



안드로이드 앱 프로그래밍

Chapter 02

안드로이드 스튜디오와 친숙해지기



이번 장에서는 무엇을 다룰까요?



안드로이드 스튜디오에 대해 먼저 알고 싶어요.



- 시작하기 전에 안드로이드 스튜디오에 대해 알아봐요.
- 뷰란 무엇일까요?
- 처음 보이는 레이아웃을 이용해 화면 배치를 해볼까요?





이번 장에서는 무엇을 다룰까요?



안드로이드 스튜디오에 대해 먼저 알고 싶나요?

- 안드로이드 스튜디오 살펴보기



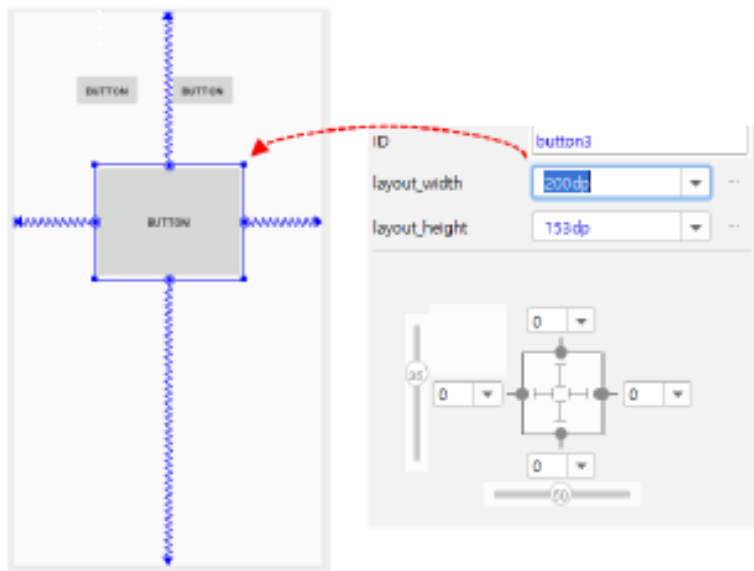
뷰란 무엇일까요?

- 뷰와 뷰의 크기 속성 이해하기



처음 보이는 레이아웃으로 화면 배치를 해볼까요?

- 제약 레이아웃 익히기





강의 주제 및 목차

강의 주제

안드로이드 스튜디오의 기본 사용법을 익히고 화면을 만들어보요.



1

안드로이드 스튜디오 제대로 익히기

2

뷰와 뷰의 크기속성 이해하기

3

레이아웃 기초 익히기

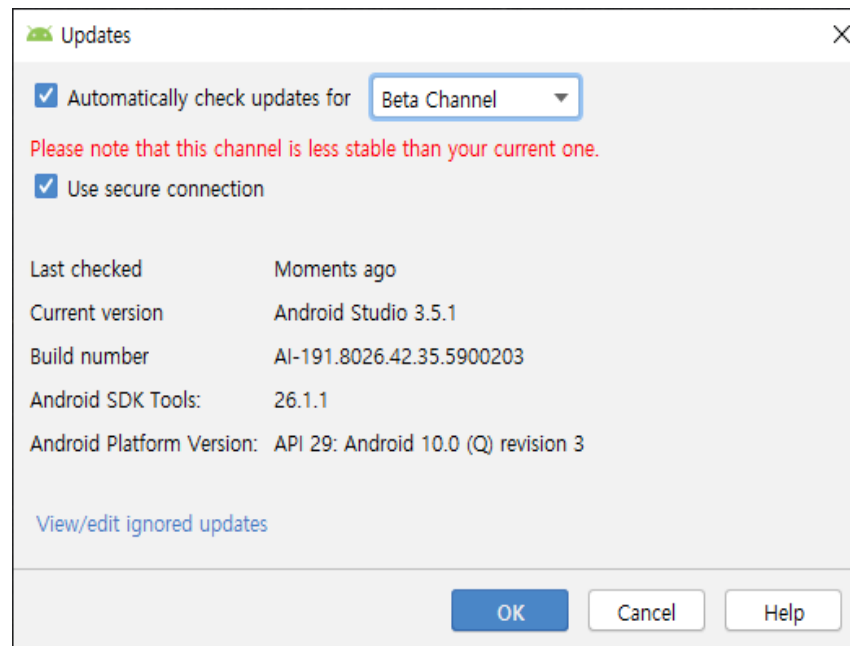
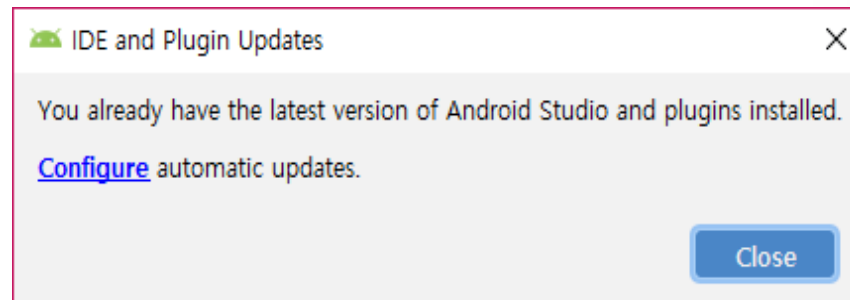
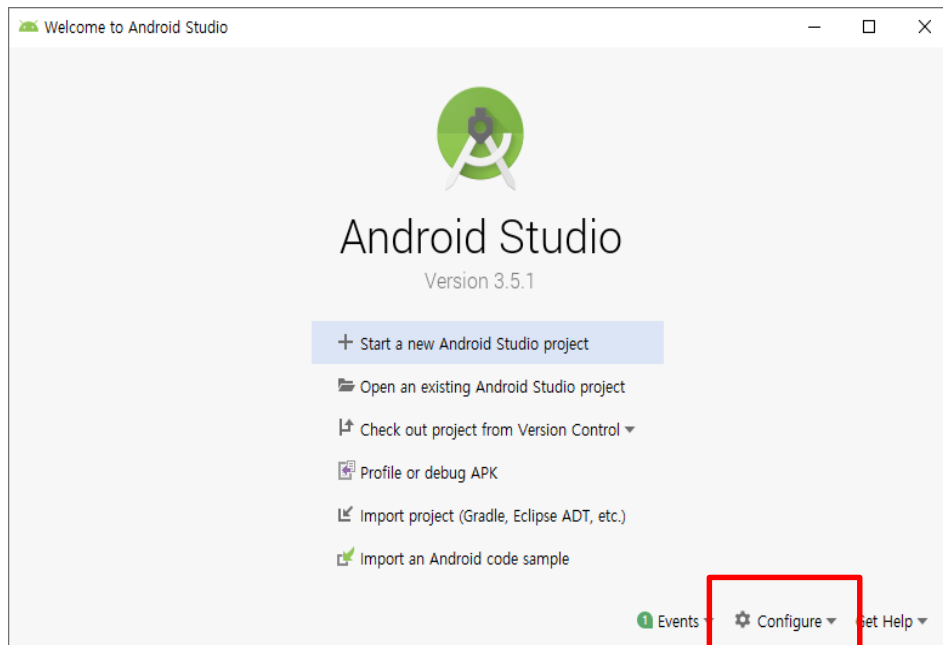
1.

안드로이드 스튜디오 제대로 익히기



시작 화면과 업데이트 여부 확인

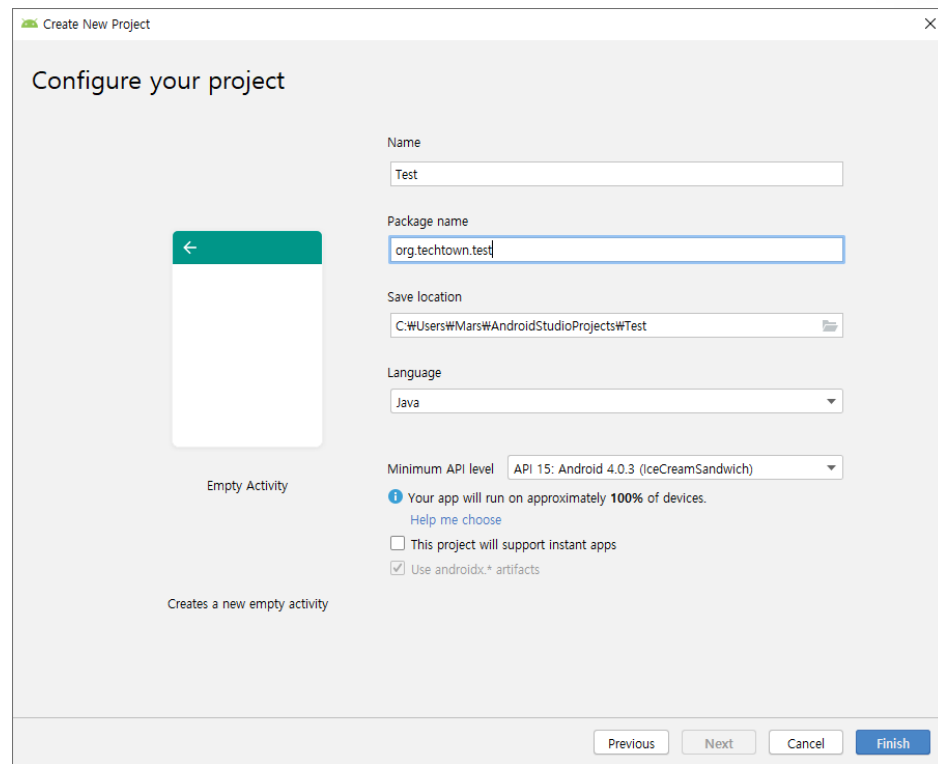
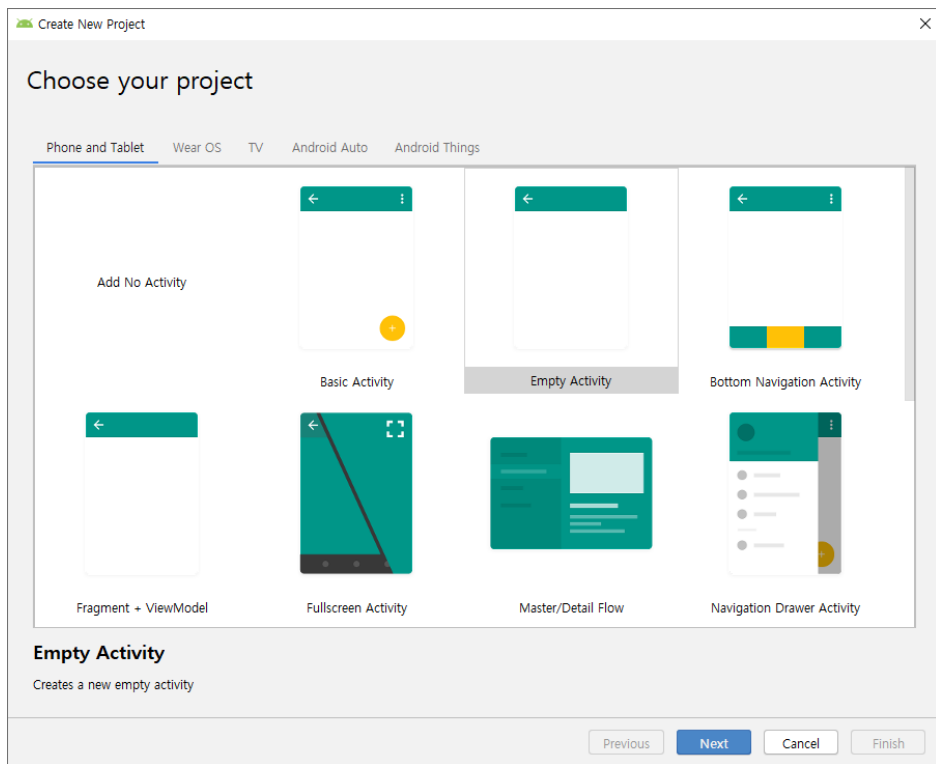
- 시작 화면 하단의 링크를 이용해 업데이트 여부 확인





새로운 프로젝트 생성

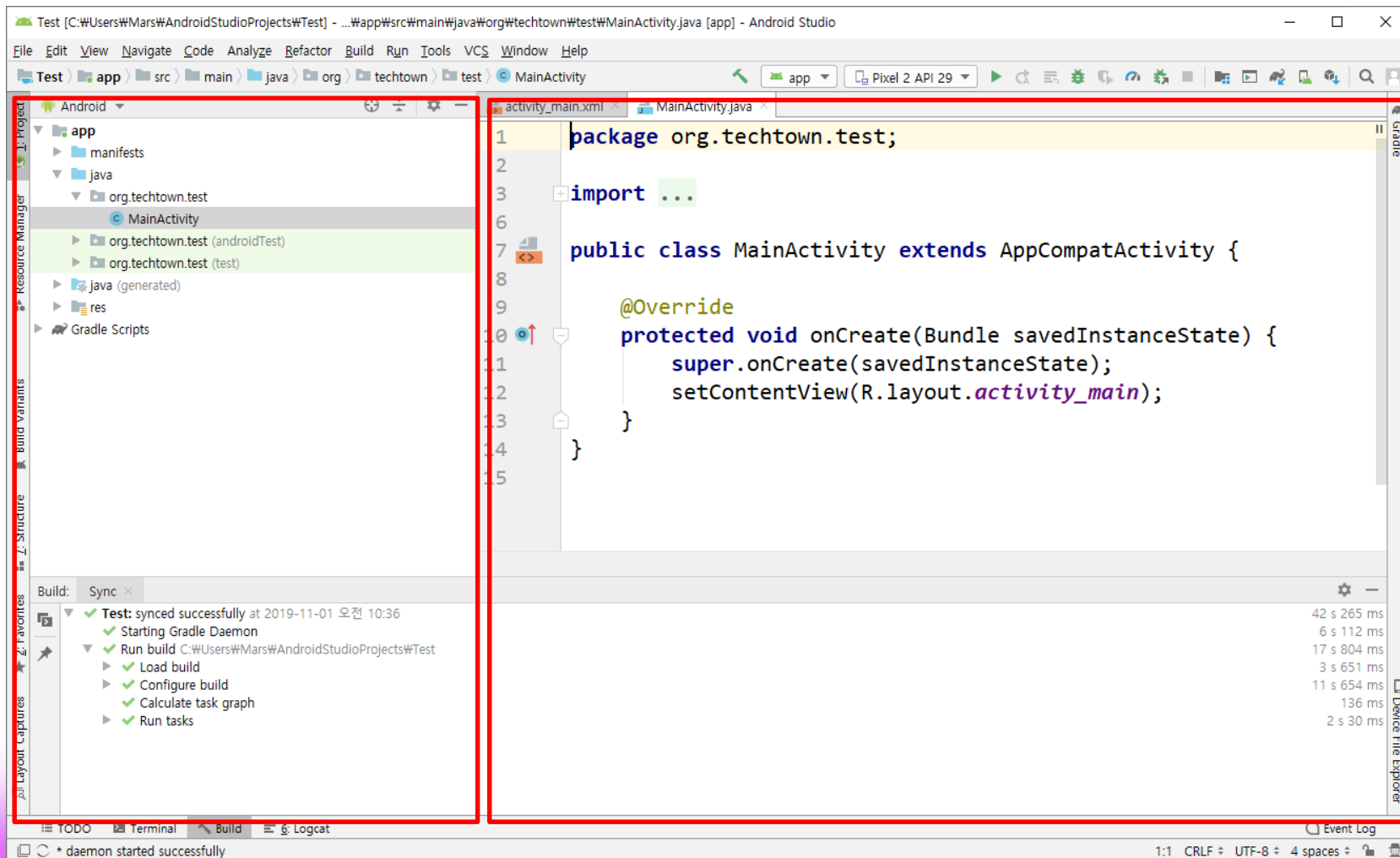
- Test 라는 이름의 새로운 프로젝트 생성





프로젝트 창 의 구성과 기능

• 프로젝트 영역과 작업 영역





테마 변경하기

- 어두운 테마로 변경하고 싶은 경우

The screenshot shows the Android Studio interface with the 'Settings' dialog open. The 'Appearance & Behavior' section is selected, and the 'Appearance' sub-section is active. Under 'UI Options', the 'Theme' dropdown is set to 'IntelliJ', and the 'Dark' theme is being selected. The background shows the IDE with a dark theme applied.

```
package org.techtown.test;

import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```



코드 편집기

• 코드에서 발생한 에러 표시

```
1 package org.techtown.test;
2
3 import ...
4
5
6 public class MainActivity extends AppCompatActivity {
7
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12     }
13 }
14
```

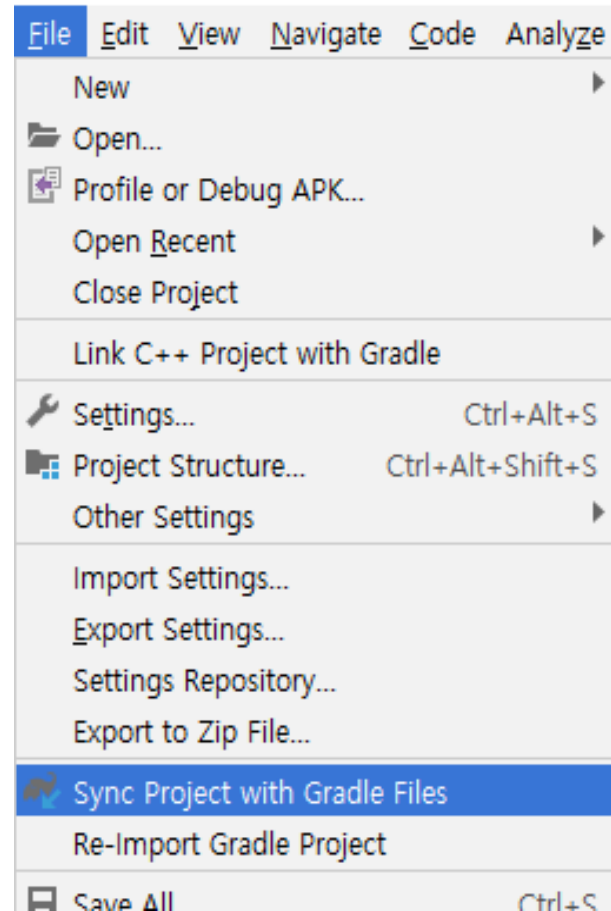
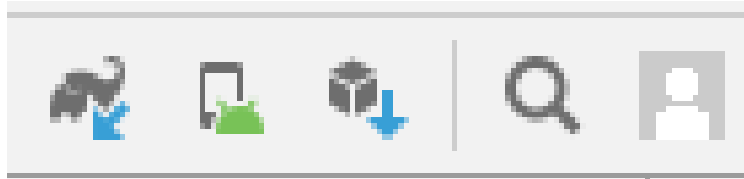
'class' or 'interface' expected

마우스 커서를 올려보세요!



파일 변경사항 반영

- Gradle을 빌드 도구로 사용하며 [Tools>Android>Sync ...] 메뉴를 누르면 파일 변경사항 반영하고 재빌드





코드 편집기

- 클래스 또는 메소드의 문서 참조 → Ctrl+Q

```
MainActivity.java x
1 package org.techtown.test;
2
3 import android.support.v7.app.AppCompatActivity;
4 import android.os.Bundle;
5
6 public class MainActivity extends AppCompatActivity {
7
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12     }
13 }
14
```

android.support.v7.app
public class AppCompatActivity
extends android.support.v4.app.FragmentActivity
implements android.support.v7.app.AppCompatActivity.Callback, android.support.v4.app.TaskS
Gradle: com.android.support:appcompat-v7:28.0.0@aar



코드 편집기

- 후보 클래스와 파라미터의 자동 표시 → Ctrl+P

The screenshot shows an IDE with two windows of MainActivity.java. The left window shows the full class code with a code completion popup for the String constructor. The right window shows the same code with a parameter list popup for the String constructor.

```
1 package org.techtown.test;
2
3 import android.support.v7.app.AppCompatActivity;
4 import android.os.Bundle;
5
6 public class MainActivity extends AppCompatActivity {
7
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12
13         String hello = new String();
14     }
15 }
16
```

String (java.lang)

- String original
- char[] value
- char[] value, int offset, int count
- int[] codePoints, int offset, int count
- byte[] ascii, int hibyte, int offset, int count
- byte[] ascii, int hibyte
- byte[] bytes, int offset, int length, String charsetName
- byte[] bytes, int offset, int length, Charset charset
- byte[] bytes, String charsetName
- byte[] bytes, Charset charset
- byte[] bytes, int offset, int length
- byte[] bytes
- StringBuffer buffer
- StringBuilder builder



코드 편집기

- 부모 클래스의 메소드 재정의 → Generate... > Override Methods...

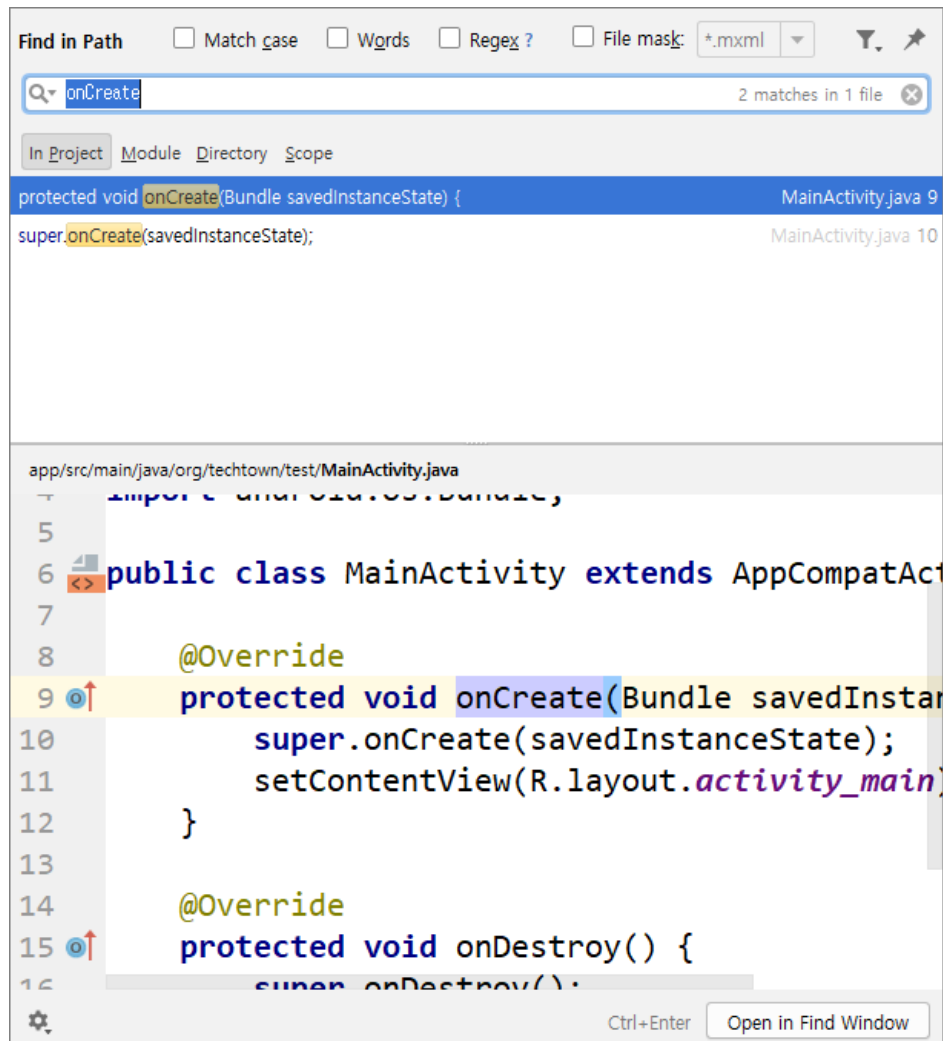
```
MainActivity.java x
1 package org.techtown.test;
2
3 import android.support.v7.app.AppCompatActivity;
4 import android.os.Bundle;
5
6 public class MainActivity extends AppCompatActivity {
7
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12     }
13
14
15
16 }
17
```

Generate	
Constructor	
toString()	
Override Methods...	Ctrl+O
Implement Methods...	Ctrl+I
Delegate Methods...	
Copyright	



코드 편집기

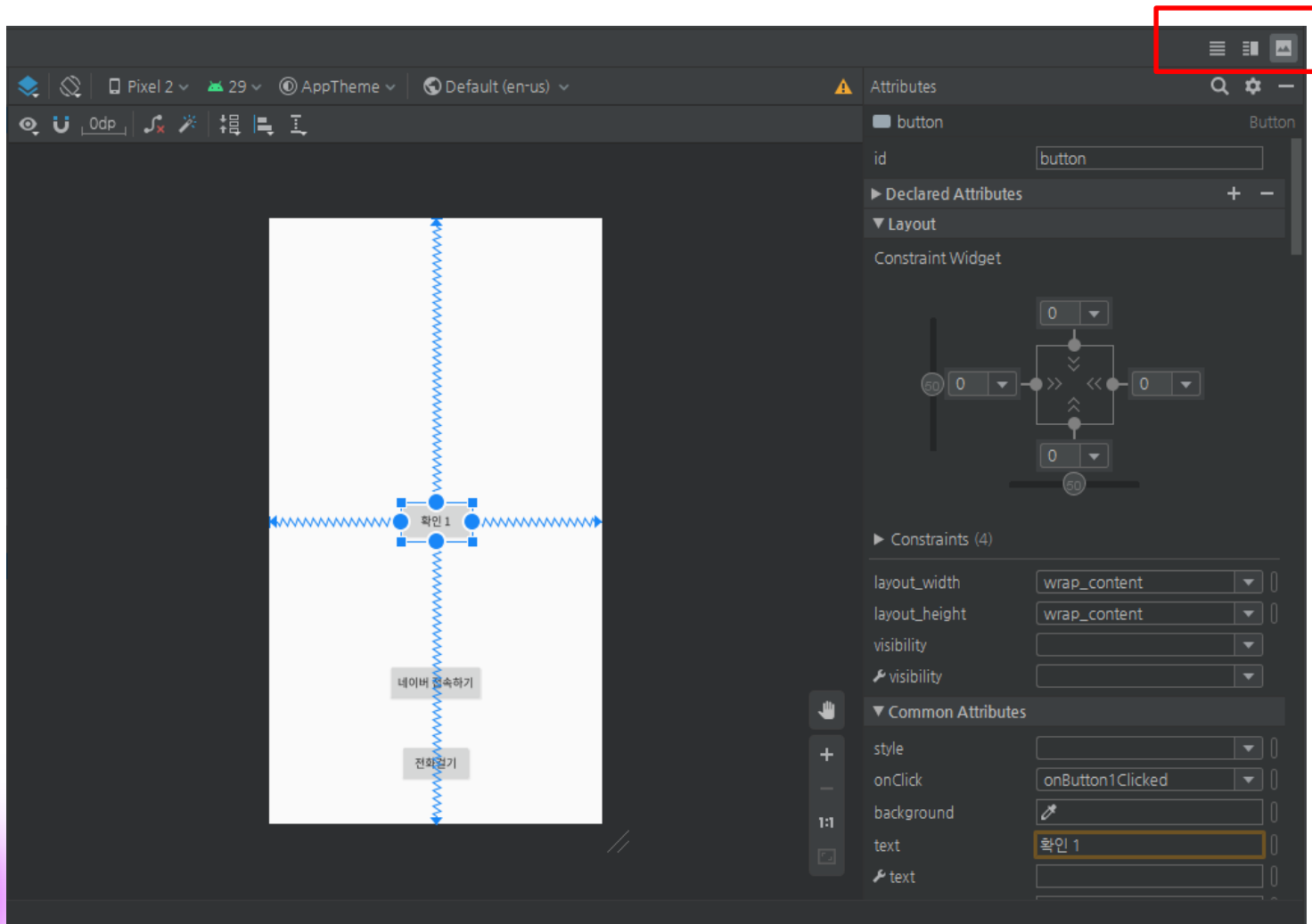
- 파일의 내용 검색 – Find in Path





디자이너 도구

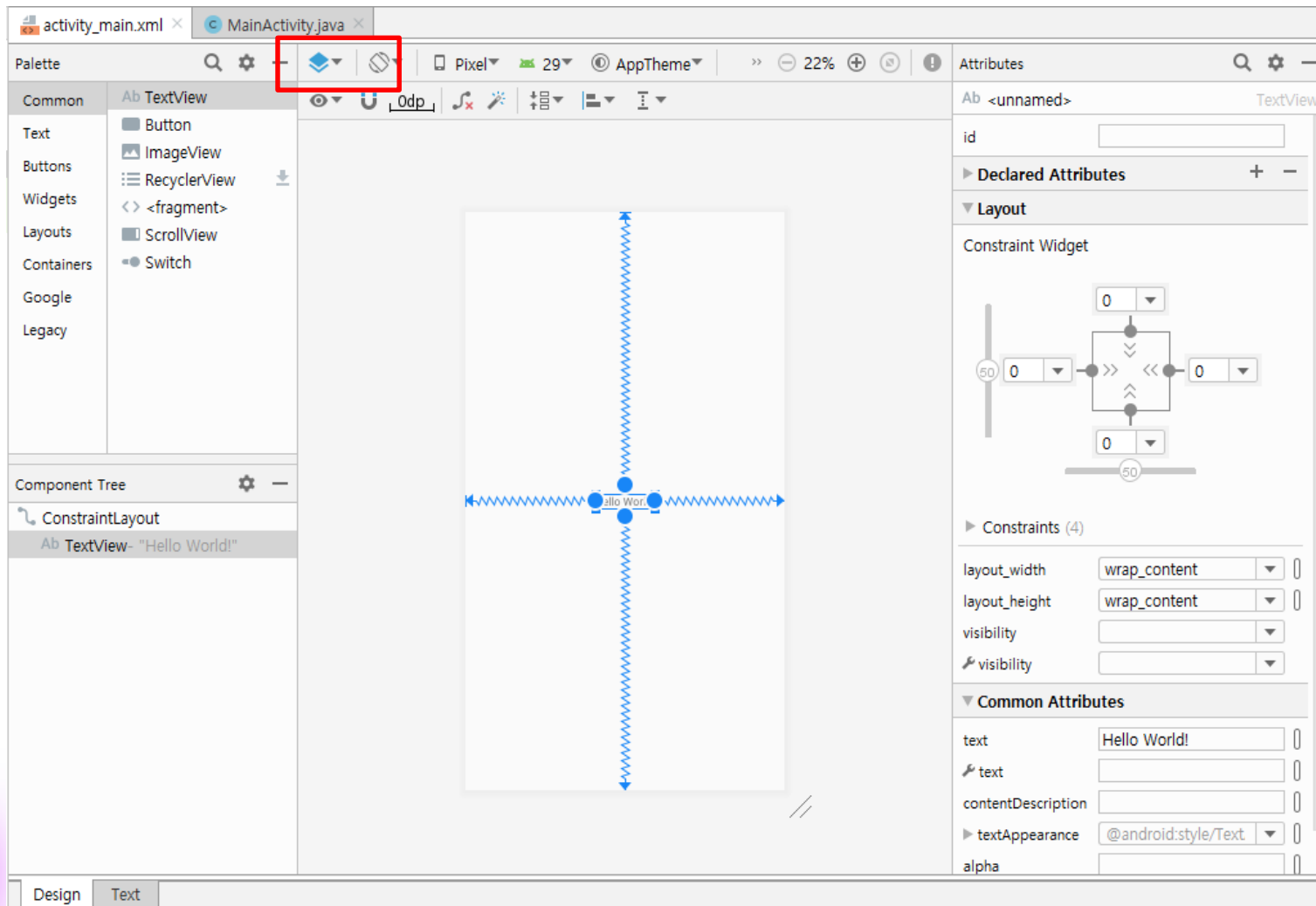
- Design 탭과 Text 탭





디자이너 도구

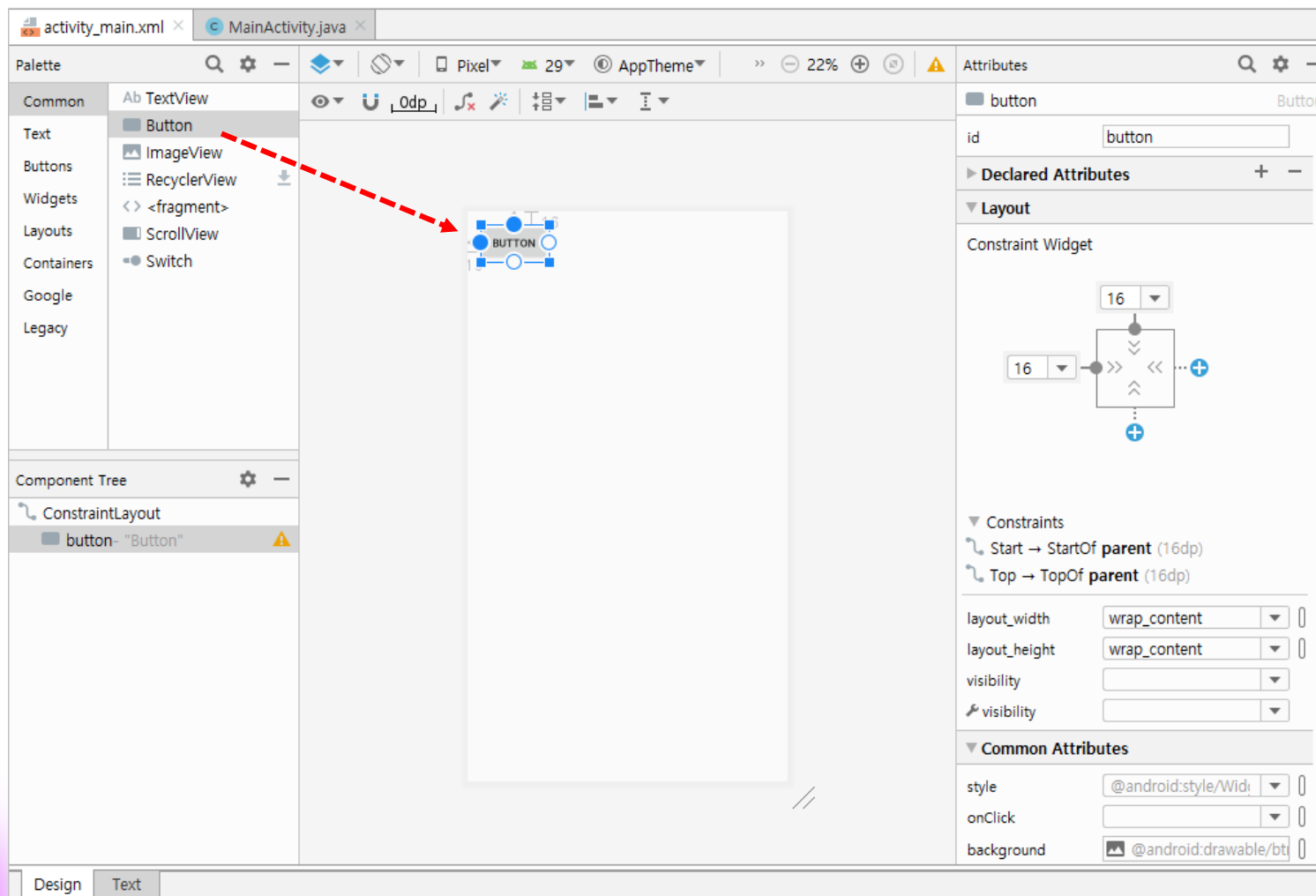
- 디자인 화면에서 [Show Design] 아이콘 선택





디자이너 도구

- 팔레트에서 위젯 끌어다 놓기와 위젯의 속성 설정하기

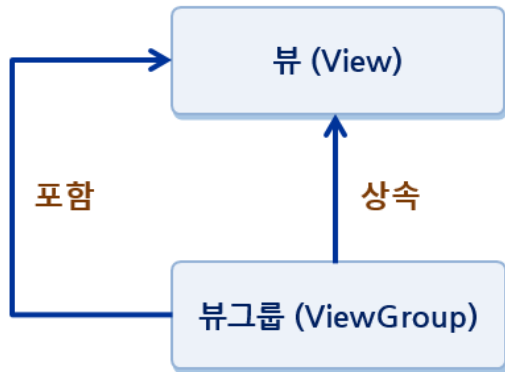


2.

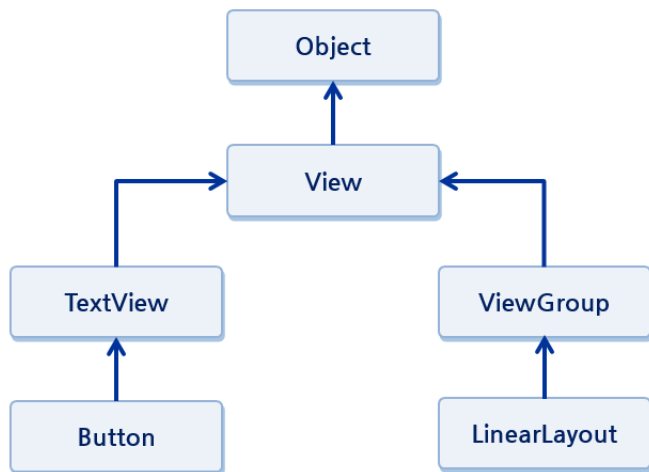
뷰와 뷰의 크기속성 이해하기



뷰와 뷰그룹의 정의



[뷰와 뷰 그룹의 관계]



[버튼과 리니어 레이아웃의 계층도]

• 뷰(View)

보여지는 놈들중 최상위

- 화면에 보이는 각각의 것들 (버튼, 텍스트 등등)
- 흔히 컨트롤(Control)이나 위젯(Widget)이라 불리는 UI 구성 요소

• 뷰 그룹(View Group)

- 뷰들을 여러 개 포함하고 있는 것
- 뷰 그룹도 뷰에서 상속하여 뷰가 됨. 즉, 위의 뷰는 버튼, 텍스트 뿐만 아니라 이것들을 포함하는 눈에 보이지 않는 영역을 포함함

• 위젯(Widget)

- 뷰 중에서 일반적인 컨트롤의 역할을 하고 있는 것
- 버튼, 텍스트 등등

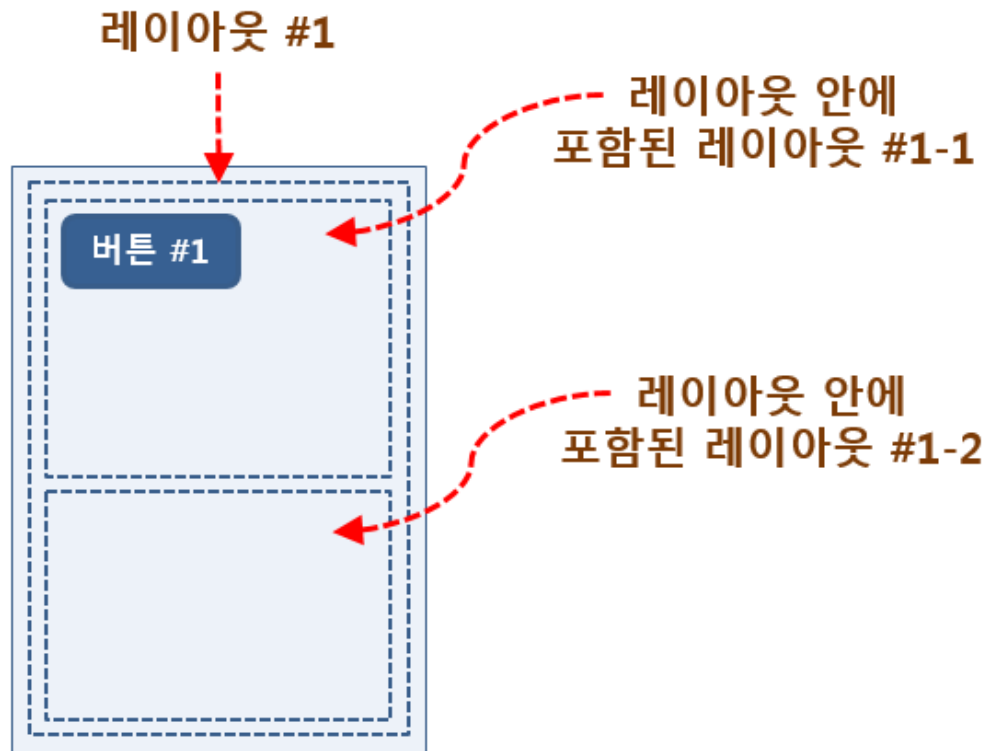
• 레이아웃(Layout)

- 뷰 그룹 중에서 내부에 뷰들을 포함하고 있으면서 그것들을 배치하는 역할을 하는 것



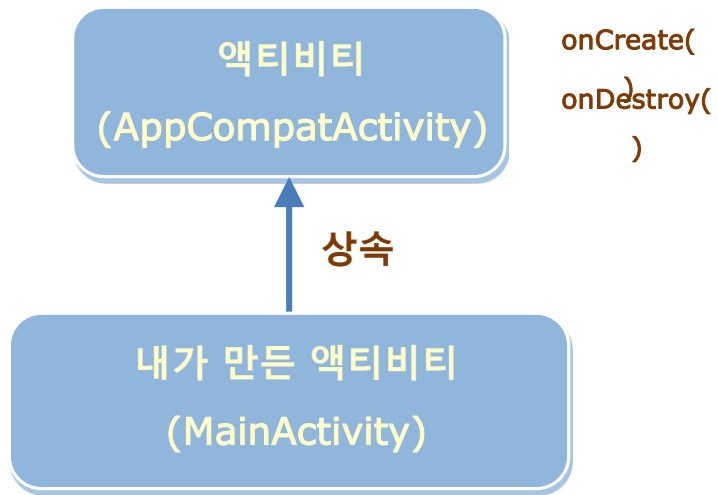
레이아웃 안에 레이아웃 포함시키기

- 레이아웃도 뷰이기 때문에 레이아웃 안에 레이아웃을 포함시킬 수 있음





상속에 대해 잘 몰라요!



• 상속

- 객체지향의 가장 기본적인 개념 중 하나
- 부모의 특성을 그대로 물려받는 것으로 변수나 메소드 재사용 가능

• 액티비티의 상속

- 처음 만들어 본 액티비티에서 extends 키워드 사용

```
public class MainActivity extends AppCompatActivity
```

• 부모 클래스의 메소드를 재정의

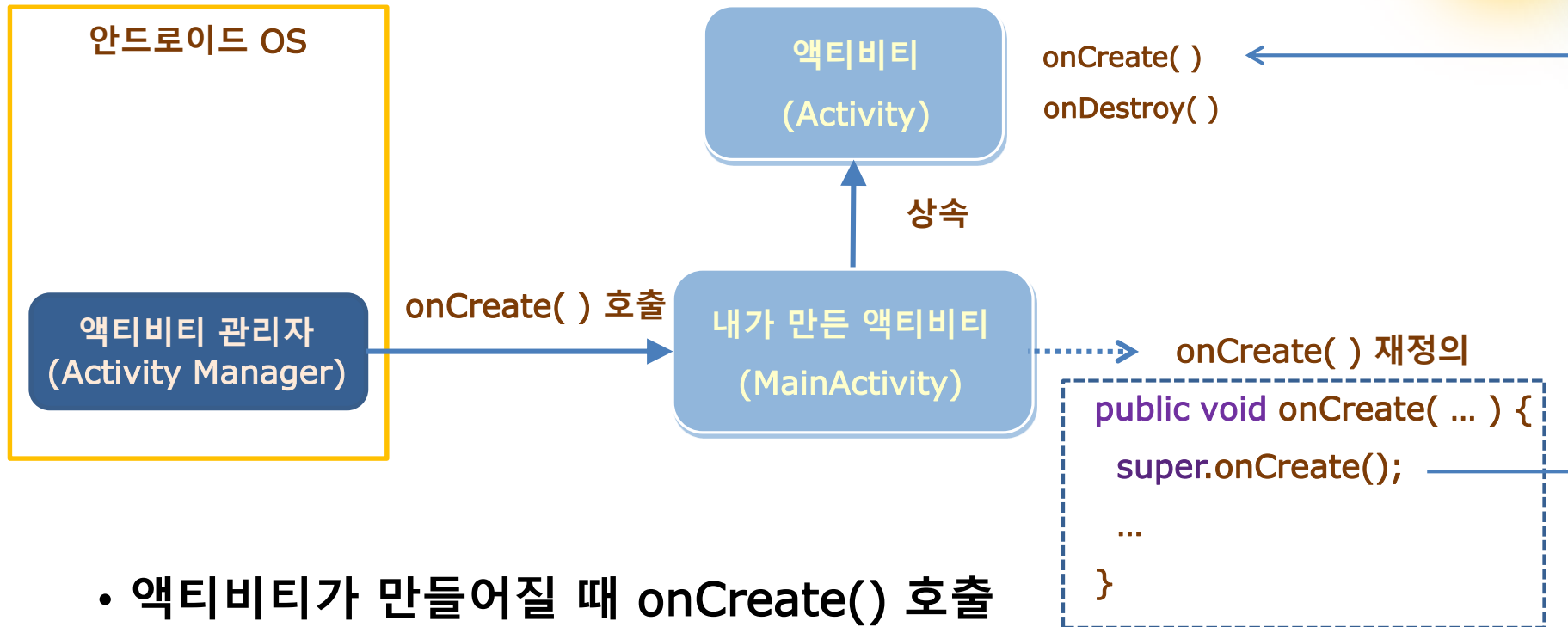
- onCreate() 메소드는 이미 부모 클래스에 정의되어 있음
- 기능을 추가하고 싶을 때 재정의(Override)

• this와 super

- 나 자신은 this, 부모는 super 를 사용하여 변수나 메소드 참조
super.onCreate(...);



재정의한 메소드를 시스템에서 호출하는 방법



• 액티비티가 만들어질 때 onCreate() 호출

- 내가 만든 액티비티의 onCreate() 메소드를 호출
- super.onCreate() 를 호출하여 Activity 클래스에 정의된 onCreate() 메소드의 기능을 사용
- 그 아래에 추가적으로 필요한 기능을 코드로 추가



XML 레이아웃의 구성

- 뷰 태그와 속성으로 구성됨

```
<시작 태그  
  속성1="속성값1"  
  속성2="속성값2"  
  ...  
</끝 태그>
```

```
<TextView  
  android:layout_width="wrap_content"  
  android:layout_height="wrap_content"  
  android:text="Hello World!" />
```

```
<android.support.constraint.ConstraintLayout  
  android:layout_width="match_parent"  
  android:layout_height="match_parent"  
  ...  
</android.support.constraint.ConstraintLayout>
```

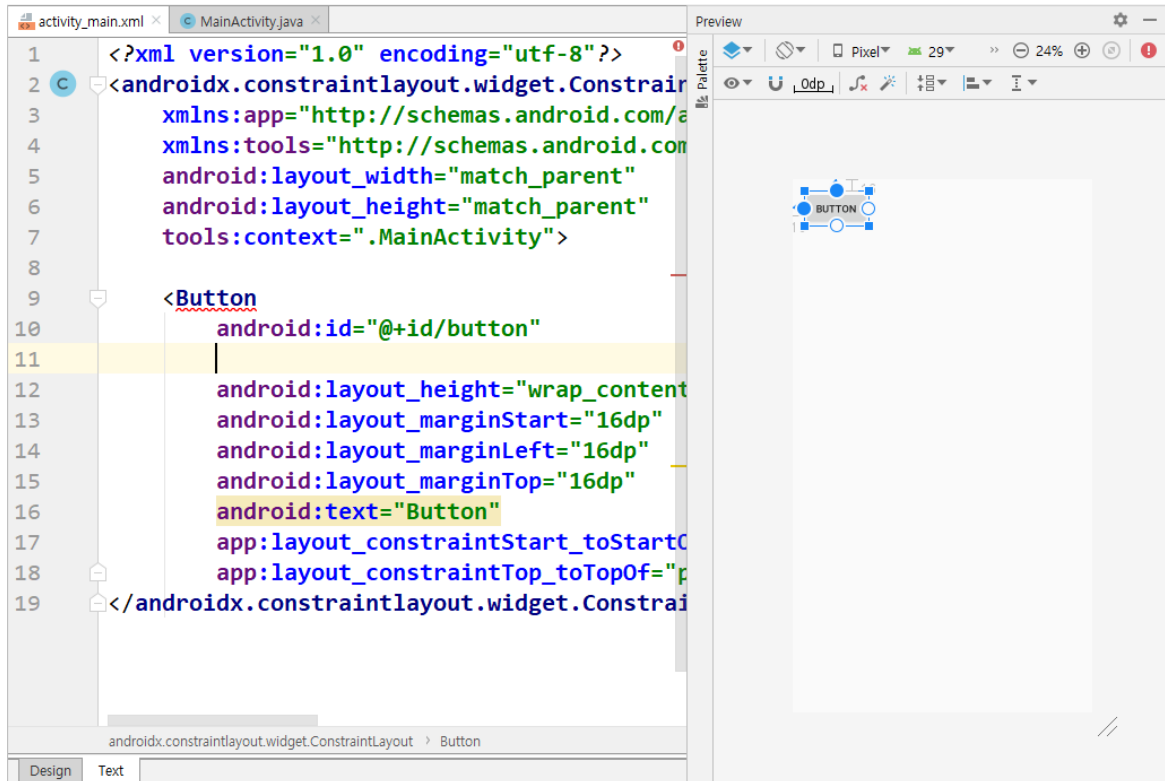
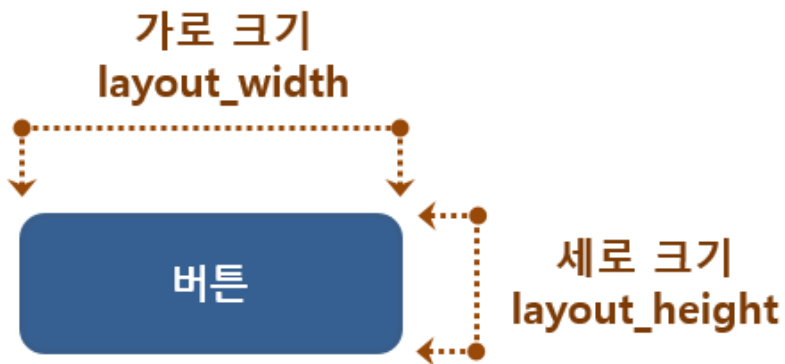
속성Prefix : 속성명1="속성값1"

```
android:layout_width="wrap_content"  
app:layout_constraintLeft_toLeftOf="parent"
```




가로 크기와 세로 크기 속성

- 가로 크기와 세로 크기는 뷰의 필수 속성
- 속성 제거 시 에러 표시





크기 지정 방법

• [필수] layout_width, layout_height

- 가장 기본적이면서 필수 속성으로 뷰의 폭과 높이를 설정함

(1) match_parent

뷰그룹에 남아 있는 여유 공간을 채움

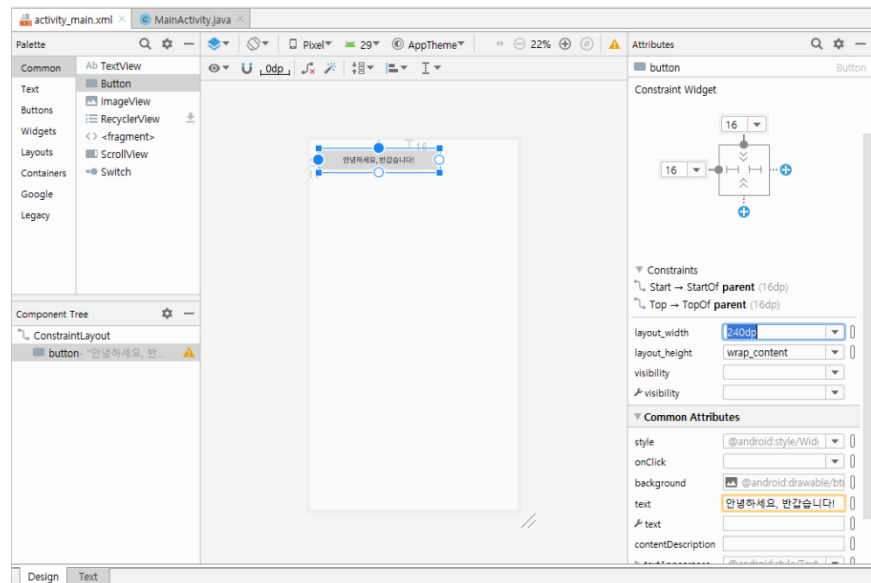
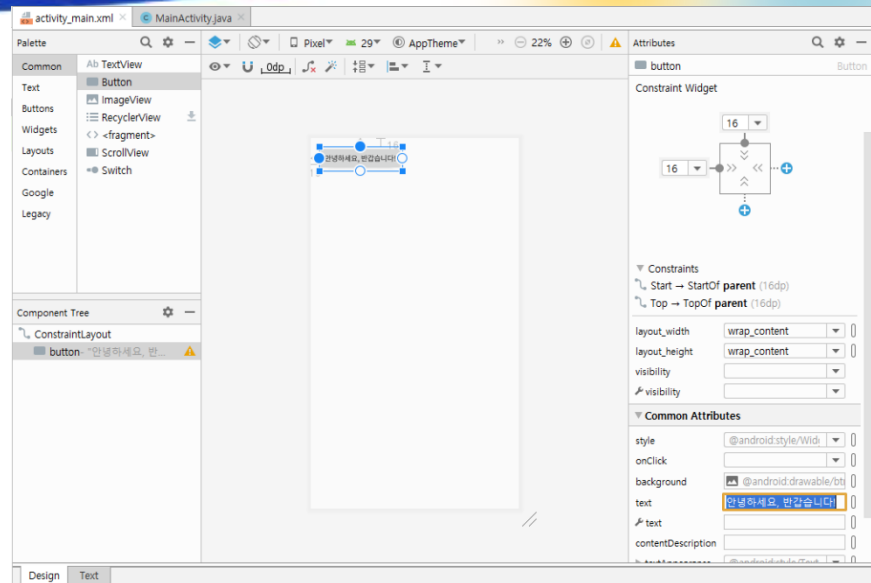
(2) wrap_content

뷰에 들어 있는 내용물의 크기에 따라 뷰의 크기가 결정됨

(3) 크기 값 지정

크기를 고정된 값으로 직접 지정하고 싶을 때 사용함

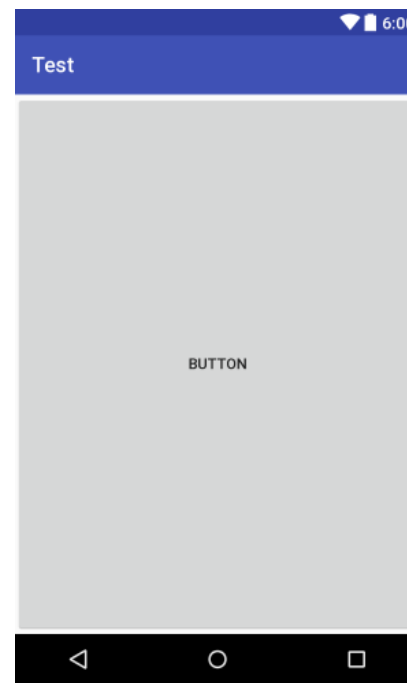
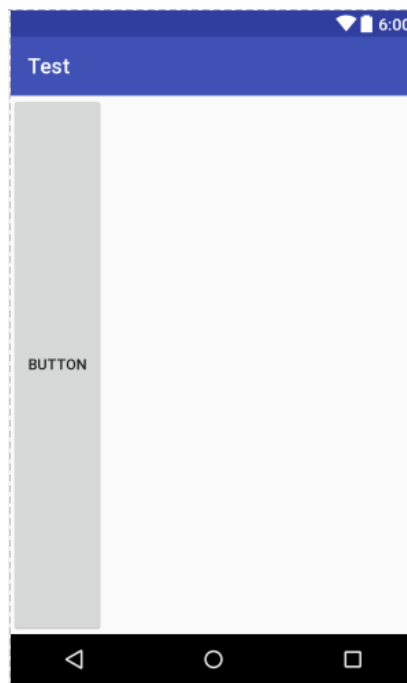
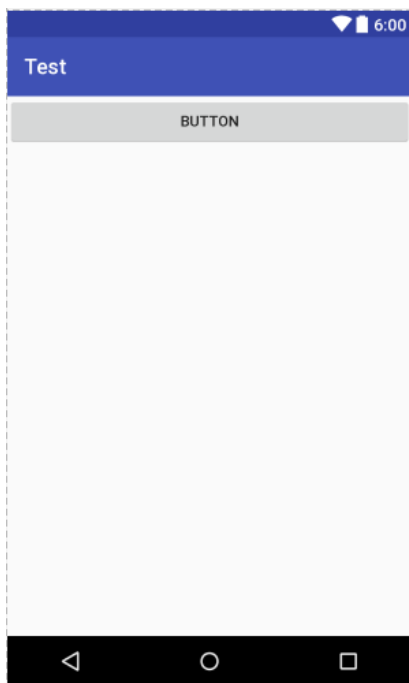
ex) "100px", "200dp"





최상위 레이아웃을 다른 것으로 변경한 경우

- match_parent와 wrap_content 적용 상태를 확실하게 구분할 수 있음



3.

레이아웃 기초 익히기

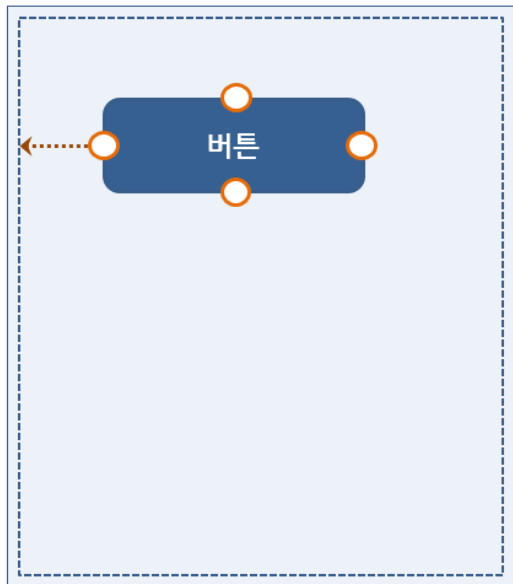


연결선으로 만드는 제약조건

- 뷰의 위, 아래, 왼쪽, 오른쪽의 연결점을 부모 레이아웃의 벽면과 연결하면 제약조건 생성
- 같은 레이아웃 안에 들어있는 다른 뷰와 연결 가능

제약조건(Constraint)

버튼의 왼쪽을 부모 레이아웃과 연결해 주세요.



• 연결점을 연결할 수 있는 타깃

- 같은 레이아웃 안에 들어 있는 다른 뷰의 연결점
- 부모 레이아웃의 연결점
- 가이드라인(Guideline)



새로운 프로젝트 생성

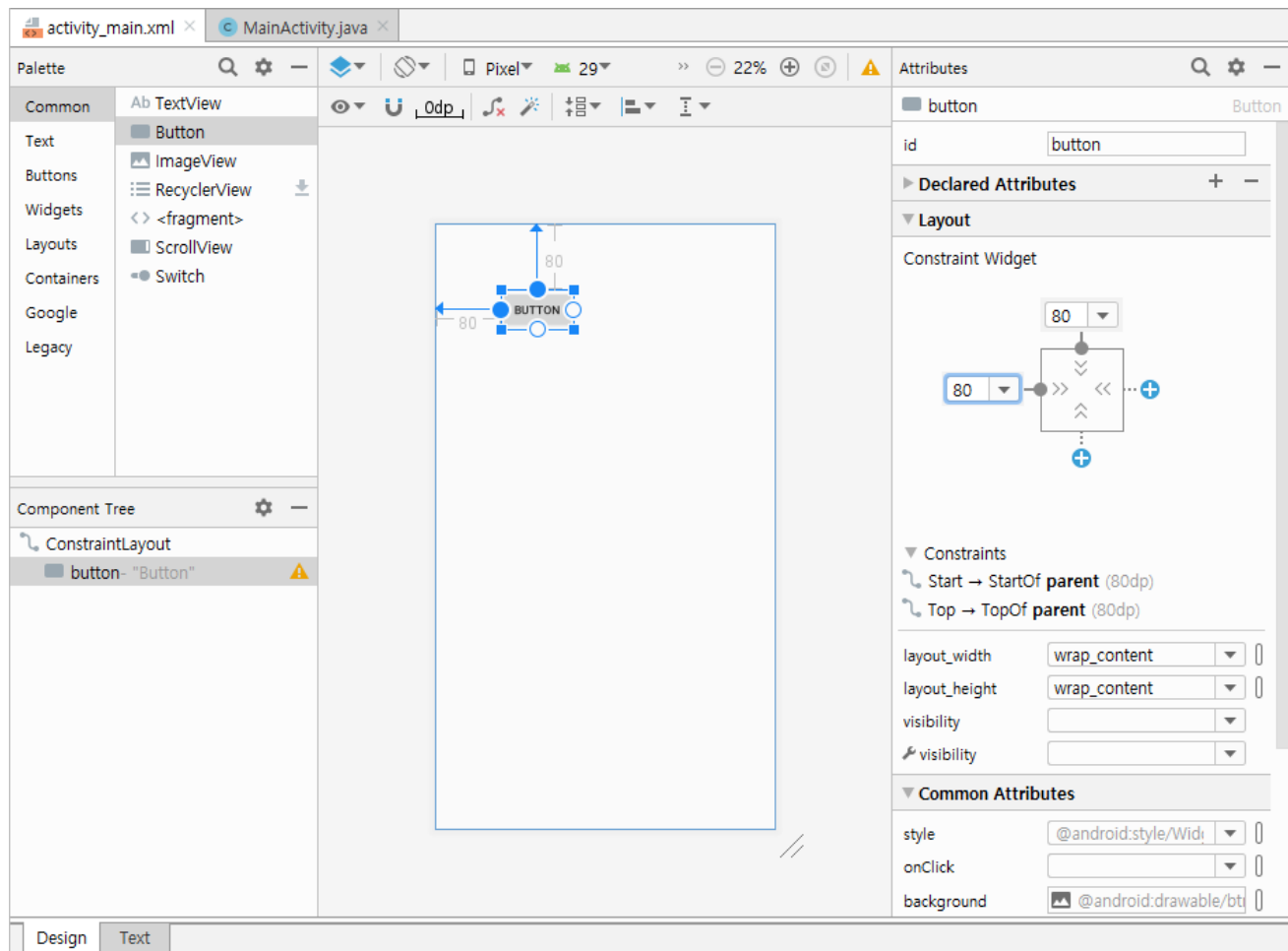
- SampleConstraintLayout이라는 이름으로 새로운 프로젝트 생성
- 화면 왼쪽 윗부분에 버튼 추가

The screenshot displays the Android Studio interface during the creation of a new project. On the left, the 'Create New Project' dialog is open, showing the 'Configure your project' screen. The project name is 'SampleConstraintLayout', the package name is 'org.techtown.sampleconstraintlayout', the save location is 'C:\Users\Mars\AndroidStudioProjects\SampleConstraintLayout', the language is 'Java', and the minimum API level is 'API 15: Android 4.0.3 (IceCreamSandwich)'. The 'Design' tab is selected, showing a visual representation of the layout. A button is added to the layout, and its properties are visible in the 'Attributes' panel on the right. The button's ID is 'button', and its text is 'Button'. The 'Constraints' panel shows the button is centered horizontally and vertically within the parent container.



연결선 생성

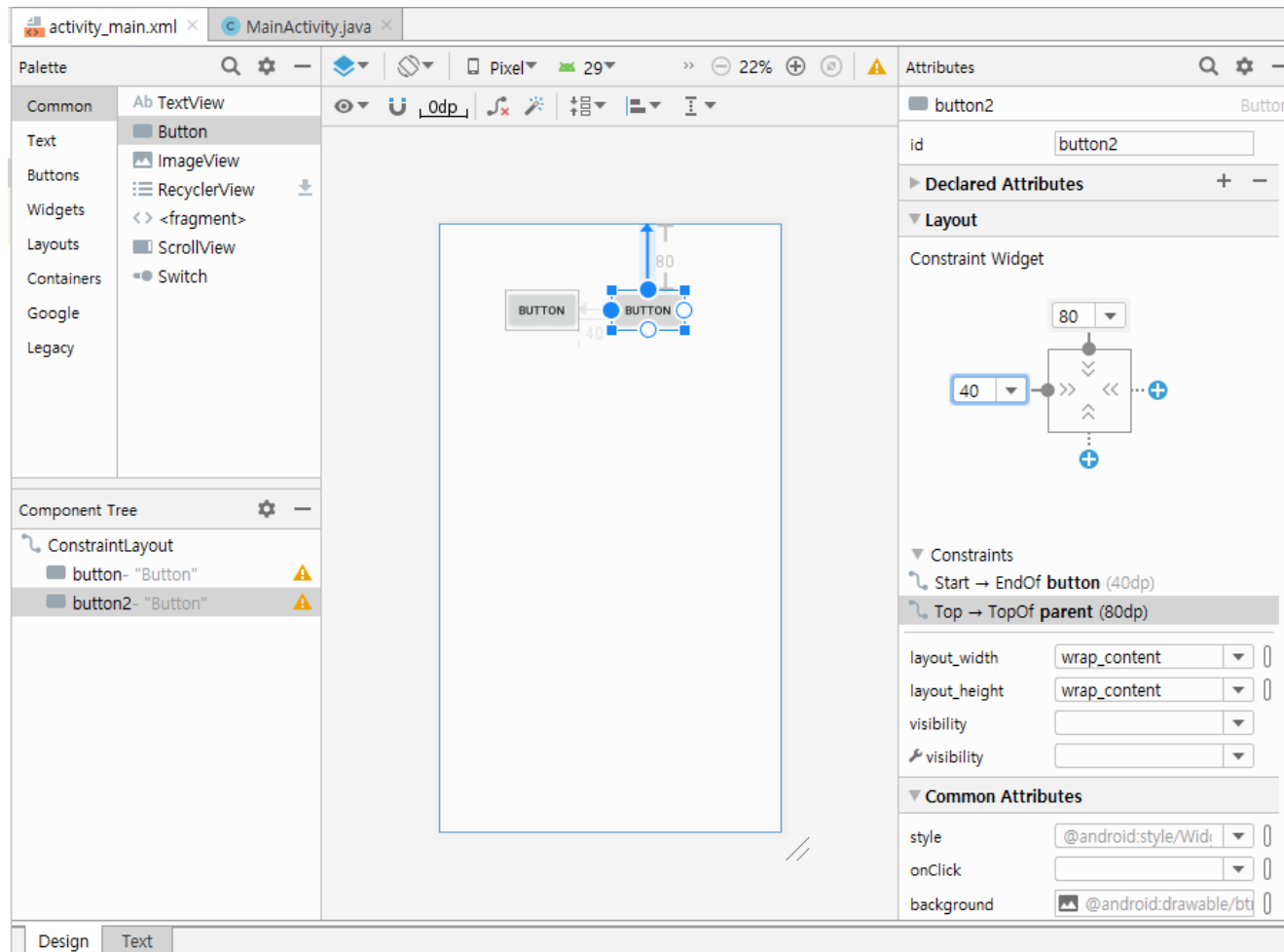
- 왼쪽과 위쪽에 있는 연결점을 부모 레이아웃의 벽면과 연결





버튼 하나 더 추가하고 연결선 생성

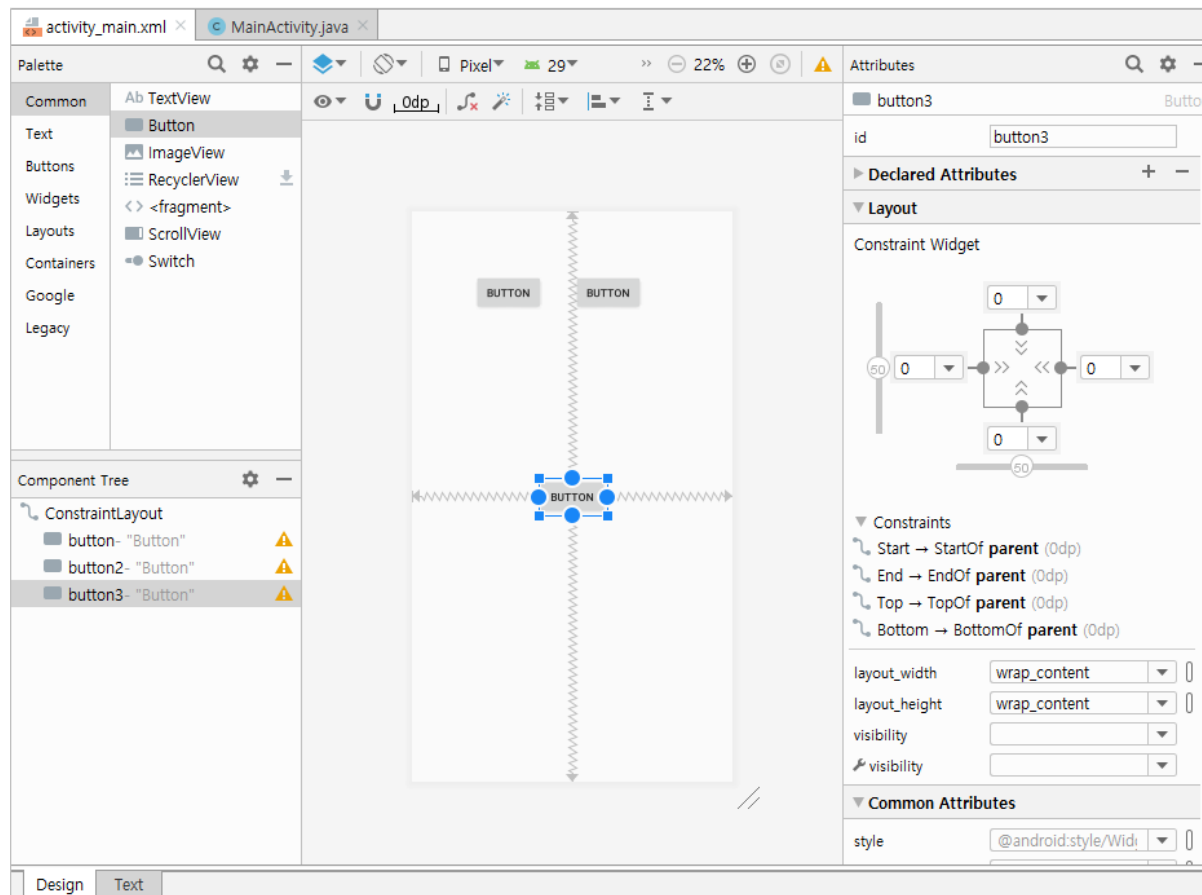
- 버튼 추가 후 부모 레이아웃 및 기존 버튼과 연결선 생성





화면 가운데에 뷰 배치하기

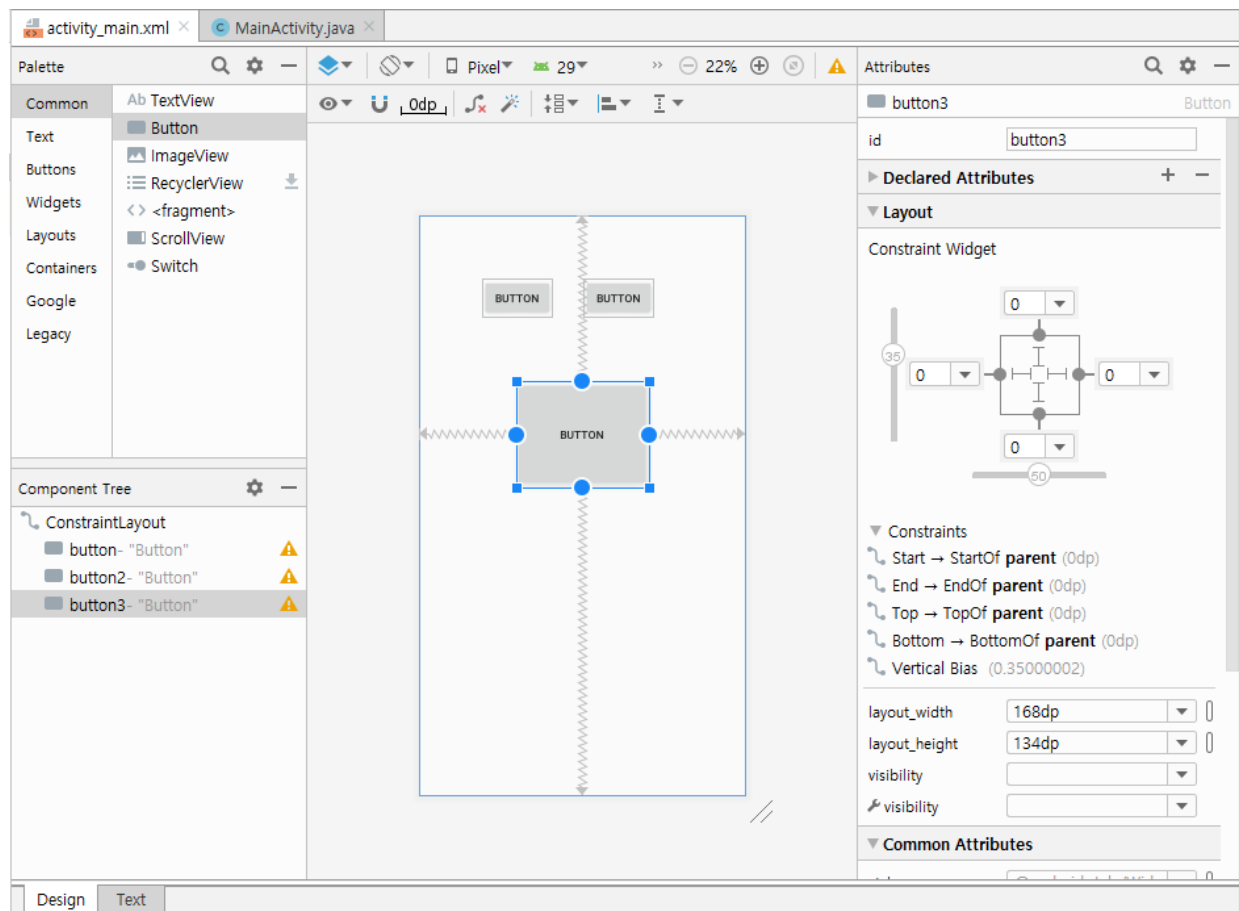
- 버튼 추가 후 위, 아래, 왼쪽, 오른쪽 모두 부모 레이아웃과 연결선 생성
- 좌우 또는 위,아래를 쌍으로 연결하면 그 가운데에 배치됨





바이어스 사용하기

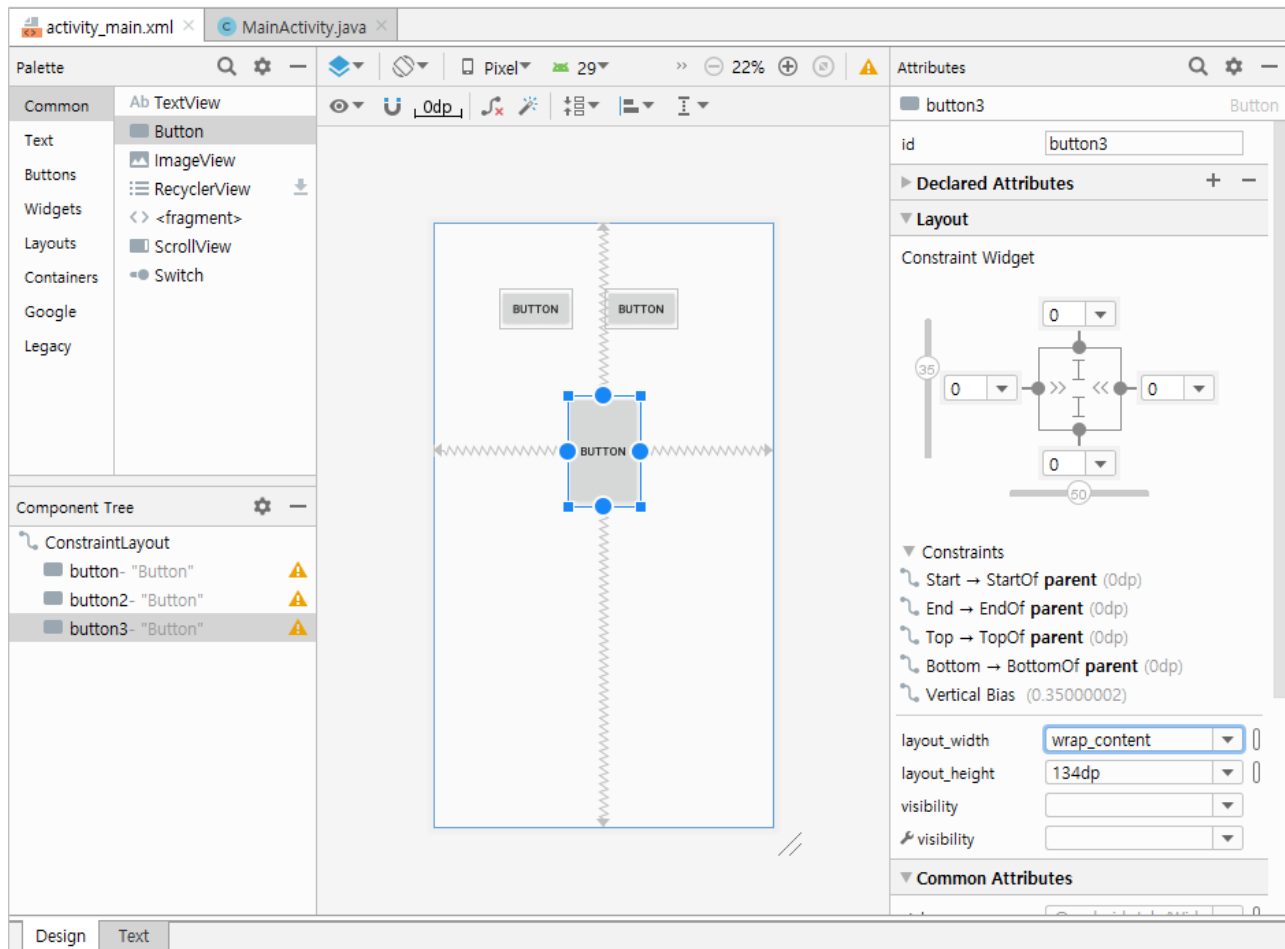
- 양쪽 또는 상하로 연결된 경우 오른쪽 제약조건 미리보기 창에서 바이어스(Bias)를 이용해 위치 조정 가능





뷰의 크기 조정

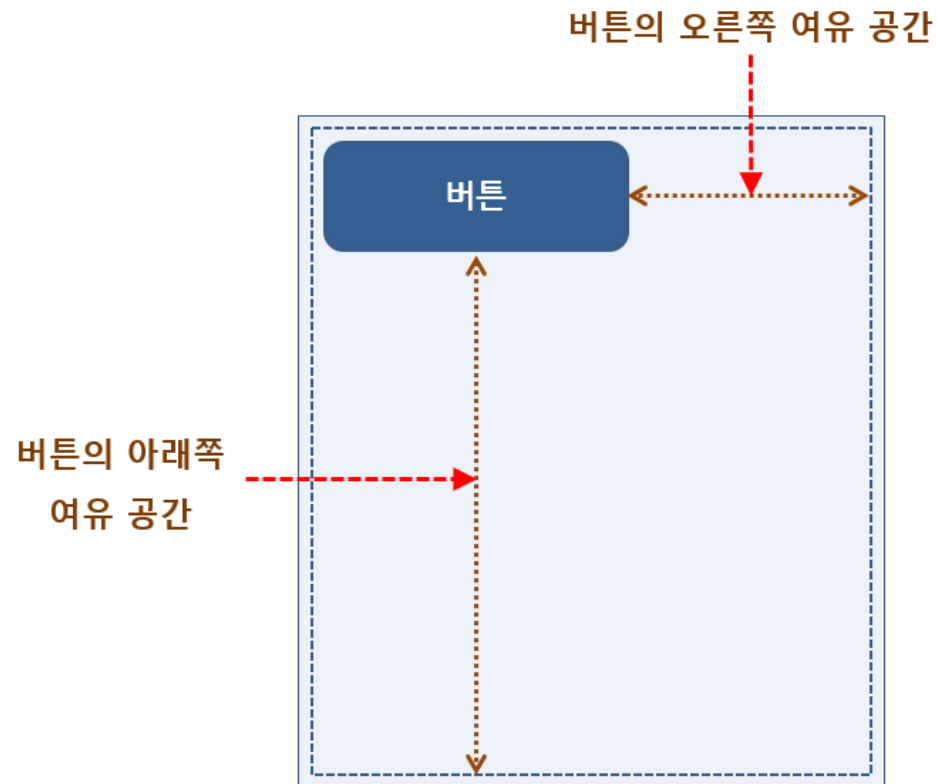
- 모서리에 있는 핸들(Handle)을 이용해 뷰의 크기 조정 가능





뷰가 차지할 수 있는 여유공간

- 부모 레이아웃 안에서 뷰가 차지할 수 있는 여유공간은 레이아웃에 의해 결정됨

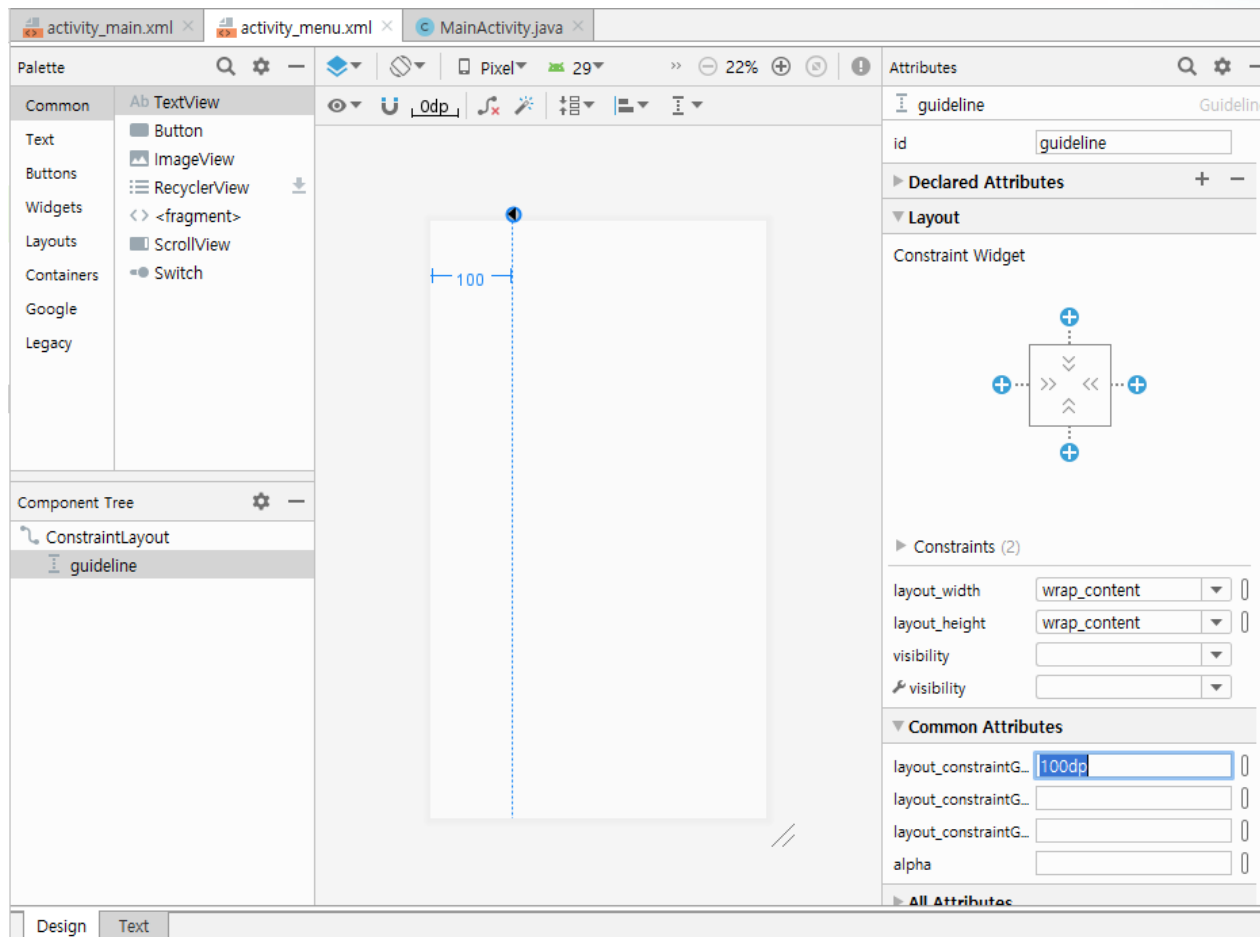
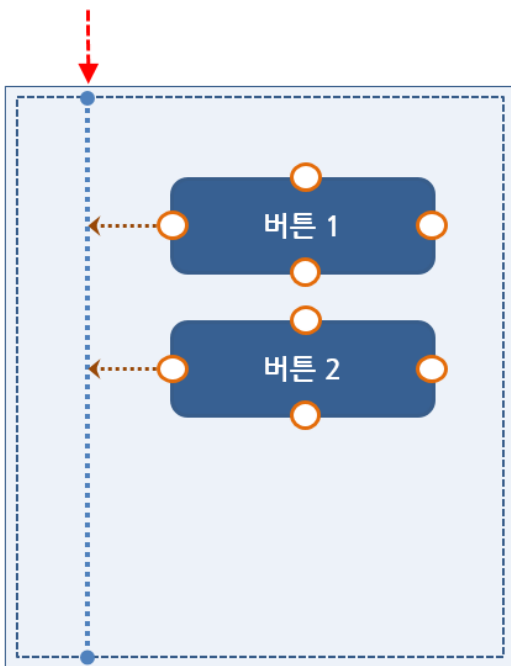




가이드라인 사용하기

- 일정 기준선으로 뷰를 정렬할 때나 기준선에 맞추어 추가할 때 사용됨

세로방향 가이드라인





XML 코드로 만들어지는 가이드라인

- XML 레이아웃 파일에서 원본 XML을 보면 Guideline 태그로 만들어짐

```
<androidx.constraintlayout.widget.Guideline
    android:id="@+id/guideline"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    app:layout_constraintGuide_begin="100dp" />
```

...



가이드라인의 원본 코드 살펴보기

- xmlns 로 시작하는 속성, android로 시작하는 속성 등이 있음

```
<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">
```

...



XML 코드의 접두어 의미

- XML 레이아웃 코드에서 접두어가 가지는 의미가 있음

(1) `xmlns:android` 안드로이드 기본 SDK에 포함되어 있는 속성을 사용합니다.

(2) `xmlns:app` 프로젝트에서 사용하는 외부 라이브러리에 포함되어 있는 속성을 사용합니다.

(3) `xmlns:tools` 안드로이드 스튜디오의 디자이너 도구 등에서 화면에 보여줄 때 사용합니다.

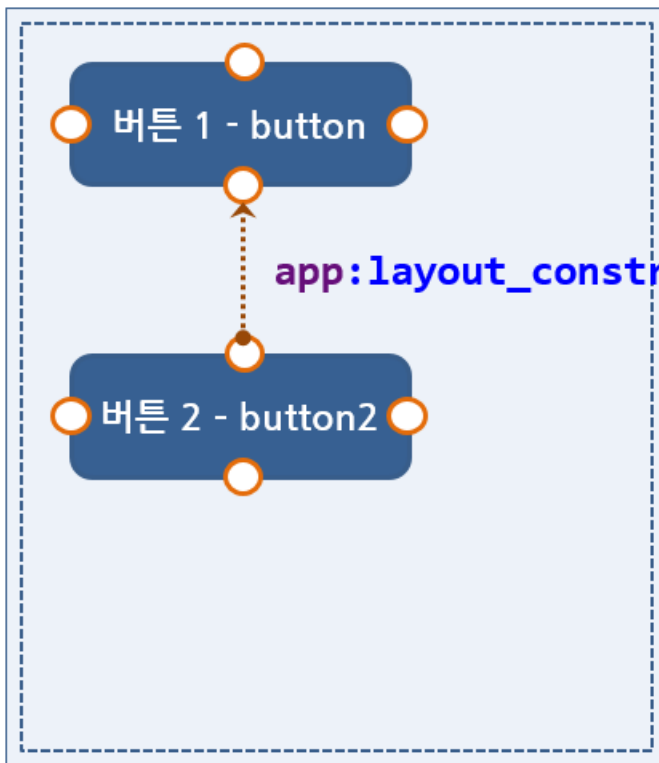
이 속성은 앱이 실행될 때는 적용되지 않고 안드로이드 스튜디오에서만 적용됩니다.



연결선이 만들어내는 XML 속성의 형식

- 일정한 규칙으로 만들어짐

`layout_constraint[소스 뷰의 연결점]_[타겟 뷰의 연결점]="[타겟 뷰의 id]"`



`app:layout_constraintTop_toBottomOf="@+id/button"`



뷰의 크기 지정에 사용되는 단위

단위	단위 표현	설 명
px	픽셀	화면 픽셀 ,해상도 따라 크기 달라짐 잘안
dp 또는 dip	밀도 독립적 픽셀 (density independent pixel)	160dpi 화면을 기준으로 한 픽셀 예) 1인치 당 160개의 점이 있는 디스플레이 화면에서 1dp는 1px와 같음. 1인치 당 320개의 점이 있는 디스플레이 화면에서 1dp는 2px와 같음.
sp 또는 sip	축척 독립적 픽셀 (scale independent pixel)	가변 글꼴을 기준으로 한 픽셀로 dp와 유사하나 글꼴의 설정에 따라 달라짐
in	인치	1인치로 된 물리적 길이
mm	밀리미터	1밀리미터로 된 물리적 길이
em	텍스트 크기	글꼴과 상관없이 동일한 텍스트 크기 표시