

● RelativeLayout: 화면에 이미 배치된 뷰를 기준으로 다른 뷰의 위치를 지정하는 레이아웃. RelativeLayout에 뷰를 포함한 후 다른 뷰를 포함하면, 이전에 뷰 위에 덮어쓰듯이 올라가게 된다. RelativeLayout은 뷰의 상대 위치를 지정하여 배치한다.

▶ 상대 위치를 지정하는 속성: 속성값은 기존 뷰의 id이다. (ex: @id/text1)

android:layout_above	기존 뷰의 윗부분에 배치
android:layout_below	기존 뷰의 아랫부분에 배치
android:layout_toLeftOf	기존 뷰의 왼쪽에 배치
android:layout_toRightOf	기존 뷰의 오른쪽에 배치

▶ align 속성: 기준이 되는 뷰와 변을 맞추는 속성. 대부분 문자열 하단을 맞추는 게 일반적이다. 속성값은 기준이 되는 뷰의 id값이다. (ex: @id/text1)

android:layout_alignTop	기존 뷰와 윗부분을 정렬
android:layout_alignBottom	기존 뷰와 아랫부분을 정렬
android:layout_alignLeft	기존 뷰와 왼쪽을 정렬
android:layout_alignRight	기존 뷰와 오른쪽을 정렬
android:layout_alignBaseline	기존 뷰와 텍스트 기준선을 정렬

▶ alignParentXXX 속성: 특정 뷰를 RelativeLayout 영역의 상하좌우로 밀 수 있는 속성. 속성값은 true/false이다.

android:layout_alignParentTop	부모의 윗부분에 뷰의 상단을 위치
android:layout_alignParentBottom	부모의 아랫부분에 뷰의 하단을 위치
android:layout_alignParentLeft	부모의 왼쪽에 뷰의 왼쪽을 위치
android:layout_alignParentRight	부모의 오른쪽에 뷰의 오른쪽을 위치
android:layout_centerHorizontal	부모의 가로 방향 중앙에 뷰를 위치
android:layout_centerVertical	부모의 세로 방향 중앙에 뷰를 위치
android:layout_centerInParent	부모의 가로세로 중앙에 뷰를 위치

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="10dp">

    <androidx.constraintlayout.utils.widget.ImageFilterView
        android:src="@drawable/profile"
        android:layout_width="70dp"
        android:layout_height="70dp"
        android:adjustViewBounds="true"
        android:id="@+id/image"
    />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="홍길동"
        android:textSize="20sp"
        android:textStyle="bold"
        android:layout_toRightOf="@id/image"
        android:id="@+id/name"
        android:layout_marginLeft="10dp"
        android:layout_marginBottom="14dp"
    />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="안녕하세요. 잘 지내시는지요?"
        android:layout_below="@id/name"
        android:layout_alignLeft="@id/name"
        android:id="@+id/text"
    />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="22. 05. 20"
        android:layout_alignParentRight="true"
        android:layout_alignBaseline="@id/name"
        android:id="@+id/date"
    />

</RelativeLayout>

```

→ 만약 LinearLayout을 사용하는 것보다 RelativeLayout을 사용하는 것이 레이아웃 중첩이 적어서 전체 생성되는 뷰 객체가 적다면, 성능을 위해 RelativeLayout을 사용하는 것이 더 좋다. 객체 생성은 공짜가 아니므로, 뷰 객체 생성에 따른 성능에 신경을 써야 한다. 객체 생성을 줄일 수 있다면 줄여 주는 게 애플리케이션의 성능 향상에 도움이 된다.