

重庆工信职业学院
2026 年合同制工作人员招聘
软件技术教师试讲内容

试讲题目：相对布局（RelativeLayout）

主要内容：

1. RelativeLayout 的基本概念与特点；
2. 相对于父容器的属性；
3. 相对于兄弟视图的属性；
4. 典型案例，如登录页面的布局实现等。

试讲要求：

1. 时间严格控制在 10 分钟内，建议准备 PPT 或板书提纲；
2. 提前提交书面教案（含教学目标、重难点、教学过程设计）；
3. 知识点讲解准确，重难点突出。

参考资料：

1. 《Android 移动应用开发（第 3 版）》 高等教育出版社 陈美莲
2. 《Android 应用开发案例教程（第 2 版）》 人民邮电出版社 张明

```
        android:layout_height="match_parent"
        android:gravity="center"
        android:orientation="vertical">
        <TextView
            android:id="@+id/textView3"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:background="#F44336"
            android:text="Hello World" />
        <TextView
            android:id="@+id/textView2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:background="#00BCD4"
            android:text="Hello World" />
    </LinearLayout>
```

运行效果如图 3.5 所示。



图 3.5 LinearLayout 居中显示的运行效果



扫码观看
微课视频

3.3.2 RelativeLayout

RelativeLayout 又称相对布局，也是一种比较常用的布局。它可以通过相对定位的方式让控件出现在布局的任何位置。在实际开发过程中，建议使用 RelativeLayout 来进行 UI 设计，因为使用 RelativeLayout 可以减少 UI 中的嵌套结构，从代码维护及运行效率上来说，具备一定的优势。

使用 RelativeLayout 前，需要将其从控件列表中拖入布局文件，如图 3.6 所示。

接下来，我们在 RelativeLayout 中加入两个按钮，供参考的代码如下：

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <Button
        android:id="@+id/btn1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="第一个按钮" />
    <Button
        android:id="@+id/btn2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="第二个按钮" />
</RelativeLayout>
```

运行效果如图 3.7 所示。

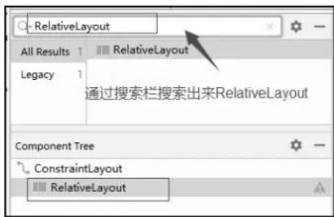


图 3.6 在控件面板搜索 RelativeLayout



图 3.7 RelativeLayout 的运行效果

通过上面的预览效果，我们看到两个按钮是叠放在一起的，这是因为我们还未设置控件的相关摆放属性。RelativeLayout 中的属性非常多，不过这些属性都有规律可循，并不难理解和记忆。下面我们来介绍 RelativeLayout 中的属性，这些属性主要分为 3 类。

- 第 1 类：属性值为 true 或者 false。
- 第 2 类：属性值必须为 ID 的引用名 “@id/id-name”。
- 第 3 类：属性值为具体的像素值，如 30dip、40dpi；

RelativeLayout 的第一类属性，如表 3.2 所示。

表 3.2 第一类属性

属性名称	描述
layout_centerHorizontal	水平居中
layout_centerVertical	垂直居中
layout_centerInparent	相对于父控件完全居中
layout_alignParentBottom	贴紧父控件的下边缘
layout_alignParentLeft	贴紧父控件的左边缘
layout_alignParentRight	贴紧父控件的右边缘
layout_alignParentTop	贴紧父控件的上边缘
layout_alignWithParentIfMissing	如果找不到对应的兄弟控件，就以父控件作为参照物

RelativeLayout 的第二类属性，如表 3.3 所示。

表 3.3 第二类属性

属性名称	描述
layout_below	在某元素的下方
layout_above	在某元素的上方
layout_toLeftOf	在某元素的左边
layout_toRightOf	在某元素的右边
layout_alignTop	本元素的上边缘和某元素的上边缘对齐
layout_alignLeft	本元素的左边缘和某元素的左边缘对齐
layout_alignBottom	本元素的下边缘和某元素的下边缘对齐
layout_alignRight	本元素的右边缘和某元素的右边缘对齐

RelativeLayout 的第三类属性，如表 3.4 所示。

表 3.4 第三类属性

属性名称	描述
layout_marginBottom	离某元素底边缘的距离
layout_marginLeft	离某元素左边缘的距离
layout_marginRight	离某元素右边缘的距离
layout_marginTop	离某元素上边缘的距离

以上 3 个表格所列的属性就是使用 RelativeLayout 进行 UI 设计时常用的属性，我们可以通过设置控件的这些属性，将控件合理地摆放在 UI 中。

下面我们通过一个例子展示将 5 个按钮分布在 UI 的 4 个角及中心的 UI 设计。供参考的代码如下：

```
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <Button
        android:id="@+id/btn1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:text="左上角" />
    <Button
        android:id="@+id/btn2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:text="右上角" />
    <Button
        android:id="@+id/btn3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:text="中间" />
    <Button
        android:id="@+id/btn4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:text="左下角" />
    <Button
        android:id="@+id/btn5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:text="右下角" />
</RelativeLayout>
```

运行效果如图 3.8 所示。



图 3.8 RelativeLayout 的运行效果



3.3.3 案例 1: Android 操作系统介绍 App

通过对上面两个布局的学习，我们大致掌握了这两个布局的基本用法。现在我们使用它们来开发一个项目：Android 操作系统介绍 App。

1. 需求描述

在本案例中，我们制作一个 Android 操作系统的介绍界面，主要使用 RelativeLayout 进行 UI 设计，其运行效果如图 3.9 所示。

2. 资源添加

在 UI 中，要显示相关的介绍文字，可以在字符串资源中将文字信息添加到资源文件 strings.xml (Android 的字符串资源文件)。在控件需要使用到相关文字时，引用对应的字符串资源，便可以在界面显示文字信息。



图 3.9 Android 操作系统介绍 App 运行效果

Android 工程的字符串资源文件 strings.xml 的配置代码如下：

```
<resources>
    <string name="app_name">Android 介绍</string>
    <string name="DetailTitme">什么是 Android 操作系统</string>
    <string name="DetailDate">2020-03-27</string>
    <string name="DetailContent">
        Android 是由谷歌公司主导开发的基于 Linux 的自由及开放源代码的操作系统。
    </string>
    <string name="BtnReply">回复</string>
    <string name="BtnRew">转发</string>
</resources>
```

3. 布局设计

在进行 UI 设计时，我们使用 RelativeLayout 进行布局。其中涉及的 TextView、Button、

Android 移动应用开发案例教程 (慕课版)

ImageView 控件，我们会在后文详细讲解。

UI 中包含了图标、标题、时间、内容及两个按钮。通过设置它们的属性进行布局，Android 操作系统介绍 App 的 UI 布局如图 3.10 所示。



图 3.10 Android 操作系统介绍 App 的 UI 布局

XML 布局文件对应的代码如下：

```
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <ImageView
        android:id="@+id/imgLogo"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_marginStart="12dp"
        android:layout_marginTop="14dp"
        app:srcCompat="@mipmap/ic_launcher" />
    <TextView
        android:id="@+id/txtTitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignTop="@+id/imgLogo"
        android:layout_marginStart="84dp"
        android:text="@string/DetailTitle" />
    <TextView
        android:id="@+id/txtDate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignTop="@+id/imgLogo"
        android:layout_marginEnd="25dp"
        android:text="@string/DetailDate" />
    <TextView
        android:id="@+id/txtContent"
        android:layout_width="275dp"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_alignStart="@+id/txtTitle"
        android:layout_marginTop="48dp"
```

```
        android:text="@string/DetailContent" />
    <Button
        android:id="@+id/btnReply"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_alignStart="@+id/txtTitle"
        android:layout_marginTop="130dp"
        android:text="@string/BtnReply" />
    <Button
        android:id="@+id/BtnFoward"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignTop="@+id/btnReply"
        android:layout_marginEnd="40dp"
        android:text="@string/BtnRew" />
</RelativeLayout>
```

4. 运行效果

完成 Android 操作系统介绍 App 的 UI 布局设计后，就可以在模拟器或者手机上运行并查看效果。其运行效果如图 3.11 所示。



图 3.11 Android 操作系统介绍 App 运行效果