

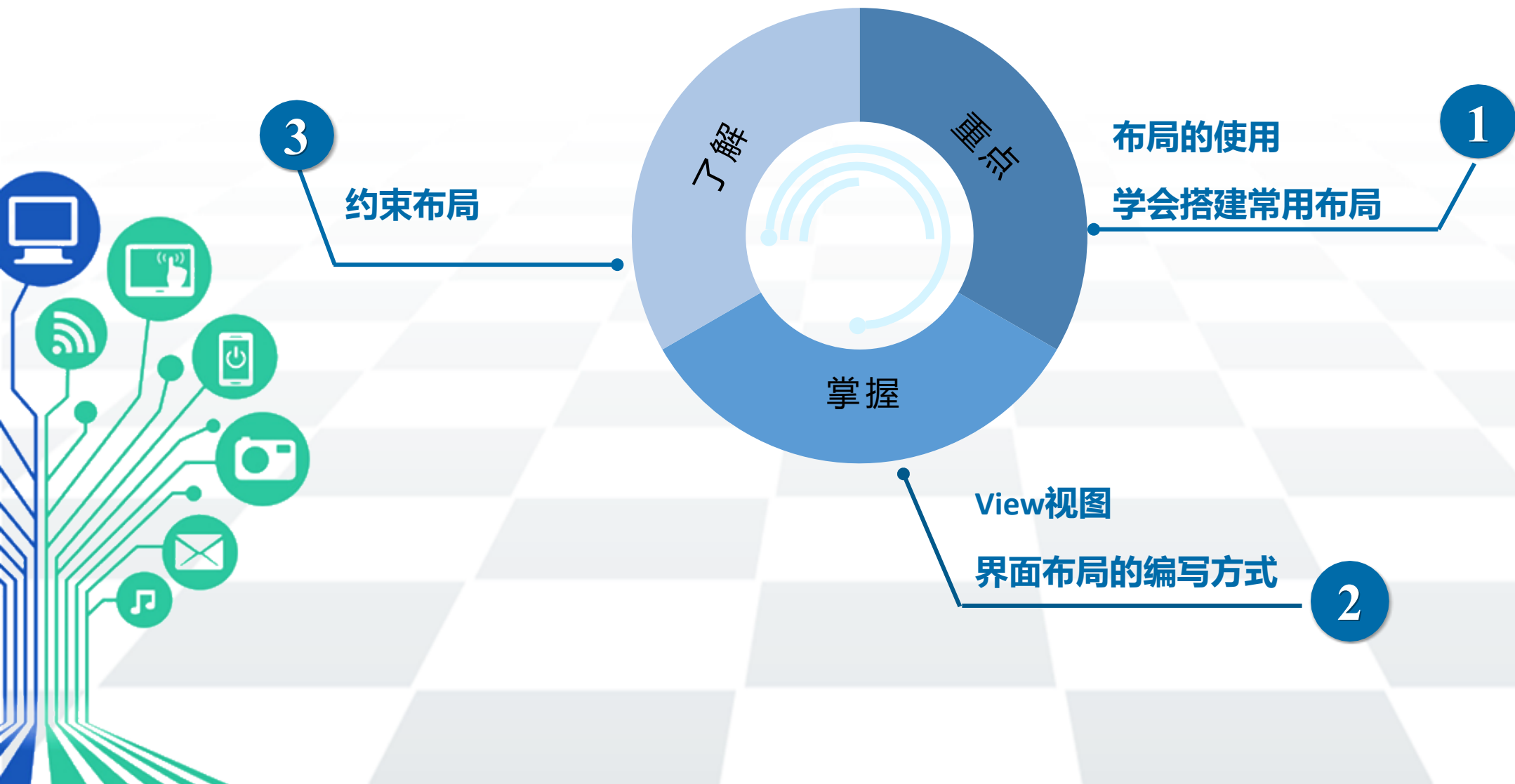


第2章Android常见界面布局



- 2.1 布局的创建
- 2.2 布局类型

学习目标



主讲内容

2.1 布局的创建

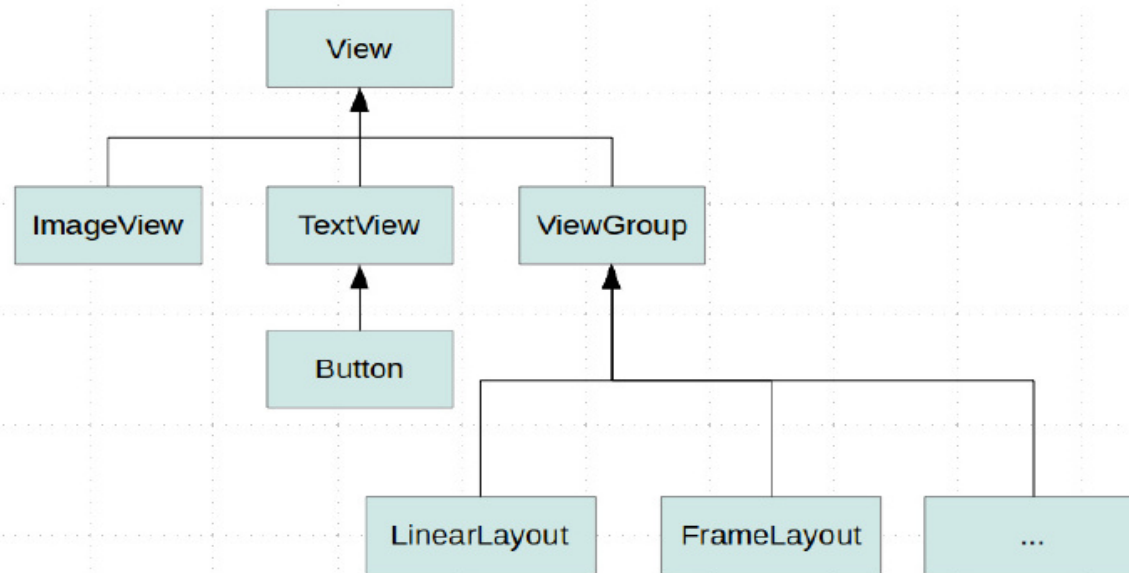
主讲内容

Speech content

2.1 布局的创建

View视图

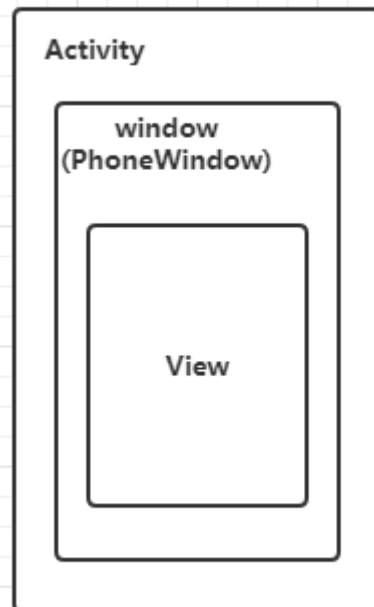
- 所有的UI元素都是通过View与ViewGroup构建的，对于一个Android应用的用户界面来说，ViewGroup作为容器盛装界面中的控件，它可以包含普通的View控件，也可以包含ViewGroup。



2.1 布局的创建

View视图

- Activity与View直接的关系



主讲内容

2.2 布局类型

2.2.1 LinearLayout 线性布局

2.2.2 RelativeLayout 相对布局

2.2.3 TableLayout 表格布局

2.2.4 FrameLayout 帧布局

2.2.5 ConstraintLayout 布局

主讲内容

Speech content

2.2.1 LinearLayout线性布局

界面布局编写方式

– 界面布局编写方式

➤ 在XML文件中编写布局：推荐此种方式布局

◆ 有效的将界面中布局的代码和Java代码隔离，使程序的结构更加清晰。

➤ 在Java代码中编写布局

◆ 在Android中所有布局和控件的对象都可以通过new关键字创建出来，将创建的View控件添加到ViewGroup布局中，从而实现View控件在布局界面中显示。

2.2.1 LinearLayout线性布局

相对布局继承自ViewGroup

在XML文件中编写布局

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android=""
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="使用XML布局文件控制UI界面"
        android:textColor="#ff0000"
        android:textSize="18sp"
        android:layout_centerInParent="true"/>
</RelativeLayout>
```

TextView控件继承自View

设置文字的样式

My Application

使用XML布局文件控制UI界面

2.2.1 LinearLayout线性布局

在Java代码中编写布局

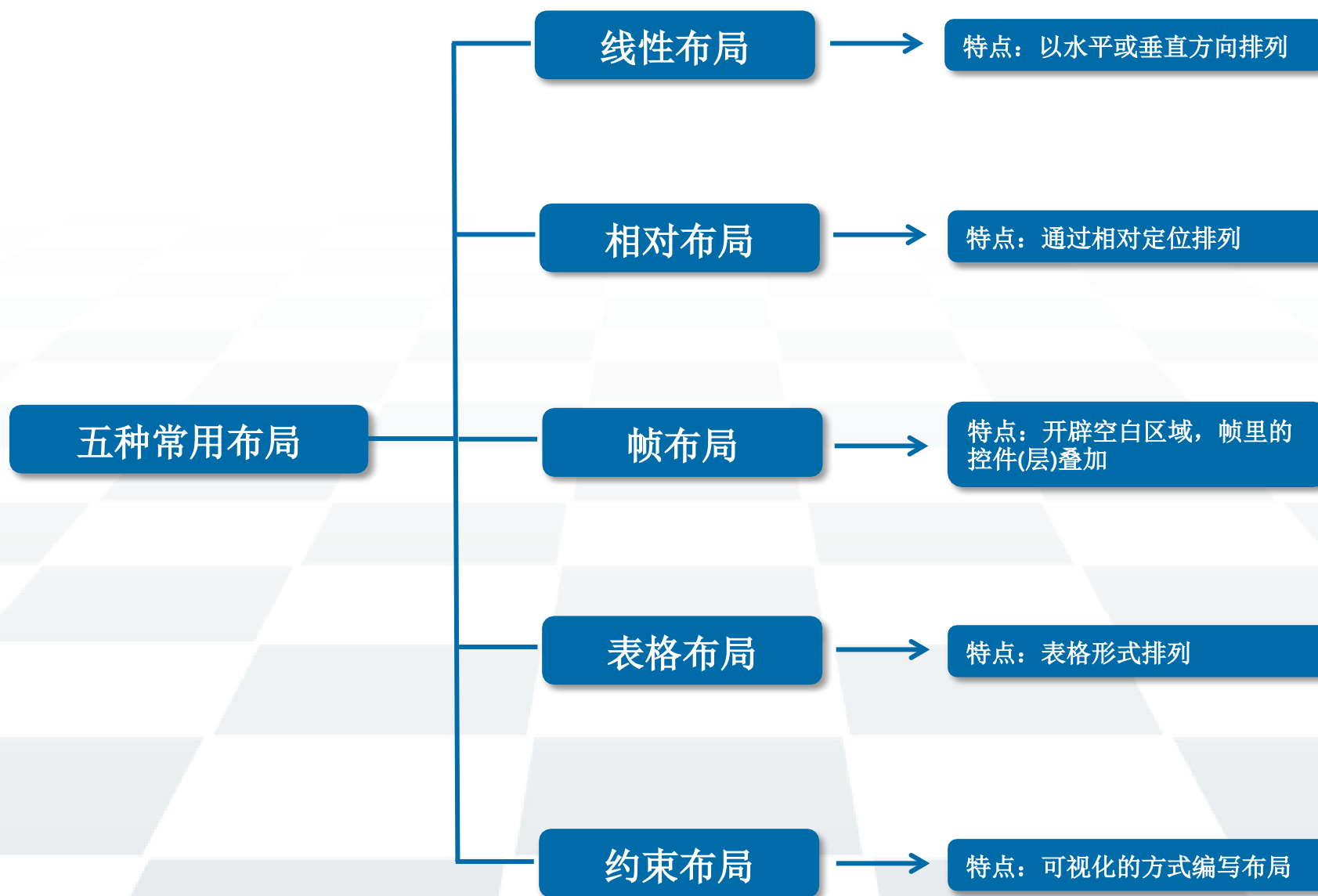
```
RelativeLayout relativeLayout = new RelativeLayout(this);
RelativeLayout.LayoutParams layoutParams = new RelativeLayout.LayoutParams(
    RelativeLayout.LayoutParams.WRAP_CONTENT,
    RelativeLayout.LayoutParams.WRAP_CONTENT);

//addRule参数对应RelativeLayout.LayoutParams.addRule(RelativeLayout.ALIGN_CENTER_HORIZONTAL); //设置居中显示
//addRule参数对应RelativeLayout.LayoutParams.addRule(RelativeLayout.ALIGN_CENTER_VERTICAL); //设置居中显示
TextView textView = new TextView(this); //w控件
textView.setText("Java"); //TextView的文字内容
textView.setTextColor(Color.RED); //TextView的文字颜色
textView.setTextSize(18); //TextView的文字大小
//添加TextView对象和RelativeLayout.addView(textView);
setContentView(relativeLayout); //设置由RelativeLayout
```

My Application

Java 代码实现界面布局

2.2.1 LinearLayout线性布局



2.2.1 LinearLayout线性布局

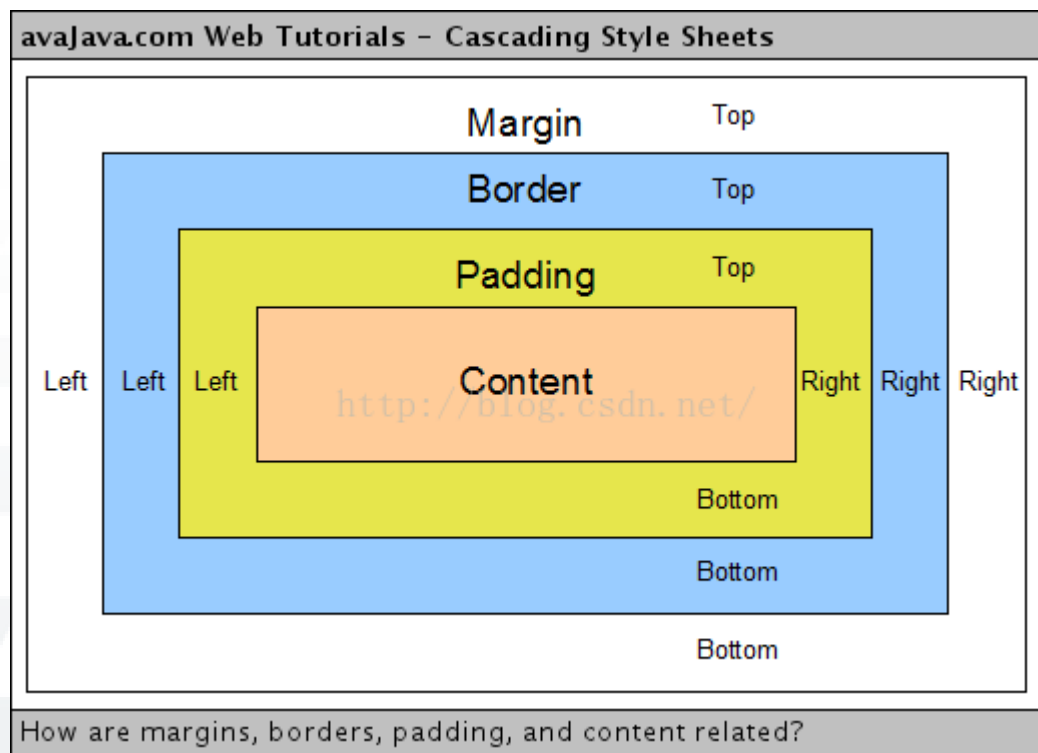
布局的通用属性

- Android系统提供的五种常用布局直接或者间接继承自ViewGroup，因此它们也支持在ViewGroup中定义的属性，这些属性可以看作是布局的通用属性。

属性名称	功能描述
android:id	设置布局的标识
android:layout_width	设置布局的宽度
android:layout_height	设置布局的高度
android:background	设置布局的背景
android:layout_margin	设置当前布局与屏幕边界或与周围控件的距离
android:padding	设置当前布局与该布局中控件的距离

2.2.1 LinearLayout线性布局

Padding、margin



2.2.1 LinearLayout线性布局

多学一招：Android支持的尺寸单位

- 尺寸单位
 - ✓ px (pixels, 像素): 每个px对应屏幕上的一个点。
 - ✓ dp (Density-independent Pixels, 设备独立像素): 是一种与屏幕密度无关的尺寸单位。
 - ✓ sp (scaled pixels, 比例像素): 主要处理字体的大小, 可以根据用户字体大小首选项进行缩放。
 - ✓ in (inches, 英寸): 标准长度单位。
 - ✓ pt (points, 磅): 屏幕物理长度单位, 1磅为1/72英寸。
 - ✓ mm (Millimeters, 毫米): 屏幕物理长度单位。

2.2.2 RelativeLayout相对布局

相对布局

- 相对布局（RelativeLayout）是通过相对定位的方式指定子控件位置，即以其它控件或父容器为参照物，摆放控件位置。
- 定义格式

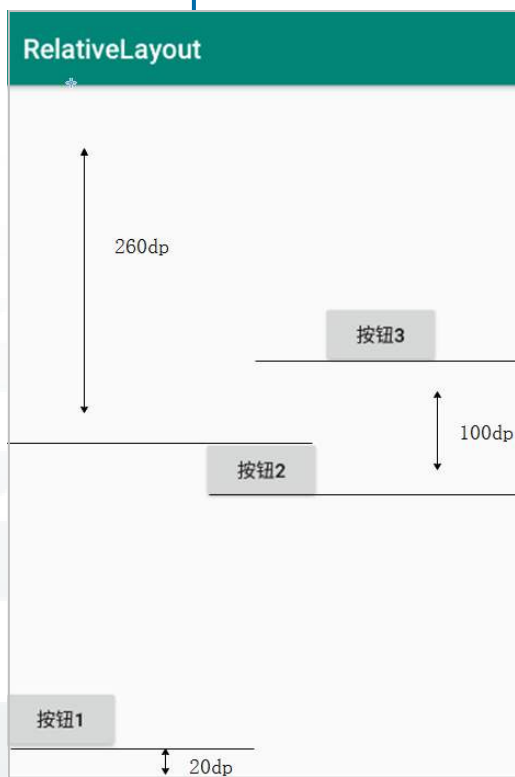
```
<RelativeLayout xmlns:android=""  
    属性 = "属性值"  
    .....>  
</RelativeLayout>
```

2.2.2 相对布局—控件位置属性

控件属性	功能描述
android:layout_centerInParent	设置当前控件位于父布局的中央位置
android:layout_centerVertical	设置当前控件位于父布局的垂直居中位置
android:layout_centerHorizontal	设置当前控件位于父控件的水平居中位置
android:layout_above	设置当前控件位于某控件上方
android:layout_below	设置当前控件位于某控件下方
android:layout_toLeftOf	设置当前控件位于某控件左侧
android:layout_toRightOf	设置当前控件位于某控件右侧
android:layout_alignParentTop	设置当前控件是否与父控件顶端对齐
android:layout_alignParentLeft	设置当前控件是否与父控件左对齐
android:layout_alignParentRight	设置当前控件是否与父控件右对齐
android:layout_alignParentBottom	设置当前控件是否与父控件底端对齐
android:layout_alignTop	设置当前控件的上边界与某控件的上边界对齐
android:layout_alignBottom	设置当前控件的下边界与某控件的下边界对齐
android:layout_alignLeft	设置当前控件的左边界与某控件的左边界对齐
android:layout_alignRight	设置当前控件的右边界与某控件的右边界对齐

2.2.2 RelativeLayout相对布局

相对布局



```
RelativeLayout xmlns:android=""  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">  
        <Button  
            android:layout_width="wrap_content"  
            android:layout_height="wrap_content"  
            android:text="按钮1" />  
        <Button  
            android:layout_width="wrap_content"  
            android:layout_height="wrap_content"  
            android:text="按钮2" />  
        <Button  
            android:layout_width="wrap_content"  
            android:layout_height="wrap_content"  
            android:text="按钮3" />  
    </RelativeLayout>
```


以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：
<https://d.book118.com/975302044342011222>