

# 안드로이드 스튜디오 레이아웃 총 정리

하웅이의 데일리

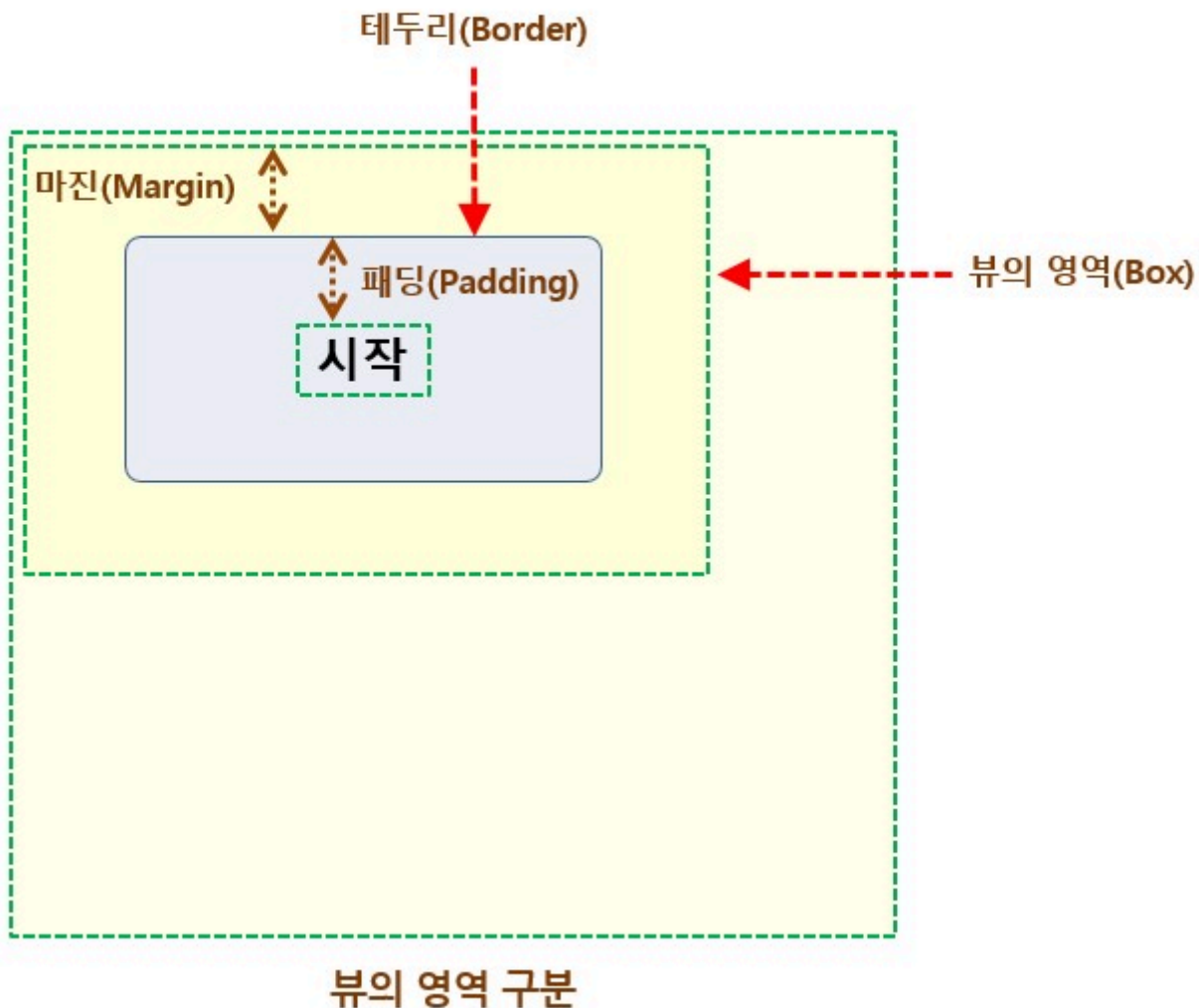
2021/07/14 21:50

<http://blog.naver.com/slovess1/222431760683>

**Layouts** : 화면의 구성 요소를 배치하는 일, 즉 화면의 틀

**뷰(View)**의 개념을 알아야 하는데, 뷰는 레이아웃, 위젯이라고도 부르며  
화면을 구성하는 인터페이스의 구성 요소이다  
(용도에 따라 뷰를 다르게 부르는데, 배치할 땐 레이아웃 / 컨트롤할 땐 위젯)

뷰의 묶음을 뷰 그룹(View Group)이라 부르며  
뷰 그룹 역시 하나의 뷰와 같은 속성을 가지기 때문에  
뷰 그룹들을 모아 더 큰 뷰 그룹을 만들 수 있다



## ■ 뷰와 상속

뷰 클래스란 모든 뷰들의 상위 관계를 맺은 클래스로

해당 뷰를 상속하는 하위 클래스들이 TextView, Button 등이다

## ■ 뷰 속성 지정

뷰의 크기를 지정하기 위해선 layout\_width와 layout\_height를 이용한다

(숫자를 넣을 땐 반드시 단위를 적어줘야 함)

dp(Density-independent Pixels) = 밀도 독립형 픽셀

시스템 폰트 사이즈와는 관계 없는 고정 값

sp(scaled pixels) = 스케일 독립형 픽셀

시스템 폰트 사이즈를 조절하면 함께 변경되는 값

-> 레이아웃 등 UI적 요소는 dp를 사용하는 것이 좋고,

글자 크기에는 sp를 사용하는 것이 좋음

· wrap\_content는 뷰 내용물의 크기에 맞게 설정되는 것!

만일 텍스트 뷰라면 글의 길이에 맞게 설정

· match\_parent는 fill\_parent를 대체하는 기능

이름에 맞게 상속 받는 상위 클래스의 뷰 크기에 맞게 설정됨

Linear Layout : 선형 레이아웃

1) orientation : 레이아웃 내의 위젯의 수직 또는 수평 방향을 설정

horizontal 수평 (가로)

vertical 수직 (세로)

2) gravity : 레이아웃 내의 위젯의 정렬 방향을 설정

## [안드로이드 스튜디오] #3-3 리니어 레이아웃(Linear Layout) : gravity 속성

본 포스팅은 국비지원으로 모바일앱을 수강하며 배웠던 수업 내용을 기록한 포스팅입니다. 언어를 배우고 ...

[blog.naver.com](http://blog.naver.com)

layout\_gravity : 부모 컴포넌트의 여유공간을 활용하여 뷰를 정렬함

gravity : 뷰내에 여유공간을 활용하여 뷰를 정렬함

- left
- right
- center
- bottom
- top

(2개를 조합해 사용할 수도 있음)

3) padding : 레이아웃 내의 위젯의 여백 설정

4) layout\_weight : 레이아웃이 전체 화면에서 차지할 공간의 가중값(상대적인 비율)으로, 여러 레이아웃이 중복될 때 사용

android:layout\_weight="1"      1만큼의 가중치를 정해서 그 비율만큼 레이아웃을 차지

여기서 상대적인 비율이 왜 중요하냐면 다양한 사이즈의 기기를 지원해야하기 때문이다.

절대적인 값을 넣어주면 아주 요상하게 될 것

weight 속성을 사용할 땐 match\_parent를 같이 사용해선 X



# 안드로이드 프로그래밍

## 안드로이드 weight (layout\_weight)

안드로이드의 layout\_weight 레이아웃 매개변수는 LinearLayout에게 자식들 배치공간을 알려준다 먼저 자식뷰들의 width (수직정렬일 경우는 height)를 기준으로 모든 뷰들을 배치한다 그 후 남은 공간이 있으..

[belll.tistory.com](http://belll.tistory.com)

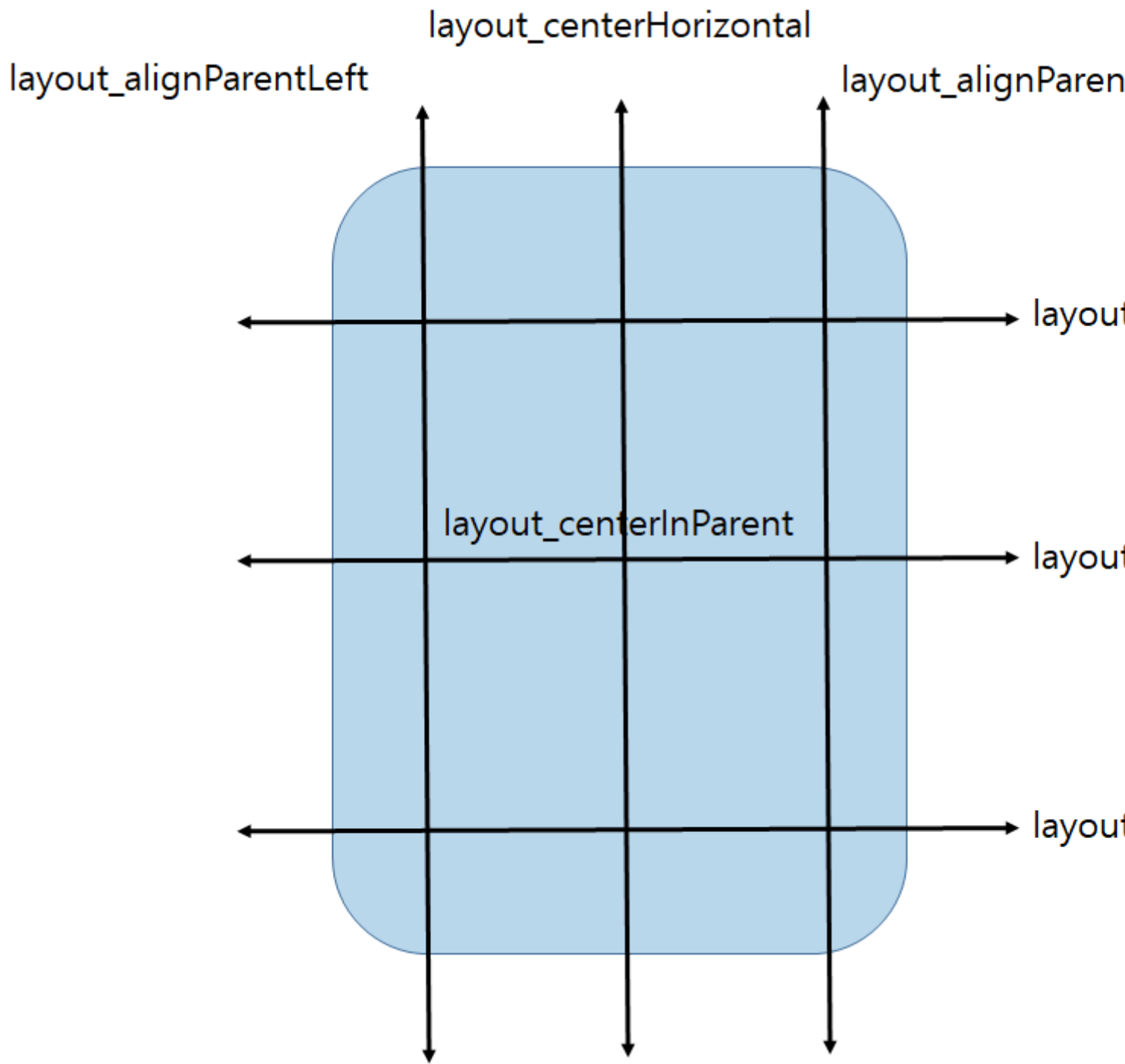
5)baselineAligned : 레이아웃 안에 배치할 위젯을 보기 좋게 정렬

## Relative Layout : 상대 레이아웃

레이아웃 내부에 포함된 위젯을 상대적인 위치로 배치

## 배치 방법

만일, 상대적인 위치를 지정하지 않으면 자식 View들은 왼쪽 최상단에 위치하게 된다



<layout\_alignParent> 계열 속성은 부모 레이아웃(RelativeLayout) 안에서 자식 View가 어디에 위치해야 하는지 지정

android:layout\_alignParentTop="true" 레이아웃 최상단

android:layout\_alignParentBottom="true" 레이아웃 최하단

android:layout\_alignParentLeft="true" 레이아웃 왼쪽

android:layout\_alignParentRight="true" 레이아웃 오른쪽

android:layout\_alignParentStart="true" 레이아웃 시작점(기본적으로 왼쪽)

android:layout\_alignParentEnd="true" 레이아웃 끝점(기본적으로 오른쪽)

<layout\_center> 계열 속성은 자식 View가 수평 또는 수직 중심부로 위치시킬 지 결정

android:layout\_centerHorizontal="true" 수평 가운데

android:layout\_centerVertical="true" 수직 가운데

android:layout\_centerParent="true" 레이아웃 정중앙

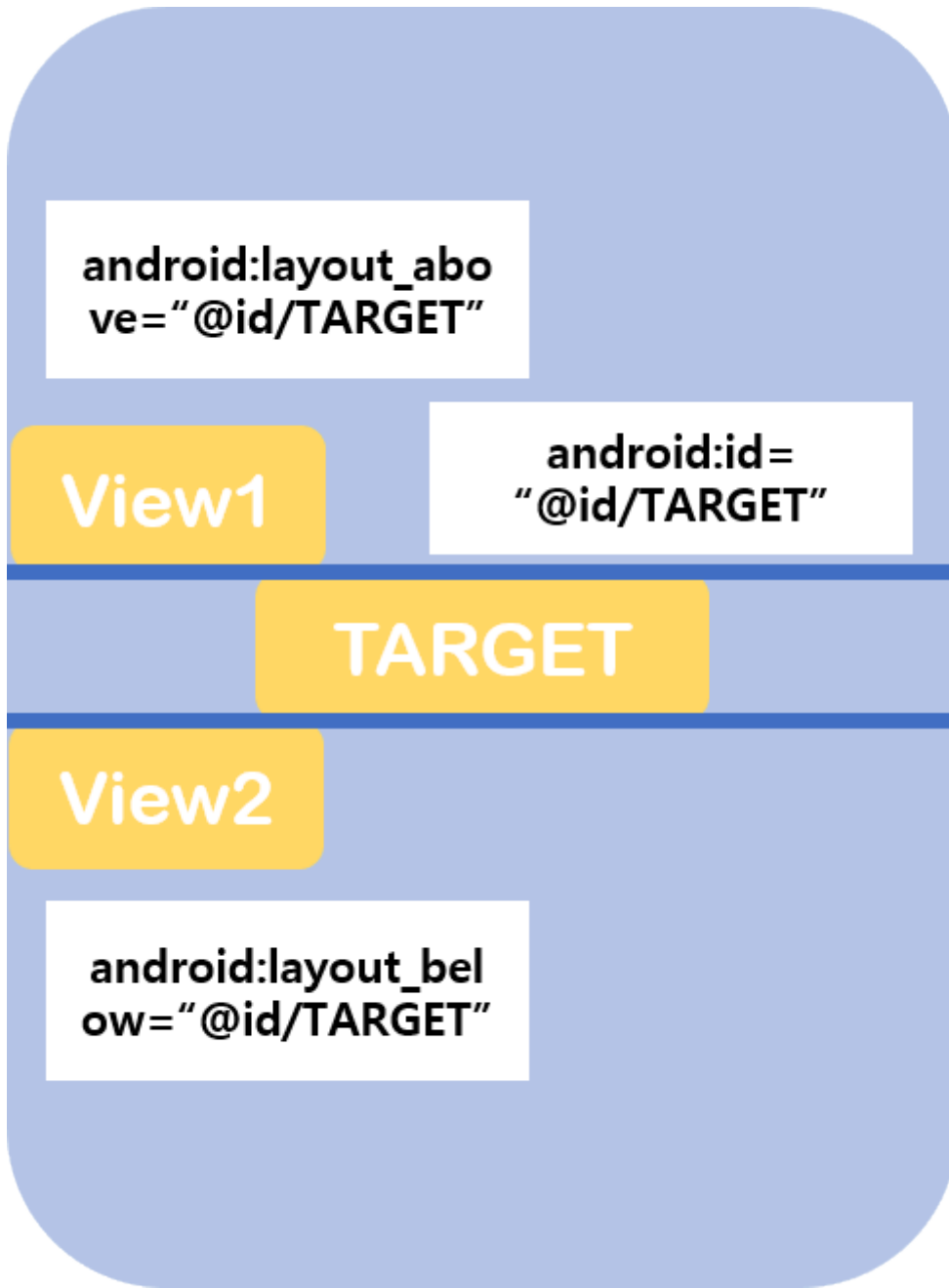
특정 View 기준(기준되는 View는 ID가 반드시 필요!)

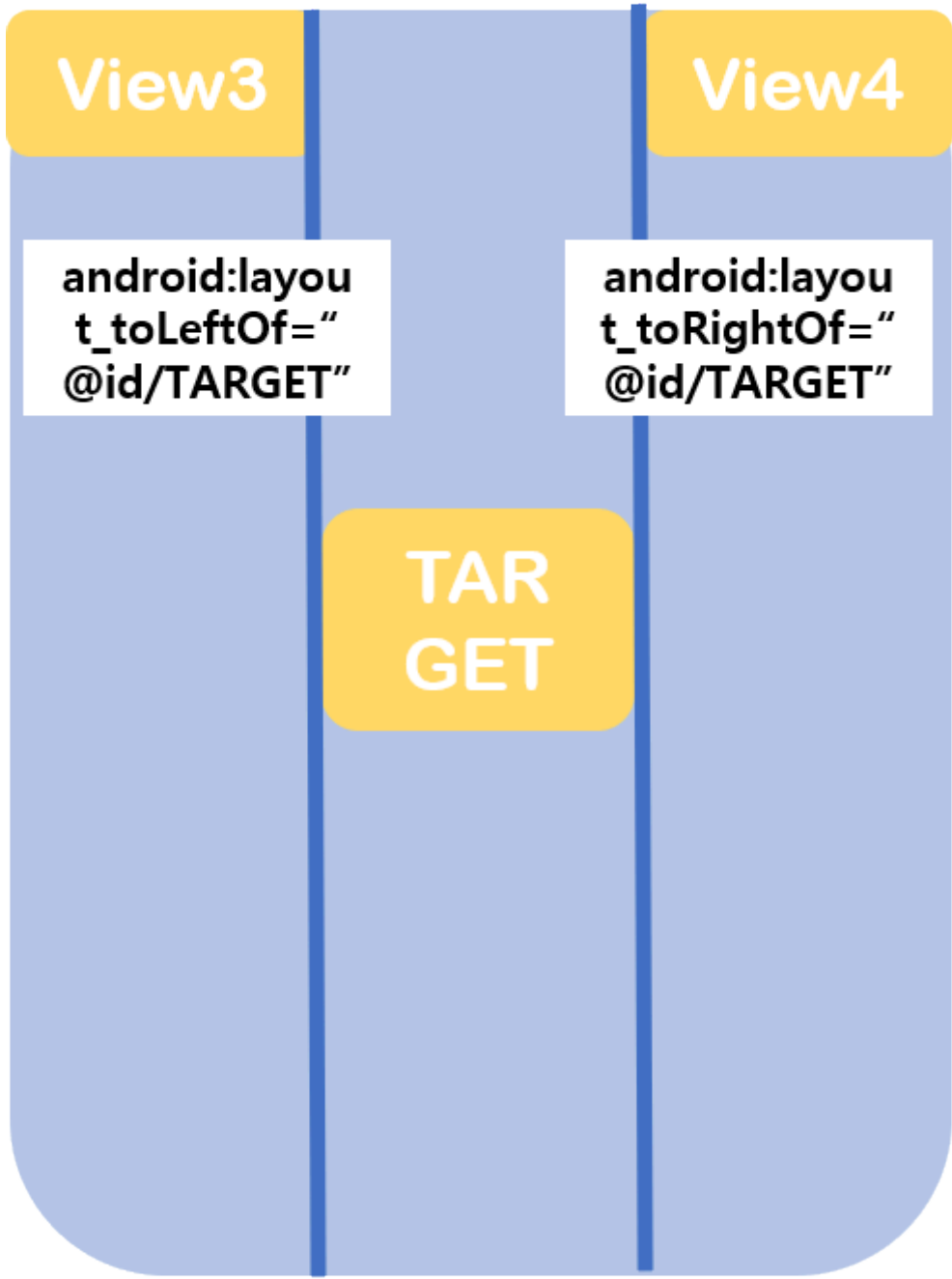
android:layout\_above="@id/기준id" 만들 뷰의 아래를 기준 뷰 위에 맞춤

android:layout\_below="@id/기준id" 만들 뷰의 위를 기준 뷰 아래에 맞춤

android:layout\_toLeftOf="@id/기준id" 기준 뷰의 왼쪽에 위치

android:layout\_toRightOf="@id/기준id" 기준 뷰의 오른쪽에 위치



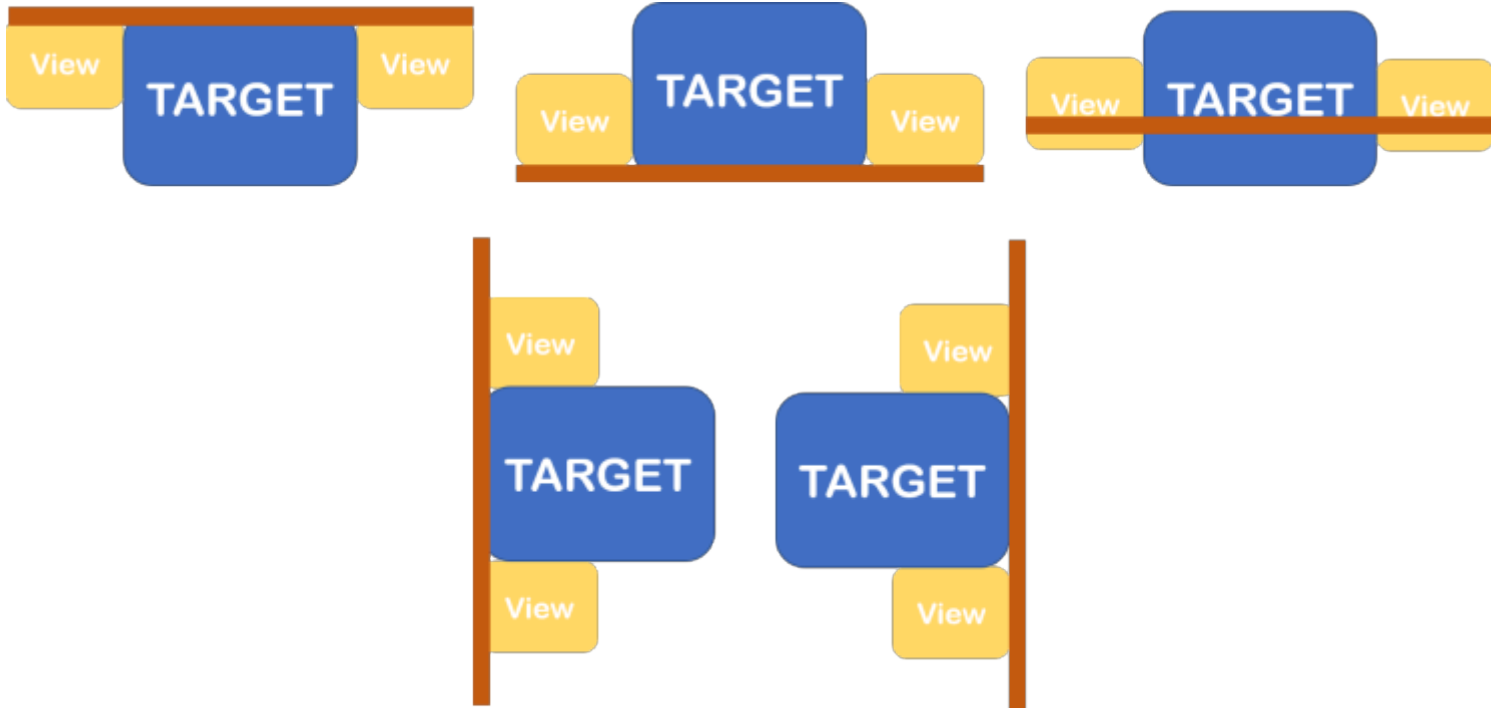






### 특정 View를 기준으로 대상을 정렬(align)

<code>android:layout_alignTop="@id/기준id"</code>	만들 뷰의 상단을 기준 뷰의 상단에 맞춤 //상단
<code>android:layout_alignBottom="@id/기준id"</code>	만들 뷰의 하단을 기준 뷰의 하단에 맞춤 //하단
<code>android:layout_alignBaseline="@id/기준id"</code>	만들 뷰의 텍스트 밑줄을 기준 뷰의 텍스트 밑줄에 맞춤 //중앙
<code>android:layout_alignLeft="@id/기준id"</code>	만들 뷰의 왼쪽을 기준 뷰의 왼쪽에 맞춤 //좌측
<code>android:layout_alignRight="@id/기준id"</code>	만들 뷰의 오른쪽을 기준 뷰의 오른쪽에 맞춤 //우측



### Frame Layout : 프레임 레이아웃

여러 개 뷰 위젯들을 중첩하고, 그 중 하나를 제일 위에 표시할 때 사용하는 레이아웃  
겹쳐서 배치할 수 있기 때문에 다양한 화면 구성이 가능



코드 작성 순서와 반대로 제일 마지막에 입력한 코드가 최상단에 나타남

foreground                      프레임 레이아웃의 전경 이미지를 지정

foregroundGravity              전경 이미지 위치를 지정                      fill, rightm left, top, bottom 등

visibility 속성은 특정 View의 보이는 상태를 결정

: 중첩된 View 중에서 특정 View만 보이게 하는 가시성 관련 속성

andorid:visibility="visible"    가장 늦게 놓는 뷰

android:visibility="invisible" 먼저 놓은 뷰

android:visibility="gone"

**Table Layout : 테이블 레이아웃**

표나 엑셀 시트와 같은 형태

원하는 행 갯수만큼 TableRow를 추가해 사용하면 됨!

(이 말은 TableRow가 행의 크기란 말)

stretchColumns : 화면 크기에 맞게 행의 너비를 늘리는 속성

(행 번호가 0부터 시작, 모든 행 지정은 '\*')

행이 3개라고 가정했을 때

android:stretchColumns="0"    1행이 길어짐

android:stretchColumns="0, 1" 1, 2행이 길어짐

android:stretchColumns="\*"    3개 모든 행이 균등함

shrinkColumns : 각 열의 폭을 자동 축소

버튼을 많이 넣다보면 화면 밖으로 넘어가버리는 경우가 생기는데, 전체적인 너비를 자동으로 맞춰줌

android:shrinkColumns="\*"   

layout\_columns: 컬럼 인덱스를 지정하여 순서 정하기

버튼을 추가한다고 할 때 빈 공간에 차례대로 오는데, 원하는 행에 배치시키고 싶을 때 사용하는 속성

android:layout\_cloumns="2" 3행에 버튼 삽입

layout\_span: 행 여러 개를 합쳐 뷰 배치하고 싶을 때(항상 자신 기준 오른쪽 행들이랑만 가능)

android:layout\_span="4"                      자신 포함 오른쪽 3개 총 4개 행 길이로 길어짐

### Grid Layout : 그리드 레이아웃

격자 모양의 표에 배치, 셀의 위치와 크기를 다양하게 변경 가능하며 셀끼리의 합병도 가능해 응용 가능성 ?



### [안드로이드] 그리드 레이아웃(Grid Layout)

※ 저는 안드로이드 프로그래밍 정복(김상형 ?, 한빛미디어) 책을 이용해 공부하고 있으며 예제와 코드는 이 책을 통해 공부중임을 밝힙니다. 개인적인 공부를 하면서 정리한 형식이기 때문에 심각한 오류가 있을..

[whatisthenext.tistory.com](http://whatisthenext.tistory.com)

### Constraint Layout : 제약 레이아웃