<6대 Layout>

레이아웃이란 다른 View를 포함해서 화면을 구성할 수 있는 클래스



-레이아웃은 ViewGroup클래스로부터 상속된다.

<자주 사용되는 속성>

\*padding: 위젯의 여백

\*baselineAligned: 보기좋게 정렬. True/ False

\*layout\_gravity/ gravity: x축과 y축을 기준으로 정렬

-top, bottom, left, right

-center\_vertical, center\_horizontal (orientation과 반대로 써야함)

-fill, fill\_vertical, fill\_horizontal

-clip\_vertical, clip\_horizontal: 부모 Layout을 넘어서는 영억 자르기.

-start, end

1. LinearLayout

Linear는 ‘선형’이란 뜻으로 화면을 가로 또는 세로 방향으로 나열할 때 사용하는 Layout

<특징>

1) Horizontal(수평, 가로)과 Vertical(수직, 세로)이 존재한다.

2) 나누는 화면을 뷰(view)라고 하고 뷰를 단순히 나누는데 편한 편이다.

3) 사용이 간편하고 표시 형태가 직관적이라는 장점

4) 무엇보다 사각형 박스 형태의 디스플레이 화면에 UI 요소들을 일렬로 배치할 수 있어서 안정감있는 화면 구성이 가능하다는 이유 때문

<사용법>

1. orientation: 자식(Children) view 위젯들이 나열되는 방향 지정

> horizontal (0): 자식(Children) View 위젯들을 가로 방향으로 나열

> vertical (1): 자식(Children) View 위젯들을 세로 방향으로 나열

> 지정이 안되었을 시 default: horizontal

2. layout\_weight: 자식(Children) View 위젯이 차지하는 가중치(weight) 지정.

>소수점 단위 사용 가능

>특정 단위가 아닌 전체에 대한 비율의 개념으로 사용.

>지정하지 않았을 때의 기본 값은 0. (내용만큼의 크기 차지)

>가중치가 적용되는 방향(width 또는 height)의 값이 “0dp”이어야 함

(테스트한 결과 value값만큼 추가로 주어진다.)

3. weightSum: LinearLayout의 최대 가중치(weight)값 지정.

>소수점 단위 사용 가능.

>지정되지 않을 시, 합산된(sum) 가중치는 자식(Children) View들의

모든 layout\_weight값을 더하여 계산

2. RelativeLayout

레이아웃 내의 자식(Children)View 위젯들 또는 부모역할을 수행하는 RelativeLayout 들이

서로 간의 상대적(Relative)위치 관계에 따라 최종적으로 표시될 영역을 결정하도록 만드는 레이아웃

<사용가능한 속성의 종류>

1)상대적 위치 정렬

\* layout\_toLeftOf, above, toRightOf, below : 뷰를 기준 뷰의 서, 북, 동, 남쪽에 배치

\* layout\_toStartOf, toEndOf: 뷰를 기준 뷰의 시작에 배치, 끝에 배치

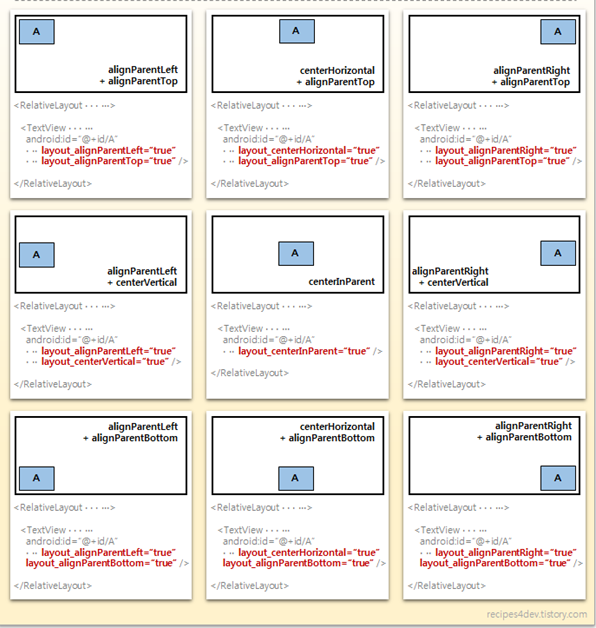


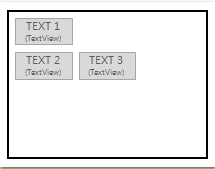
2) 부모인 RelativeLayout을 기준으로 상대적 위치 정렬

\* layout\_alignParent(Left,Top,Right,Bottom): 부모의 영역 내에서 그 방향에 배치

\* layout\_centerHorizontal, centerVertical, centerInParent: 가로 가운데, 세로 가운데, 정 중앙 배치

\* layout\_alignParentStart, alignParentEnd: 부모 영역의 시작 지점, 끝 지점에 배치





예시1) 오른쪽 그림과 같은 TEXT3을 TEXT2를 기준으로 옆에 두고 싶다면

TEXT2가 TEXT1을 기준으로 below를 잡아줬듯이 TEXT3도 TEXT1을

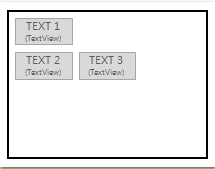
기준으로 below를 잡아줘야한다.

3)맞춤 정렬

\* layout\_align(Left,Top,Right,Bottom): 뷰의 방향을 기준으로 뷰의 방향에 맞춤.

\* layout\_alignBaseline: 뷰의 폰트 기준선(Baseline)을 기준 뷰의 폰트 기준선에 맞춤.





아까의 예시1)과 같은 그림이지만 TEXT3에 TEXT2에 해당하는 alignTop를

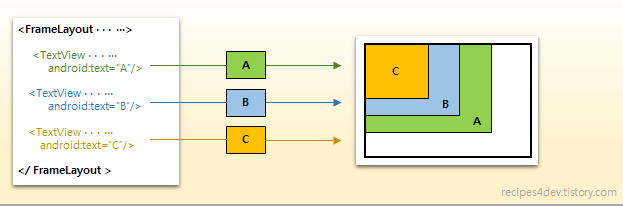
잡아준다면 바로 오른쪽으로 오게 할 수 있다.

3. FrameLayout

자식으로 추가된 여러 뷰 위젯들 중 하나를 Layout의 전면에 표시할 때 사용하는 클래스 ex)액자

구글에서는 오직 하나의 자식(Children) 뷰 위젯만을 표시하는데 사용하는 것을 권고했습니다.

FrameLayout은 또한 Fragment 나 Tab처럼 특정 위젯이나 페이지가 변경되는 UI에 적합한 UI입니다.

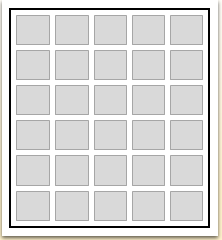


FrameLayout에서 자식(Children)뷰 변경하기

1. FrameLayout의 addView()와 removeView() 또는 removeViewAt() 함수를 사용하여 수행할 수 있다.

2. visibility속성을 이용해서 변경하는 방법

3. LayoutInflater를 이용해서 xml을 view객체로 변환할 수 있다.

4. TableLayout

뷰를 테이블 형식으로 표시하는 Layout이다.

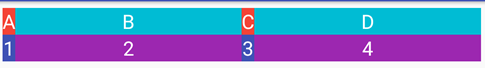
테이블은 행과 열로 구성되며 표와 같은 형태를 보여준다.

<특징>

\* 최대 열의 개수는 가장 많은 자식을 가진 TableRow에 맞춰짐.

\* stretchColumns를 통해 늘이고자하는 열의 인덱스를 지정할 수 있다.

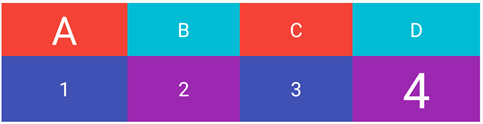
>첫번째와 세번째 열은 원래 크기대로 표시하고 싶다면 android:stretchColumns=”1,3”으로 지정.



>전체 열의 너비가 늘어나게 만들려면 stretchColumns=”\*”으로 지정하면 됩니다.



\* 가장 높은 셀의 높의 너비, 가장 높은 셀의 높이에 맞춰집니다.



\* layout\_column을 사용하여 해당 인덱스에 값을 넣을 수 있습니다.



\* layout\_span을 이용하면 셀을 합칠 수 있습니다.

\* collapsecolumns을 이용하면 특정 열을 감출 수 있습니다. (0부터 시작)

\* shrinkColumns을 이용하면 전체 너비를 넘지않도록 만들 수 있습니다.

5. GridLayout

TableLayout과 유사한 점을 보여주나 이의 단점을 보완하고 다른 레이아웃들의 장점을 합친 Layout이다.

다른 레이아웃과 중첩으로 사용할 필요가 없어 메모리 사용량을 줄일 수 있다는 장점이 있다.

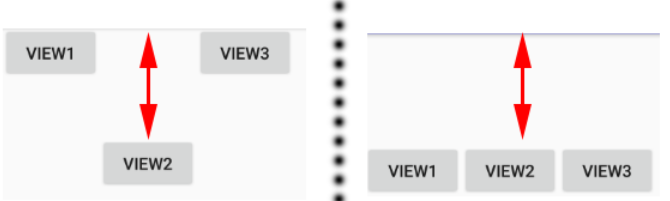
<특징>

\* orientation을 이용해 방향을 정할 수 있다.

\* rowCount, columnCount를 이용해 행의 개수, 열의 개수를 지정할 수 있다.

\* alignmentMode를 이용해 기준이 되는 뷰의 여백을 포함하여 나머지 뷰들을 정렬할 수 있습니다.

>alignBounds(차일드 외곽 기준), alignMargins(마진을 기준, 디폴트)



\* layout\_column/layout\_row를 이용해 특정 행과 열의 위치를 지정할 수 있습니다.

\* layout\_columnspan / layout\_rowspan을 이용해 행과 열을 합칠 수 있다.

그렇다면 TableLayout과 GridLayout의 차이점은 무엇일까?

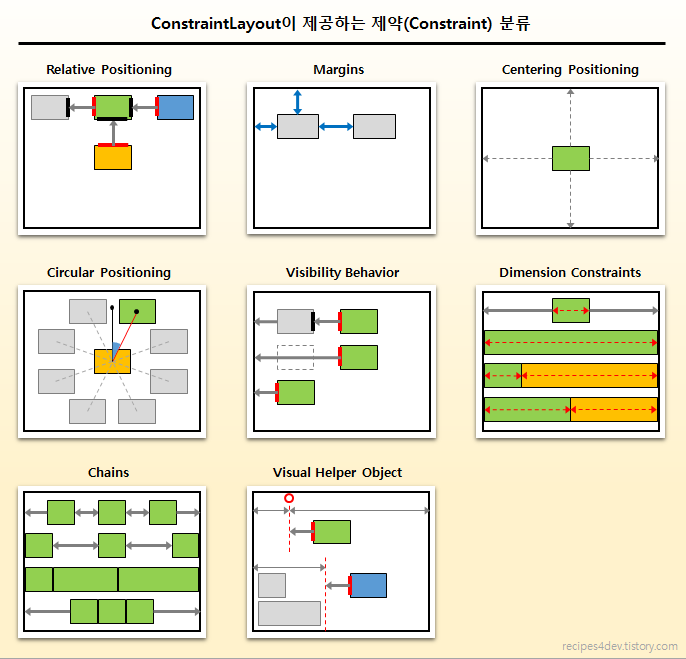
=>column을 통합할 수 있지만 row는 통합이 불가능하다. 또한 TableLayout에서도

세로순서로 넣을 수는 있지만 불편하다.

6. ConstranitLayout

Layout 구성 시, 뷰의 위치와 크기를 유연하게 조절할 수 있는 레이아웃.

레이아웃 사이의 관계에 따라 모든 보기의 레이아웃이 결정된다는 점에서 RelativeLayout과 비슷하지만, 보다 유연하고 Android 스튜디오의 Layout Editor와 함께 사용하기가 좀 더 쉽다.



<제약조건>

보기를 Layout Editor에 놓으면 제약조건이 없어도 둔 위치에 그대로 남아 있습니다. 이는 단지 더 쉽게 편집하도록 한 것이며, 기기에서 레이아웃을 실행할 때 보기에 제약조건이 없으면 [0, 0](맨 위 왼쪽 모서리) 위치에 그립니다.,

<속성 기본 사용법>

Layout\_constraintXXX <= XXX는 시작하는 위치입니다.