

# User Interfaces 기초

Mobile Software


2019 Fall

# developer.android.com

## 메인 페이지

### 앱 빌드 시작하기


숙련된 개발자이든 Android 앱을 처음 만드는 초보자이든 상관없이 다음의 몇 가지 리소스를 사용하여 시작할 수 있습니다.

[다운로드](#)

### 개발자 가이드

여기에는 앱을 빌드하는 방법, 다양한 화면에 맞게 변하는 레이아웃을 빌드하는 방법, 데이터를 로컬 데이터베이스에 저장하는 방법, 기기 센서 및 카메라를 사용하는 방법 등등을 알려주는 광범위한 문서가 있습니다.


[시작하기](#)



#### 샘플 코드

이 샘플 프로젝트를 사용하여 개발을 시작해 보세요.


[샘플 보기](#)



#### 앱 테스트

출시 전에 앱의 동작과 사용성을 확인하세요.


[테스트 방법 알아보기](#)



#### 품질 가이드라인

이 디자인 및 동작 가이드라인에 따라 고품질의 앱을 빌드하세요.

[가이드라인 보기](#)



#### Google Play에 배포

글로벌 사용자에게 접근하여 수익 창출

[PLAY 알아보기](#)

# Android 개발자 사이트를 최대한 활용하자!

<http://developer.android.com/docs>

## 시작하기

첫 앱 빌드하기

샘플 코드

API 참조 문서

디자인 가이드라인

Codelab 가이드 [🔗](#)

교육 과정 [🔗](#)

## 핵심 개발자 주제

액티비티	인텐트 및 인텐트 필터	UI 및 탐색
애니메이션 및 전환	이미지 및 그래픽	오디오 및 비디오
백그라운드 작업	앱 데이터 및 파일	사용자 데이터 및 신원
사용자 위치	터치 및 입력	카메라
CameraX	센서	연결
RenderScript	웹 기반 콘텐츠	인스턴트 앱
모든 개발자 가이드 보기		

### Android Codelab

Codelab은 짧은 자기 주도형 자습서로 특정 유형의 앱을 만들거나 기존 앱에 새 기능을 추가하는 과정을 안내합니다.

[CODELAB 사용해 보기](#)

### 온라인 동영상 과정

Google은 Udacity와 협력하여 온라인 동영상상을 통해 Android 개발을 가르칩니다. 일부 과정이 무료로 제공되며 Nanodegree 프로그램에 등록할 수도 있습니다.

[UDACITY로 이동](#)

### 교육 과정

이 온라인 교육 과정에서는 Android 앱을 빌드하는 방법을 차근차근 안내해 줍니다. 신규 개발자와 숙련된 개발자 모두를 위한 과정이 모두 제공됩니다.

[과정 보기](#)

# API 참조 문서 활용

<http://developer.android.com/docs>

시작하기

첫 앱 빌드하기

샘플 코드

API 참조 문서

디자인 가이드라인

Codelab 가이드

교육 과정

Overview

Android Platform

Android Support Library

AndroidX

Architecture Components

Databinding Library

Constraint Layout Library

Material Components

Test Support Library

Wearable Library

Play Billing Library

Play Core Library

Play Install Referrer Library

Android Things

Kotlin API Reference (Preview)

## Package Index

These are the API packages. See all [API classes](#).

<a href="#">android</a>	Contains resource class permissions for system
<a href="#">android.accessibilityservice</a>	The classes in this package provide an alternative or augment
<a href="#">android.accounts</a>	
<a href="#">android.animation</a>	These classes provide

<a href="#">android.webkit</a>	Provides tools for browsing the web.
<a href="#">android.widget</a>	The widget package contains (mostly visual)

<a href="#">BaseExpandableListAdapter</a>	Base class for a class that provides data for a
<a href="#">Button</a>	A user interface class that displays a button
<a href="#">CalendarView</a>	This class is a subclass of <a href="#">TextView</a>

# 샘플 코드

TECHNOLOGY ▾ LANGUAGE ▾ RESET

☆☆☆☆☆

15 30 60 per page < 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 > 216 results

JAVA FEATURED  
**AppShortcuts**  
This sample demonstrates how to use the Launcher Shortcuts API introduced in Android 7.1 (API 25). This API allows an...

JAVA FEATURED  
**AutofillFramework**  
This sample demonstrates the use of the Autofill Framework. It includes implementations of client Activities with...

KOTLIN FEATURED  
**AutofillFramework (Kotlin)**  
This sample demonstrates the use of the Autofill Framework. It includes implementations of client Activities with...

JAVA FEATURED  
**DownloadableFonts**  
This sample demonstrates how to use the Downloadable Fonts feature introduced in Android O. Downloadable Fonts is a featur...

KOTLIN FEATURED  
**DownloadableFonts (Kotlin)**  
This sample demonstrates how to use the Downloadable Fonts feature introduced in Android O. Downloadable Fonts is a featur...

JAVA FEATURED  
**EmojiCompat**  
This sample demonstrates usage of EmojiCompat support library. You can use this library to prevent your app from...

KOTLIN FEATURED  
**EmojiCompat (Kotlin)**  
This sample demonstrates usage of EmojiCompat support library. You can use this library to prevent your app from...

JAVA FEATURED  
**JobScheduler**  
Demonstration of the JobScheduler API, which provides an interface for scheduling background tasks when certain tasks apply.

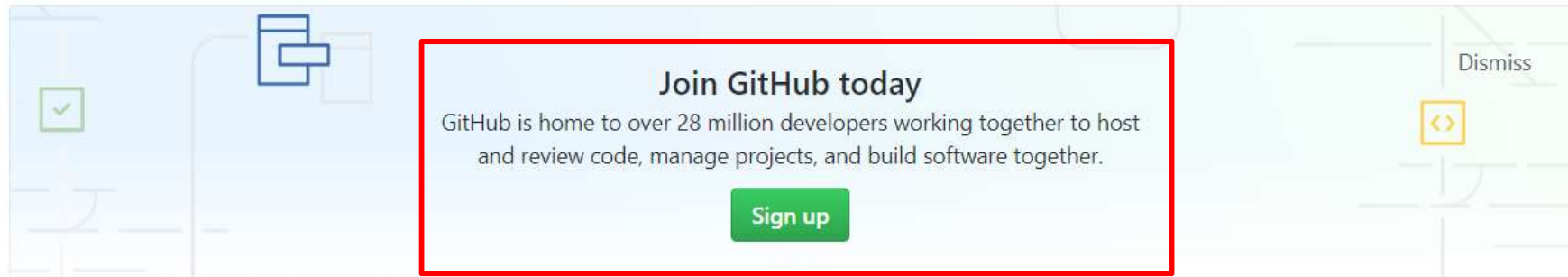
JAVA FEATURED  
**NotificationChannels**  
Demonstration of using channels to categorize notifications by topic. This feature was added in Android O, and allow...

KOTLIN FEATURED  
**NotificationChannels (Kotlin)**  
Demonstration of using channels to categorize notifications by topic. This feature was added in Android O, and allow...

JAVA FEATURED  
**PictureInPicture**  
This sample demonstrates basic usage of Picture-in-Picture mode for handheld devices. The sample plays a video. The...

KOTLIN FEATURED  
**PictureInPicture (Kotlin)**  
This sample demonstrates basic usage of Picture-in-Picture mode for handheld devices. The sample plays a video. The...

# 샘플 : AppShortCuts



*No description, website, or topics provided.*

7 commits1 branch0 releases1 contributorApache-2.0

Branch: master ▾New pull requestFind fileClone or download ▾

[android-AppShortcuts](#) / [app](#) / [src](#) / [main](#) / [java](#) / [com](#) / [example](#) / [android](#) / [appshortcuts](#) /

google-automerger Auto-update

Latest commit 3a78cfc on 20 Oct 2016

..		
<a href="#">Main.java</a>	Auto-update	2 years ago
<a href="#">MyReceiver.java</a>	Auto-update	2 years ago
<a href="#">ShortcutHelper.java</a>	Auto-update	2 years ago
<a href="#">Utils.java</a>	Auto-update	2 years ago

# 첫 앱 빌드하기

## App Basics

### Introduction

#### Build your first app

##### Overview

Create an Android project

Run your app

Build a simple user interface

Start another activity

#### App fundamentals

- ▶ App resources
- ▶ App manifest file
- ▶ App permissions

#### Devices

- ▶ Device compatibility
- ▶ Wear
- ▶ Android TV
- ▶ Android for Cars

Android Developers > Docs > 안내

☆☆☆☆☆

## 첫 앱 빌드

다음 페이지에서는 간단한 Android 앱을 빌드하는 방법을 설명합니다. Android 스튜디오로 'Hello World' 프로젝트를 만들고 실행하는 방법에 대해 배웁니다. 그런 다음, 사용자 입력을 받아 앱의 두 번째 화면을 여는 새로운 앱 인터페이스를 만듭니다.

시작하기 전에, Android 앱에 대해 알아야 할 두 가지 기본 개념이 있습니다.

앱은 여러 개의 진입 지점을 제공합니다.

Android 앱은 여러 구성요소의 조합으로 빌드되며, 이러한 구성요소는 개별적으로 호출될 수도 있습니다. 예를 들어 **활동**은 사용자 인터페이스를 제공하는 하나의 앱 구성요소 유형입니다.

앱은 여러 가지 기기에 맞게 변경됩니다.

Android를 사용하면 다양한 기기에 다양한 리소스를 제공할 수 있습니다. 예를 들어 다양한 화면 크기에 따라 다양한 레이아웃을 만들 수 있습니다. 그러면 시스템이 현재 기기의 화면 크기에 따라 사용할 레이아웃을 결정합니다.

# 교육 과정

Find the training option that's right for you.



COURSE OVERVIEWS

COURSE EXTRAS

## New! Android Kotlin Fundamentals

 Codelab-based

In our free, self-paced [Android Kotlin Fundamentals](#) training, you learn basic Android programming concepts using the Kotlin programming language, and you build a variety of apps. Each codelab includes a tutorial with solution code in GitHub.



Training level: Intermediate

START COURSE



## Android Kotlin Fundamentals Course

Android Kotlin Fundamentals leads you through a series of codelabs that teach you the fundamentals of building Android apps in Kotlin.

A-Z RECENT DURATION

Android Kotlin Fundamentals: Welcome to the course

Updated Aug 13, 2019



Start

Android Kotlin Fundamentals 01.0: Install Android Studio

Updated May 28, 2019



Start

Android Kotlin Fundamentals 01.1: Get started

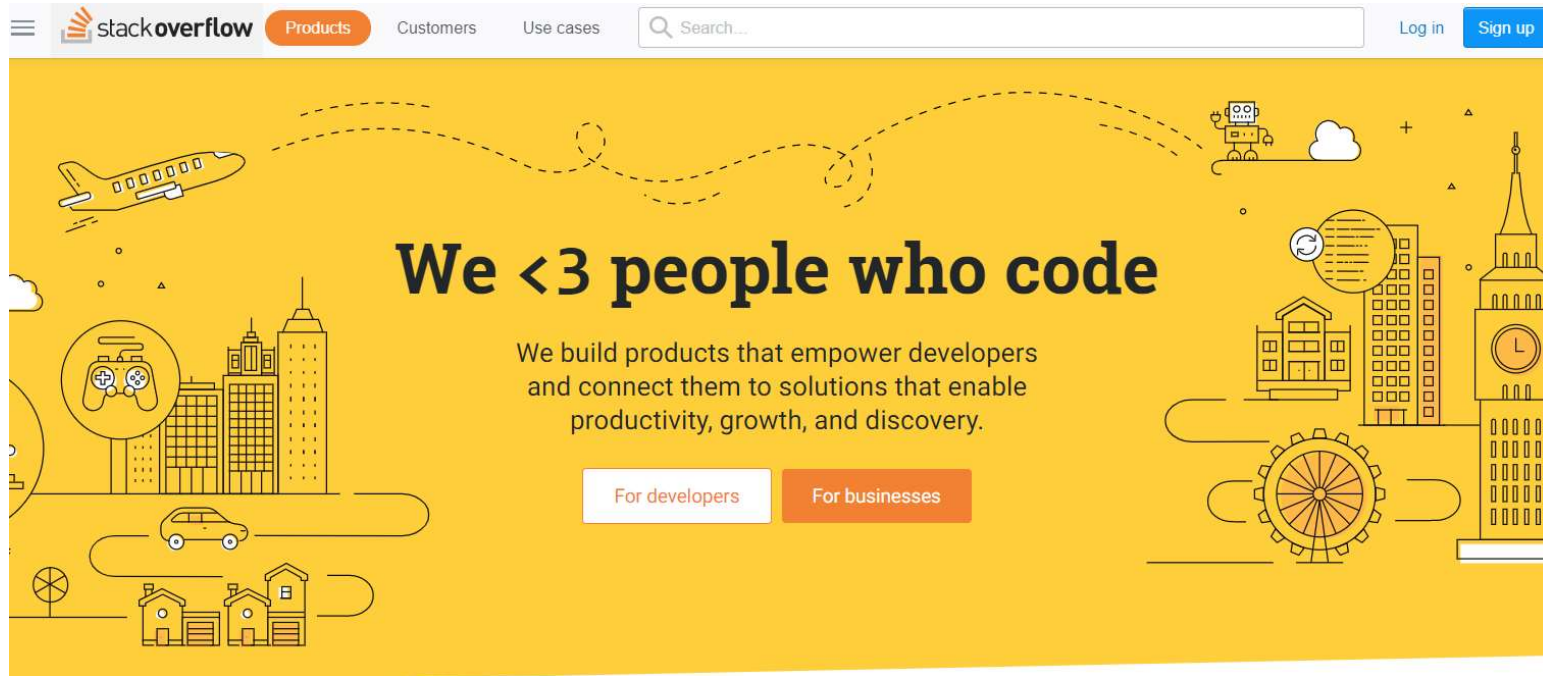
Updated May 28, 2019



Start



# 질문은 stackoverflow 활용

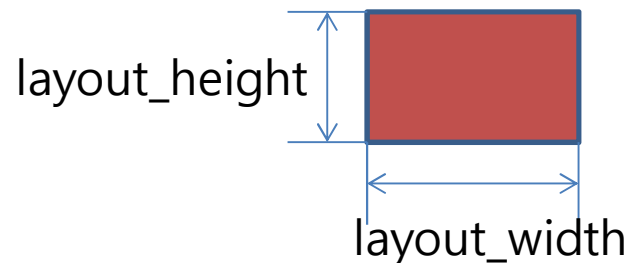


## For developers, by developers

Stack Overflow is an **open community** for anyone that codes. We help you get answers to your toughest coding questions, share knowledge with your coworkers in private, and find your next dream job.

# View 의 크기

- view의 크기 : View의 **높이**, **폭**을 지정
  - 높이 : **layout\_height**, 폭 : **layout\_width**
  - View의 모양에 관계없이 view가 차지하는 공간은 직사각형
- 값 지정(**필수** – 지정하지 않으면 오류 발생)
  - **match\_parent** : view를 포함하는 ViewGroup의 크기와 같음
  - **wrap\_content** : view에 포함된 내용 크기와 같음
  - 절대 값 : 숫자
    - 숫자 뒤에 단위 **dp**를 붙여야 함
    - **0dp** → 여유 공간에 따라 크기가 정해짐



# 화면 밀도 (screen density)

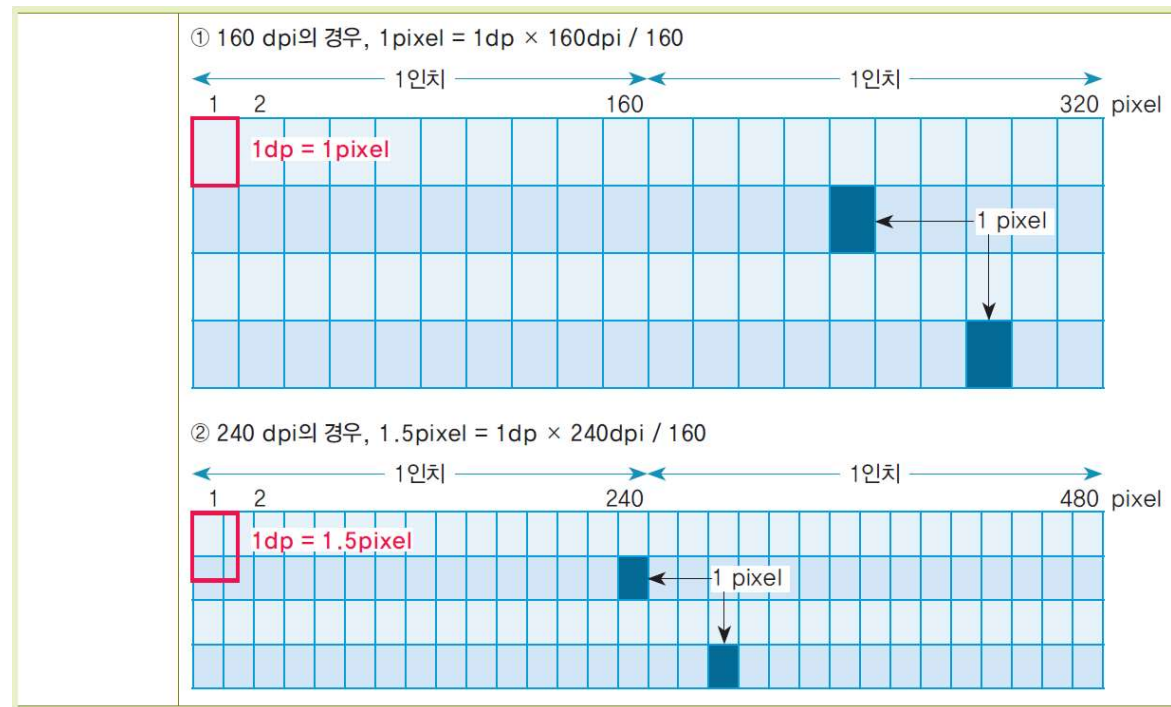
화면 밀도 (screen density)	<ul style="list-style-type: none"> <li>■ 화면 면적 당 픽셀 수, 대개 1인치 당 픽셀 수를 나타내는 <u>dpi(dots per inch)</u>를 사용함</li> <li>■ 안드로이드의 6가지 화면밀도 분류와 dpi수</li> </ul>	
	화면밀도 분류	dpi
	ldpi(low)	120
	mdpi(medium)	160
	hdpi(high)	240
	xhdpi(extra-high)	320
	xxhdpi(extra-extra-high)	480
	xxxhdpi(extra-extra-extra-high)	640

px	pixel(picture element)	디스플레이 기본 단위
dp(dip)	density-independent pixel	160dpi 밀도를 기준으로 한 픽셀.
sp(sip)	scale-independent pixel	글꼴 크기에 따라 픽셀 수가 달라짐. 텍스트 크기를 지정할 때 사용.

참고 <http://angrytools.com/android/pixelcalc>

# dp 와 px 관계

- dp : screen density와 무관한 가상 pixel
$$\text{px} = \text{dp} \times \text{dpi} / 160$$
  - Screen density가 medium(=160dpi)  $\rightarrow 1\text{dp} = 1\text{px}$
  - Screen density가 high(=240dpi)  $\rightarrow 1\text{dp} = 1.5\text{px}$



# 실습 준비

- 새 프로젝트 생성
  - Activity : **Empty Activity**
  - Application name : **Basic UI**
  - Package name : **edu.ourincheon.basicui**
  - Minimum API level : **API 24** (Nougat)
  - Activity name : **MainActivity.kt**
  - Layout name : **activity\_main.xml**
- 자동 생성된 **레이아웃 파일**은 기본 layout으로 **ConstraintLayout** 이 지정되어 있음.
  - **ConstraintLayout** 대신 **LinearLayout** 으로 변경
  - **TextView** 도 삭제

# 실습 1: 단위 변환 (dp → px)

Type	Name	Play Store	Resolution	API
	Pixel 2 API 28		1080 × 1920 420dpi	28

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    val dp = 16
    val scale: Float = resources.displayMetrics.density
    val px = (scale*dp + 0.5).toInt()

    Log.i("Unit Conversion>>>", "density = $scale")
    var dpi: Int = (scale * 160).toInt()
    Log.i("Unit Conversion>>>", "dpi = $dpi")
    Log.i("Unit Conversion>>>", "dp = $dp, px = $px")
}
```

Logcat

Emulator Pixel\_2\_API\_28 Android    edu.ourincheon.basicui (8680)    Verbose    Unit

```
2019-09-09 19:08:48.589 8680-8680/edu.ourincheon.basicui I/Unit Conversion>>>: density = 2.625
2019-09-09 19:08:48.590 8680-8680/edu.ourincheon.basicui I/Unit Conversion>>>: dpi = 420
2019-09-09 19:08:48.590 8680-8680/edu.ourincheon.basicui I/Unit Conversion>>>: dp = 16, px = 42
```

## 실습 2: 다양한 단위 변환(dp→ px) 방법

```
val scale: Float = resources.displayMetrics.density
```

### Method 1

```
val dp = 16  
val px = (scale*dp + 0.5).toInt()
```

### Method 2

```
val dp = 16  
val px = Math.round(scale*dp)
```

### Method 3

```
val dp: Float = 16f  
val px = TypedValue.applyDimension(  
    TypedValue.COMPLEX_UNIT_DIP, dp,  
    resources.displayMetrics  
)
```

# 색상

- 투명도와 빛의 3원색인 RGB값을 지정
  - “#RRGGBB”
  - “#AARRGGBB”
    - 16진수 사용
    - R-빨간색, G-녹색, B-파란색
    - A(alpha)-투명도
      - 00: 투명, FF: 불투명

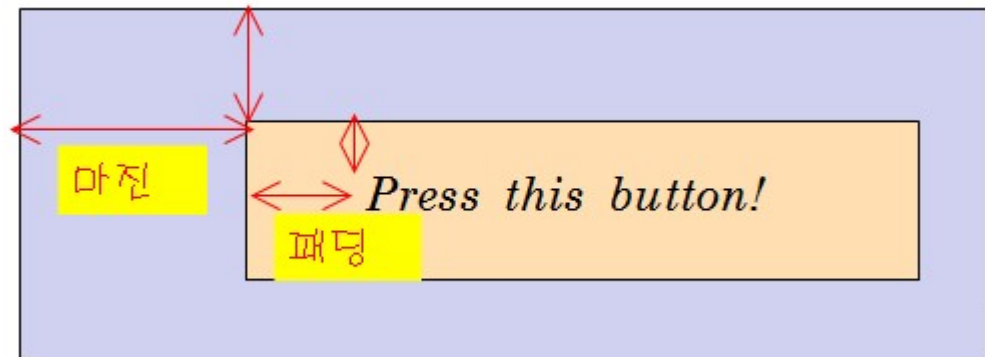
■ 색상표

	#000000		#800080
	#000080		#808000
	#0000ff		#808080
	#008000		#c0c0c0
	#008080		#ff0000
	#00ff00		#ff00ff
	#00ffff		#ffff00
	#800000		#ffffff



# margin과 padding

- **layout\_margin**
  - 부모 container와 자식 view 사이 여백
- **padding**
  - View의 (바깥) 테두리와 view의 실제 내용 사이의 간격
- Padding
  - **paddingLeft**
  - **paddingRight**
  - **paddingTop**
  - **paddingBottom**
- Margin
  - **layout\_marginLeft**
  - **layout\_marginRight**
  - **layout\_marginTop**
  - **layout\_marginBottom**

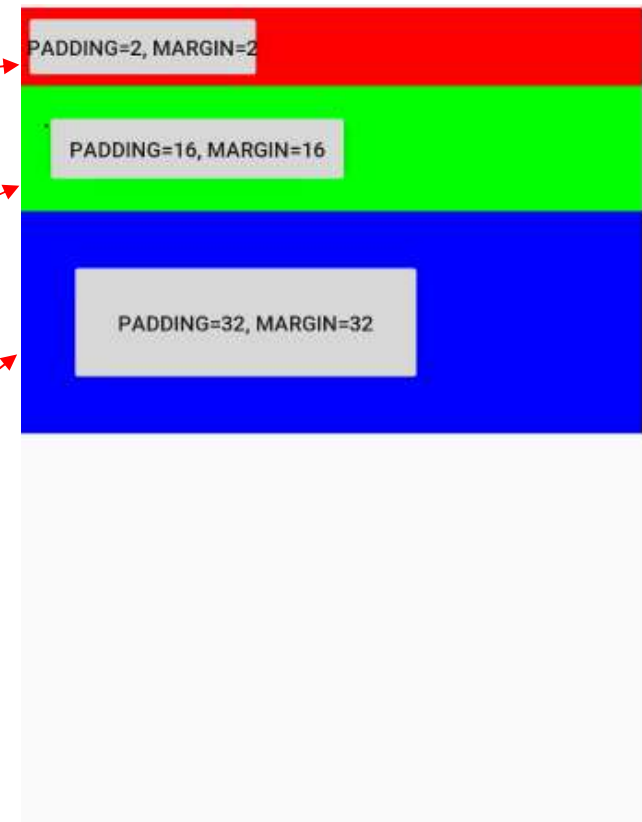


# 실습 3: margin과 padding

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="2dp"
    android:layout_margin="2dp"
    android:text="Padding=2, Margin=2" />
```

```
<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="16dp"
    android:layout_margin="16dp"
    android:text="Padding=16, Margin=16" />
```

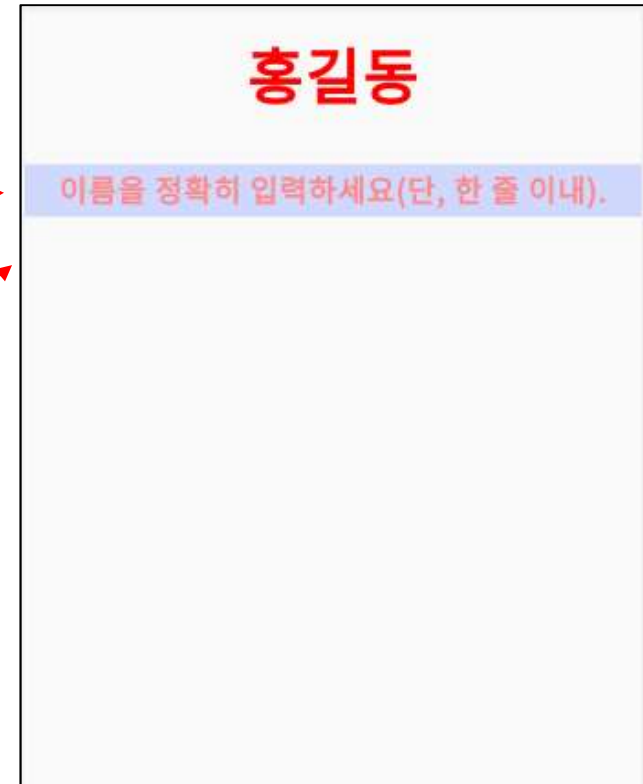
```
<Button
    android:id="@+id/button3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="32dp"
    android:layout_margin="32dp"
    android:text="Padding=32, Margin=32" />
```



# 실습 4: TextView


```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="홍길동"
    android:textColor="#ffff0000"
    android:textSize="40dp"
    android:textStyle="bold"
```

```
<TextView
    android:id="@+id/TextView01"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:background="#88aabbff"
    android:padding="4dp"
    android:text="이름을 정확히 입력하세요(단, 한 줄 이내)."
```



# ImageView

- Icon 과 같은 이미지들을 나타내기 위해 사용

속성	설정 메소드	설명
 <code>android:adjustViewBounds</code>	<code>setAdjustViewBounds(boolean)</code>	drawable의 종횡비를 유지하기 위하여 이미지 뷰의 가로, 세로를 조정
<code>android:cropToPadding</code>		true이면 패딩 안에 맞추어서 이미지를 자른다.
<code>android:maxHeight</code>	<code>setMaxHeight(int)</code>	이미지 뷰의 최대 높이
<code>android:maxLength</code>	<code>setMaxLength(int)</code>	이미지 뷰의 최대 너비
<code>android:scaleType</code>	<code>setScaleType(ImageView.ScaleType)</code>	이미지 뷰의 크기에 맞추어 어떻게 확대나 축소할 것인지 방법 선택
 <code>android:src</code>	<code>setImageResource(int)</code>	이미지 소스
<code>android:tint</code>	<code>setColorFilter(int, PorterDuff.Mode)</code>	이미지 배경 색상

# 실습 5: ImageView 와 ImageButton

- Image 원본 : [starry\\_night.png](#), [ok\\_btn.png](#)
- View > Resource Manager > drag & drop > import

```
<ImageView  
    android:id="@+id/imageView"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:maxLength="350dp"  
    android:maxLength="290dp"  
    android:adjustViewBounds="true"  
    android:src="@drawable/starry_night"  
    android:layout_gravity="center"  
    android:layout_marginTop="20dp"  
    android:layout_marginBottom="20dp"/>
```

```
<ImageButton  
    android:id="@+id/imageButton"  
    android:layout_width="50dp"  
    android:layout_height="50dp"  
    android:layout_marginLeft="40dp"  
    android:background="@drawable/ok_btn"  
    android:contentDescription="ok button" />
```

