개발자를 위한 레시피

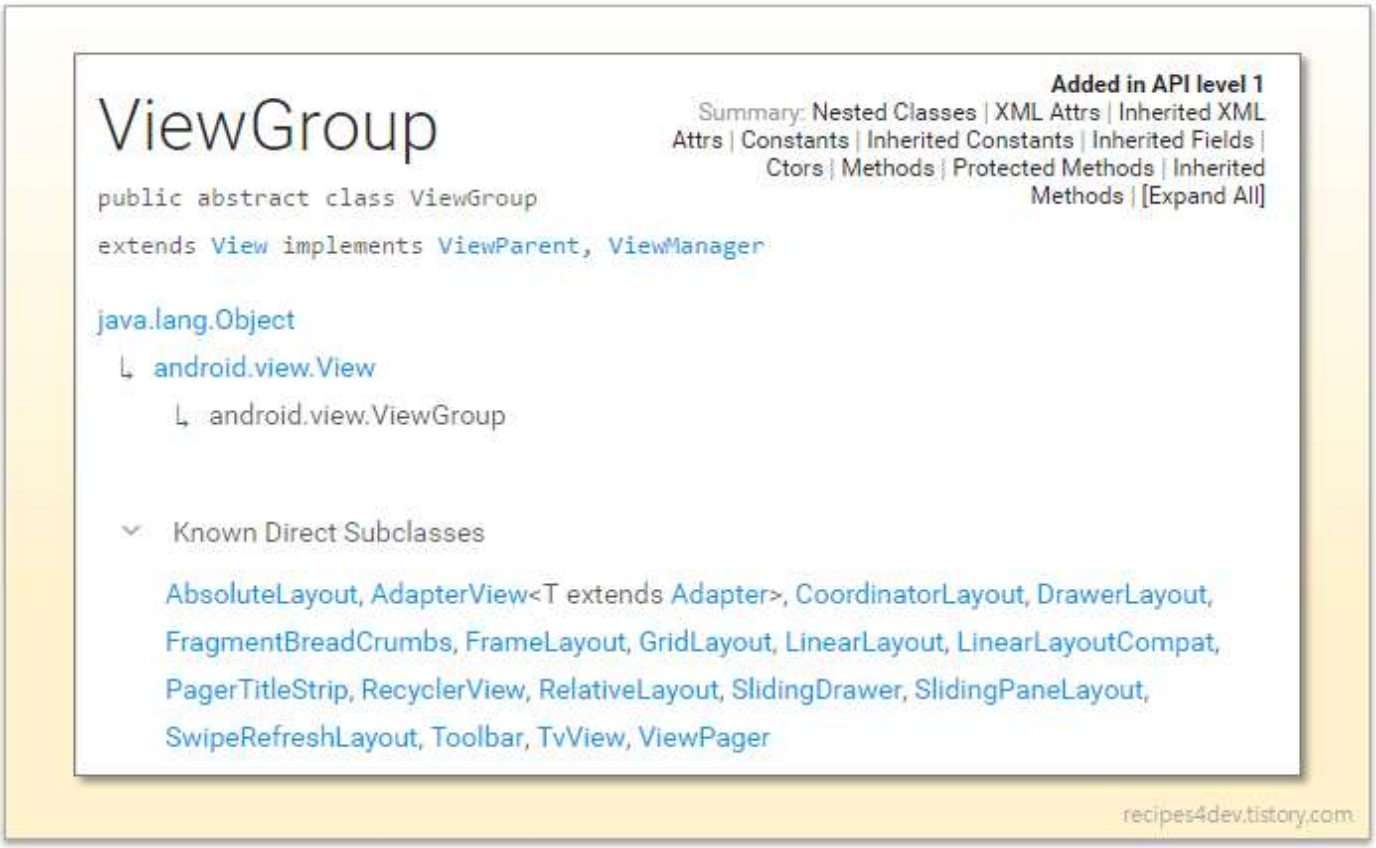
Recipes for Developer.

안드로이드 레이아웃 (Android Layout)

2016.07.27 14:35

1. 안드로이드 Layout 클래스

안드로이드 Layout 클래스는 View 위젯들을 화면에 배치하는 과정에서, 위젯의 위치를 정렬하거나, 연관된 위젯들을 그룹화하는 역할을 수행합니다. 즉, Layout 클래스는 View 위젯들을 그룹화하여 배치하기 위한 용도로 사용되는 ViewGroup이며, 자체적인 UI 표시 기능이나 사용자 이벤트 처리 기능은 매우 제한적으로 사용됩니다.



안드로이드의 ViewGroup 정의에 따르면 "ViewGroup은 (자식이라 불리는)다른 View를 포함할 수 있는 View"를 말합니다. 또한 안드로이드에서 제공되는 모든 Layout 클래스의 부모는 ViewGroup 클래스이기 때문에, Layout클래스는 View 위젯들을 포함하는 컨테이너 역할을 수행할 수 있는 것이죠.

> A `ViewGroup` is a special view that can contain other views (called children.) The view group is the base class for layouts and views containers. This class also defines the `ViewGroup.LayoutParams` class which serves as the base class for layouts parameters.

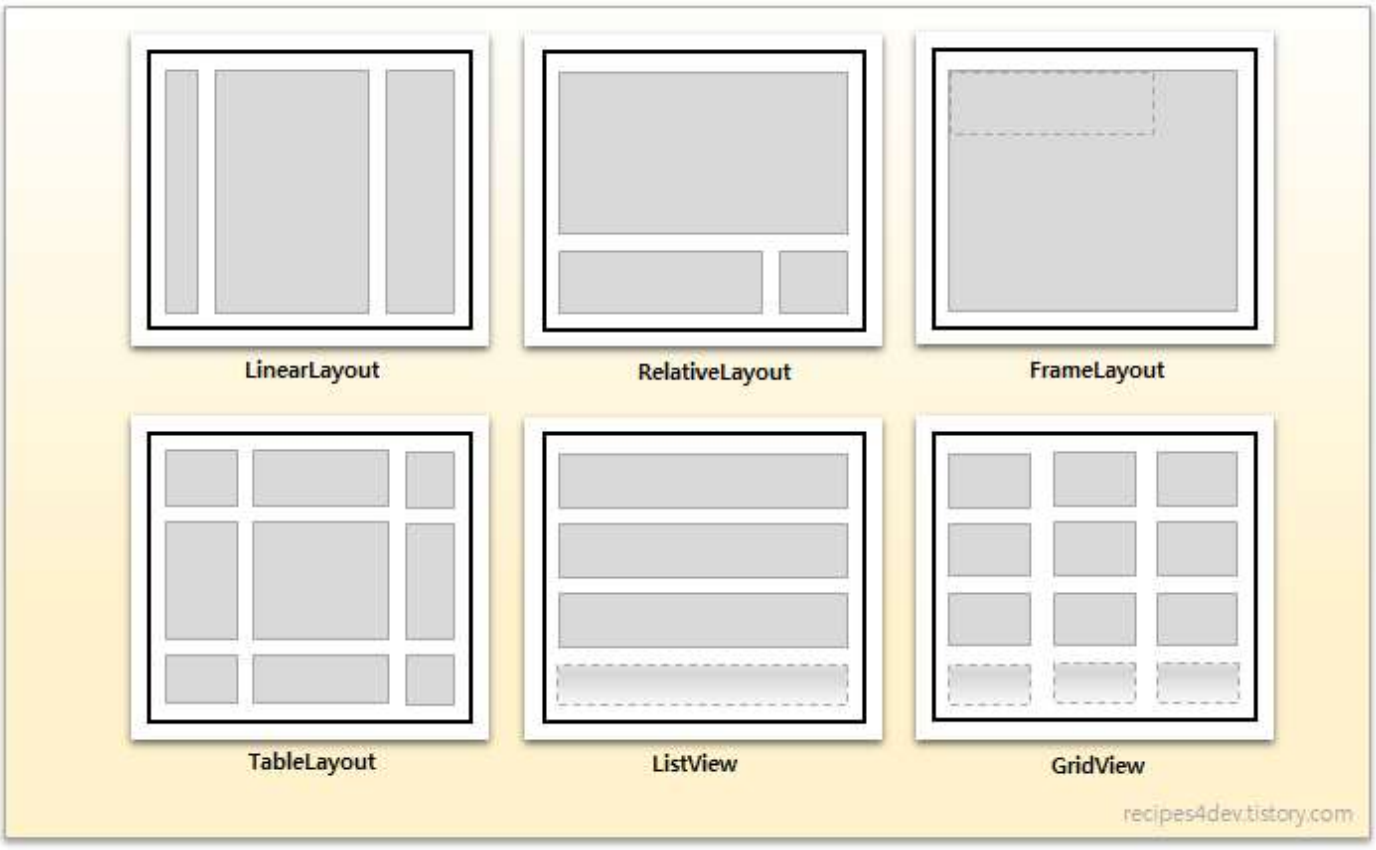
https://developer.android.com/reference/android/view/ViewGroup.html

2. Layout 클래스의 종류

Layout 클래스는 많은 종류가 있습니다. 하지만 모든 Layout 클래스의 사용법을 숙지하고 있을 필요는 없습니다. 대신, 자주 사용되는 몇 가지 Layout에 대한 특징과 사용법 정도만 익혀둬도 큰 문제 없이 Layout을 이용한 UI 구성 작업을 할 수 있습니다.

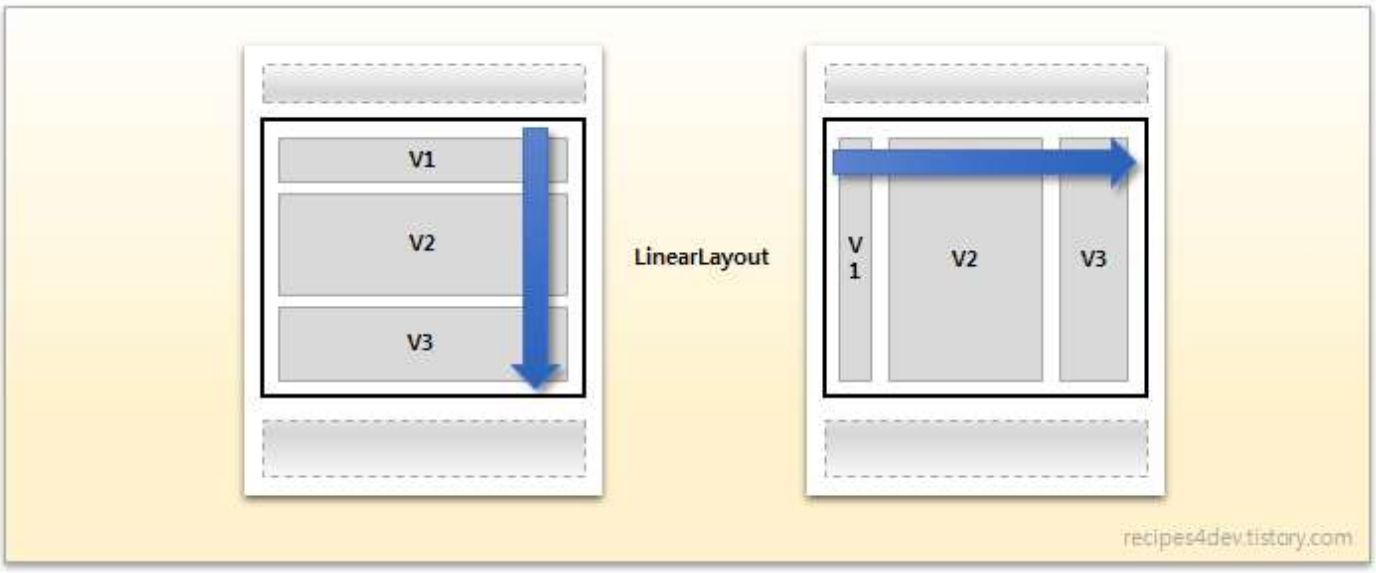
UI 구성 시 빈번하게 사용되는 보편적인 Layout 클래스에는 다음과 같은 것들이 있습니다.

1. LinearLayout
2. RelativeLayout
3. FrameLayout
4. TableLayout
5. ListView와 GridView

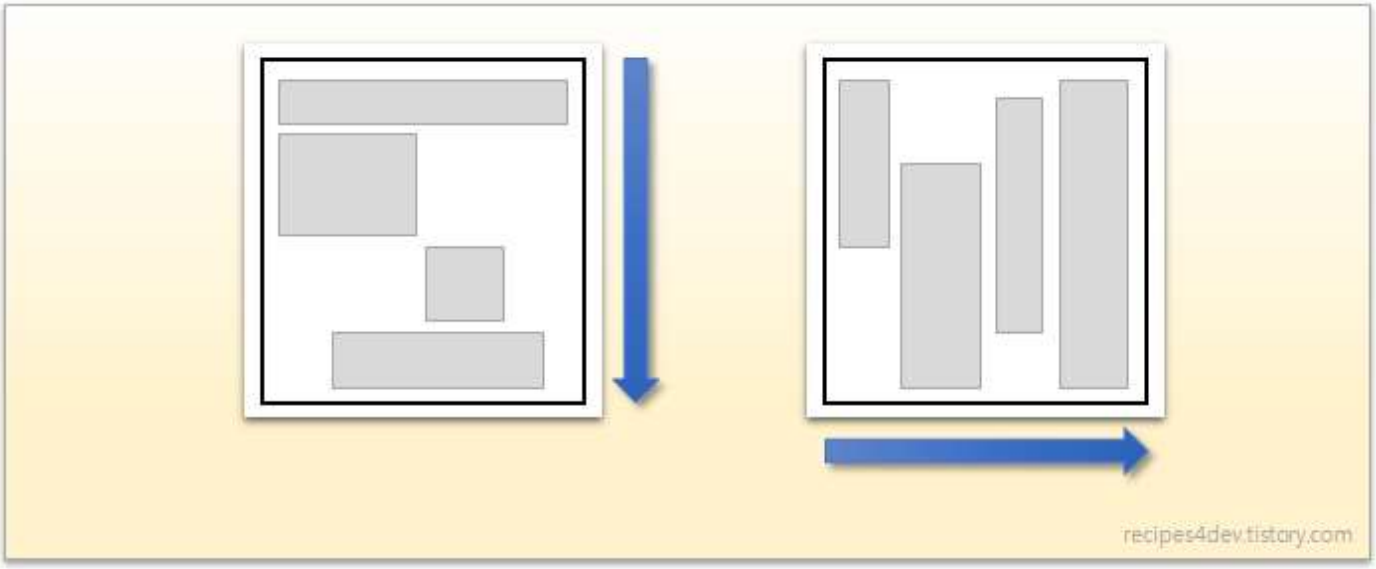


2.1 LinearLayout

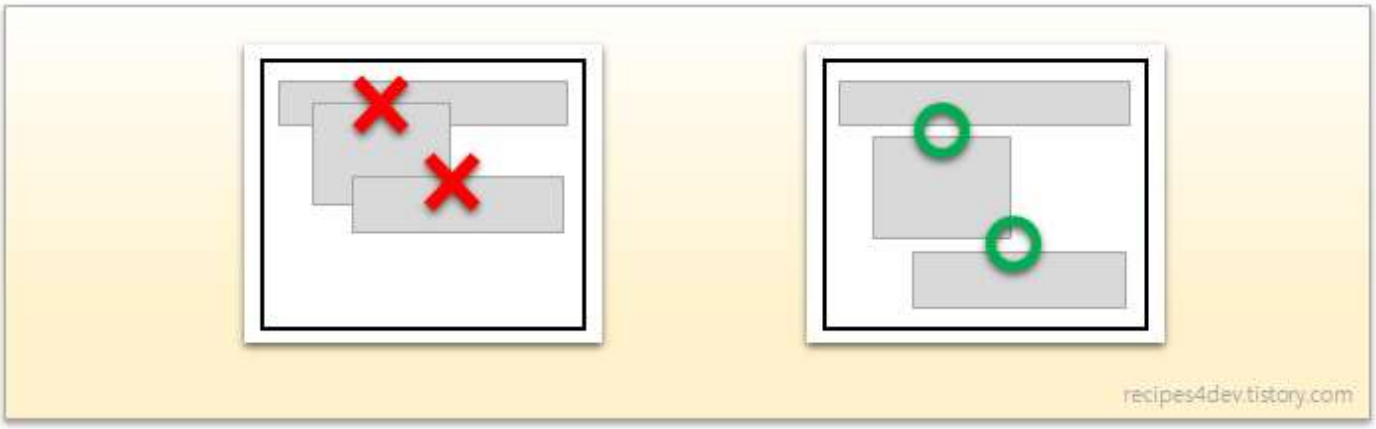
LinearLayout은 Linear(선형, 선으로 된)라는 단어가 포함하는 의미대로, 여러 View 위젯들을 가로 또는 세로 방향으로 나열할 때 사용하는 Layout 클래스입니다.



LinearLayout의 자식(Children)으로 배치되는 View 위젯들은 오직 한 방향(가로 또는 세로)으로만 배치가 가능하기 때문에 위젯의 크기(높이 또는 너비)와 관계없이 한 줄로만 배열됩니다. 즉, 아래 그림과 같이 가로 방향으로 배치될 때는 가로로 한줄(only one row), 세로 방향으로 배치될 때는 세로로 한줄(only one column)로 표시됩니다.

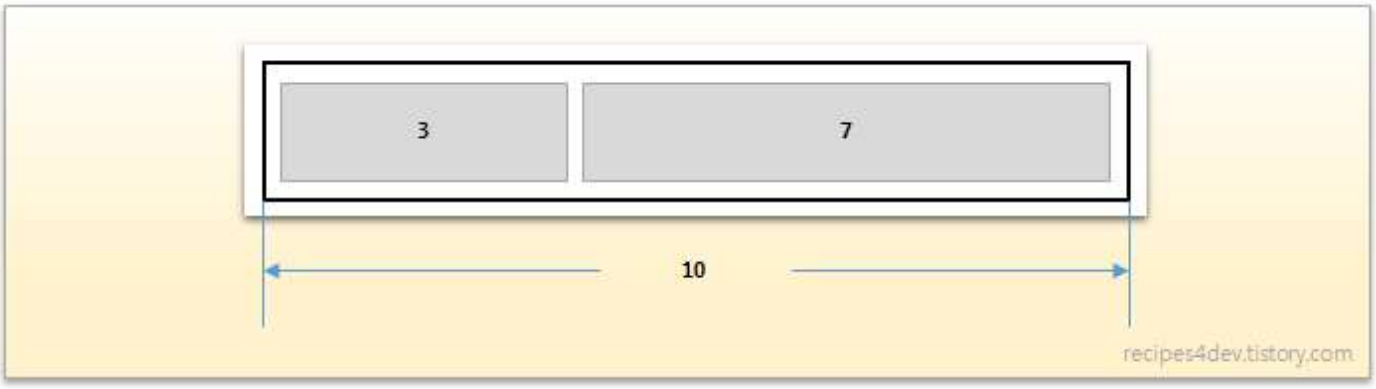


또한 LinearLayout의 자식들은 중첩(overrap)되지 않고, 지정한 방향으로 쌓이는(stacked) 형태로 표시됩니다.



2.1.1 Layout Weight

LinearLayout은 자식(Children)들이 배치될 때, 각 View 위젯들이 차지하는 영역을 dp 등의 고정 치수 단위 뿐만 아니라, 전체 영역 대비 비율의 개념으로 지정할 수 있는 Weight(가중치) 설정 기능을 제공합니다. 쉽게 말해 전체 크기를 10으로 본다면, 첫 번째 View 위젯은 3, 두 번째 View 위젯은 7의 영역을 차지하도록 만들 수 있다는 것이죠.

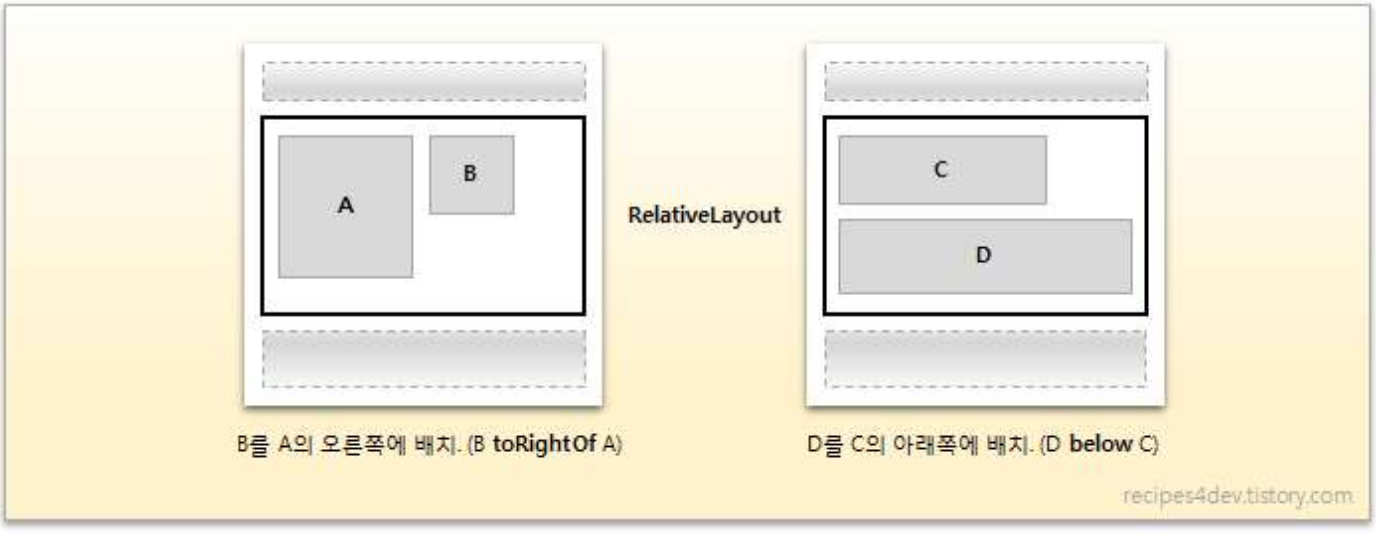


Weight가 개별 View 위젯의 가중치를 설정하는 값이라 할지라도, 그 값이 전체 영역 대비 %(Percentage)를 의미하거나, 전체 영역이 10 또는 100으로 고정된 것은 아니라는 것에 주의해야 합니다. 즉, 자식(Children)들 간의 상대적인 값으로 적용된다는 것입니다. (두 개의 View 위젯이 각각 1, 2의 Weight 값을 가지면, 첫 번째는 1/3만큼, 두 번째는 2/3만큼의 영역을 차지하게 됩니다.)

마지막으로 자식 View 위젯의 Weight 값이 크면 클수록, 화면에서 더 많은 영역을 차지하게 됩니다.

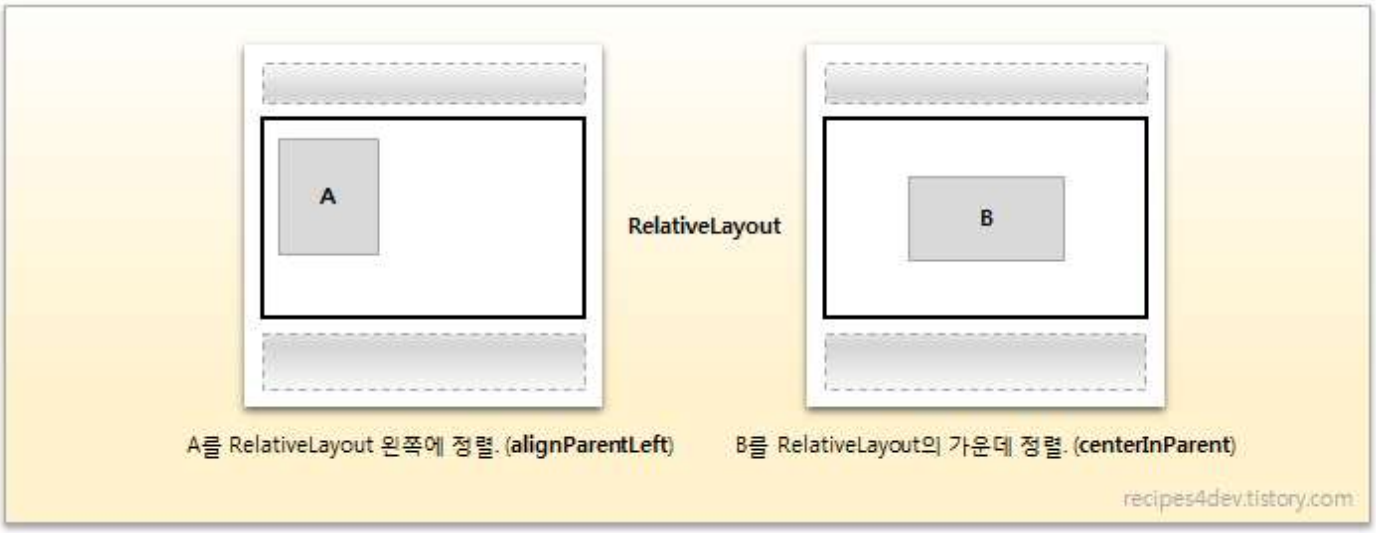
2.2 RelativeLayout

RelativeLayout은 자식(Children) View 위젯들이 서로 간의 상대적 배치 관계에 따라 화면에 표시될 위치가 결정되도록 만들어주는 Layout 클래스입니다. 알기 쉽게 예를 들자면, "A를 화면에 표시하고 B는 A의 오른쪽에 표시"하거나, "C를 첫번째 자식으로 두고 D가 C 아래에 위치하도록 배치"하는 경우죠.

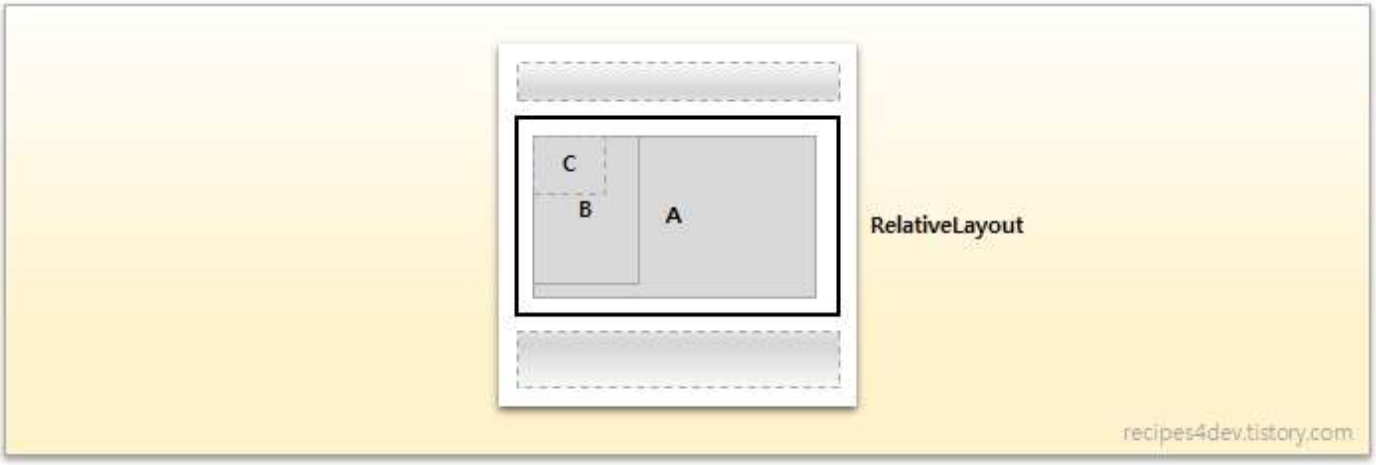


이렇게 "Relative"라는 단어의 사전적 의미대로 "상대적인" 위치를 지정할 수 있지만, 그 대상이 반드시 RelativeLayout에 포함된 View 위젯이어야만 하는 것은 아닙니다. View 위젯들을 포함하는 RelativeLayout(View 위젯 입장에서는 부모(Parent)) 자체가 상대적 위치의 기준점으로 사용될 수도 있습니다.

"A를 RelativeLayout 내의 왼쪽을 기준으로 배치"하거나 "B를 RelativeLayout의 한 가운데 배치"하도록 하는 경우를 예로 들 수 있습니다.



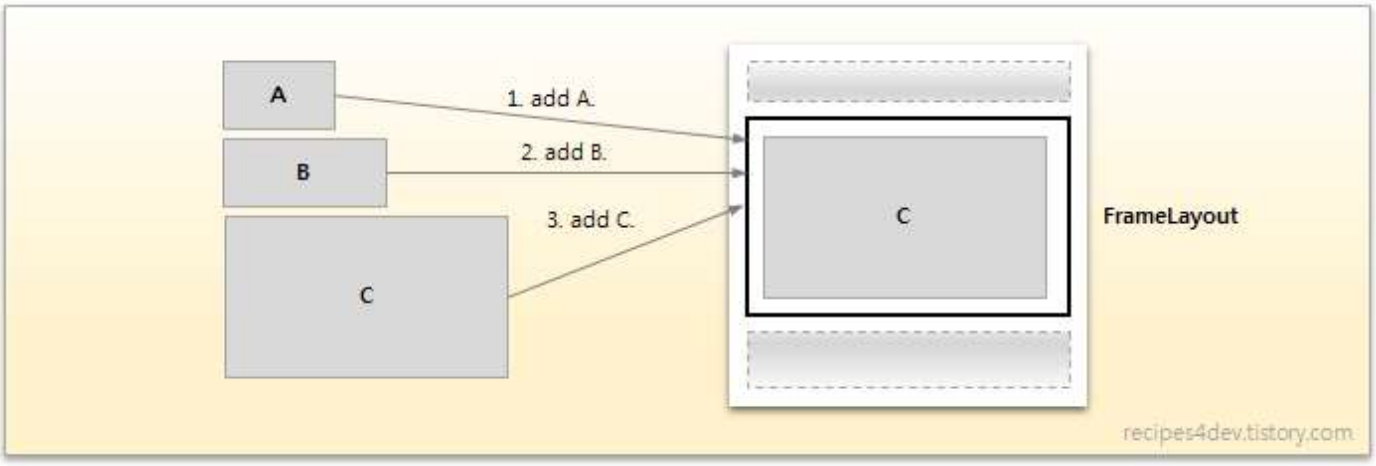
만약 RelativeLayout의 자식(Children)들에 "상대적인" 배치 기준을 지정하지 않는다면, RelativeLayout 내부에서 중첩되어 표시됩니다.



다수의 중첩 LinearLayout들로 구성된 사용자 인터페이스를 하나의 RelativeLayout으로 대체 가능할 정도로 RelativeLayout은 매우 효율적이고 강력한 활용성을 제공하지만, 개발자가 원하는 결과물을 만들기가 다소 어려운 것 또한 사실입니다. 그러므로 사용자 인터페이스를 구성할 때는 무조건 RelativeLayout 하나로만 구성하려 하지 말고, 상황에 맞게 다른 종류의 Layout을 섞어쓰는 것이 개발 시간을 단축시키는 하나의 방법이 될 수도 있습니다.

2.3 FrameLayout

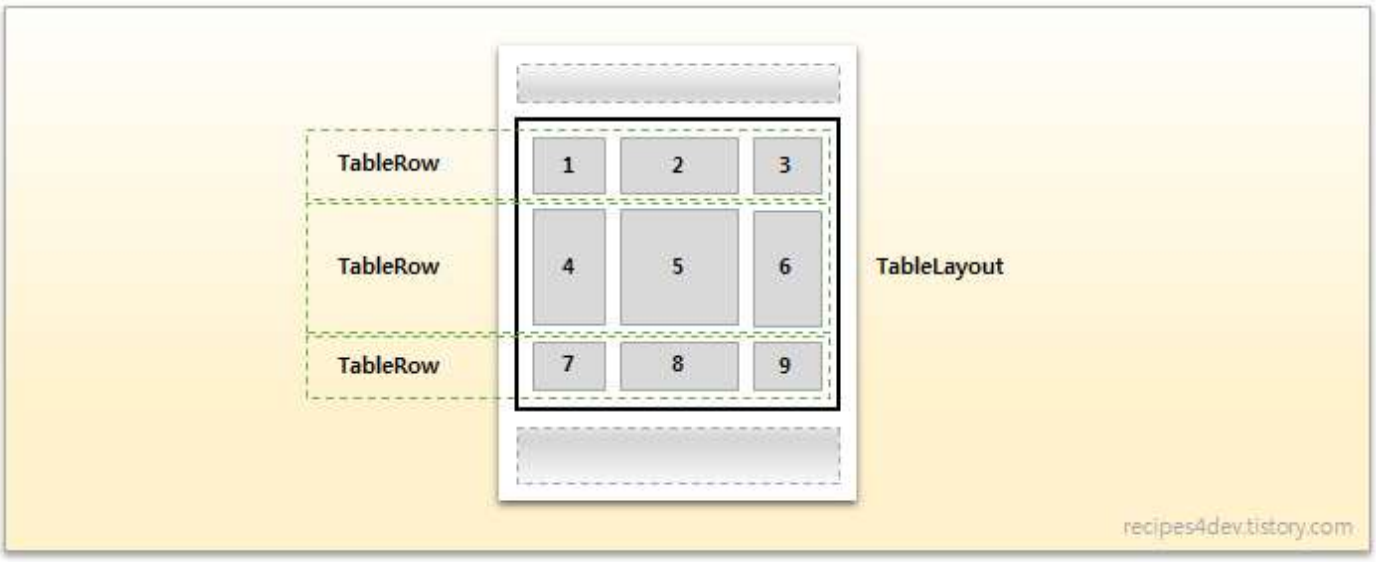
FrameLayout은 주로 하나의 자식 View 위젯만 표시할 때 사용하는 Layout 클래스입니다. 하지만 오직 하나의 자식 View 위젯만 가질 수 있다는 의미는 아닙니다. FrameLayout에 여러 View 위젯을 자식으로 추가하면 겹쳐진 형태로 표시되며, 가장 최근에 추가된 View 위젯이 가장 상위(on top)에 표시됩니다. 이러한 특징을 이용해 가장 상위의 View 위젯만 표시하고 나머지는 보이지 않게 만듦으로써 하나의 자식 View만 표시되도록 만드는 것이죠.



FrameLayout이 많이 사용되는 예 중에 한 가지는 Fragment를 사용하는 경우입니다. 특히 여러 Fragment를 동일한 위치 내에서 교체하여 표시하고자 할 때, Fragment의 컨테이너 역할로써 FrameLayout을 주로 사용합니다. Fragment 표시를 위해 FrameLayout을 사용하는 예제는 [안드로이드 프래그먼트 기본 사용법]에서 확인할 수 있습니다.

2.4 TableLayout

TableLayout은 자식(Children) View 위젯들을 테이블(행과 열로 구성)로 나누어 표시하는 Layout 클래스입니다.



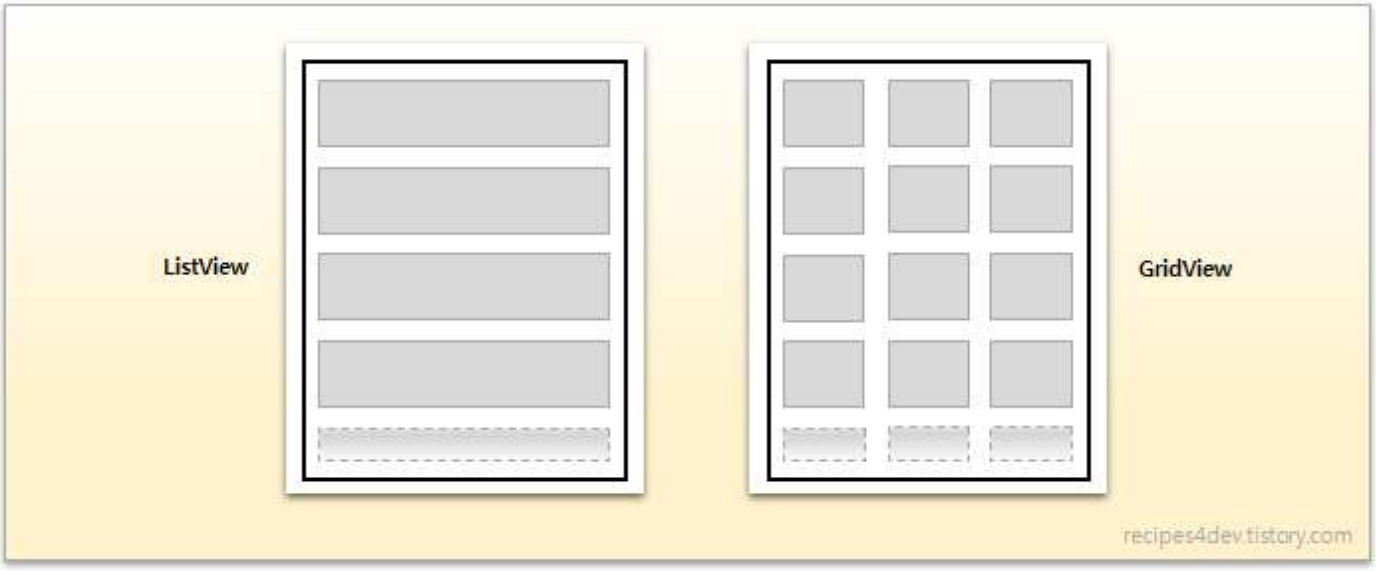
TableLayout에 View 위젯을 추가하기 위해서는 먼저 TableRow 클래스를 사용하여 하나의 행을 추가해야 합니다. 추가된 각 행에 View 위젯을 추가하면 테이블 형태로 정렬되어 표시됩니다. TableLayout의 전체 열(Column) 개수는 TableRow 중 가장 많은 열(Column)의 개수에 맞춰집니다.

HTML에서의 "<table>", "<tr>", "<td>" 태그를 사용하는 것과 매우 유사하지만, 행(Row)을 Span할 수 없다는 것(=두 개 이상의 행(Row) 합치기 허용 안됨)과 "<td>" 역할을 하는 클래스가 별도로 존재하지 않는 것이 차이점이라 할 수 있습니다.

2.5 ListView와 GridView

ListView와 GridView의 경우, 클래스 이름에 "Layout"이라는 단어가 포함되지 않고 위에서 언급한 보편적인 Layout 들과 사용 방법이 달라 Layout이 아닌 것으로 착각 할 수도 있지만, 엄연히 ViewGroup으로부터 상속받은 Layout 클래스의 한 종류입니다.

특히, 동일한 자식(Children) View 위젯을, 내용만 달리하여 반복적으로 표시해야 하는 경우 ListView와 GridView가 매우 유용하게 사용될 수 있습니다.



ListView에 대한 자세한 사용법은 [ANDROID 프로그래밍/LISTVIEW]의 내용을 참고하시기 바랍니다.

3. Layout 클래스 사용하기

Layout 클래스를 사용하는 방법은 리소스 XML 파일에 추가하는 것과 Java 소스에서 Layout의 인스턴스를 생성하는 것, 이 두 가지가 존재합니다. 하지만 Layout 인스턴스를 생성하는 방법은 거의 사용되지 않고, 거의 모든 경우에 리소스 XML 파일에 Layout 클래스를 추가하는 방법이 사용됩니다. (구글의 안드로이드 개발 가이드라인에 따르면 XML로 선언하는 것이 UI의 구조를 시각화하기가 쉽고 문제 발생 시 디버깅하기도 쉽기 때문이라고 합니다.)

아래의 코드는 LinearLayout에 EditText와 Button을 각각 배치한 예제 소스입니다.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
              android:layout_width="match_parent"
              android:layout_height="match_parent"
              android:orientation="vertical" >
    <EditText android:id="@+id/text1"
              android:layout_width="wrap_content"
              android:layout_height="wrap_content"
              android:text="TEXTVIEW1" />
    <Button android:id="@+id/button1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="BUTTON1" />
</LinearLayout>
```

5. 참고.

- ViewGroup에 대한 자세한 도움말.
  - [안드로이드 개발 참조문서 ViewGroup 항목]을 참고하세요.
- 레이아웃에 대한 자세한 도움말.
  - [안드로이드 개발 API 가이드 레이아웃 항목]을 참고하세요.
- LinearLayout에 대한 자세한 도움말.

    - [안드로이드 개발 API 가이드 LinearLayout]을 참고하세요.
    - [안드로이드 개발 참조문서 LinearLayout]을 참고하세요.
- RelativeLayout에 대한 자세한 도움말.
    - [안드로이드 개발 API 가이드 RelativeLayout]을 참고하세요.
    - [안드로이드 개발 참조문서 RelativeLayout]을 참고하세요.
- TableLayout에 대한 자세한 도움말.
    - [안드로이드 개발 참조문서 TableLayout]을 참고하세요.
- FrameLayout에 대한 자세한 도움말.
    - [안드로이드 개발 참조문서 FrameLayout]을 참고하세요.

.END.

7

'ANDROID 프로그래밍 > LAYOUT' 카테고리의 다른 글

안드로이드 프레임레이아웃 뷰 변경하기 1. [addView(), removeView()] (How to change a View in Android FrameLayout) (0)	2017.05.19
안드로이드 프레임레이아웃. (Android FrameLayout) (2)	2017.05.15
안드로이드 렐러티브레이아웃. (Android RelativeLayout) (2)	2017.05.10
안드로이드 리니어레이아웃. (Android LinearLayout) (0)	2016.10.12
안드로이드 레이아웃 공통사항. (Android Layout Common) (2)	2016.10.07
안드로이드 레이아웃 (Android Layout) (0)	2016.07.27

ANDROID 프로그래밍/LAYOUT Android, FrameLayout, layout, Layout Weight, layout height, layout_margin, layout_width, linearlayout, MARGIN, match_parent, padding, RelativeLayout, tablelayout, wrap_content, 레이아웃, 안드로이드

Leave a Comment Leave a Trackback

◀ PREV [1] [···] [64] [65] [66] [67] [68] [69] [70] [71] [72] [···] [90] NEXT ▶

Categories

Recent Posts

Recent Comments

- 지나가던 초보:우선 매번 검색할때마다 좋은 설명글들을..
- 뽀따:음, 죄송하지만 질문 글에 올려주신 내용..
- 뽀따:일단 질문글에 올려주신 코드를 보면, 잘..
- 뽀따:아래 내용 참고 바랍니다.
- 봉봉:제가 질문을 명확하게 드리지 못한 것 같..
- 요누스:insert 쿼리 String sqlInser..
- 요누스:쿼리는 수정핸쓴데 잘못된게 없는거 같..