# UI

## 配置文件

main.xml。

## 视图

视图，基类为：android.view.View。为屏幕上的小部件。

视图组，基类：android.view.ViewGroud。布局，可以包含多个视图，本身也是视图。

## LinearLayout

单行或者单列排列，属性：

layout\_width,layout\_height。

值：

fill\_parent：填满父元素

fill\_wrap：当前内容区域

Layout\_marginTop，Layout\_marginBottom，

Layout\_marginLeft，Layout\_marginRight,layout\_weight

额外空间。

Layout\_gravity：子视图定位

layout\_x，layout\_y.：定位。

orientation：排版方式 vertical垂直

## 度量单位

常用

dp:在160dpi中相当1px，会根据手机的dpi自动调整

sp:字体

## AbsoluteLayout

绝对定位。layout\_x，layout\_y.：定位

## TableLayout

行列形式组织视图，<TableRow>指定行。

## RelativeLayout

指定子视图彼此之间的定位，

属性 值

layout\_alignParentTop bool

layout\_alignParentLeft bool

layout\_alignLeft id

layout\_alignRight

layout\_below

layout\_centerHorizontal

ID:

设置 id=”@+id/name”

引用 id=”@+id/name”

## FrameLayout

单个视图的占位符，可以嵌套在其他Layout中，其中的多个视图都在左上角，而且会一个覆盖一个。

## scollView

滑动显示一个大于屏幕的视图列表，但是最好取消其中视图自动获取焦点的功能

HorizontalScrollView，水平滚动条

## 适应屏幕方向

当手机改变方向改变是，都会重新布局。

解决方案：定位到屏幕的四边，可以用RelativeLayout实现，

重新设置大小和定位，新建layout\_land文件夹

控制活动方向：setRequstOrientation(ActivityInfo.SCREEN\_OPRENTATION\_LANDSCAPE).

或者配置：screenOrientation=”landscape”

## Action Bar

标题栏，可以配置：.noActionBar隐藏。也可以调用.hide()方法

添加动作项：

onCreateOptionsMenu(),

其中MenuItem.setShowAsAction(MenuItem.SHOW\_AS\_ACTION\_IF\_FOOM),表示如果有空间则显示动作项。

点击时：

onOptionsItemSelected(menuItem item)；

## 代码创建界面，监听

在onCreate()方法中，

点击事件：onKeyDown()

绑定事件：OnclickLIstenner类 click()

元素的:.setOnFocusChangeListener()方法

# 常用控件

TextView，文本控件（支持有限的Html）

 EditText，文本编辑控件（支持有限的Html）

 Button，按钮控件

ImageView，图片控件

ImageButton，图片按钮控件

Spinner，下拉选择控件

Dialog是android开发过程中最常用到的组件之一，通常是在当前activity前面出现的小窗口。Dialog通常是用来通知打断用户，执行直接跟应用程序进度相关的短暂任务(比如进度条或者登录框)。Dialog可分为以下几种类型：

警告对话框：Alertialog

进度对话框：ProgressDialog

日期选择对话框：DatePickerDialog

时间选择对话框：TimePickerDialog

自定义对话框：从Dialog继承

Toast，是一种简易的消息提示框

## 去掉标题栏

1. this.requestWindowFeature(Window.FEATURE\_NO\_TITLE);

//去掉标题栏

记住：这句代码要写在setContentView()前面。

2.在清单文件（manifest.xml）里面实现

<application android:icon="@drawable/icon"

        android:label="@string/app\_name"

        android:theme="@android:style/Theme.NoTitleBar">

这样用可以将整个应用设置成无标题栏，如果只需要在一个Activity设置成一个无标题栏的形式，只要把上面的第三行代码写到某一个Activity里面就可以了。

3.在style.xml文件里定义

<?xml version="1.0" encoding="UTF-8" ?>

<resources>

    <style name="notitle">

        <item name="android:windowNoTitle">true</item>

    </style>

</resources>

然后面manifest.xml中引用就可以了，这种方法稍麻烦了些。

<application android:icon="@drawable/icon"

        android:label="@string/app\_name"

        android:theme="@style/notitle">

## 状态栏一体化

/\*\* android 4.4

\* 设置状态栏背景状态

\*/

private void setTranslucentStatus()

{

if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.KITKAT)

{

Window win = getWindow();

WindowManager.LayoutParams winParams = win.getAttributes();

final int bits = WindowManager.LayoutParams.FLAG\_TRANSLUCENT\_STATUS;

winParams.flags |= bits;

win.setAttributes(winParams);

}

SystemStatusManager tintManager = new SystemStatusManager(this);

tintManager.setStatusBarTintEnabled(true);

tintManager.setStatusBarTintResource(0);//状态栏无背景

}