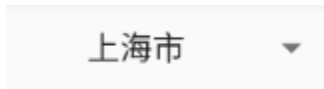
Spinner、ListView、GridView都可以用于这三种适配器，也可以用于自定义适配器。自定义适配器继承这些适配器，并可以进行改变。BaseAdapter可以对表项进行最大限度的定制。

下拉框 (Spinner)

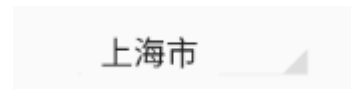
[参考](#)

□ 概述

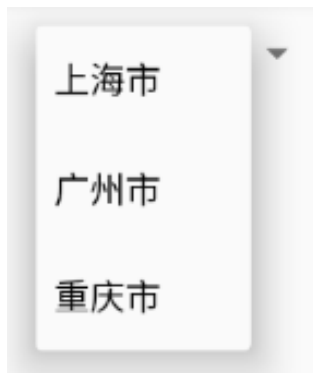
- 下拉框Spinner采用下拉框(dropdown)的方式在多个项目中选择一个项目，它可以使用ArrayAdapter(数组、ArrayList<String>)、SimpleAdapter(ArrayList<HashMap>)和SimpleCursorAdapter(ArrayList<HashMap>)，它的显示方式取决于属性style和spinnerMode。



`style="spinnerStyle"` (默认)



`style="@style/Platform.Widget.AppCompat.Spinner"`



`spinnerMode="dropdown"`



`spinnerMode="dialog"`

□ 直接使用XML数组

MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Spinner spinner=(Spinner)findViewById(R.id.spinner1);
        spinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?> parent, View view,
                                     int position, long id) {
                Toast toast=Toast.makeText(MainActivity.this, "选择第"+(position+1)+"项",
                                           Toast.LENGTH_SHORT);
                toast.setGravity(Gravity.TOP|Gravity.CENTER_HORIZONTAL, 0, 360);
                toast.show();
            }
            @Override
            public void onNothingSelected(AdapterView<?> parent) { }
        });
    }
}
```

工程名: SpinnerStatic

事件见后面的ListView

Toast显示定位 {

- setGravity(int gravity, int xOffset, int yOffset)
- setMargin(float horiMargin, float vertMargin)

深圳市

选择第3项

- 下面是spinner直接使用资源数组的例子。

activity_main.xml

```
<Spinner
    android:id="@+id/spinner1"
    android:entries="@array/city"
    android:spinnerMode="dropdown"
    android:style="spinnerStyle" />
```

res/values/arrays.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name="city">
        <item>上海市</item>
        <item>厦门市</item>
        <item>深圳市</item>
        <item>北京市</item>
        <item>珠海市</item>
        <item>广州市</item>
        <item>重庆市</item>
    </string-array>
</resources>
```



□ 自定义ArrayAdapter

- 定义MyArrayAdapter为ArrayAdapter的子类

(1) 通过构造器带入Activity实例（上下文）和字符串数组。

```
public MyArrayAdapter(Context context, String[] stringArray) { ... }
```

(2) 通过调用回调函数getDropDownView()返回每个下拉项的视图。

```
public View getDropDownView(int position, View convertView, ViewGroup parent)
```

其中， position为第一个下拉项（从0开始）， convertView用于返回的下拉项， parent为当前Spinner。带入convertView主要目的是减少布局填充(inflate)的次数。有关填充器见本课件最后面的一节

(3) 通过调用回调函数getView()返回选中项目的视图。参数的含义与上面类似。

```
public View getView(int position, View convertView, ViewGroup parent)
```

- 通过MyArrayAdapter定义ArrayAdapter， 其中cities可以为数组和ArrayList。

```
Spinner spinner=(Spinner) findViewById(R.id. spinner);  
ArrayAdapter<String> arrayAdapter =  
    new MyArrayAdapter(MainActivity. this, cities);  
spinner.setAdapter(arrayAdapter);
```

```

public class MainActivity extends AppCompatActivity {
    private String [] cities;
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Spinner spinner=(Spinner) findViewById(R.id.spinner);
        cities=getResources().getStringArray(R.array.city);
        ArrayAdapter<String> arrayAdapter =
            new MyArrayAdapter(MainActivity.this,cities);
        spinner.setAdapter(arrayAdapter);
        spinner.setOnItemClickListener(new ItemSelListener());
    }
    private class ItemSelListener
        implements AdapterView.OnItemClickListener{
        @Override
        public void onItemClick(AdapterView<?> parent, View view,
            int position, long id) {
            Toast.makeText(MainActivity.this,"选中了:"+ cities[position],
                Toast.LENGTH_SHORT).show();
        }
        @Override
        public void onNothingSelected(AdapterView<?> parent) {}
    }
}

```

• 设置下拉方式(默认): `arrayAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);`

深圳

广州

上海

北京

深圳

天津

重庆

- 自定义下拉框

```
public class MainActivity extends AppCompatActivity {  
    private String [] cities;  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        Spinner spinner=(Spinner) findViewById(R.id.spinner);  
        cities=getResources().getStringArray(R.array.city);  
        ArrayAdapter<String> arrayAdapter =  
            new MyArrayAdapter(MainActivity.this,cities);  
        spinner.setAdapter(arrayAdapter);  
        spinner.setOnItemSelectedListener(new ItemSelListener());  
    }  
    private class ItemSelListener implements AdapterView.OnItemSelectedListener{  
        @Override  
        public void onItemSelected(AdapterView<?> parent, View view,  
                                    int position, long id) {  
            Toast.makeText(MainActivity.this,"选中了:" + cities[position],  
                           Toast.LENGTH_SHORT).show();  
        }  
        @Override  
        public void onNothingSelected(AdapterView<?> parent) {}  
    }  
}
```

项目: SpinnerCustom



•设置下拉方式(默认): `arrayAdapter.setDropDownViewResource(
 android.R.layout.simple_spinner_dropdown_item);`

MyArrayAdapter.java

```
public class MyArrayAdapter extends ArrayAdapter<String> {
    private Context context;
    private String [] stringArray;
    public MyArrayAdapter(Context context, String[] stringArray) {
        super(context, android.R.layout.simple_spinner_item, stringArray);
        this.context = context;
        this.stringArray = stringArray;
    }
    @Override // Spinner下拉时每一行显示的内容由convertView返回
    public View getDropDownView(int position, View convertView, ViewGroup parent) {
        if (convertView == null) { //减少布局填充 (inflate)的次数
            LayoutInflater inflater = LayoutInflater.from(context);
            convertView = inflater.inflate(
                android.R.layout.simple_spinner_dropdown_item, parent, false);
        }
        TextView tv = (TextView) convertView.findViewById(android.R.id.text1);
        tv.setText(stringArray[position]);
        tv.setTextSize(18f); // 浮点数, 默认sp
        tv.setTextColor(Color.BLUE);
        return convertView;
    }
}
```

inflater为当前Activity的填充器，用于把XML布局转换为视图。具体的用法见后面章节。

```

@Override //Spinner选中一行后的视图由convertView返回
public View getView(int position, View convertView, ViewGroup parent) {
    if (convertView == null) { //减少填充的次数
        LayoutInflater inflater = LayoutInflater.from(context);
        convertView = inflater.inflate(android.R.layout.simple_spinner_item,
            parent, false);
    };
    TextView tv = (TextView) convertView.findViewById(android.R.id.text1);
    tv.setText(stringArray[position]);
    tv.setTextSize(18f);
    tv.setTextColor(Color.RED);
    return convertView;
}
}

```

activity_main.xml

```

<Spinner
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:id="@+id/spinner"/>

```

android.R.layout. *simple_spinner_item.xml*

```
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@android:id/text1"
    style="?android:attr/spinnerItemStyle"
    android:singleLine="true"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ellipsize="marquee"
    android:textAlignment="inherit"/>
```

系统内置布局文件

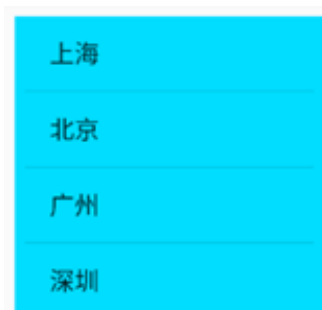
android.R.layout. *simple_spinner_dropdown_item.xml*

```
<?xml version="1.0" encoding="utf-8"?>
<CheckedTextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@android:id/text1"
    style="?android:attr/spinnerDropDownItemStyle"
    android:singleLine="true"
    android:layout_width="match_parent"
    android:layout_height="?android:attr/dropDownListPreferredItemHeight"
    android:ellipsize="marquee"/>
```


列表视图 (ListView)

ListView可以用于列表展示、单项选择和多项选择，主要适配器包括ArrayList、SimpleAdapter和SimpleCursorAdapter，数据来源可以是数组、ArrayList<String>和ArrayList(<HashMap>)。

每行单项的列表使用ArrayList和ArrayAdapter，每行多项的列表需要使用SimpleAdapter和SimpleCursorAdapter以及ArrayList(<HashMap>)。它们都可以使用自定义适配器。



普通列表显示



单选列表显示



多选列表显示



□ ArrayList（XML数组和资源ListView）

- 本案例主要利用资源数组和资源ListView实现普通列表。

```
<ListView
    android:entries="@array/city"
    ...
/>
```

```
<resources>
    <string-array name="city">
        <item>广州</item>
        ...
    </string-array>
</resources>
```

- ListView的事件主要有ItemClick（普通点击）和ItemLongClick（长按点击）。

onItemClick(AdapterView<?> parent, View view, **int** position, **long** id)

onItemLongClick(AdapterView<?> parent, View view, **int** position, **long** id)

- ✓ view是被点击item(行)的句柄，用view.findViewById()方法可以获取所点击item中的控件。
- ✓ position是被点击item在适配器里第几行（从0开始）。
- ✓ id为所点击item位于第几行。大部分时候其值与position一样。
- ✓ parent指向listview适配器的一个指针。

```
ListView listView = (ListView) parent;
```

```
ListAdapter listAdapter = listView.getAdapter();
```

```
TextView tv = (TextView) listAdapter.getItem(position);
```

项目：ListViewStatic

```
public class MainActivity extends AppCompatActivity {  
    private ListView lv;  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        lv = (ListView) findViewById(R.id.listView);  
        lv.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
            public void onItemClick(AdapterView<?> parent, View view,  
                                    int position, long id) {  
                Toast.makeText(MainActivity.this,  
                    "第" + (position + 1) + "项被单击按下", Toast.LENGTH_SHORT)  
                    .show();  
            }  
        });  
        lv.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() {  
            public boolean onItemLongClick(AdapterView<?> parent, View view,  
                                            int position, long id) {  
                Toast.makeText(MainActivity.this,  
                    "第" + (position + 1) + "项被长时间按下",  
                    Toast.LENGTH_LONG).show();  
                return true;  
            }  
        });  
    }  
}
```

点击 →



第3项被单击按下

activity_main.xml

```
<ListView
    android:id="@+id/listView"
    android:layout_width="120dp"
    android:layout_height="wrap_content"
    android:entries="@array/city"
    android:background="@android:color/holo_blue_bright"
    android:longClickable="true"
    android:textColor="#F00" /> —— 没有效果
```

arrays.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name="city">
        <item>广州</item>
        <item>上海</item>
        <item>北京</item>
        <item>深圳</item>
        <item>天津</item>
        <item>重庆</item>
    </string-array>
</resources>
```

□ ArrayAdapter (Array或ArrayList, 普通、单选、多选)

- 把数组和List用于普通列表显示、单选列表显示和多选列表显示。

```
String[] cities = {"上海", "北京", "广州", "深圳"};
adapter = new ArrayAdapter<String>(this,
    android.R.layout.simple_list_item_multiple_choice, cities);
lv = (ListView) findViewById(R.id.listView);
lv.setAdapter(adapter);
lv.setChoiceMode(ListView.CHOICE_MODE_MULTIPLE);
```

- 显示方式 `new ArrayAdapter<String>()`

普通列表显示: `android.R.layout.simple_list_item_1`

单选列表显示: `android.R.layout.simple_list_item_single_choice`

多选列表显示: `android.R.layout.simple_list_item_multiple_choice`

自定义列表显示: `custom_simple_list_item_1` (用于改变显示)

- 选择方式 `setChoiceMode()`

不可选择: `CHOICE_MODE_NONE` (默认)

单项选择: `CHOICE_MODE_SINGLE`

多选选择: `CHOICE_MODE_MULTIPLE`

多选显示(不能改变): `CHOICE_MODE_MULTIPLE_MODAL`

各种显示方式都可以设置选择方式, 而且都有效!

- 其它生成list的方法:

```
List<String> cities = Arrays.asList("上海", "北京", "广州", "深圳");
```

```
String[] cities = getResources().getStringArray(R.array.city);
```

上海
北京
广州
深圳

普通列表显示

上海	<input type="radio"/>
北京	<input type="radio"/>
广州	<input checked="" type="radio"/>
深圳	<input type="radio"/>

单选列表显示

上海	<input checked="" type="checkbox"/>
北京	<input checked="" type="checkbox"/>
广州	<input type="checkbox"/>
深圳	<input checked="" type="checkbox"/>

多选列表显示

项目：ListViewArray

```
public class MainActivity extends AppCompatActivity {
    private ListView lv;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ArrayList list = new ArrayList<String>(); // 生成动态list
        list.add("广州"); list.add("上海"); list.add("北京");
        list.add("深圳"); list.add("天津"); list.add("重庆");
        ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
            android.R.layout.simple_list_item_multiple_choice, list);
        lv = (ListView) findViewById(R.id.listView);
        lv.setAdapter(adapter);
        lv.setChoiceMode(ListView.CHOICE_MODE_MULTIPLE);
        lv.setPadding(20, 0, 20, 0);
        Button btn = (Button) findViewById(R.id.button);
        btn.setOnClickListener(new Button.OnClickListener() {
            public void onClick(View view) {
                SparseBooleanArray arr = lv.getCheckedItemPositions();
                String selected = "";
                for (int i = 0; i < arr.size(); i++) {
                    if(arr.valueAt(i)) //int pos = arr.keyAt(i);
                        selected = selected + (selected.isEmpty() ? "" : ", ") + arr.keyAt(i);
                }
                Toast.makeText(MainActivity.this, "第" + selected + "项被选择. 共"
                    + lv.getCount() + "项", Toast.LENGTH_LONG).show();
            }
        });
    }
}
```

上海	<input checked="" type="checkbox"/>
北京	<input checked="" type="checkbox"/>
广州	<input type="checkbox"/>
深圳	<input checked="" type="checkbox"/>

BUTTON

↓ 点击

第0,1,3项被选择. 共6项

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout ...>
    <ListView
        android:id="@+id/listView"
        ...
        android:background="@android:color/holo_blue_bright" />
    <Button
        android:id="@+id/button"
        ...
        android:text="Button" />
</RelativeLayout>
```

android.R.layout.simple_list_item_multiple_choice.xml

```
<?xml version="1.0" encoding="utf-8"?>
<CheckedTextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@android:id/text1"
    android:layout_width="match_parent"
    android:layout_height="?android:attr/listPreferredItemHeightSmall"
    android:textAppearance="?android:attr/textAppearanceListItemSmall"
    android:gravity="center_vertical"
    android:checkMark="?android:attr/listChoiceIndicatorMultiple"
    android:paddingStart="?android:attr/listPreferredItemPaddingStart"
    android:paddingEnd="?android:attr/listPreferredItemPaddingEnd" />
```

android.R.layout.simple_list_item_single_choice.xml

```
.....
    android:checkMark="?android:attr/listChoiceIndicatorSingle"
```

android.R.layout.simple_list_item_1

```
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@android:id/text1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceListItemSmall"
    android:gravity="center_vertical"
    android:paddingStart="?android:attr/listPreferredItemPaddingStart"
    android:paddingEnd="?android:attr/listPreferredItemPaddingEnd"
    android:minHeight="?android:attr/listPreferredItemHeightSmall" />
```

custom_simple_list_item_1.xml

```
<?xml version="1.0" encoding="utf-8"?>
<CheckedTextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@android:id/text1"
    android:layout_width="match_parent"
    android:layout_height="?android:attr/listPreferredItemHeightSmall"
    android:textAppearance="?android:attr/textAppearanceListItemSmall"
    android:textColor="#FF0000"
    android:textSize="40sp"
    android:gravity="center_vertical"
    android:checkMark="?android:attr/listChoiceIndicatorMultiple"
    android:paddingLeft="?android:attr/listPreferredItemPaddingLeft"
    android:paddingRight="?android:attr/listPreferredItemPaddingRight" />
```

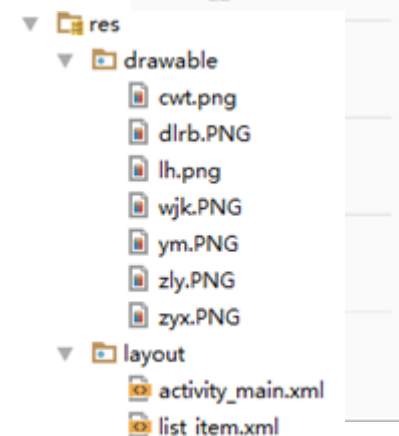

❑ SimpleAdapter (ArrayList<HashMap>)

- SimpleAdapter 结合 ArrayList<HashMap> 可以在列表每行显示多个项目。

```
int[] images = { R.drawable.cwt, R.drawable.dlrb,...};
String[] names = { "陈伟霆", "迪丽热巴", ...};
int[] ages = { 18, 21, ...};
list = new ArrayList<Map<String, Object>>();
for (int i = 0; i < names.length; i++) {
    Map<String, Object> map = new HashMap<String, Object>();
    map.put("icon", images[i]);
    map.put("name", names[i]);
    map.put("age", ages[i]);
    list.add(map);
}
SimpleAdapter adapter = new SimpleAdapter(this, list,
    R.layout.list_item,
    new String[] { "icon", "name", "age" },
    new int[] { R.id.icon, R.id.name, R.id.age });
listView.setAdapter(adapter);
```

HashMap(列)

ArrayList(行)



layout/list_item.xml

```
<RelativeLayout ...>
    <ImageView android:id="@+id/icon" .../>
    <TextView android:id="@+id/name" .../>
    <TextView android:id="@+id/age" .../>
</RelativeLayout>
```

```

public class MainActivity extends AppCompatActivity {
    private ListView listView;
    ArrayList<Map<String, Object>> list;
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        listView = (ListView) findViewById(R.id.listView);
        int[] images = { R.drawable.cwt, R.drawable.dlr, R.drawable.lh,
            R.drawable.wjk, R.drawable.ym, R.drawable.zly, R.drawable.zyx };
        String[] names = { "陈伟霆", "迪丽热巴", "鹿晗",
            "王俊凯", "杨幂", "赵丽颖", "张艺兴" };
        int[] ages = { 18, 21, 22, 32, 31, 18, 20, 25 };
        list = new ArrayList<Map<String, Object>>();
        for (int i = 0; i < names.length; i++) {
            Map<String, Object> map = new HashMap<String, Object>();
            map.put("icon", images[i]);    map.put("name", names[i]);
            map.put("age", ages[i]);
            list.add(map);
        }
        /** 参数: context(上下文对象) datasource(数据源) itemlayout(每个Item的布局页面)
         *   from String[] 数据源中key的数组, to int[] 布局页面中id的数组 */
        SimpleAdapter adapter = new SimpleAdapter(this, list,
            R.layout.list_item, new String[] { "icon", "name", "age" },
            new int[] { R.id.icon, R.id.name, R.id.age });
        listView.setAdapter(adapter);
    }
}

```

activity_main.xml

```
<ListView
    android:id="@+id/listView"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:background="@android:color/holo_blue_bright" />
```

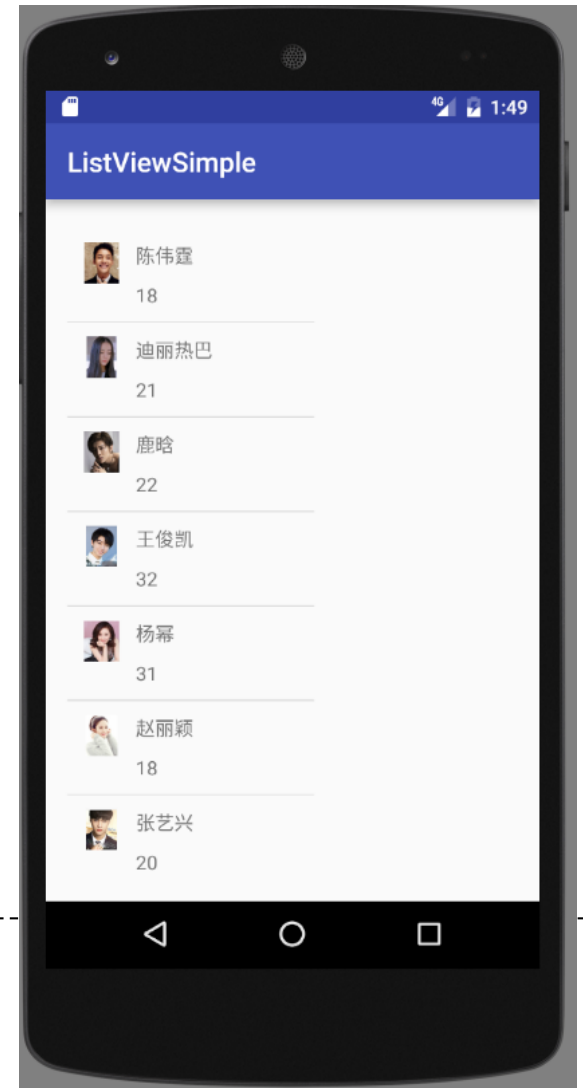
list_item.xml

```
?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="10dp" >
    <ImageView
        android:id="@+id/icon"
        android:layout_width="30dp"
        android:layout_height="30dp"
        android:src="@drawable/cwt"
        android:layout_marginRight="10dp"/>
```

```
<TextView
    android:id="@+id/name"
    android:text="姓名"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_toRightOf="@id/icon" />
```

```
<TextView
    android:id="@+id/age"
    android:text="年龄"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/name"
    android:layout_marginTop="10dp"
    android:layout_alignLeft="@id/name" />
```

```
</RelativeLayout>
```



- 可以增加事件:

```
listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        Toast.makeText(MainActivity.this, "Short Click: " +
            list.get(position).get("name").toString(), Toast.LENGTH_SHORT).show();
    }
});

listView.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() {
    @Override
    public boolean onItemLongClick(AdapterView<?> parent, View view,
                                    int position, long id) {
        Toast.makeText(MainActivity.this, "Long Click: " +
            list.get(position).get("name").toString(), Toast.LENGTH_SHORT).show();
        return true; //true: 只执行长按事件(ShortClick事件失效)
    }
});
```



* id与position取值相同

□ SimpleCursorAdapter

- Cursor是指向从数据库中取出的记录集的指针。SimpleCursorAdapter的使用与SimpleAdapter相似，只是把ArrayList(<HashMap>)替换成了Cursor。

```
Cursor cursor = db.rawQuery("select _id,photo,num,name from stu", null);
SimpleCursorAdapter adapter = new SimpleCursorAdapter(MainActivity.this,
    R.layout.list_item, cursor,
    new String[] {"num", "name"},
    new int[] {R.id.num, R.id.name}, 0);
listView.setAdapter(adapter);
```

- 由于数据库中保存的是图片名称，本例通过自定义SimpleCursorAdapter显示图片。

```
MySimpleCursorAdapter adapter = new MySimpleCursorAdapter(MainActivity.this,
    R.layout.list_item, cursor,
    new String[] {"photo", "num", "name"},
    new int[] {R.id.photo, R.id.num, R.id.name}, 0);
listView.setAdapter(adapter);
```

SimpleCursorAdapter的回调函数newView()用于生成每行的视图。

SimpleCursorAdapter的回调函数bindView()用于显示每行视图。因此，先在newView()中填充(inflate)布局list_item作为列表每一行的视图，然后再利用bindView()修改每一行的显示内容。其中的要显示的图片需要通过资源名称来获得资源Id。

MainActivity.java

工程名: ListViewCursor

```
public class MainActivity extends AppCompatActivity {
    ListView listView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        DBOpenHelper dbOpenHelper = new DBOpenHelper(this, "dbStu.db3", null, 1);
        SQLiteDatabase db = dbOpenHelper.getWritableDatabase(); //insert/upatde/delete
        //db.execSQL("DROP TABLE stu");
        String CREATE_TABLE = "CREATE TABLE IF NOT EXISTS stu(_id INTEGER PRIMARY KEY
                                autoincrement,photo varchar(24), num INTEGER, name varchar(24))";
        db.execSQL(CREATE_TABLE);
        Cursor cursor = db.rawQuery("select _id,photo,num,name from stu", null);
        if (cursor != null) {
            MySimpleCursorAdapter adapter=new MySimpleCursorAdapter(MainActivity.this,
                R.layout.list_item, cursor,new String[]{"photo","num","name"},
                new int[]{R.id.photo,R.id.num, R.id.name}, 0);
            listView = (ListView) findViewById(R.id.listView);
            listView.setAdapter(adapter);
        }
        db.close();
    }
}
```

```

Button btn=(Button)findViewById(R.id.button);
btn.setOnClickListener(new Button.OnClickListener() {
    @Override
    public void onClick(View v) {
        AddRows();
    }
});
listView.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
    @Override
    public void onItemSelected(AdapterView<?> parent,View v,int pos,long id) {
        // TODO Auto-generated method stub
        Cursor cursor = (Cursor) listView.getSelectedItem();
        String name = cursor.getString(1);
        String num = cursor.getString(2);
        System.out.println(num+" "+name);
    }

    @Override
    public void onNothingSelected(AdapterView<?> arg0) {
        // TODO Auto-generated method stub
    }
});
}

```



```

void AddRows() {
    DBOpenHelper dbOpenHelper = new DBOpenHelper(this, "dbStu.db3", null, 1);
    SQLiteDatabase db = dbOpenHelper.getWritableDatabase();//ins/upatde/del
    //db.execSQL("DROP TABLE stu");
    String CREATE_TABLE = "CREATE TABLE IF NOT EXISTS stu(_id INTEGER PRIMARY KEY
        autoincrement,photo varchar(24), num INTEGER, name varchar(24))";
    db.execSQL(CREATE_TABLE);
    String INSERT_DATA = "";
    INSERT_DATA="INSERT INTO stu(photo,num, name)values(' cwt',150010121,' 陈伟霆')";
    db.execSQL(INSERT_DATA);
    INSERT_DATA="INSERT INTO stu(photo,num, name)values(' dlrb',150010122,' 迪丽热巴')";
    db.execSQL(INSERT_DATA);
    INSERT_DATA="INSERT INTO stu(photo,num, name) values(' zyx',150010123, ' 张艺兴')";
    db.execSQL(INSERT_DATA);
    INSERT_DATA="INSERT INTO stu(photo,num, name)values(' lh',150010124, ' 鹿晗')";
    db.execSQL(INSERT_DATA);
    //db.execSQL("DELETE FROM stu;");
    db.close();
}
}

```

MySimpleCursorAdapter.java

```
class MySimpleCursorAdapter extends SimpleCursorAdapter {
    private LayoutInflater mInflater;
    private Context mContext;
    Cursor cur;    int layout;
    public MySimpleCursorAdapter(Context context, int layout, Cursor c,
                                String[] from, int[] to, int flags) {
        super(context, layout, c, from, to, flags);
        this.layout = layout; mContext = context;
        mInflater = LayoutInflater.from(context); cur = c;
    }
    @Override
    public void bindView(View view, Context context, Cursor cursor) {
        ImageView notePhoto = (ImageView) view.findViewById(R.id.photo);
        TextView noteNum = (TextView) view.findViewById(R.id.num);
        TextView noteName = (TextView) view.findViewById(R.id.name);
        Log.d("xxxx", "cur="+cursor.getCount()+" , c_count="+cursor.getColumnCount());
        notePhoto.setImageResource(getResourceByReflect(cur.getString(1)));
        noteNum.setText("" + cur.getLong(2));
        noteName.setText("" + cur.getString(3));
    }
    @Override
    public View newView(Context arg0, Cursor arg1, ViewGroup arg2) {
        return mInflater.inflate(layout, arg2, false);
    }
}
```

```

int getResource(String imageName) {
    //Context ctx = getBaseContext();
    int resId = mContext.getResources().getIdentifier(imageName, "drawable",
                                                         mContext.getPackageName());

    return resId;
}

public int getResourceByReflect(String imageName) {
    Class drawable = R.drawable.class;
    Field field = null;
    int r_id;
    try {
        field = drawable.getField(imageName);
        r_id = field.getInt(field.getName());
    } catch (Exception e) {
        r_id = R.drawable.cwt;
        Log.e("ERROR", "PICTURE NOT FOUND!");
    }
    return r_id;
}
}

```

DBOpenHelper.java

```
class DBOpenHelper extends SQLiteOpenHelper {
    private final String TAG = "MySQLiteOpenHelper";

    public DBOpenHelper(Context context, String dbName,
                        SQLiteDatabase.CursorFactory factory, int dbVersion) {
        //上下文, 数据库名称, 游标工厂, 数据库版本号 (>=1)    优点: 可以处理版本变化
        super(context, dbName, factory, dbVersion);
        Log.d(TAG, "MySQLiteOpenHelper");
    }
    //可以在这里创建表
    @Override
    public void onCreate(SQLiteDatabase db) {
        Log.d(TAG, "onCreate");
    }
    //如果版本号发生变化, 可以在这里做一些事, 例如, 为某个表增加一列
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        Log.d(TAG, "onUpgrade");
    }
}
```

activity_main.xml

```
<ListView
    android:id="@+id/listView"
    android:layout_width="300dp"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:fadeScrollbars="true"
    android:requiresFadingEdge="vertical"
    android:fadingEdgeLength="200dp"/>
```

list_item.xml

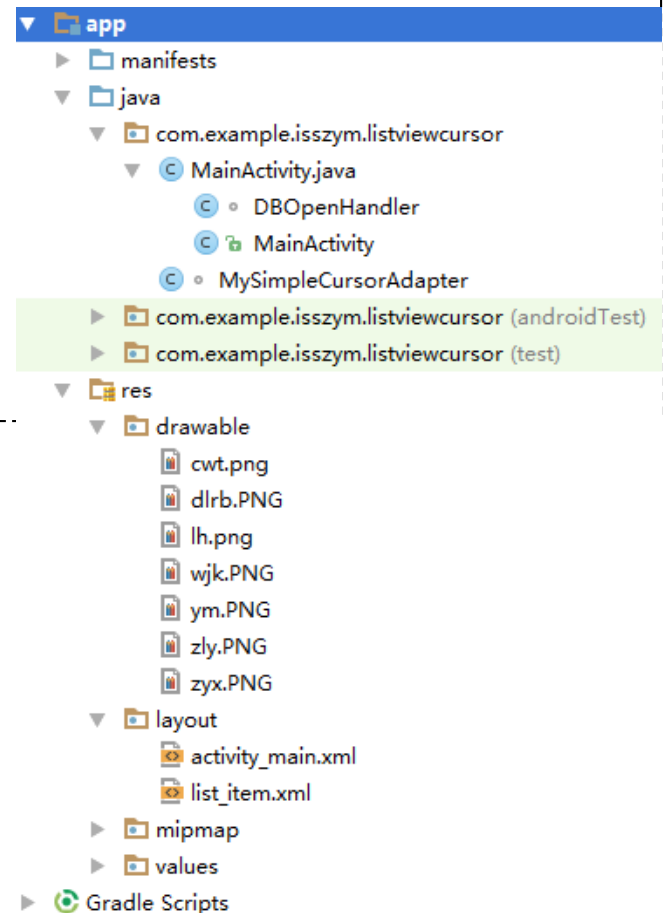
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="10dp" >
    <ImageView
        android:layout_width="60dp"
        android:layout_height="60dp"
        android:id="@+id/photo"
        android:text="photo"
        android:src="@drawable/cwt"
        android:layout_marginLeft="@dimen/activity_horizontal_margin"/>
```



```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/name"
    android:text="姓名"
    android:layout_marginLeft="@dimen/activity_horizontal_margin" />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/num"
    android:text="学号"
    android:layout_marginLeft
        ="@dimen/activity_horizontal_margin" />
</LinearLayout>

```



□ ListView的其他常用属性

- `android:stackFromBottom="true"`
- `android:transcriptMode="alwaysScroll"`
- `android:cacheColorHint="#F00"`
- `android:cacheColorHint="#00000000"`
- `android:divider="@drawable/divider"`
- `android:divider="@null"`
- `android:requiresFadingEdge="true"`
- `android:fadingEdgeLength="16dp"`
- `android:scrollbars="none"`
- `android:fadeScrollbars="true"`
- `android:entries="@array/city"`
- `setVerticalScrollBarEnabled(true)`

显示列表的末尾内容

最新的条目自动滚动到可视范围内

设置背景颜色，如果用图片做背景

如果要用background设置背景图

每一项之间用一个图片做为间隔

去掉item之间的分割线

上边和下边有黑色的阴影

上下边阴影的长度。

隐藏listView的滚动条

实现滚动条的自动隐藏和显示。

设置列表内容

用于设置垂直滚动条

网格框(GridView)

[参考](#)

- GridView可以采用多行多列方式排布单元(item)，GridView与ListView一样，都需要通过Adapter来提供显示的数据，ListView可以使用android:entries 来得到数据，但GridView不可以，必须通过适配器来为其添加数据。其常用属性：

`android:numColumns="3"`

用于设置列数

`android:columnWidth="120dp"`

用于设置列的宽度

`android:stretchMode="none"`

用于设置拉伸模式

`android:horizontalSpacing`

用于设置各元素之间的水平间距

`android:verticalSpacing`

用于设置各元素之间的垂直间距

`android:gravity`

用于设置对齐方式

- GridView的事件和ListView一样，都是设置
`setOnItemClickListener(OnItemClickListener listener);`

stretchMode

none(不拉伸)

spacingWidth(仅拉伸元素之间的间距)

columnWidth(仅拉伸表格元素本身)

spacingWidthUniform(元素之间的间距一起拉伸)



动态生成控件

- 前面的控件都是来自配置文件，如果控件数量不确定，可以临时编程生成。

第一步、生成新控件

```
TextView myTextView = new TextView(MainActivity.this);
```

第二步、设置好新控件的尺寸和边距后把新控件放入布局中

```
LinearLayout.LayoutParams textViewLP  
    = new LinearLayout.LayoutParams(  
        LinearLayout.LayoutParams.WRAP_CONTENT,    // width  
        LinearLayout.LayoutParams.WRAP_CONTENT);    // height  
textViewLP.setMargins(0,20,0,0);  
LinearLayout ll =(LinearLayout)findViewById(R.id.activity_main);  
ll.addView(myTextView, textViewLP);
```

MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        LinearLayout ll =(LinearLayout)findViewById(R.id.activity_main);
        for(int i=0;i<16;i++) {
            TextView myTextView = new TextView(MainActivity.this);
            myTextView.setText("第"+i+"行, Hello World!");
            myTextView.setTextSize(30);
            myTextView.setBackgroundColor(Color.argb(255,200,200,255));
            LinearLayout.LayoutParams textViewLP
                = new LinearLayout.LayoutParams(
                    LinearLayout.LayoutParams.WRAP_CONTENT,
                    LinearLayout.LayoutParams.WRAP_CONTENT);
            textViewLP.setMargins(0,20,0,0);
            ll.addView(myTextView, textViewLP);
        }
    }
}
```

项目名: DynamicView



```

activity_main.xml
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context="com.example.isszym.dynamicview.MainActivity"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp">

    <LinearLayout
        android:id="@+id/activity_main"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical" />
</ScrollView>

```



ScrollView只能包含一个子女元素。

布局(Layout)目录

线性布局(LinearLayout)

orientation

相对布局 (RelativeLayout)

layout_centerHorizontal

layout_centerInparent

layout_alignParentTop

layout_alignParentStart

layout_alignBaseline

layout_alignWithParentIfMissing

layout_below

layout_toLeftOf

layout_toStartOf

layout_alignTop

layout_alignStart

帧布局(FrameLayout)

layout_gravity

top|left|center_vertical|center
gravity

setVisibility

TextView.INVISIBLE

TextView.GONE

表布局(TableLayout)

shrinkColumns

stretchColumns

collapseColumns

网格布局(GridLayout)

rowCount

columnCount

layout_columnSpan

layout_rowSpan

布局扩展(LayoutInflater)

LayoutInflater.inflate(R.layout.activity_main, null)

动态创建布局(DynamicLayout)

LinearLayout ll = new LinearLayout(this);

ll.setOrientation(LinearLayout.VERTICAL);

addContentView(ll, LayoutParams)

布局(Layout)

- 线性布局 (LinearLayout)
布局一种放置控件的容器。线性布局把所包含的控件垂直或水平排列。
- 相对布局 (RelativeLayout)
控件相对于其它控件摆放。
- 帧布局 (FrameLayout)
所有控件叠放在一起，越后面越在上层。
- 表格布局 (TableLayout)
多行布局，每行可以放入若干控件作为列，并可以统一设置每列属性。
- 网格布局 (GridLayout)
多行多列布局，一个控件可以占据一行一列，也可以跨越多列或多行。

* 一个布局中可以包含其它布局。

* 控件是View，Layout是GroupView，GroupView也是View。

[Constraint Layout](#) [2](#)
[FlexboxLayout](#)

• 线性布局

项目名: **LinearLayout**

[参考](#)

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android=
    "http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="left|center_vertical">
    <Button
        android:id="@+id/bn1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="20sp"
        android:text="按钮1"/>

    .....
    <Button
        android:id="@+id/bn4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="20sp"
        android:text="按钮4"/>
</LinearLayout>
```

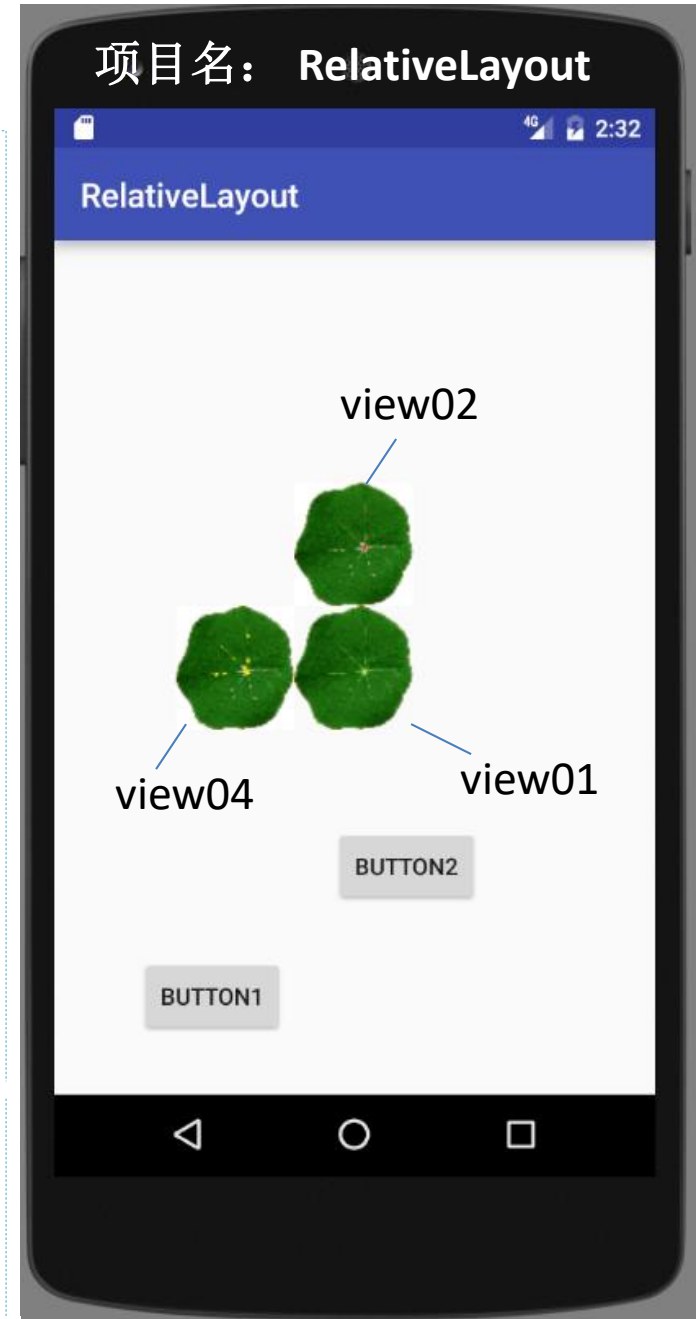


- 水平布局: `android:orientation="horizontal"`
- `android:gravity`用于说明控件或文字在其内部的位置; `android:layout_gravity`说明控件在父元素中的位置。

• 相对布局 (RelativeLayout)

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android=
    "http://schemas.android.com/apk/res/android"
    android:id="@+id/_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:id="@+id/view01"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/leaf"
        android:layout_centerInParent="true"/>
    <TextView
        android:id="@+id/view02"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/leaf2"
        android:layout_above="@+id/view01"
        android:layout_alignLeft="@+id/view01"/>
    <TextView
        android:id="@+id/view04"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/leaf3"
        android:layout_toLeftOf="@+id/view01"
        android:layout_alignTop="@+id/view01"/>
```



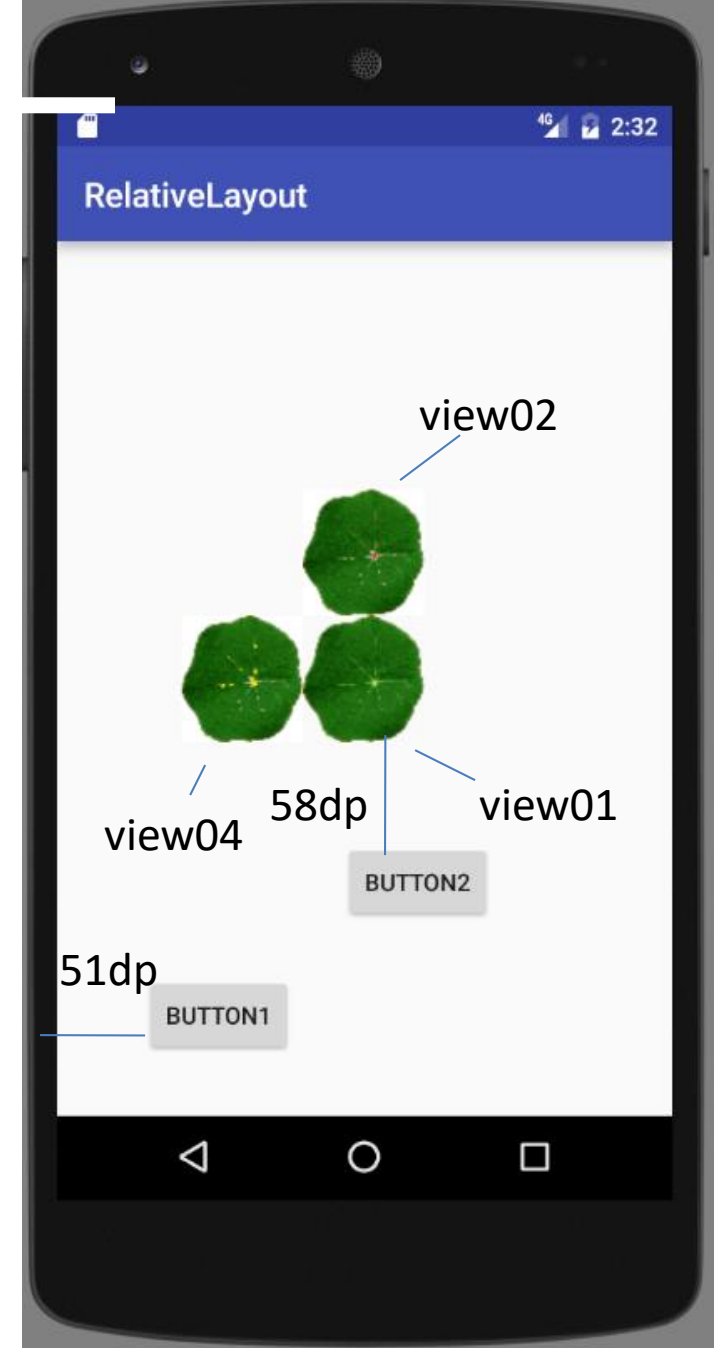
<Button

```
android:text="Button1"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignParentTop="true"  
android:layout_alignParentLeft="true"  
android:layout_marginLeft="51dp"  
android:layout_marginTop="429dp"  
android:id="@+id/button" />
```

<Button

```
android:text="Button2"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:id="@+id/button2"  
android:layout_below="@+id/view01"  
android:layout_toRightOf="@+id/view04"  
android:layout_marginLeft="23dp"  
android:layout_marginTop="58dp" />
```

</RelativeLayout>



`android:layout_centerHorizontal`
`android:layout_centerVertical`
`android:layout_centerInparent`
`android:layout_alignParentTop`
`android:layout_alignParentRight`
`android:layout_alignParentBottom`
`android:layout_alignParentLeft`
`android:layout_alignParentStart`
`android:layout_alignParentEnd`
`android:layout_alignBaseline`
`android:layout_alignWithParentIfMissing`

`android:layout_below`
`android:layout_above`
`android:layout_toLeftOf`
`android:layout_toRightOf`
`android:layout_toStartOf`
`android:layout_toEndOf`

在父容器中水平居中
在父容器中垂直居中
在父容器中水平垂直都居中
贴紧父容器的上边缘
贴紧父容器的右边缘
贴紧父容器的下边缘
贴紧父容器的左边缘
紧贴父容器的开始位置(默认为左边缘)
紧贴父容器的结束位置(默认为右边缘)
本元素的文本与父容器基线对齐
以父容器做参照物(如果没有参照元素)

在某控件的下方
在某控件的上方
在某控件的左边
在某控件的右边
在某控件开始之前 (默认为toLeftOf)
在某控件结束之后 (默认为toRightOf)

`android:layout_alignTop`
`android:layout_alignLeft`
`android:layout_alignBottom`
`android:layout_alignRight`
`android:layout_alignStart`
`android:layout_alignEnd`

本元素的上边缘和某元素的的上边缘对齐
本元素的左边缘和某元素的的左边缘对齐
本元素的下边缘和某元素的的下边缘对齐
本元素的右边缘和某元素的的右边缘对齐
本元素与父元素开始处对齐（默认为左边缘）
本元素与父元素结束处对齐（默认为右边缘）

`android:layout_gravity`
`android:gravity`

控件在父元素中的摆放方式（见FrameLayout）
控件内容的摆放方式

* 对于文字从左到右显式模式，Start与Left相同，End与Right相同，对于从右到左模式，正好相反，Start与Right相同，End与Left相同。

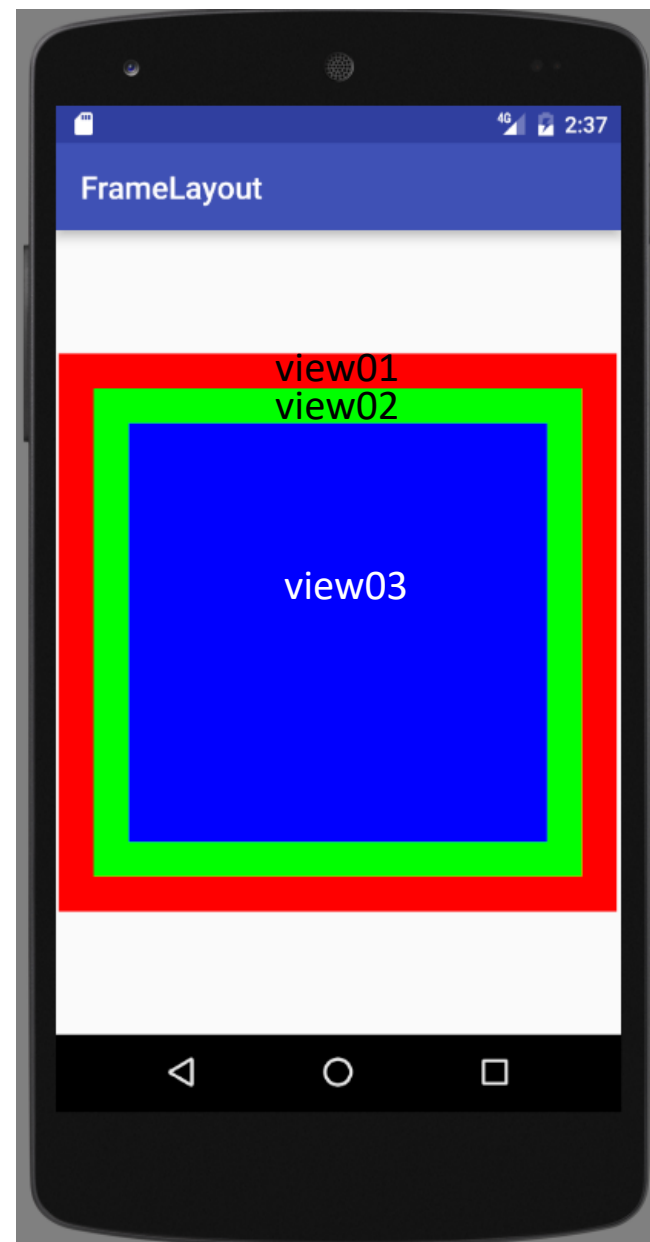
• 帧布局(FrameLayout)

[参考](#)

activity_main.xml 项目名: **FrameLayout**

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android=
    "http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:id="@+id/view01"
        android:layout_width="160pt"
        android:layout_height="160pt"
        android:layout_gravity="center"
        android:background="#f00"/>
    <TextView
        android:id="@+id/view02"
        android:layout_width="140pt"
        android:layout_height="140pt"
        android:layout_gravity="center"
        android:background="#0f0"/>
    <TextView
        android:id="@+id/view03"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:width="120pt"
        android:height="120pt"
        android:background="#00f"/>
</FrameLayout>
```

先定义的
TextView位于
底层后定义的
TextView位于
上层



- layout_gravity在LinearLayout中按方向取值才有效，在FrameLayout中才可以使用全部的值。

activity_main.xml

```
<FrameLayout xmlns:android
=http://schemas.android.com/apk/res/android
    android:layout_width="match_parent" 省略了padding
    android:layout_height="match_parent">
    <TextView android:layout_width="200dp"
        android:layout_height="200dp"
        android:background="#8F8"
        android:layout_gravity="left|top"
        android:gravity="right|top"
        android:textSize="24sp"
        android:text="Hello World!" />
    <TextView android:layout_width="200dp"
        android:layout_height="200dp"
        android:background="#F8F"
        android:layout_gravity="center"
        android:gravity="center"
        android:textSize="24sp"
        android:text="Hello World!" />
    <TextView android:layout_width="200dp"
        android:layout_height="200dp"
        android:background="#88F"
        android:layout_gravity="bottom|right"
        android:gravity="bottom|left"
        android:textSize="24sp"
        android:text="Hello World!" />
</FrameLayout>
```



android:layout_gravity和android:gravity的取值:

值	说明
top bottom left right	将对象放在其容器的顶部/底部/左侧/右侧, 不改变其大小。
center_vertical center_horizontal center	垂直、水平、双方向居中, 不改变其大小。
fill_vertical fill_horizontal fill	必要的时候增加对象的垂直(水平或双向)方向大小, 以完全充满其容器。
clip_vertical clip_horizontal	clip_vertical: 附加选项, 用于按照容器的边来剪切对象的顶部和/或底部的内容。 剪切基于其纵向对齐设置: 顶部对齐时, 剪切底部; 底部对齐时剪切顶部; 除此之外剪切顶部和底部.垂直方向裁剪。 clip_horizontal类似。

- 如果要想view03完全消失(不占空间), 要如下设置:

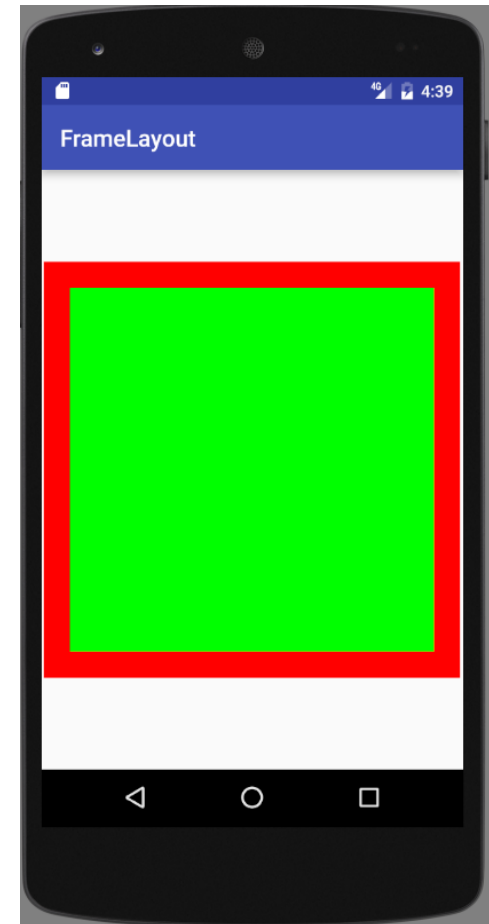
```
TextView tv3=(TextView)findViewById(R.id.view03);  
tv3.setVisibility(TextView.GONE);
```

- 如果只是让view03看不见(占空间), 要如下设置:

```
tv3.setVisibility(TextView.VISIBILITY);
```

- 在界面XML中配置visibility无效

```
android:visibility="invisible"  
android:visibility="gone"
```



• 表布局(TableLayout)

[参考](#)

项目名: TableLayout

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!-- 定义第一个表格布局, 指定第2列允许收缩, 第3、4列允许
拉伸 第5列隐藏-->
<TableLayout xmlns:android
    ="http://schemas.android.com/apk/res/android"
    android:id="@+id/TableLayout01"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:shrinkColumns="1"
    android:stretchColumns="2,3"
    android:collapseColumns="4">
    <Button
        android:id="@+id/ok1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="16sp"
        android:text="独自一行的按钮" />
    <TableRow>
        <Button
            android:id="@+id/ok2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="16sp"
            android:text="普通按钮" />
```



```

<Button
    android:id="@+id/ok3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="16sp"
    android:text="收缩的按钮" />
<Button
    android:id="@+id/ok4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="16sp"
    android:text="拉伸的按钮" />
<Button
    android:id="@+id/ok5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="16sp"
    android:text="拉伸的按钮" />
<Button
    android:id="@+id/ok6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="16sp"
    android:text="隐藏的按钮" />
</TableRow>

```



```
<TableRow>
    <!-- 为该表格行添加三个按钮 -->
    <Button
        android:id="@+id/ok7"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="16sp"
        android:text="普通按钮" />

    <Button
        android:id="@+id/ok8"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="16sp"
        android:text="收缩的按钮" />
    <Button
        android:id="@+id/ok9"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="16sp"
        android:text="拉伸的按钮" />
</TableRow>
</TableLayout>
```


• 网格布局(GridLayout)

[参考](#)

项目名: GridLayout

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<GridLayout xmlns:android
="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:rowCount="6"
    android:columnCount="4"
    android:id="@+id/root">
    <TextView android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_columnSpan="4"
        android:textSize="50sp"
        android:layout_marginLeft="2pt"
        android:layout_marginRight="2pt"
        android:padding="3pt"
        android:layout_gravity="right"
        android:background="#eee"
        android:textColor="#000"
        android:text="0"/>
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_columnSpan="4"
        android:textSize="20sp"
        android:text="清除"/>
</GridLayout>
```



MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    GridLayout gridLayout;
    String[] chars = new String[] {           // 定义16个按钮的文本
        "7" , "8" , "9" , "÷" , "4" , "5" , "6" , "×",
        "1" , "2" , "3" , "-", "." , "0" , "=", "+" };

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        gridLayout = (GridLayout) findViewById(R.id.root);
        for(int i = 0 ; i < chars.length ; i++) {
            Button bn = new Button(this);
            bn.setText(chars[i]);
            bn.setTextSize(40);                // 设置该按钮的字号大小
            bn.setPadding(5 , 35 , 5 , 35); // 设置按钮四周的空白区域
            GridLayout.Spec rowSpec=GridLayout.spec(i/4+2); //指定所在的行
            GridLayout.Spec columnSpec = GridLayout.spec(i%4); //指定所在的列
            GridLayout.LayoutParams params = new GridLayout.LayoutParams(
                rowSpec , columnSpec);
            params.setGravity(Gravity.FILL); // 指定该组件占满父容器
            gridLayout.addView(bn , params);
        }
    }
}
```

动态创建布局

- 先产生一个布局实例，再修改宽度和高度等参数，最后用addContentView()加入到Activity实例中。

```
LinearLayout ll = new LinearLayout(this);  
ll.setOrientation(LinearLayout.VERTICAL);  
LinearLayout.LayoutParams lp  
    = new LinearLayout.LayoutParams(  
        LinearLayout.LayoutParams.MATCH_PARENT,  
        LinearLayout.LayoutParams.MATCH_PARENT);  
this.addContentView(ll, lp); // 可省略this
```

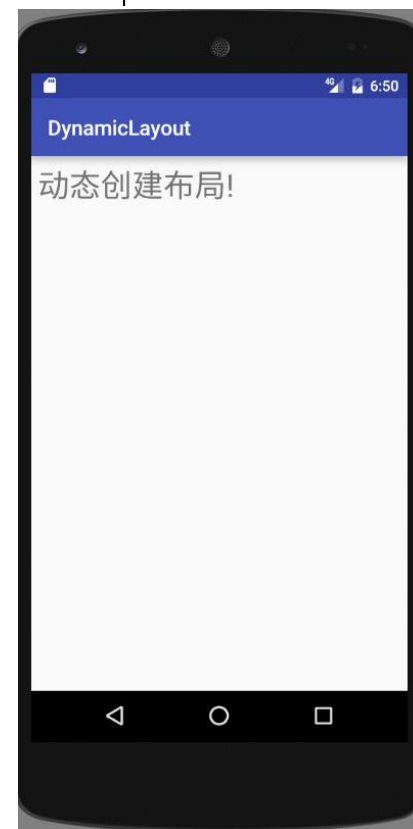
- Activity常用页面有一个叫做DecorView的顶级View，DecorView包含一个垂直方向的LinearLayout，LinearLayout由标题栏和内容栏两部分组成。
- 内容栏是一个FrameLayout，在Activity中调用方法setContentView()就是将一个布局inflate转换成View添加到这个FrameLayout中。
- 在已有的布局中加入布局或控件采用addView()。



项目名: DynamicLayout

```
----- MainActivity.java -----
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        LinearLayout ll = new LinearLayout(MainActivity.this);
        ll.setOrientation(LinearLayout.VERTICAL);
        LinearLayout.LayoutParams lp;    // 向父视图传递布局参数
        lp = new LinearLayout.LayoutParams(
            LinearLayout.LayoutParams.MATCH_PARENT,
            LinearLayout.LayoutParams.MATCH_PARENT);
        this.addView(ll, lp);    // 加入Activity, 可省略this
        TextView myTextView = new TextView(MainActivity.this);
        myTextView.setText("动态创建布局!");
        myTextView.setTextSize(30);
        LinearLayout.LayoutParams textViewLP; // 向父视图传布局参数
        textViewLP = new LinearLayout.LayoutParams(
            LinearLayout.LayoutParams.MATCH_PARENT,    // 宽
            LinearLayout.LayoutParams.WRAP_CONTENT);    // 高
        ll.setPadding(20, 20, 10, 10);
        ll.addView(myTextView, textViewLP);
    }
}
```

*不需要界面文件



[LayoutParams](#)

布局填充器 (LayoutInflater)

把一个新布局用于加入到已存在的布局中需要使用布局填充器 **LayoutInflater**。获得布局填充器有三种方法，前两种方式都会被转换成第三种方式，它们都是获取某个**Activity**实例的布局填充器服务：

```
LayoutInflater inflater = getLayoutInflater();           // 当前Activity实例
LayoutInflater inflater = LayoutInflater.from(context);  // 某个Activity实例
LayoutInflater inflater = (LayoutInflater)context.getSystemService(
    Context.LAYOUT_INFLATER_SERVICE);
```

LayoutInflater通过方法inflate()可以把布局转换为View：

```
public View inflate(int resource, ViewGroup parent, boolean attachToRoot)
```

其中，resource为布局资源的ID，parent为null时，填充布局的layout_width和layout_height无效，而parent是一个View时它们有效。attachToRoot为true时inflate产生的view直接加入到parent中，否则需要用addView()加入。

```
public View inflate(int resource, ViewGroup parent)
```

这个格式调用了上一种格式，第三个参数为parent!=null，也就是说，parent不为null时inflate得到的view将自动加入parent，否则需要用addView加入布局或者用setContentView加入到FrameLayout中。

项目名: LayoutInflater

```
public class MainActivity extends AppCompatActivity {
    LayoutInflater inflater;
    LinearLayout layout;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        inflater = LayoutInflater.from(this);
        LinearLayout mainlayout = (LinearLayout)inflater.inflate(R.layout.activity_main, null);
        setContentView(mainlayout);
        layout = (LinearLayout)findViewById(R.id.browse);
        Button btn=(Button)findViewById(R.id.button);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                LinearLayout ll = (LinearLayout)inflater.inflate(R.layout.line_item, null);
                TextView num = (TextView)ll.findViewById(R.id.num);
                num.setText(((EditText)findViewById(R.id.num)).getText());
                TextView name = (TextView)ll.findViewById(R.id.name);
                name.setText(((EditText)findViewById(R.id.name)).getText());
                TextView age = (TextView)ll.findViewById(R.id.age);
                age.setText(((EditText)findViewById(R.id.age)).getText());
                LinearLayout.LayoutParams lp = new LinearLayout.LayoutParams(
                    LinearLayout.LayoutParams.MATCH_PARENT,
                    LinearLayout.LayoutParams.WRAP_CONTENT);
                ll.setPadding(20, 20, 10, 10);
                layout.addView(ll, lp);
            }
        });
    }
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    ...
>
<LinearLayout
    android:id="@+id/editline"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <EditText
        android:id="@+id/num"
        android:hint="学号"
        android:textSize="20sp"
        android:layout_width="120dp"
        android:layout_height="wrap_content"
        android:layout_marginRight="10dp" />
    <EditText
        android:id="@+id/name"
        android:hint="姓名"
        ... />
    <EditText
        android:id="@+id/age"
        android:hint="年龄"
        ... />
</LinearLayout>
```

<Button

```
    android:id="@+id/button"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="21dp"  
    android:gravity="center_horizontal"  
    android:text="增加"  
    android:textSize="18sp" />
```

<ScrollView

```
    android:layout_width="match_parent"  
    android:layout_height="match_parent">
```

<LinearLayout

```
    android:id="@+id/browse"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:orientation="vertical"></LinearLayout>
```

</ScrollView>

</LinearLayout>

line_item.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout android:id="@+id/line_item"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    ...>
    <TextView
        android:id="@+id/num"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginRight="16dp"
        android:text="001"
        android:textSize="24sp" />

    <TextView
        android:id="@+id/name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginRight="16dp"
        android:text="Hello"
        android:textSize="24sp" />

    <TextView
        android:id="@+id/age"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="20"
        android:textSize="24sp" />
</LinearLayout>
```

附录1、课件所学的控件

【列表和适配器】 73

适配器(Adapter)

MVC模式

下拉框(Spinner)

SpinnerStatic

arrays.xml

entries

spinnerMode

setOnItemSelectedListener()

SpinnerCustom

LayoutInflater

getResources().getStringArray()

列表框(ListView)

概述

stackFromBottom

transcriptMode

cacheColorHint

divider

requiresFadingEdge

fadingEdgeLength

scrollbars

fadeScrollbars

setVerticalScrollBarEnabled()

Static(Arrays.xml)

longClickable

setOnItemClickListener()

setOnItemLongClickListener()

ArrayAdapter

setAdapter

simple_list_item_multiple_choice

simple_list_item_single_choice

simple_list_item_1

custom_simple_list_item_1

setChoiceMode

CHOICE_MODE_MULTIPLE

CHOICE_MODE_SINGLE

setOnClickListener

SimpleAdapter

setAdapter(Context,ArrayList<HashMap>,item_list
String[]from,int[] to)

CursorAdapter

requiresFadingEdge

fadingEdgeLength

getResource(String imageName)

getResourceByReflect(String imageName)

SQLiteOpenHelper

setOnItemSelectedListener()

网格框(GridView)

numColumns

columnWidth

stretchMode

horizontalSpacing

verticalSpacing

setOnItemClickListener()

动态生成控件

Layout.addView(view,LayoutParams)

ScrollView

【布局】 114

线性布局(LinearLayout)

orientation

相对布局 (Relative Layout)

layout_centerHorizontal

layout_centerInparent

layout_alignParentTop

layout_alignParentStart

layout_alignBaseline

layout_alignWithParentIfMissing

layout_below

layout_toLeftOf

layout_toStartOf

layout_alignTop

layout_alignStart

帧布局(FrameLayout)

layout_gravity

top|left|center_vertical|center

gravity

setVisibility

TextView.VISIBILITY

TextView.GONE

表布局(TableLayout)

shrinkColumns

stretchColumns

collapseColumns

网格布局(GridLayout)

rowCount

columnCount

layout_columnSpan

layout_rowSpan

动态创建布局

```
LinearLayout ll = new LinearLayout(this);
```

```
ll.setOrientation(LinearLayout.VERTICAL);
```

```
addContentView(ll, LayoutParams)
```

布局扩展(LayoutInflater)

```
LayoutInflater.inflate(R.layout.activity_main, null)
```

附录1、课件所学的控制件

view(视图) -- box(框)