

레이아웃의 구성요소 VIEW와 VIEWGROUP

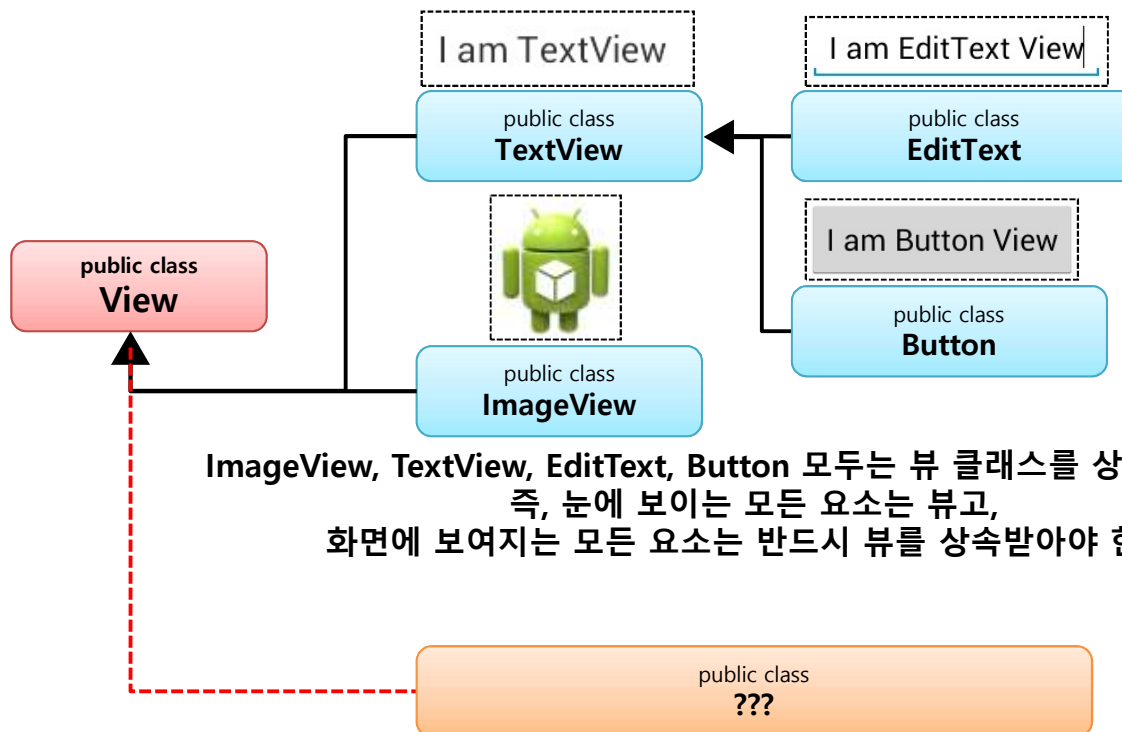


2

뷰와 뷰그룹

뷰에 대해서

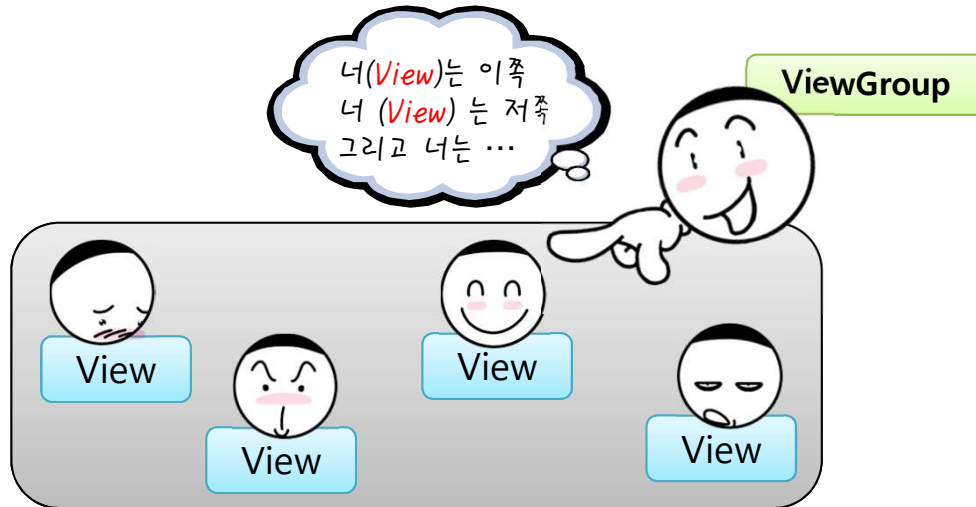
3



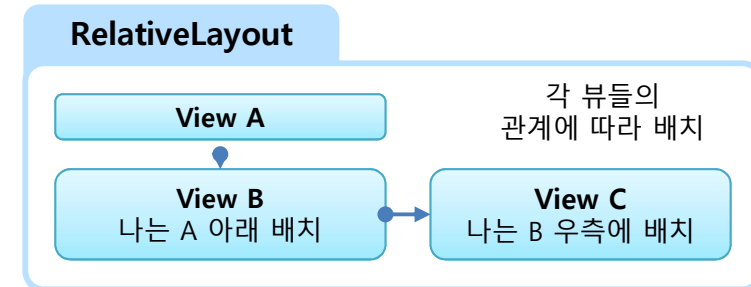
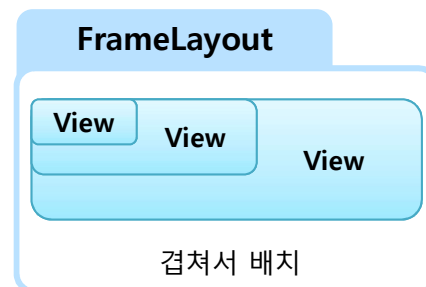
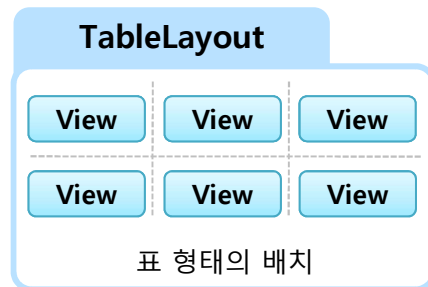
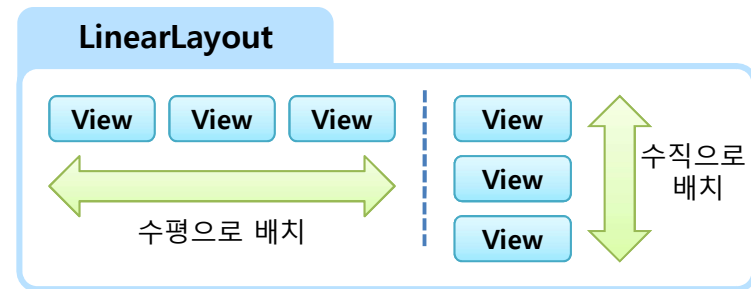
ImageView, TextView, EditText, Button 모두는 뷰 클래스를 상속받고 있다.
즉, 눈에 보이는 모든 요소는 뷰고,
화면에 보여지는 모든 요소는 반드시 뷰를 상속받아야 한다.

뷰그룹에 대해서

4

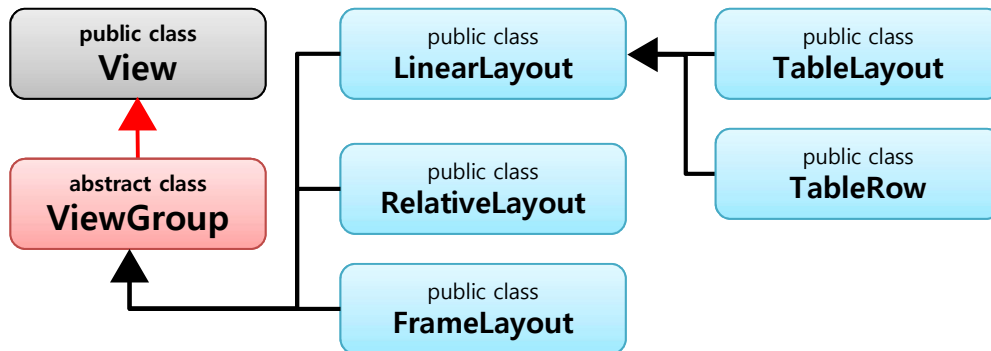


뷰의 배치를 담당하는 뷰그룹



뷰그룹에 대해서

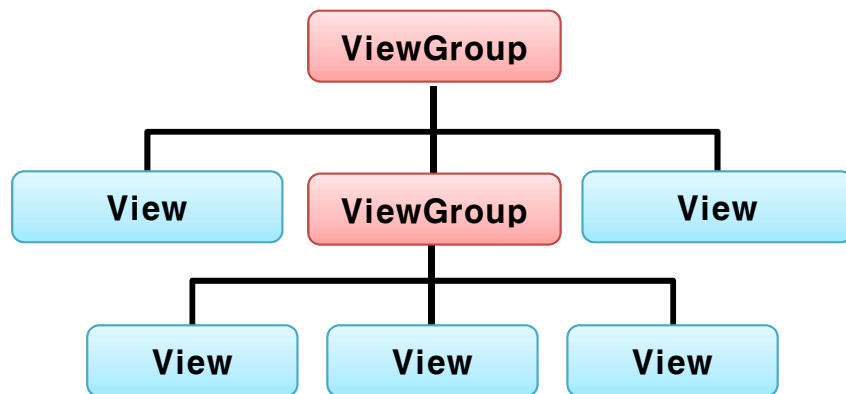
5



결국 뷰그룹도 뷰다.

결국 안드로이드에서의 화면은 오직 뷰만으로 구성되어 있다.

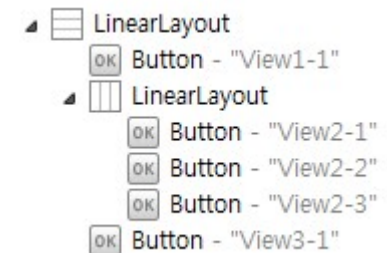
■ 왜 뷰그룹이 뷰를 상속받아야 할까?



뷰그룹 자체도 부모 뷰그룹에 의해 배치되어야 하기 때문이다.



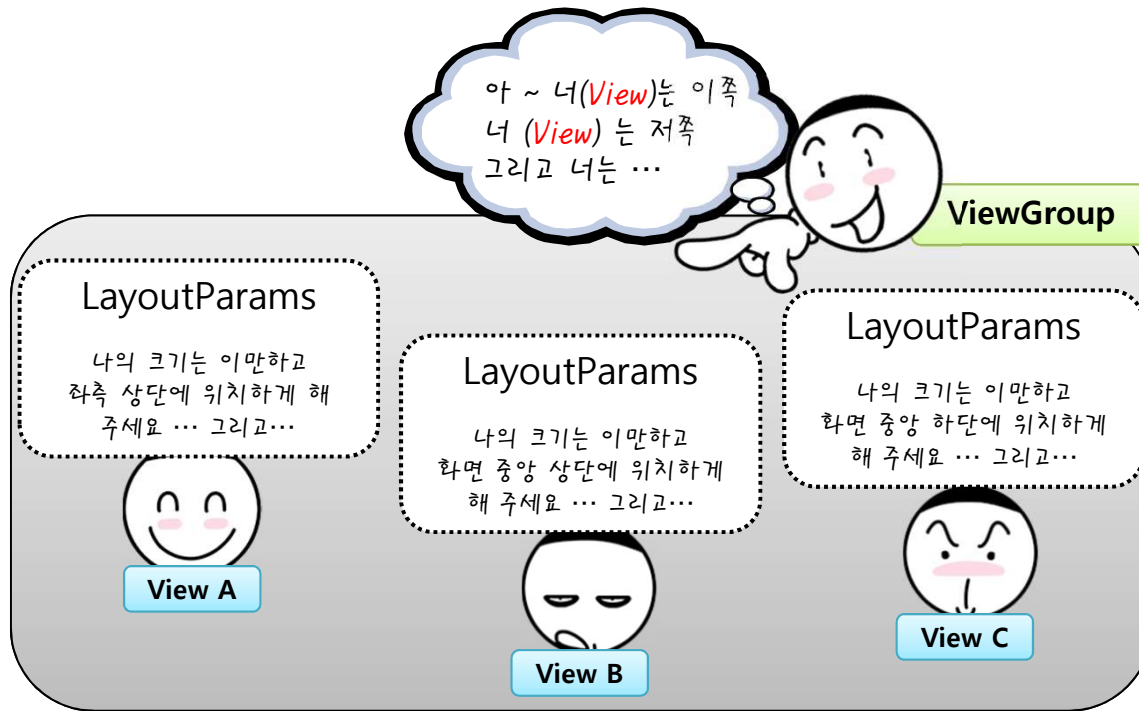
● 레이아웃 구조



6

화면 배치 정보 LayoutParams

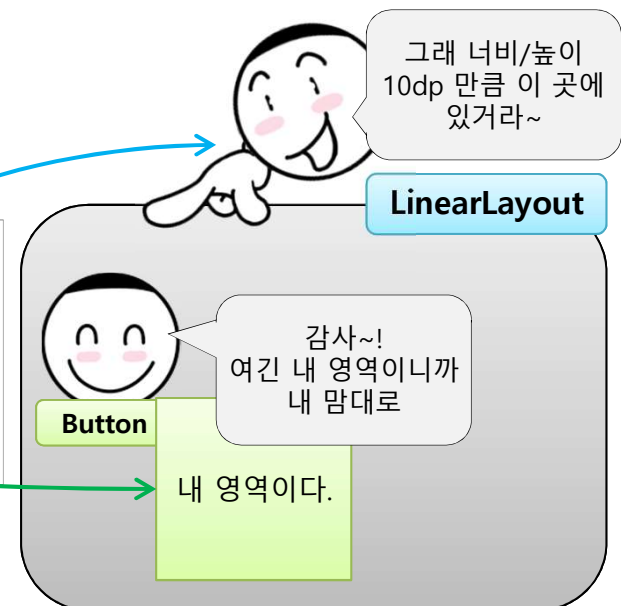
화면 배치정보 LayoutParams



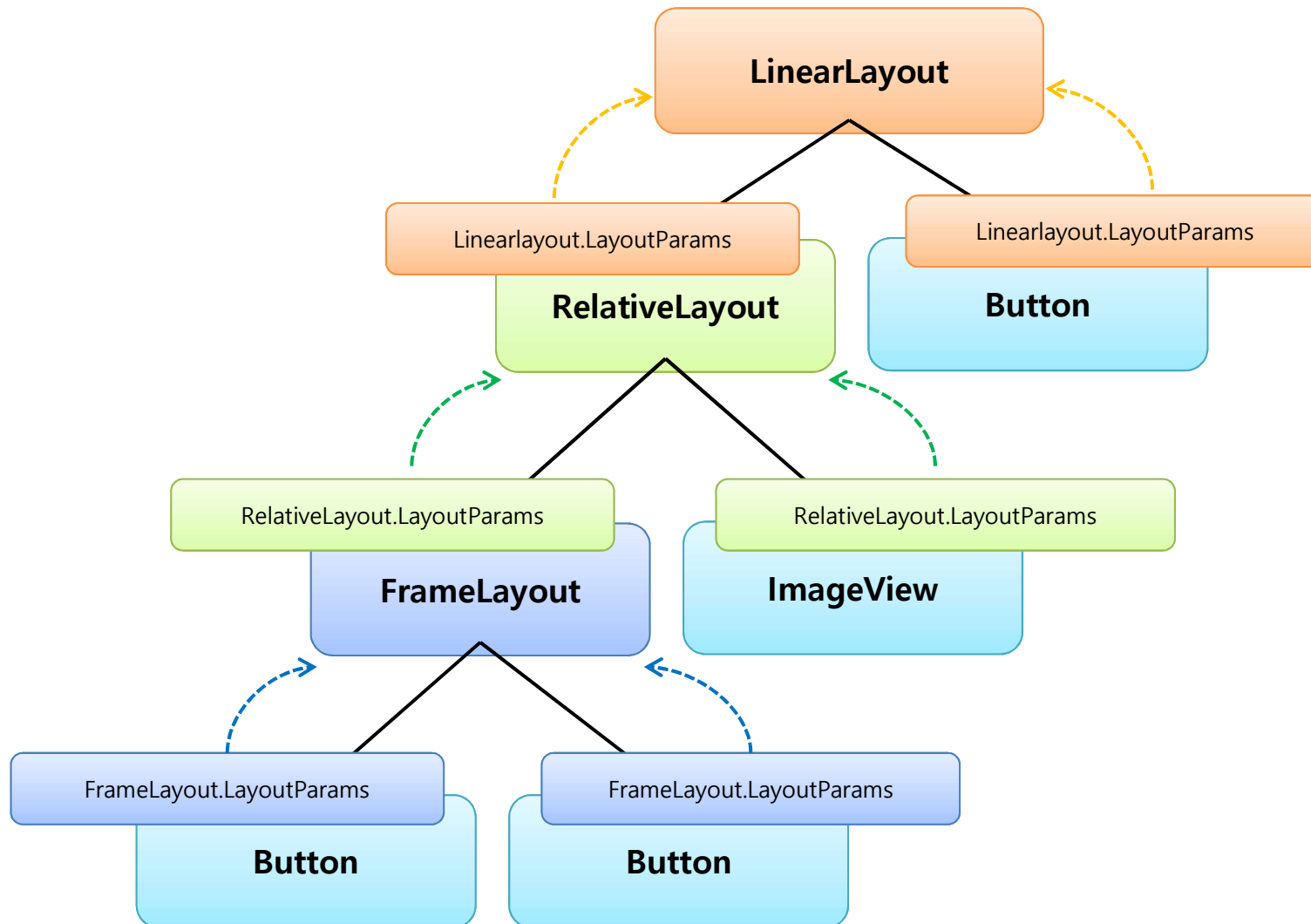
● 레이아웃 XML 내용

```
<LinearLayout>  
  <Button  
    android:layout_width="10dp"  
    android:layout_height="10dp"  
    android:text="내 영역이다." />  
</LinearLayout>
```

아빠~!
나에게 요만한
영역을 주삼.

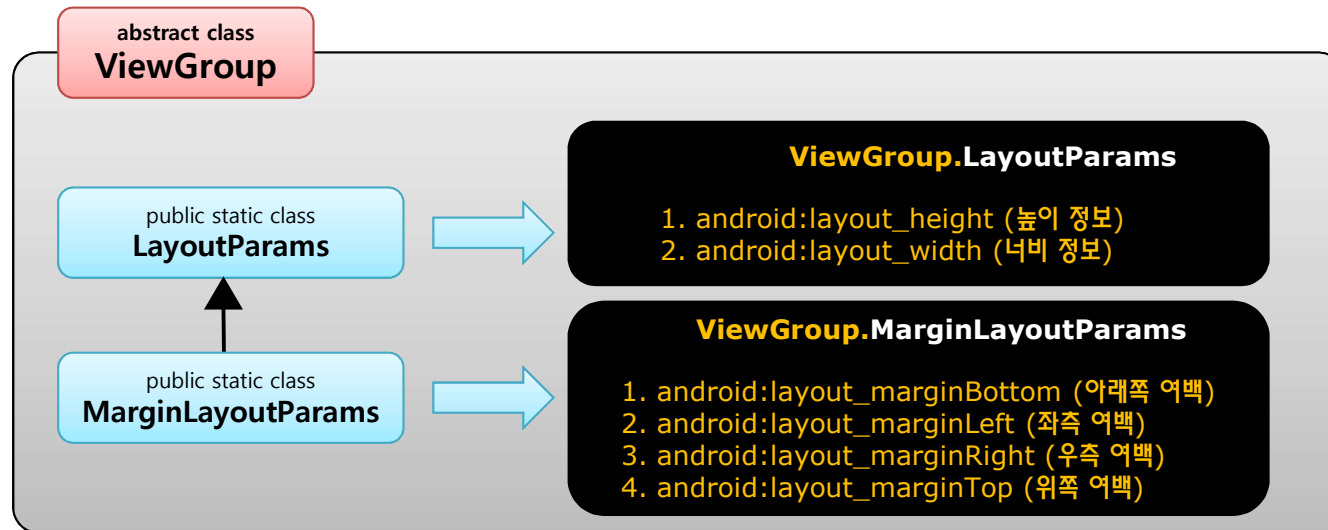


뷰가 가지는 다양한 LayoutParams 정보



최상위 뷰그룹의 LayoutParams 정보

9



최상위 뷰그룹의 LayoutParams 정보

res/layout/activity_main.xml

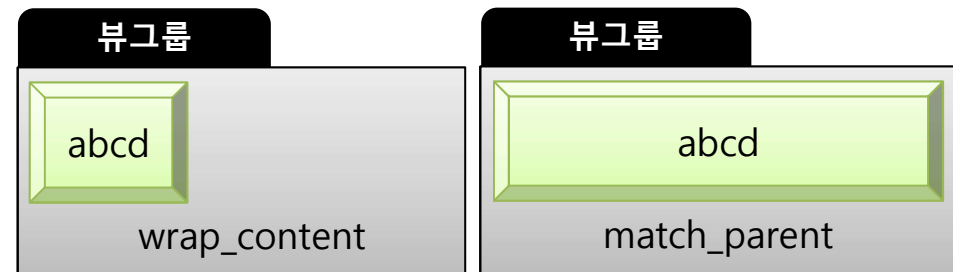
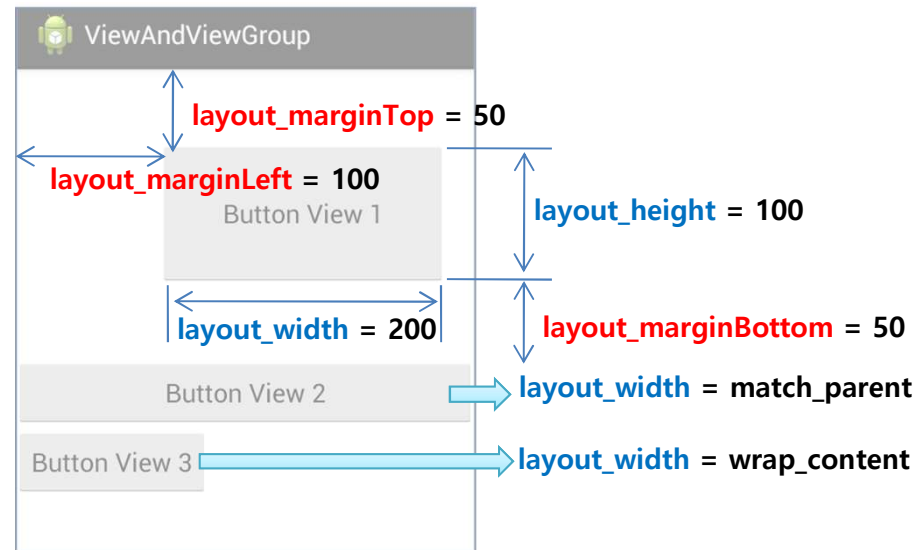
```
<LinearLayout xmlns:android="..."
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <Button
        android:layout_width="200dp"
        android:layout_height="100dp"
        android:layout_marginTop="50dp"
        android:layout_marginLeft="100dp"
        android:layout_marginBottom="50dp"
        android:text="Button View 1" />

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Button View 2" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button View 3" />

</LinearLayout>
```



뷰그룹의 파생된 클래스와 LayoutParams 정보

11

