Android 5.0新添加了Z轴，使得组件在界面上显示有立体层次感，**注意**：只有子View比父View小，才能显示出子View的阴影效果，如果子View比父View大，则父View直接把子View截取一部分界面显示出来，无法显示出阴影效果，

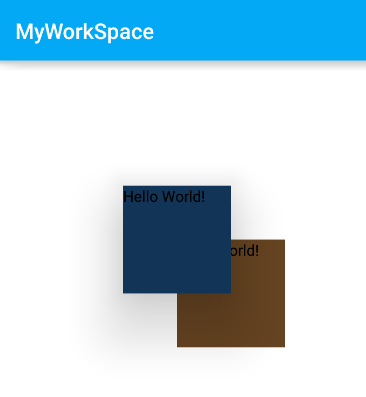
具体“参考文档”文件夹下的网页文档：

shadow.html文档

一、通过elevation属性设置控件与屏幕的距离

<TextView  
 android:layout\_width="100dp"  
 android:layout\_height="100dp"  
 android:layout\_margin="100dp"  
 android:text="Hello World!"  
 android:elevation="50dp"  
 android:background="#123456"  
 />  
<TextView  
 android:layout\_width="100dp"  
 android:layout\_height="100dp"  
 android:layout\_margin="150dp"  
 android:text="Hello World!"  
 android:background="#654321"  
 />

为第一个TextView设置距离屏幕的距离为50dp，第二个TextView不设置该属性，默认为0，下面是显示的效果：



该布局选取的是帧布局，如果没有设置elevation属性的话，第二个TextView应该显示在第一个TextView之上，但设置了elevation属性，第一个距离屏幕的距离为50dp，所以第一个TextView显示在第二个TextView的上面，且第一个TextView四周有背景阴影

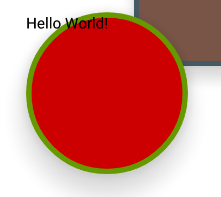
二、通过剪裁设置控件阴影的形状

在Android的世界里，所有的View都是矩形的，虽然可以给View设置背景圆形的图片，即可在界面显示出圆形的内容，但是View的大小实际上仍然是矩形，并且设置的图片也是实际上也是矩形，只是圆形以外的区域为透明色。

控件的原始形状都是矩形，如果是矩形，则阴影效果也是矩形，如果背景直接用颜色设置，则可以显示出阴影：



如果是通过shape的xml文件绘制的形状，如圆形，则阴影效果也是圆形的（根据该形状的轮库判断）：



为圆形的背景设置阴影效果：

如果背景只是设置一张普通的背景图片，则无法获得该view的轮廓，所以不会显示阴影效果：

内容是圆形，但实际上是矩形的，周边为透明色，无法显示出阴影效果：



在手机上设置了elevation属性后并没有显示出阴影的效果： 

如果想要上面的图片显示出实际形状（如上图是圆形）的阴影效果，则只能通过代码动态设置：

iv = (ImageView) findViewById(R.id.*iv*);  
ViewOutlineProvider provider = new ViewOutlineProvider() {  
 @Override  
 public void getOutline(View view, Outline outline) {

//获取轮廓

outline.setOval(0,0,view.getWidth(),view.getHeight());  
 }  
};  
iv.setOutlineProvider(provider);

通过代码为控件设置轮廓后的效果：



裁剪圆形：

ViewOutlineProvider provider = new ViewOutlineProvider() {  
 @Override  
 public void getOutline(View view, Outline outline) {  
 //获取圆形轮廓 outline.setOval(0,0,view.getWidth(),view.getHeight());  
 }  
 };  
 tv1.setOutlineProvider(provider);  
// 裁剪成圆形轮廓  
 tv1.setClipToOutline(true);

裁剪的圆形：



裁剪圆角矩形：

ViewOutlineProvider provider = new ViewOutlineProvider() {  
 @Override  
 public void getOutline(View view, Outline outline) {  
 //获取圆角矩形轮廓 outline.setRoundRect(0,0,view.getWidth(),view.getHeight(),25);  
 }  
 };  
 tv2.setOutlineProvider(provider);  
// 裁剪成圆形轮廓  
 tv2.setClipToOutline(true);

裁剪的圆角矩形：

