project（项目，工作空间）包含一个或多个module。

module（模块，应用）包含一个或多个app components。

app components（应用组件）

1、每个应用组件都可以作为应用的入口点。

2、任何应用都可以启动其它应用的组件（不能直接启动，需要向系统发消息）。

|  |  |  |
| --- | --- | --- |
| activity（活动） | 一屏。 | 通过Intent的异步消息进行启动。Activity能通过Intent返回结果。  startActivity()、startActivityForResult()  startService()  sendBroadcast()、sendOrderedBroadcast()、sendStickyBroadcast()等 |
| service（服务） | 后台运行，无界面。 |
| broadcast receiver（广播接收器） | （广播可以由系统或应用发起。） |
| content provider（内容提供程序） | 既可以管理私有数据也可以管理共享数据。 | 成为ContentResolver（内容解析程序）的请求目标时启动。使用数据的组件在ContentResolver对象上调用方法（query()），不直接访问content provider。 |

3、启动组件前，系统读取AndroidManifest.xml，确认组件存在、应用所需API级别、应用所需软硬件、应用所需API库等。

|  |  |  |
| --- | --- | --- |
| activity | <activity> | 未在清单文件中声明的 Activity、service和content provider对系统不可见，因此永远不会运行。 |
| service | <service> |
| content provider | <provider> |
| broadcast receiver | <receiver> | 可以动态创建并注册（registerReceiver()）。 |

声明activity时可以用Intent过滤器来让组件响应[要求一类应用中来一个的]Intent。

<uses-feature>android:required="true"必须要，android:required="false"不是必须要（运行时自己检查有没有）

<uses-sdk>

<uses-permission>

资源

1、SDK build tools（SDK构建工具）为project中的每个资源（在**应用名**build\generated\source\r\debug\**包路径**\R.java中）生成1个ID。编译project或module时，aapt（Android Asset Packaging Tool）生成 R 类。

默认资源：不带限定符的

1、资源放在res文件夹下的子文件夹中，子文件夹名可以是

|  |  |
| --- | --- |
| animator |  |
| anim |  |
| color |  |
| drawable | <shape xmlns:android="http://schemas.android.com/apk/res/android"  android:shape="rectangle">  <solid android:color="**资源**">  <corners android:radius="**数值**">  <padding  android:top="**数值**"  android:right="**数值**"  android:bottom="**数值**"  android:left="**数值**"/>  </shape> |
| mipmap |  |
| layout |  |
| menu |  |
| raw |  |
| values | res的其它子文件夹都是每个文件生成1个ID，values文件夹中<resources>元素的每个子元素生成1个ID。  arrays.xml、colors.xml、dimens.xml、strings.xml、styles.xml |
| xml |  |

备用资源：带限定符的

默认资源名和限定符之间、限定符和限定符之间用“-”分隔。

！！！若没有匹配当前设备的资源文件夹，应用崩溃，所以应该总是提供默认资源。

1、文件夹名不区分大小写。

2、每组限定符使用0个或1个。

3、若使用限定符，限定符顺序与表中的组的顺序相同。

4、匹配规则：根据限定符淘汰与设备配置冲突的资源文件夹（默认资源文件夹一定会被保留）；对表中每一个限定符组，若有文件夹有限定符，就淘汰所有没有限定符的文件夹（默认资源文件夹一定会被淘汰）；直到只剩下1个文件夹。所以表中的组的顺序比设备匹配的限定符数量更重要。

创建资源别名

访问资源

|  |  |
| --- | --- |
| 在XML中访问资源 | @(**包名**:)?**资源类型**/**资源名**  1、**资源名**不包括扩展名。 |
| 在代码中访问资源 | (**包名**.)?R.**资源类型**.**资源名**  1、**资源名**不包括扩展名。 |

package name每个成分不能含有中文、空格，不能只有数字。

project location不能含有中文、空格。

AndroidMenifest.xml

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

<manifest>

|  |  |
| --- | --- |
| xmlns:android="http://schemas.android.com/apk/res/android" | 声明包的命名空间。 |
| package="**包名**" | 声明应用的包。 |
|  |  |
|  |  |
|  |  |

<application>

|  |  |
| --- | --- |
| android:allowBackup="true" |  |
| android:icon="**资源**" |  |
| android:roundIcon="**资源**" |  |
| android:label="**资源**" | 应用标签，为应用指定名称。 |
| andoird:supportsRtl="true" |  |
| android:theme="**资源**" | 应用采用的主题。  "@style/AppTheme"  "@style/Theme.AppCompat "  "@style/Theme.AppCompat.Light.DarkActionBar" |

android.app.Activity类

<activity>

|  |  |
| --- | --- |
| android:name=".**类名**" |  |
| android:screenOrientation="landscape" |  |

getWindow().setFlags(WindowManager.LayoutParams.FLAG\_FULLSCREEN,WindowManager.LayoutParams.FLAG\_FULLSCREEN);

<intent-filter>

<action>

|  |  |
| --- | --- |
| android:name="android.intent.action.MAIN" |  |

<category>

|  |  |
| --- | --- |
| android:name="android.intent.category.LAUNCHER" |  |

AppCompatActivity类

继承该类的Activity带有Action Bar

布局资源文件，文件名命名符合Java命名规则，Ctrl+Alt+Space显示可选值列表

android.view.View类

findViewById(R.id.**控件id**)

属性

|  |  |  |  |
| --- | --- | --- | --- |
|  | android:id | setId(int) | 不能使用Java关键字。 |
|  | android:background="**资源**" | setBackgroundRersource(int) |  |
|  | android:visibility | setVisibility(int) |  |
| 尺寸 | android:minWidth | setMinimumWidth(int) |  |
| android:minHeight | setMinimumHeight(int) |  |
| android:padding |  |  |
| android:padding(Top|Right|Bottom|Left|Start|End) |  |  |
|  | android:elevation  z轴深度 | setElevation(float) |  |
| 是否响应事件 | android:clickable | setClickable(boolean) |  |
| android:longClickable | setLongClickable(boolean) |  |

事件

|  |  |  |
| --- | --- | --- |
|  | android:onClick |  |

ViewGroup类，继承View类，抽象类

|  |  |  |
| --- | --- | --- |
| ViewGroup.LayoutParams类，封装布局的位置、尺寸等信息 | android:layout\_width="(match\_parent|wrap\_content|**数值**)" | 视图在屏幕上占的宽度。 |
| android:layout\_height="(match\_parent|wrap\_content|**数值**)" | 视图在屏幕上占的高度。 |
| ViewGroup.MarginLayoutParams，控制子组件的外边距 | android:margin(Top|Right|Bottom|Left|Start|End) |  |

RelativeLayout相对布局管理器，继承ViewGroup类

|  |  |  |
| --- | --- | --- |
|  | android:gravity |  |
|  | android:ignoreGravity |  |
| RelativeLayout.LayoutParams | android:layout\_(above|toRightOf|below|toLeftOf)="**控件id**" |  |
| android:layout\_align(Top|Right|Bottom|Left)="**控件id**" |  |
| android:layout\_alignParent(Top|Right|Bottom|Left)="(false|true)" |  |
| android:layout\_center(InParent|Horizontal|Vertical)="(false|true)" |  |

LinearLayout线性布局管理器，继承ViewGroup类

|  |  |  |
| --- | --- | --- |
|  | android:orientation="(horizontal|vertical)" |  |
|  | android:gravity | top、right、bottom、left、(center|fill|clip)\_(horizontal|vertical)、center、fill用按位或组合。 |
| 子组件 | android:layout\_gravity | top、right、bottom、left、(center|fill|clip)\_(horizontal|vertical)、center、fill用按位或组合。 |
| android:layout\_weight | 默认为0，表示需要多大的空间就占多大的空间。  大于0时，按权重比将父容器剩余空间分割。 |

水平线性布局管理器中子组件的layout\_width设置为match\_parent，垂直线性布局管理器中子组件的layout\_height设置为match\_parent，都会导致只显示1个组件。

FrameLayout帧布局管理器，继承ViewGroup类

|  |  |  |
| --- | --- | --- |
|  | android:foreground="**资源**" |  |
|  | android:foregroundGravity | top、right、bottom、left、(center|fill|clip)\_(horizontal|vertical)、center、fill用按位或组合。 |
| 子组件 | android:layout\_gravity="center" |  |

前面的组件在下，后面的组件在上。

TableLayout表格布局管理器，继承LinearLayout类

<tableRow>每添加1个组件，表格增加1列。

直接向TableLayout中添加的组件独占1行。

|  |  |
| --- | --- |
| android:collapseColumns | 列号从0开始，用“,”分隔。 |
| android:shrinkColumns | 允许被收缩的列，直到表格匹配屏幕大小。  列号从0开始，用“,”分隔。 |
| android:stretchColumns | 允许被拉伸的列，用来填充屏幕空间。  列号从0开始，用“,”分隔。 |

AbsoluteLayout绝对布局管理器，继承ViewGroup类

已过时。

GridLayout网格布局管理器，继承ViewGroup类

|  |  |  |
| --- | --- | --- |
|  | android:orientation="(horizontal|vertical)" | 没有为子组件分配行和列时的排列方式。 |
|  | android:rowCount | 最大行数 |
|  | android:columnCount | 最大列数 |
|  | android:rowOrderPreserved="(false|true)" | 行边界显示的顺序是否和行索引的顺序相同。 |
|  | android:columnOrderPreserved="(false|true)" | 列边界显示的顺序是否和列索引的顺序相同。 |
|  | android:useDefaultMargins="(false|true)" | 是否使用默认的边距。 |
|  | android:alignmentMode | 对齐模式  alignBounds对齐边界  alignMargins对齐边距 |
| GridLayout.LayoutParams | android:layout\_gravity | top、right、bottom、left、(center|fill|clip)\_(horizontal|vertical)、center、fill、start、end用按位或组合。 |
| android:layout\_row | 从0开始。 |
| android:layout\_column | 从0开始。 |
| android:layout\_rowSpan | 纵向跨几行。 |
| android:layout\_columnSpan | 横向跨几列。 |
| android:layout\_columnWeight |  |

<android.support.constraint.ConstraintLayout>

|  |  |  |
| --- | --- | --- |
|  | xmlns:android="http://schemas.android.com/apk/res/android" | 声明命名空间。 |
|  | xmlns:app="http://schemas.android.com/apk/res-auto" |  |
|  | xmlns:tools="http://schemas.android.com/tools" | 指定布局的默认工具。 |
|  | tools:context="**包名**.**类名**" |  |
| 子组件 | app:layout\_constraintBottom\_to(Top|Right|Left|Bottom)Of="parent" |  |

android.widget.TextView类，View的子类

|  |  |
| --- | --- |
| android:text="**资源**" |  |
| android:textColor="**资源**" |  |
| android:textSize="**数值**" |  |
| android:autoLink | 将文本转换为可单击的超链接。  none、web、email、phone、map、all |
| android:singleLine="(false|true)" | 是否不换行。不换行时，超出部分显示成“...”。 |
| android:minLines="**数值**" |  |
| android:width="**数值**" | ！！！不能是match\_parent、wrap\_content。 |
| android:height="**数值**" |  |
| android:lines="**数值**" | 文本框高度为行高的整数倍。 |
| android:maxWidth="**数值**" |  |
| android:drawable(Top|Right|Bottom|Left|Start|End) |  |
| android:gravity | top、right、bottom、left、(center|fill|clip)\_(horizontal|vertical)、center、fill用按位或组合。 |
| android:inputType | 显示内容的文本类型。  textPassword、textEmailAddress、phone、date用按位或组合。 |
| android:hint="**资源**" | 文本框中内容为空时显示的文本。 |
| android:drawableLeft="**资源**" |  |
| android:drawablePadding="**数值**" |  |

EditText，TextView的子类

|  |  |
| --- | --- |
| String getText().toString() |  |

Button类，TextView的子类

|  |  |
| --- | --- |
| android:onClick="**方法名**" | import android.view.View.OnClickListener  ((Button)findViewById(R.id.**资源id**)).setOnClickListener(new (View.)?OnClickListener(){  @Override  public void onClick(View v){  }  }) |

ImageView类，View的子类

|  |  |
| --- | --- |
| android:src="**资源**" |  |
| android:adjustViewBounds="(false|true)" | 是否调整边界来保持图片的长宽比。 |
| android:maxWeight | 需要android:adjustViewBounds为true。 |
| android:maxHeight | 需要android:adjustViewBounds为true。 |
| android:scaleType | 缩放方式。  martrix、fitXY、fitStart、fitCenter、fitEnd、center、centerCrop、centerInside  fitCenter会放大图片，让图片尽量充满；centerInside不会放大图片。  <https://stackoverflow.com/questions/11353973/whats-the-difference-between-center-inside-and-fit-center-scale-types> |
| android:tint="**资源**" | 在图片上着一层颜色。颜色一般是透明的。 |

ImageButton类，ImageView的子类

|  |  |
| --- | --- |
| android:background | 若不设置，则自动拥有1个可以按下去的灰色按钮背景。  StateListDrawable资源 |

CompoundButton类，Button的子类

<RadioGroup>

|  |  |
| --- | --- |
| android:orientation="(horizontal|vertical)" |  |

子组件

|  |  |
| --- | --- |
| android:layout\_width="(match\_parent|wrap\_content|**数值**)" |  |
| android:layout\_height="(match\_parent|wrap\_content|**数值**)" |  |

<RadioButton>

RadioButton类，CompoundButton的子类

|  |  |
| --- | --- |
| android:checked="(false|true)" | ((RadioGroup)findViewById(R.id.**资源id**)).setOnCheckedChangedListener(new RadioGroup.OnCheckedChangeListener(){  @Override  pubic void onCheckedChanged(RadioGroup group,int checkedId){  ((RadioButton)findViewById(checkedId)).getText();  }  })  for(int i=0;i!=**RadioGroup对象**.getChildCount();++i)  {  ((RadioButton)**RadioGroup对象**.getChildAt(i)).isChecked();  } |
| android:text="**资源**" |  |

CheckBox类，CompoundButton的子类

|  |  |
| --- | --- |
| android:checked="(false|true)" | ((CheckBox)findViewById(R.id.**资源id**)).setOnCheckedChangedListener(new CompoundButton.OnCheckedChangeListener(){  @Override  pubic void onCheckedChanged(CompoundButton buttonView,boolean isChecked){  }  }) |

ProgressBar类，View的子类

|  |  |  |
| --- | --- | --- |
| android:max |  |  |
| android:progress | setProgress(int)  incrementProgressBy(int diff)进度增加或减少 |  |
| android:progressDrawable |  | 进度条轨道的绘制形式。  ?android:attr/progressBarStyleHorizontal细的长条水平进度条  ?android:attr/progressBarStyleLarge大圆形进度条  ?android:attr/progressBarStyleSmall小圆形进度条  @android:style/Widget.ProgressBar.Large大跳跃、旋转画面的进度条  @android:style/Widget.ProgressBar.Small小跳跃、旋转画面的进度条  @android:style/Widget.ProgressBar.Horizontal粗的长条水平进度条 |

AbsSeekBar类，ProgressBar的子类

SeekBar类，AbsSeekBar的子类

|  |  |
| --- | --- |
| android:thumb="**资源**" | 滑块的外观。 |

|  |
| --- |
| **SeekBar对象**.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener(){  @Override  public void onStartTrackingTouch(SeekBar seekBar){  }  @Override  public void onStopTrackingTouch(SeekBar seekBar){  }  @Override  public void onProgressChanged(SeekBar seekBar,int progress,boolean fromUser){  }  }) |

RatingBar类，AbsSeekBar的子类

|  |  |  |
| --- | --- | --- |
| android:isIndicator="(false|true)" |  | true，不允许用户改变。 |
| android:numStars |  | 星星总数。 |
| android:rating | float getRating() | 默认星级。 |
| android:stepSize | float getStepSize() | 默认值：0.5。 |
|  | int getProgress() | getRating()/getStepSize() |

AdapterView类，抽象类，ViewGroup的子类

可以包括多个列表项。指定多个列表项时，使用Adapter对象提供。

AbsListView类，AdapterView的子类

GridView类，AbsListView的子类

|  |  |
| --- | --- |
| android:horizontalSpacing | 元素之间水平间距。 |
| android:verticalSpacing | 元素之间垂直间距。 |
| android:gravity |  |
| android:numColumns="(auto\_fit|**数值**)" | 列数。若是1，则建议用ListView。 |
| android:columnWidth |  |
| android:stretchMode | none、spacingWidth拉伸元素间距、columnWidth拉伸元素、spacingWidthUniform拉伸元素间距和元素 |

**GridView对象**.setAdapter(**BaseAdapter对象**);

Adapter

ArrayAdapter

SimpleAdapter

SimpleCursorAdapter

BaseAdapter

android.os.Handler类

|  |
| --- |
| **Handler对象**=new Handler(){  @Override  public void handleMessage(Message msg){  msg.what==**整数**  }  }  **Message对象**.what=**整数**;  **Handler对象**.sendMessage(**Message对象**); |

|  |
| --- |
| Toast.makeText(MainActivity.this,String,Toast.(LENGTH\_SHORT|LENGTH\_LONG)).show(); |
| AlertDialog.Builder b=new AlertDialog.Builder(MainActivity.this);  b.setMessage(String);  b.setPositiveButton(String,null).show(); |

自定义View类

|  |
| --- |
| public **类名**(context)  super(context); |
| @Override  protected void onDraw(Canvas canvas)  super.onDraw(canvas) |
| 对象.setOnTouchListener(new View.OnTouchListener()  {    对象.invalidate();  return true;  }) |