**Android 6.0新控件**

**1. TextInputLayout的使用**

* 高级炫酷带有提示的输入框，相当于输入框中的战斗框
* 使用需要依赖design类库：

compile 'com.android.support:design:23.0.0+'

* 使用步骤：
  + 先在TextInputLayout中包裹一个EditText，如：
  + 然后给EditText添加文本变化监听器：
  + editText.addTextChangedListener(new TextWatcher() {
  + @Override
  + public void beforeTextChanged(CharSequence s, int start, int count, int after) {
  + }
  + @Override
  + public void onTextChanged(CharSequence s, int start, int before, int count) {
  + }
  + @Override
  + public void afterTextChanged(Editable s) {
  + if(s.length() > 5){
  + //设置错误提示信息
  + inputLayout.setError("不能超过5个");
  + //启用错误提示
  + inputLayout.setErrorEnabled(true);
  + }else{
  + //关闭错误提示
  + inputLayout.setErrorEnabled(false);
  + }
  + }

});

**2. FloatingActionButton的使用**

* 总是能悬浮在界面上的Button，可以设置点击事件
* 使用需要依赖design类库：

compile 'com.android.support:design:23.0.0+'

* 可以设置以下属性：
* app:fabSize="normal"//2个取值，normal=56dp,mini=48dp
* app:elevation="10dp"//高度
* app:rippleColor="#0000ff"//按下水波纹颜色
* app:pressedTranslationZ="20dp"//按下Z轴移动距离
* 注意：设置android:clickable="true"才有按下的效果

**3. Snackbar的使用**

* 一个介于Toast和Dialog之间的和用户交互的控件，显示在界面底部，并且可以设置点击行为，也可以滑动出去
* 在和FloatingActionButton一起使用时，需要让CoordinatorLayout作为父布局才能使FloatingActionButton给SnackBar腾出空间

**4. TabLayout的使用**

* 相当于ViewPagerIndicator的指示器效果，一般用来跟ViewPager结合使用
* 使用需要依赖design类库：

compile 'com.android.support:design:23.0.0+'

* 单独使用TabLayout的步骤：
  + 先添加Tab，使用tabLayout.newTab()方法创建Tab：
  + //1.添加Tab

tabLayout.addTab(tabLayout.newTab().setText("Tab1"));

* + 给tabLayout设置tab改变的监听器：
  + //2.给tabLayout添加tab改变的监听器
  + tabLayout.setOnTabSelectedListener(new TabLayout.OnTabSelectedListener() {
  + @Override
  + public void onTabSelected(TabLayout.Tab tab) {
  + Log.e("TAG", "onTabSelected: "+tab.getText());
  + }
  + @Override
  + public void onTabUnselected(TabLayout.Tab tab) {
  + }
  + @Override
  + public void onTabReselected(TabLayout.Tab tab) {
  + }

});

* + 在xml中给TabLayout设置属性：
  + app:tabIndicatorColor="#00f"//横线颜色
  + app:tabSelectedTextColor="#f00"//选中字体颜色
  + app:tabTextColor="#0f0"//默认字体颜色
  + app:tabMode="fixed"//fixed：不能滑动，每个Tab平分宽度，scrollable:可以滑动tab，每个tab宽度很小，适用于tab很多的情景
  + app:tabGravity="fill"//fill:平分宽度，center:让tab居中,如果tabMode是scrollable，那tabGravity会被忽略
* 和ViewPager关联使用步骤：
  + 先给ViewPager填充数据，然后关联TabLayout和ViewPager：
  + //给ViewPager填充数据
  + viewpager.setAdapter(new MyAdapter());
  + //关联TabLayout和ViewPager

tabLayout.setupWithViewPager(viewpager);

* + 需要注意的是，ViewPager的adapter的getPageTitle()方法的返回值将会设置给Tab的标题

**5. CoordinatorLayout的使用**

* 协调布局，能够协调多个布局的位置关系，可以实现让FAB上下滑动，展开或折叠ToolBar，控制View扩展收缩以及放大缩小
* 使用需要依赖design类库：

compile 'com.android.support:design:23.0.0+'

* 使用它结合AppBarLayout实现向上滑动隐藏ToolBar的效果：
  + AppBarLayout会将包裹的所有子View作为一个整体的AppBar，有着统一的界面着色;
  + app:layout\_scrollFlags属性介绍：
  + scroll:表示该View可以被滑动出CoordinatorLayout的范围，所有想滚动出屏幕的view都需要设置这个flag， 没有设置这个flag的view将被固定在屏幕顶部。例如，TabLayout 没有设置这个值，将会停留在屏幕顶部
  + enterAlways:表示任意向下的滚动都会导致该View可见
  + exitUntilCollapsed:滚动退出屏幕，最后折叠在顶端
  + enterAlwaysCollapsed:当你的视图已经设置minHeight属性又使用此标志时，你的视图只能以最小高度进入，只有当滚动视图到达顶部时才扩大到完整高度
  + 给想滑动出范围的View设置属性，比如ToolBar：

app:layout\_scrollFlags=”scroll|enterAlways

* + 给发出滑动行为的View设置属性，比如ViewPager：

app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"

* + 实现步骤：
  + 1.CoordinatorLayout作根控件，包裹AppBarLayout和可滚动的控件，比如ViewPager
  + 2.AppBarLayout包裹 ToolBar 及TabLayout，
  + ToolBar要滑动，给其设置app:layout\_scrollFlags
  + 3.ViewPager是发出滑动行为的控件，设置属性 app:layout\_behavior

注意：带layout\_scrollFlags的view需要放在固定View的前面，这样滑动的view才能够正常退出，而固定的view继续留在顶部