

# 모바일 프로그래밍

## 02 사용자 인터페이스 기초 2

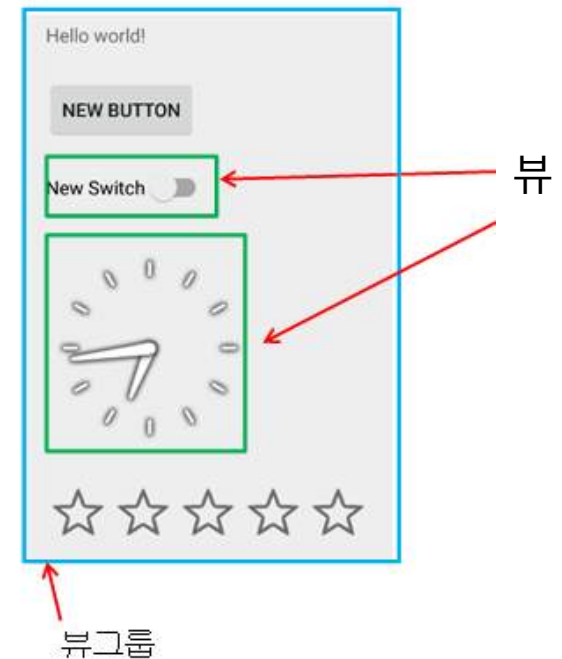
2017 2학기

강승우

레이아웃 (Layout)

# 레이아웃

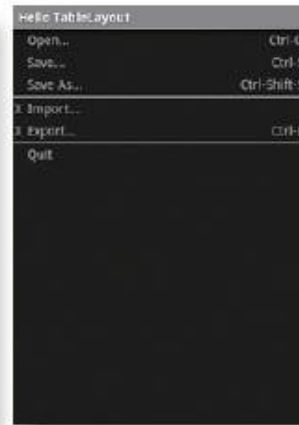
- 뷰를 화면에 배치하는 방법
- 절대 위치(좌표) vs. 상대 위치
  - 안드로이드 기기 (스마트폰, 태블릿)는 화면 크기와 해상도가 다양하기 때문에 절대 좌표를 가지고 뷰의 위치를 지정하는 방식이 아니라 뷰들의 상대적인 배치를 지정하는 방식이 쓰임



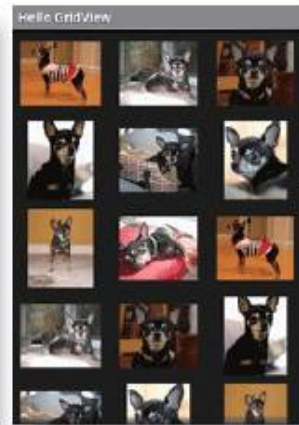
# 레이아웃 종류



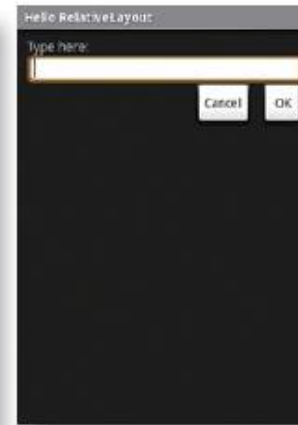
LinearLayout



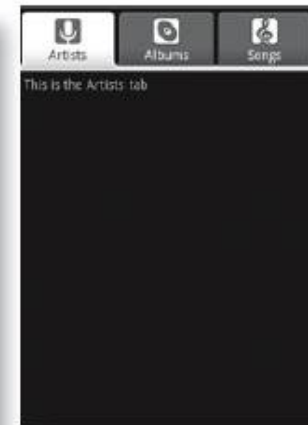
TableLayout



GridLayout



RelativeLayout

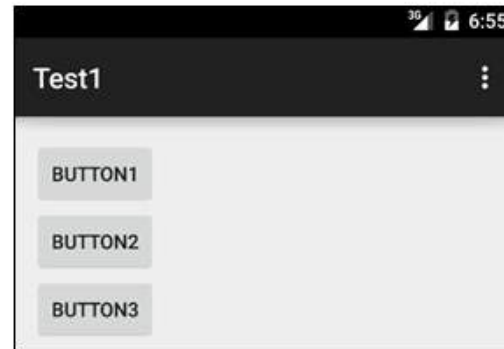
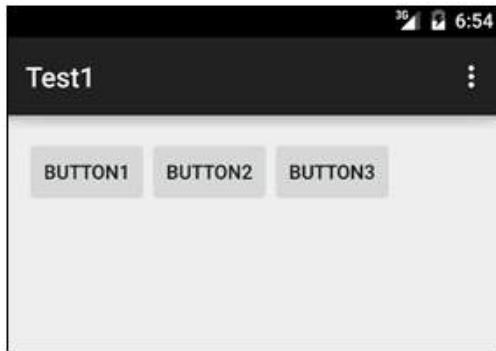


TabLayout

레이아웃 클래스	설명
LinearLayout	하위 뷰(자식)를 수직이나 수평으로 배치
TableLayout	하위 뷰(자식)를 테이블 형태로 배치 (행과 열)
GridLayout	바둑판 모양으로 배치
RelativeLayout	상대적으로 배치 (예: 다른 뷰의 왼쪽, 오른쪽)
TabLayout	탭을 이용하여 겹쳐진 하위 뷰 중 하나를 선택할 수 있도록 배치
FrameLayout	모든 하위 뷰(자식)를 좌측 상단에 겹치게 배치

# LinearLayout

예제 프로젝트 이름:  
Ch5LinearLayoutTest



속성	관련 메소드	설명
orientation	setOrientation(int)	“horizontal”: 수평 배치, “vertical”: 수직 배치
gravity	setGravity(int)	x축과 y축 상에 하위 뷰를 어떻게 배치할 것인지를 지정
baselineAligned	setBaselineAligned(boolean)	하위 뷰의 하단 기준선 정렬 여부 지정 (true: 정렬)

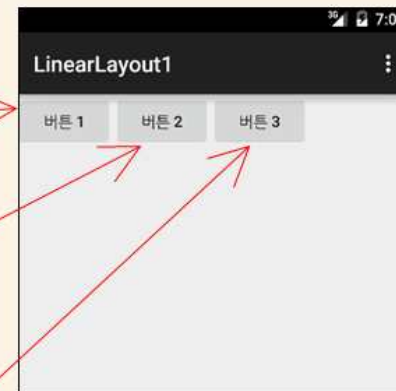
<https://developer.android.com/reference/android/widget/LinearLayout.html>

# horizontal 이용 예제

main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <Button
        android:id="@+id/button01"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="버튼 1"/>
    <Button
        android:id="@+id/button02"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="버튼 2"/>
    <Button
        android:id="@+id/button03"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="버튼 3"/>
</LinearLayout>
```

자식을 수평으로 배치



# gravity 속성 값

상수	값	설명
top	0x30	객체를 컨테이너의 상단에 배치, 크기를 변경하지 않음
bottom	0x50	객체를 컨테이너의 하단에 배치, 크기를 변경하지 않음
left	0x03	객체를 컨테이너의 좌측에 배치, 크기를 변경하지 않음
right	0x05	객체를 컨테이너의 우단에 배치, 크기를 변경하지 않음
center_vertical	0x10	객체를 컨테이너의 수직의 중앙에 배치, 크기를 변경하지 않음
fill_vertical	0x70	객체를 컨테이너의 수직을 채우도록 배치
center_horizontal	0x01	객체를 컨테이너의 수평의 중앙에 배치, 크기를 변경하지 않음
fill_horizontal	0x07	객체를 컨테이너의 수평을 채우도록 배치
center	0x11	객체를 컨테이너의 수평, 수직의 중앙에 배치
fill	0x77	객체가 컨테이너 가득 채우도록 배치

# gravity 속성 예제

main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.
    android:orientation="vertical"
    android:layout width="match parent"
    android:layout height="match parent"
    android:gravity="center"
>
<Button android:id="@+id/button01"
    android:layout width="wrap content"
    android:layout height="wrap content"
    android:text="버튼1"
/>
</LinearLayout>
```

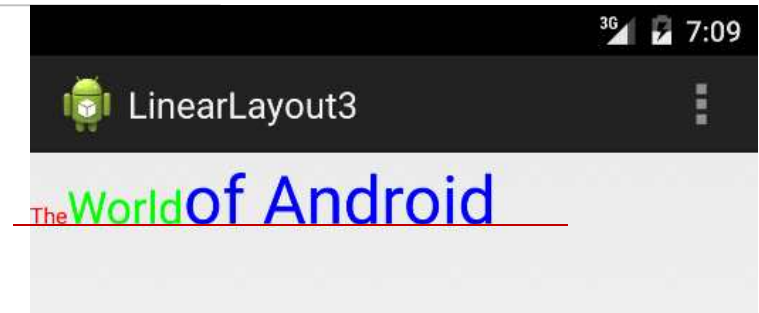
“자식뷰를 중앙에 배치할  
것!”이라는 의미이다.



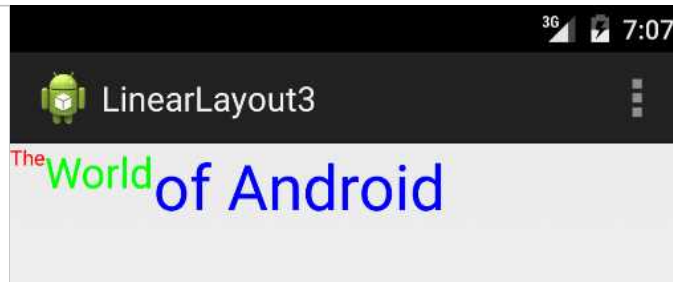


# baselineAligned 예제

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:baselineAligned="true"
>
```

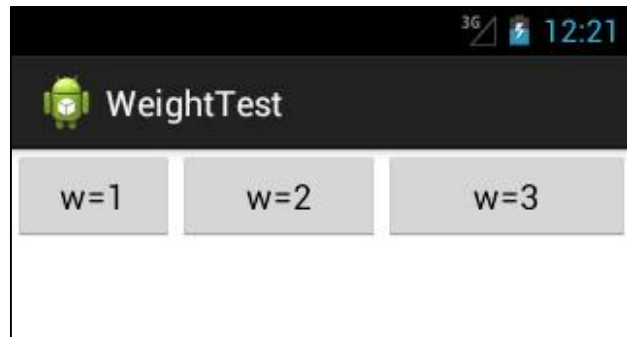


```
<LinearLayout
...
...
    android:baselineAligned="false"
>
```



# weight 속성

- 공간 할당의 가중치를 줄 수 있음
  - 예: 자식 뷰들의 가중치가 각각 1, 2, 3이면,  
남아있는 공간의 1/6, 2/6, 3/6을 각각 할당 받음
- 각 View에서 weight 속성을 설정하지 않으면 0으로 인식
  - 확장되지 않음



남은 공간을 동일하게 나누어서 차지

# weight 예제

- 각 view의  
android:layout\_weight 속성

```
<Button
    android:id="@+id/button01"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="0"
    android:text="클 등록" />

<TextView
    android:id="@+id/textview01"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="0"
    android:text="이름" />

<EditText
    android:id="@+id/edittext01"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="0" />

<TextView
    android:id="@+id/textview02"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="0"
    android:text="의견" />

<EditTextlove
    android:id="@+id/edittext02"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1" />

</LinearLayout>
```

가중치를 0으로 설정

가중치를 1으로 설정

LinearLayout4

클 등록

이름

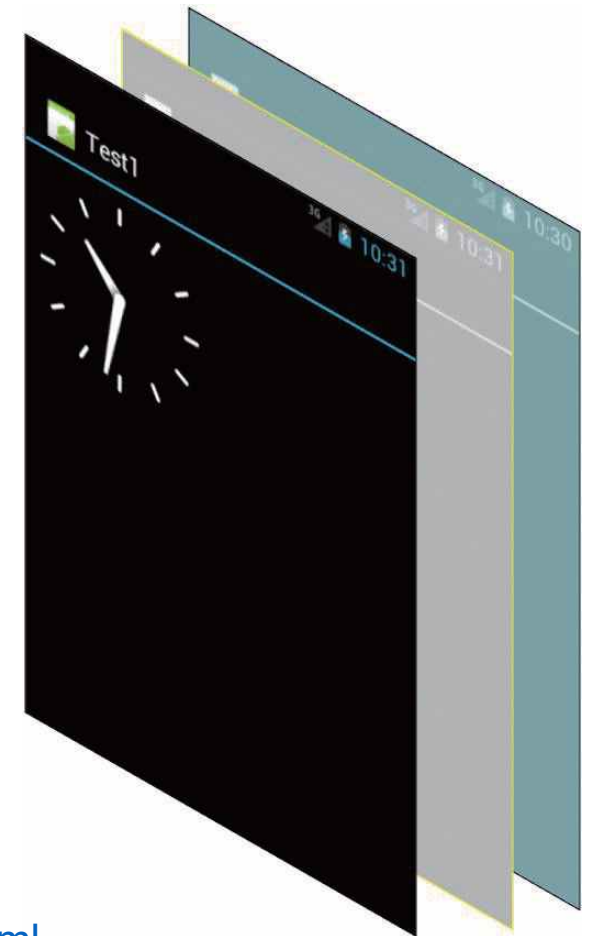
love is a open door

의견

Okay, can I ...

# FrameLayout

- 프레임 레이아웃 안에서 순서 대로 자식 뷰를 화면에 표시
  - 기준점은 화면의 좌측 상단
  - 만약 자식 뷰가 두 개 이상이면 이전에 추가된 뷰 위에 새로운 자식 뷰가 겹쳐져 표시
  - 뷰의 가시성(visibility) 속성을 코드로 변경하여 어떤 순간에 하나의 뷰만 표시하도록 함



<https://developer.android.com/reference/android/widget/FrameLayout.html>

# FrameLayout 예제

*main.xml*

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout width="match parent"
  android:layout height="match parent" >
  <ImageView
    android:id="@+id/ImageView01"
    android:layout width="wrap content"
    android:layout height="wrap content"
    android:src="@drawable/marketbaq"
  />
  <Button
    android:id="@+id/Button01"
    android:layout width="wrap content"
    android:layout height="wrap content"
    android:text="여기를 누르세요"
  />
</FrameLayout>
```



# TableLayout

- 자식 뷰들을 테이블 형태로 배치
  - 하나의 테이블에는 여러 개의 TableRow로 구성
  - 각 TableRow에 여러 개의 셀이 포함. 하나의 셀에는 하나의 뷰 객체

```
<TableLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout width="match parent"
    android:layout height="match parent">

    <TableRow>
        <TextView android:text="주소"/>
        <EditText android:text="서울시종로구 120"/>
    </TableRow>

    <TableRow>
        <TextView android:text="이름"/>
        <EditText android:text="홍길동"/>
    </TableRow>

    <TableRow>
        <Button android:text="저장"/>
        <Button android:text="취소"/>
    </TableRow>
</TableLayout>
```



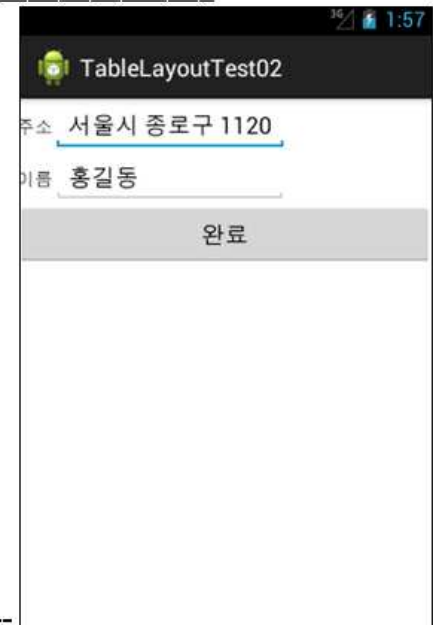


# TableLayout 예제 2

main.xml

```
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TableRow>
        <TextView android:text="주소" />
        <EditText android:text="서울시 종로구 1120" />
    </TableRow>
    <TableRow>
        <TextView android:text="이름" />
        <EditText android:text="홍길동" />
    </TableRow>
    <Button android:text="완료" />
</TableLayout>
```

TableRow 안에 넣지  
않더라도 뷰 객체가  
하나의 행으로 배치



<https://developer.android.com/reference/android/widget/TableLayout.html>

# RelativeLayout

- 자식 뷰의 위치를 부모 뷰나 다른 자식 뷰 들에 상대적으로 지정
  - 먼저 등장하는 뷰를 기준으로 상대적인 위치를 결정하게 됨
  - 따라서 기준이 되는 뷰는 먼저 나와야 함

속성	설명
layout_above	만약 <b>true</b> 이면 현재 뷰의 하단을 기준 뷰의 위에 일치시킨다.
layout_below	현재 뷰의 상단을 기준 뷰의 하단에 위치시킨다.
layout_centerHorizontal	수평으로 현재 뷰의 중심을 부모와 일치시킨다.
layout_centerInParent	부모의 중심점에 현재 뷰를 위치시킨다.
layout_centerVertical	수직으로 현재 뷰의 중심을 부모와 일치시킨다.
layout_toLeftOf	현재 뷰의 우측단을 기준 뷰의 좌측단에 위치시킨다.
layout_toRightOf	현재 뷰의 좌측단을 기준 뷰의 우측단에 위치시킨다.

<https://developer.android.com/reference/android/widget/RelativeLayout.html>



# RelativeLayout 예제

예제 프로젝트 이름:  
Ch5RelativeLayoutTest

main.xml

```
<?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">
    <TextView
        android:id="@+id/address"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:text="주소를 입력하세요" />
    <EditText
        android:id="@+id/input"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@android:drawable/editbox_background"
        android:layout_below="@id/address" />
    <Button
        android:id="@+id/cancel"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/input"
        android:layout_alignParentRight="true"
        android:layout_marginLeft="10dip"
        android:text="취소" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_toLeftOf="@id/cancel"
        android:layout_alignTop="@id/cancel"
        android:text="확인" />
</RelativeLayout>
```



address 아래에 배치

input 아래에 배치

cancel의 왼쪽에 배치