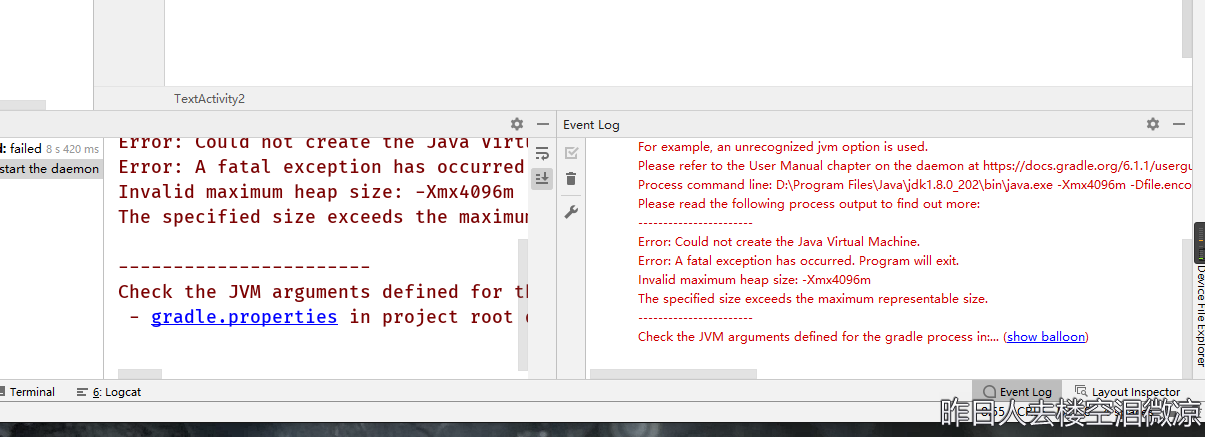
## 安卓模拟器换地方

<https://blog.csdn.net/u010940300/article/details/43909509>



<https://blog.csdn.net/dhx20022889/article/details/44919905>

AdbHostServer.cpp:102: Unable to connect to adb daemon on port: 5037解决

AndroidStudio Cause: unable to find valid certification path to requested target

新建项目的时候，语言选择正确的（Java）

<**View  
 android:layout\_width="50dp"  
 android:layout\_height="match\_parent"  
 android:background="#fffff5"  
 android:layout\_weight="1"**/>  
<**View  
 android:layout\_width="0dp"  
 android:layout\_height="match\_parent"  
 android:background="#eeeee5"  
 android:layout\_weight="1"**/>

Layout\_weight:是权重的意思，表示这个模块占父空间剩余部分的多少；把剩余内容按照权重分配

**private** Button **mBtnTextView**;

**mBtnTextView** = findViewById(R.id.***btn\_textview***);

得到一个元素

**mBtnTextView**.setOnClickListener(**new** View.OnClickListener() {  
 **public void** onClick(View view) {  
 *//跳转到TextViewActivity* Intent intent = **new** Intent(TextActivity.**this**,TextViewActivity.**class**);  
 startActivity(intent);  
 }  
});

在该元素上创建一个鼠标点击监听事件

<**TextView  
 android:id="@+id/T1\_4"//注册id  
 android:layout\_width="wrap\_content"//包裹内容  
 android:layout\_height="wrap\_content"  
 android:text="稻草"  
 android:drawableBottom="@mipmap/ic\_launcher\_round"//在文档底部添加图片  
 android:textColor="#000000"//文字颜色  
 android:textSize="24dp"//文字大小  
 android:layout\_marginTop="10dp"//设置外边距** />

中划线S设置

**private** TextView **mTv4**;

**mTv4** = findViewById(R.id.***T1\_5***);  
**mTv4**.getPaint().setFlags(Paint.***STRIKE\_THRU\_TEXT\_FLAG***);

**mTv4**.getPaint().setAntiAlias(**true**);*//去除锯齿*

下划线设置

找到控件

**mtv6** = findViewById(R.id.***T1\_6***);  
**mtv6**.getPaint().setFlags(Paint.***UNDERLINE\_TEXT\_FLAG***);

删除按钮格式

**android:button="@null"**

**安卓控件置底**

<https://blog.csdn.net/wingsofpiano/article/details/45950069>

# ArrayAdapter的使用：

1. 声明spinner和ArrayAdapter
2. ArrayAdapter.createFromResource方法：当前实例（this），资源文件id（R.array.XXX）,列表显示风格id（R.layout.XXX）
3. spinner.setAdapter(ArrayAdapter)

## ArrayAdapter的参数说明

//初始化适配器

adapter=new ArrayAdapter<>(this,android.R.layout.simple\_expandable\_list\_item\_1,datas);

第一个参数：context上下文对象

第二个参数：每一个item的样式，可以使用系统提供，也可以自定义就是一个TextView

第三个参数：数据源，要显示的数据

系统提供的item的样式，可以试一试

simple\_list\_item1:单独的一行文本框

simple\_list\_item2:有两个文本框组成

simple\_list\_item\_checked每项都是由一个已选中的列表项

simple\_list\_item\_multiple\_choice:都带有一个复选框

simple\_list\_item\_single\_choice：都带有一个单选框

## edittext与Spinner

拿到edittext ，给spinner setOnItemSelectedListener 匿名方法注意也是XXXselectedXXX，

public void onItemSelected(AdapterView<?> parent, View view,int position, long id)//选项被选中时触发该方法

AdapterView:选项被选中的视图列表

view：当前选中的item选项

position：被选中的item选项在视图中的位置

id:被选中item选项的id

public void onNothingSelected(AdapterView<?> parent)//没有选项时触发

选择和未选择的方法都要有

# ListView

entries提供列表项

适配器Adapter提供列表项

创建步骤：

第一步：在布局文件中定义一个空的ListView

第二步：在Activity中：

根据id得到ListView

创建数据源，如：String str[]={“China”,”American”。。};

创建ArrayAdapter对象

将ListView和ArrayAdapter对象绑定

使用ListView中的方法：

public void setAdapter (ListAdapter adapter)

SimpleAdapter 构造函数如下：

SimpleAdapter（Context context，List<? extends Map <String, ?>> data, int resource, String[] from,int[] to）

Context context,就是当前的上下文,

<Map<String, ?>> data就是绑定的数据,这个式吧数据绑定到手机上的那个数据源,

resource参数值就指定的ListView的列表的布局。(各种LinearLayout)，现在假设布局文件中包含了两个组件：TextView和EditText，R.id.textview , R.id.editext分别为TextVieew和EditText的id。那么在加载列表项式，date是数据源。

From参数Map对象的key，而to表示组件的id，例如，本例中的参数值为from=new String[]{‘textview’,’editext’}。意思就是将Map对象中key为editext的value绑定到R.id.editext上。

## SimpleAdapter 创建使用过程：

1. 定义好每个Item中要存放的东西；
2. 定义好每个ListItem的样式，在layout中；
3. **创建好List<Map<String, Object>> 的ArrayList对象；**

**private List<Map<String,Object>> landlordList;**

**landlordList = new ArrayList<>();**

**landlordList.add(landlordMap);**

1. 在插入item循环中，每新建好，**put**好一个HashMap（Map<String, Object> listItem = **new** HashMap<String, Object>();）就要添加到List<Map>当中



1. 构造SimpleAdapter ，注意五个参数（）
2. ListView.setAdapter()

## 给挂载了SimpleAdapter的ListView的Item加监听事件

listview .setOn**Item**ClickListener(new **AdapterView**.OnItemClickListener(){

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

TextView tvname = view.findViewById(R.id.four\_name);

TextView tvdesc = view.findViewById(R.id.four\_desc);

mytextview.setText("请选择：："+tvname.getText()+tvdesc.getText());

})

PS：：view（小布局）：你点击的Listview的某一项的内容，来源于adapter。如用((TextView)view).getText().toString()，可以取出点击的这一项的内容，转为string类型。

**获得小布局View组件中的id为four\_name的TextView组件**

position：是Adapter的某一项的位置，如点击了listview第2项，而第2项对应的是adapter的第2个数值，那此时position的值就为1了。

id：值为点击了Listview的哪一项对应的数值，点击了listview第2项，那id就等于1。一般和position相同。这里也有可能对应别的数值

## 构建自己的Adapter

StudentAdapter类继承了BaseAdapter类， 实现了ListView中的列表项的完全控制

\*listStudent集合里保存了要在Listview中显示的所有数据

\*context是上下文对象，在这里是指当前Activity对象，调用时用当前的Acticity.this即可

因为进城了BaseAdapter类，所以需要实现五个方法：

public StudentAdapter（List<> XXList, Context context）本身的构造方法

getCount（）获取当前Adapter适配器的所承接的List的对象个数

getItem（int position）一般返回null

getItemId（int position）返回第Position的id，一般用position代替

重点了：getView（int position, View converView, ViewGroup parent）

思想：获得布局文件，转化成一个View对象，从而修改布局文件的内容

获得一个xml文件，用它来生成View对象convertView，里面定义了，例如TextView，Button，ProcessBar等控件对象，用到一个叫LayoutInflater的系统加载布局服务，把xml布局文件实例化为一个View对象，就是将Layout xml文件创建一个view对象，都要通过LayoutInflater.inflate（）来载入；

必须先获得LayoutInflater实例才能使用，有：：

LayoutInflater.from(this)

getLayoutInflater()

(LayoutInflater)getSystemService(LAYOUT\_INFLATER\_SERVICE)

inflate(每个Item的xml文件id，ViewGroup parent，false/true获得小/大布局)

convertView获得的ProgressBar需要final？？

通过 inflate得到你的xml对象，然后调用对象的getViewById得到构件后，就可以把自己Adapter类中的List元素的值传入到对象的构件里，例如实验中的LinkedList<Student>，我们就可以，将这一堆Student元素中的Name，ProgressBar等写入xml实例的View类当中。

定义LinkedList的时候要new

例：static LinkedList<Student> studentLinkedList = new LinkedList<>();

## Toast的用法

创建：：

Toast.makeText(getApplicationContext(),R.string.toast\_string,Toast.LENGTH\_SHORT).show();

最好写getApplication，而不是this；toast的文字，可以自己定义样式；

toast的显示长度；；最后最重要的，加上show();

toast优化

private static Toast toast ;

static boolean isShowing = false;

public void showToast(){

if (isShowing == false) {

toast = Toast.makeText(getApplicationContext(), R.string.toast\_string, Toast.LENGTH\_SHORT);

toast.show();

isShowing = true;

}else {

toast.setText(R.string.toast\_string);

isShowing = false;

toast.cancel();

}

}

在onclick调用showToast即可

## ListView中如何使用Button，让onClick和onItemClick事件共存

假如ListView的一条记录包含一个TextView文本框，一个Button按钮，怎样能让点击按钮有事件响应，同时点击这条记录也有事件响应?

解决方法如下：

将具有点击事件控件的Android:focusable属性置成false就可以了，（如这里就是按钮的控件，只要把按钮的控件的android:focusable属性置成false）；这样可以既捕捉onItemClick事件（在非button区域），又可以捕捉onClick等事件。

## ListView给item添加onitemclicklistener方法 暂时保留

problem solved

首先获得你要在当前activity的所需要的传入到下一个acticity的控件的数值，例如，在当前activity下的有两个TextView 和ProgressBar，那么在ListView的setOnItemClickListener方法下，就需要get到这两个控件，将他们的Textview的数值传到Bundle中，

bundle.putString/putInt等，然后由intent实例统一putExtras（bundle），传入到下一个activity,

非常重要：：Bundle 传数据的时候，putString，putInt等，到了跳转页面之后，取得数据就需要根据getText还是getInt来将Bundle中的数据进行类型的转换。