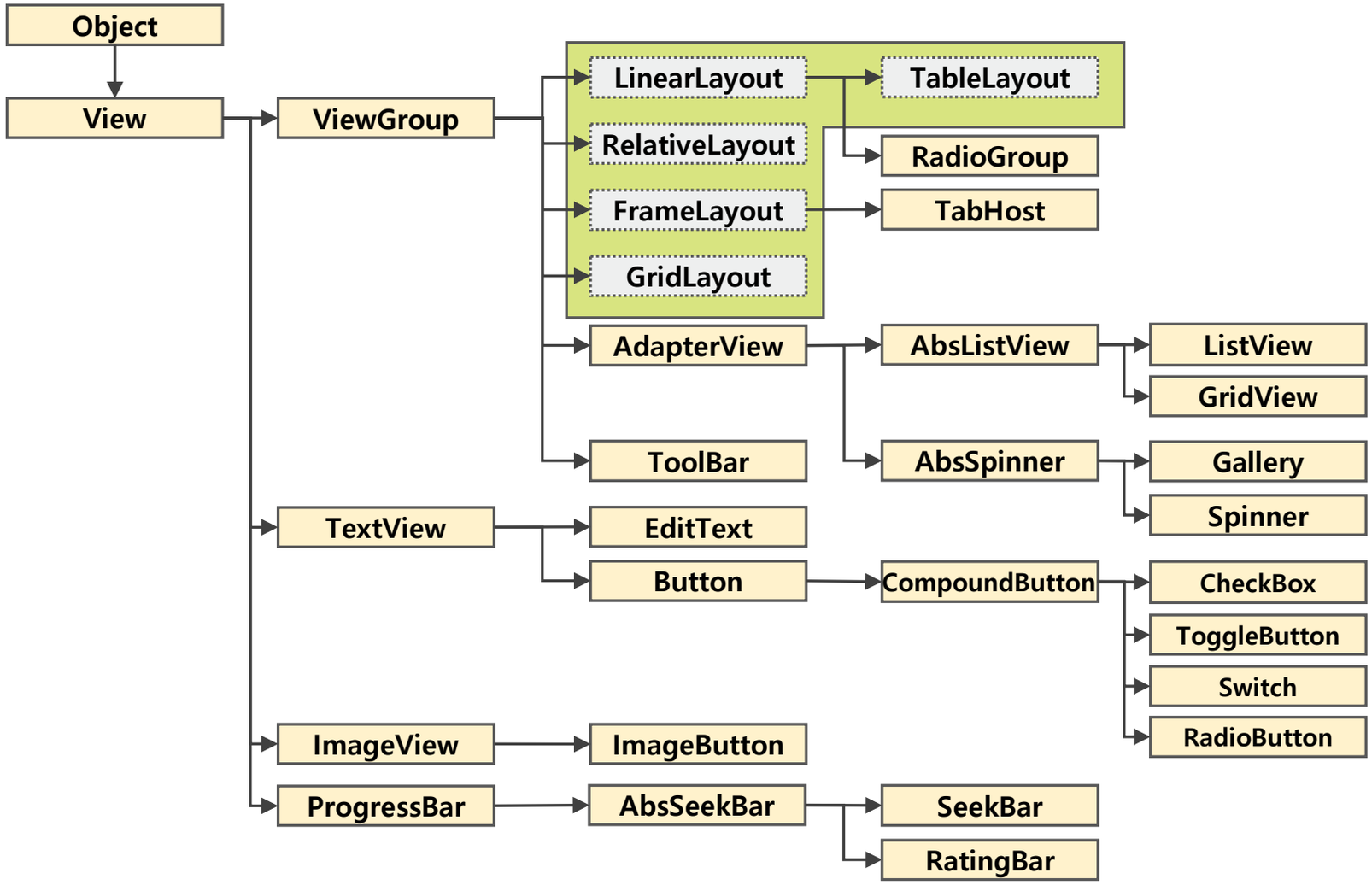
Chapter 04 배치를 담당하는 레이아웃

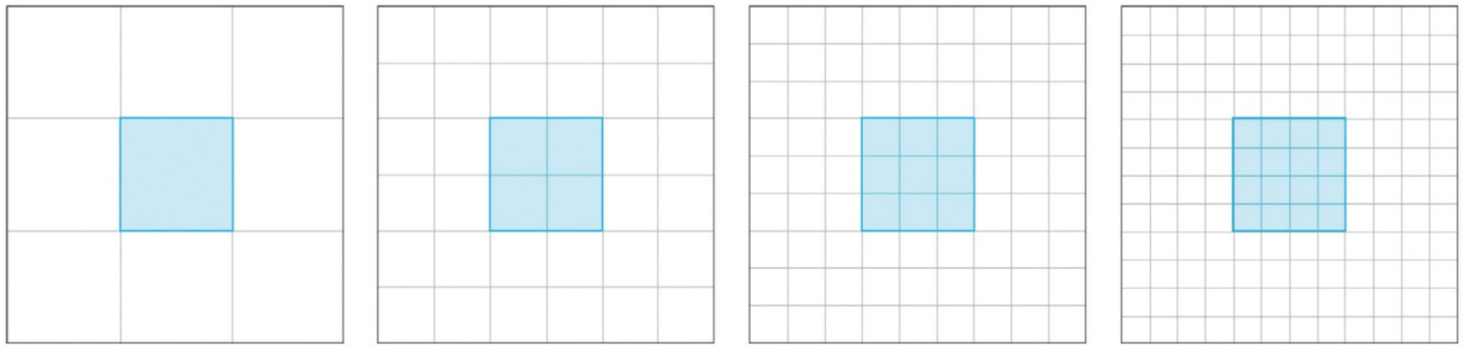
1.1 레이아웃 파일

- 하나의 activity 파일(MainActivity.kt)이 생성
- 하나의 layout 파일(activity_main.xml)이 생성



** 참고: dp의 이해

- dp: 어떤 크기의 화면에서도 동일한 비율로 보이게 하기 위해 안드로이드에서 정의한 좌표 단위
- dip; density independent pixel: 밀로 독립 픽셀
- 화면의 픽셀 미리도와 해상도는 플랫폼에 따라 다르므로
  1인치에 들어가는 픽셀 수를 픽셀 밀도로 계산해서 화면의 크기를 조절함

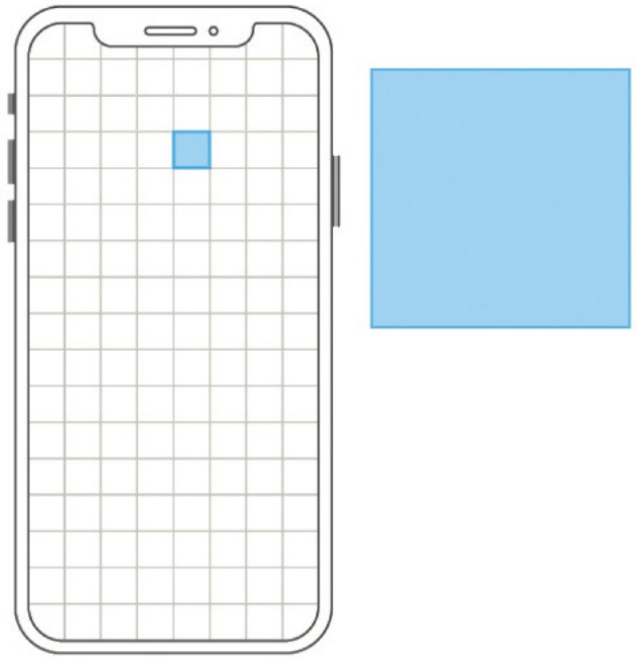


1dp = 1px　　1dp = 2px　　1dp = 3px　　1dp = 4px

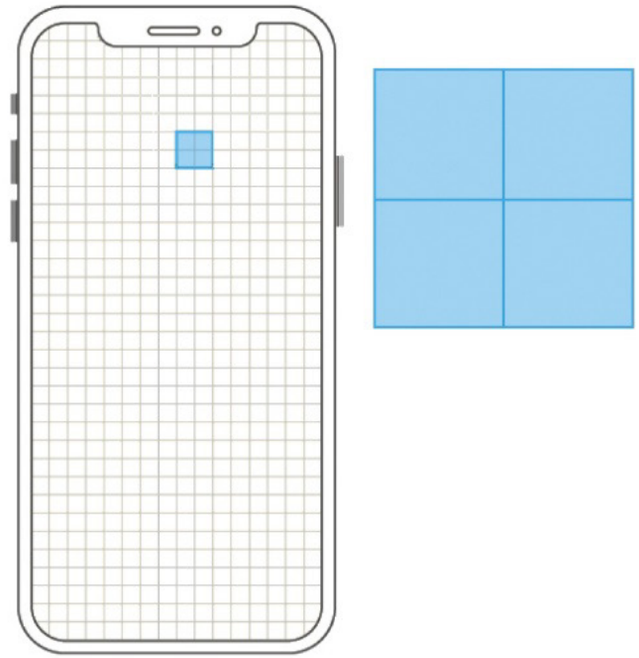
- 해상도가 달라도 크기가 같아 보이게하려면, dp를 사용해서 밀도를 다르게 제작해야 함

출처: 디자인 스타일 가이드로 배우는 UI/UX 디자인 이론과 실무 with 어도비 XD / 이영주 / 한빛미디어

· 해상도가 달라도 크기가 같아 보이게하려면, dp를 사용해서 밀도를 다르게 제작해야 함

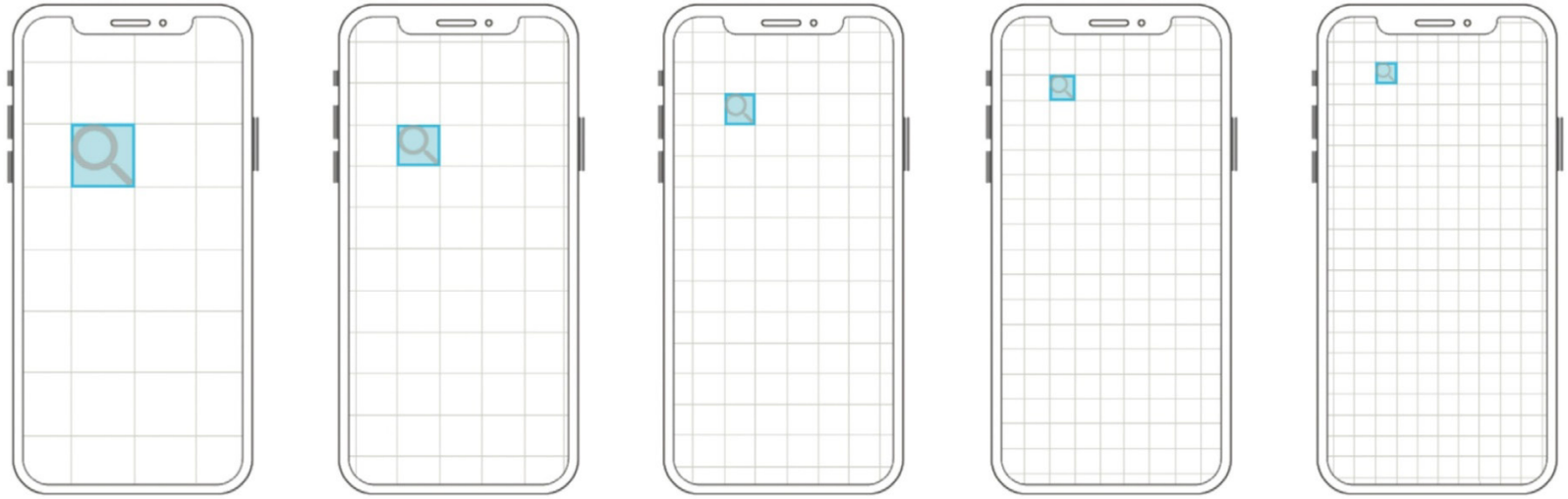


1dp = 1px



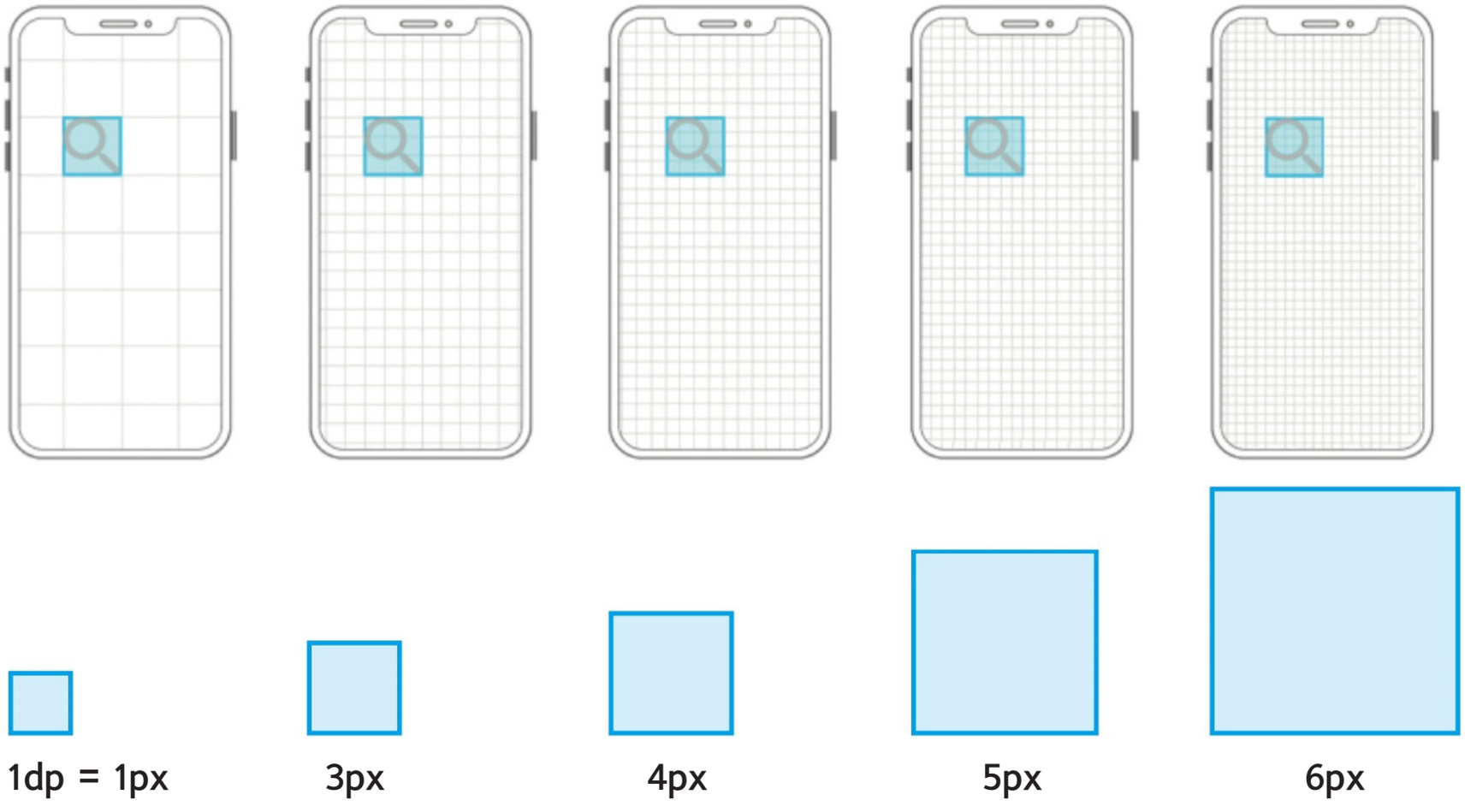
1dp = 2px

· 해상도와 관계없이 동일한 이미지 크기를 적용할 때,
  - 해상도가 높아질 수록 그림이 작아지는 문제 발생



· 해상도에 따라 같은 크기의 이미지를 나타낼 때

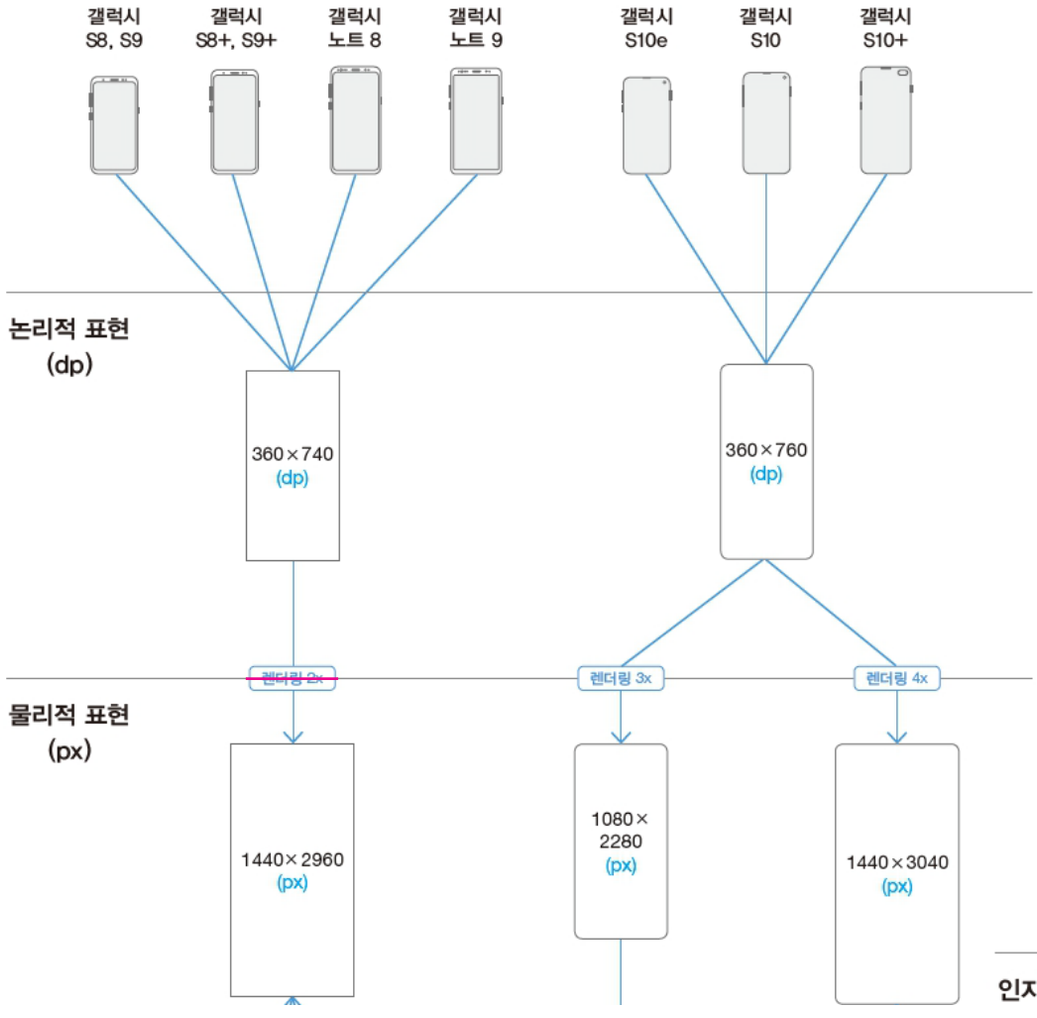
  - 해상도에 따라 다른 밀도 사용

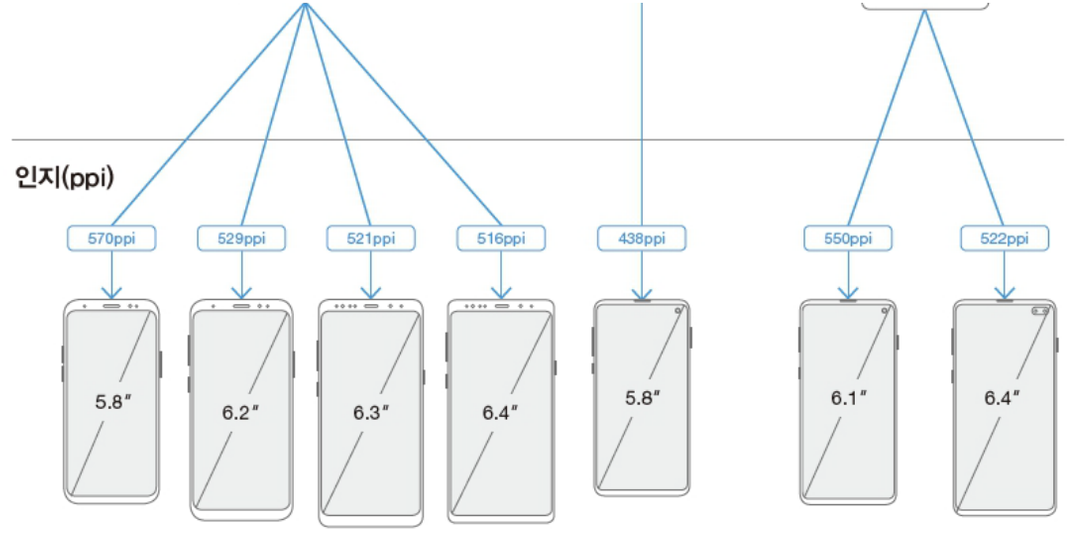


· **래스터화**

- **계산된 픽셀을 물리적 화면에 표시하는 과정**
- **레스트화 단계**

레스터화	설명
1단계 논리적 표현(dp, point)	dp 또는 point는 추상 단위로, 수학적 좌표 공간에서만 의미가 있다. 실제 디자인을 위해서는 래스트화가 필요하다.
2단계 물리적 표현(px)	dp 또는 point로 렌더링되는 과정을 래스터화라고 한다. px은 에 또는 point에 배율을 곱해서 얻을 수 있으며, 이 과정을 거쳐아먄 실제로 디자인할 크기를 얻을 수 있다.
3단계 다운샘플링	디바이스 해상도는 이전 단계에서 랜더링된 이미지보다 낮을 수 있다. 이미지를 표시하기 위해 해상도를 낮추는 다운샘플링, 즉 이미지의 크기 조정이 필요한 단계이다. 이는 아이폰 6+, 6s+, 7+, 8+의 비율이 다른 디바이스보다 약간 작아서 생긴 과정으로 아이폰에만 해당한다.
4단계 인지(ppi)	계산된 픽셀을 물리적 화면에 표시한다. ppi는 1인치에 맞는 픽셀 수와 실제 세계에 나타나는 픽셀 크기를 알려준다.

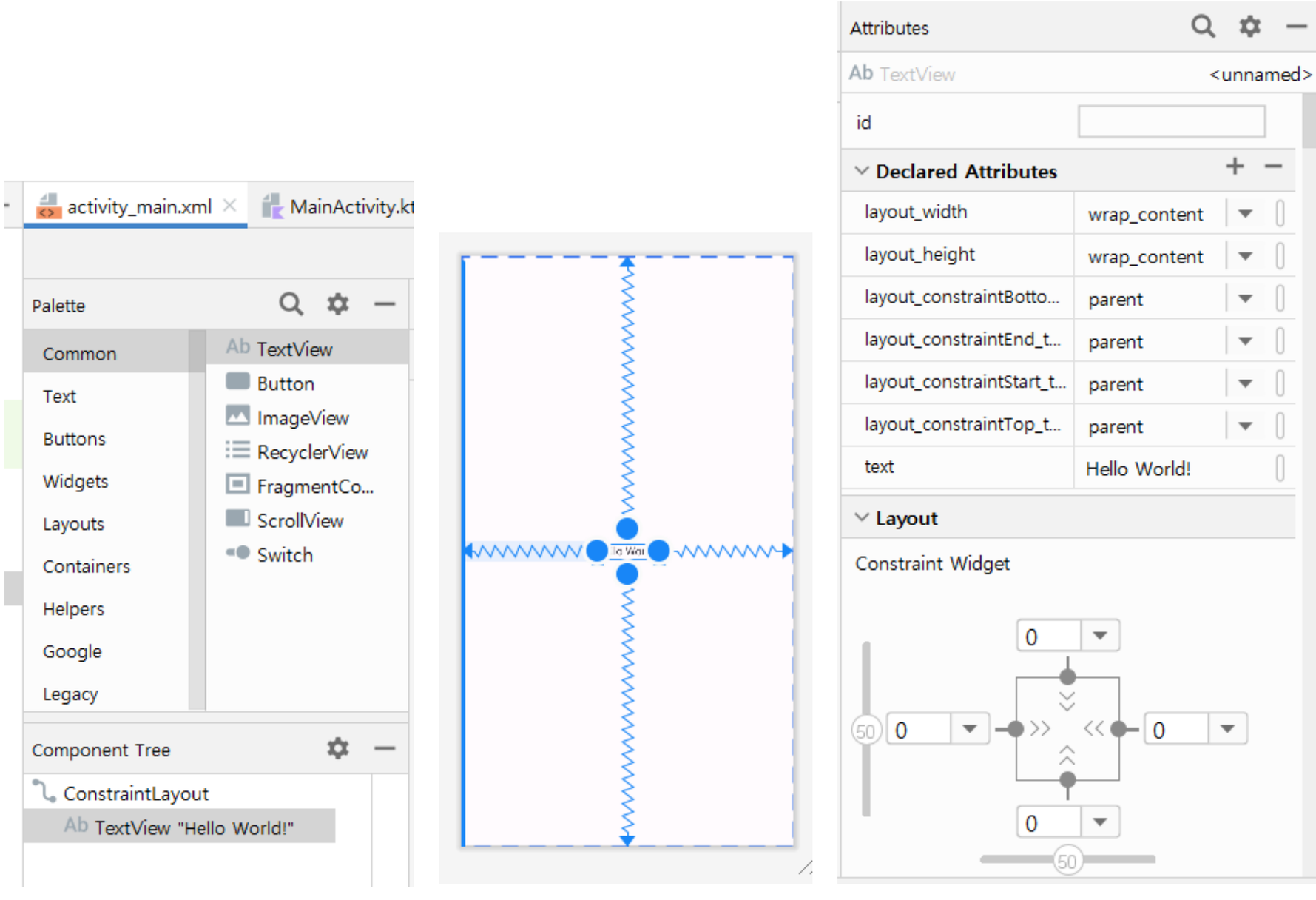

갤럭시 S8, S9
갤럭시 S8+, S9+
갤럭시 노트 8
갤럭시 노트 9
갤럭시 S10e
갤럭시 S10
갤럭시 S10+
논리적 표현 (dp)
360×740 (dp)
360×760 (dp)
렌더링 2x
렌더링 3x
렌더링 4x
물리적 표현 (px)
1440×2960 (px)
1080×2280 (px)
1440×3040 (px)


인지(ppi)
570ppi
529ppi
521ppi
516ppi
438ppi
550ppi
522ppi
5.8″
6.2″
6.3″
6.4″
5.8″
6.1″
6.4″

1.2 컨스트레인트 레이아웃 ConstraintLayout

- 안드로이드의 기본 레이아웃
- 화면의 위젯 사이에 제약조건(constraint) 설정으로 화면을 쉽게 구성

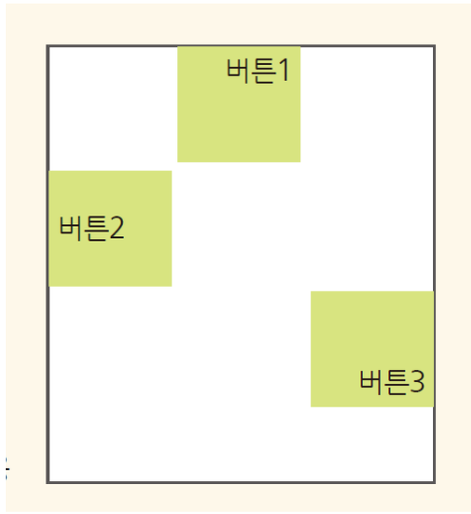
- 화면 용어
  - 위젯 / 콤포넌트
  - 핸들러: 화면에 배치된 위젯을 선택했을 때 표시되는 상하좌우 4개의 원 모양의 점
    - 연결된 핸들러와 연결되지 않은 핸들러 구분(파란색과 흰색)
  - 컨스트레인트: 위젯 주변에 표시되는 주름 무늬 선
  - 앵커포인트: 컨스트레인트가 연결될 수 있는 선(화면 예시에서 위젯의 상하좌우에 주름 선 끝의 화살표 위치)
  - 컨스트레인트 편집기: Attributes 창에 나타나는 Layout(Constraint Widget) 영역


activity_main.xml
MainActivity.kt
Palette
Common
Text
Buttons
Widgets
Layouts
Containers
Helpers
Google
Legacy
TextView
Button
ImageView
RecyclerView
FragmentCo...
ScrollView
Switch
Component Tree
ConstraintLayout
TextView "Hello World!"
Attributes
TextView
<unnamed>
id
Declared Attributes
layout_width
wrap_content
layout_height
wrap_content
layout_constraintBotto...
parent
layout_constraintEnd_t...
parent
layout_constraintStart_t...
parent
layout_constraintTop_t...
parent
text
Hello World!
Layout
Constraint Widget
0
0
0
0
50
50

1.2 리니어 레이아웃 LinearLayout

- 위젯을 가로 또는 세로 한 줄로 배치하기 위한 레이아웃
- 새 프로젝트 생성 후 activity_main.xml 파일에서 레이아웃 변경 필요
  - <androidx.constraintlayout.widget.ConstraintLayout
    ...
  - <LinearLayout
- [Design] 모드에서 Component Tree를 통해 변경 결과 확인

예 다음 그림과 같은 화면을 activity_main.xml 파일을 통해 작성(ex_001)



- LinearLayout 사용
- orientation="vertical"
- 각 버튼의 layout_width=100dp
- 각 버튼의 layout_height=100dp
- 각 버튼에 layout_gravity=true

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">


    <Button
        android:id="@+id/button"
        android:layout_width="200dp"
        android:layout_height="100dp"
        android:layout_gravity="end"
        android:text="오른쪽" />
        <!--  right가 아닌 end로 사용 -->

    <Button
        android:id="@+id/button2"
        android:layout_width="200dp"
        android:layout_height="100dp"
        android:layout_gravity="center"
        android:text="중앙" />

    <Button
        android:id="@+id/button3"
        android:layout_width="200dp"
        android:layout_height="100dp"
        android:layout_gravity="start"
        android:text="왼쪽" />
        <!-- left가 아닌 start로 사용 -->
```

```
    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="46dp"
        android:layout_marginTop="20dp"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:layout_marginBottom="30dp"
        android:textSize="30sp"
        android:textStyle="bold|italic"
        android:text="여기에 결과 표시" />
```

```
package com.android.ex_001

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import com.android.ex_001.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        //setContentView(R.layout.activity_main)
        val binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        binding.button.setOnClickListener() {
            binding.textView.text = binding.button.text
        }

        binding.button2.setOnClickListener() {
            binding.textView.text = binding.button2.text
        }

        binding.button3.setOnClickListener() {
            binding.textView.text = binding.button3.text
        }
    }
}
```

```
android {
    buildFeatures {
        viewBinding = true
    }
    namespace = "com.android.ex_001"
    compileSdk = 33
```

1.2 리니어 레이아웃 LinearLayout(계속)

- orientation
  - vertical / horizontal
- layout_weight
  - 위젯의 가로세로 비율
- layout_gravity
  - start(~~left~~) / center / end(~~right~~)
- gravity