بِسْمِ اللهِ الرَّحْمٰنِ الرَّحِيْمِ

قَالَ رَبِّ ٱشْرَحْ لِى صَدْرِى وَيَسِّرْ لِىٓ أَمْرِى
وَٱحْلُلْ عُقْدَةً مِّن لِّسَانِى يَفْقَهُوا۟ قَوْلِى

وَقُل رَّبِّ زِدْنِى عِلْمًا

قَالُوا۟ سُبْحَٰنَكَ
لَا عِلْمَ لَنَآ إِلَّا مَا عَلَّمْتَنَآ
إِنَّكَ أَنتَ ٱلْعَلِيمُ ٱلْحَكِيمُ

Android Layouts

Mobile Computing

LECTURE 05-06

Layouts

Instructor:

Maulana Haq Nawaz

تصحیح نیت

حضرت محمد صلی اللہ علیہ وسلم نے فرمایا

اِنَّمَا الْاَعْمَالُ بِالنِّـیَّاتِ

ترجمہ

اعمال کا دارومدار نیتوں پر

دُعا

((اللَّهُـــمَّ

إنِّي أَسْأَلُكَ الهُـدَى،

و التُّقَى، و العَفَافَ،

و الغِنَـــى))

عن عبدالله بن مسعود رضي الله عنه عن النبي ﷺ أنه كان يقول :

((اللهم إني أسألك الهدى ...)).

رواه مسلم (2721)

مجالس الذكر

Little Efforts Daily Will Make You the Greatest

To systematically learn and get excellence in any concept / subject

- روز کا کام روز کریں
- اک مہینے کا کھانا ایک دن میں نہیں کھایا جا سکتا،
ایسے ہی ایک مہینے کا کام ایک دن میں نہیں ہو سکتا

Little Efforts Daily Will Make You the Greatest

Importance of completing tasks on daily basis

Main Reason of Failure in Life

- یہ کام کل کریں گے
- جو کام کبھی بھی ہو سکتا ہے وہ کبھی نہیں ہوتا
- زندگی ایک دن ہے اور وہ ہے آج۔ زندگی میں کل نام کی کوئی چیز نہیں ہے
- جو دن آپ کی زندگی سے چلا گیا اب واپس نہیں آئے گا
- آج کا کام آج ہی ہوسکتا ہے
- جو گز گیا وہ آنا نہیں ، آنے والے دن کا پتہ نہیں ، آج میدان جما ہے تو اینے جوہر دکھاؤ

How to Achieve BIG Goals in Life

Balanced Life is Ideal Life ☺

To achieve BIG Goals in Life

- **Make a Schedule of 24 Hours with a focus on Five main components of Human Life**
  - **Health**
    - **Physical Health**
    - **Mental Health**
    - **Social Health**
  - **Spirituality**
  - **Work**
  - **Family**
  - **Friends**

جو کام کریں دل سے کریں

- کام کرنا.
- خوشی خوشی کام کرنا.
- اللہ کو ساتھ لے کرخوشی خوشی کام کرنا.

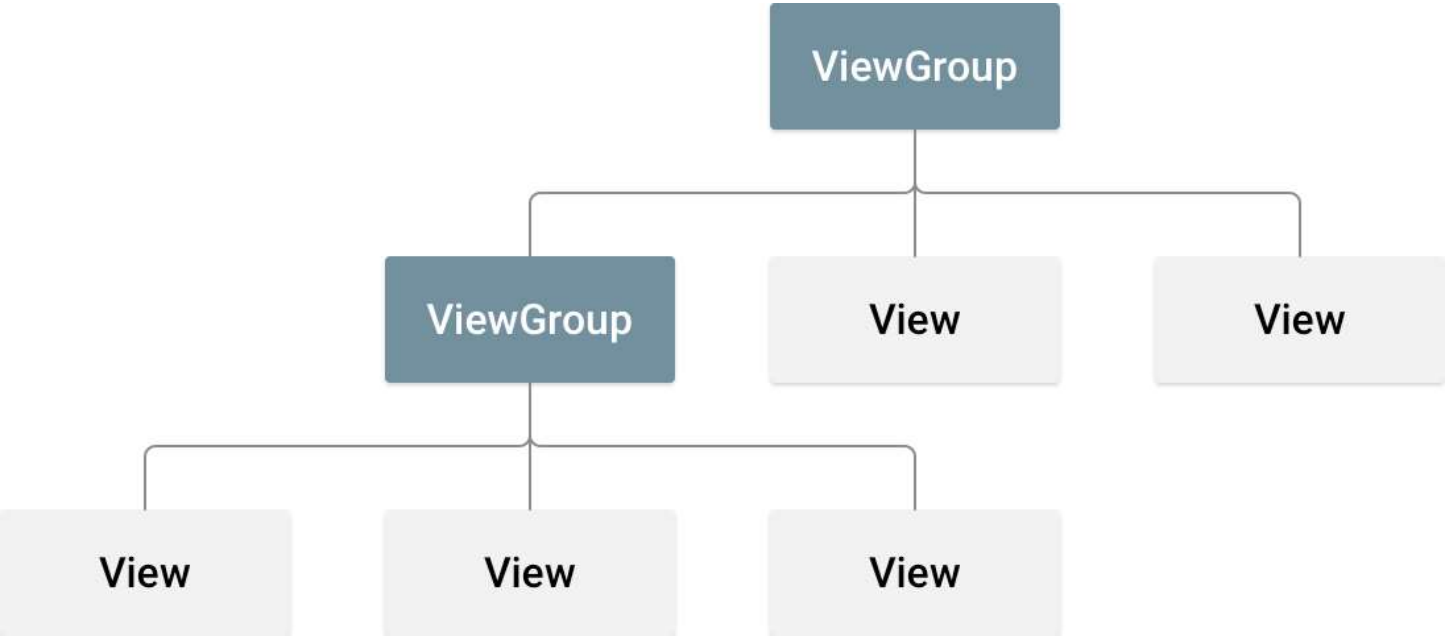
آیت: إِيَّاكَ نَعْبُدُ وإِيَّاكَ نَسْتَعِينُ

ترجمہ: یا اللہ ہم تیری ہی عبادت کرتے ہیں.
اور تجھ ہی سے مد د مانگتے ہیں

Lecture Outline

1. Revision
2. View ViewGroup
3. Layout
4. Layout Parameters
5. Constraint layout
6. Constraint Baseline
7. Constraint Centre
8. Chains
9. Constraint bias
10. Infer Constraints
11. Stop Complaining! Stop Criticizing! Let’s Start Contributing
12. Lecture Summary

View and ViewGroup



- A **View** usually draws something the user can see and interact with.
- A **ViewGroup Invisible** container

View and ViewGroup

The View objects are usually called "widgets" and can be one of many subclasses, such as Button or TextView.

The ViewGroup objects are usually called "layouts" can be one of many types that provide a different layout structure, such as LinearLayout or ConstraintLayout .

View and ViewGroup

Each layout file must contain exactly one root element, which must be a View or ViewGroup object. Once you've defined the root element, you can add additional layout objects or widgets as child elements to gradually build a View hierarchy that defines your layout. For example, here's an XML layout that uses a vertical LinearLayout to hold a TextView and a Button:

Layout Anatomy

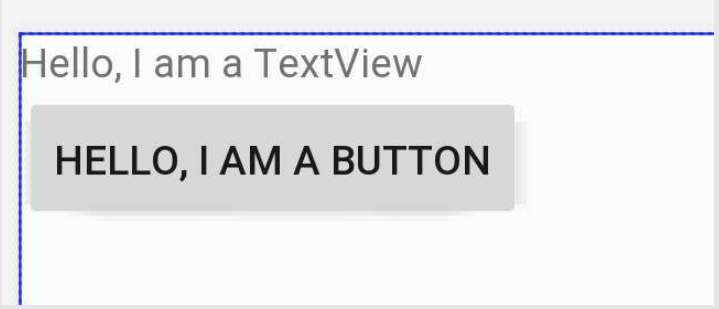
1. Layout regions are the foundation for interactive experiences. They're a layout's building blocks and are composed of elements and components that share a similar function. Layout regions can also group smaller containers such as cards. A large screen layout has three main regions:
    1. App bars
    2. Navigation
    3. Body

https://material.io/design/layout/understanding-layout.html#layout-anatomy

Layout Anatomy

View and ViewGroup

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical" >
   <TextView android:id="@+id/text"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Hello, I am a TextView" />
   <Button android:id="@+id/button"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Hello, I am a Button" />
</LinearLayout>
```



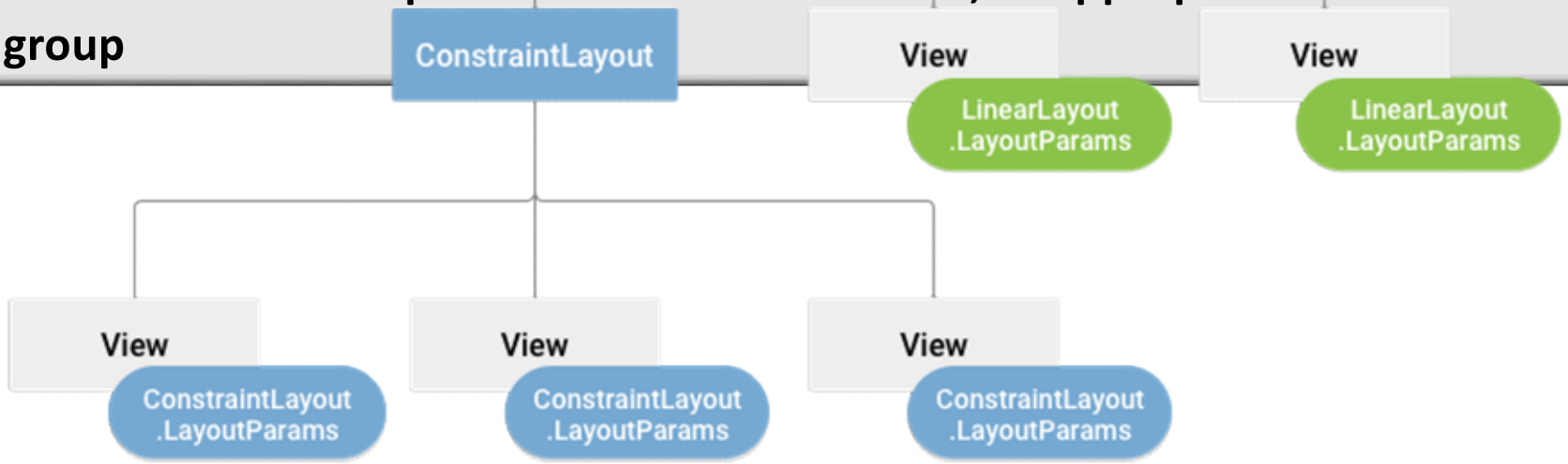
Load XML Resources

```
package org.haqnawaz.layout;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

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

Layout Parameters

XML layout attributes named layout_something define layout parameters for the View that are appropriate for the V LinearLayout which it resides.

Every ViewGroup class implements a nested class that extends ViewGroup.LayoutParams. This subclass contains property types that define the size and position for each child view, as appropriate for the view group



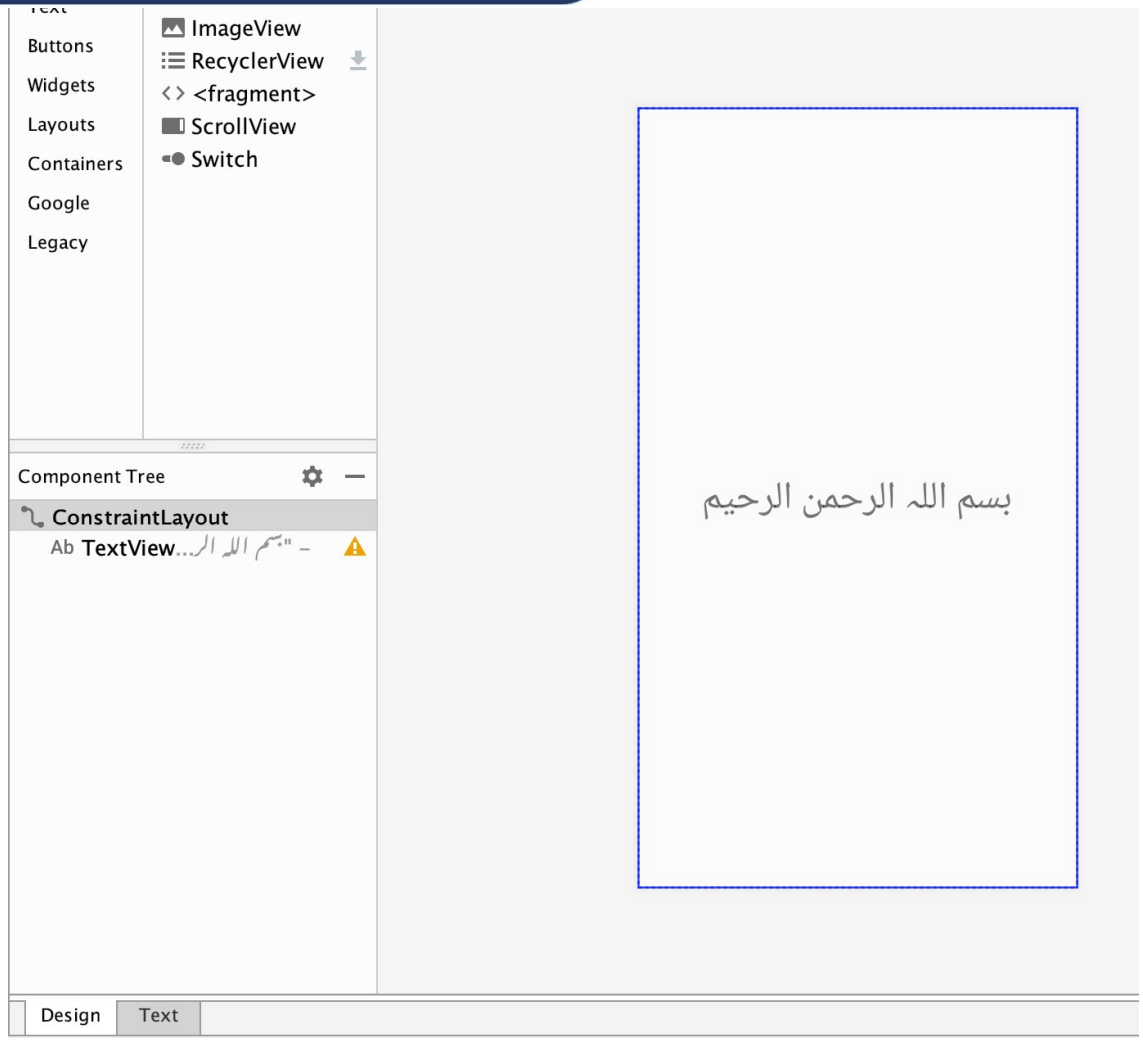
XML

1. Android's XML
    1. Web’ HTML
2. Root element View or ViewGroup object.
3. Once you've defined the root element, you can add additional layout objects or widgets as child elements to gradually build a View hierarchy that defines your layout

XML

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
              android:layout_width="match_parent"
              android:layout_height="match_parent"
              android:orientation="vertical" >
    <TextView android:id="@+id/text"
              android:layout_width="wrap_content"
              android:layout_height="wrap_content"
              android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hello, I am a Button" />
</LinearLayout>
```

XML Design view



XML Text view

PUCIT2021 > app > src > main > res > layout > activity_main.xml

MainActivity.java | activity_main.xml

```xml
<?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"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="بسم الله الرحمن الرحيم"
        android:textSize="36sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Design | Text

XML Text view

XML Anatomy

```
<?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"
```

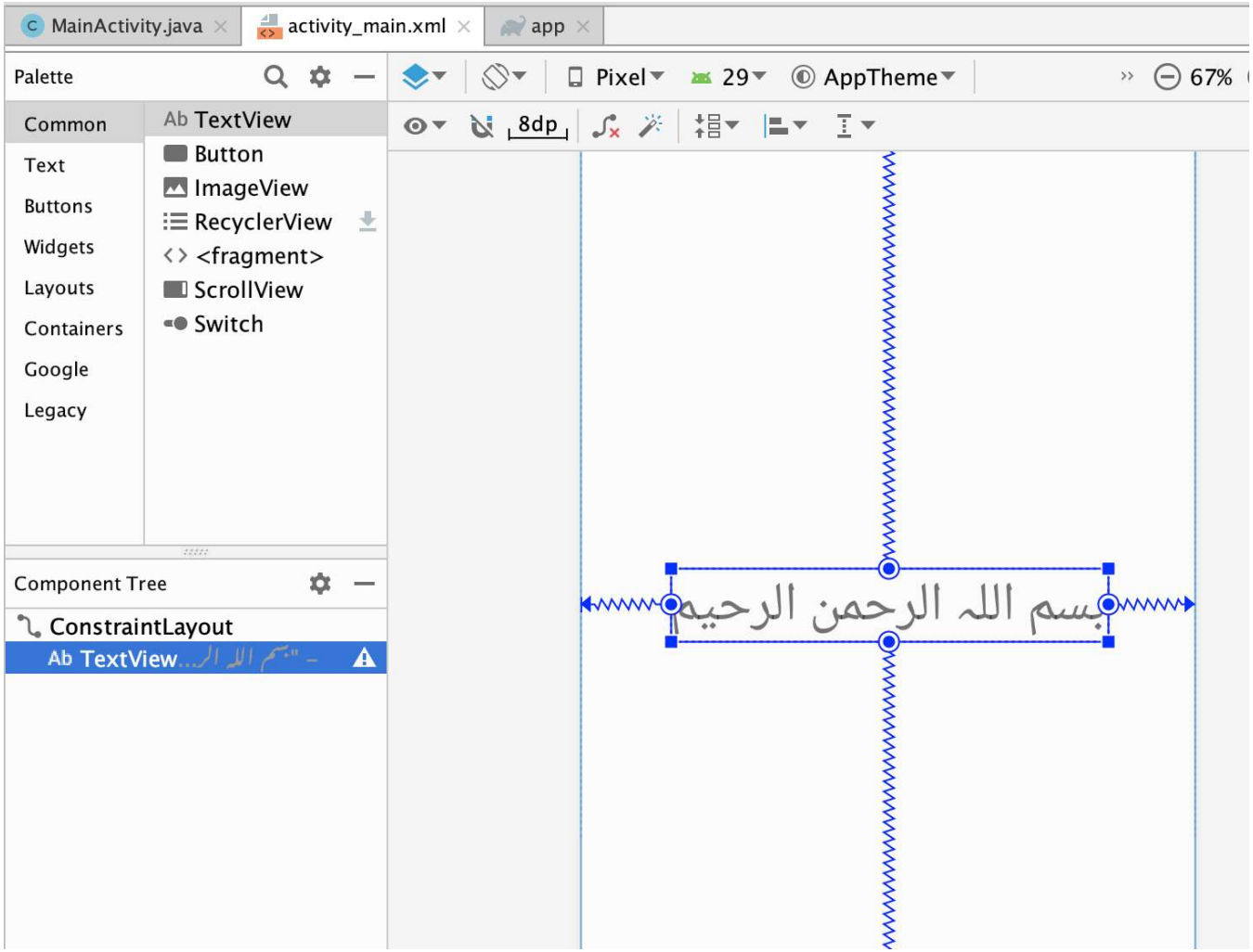
- **Look at the tag for the ConstraintLayout, and notice that it says androidx.constraintlayout.widget.ConstraintLayout instead of just ConstraintLayout like the TextView. This is because ConstraintLayout is part of Android Jetpack, which contains libraries of code which offers additional functionality on top of the core Android platform. Jetpack has useful functionality you can take advantage of to make building apps easier. You'll recognize this UI component is part of Jetpack because it starts with "androidx"**

XML Anatomy

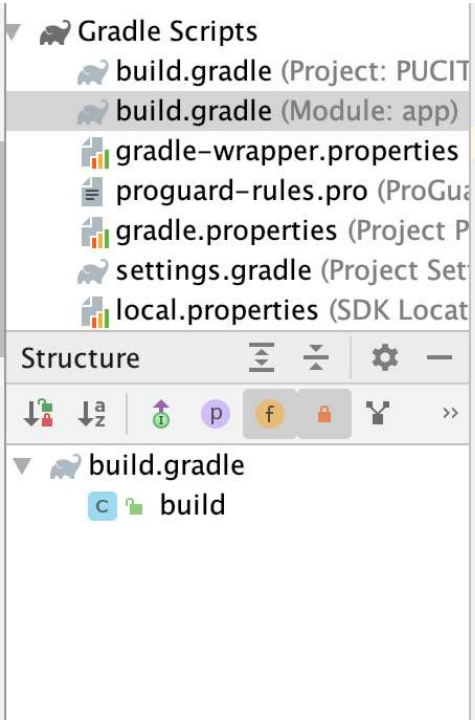
```
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">
```

- The xmlns stands for XML namespace, and each line defines a schema, or vocabulary for attributes related to those words. The android: namespace, for example, marks attributes that are defined by the Android system. All of the attributes in the layout XML begins with one of those namespaces.

Constraint Layout



Constraint Layout



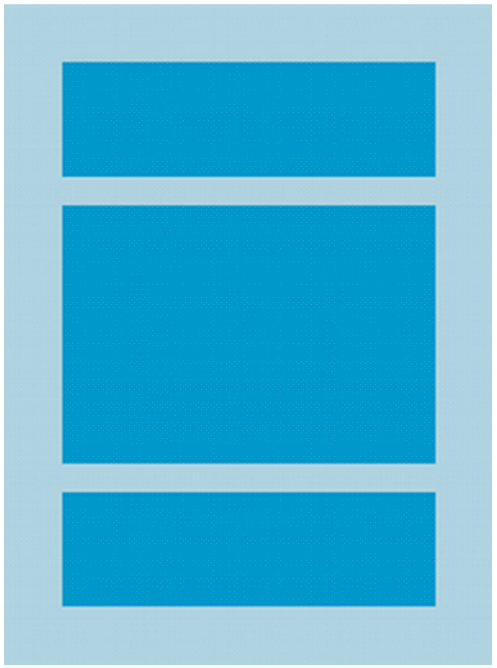

```
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'androidx.appcompat:appcompat:1.2.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.0.1'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'androidx.test:runner:1.3.0'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'
}
```

Load XML Resources

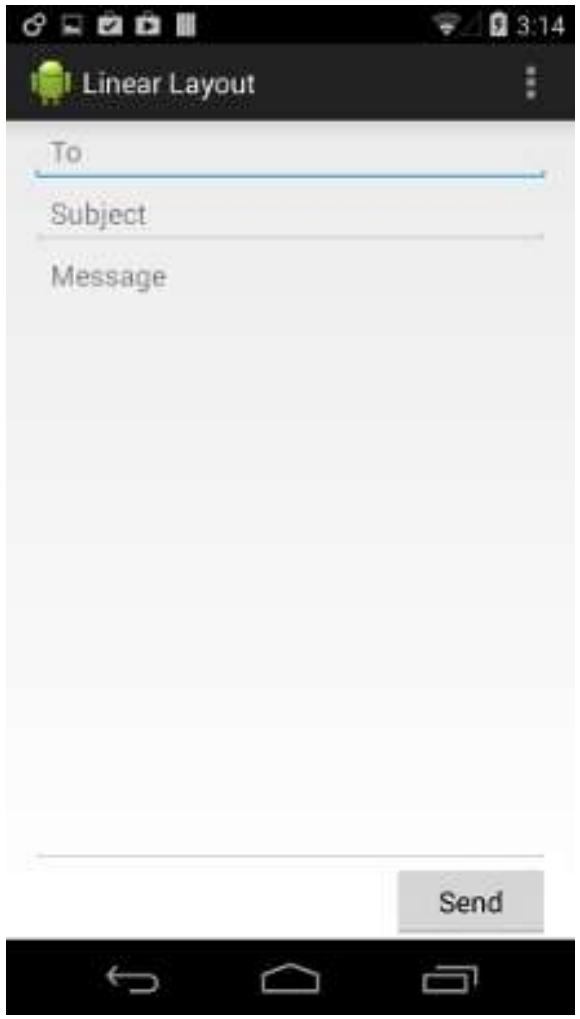
```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main_layout);
}
```

```
Button myButton = (Button) findViewById(R.id.my_button);
```

Linear Layout

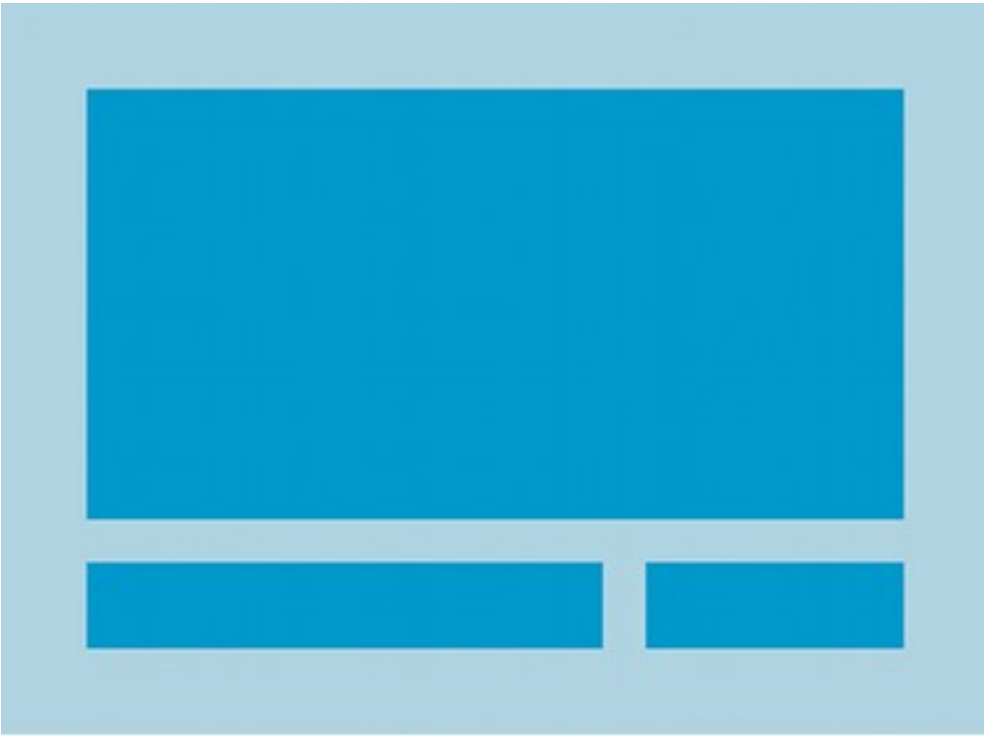


Linear Layout

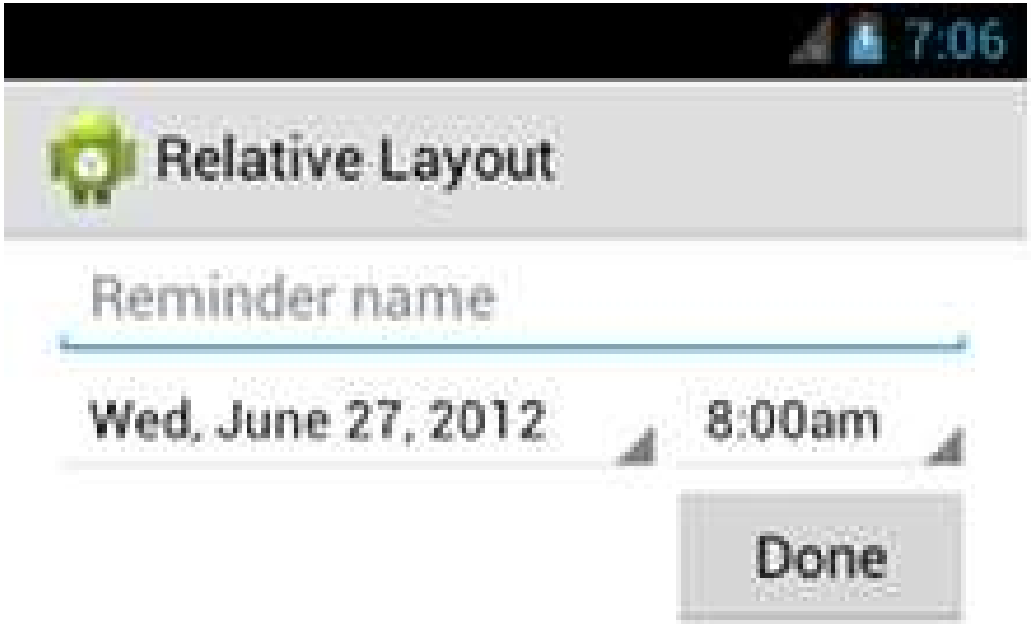



```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.and
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:orientation="vertical" >
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/to" />
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/subject" />
    <EditText
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:gravity="top"
        android:hint="@string/message" />
    <Button
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_gravity="right"
        android:text="@string/send" />
</LinearLayout>
```

Relative Layout

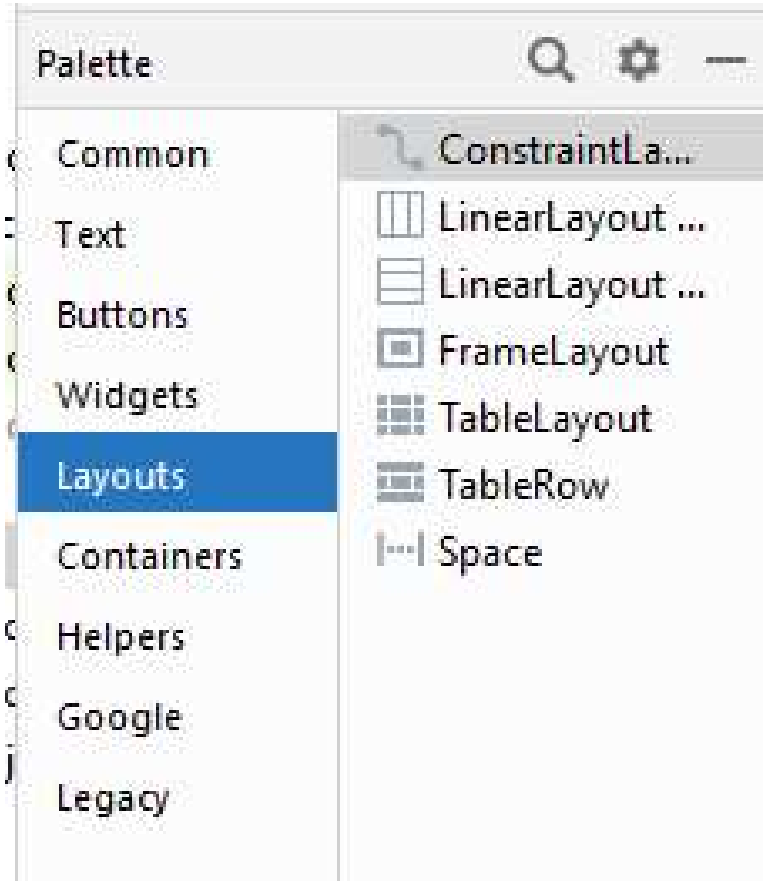


Relative Layout




```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="16dp"
    android:paddingRight="16dp" >
    <EditText
        android:id="@+id/name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/reminder" />
    <Spinner
        android:id="@+id/dates"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_below="@id/name"
        android:layout_alignParentLeft="true"
        android:layout_toLeftOf="@+id/times" />
    <Spinner
        android:id="@id/times"
        android:layout_width="96dp"
        android:layout_height="wrap_content"
        android:layout_below="@id/name"
        android:layout_alignParentRight="true" />
    <Button
        android:layout_width="96dp"
        android:layout_height="wrap_content"
        android:layout_below="@id/times"
        android:layout_alignParentRight="true"
        android:text="@string/done" />
</RelativeLayout>
```

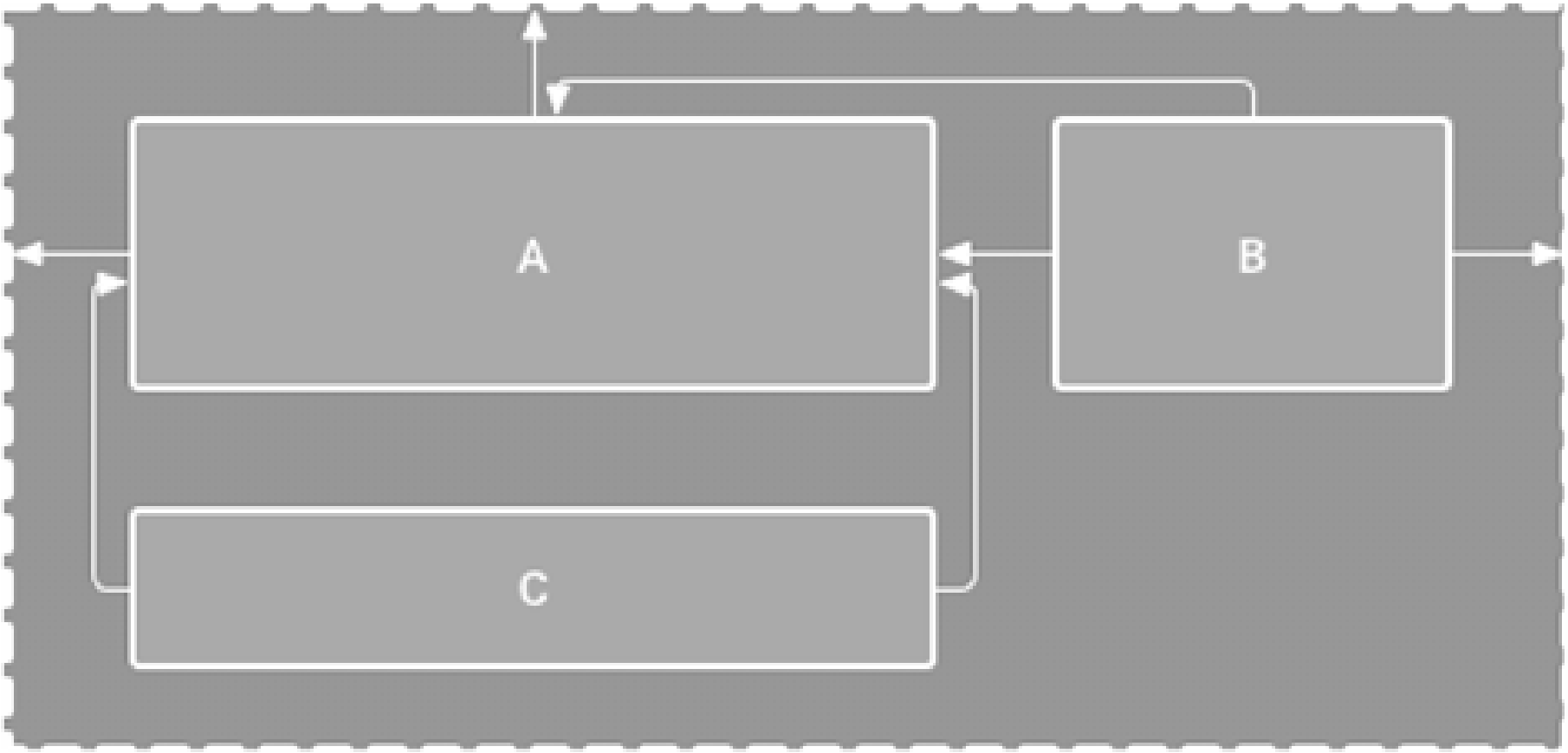
Layouts


Palette
Common
Text
Buttons
Widgets
Layouts
Containers
Helpers
Google
Legacy
ConstraintLa...
LinearLayout ...
LinearLayout ...
FrameLayout
TableLayout
TableRow
Space

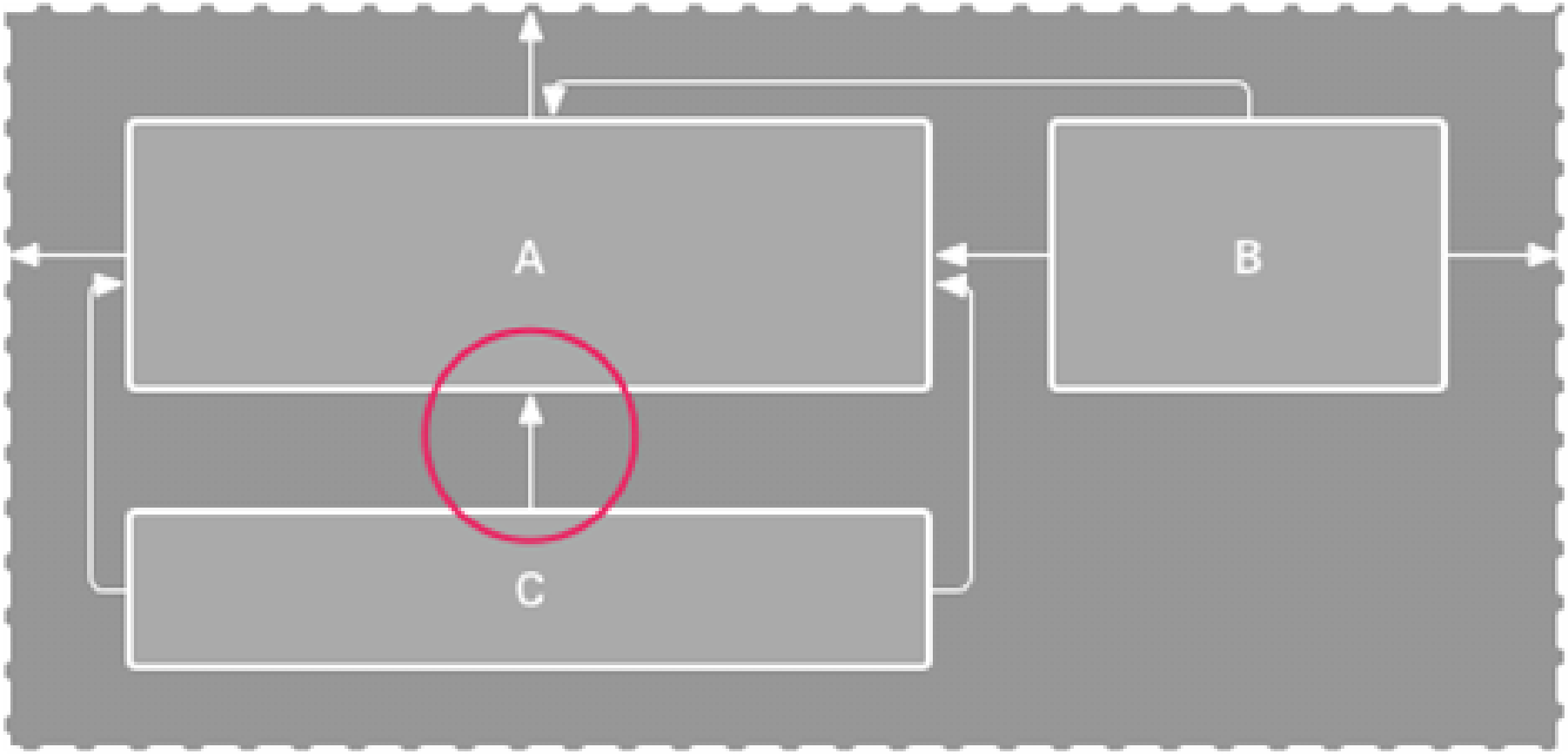
Constraint Layout

1. To define a view's position in ConstraintLayout, you must add at least one horizontal and one vertical constraint for the view.
2. Each constraint represents a connection or alignment to another view, the parent layout, or an invisible guideline.
3. Each constraint defines the view's position along either the vertical or horizontal axis; so each view must have a minimum of one constraint for each axis, but often more are necessary.
4. If a view has no constraints when you run your layout on a device, it is drawn at position [0,0]

Guess



Answer

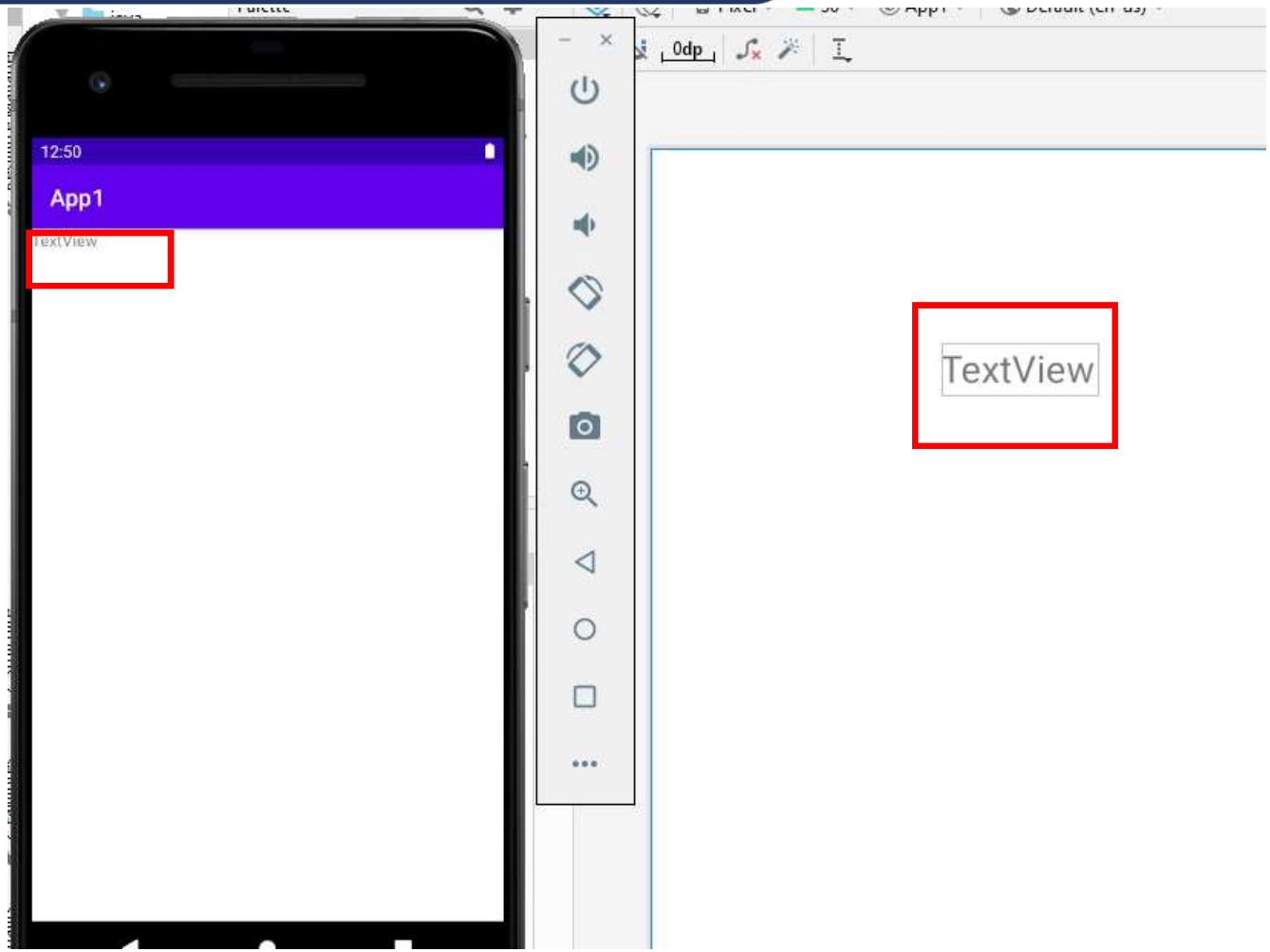


Required Dependency

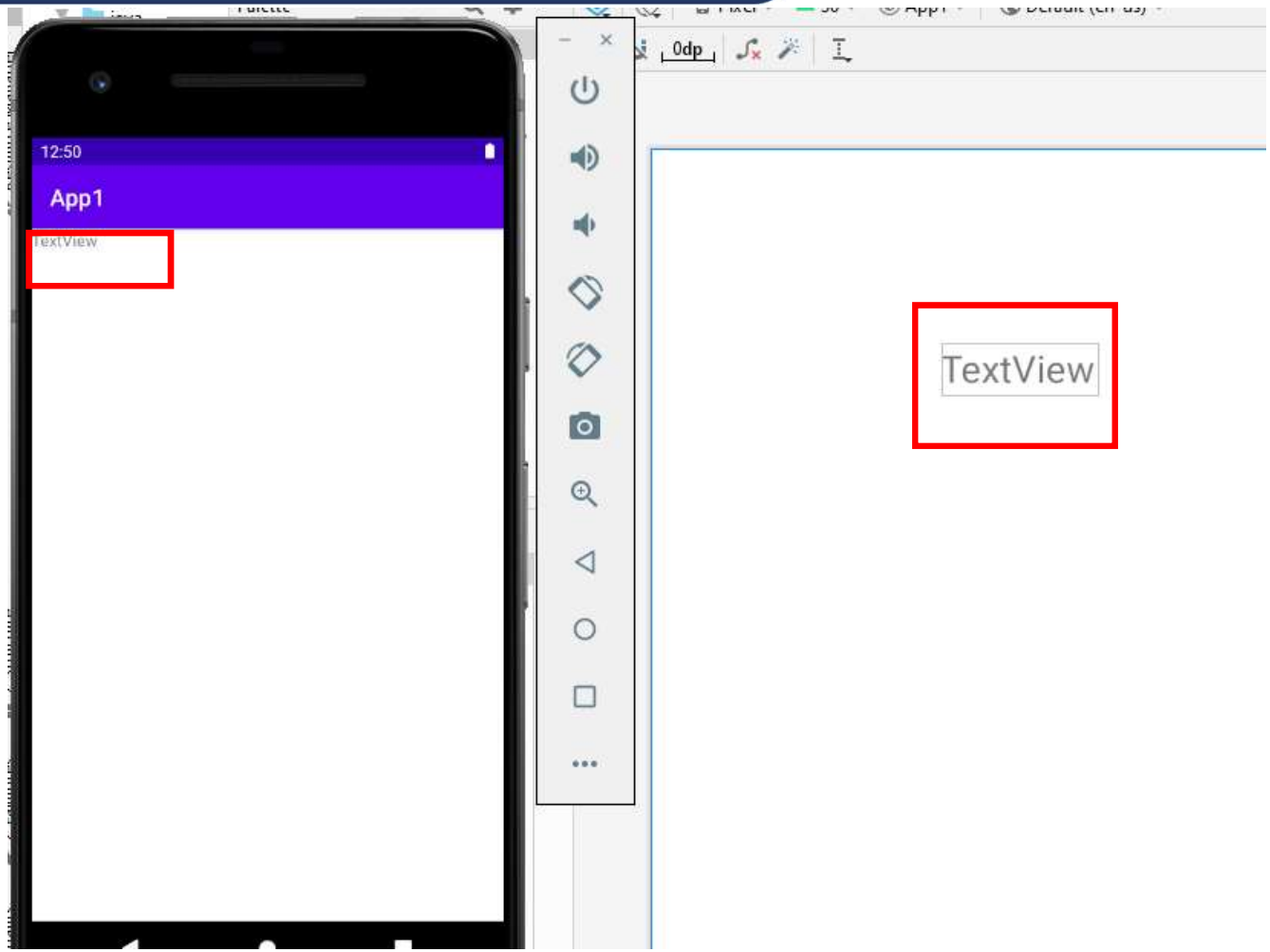
```
dependencies {
    implementation "androidx.constraintlayout:constraintlayout:2.0.4"
}
```

```
22  dependencies {
23      implementation fileTree(dir: 'libs', include: ['*.jar'])
24      implementation 'androidx.appcompat:appcompat:1.2.0'
25      implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
26      testImplementation 'junit:junit:4.12'
27      androidTestImplementation 'androidx.test:runner:1.3.0'
28      androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'
29  }
30
```

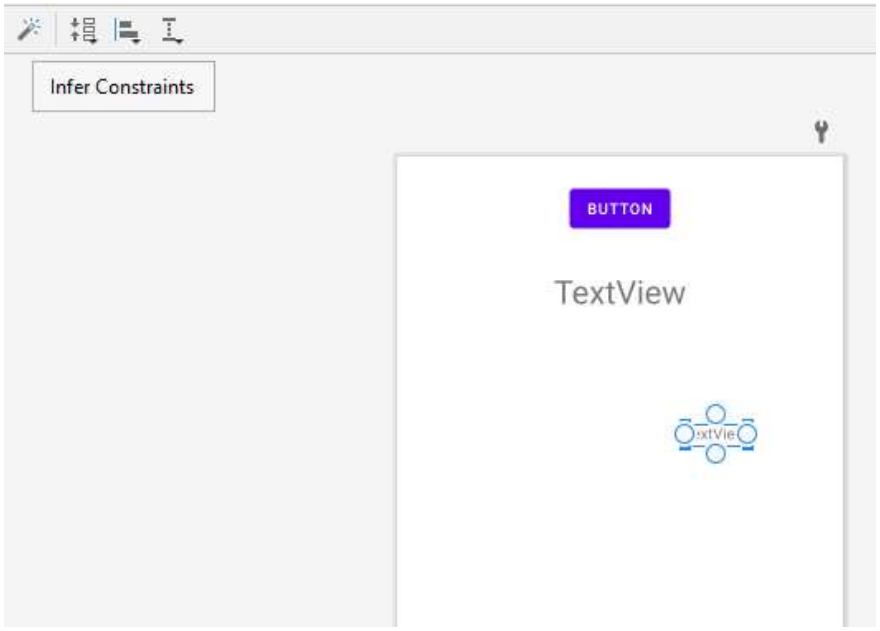
Default position of view

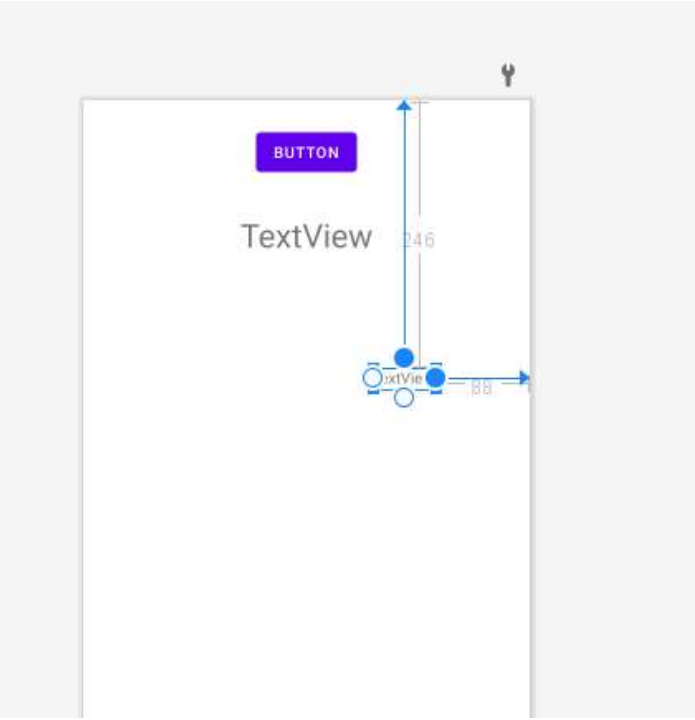


Default position of view

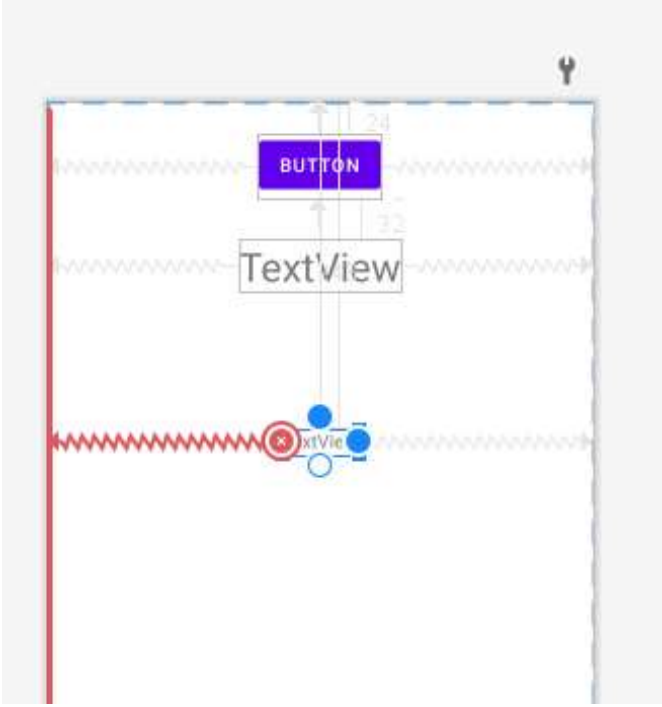


Infer Constraints

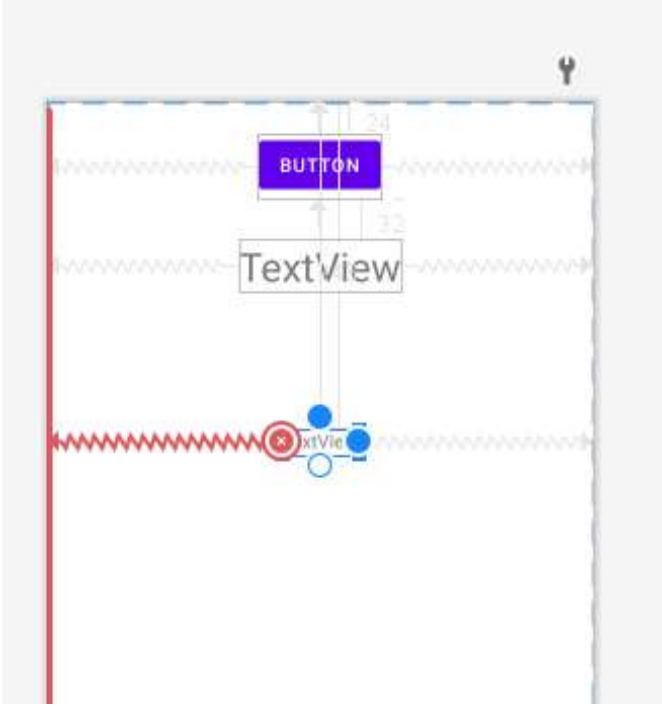




Delete Constraints

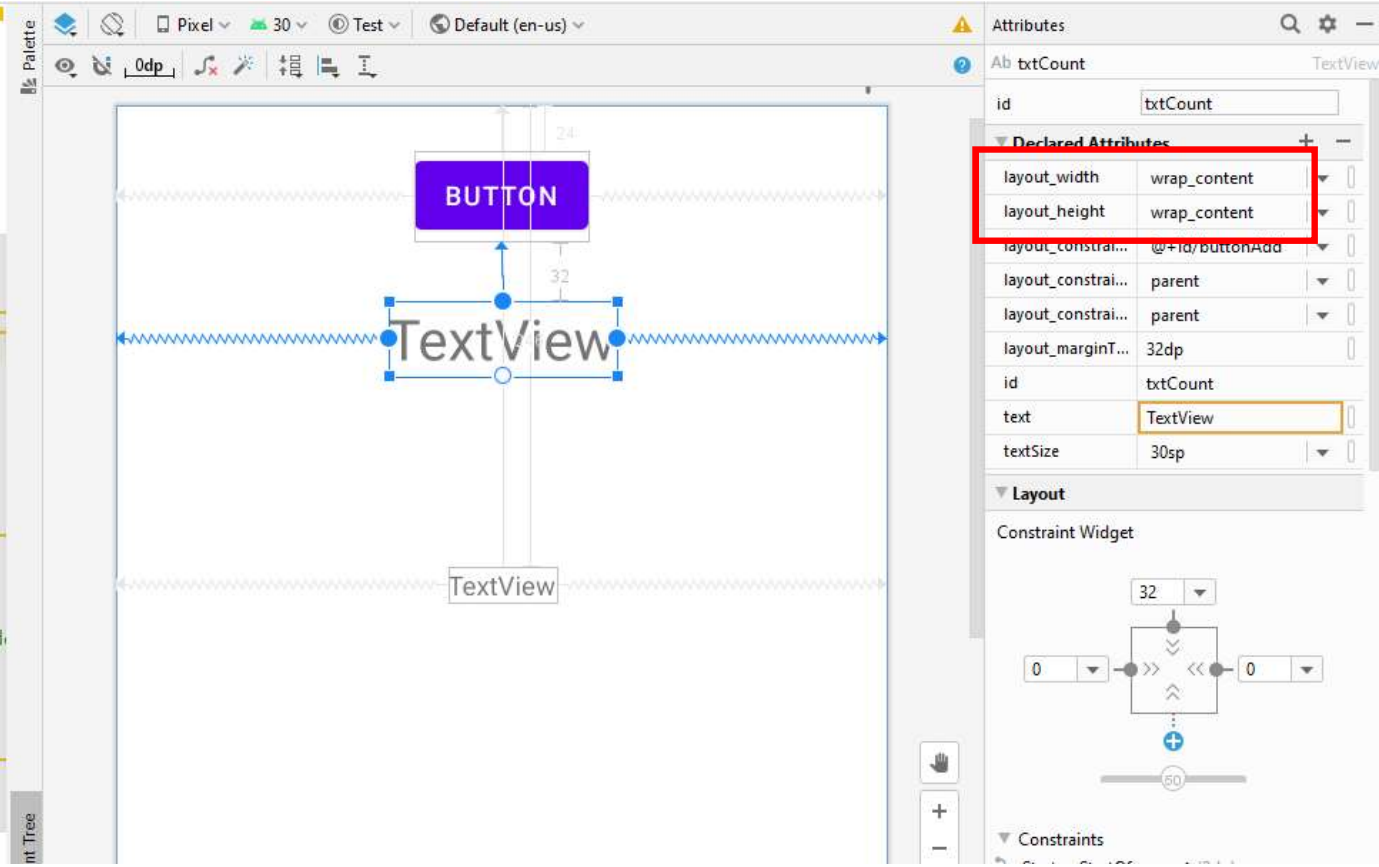


Delete Constraints



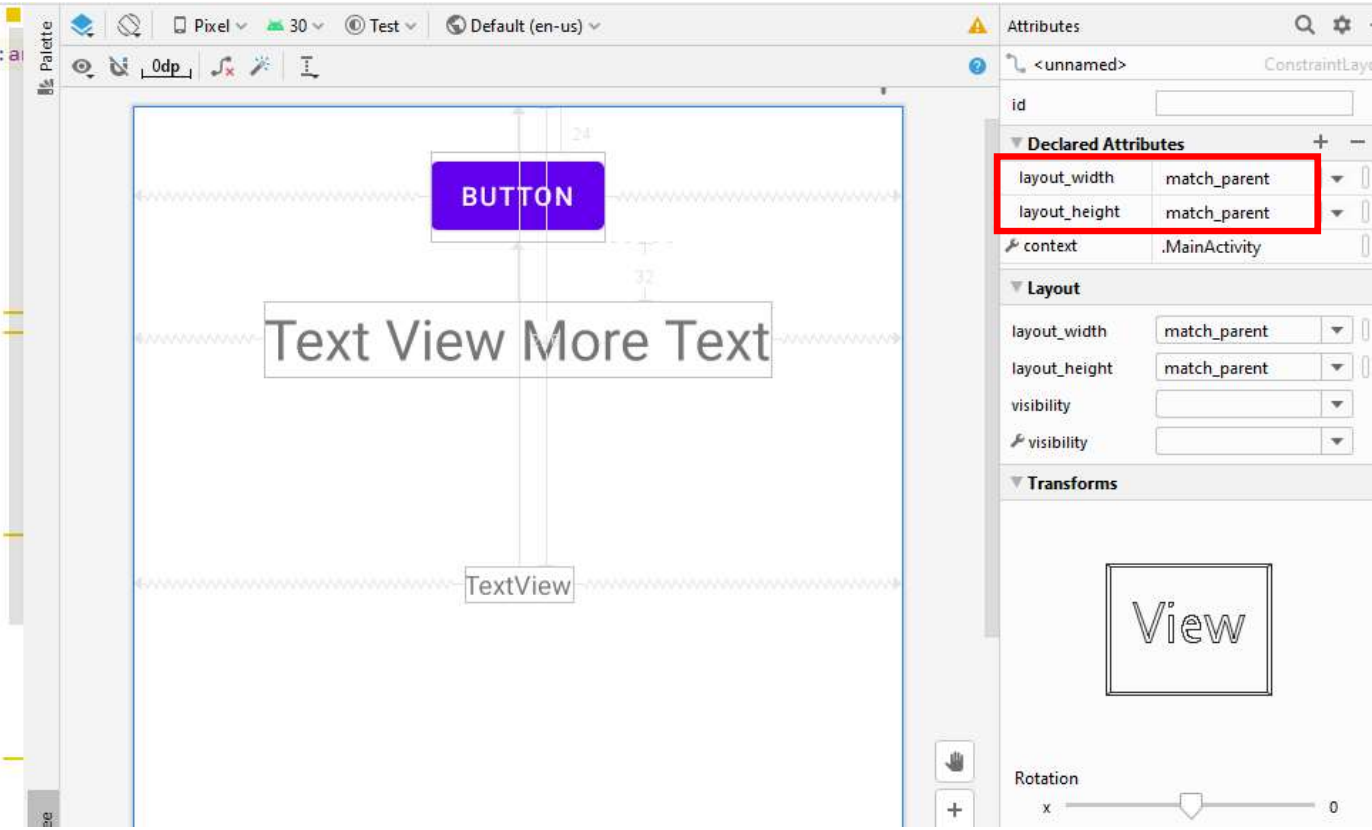
match_parent Attribute

```
    android:id="@+id/buttonAdd"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="24dp"
    android:onClick="Add"
    android:text="Button"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<TextView
    android:id="@+id/txtCount"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:text="TextView"
    android:textSize="30sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/buttonAd

<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="246dp"
```



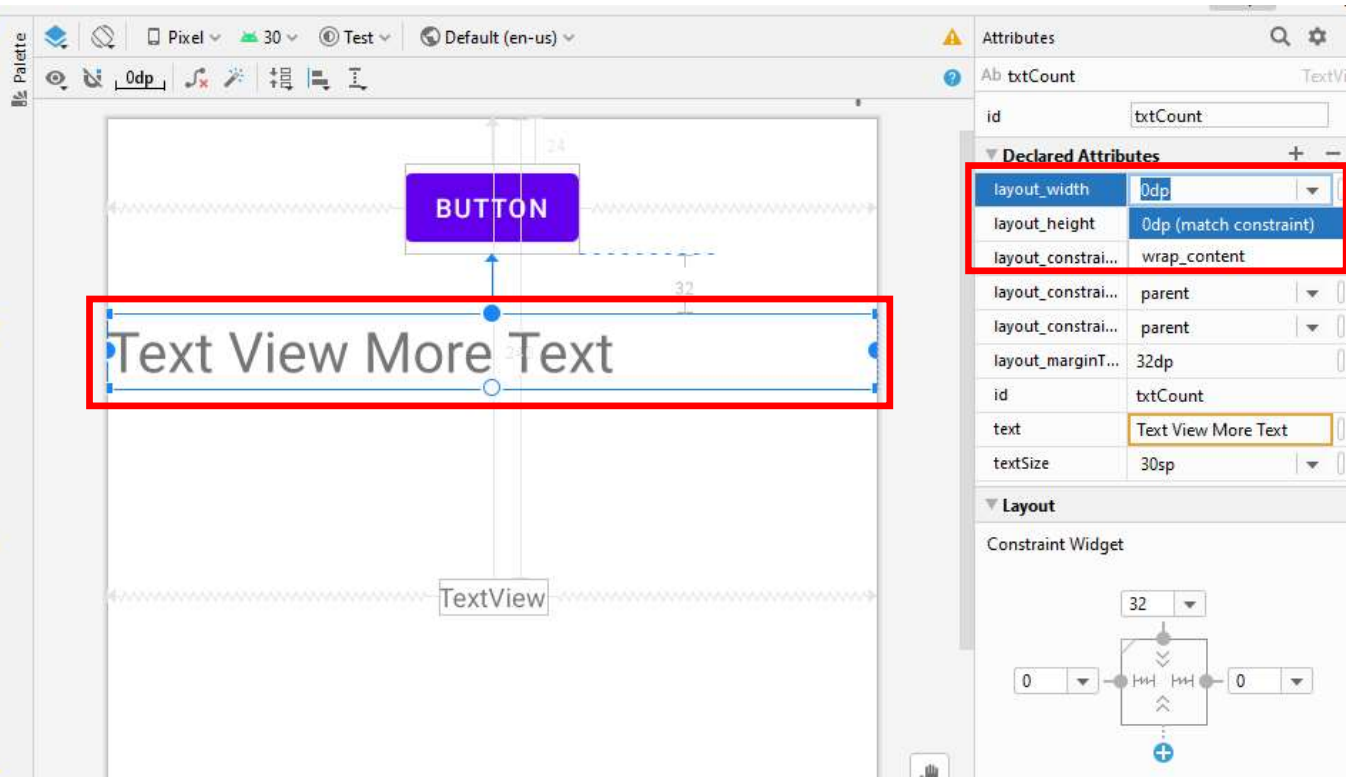
match_parent Attribute

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:a
    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"
    tools:context=".MainActivity">


    <Button
        android:id="@+id/buttonAdd"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="24dp"
        android:onClick="Add"
        android:text="Button"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/txtCount"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="32dp"
```

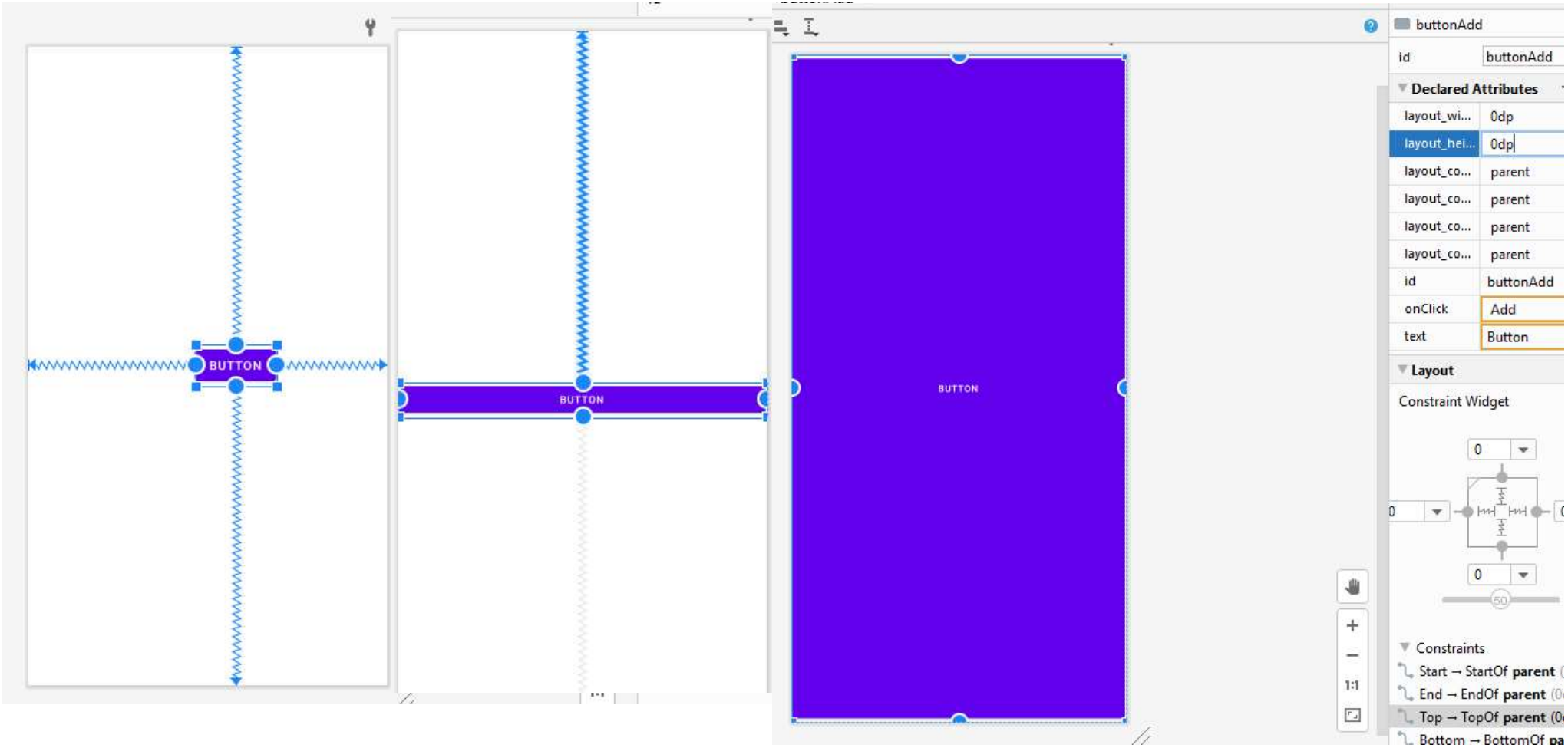


match constraint Attribute

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:a
    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"
    tools:context=".MainActivity">


    <Button
        android:id="@+id/buttonAdd"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="24dp"
        android:onClick="Add"
        android:text="Button"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/txtCount"
        android:layout_width="0dp"
```

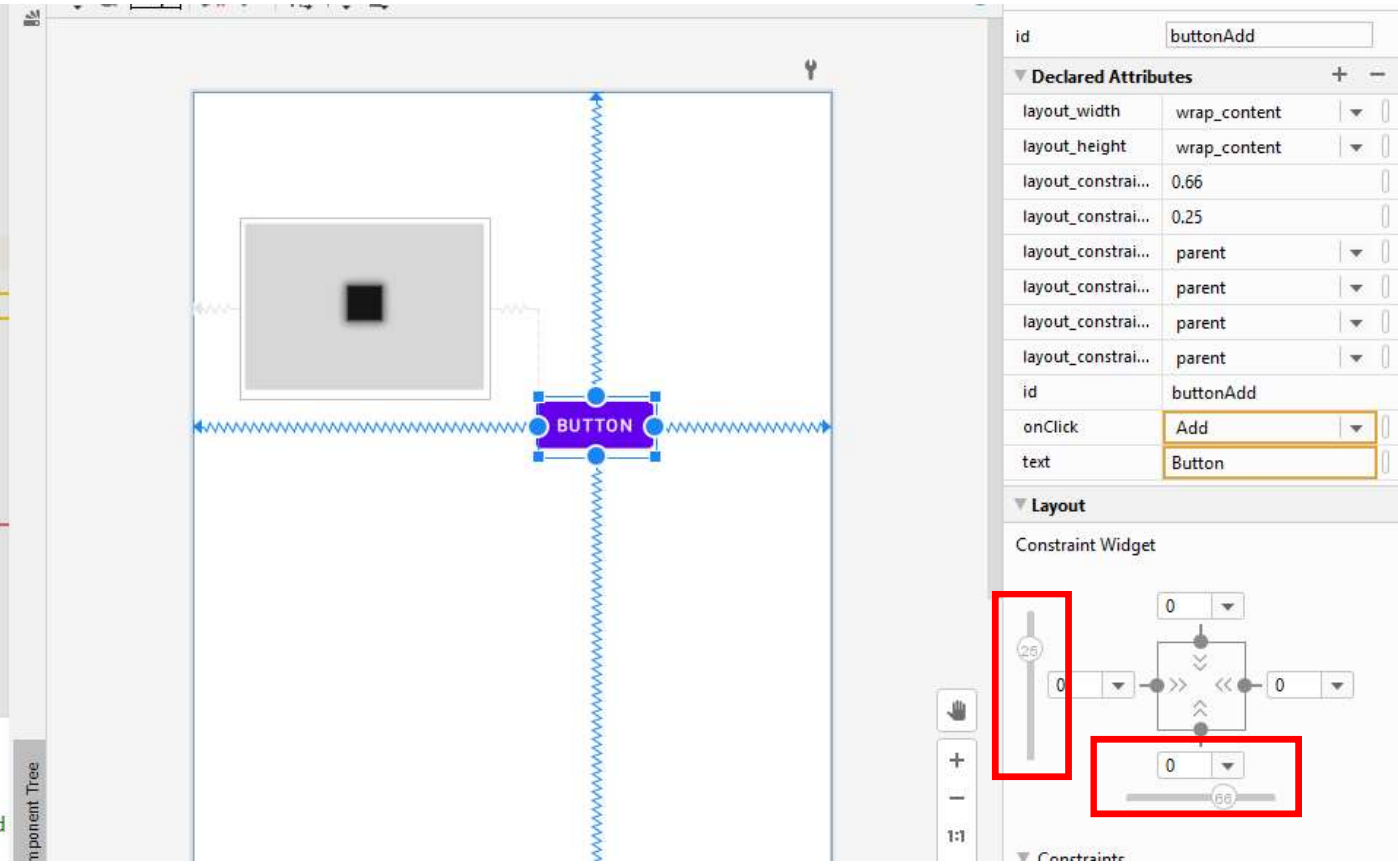


match constraint Attribute



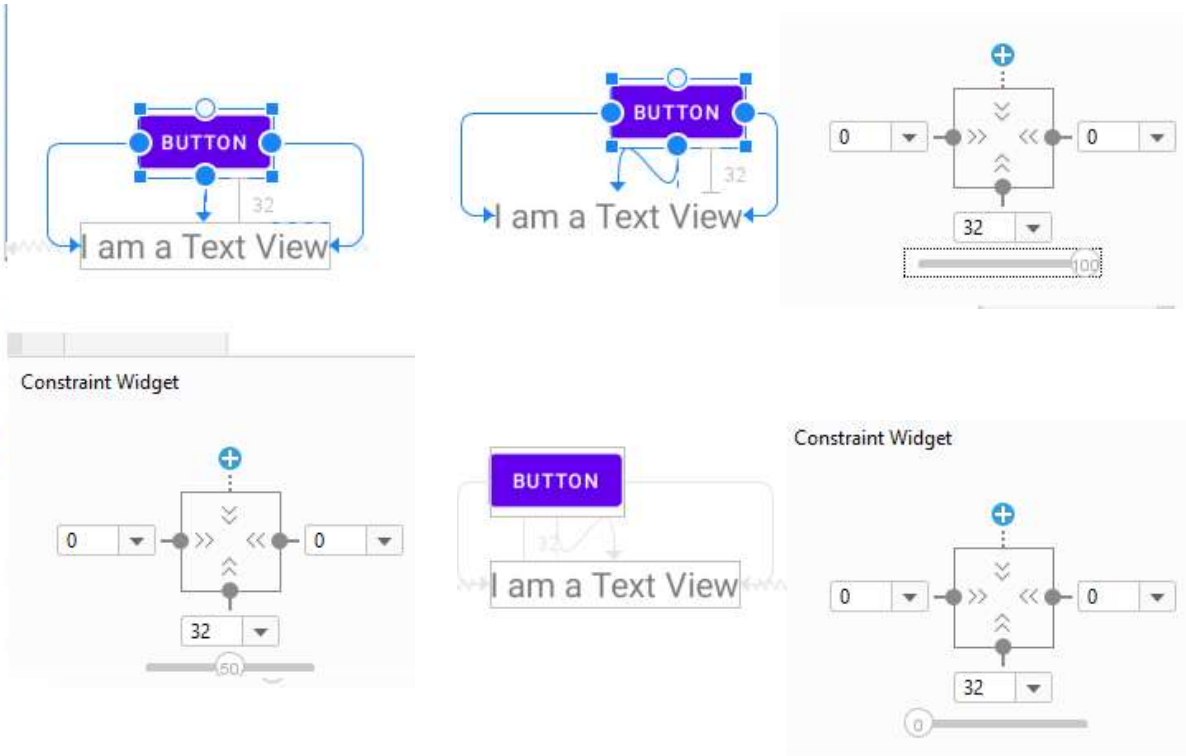
bias Attribute

```
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"
tools:context=".MainActivity">

<Button
    android:id="@+id/buttonAdd"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="Add"
    android:text="Button"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.66"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.25" />

<ImageButton
    android:id="@+id/imageButton"
    android:layout_width="197dp"
    android:layout_height="143dp"
    app:layout_constraintEnd_toStartOf="@+id/buttonAdd
    app:layout_constraintStart_toStartOf="parent"
```



bias Attribute

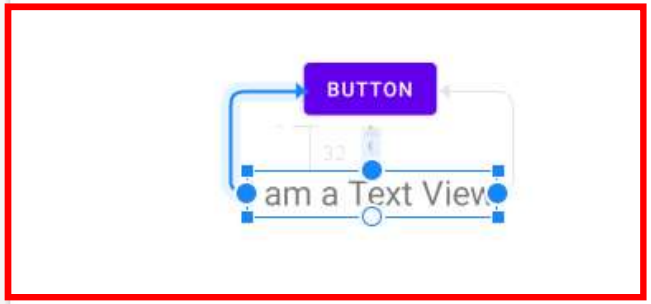
```
<Button
    android:id="@+id/buttonAdd"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="32dp"
    android:onClick="Add"
    android:text="Button"
    app:layout_constraintBottom_toTopOf="@+id/textVie
    app:layout_constraintEnd_toEndOf="@+id/textView4"
    app:layout_constraintStart_toStartOf="@+id/textVi
```

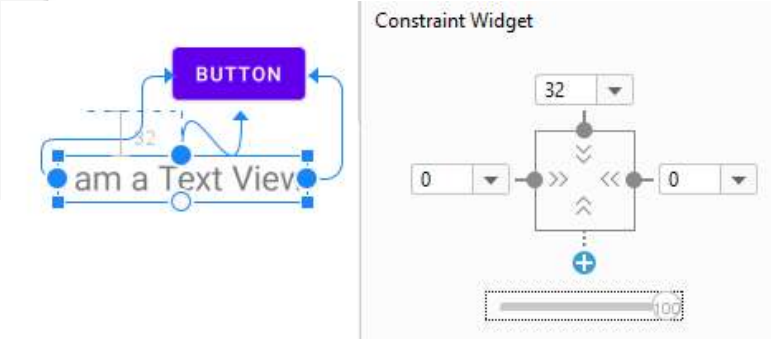


محدود اختيارات

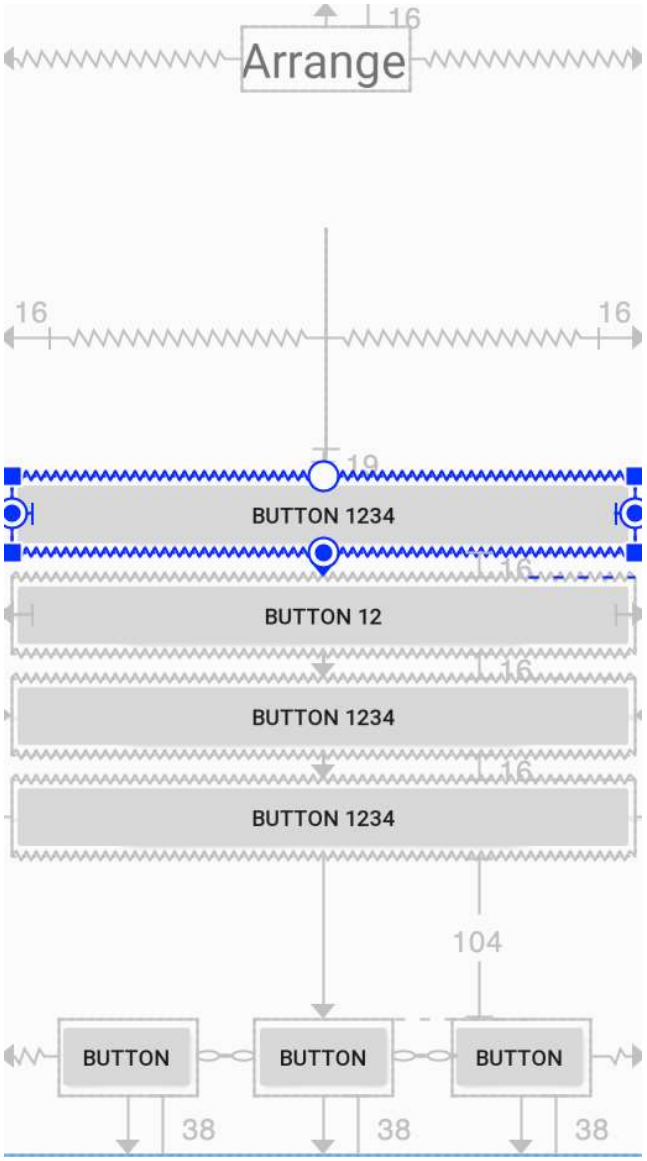
bias Attribute

```
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.25" />

<TextView
    android:id="@+id/textView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:text="I am a Text View"
    android:textSize="24dp"
    app:layout_constraintEnd_toEndOf="@+id/buttonAdd"
    app:layout_constraintStart_toStartOf="@+id/buttonA
    app:layout_constraintTop_toBottomOf="@+id/buttonAd

androidx.constraintlayout.widget.ConstraintLayout>
```

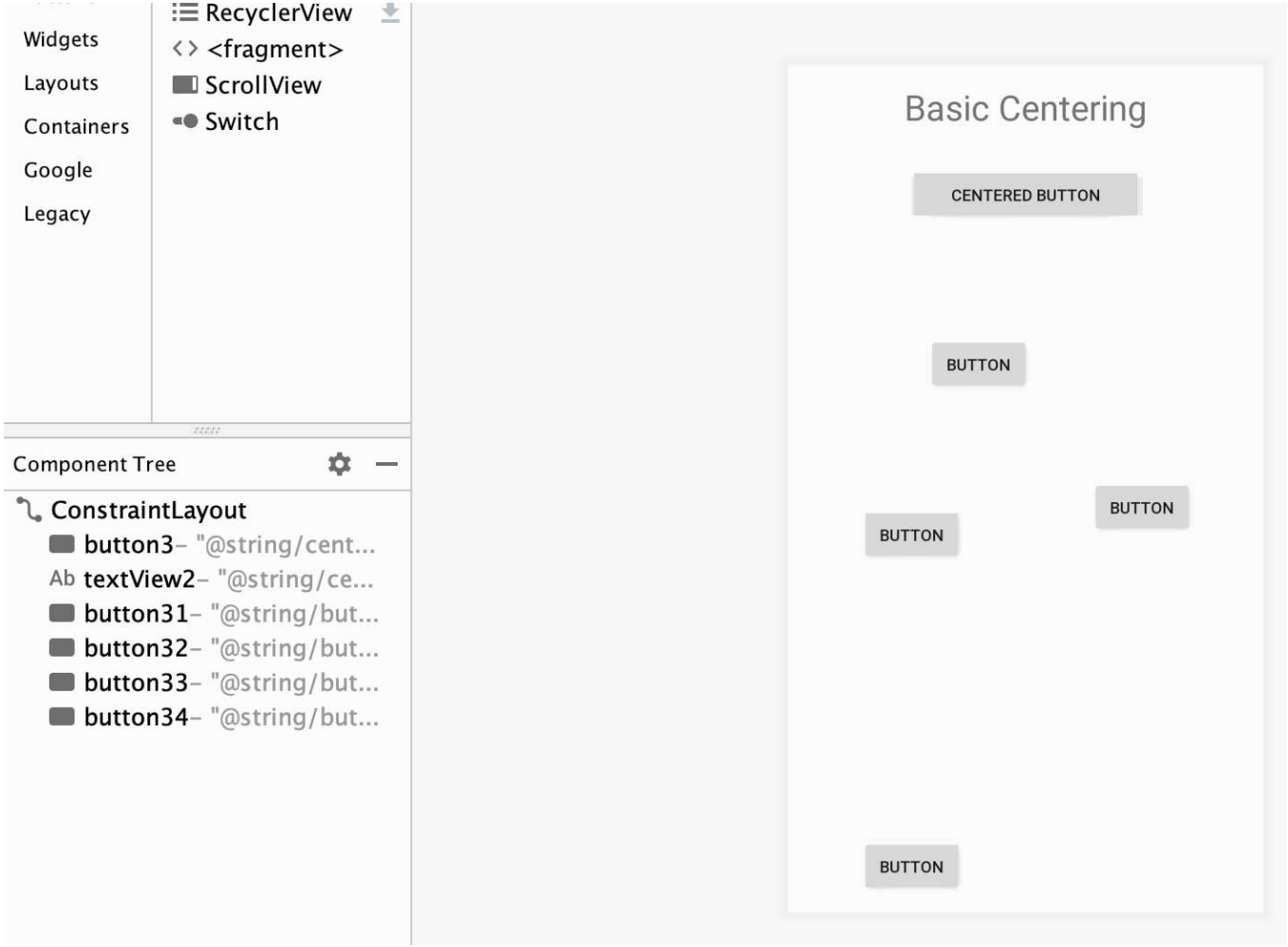




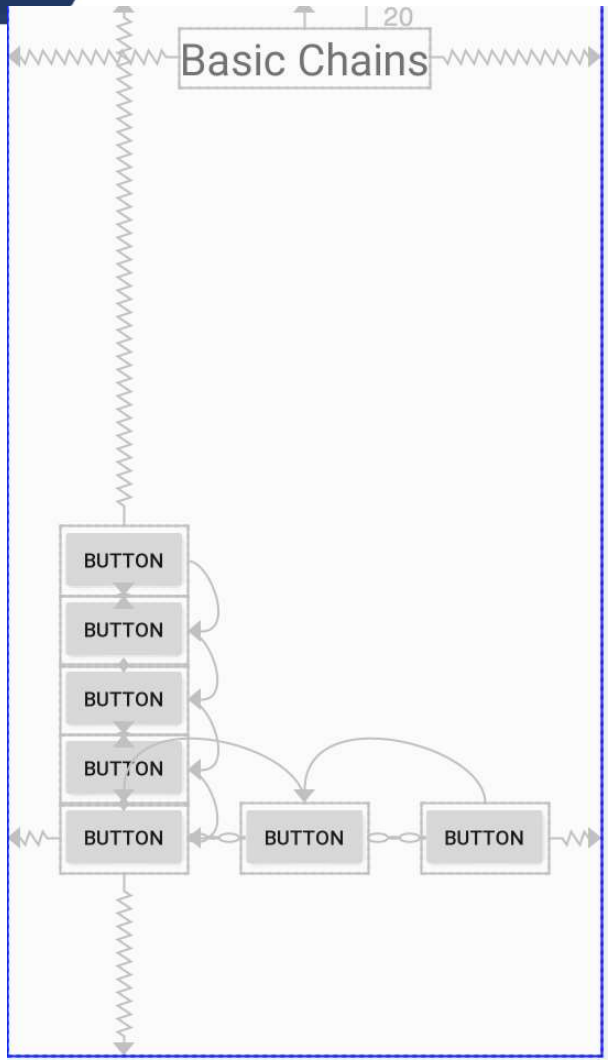
Constraint layout



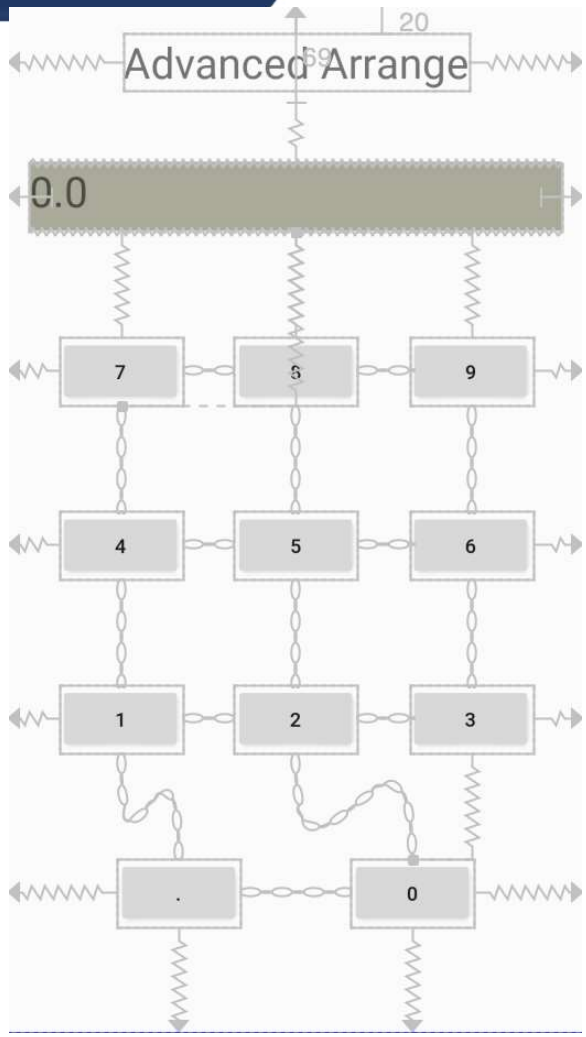
Constraint layout



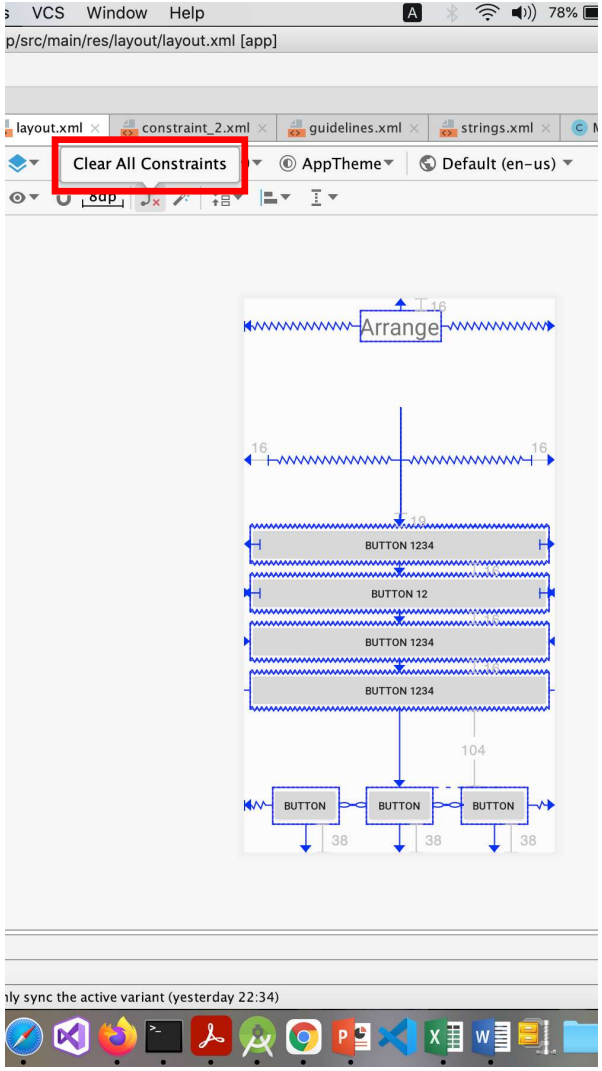
Constraint layout



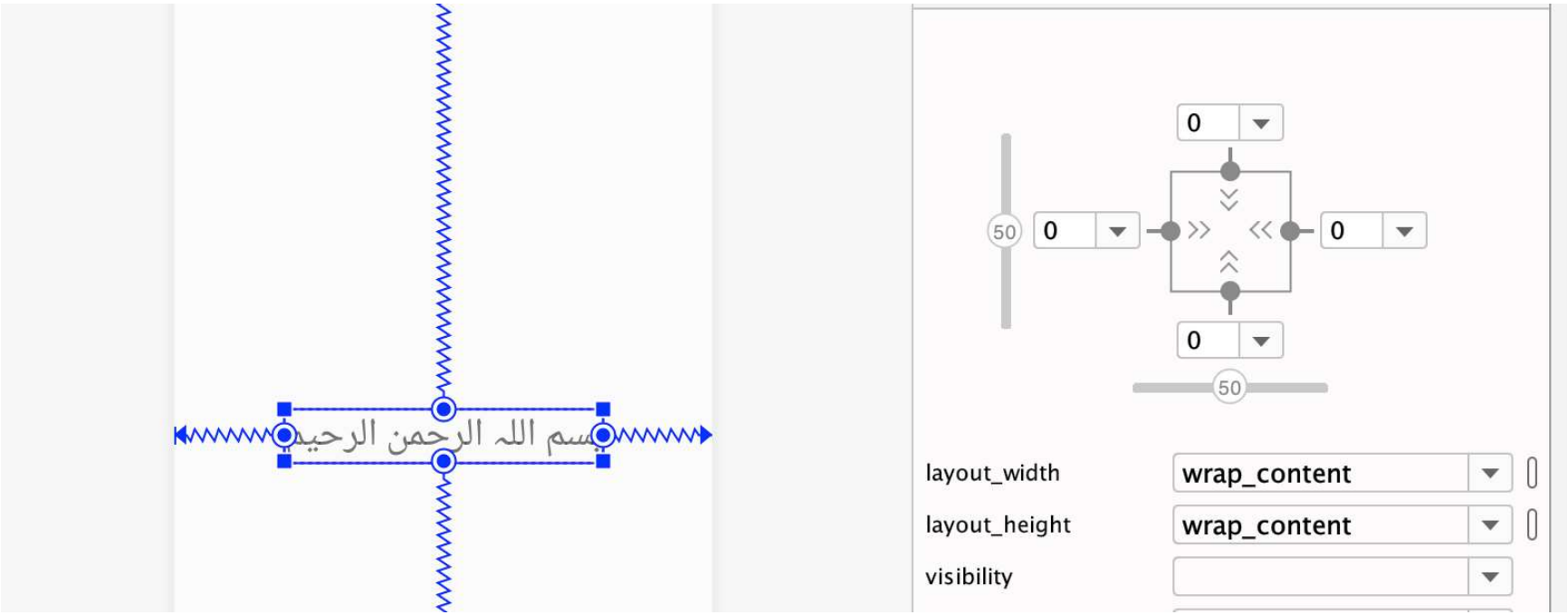
Constraint layout



Clear all constraints



Layout Constraint



Add or Remove Constraint

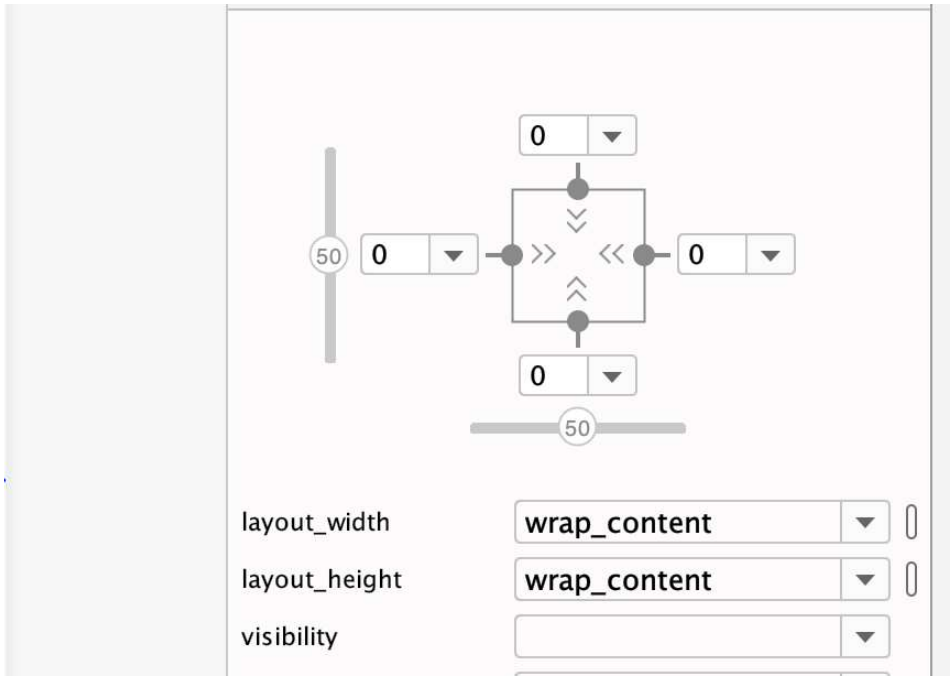
1. Drag a view from the **Palette** window into the editor.

   When you add a view in a `ConstraintLayout`, it displays a bounding box with square resizing handles on each corner and circular constraint handles on each side.

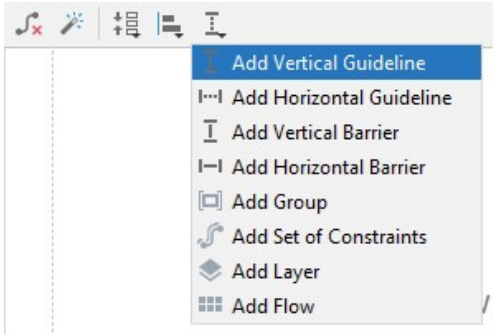
2. Click the view to select it.

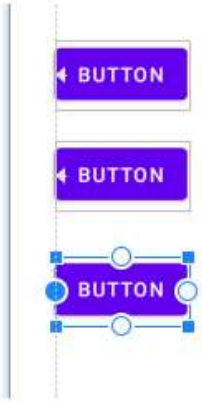
3. Do one of the following:

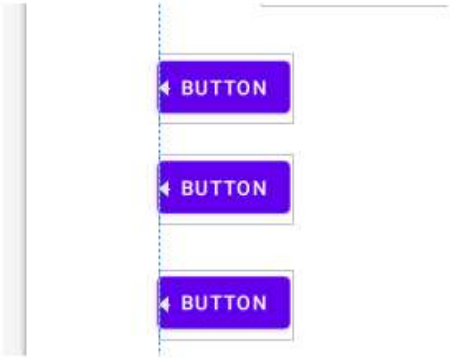
   - Click a constraint handle and drag it to an available anchor point. This point can be the edge of another view, the edge of the layout, or a guideline. Notice that as you drag the constraint handle, the Layout Editor shows potential connection anchors and blue overlays.



Guidelines


Add Vertical Guideline
Add Horizontal Guideline
Add Vertical Barrier
Add Horizontal Barrier
Add Group
Add Set of Constraints
Add Layer
Add Flow


BUTTON
BUTTON
BUTTON


BUTTON
BUTTON
BUTTON

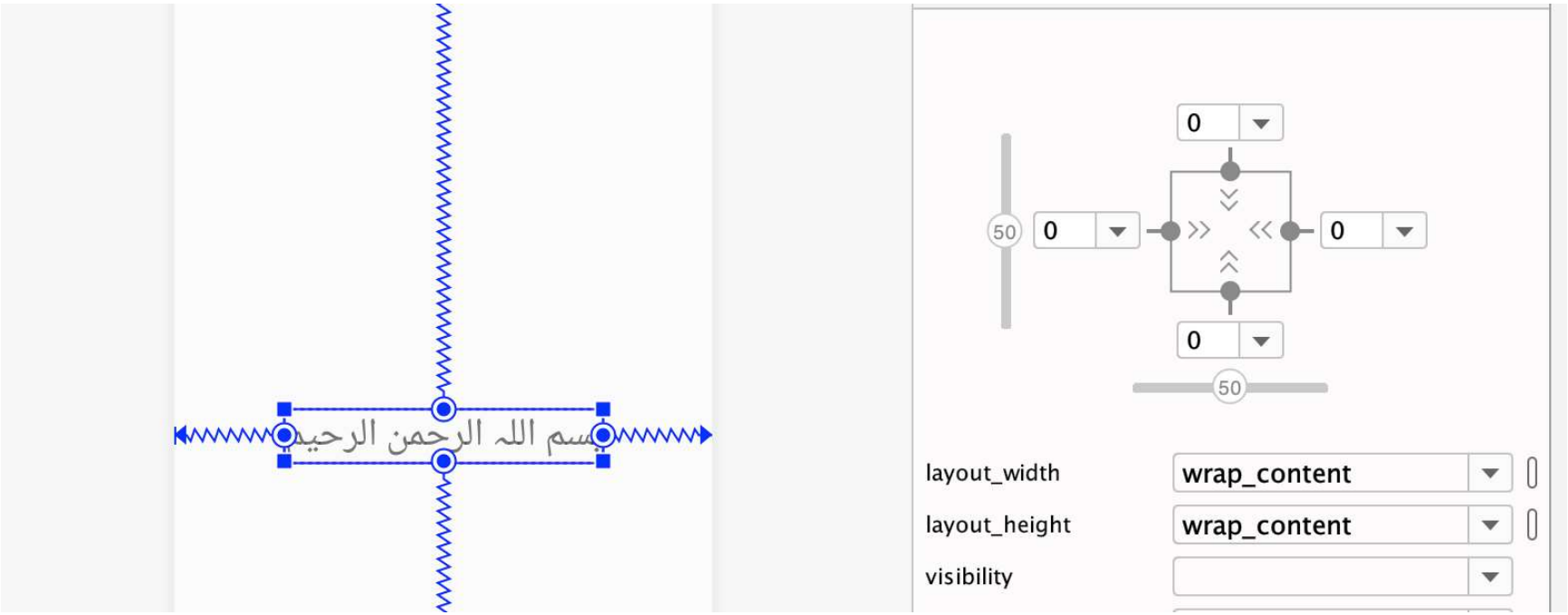
Barrier

A Barrier references multiple widgets as input, and creates a virtual guideline based on the most extreme widget on the specified side. For example, a left barrier will align to the left of all the referenced

With the barrier direction set to start, we will have the following result:

Reversely, with the direction set to end, we will have:

Constraint Layout



زخم کا مرہم درد کا اپنے درماں بیچ کے آئے
ہیں
ہم لمحوں کا سودا کر کے صدیاں بیچ کے آئے
ہیں
بکنے پر جب آ ہی گئے تھے اونچے مول تو
بکتے ہم
ہم کو ہمارے رہبر لیکن ارزاں بیچ کے آئے
ہیں

حقیر ذرّے فضا میں بکھر گئے ہوتے
تباہیوں کی حدوں سے گذر گئے ہوتے
گناہ گار تو ایسے تھے ہم کہ بس توبہ
خدا کریم نہ ہوتا تو مر گئے ہوتے

•

دلِ حزیں تو جو دنیا میں الجھا رہتا ہے
خدا کرے کہ تجھے عشقِ مصطفیٰؐ ہو جائے

•

ویسے تو ہر اک خواب بھی سچا نہیں ہوتا
جس خواب میں وہؐ آئیں وہ جھوٹا نہیں ہوتا

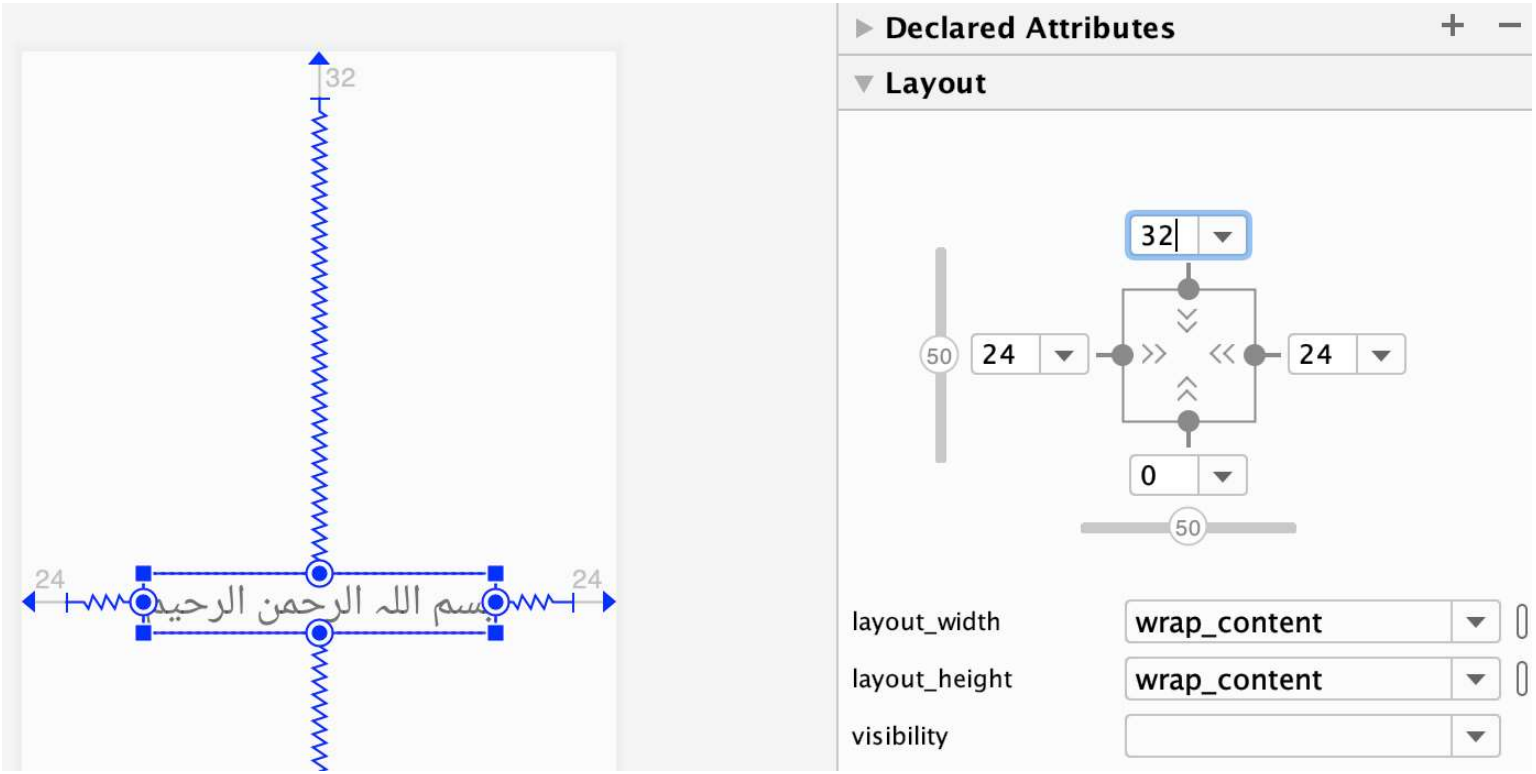
•

دیارِ مصطفیٰؐ کی حاضری ہے
دلِ مضطر سلیقے سے دھڑکنا

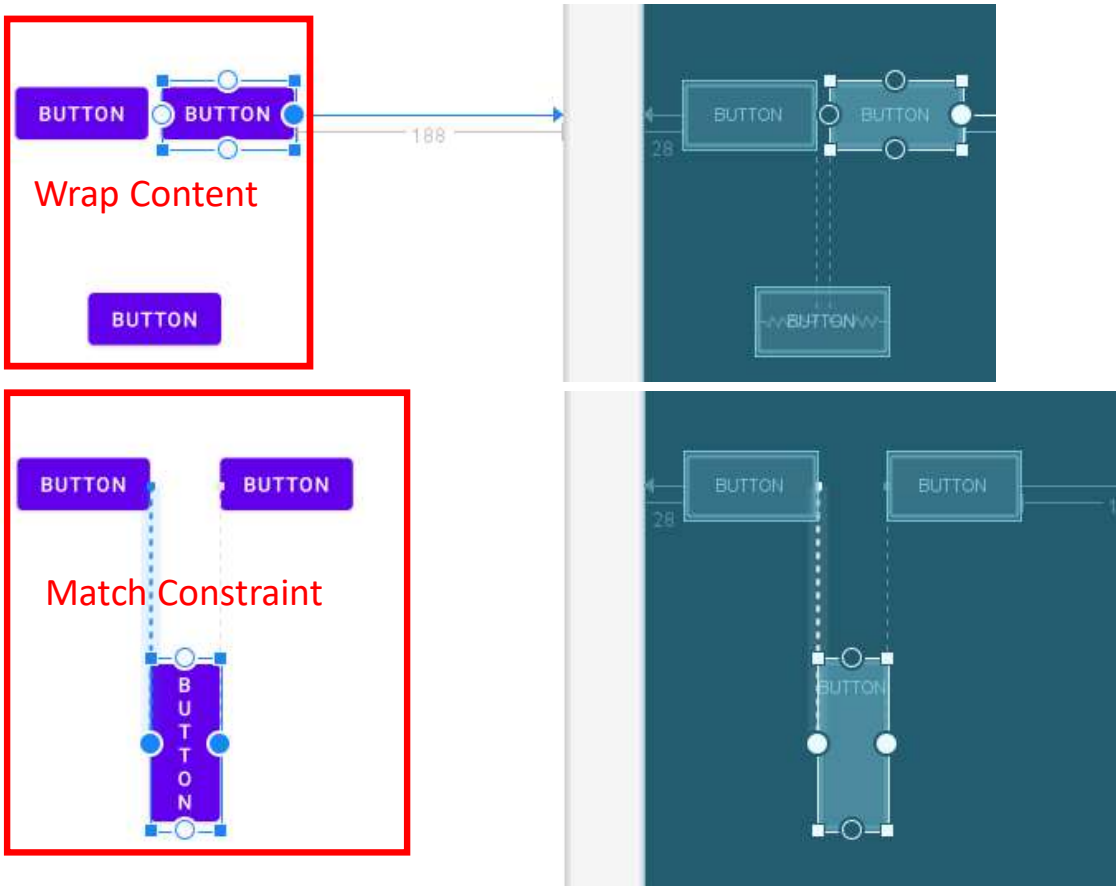
•

آپؐ کے گھر سے ایسے لوٹے ہم
آج تک اپنے گھر نہیں پہنچے

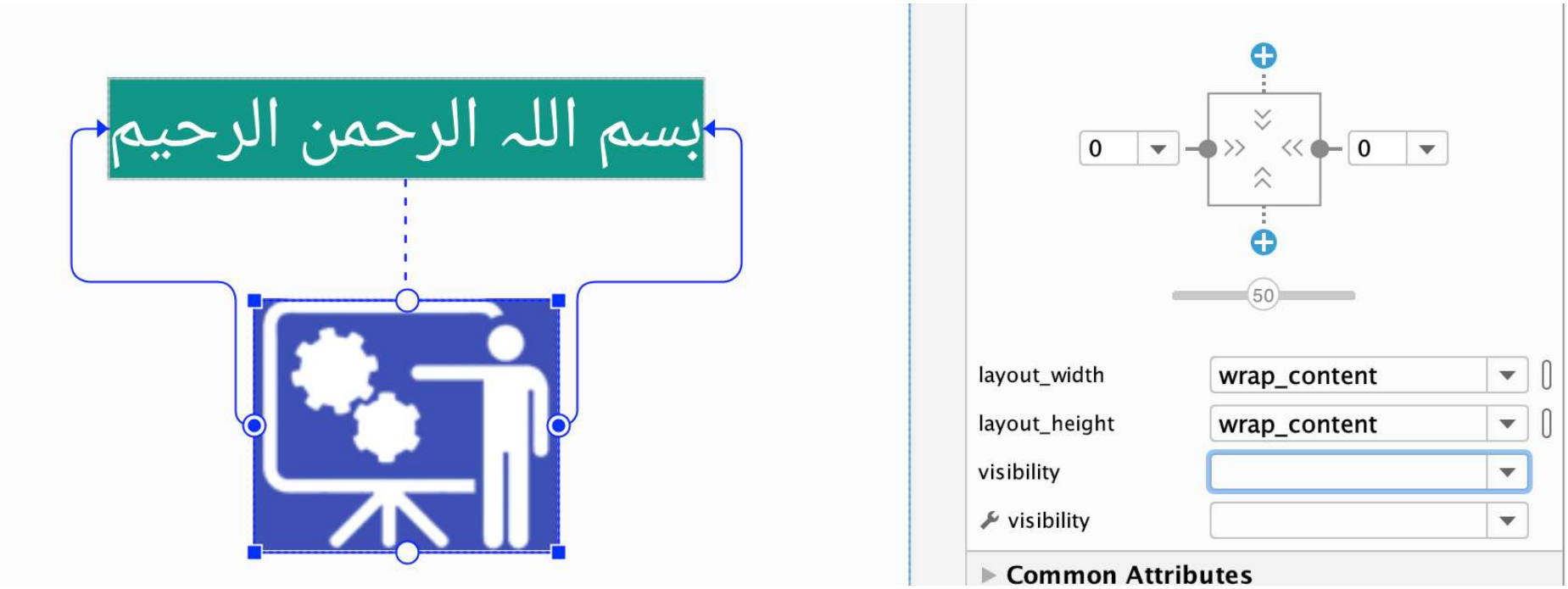
Layout Constraint



Layout Wrap Content, Match Constraint

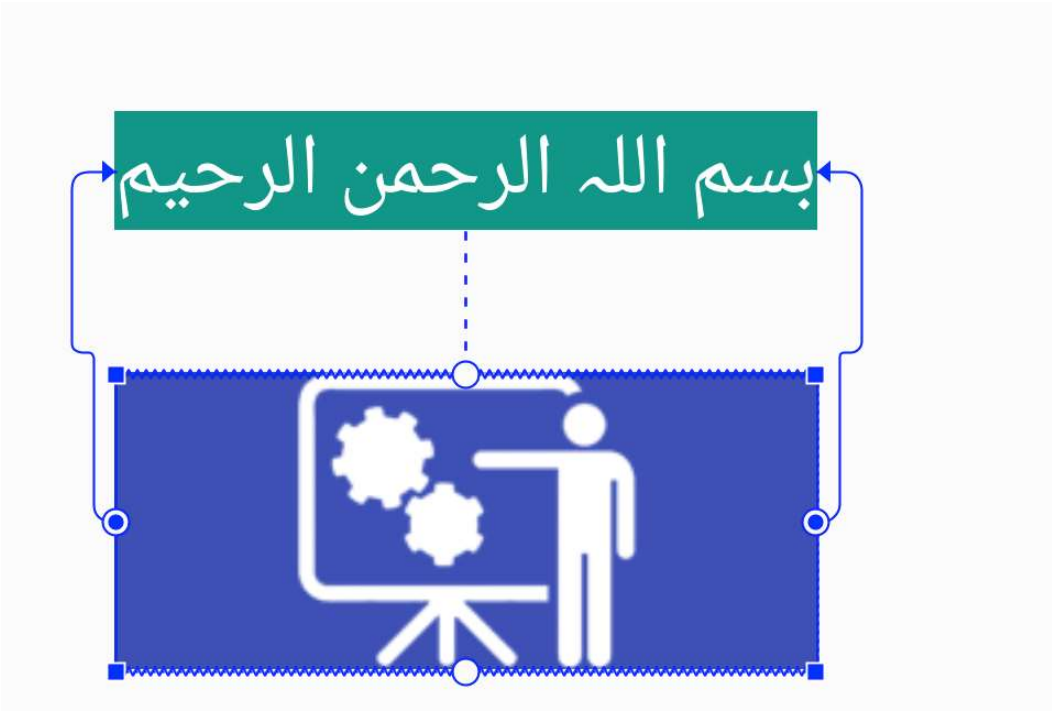


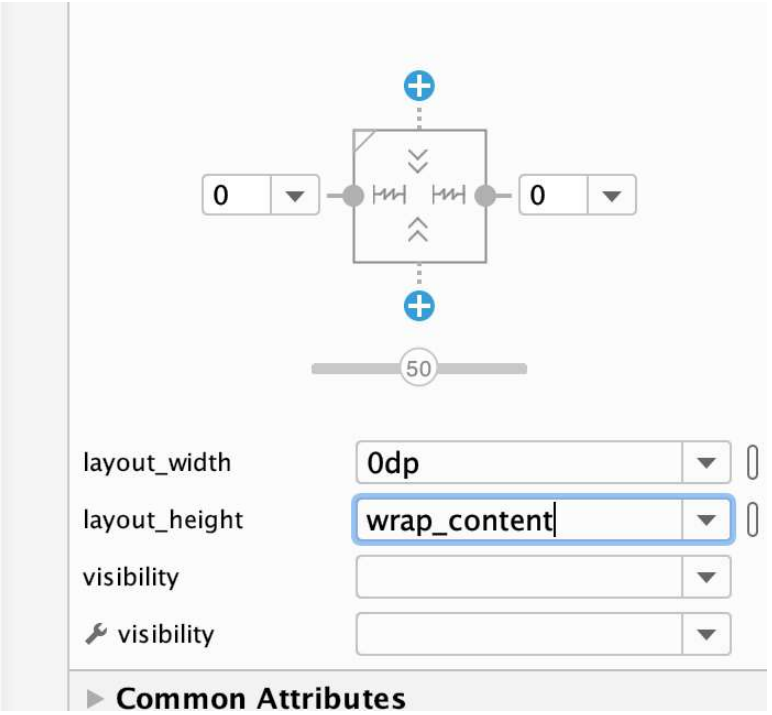
Layout Constraint Aspect Ratio



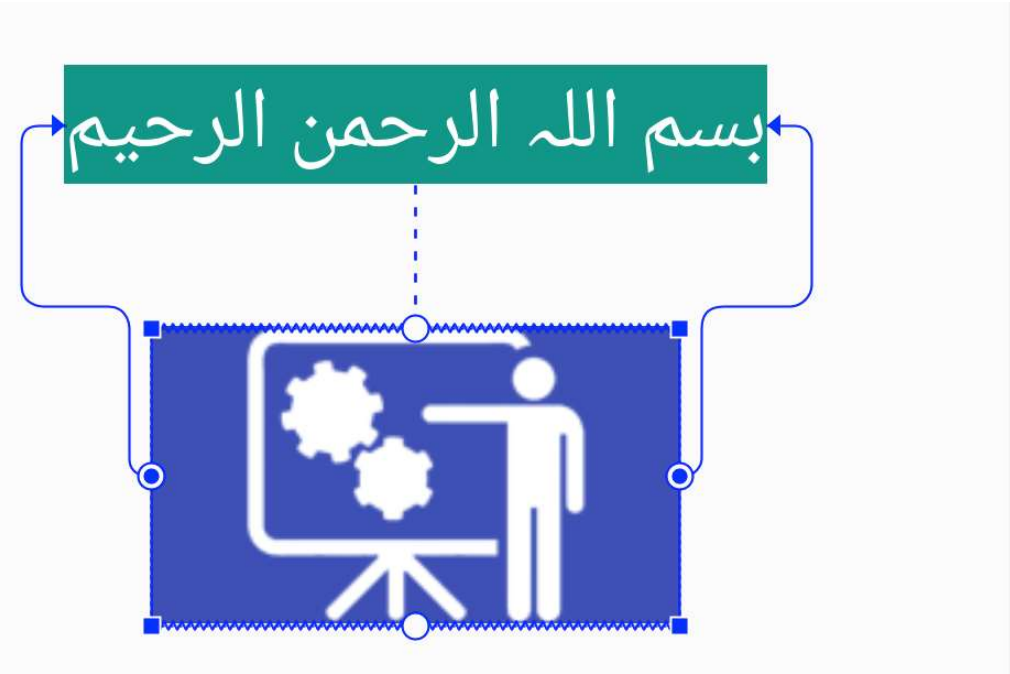
? If layout width = 0

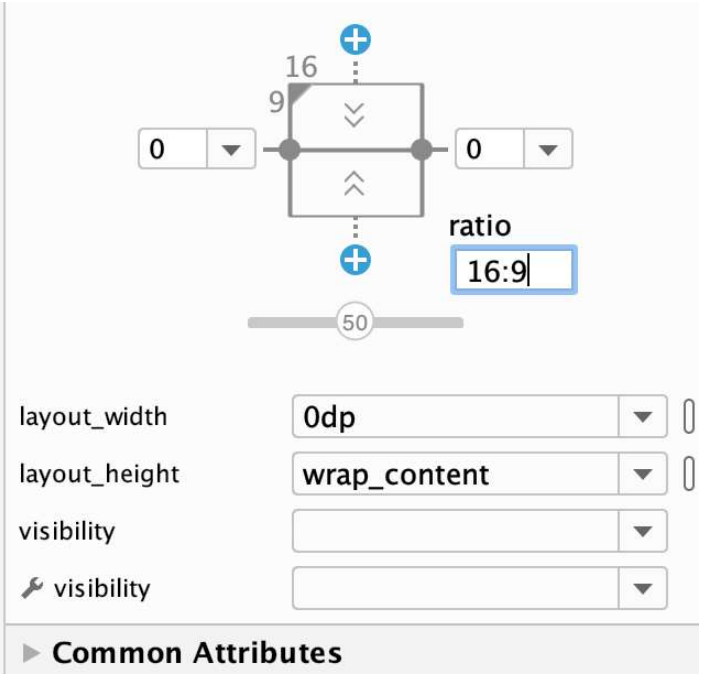
Layout Constraint Aspect Ratio



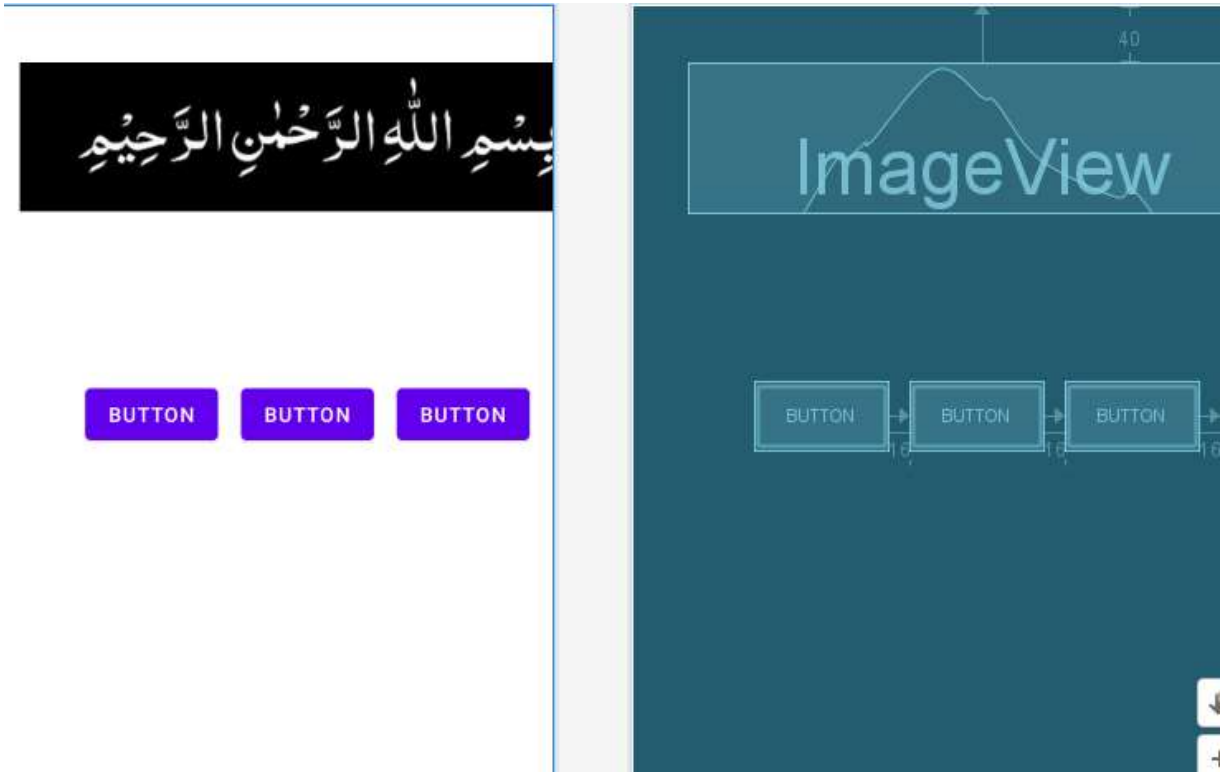


Layout Aspect Ratio 16:9

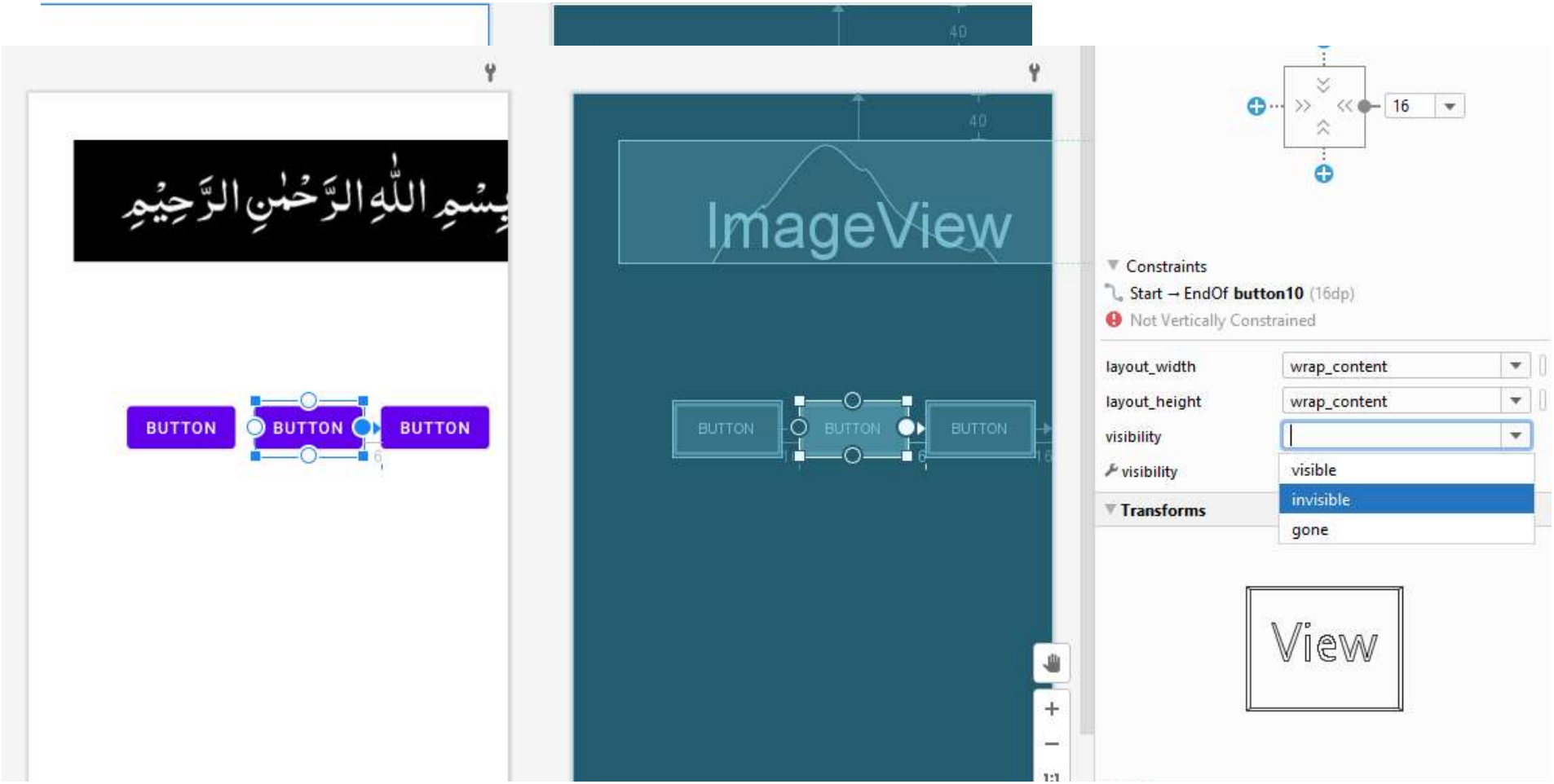




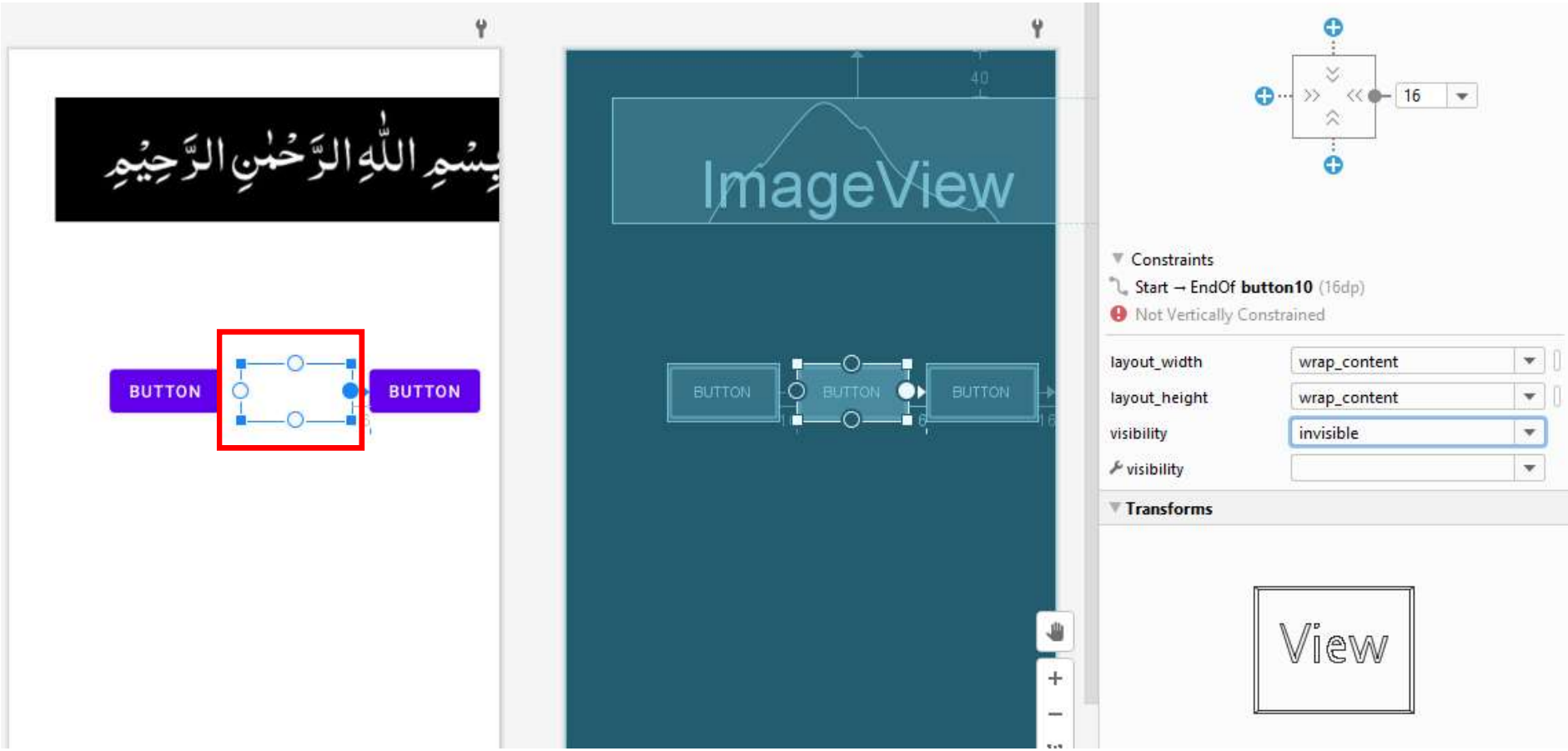
Layout Constraint visible



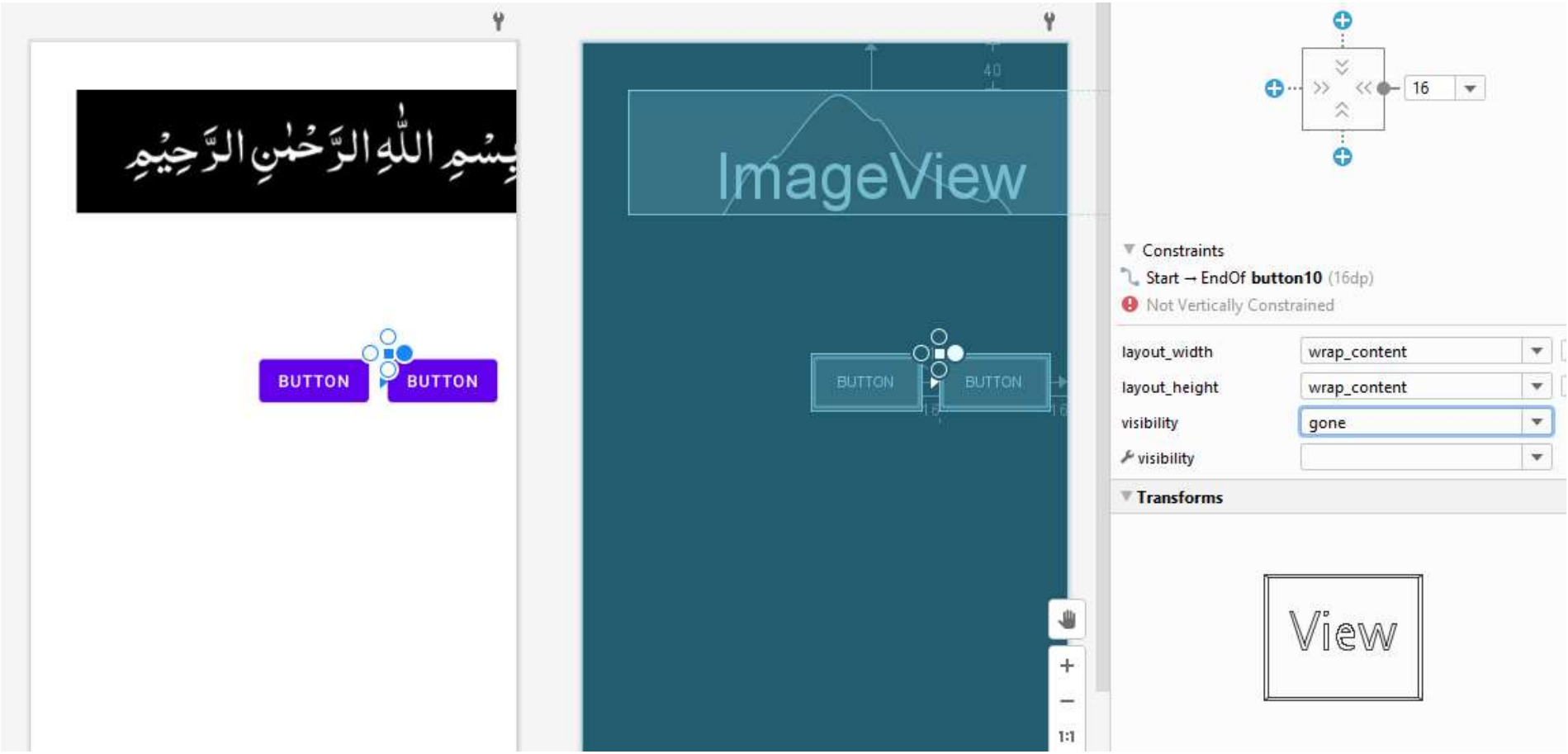
Layout Constraint invisible



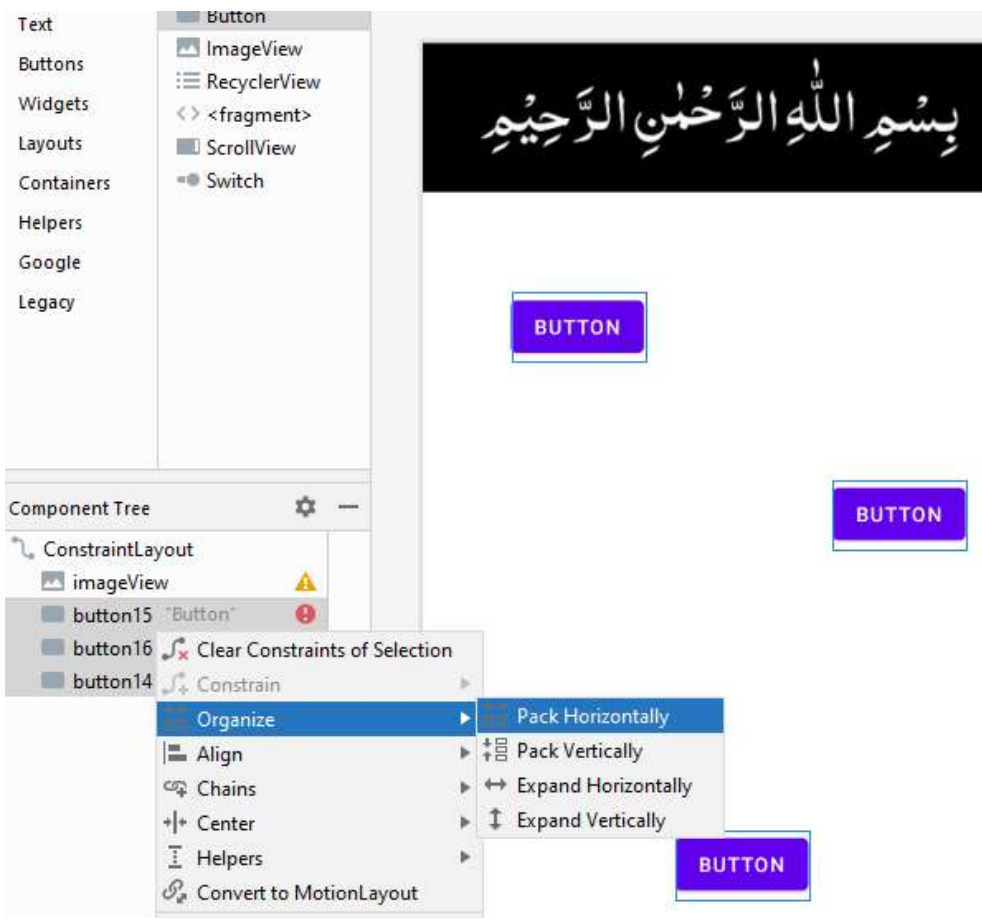
Layout Constraint invisible

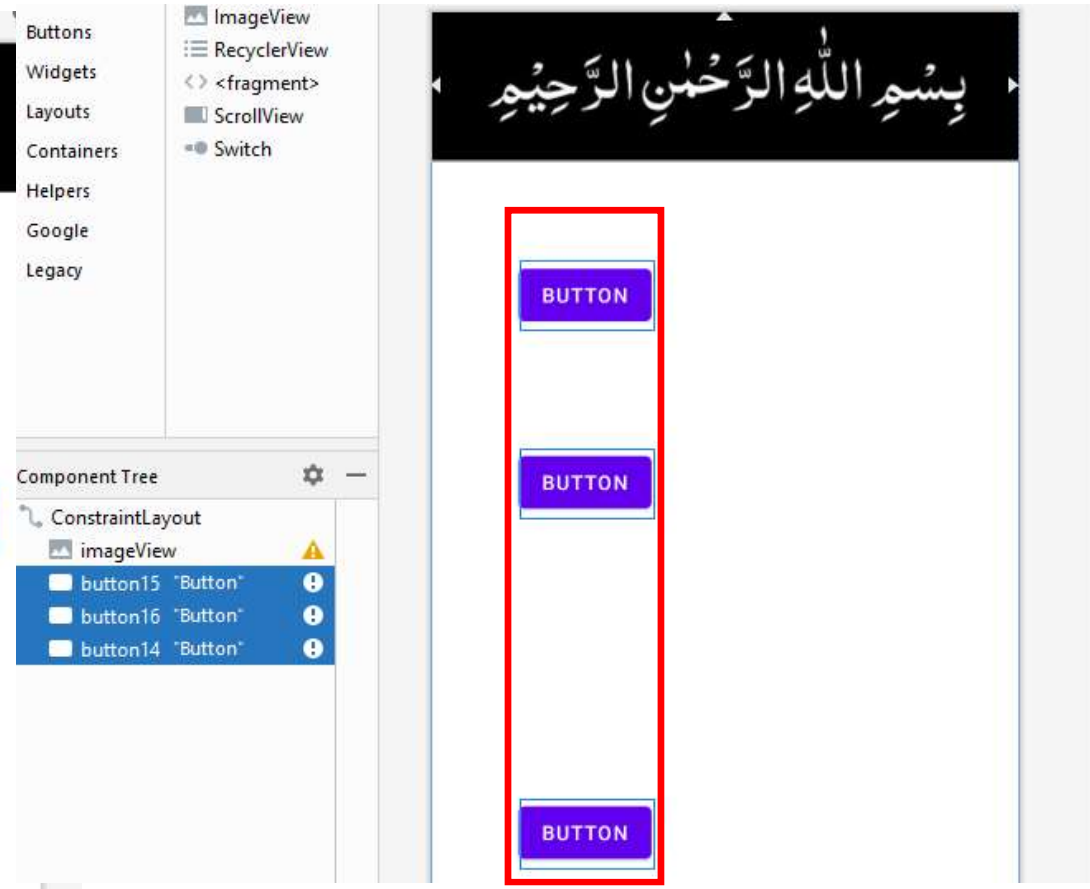


Layout Constraint gone

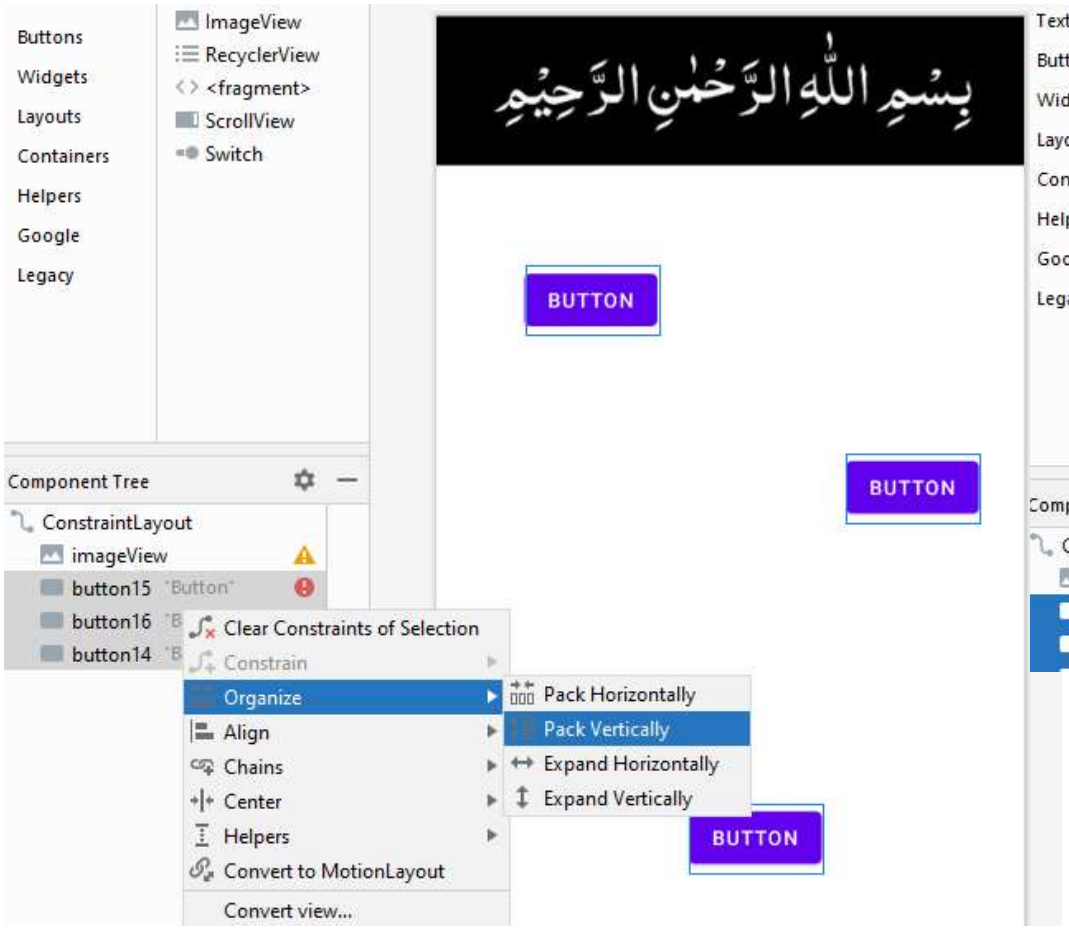


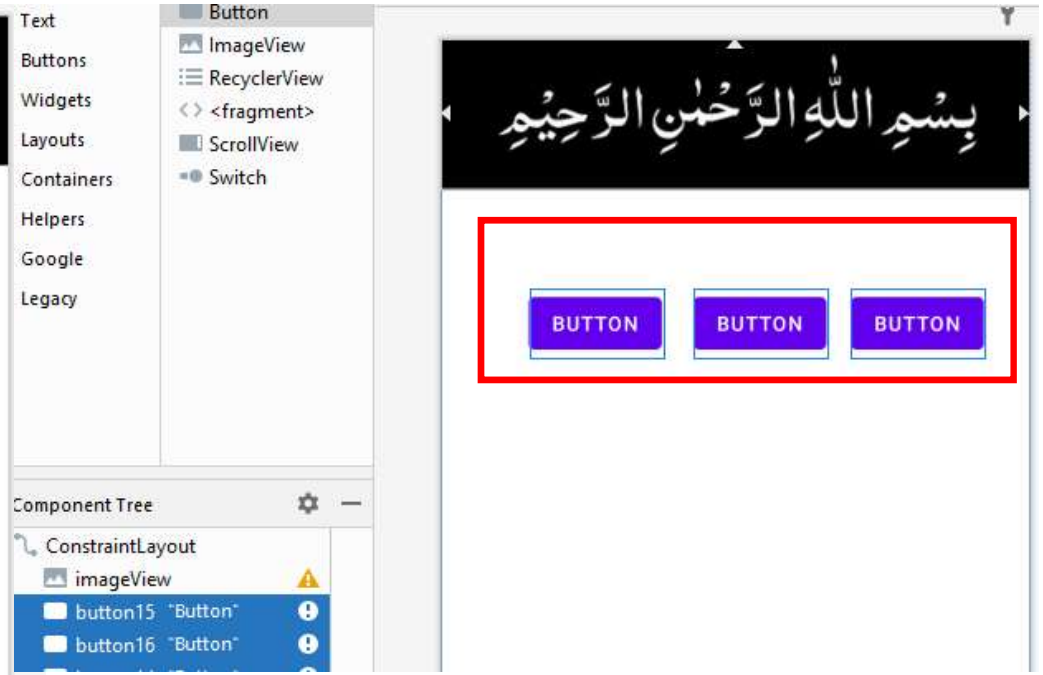
organize Pack Horizontally



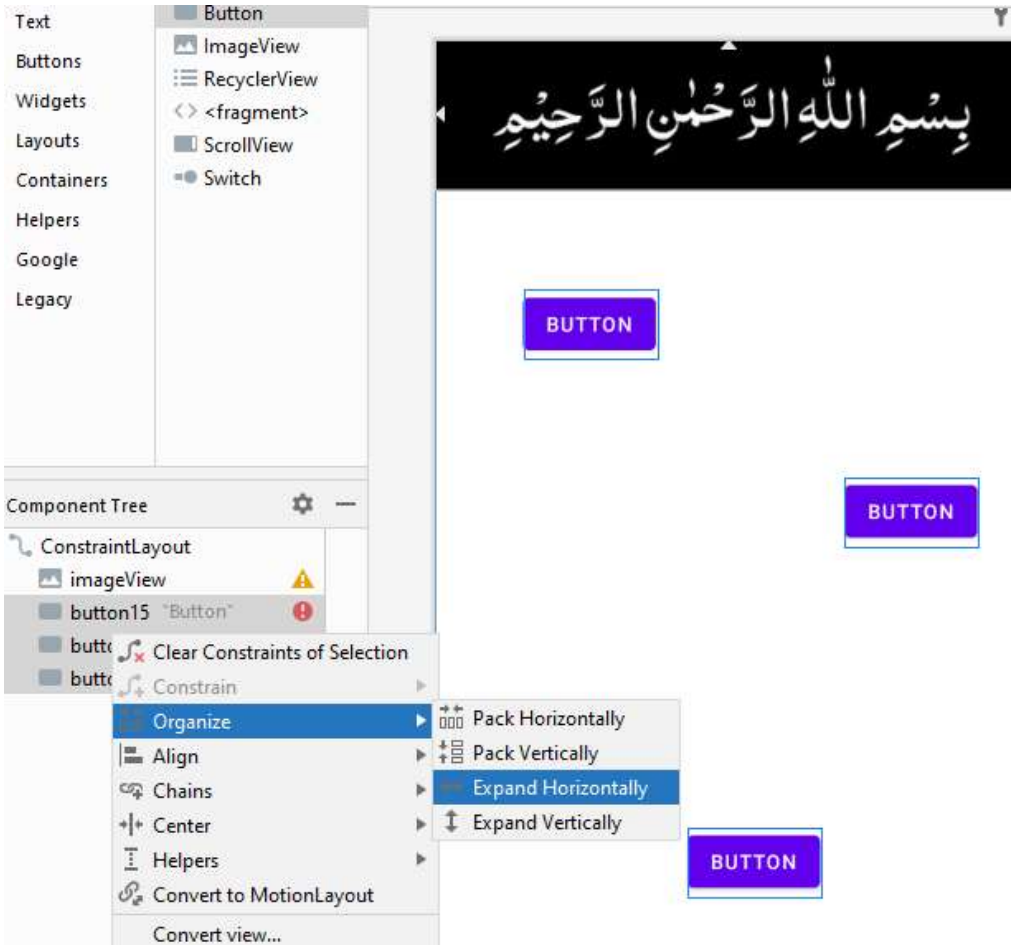


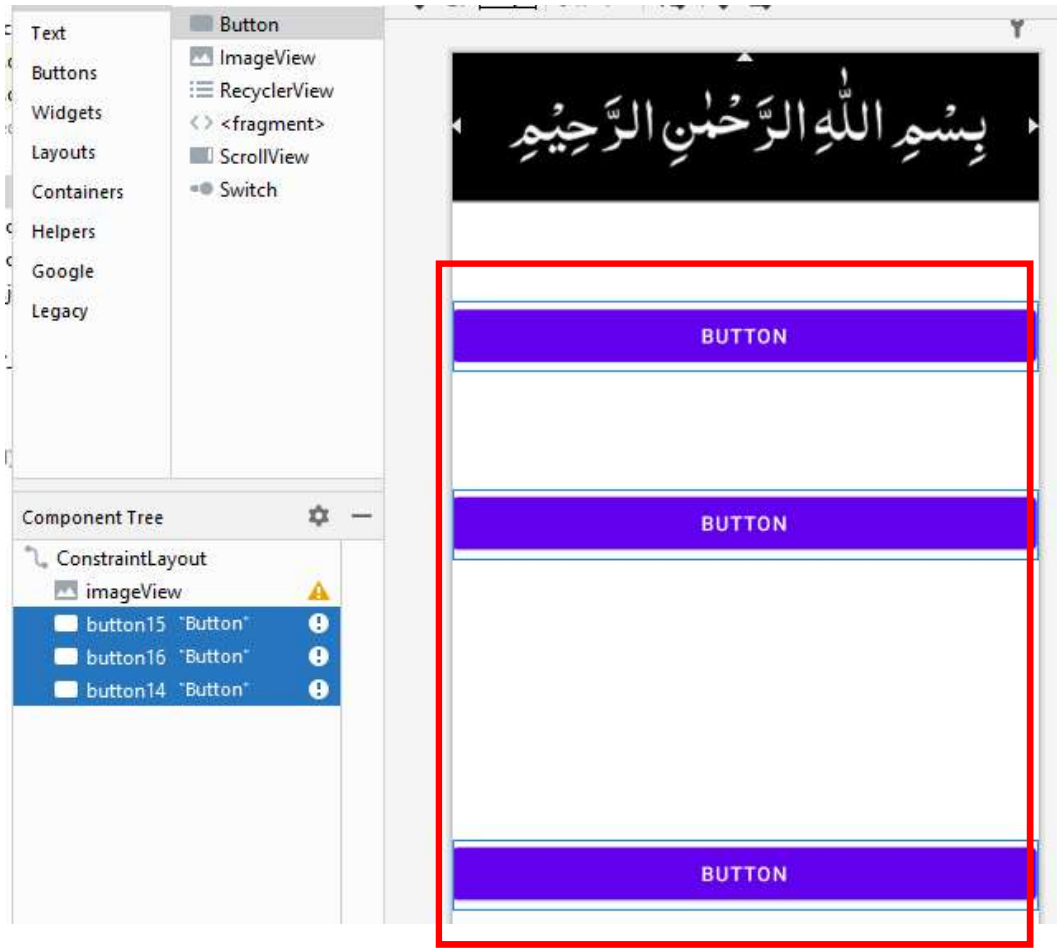
organize Pack Vertically



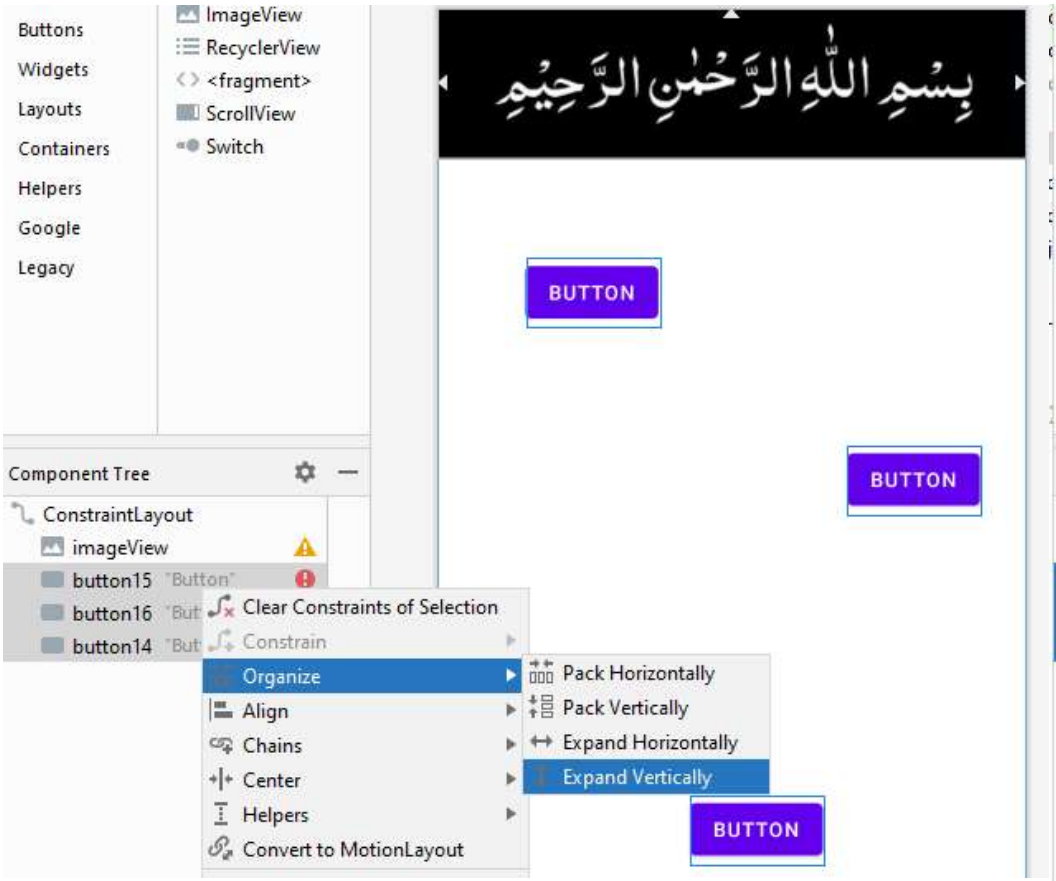


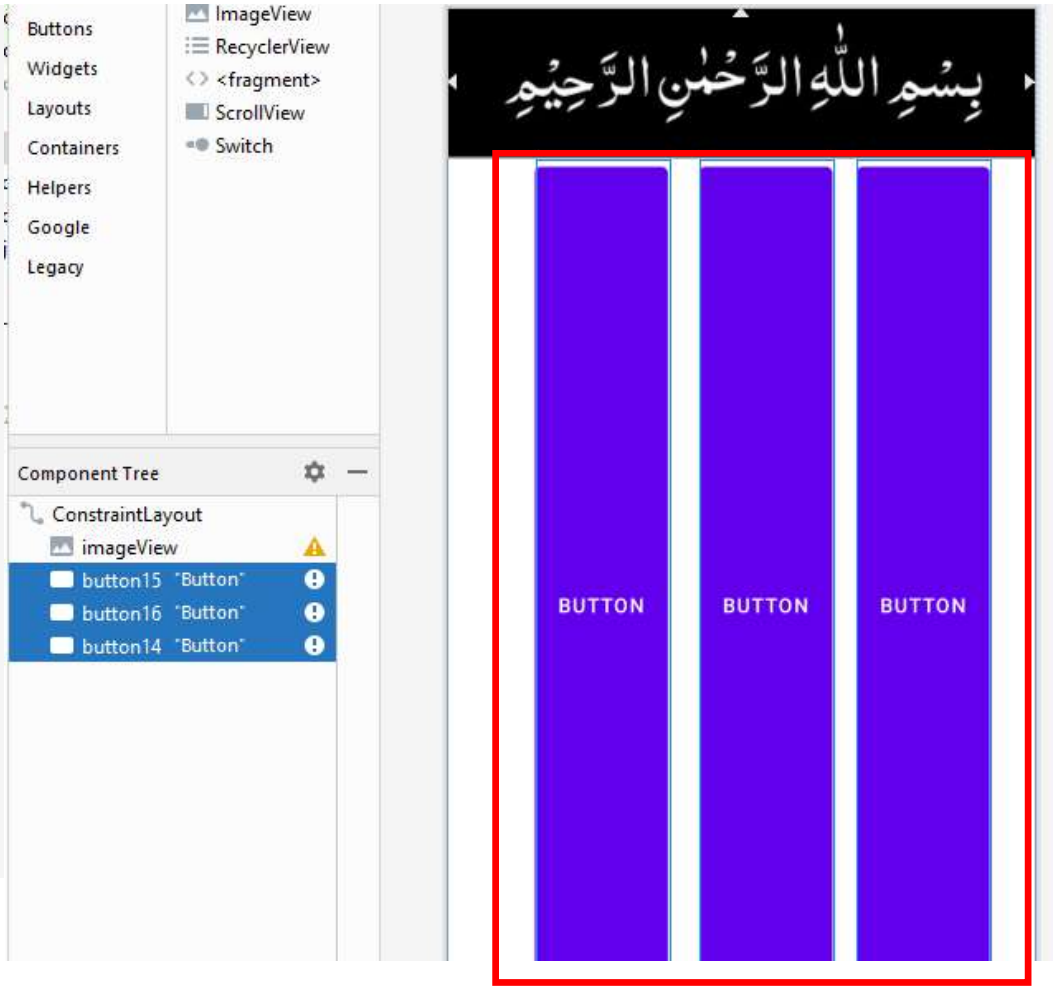
organize Expand Horizontally



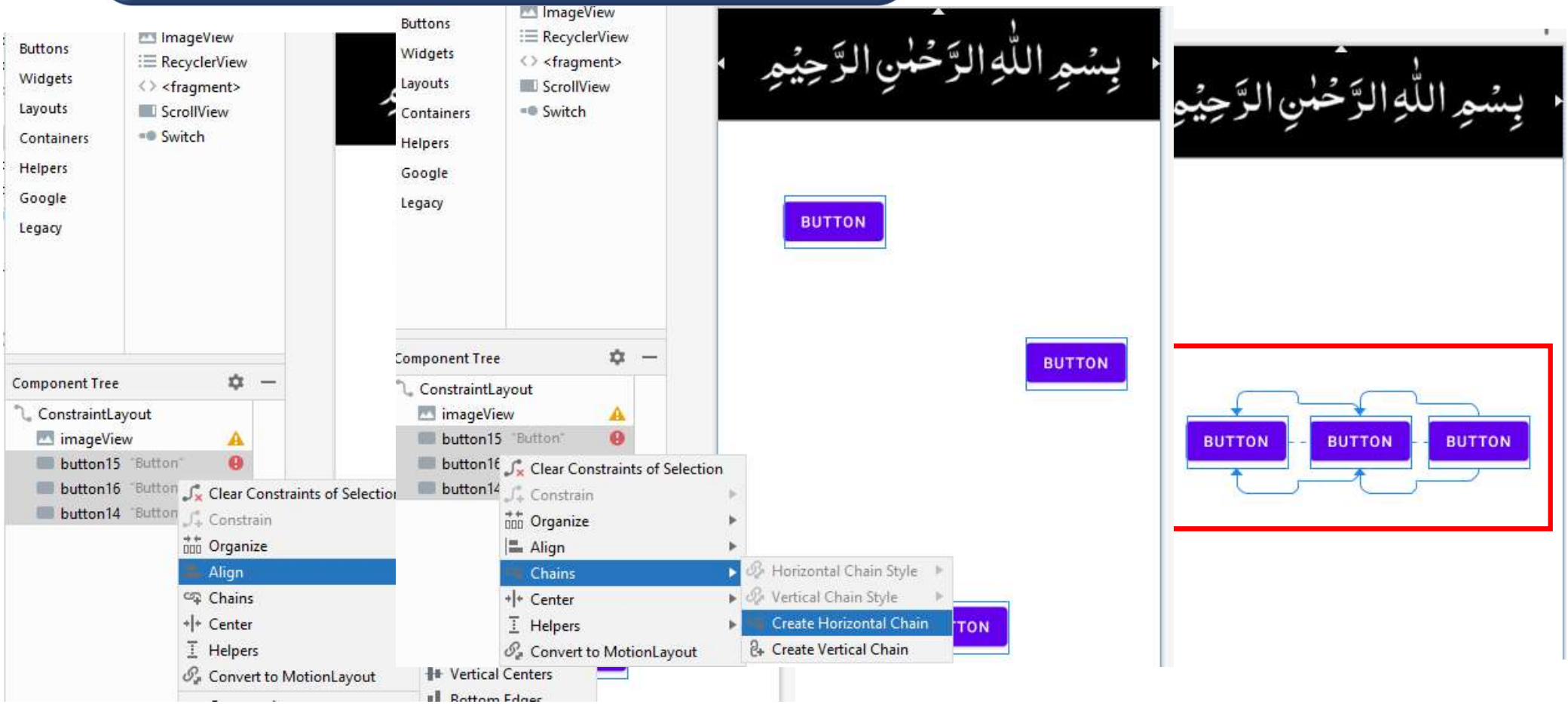


organize Expand Vertically

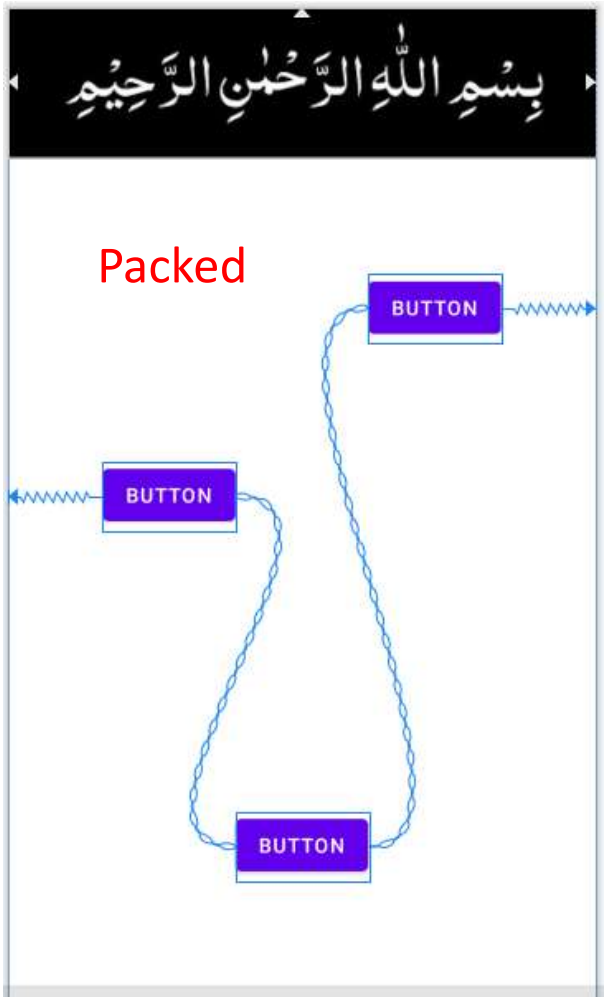


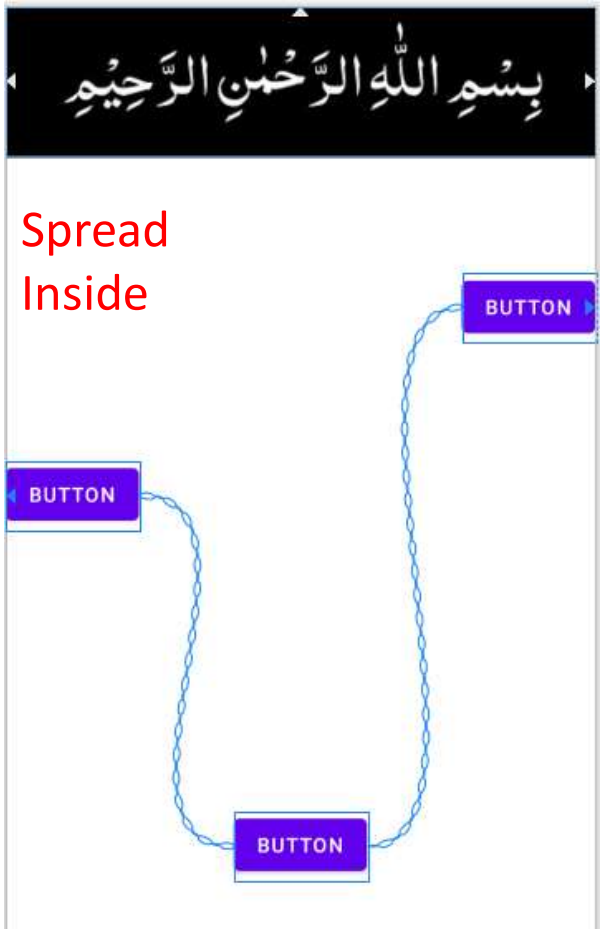


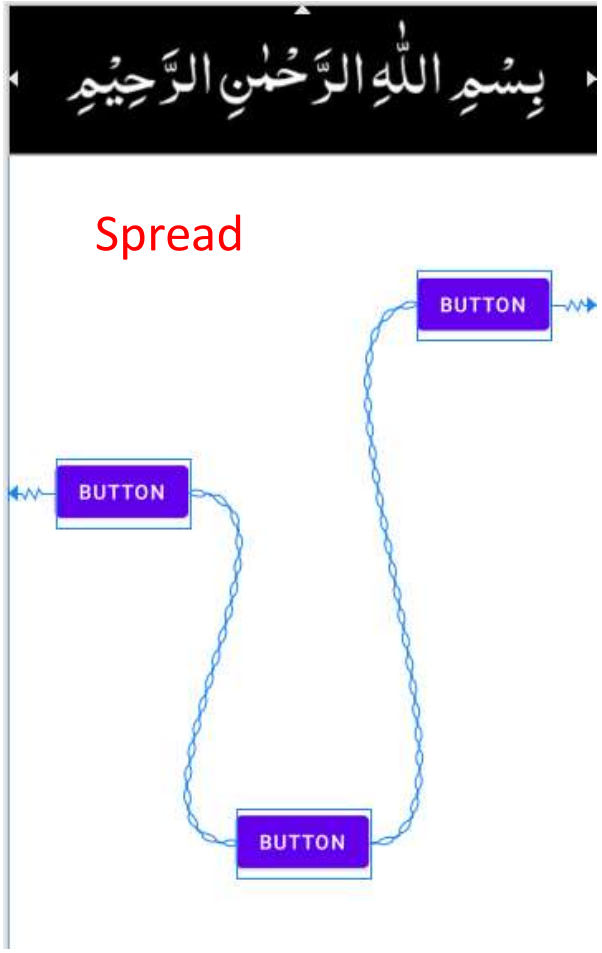
organize Vertical Centers



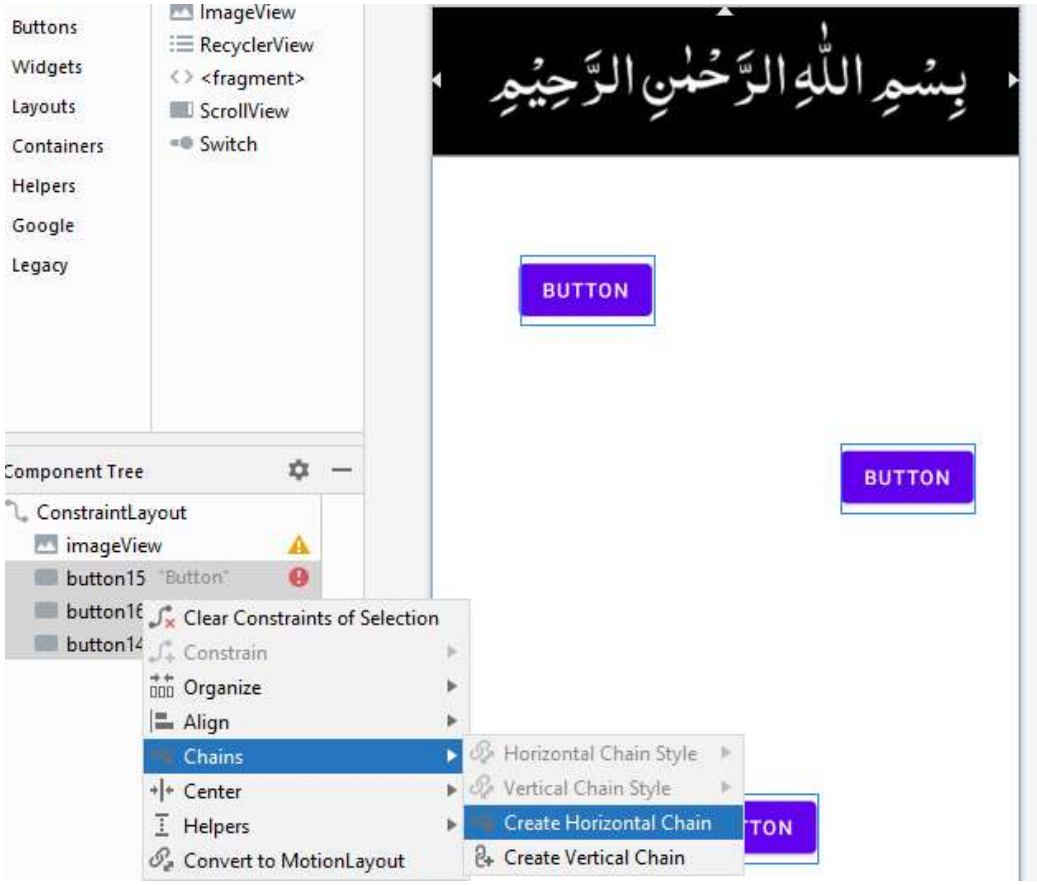
Chains Horizontal

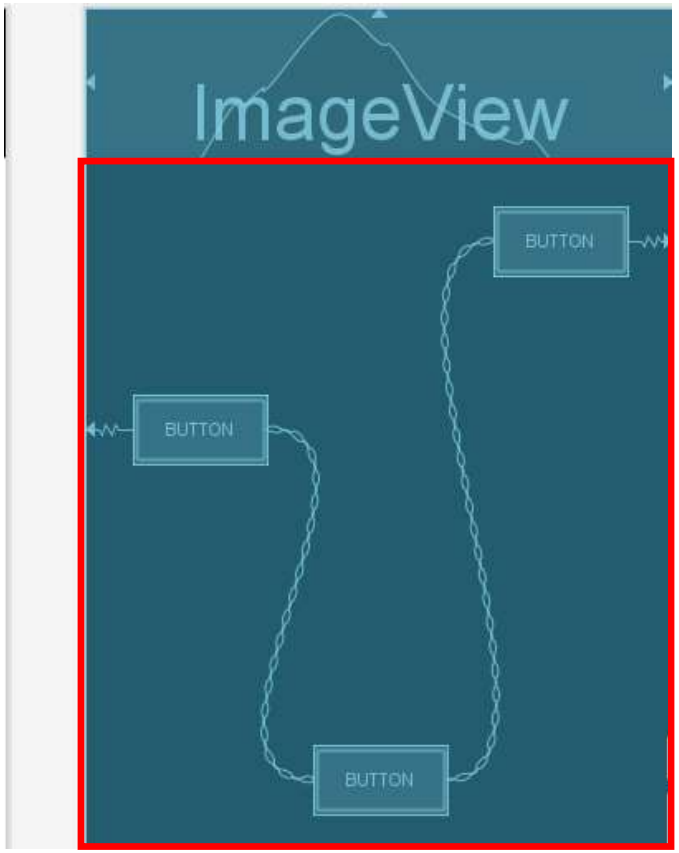




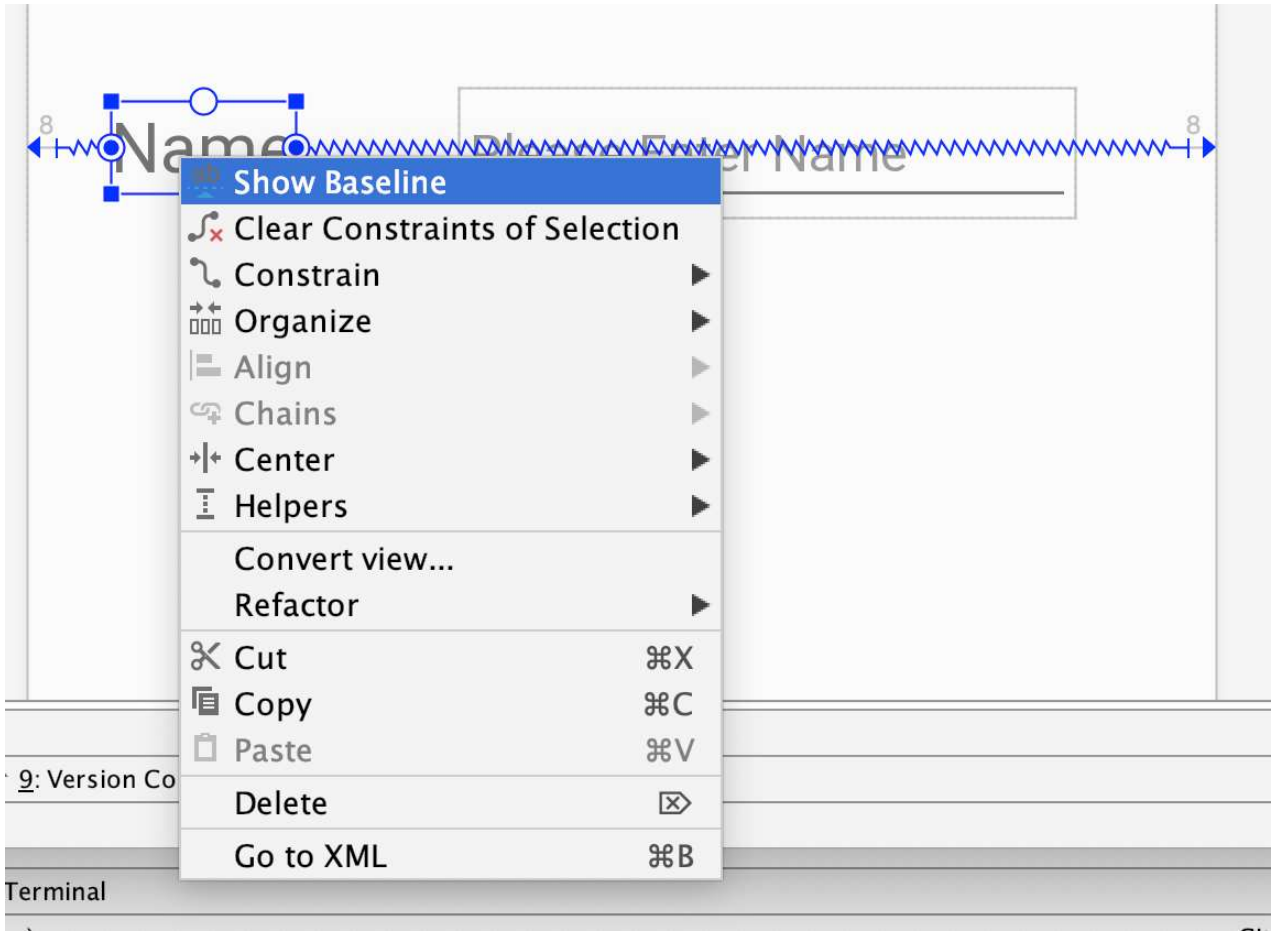


Chains Horizontal Chain

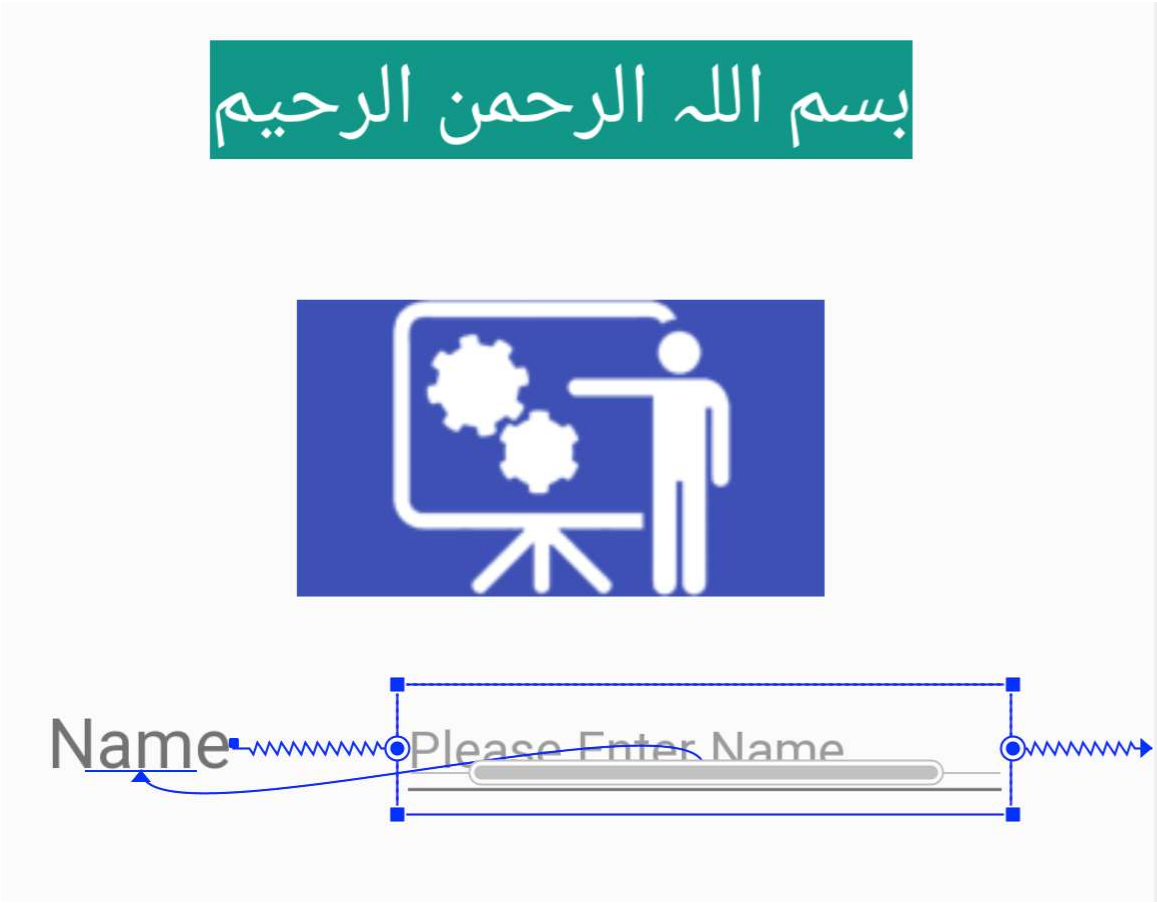


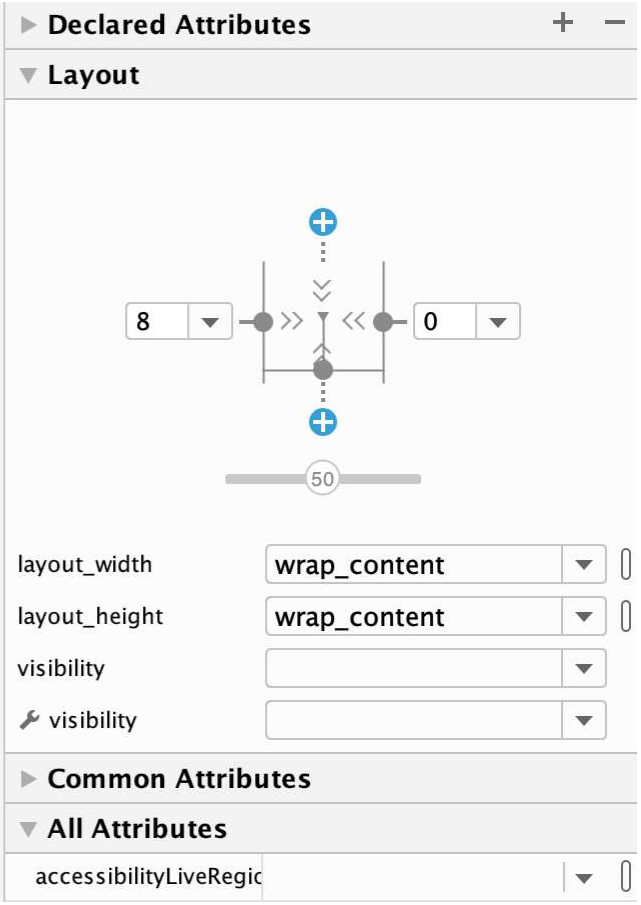


Layout Constraint Baseline

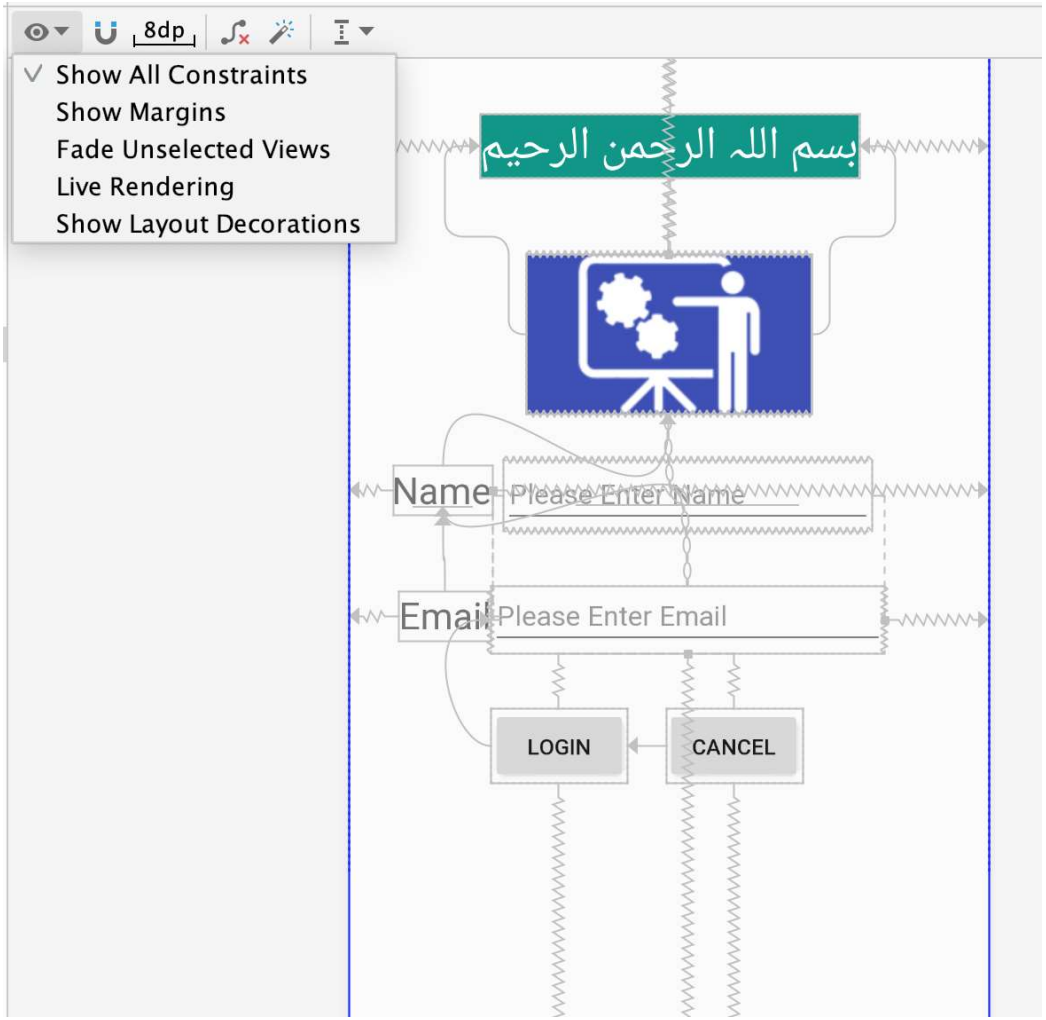


Layout Constraint Baseline

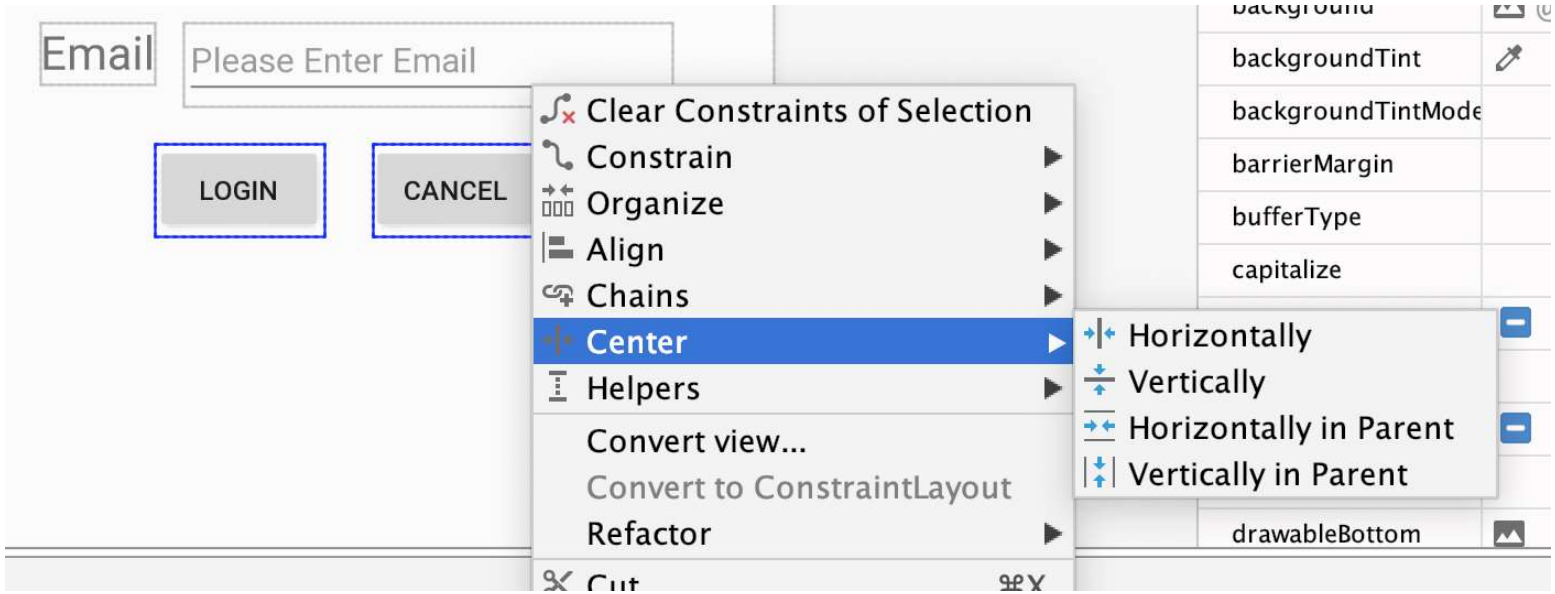




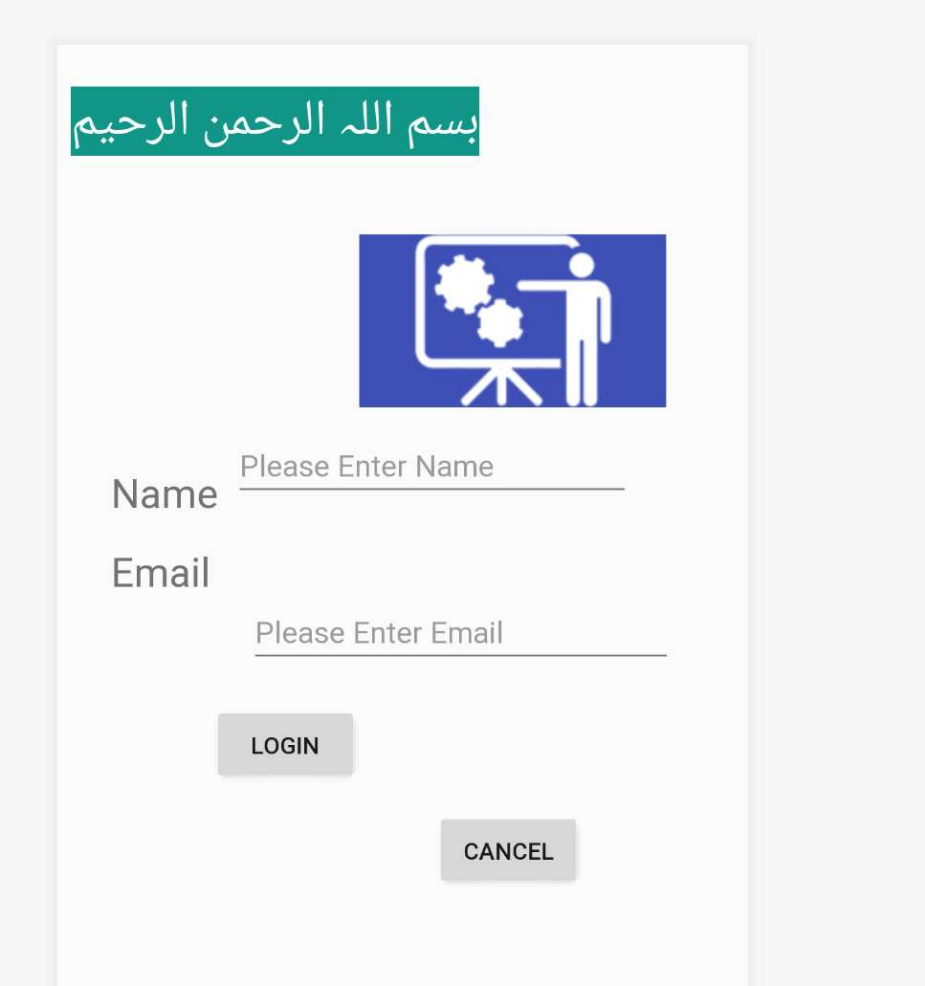
Layout Constraint Baseline



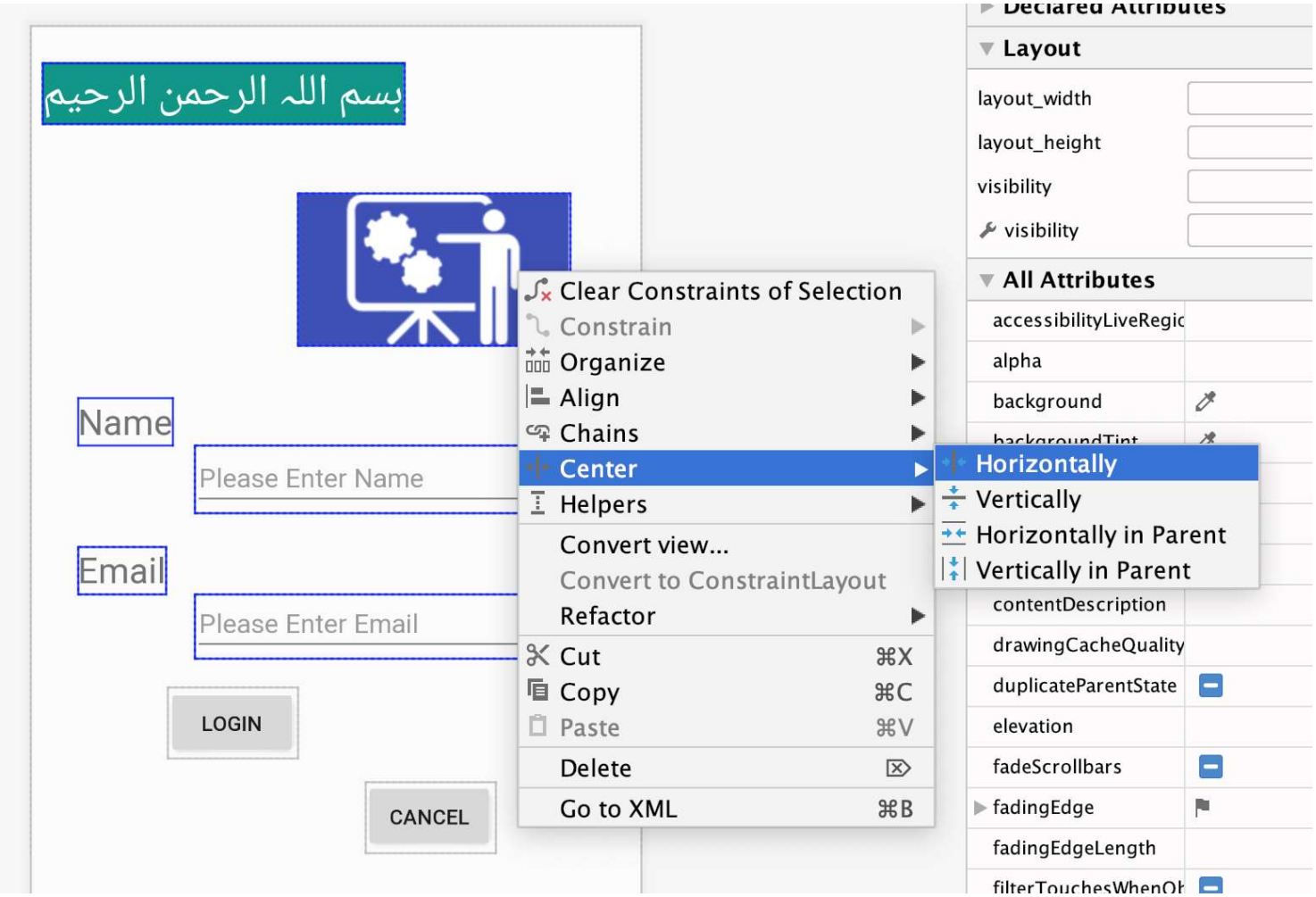
Layout Constraint Centre



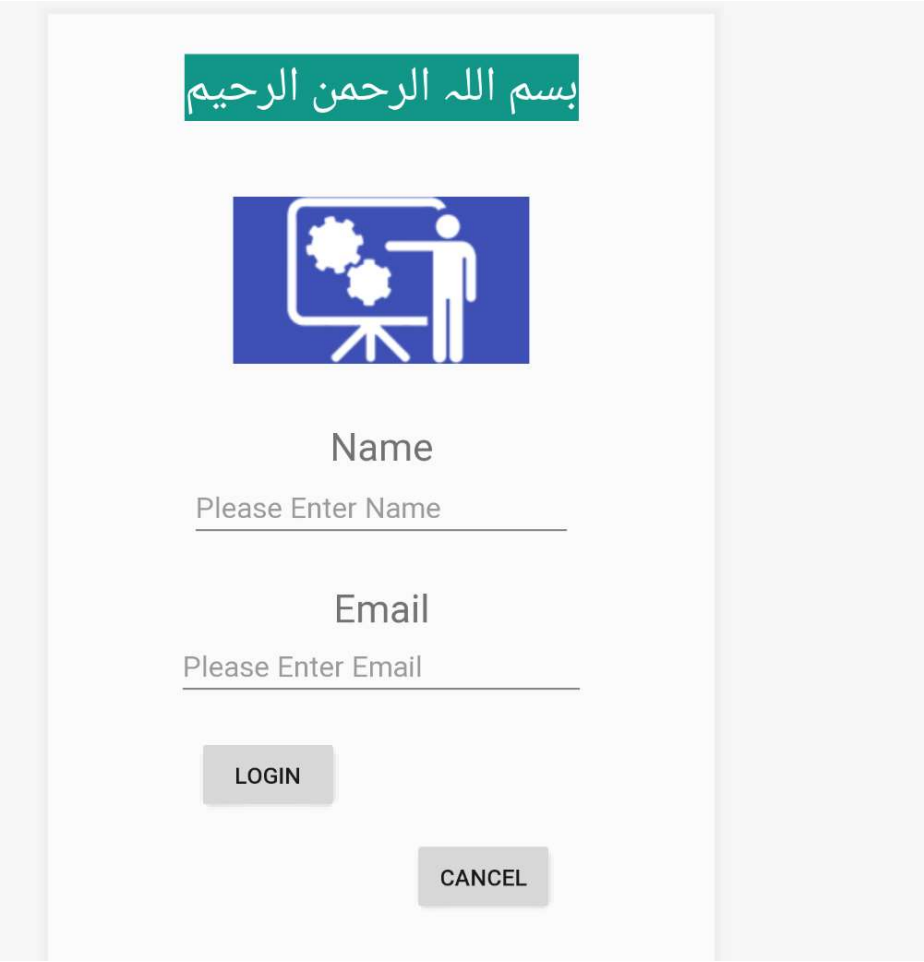
Layout Constraint

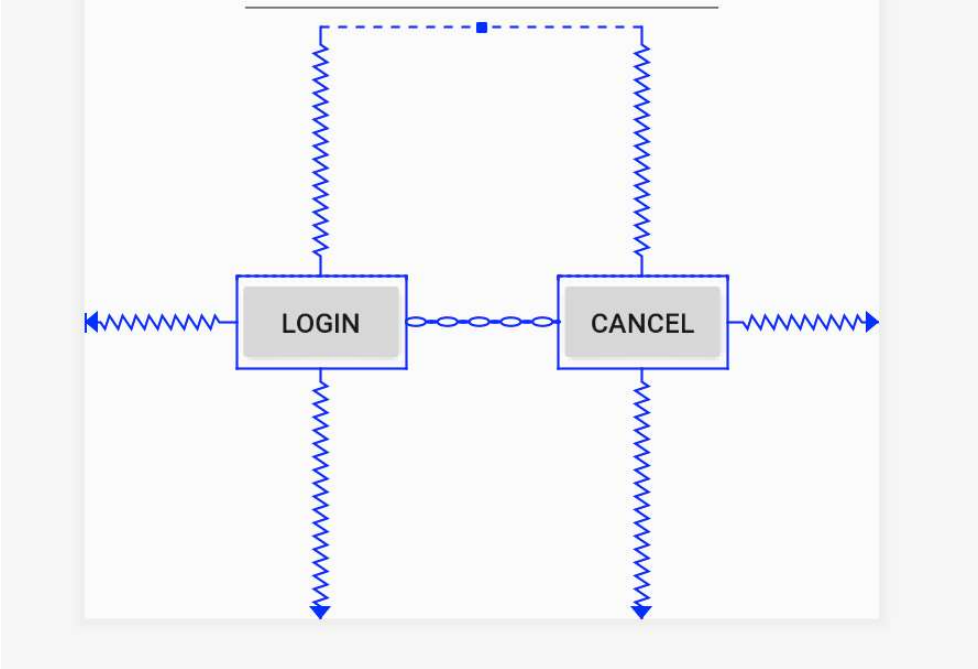


Layout Constraint Center

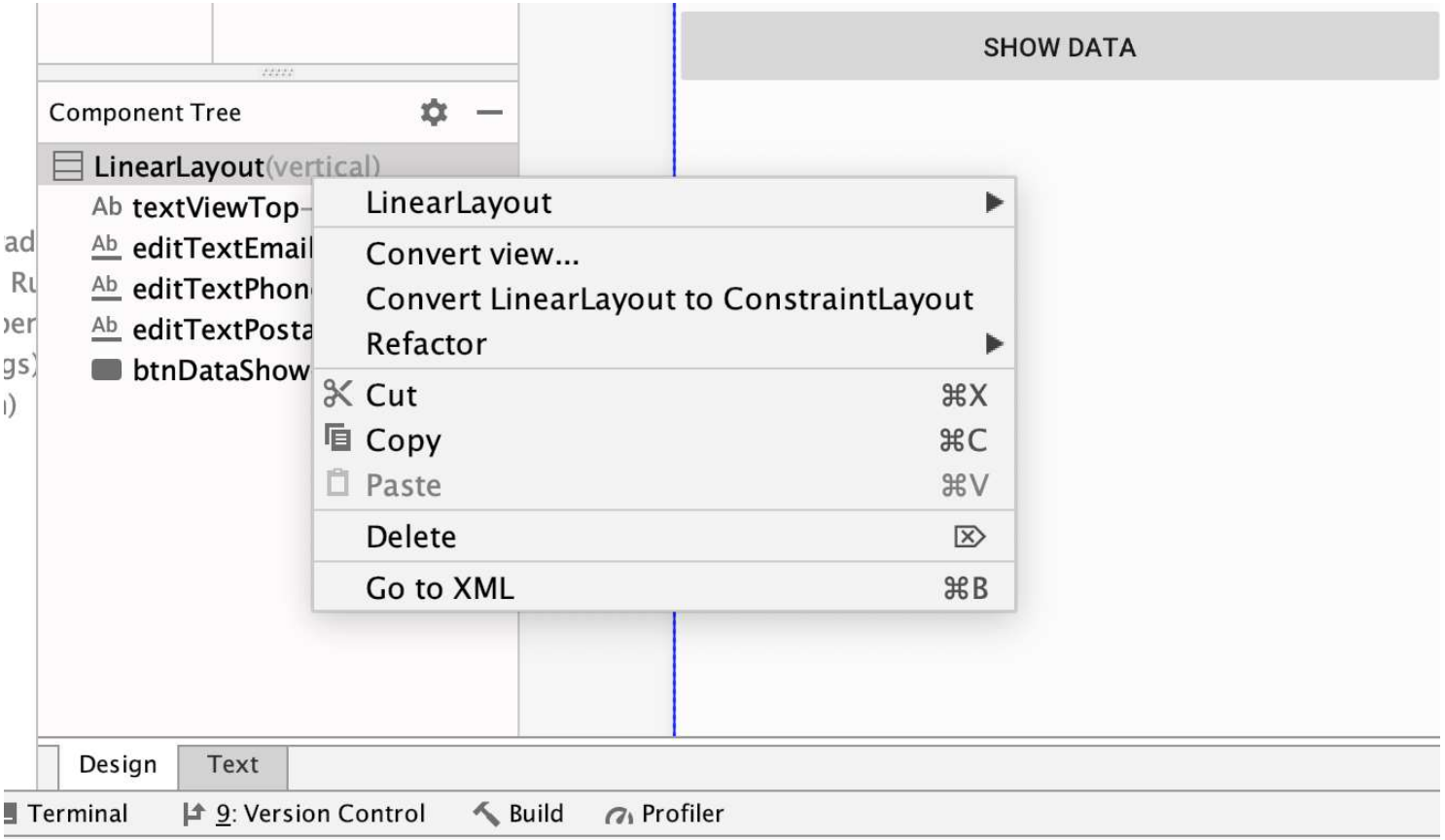


Layout Constraint Center

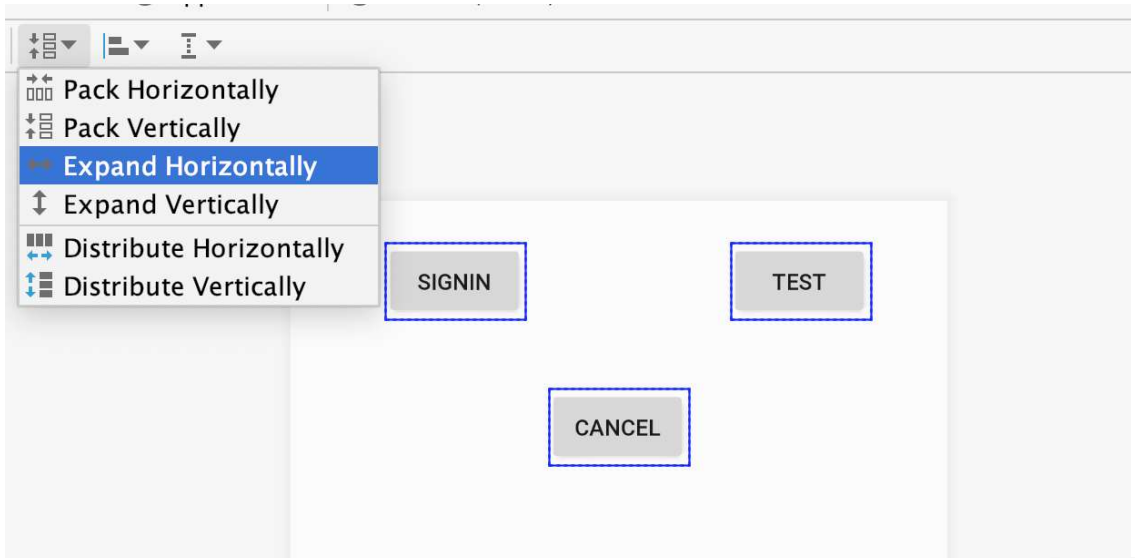


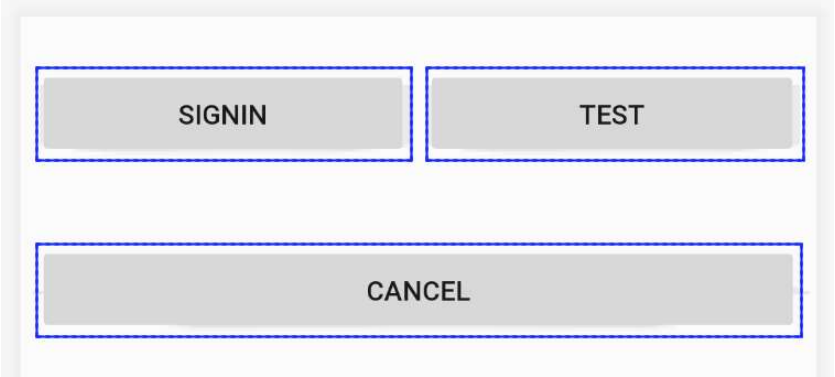


Layout Conversion

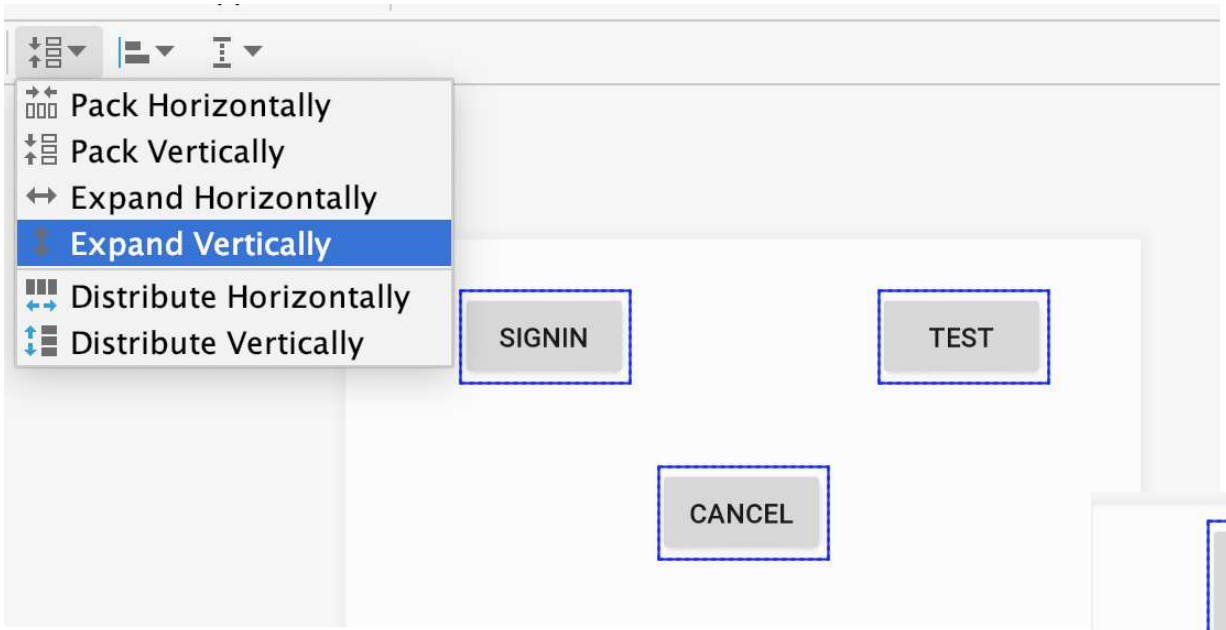


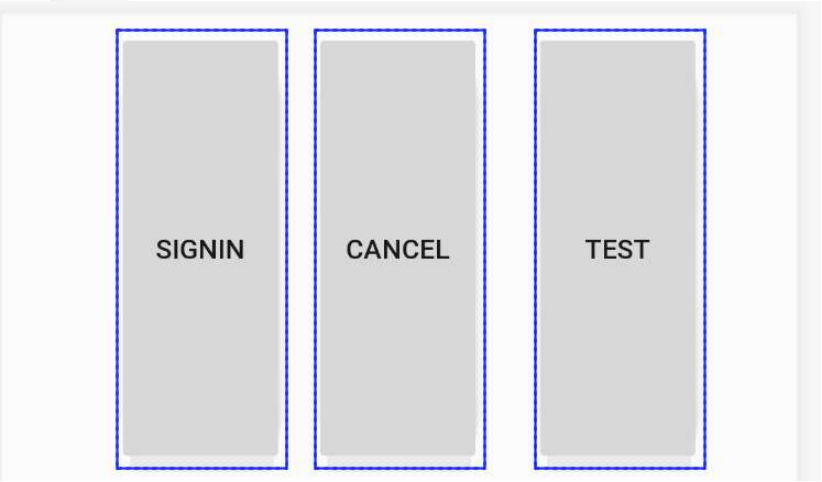
Layout constraint Expand Horizontally





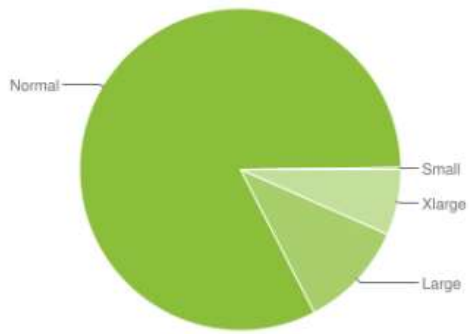
Layout constraint Expand Vertically

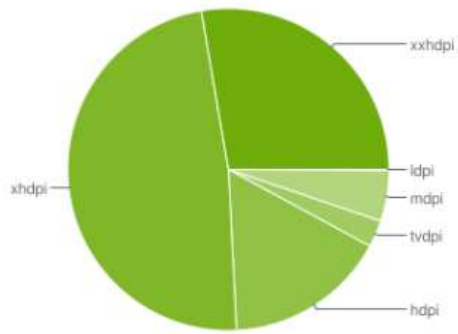




Screen Size And Density

	ldpi	mdpi	tvdpi	hdpi	xhdpi	xxhdpi	Total
Small	0.1%				0.1%		0.2%
Normal		0.2%	0.3%	12.9%	43.6%	25.4%	82.4%
Large		1.4%	2.4%	0.7%	4.0%	2.3%	10.8%
Xlarge		3.5%		2.7%	0.4%		6.6%
Total	0.1%	5.1%	2.7%	16.3%	48.1%	27.7%	

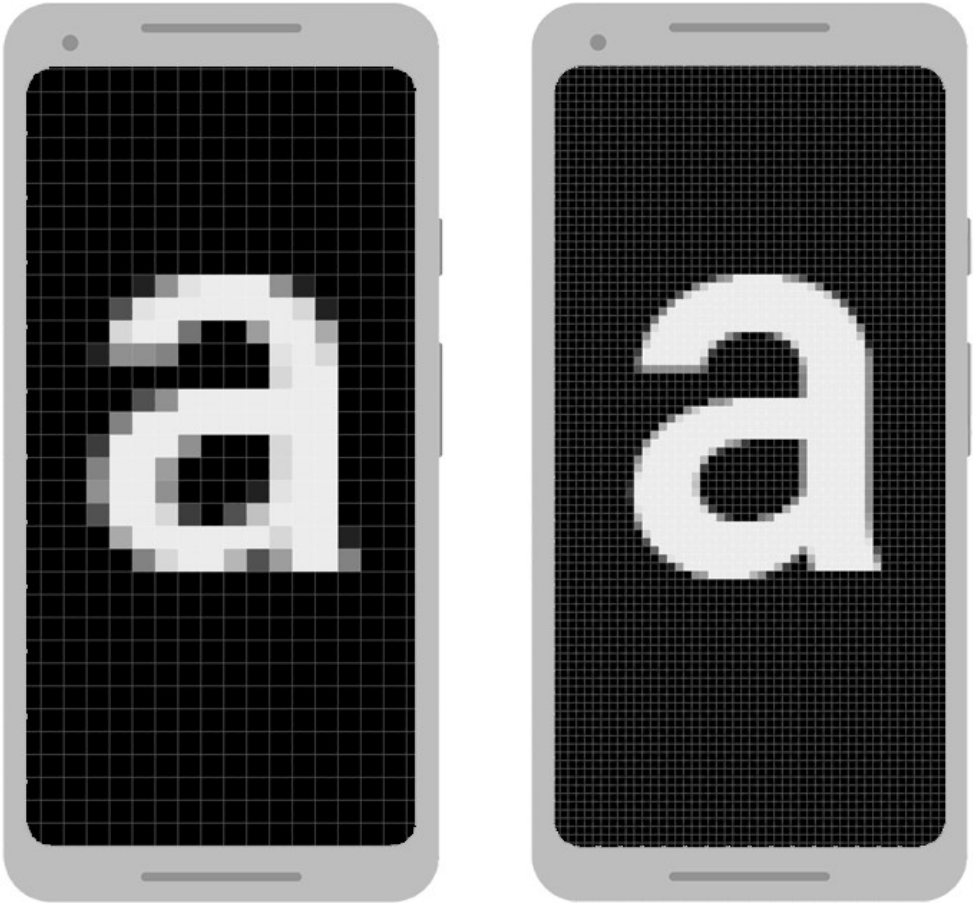




Data collected during a 7-day period ending on July 31, 2021.
Any screen configurations with less than 0.1% distribution are not shown.

https://developer.android.com/about/dashboards/index.html

Same Size different Pixels

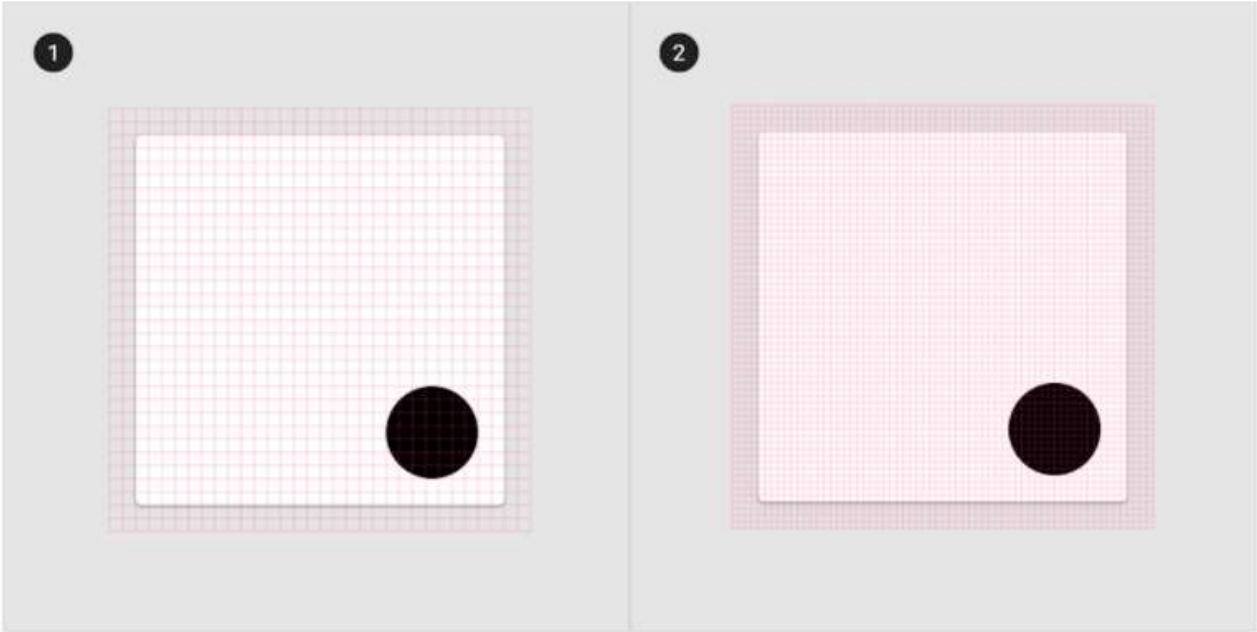


https://developer.android.com/training/multiscreen/screendensities

Density-independent pixels

Density-independent pixels, written as dp (pronounced "dips"), are flexible units that scale to have uniform dimensions on any screen. They provide a flexible way to accommodate a design across platforms.

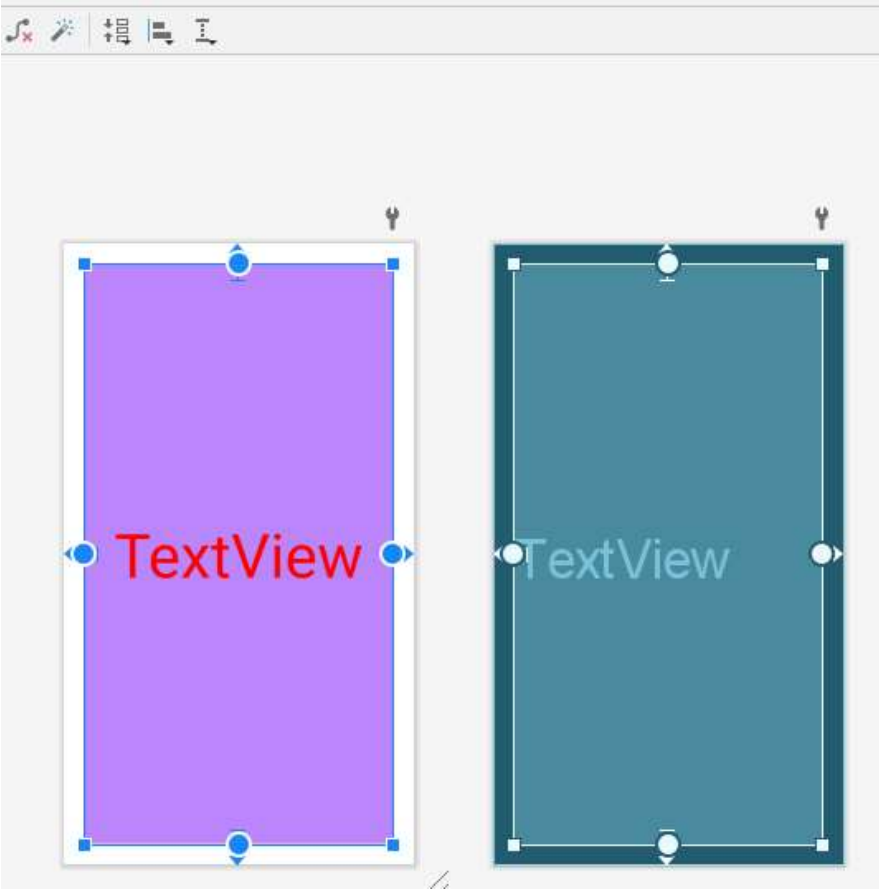
Material UIs use density-independent pixels to display elements consistently on screens with different densities.



1. Low-density screen displayed with density independence
2. High-density screen displayed with density independence

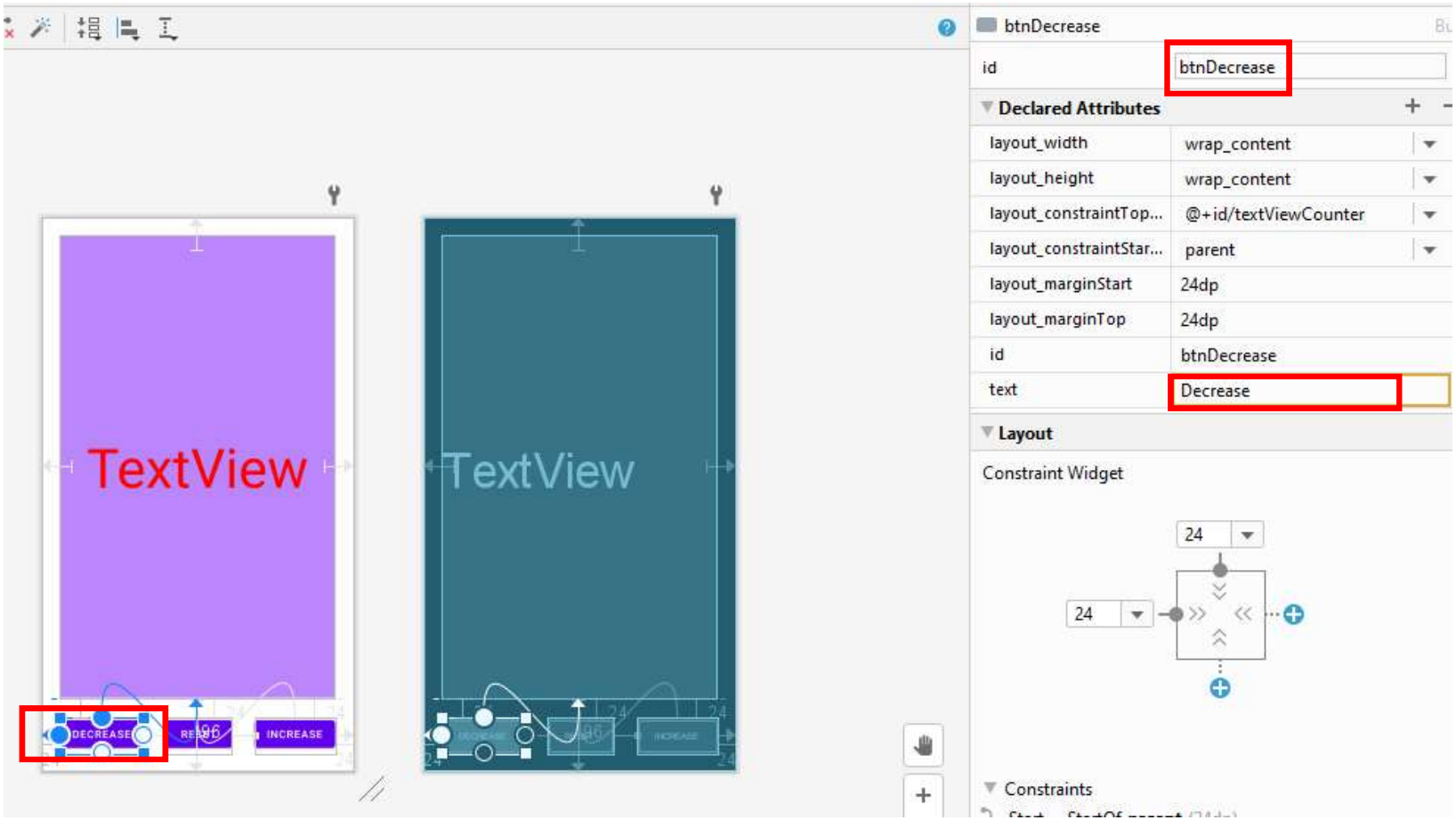
https://material.io/design/layout/pixel-density.html#pixel-density-on-android

XML Design




```xml
<TextView
    android:id="@+id/textViewCounter"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_marginStart="24dp"
    android:layout_marginTop="24dp"
    android:layout_marginEnd="24dp"
    android:layout_marginBottom="96dp"
    android:background="@color/purple_200"
    android:gravity="center_horizontal|center_vertical"
    android:text="TextView"
    android:textAlignment="gravity"
    android:textColor="#FF0000"
    android:textSize="72sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0" />
```

Load XML Resources



Load XML Resources

```
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.Button;

public class ActivityOne extends AppCompatActivity {

    Button buttonIncrease, buttonDecrease, buttonReset;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_one);
```

Coding

```
Button buttonIncrease, buttonDecrease, buttonReset;
TextView textView;
int counter= 0;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_one);

    buttonIncrease = findViewById(R.id.btnIncrease);
    buttonDecrease = findViewById(R.id.btnDecrease);
    buttonReset = findViewById(R.id.btnReset);
    textView = findViewById(R.id.textViewCounter);

    buttonIncrease.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            counter++;

        }
    });
}
```

Coding

```
    buttonIncrease = findViewById(R.id.btnIncrease);
    buttonDecrease = findViewById(R.id.btnDecrease);
    buttonReset = findViewById(R.id.btnReset);
    textView = findViewById(R.id.textViewCounter);

    buttonIncrease.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            counter++;
            textView.setText(Integer.toString(counter));
        }
    });
}
```

Coding

```
<Button
    android:id="@+id/btnReset"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:layout_marginTop="24dp"
    android:layout_marginEnd="24dp"
    android:text="Reset"
    android:onClick="ResetNumber"
    app:layout_constraintEn
    app:layout_constraintSt
    app:layout_constraintTo
</androidx.constraintlayout.wid
```

- Provide feedback on this warning
- Provide feedback on this warning
- Suppress: Add tools:ignore="OnClick" attribute
- Create 'ResetNumber(View)' in 'ActivityOne'
- Suppress: Add tools:ignore="UsingOnClickInXml" attribute
- Create onClick event handler

Coding

```
    public void ResetNumber(View view) {
        counter=0;
        textView.setText(Integer.toString(counter));
    }
}
```

Coding

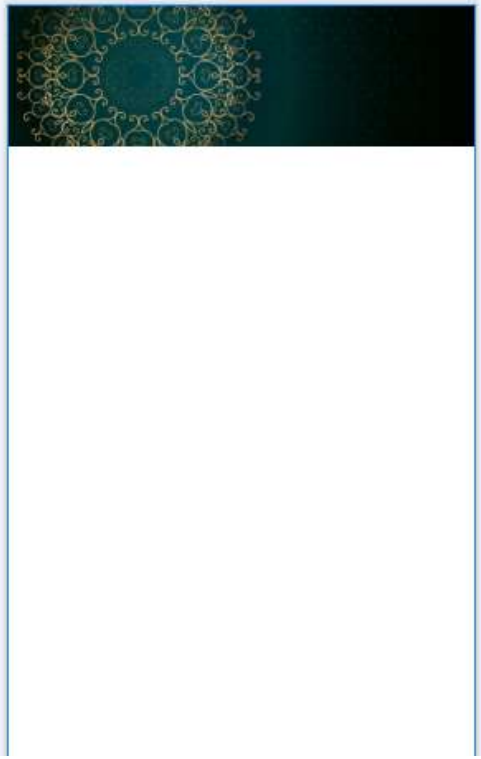
```
buttonDecrease.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        counter--;
        textView.setText(Integer.toString(counter));
    }
});
```

Coding

```
        buttonIncrease.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                counter++;
                setTextView(counter);
            }
        });

        buttonDecrease.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                counter--;
                setTextView(counter);
            }
        });
    }

    public void ResetNumber(View view) {
        counter=0;
        setTextView(counter);
    }

    private  void setTextView(int counter)
    {
        textView.setText(Integer.toString(counter));
    }
}
```

XML Design

```
    xmlns:app="http://schemas.android.com/apk/res-a
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".DesignActivity">

    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="wrap_content"
        android:layout_height="125dp"
        android:scaleType="centerCrop"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parer
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/islamic_header" />
</androidx.constraintlayout.widget.ConstraintLayout
```

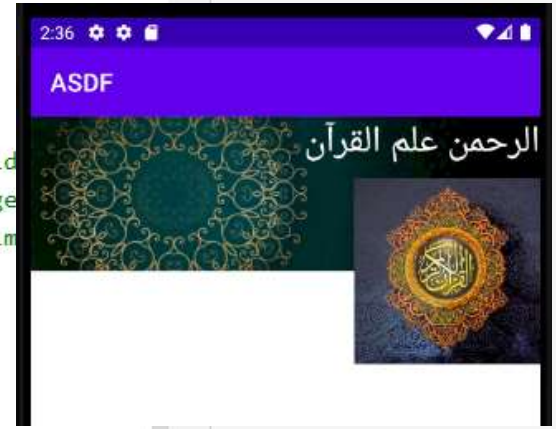


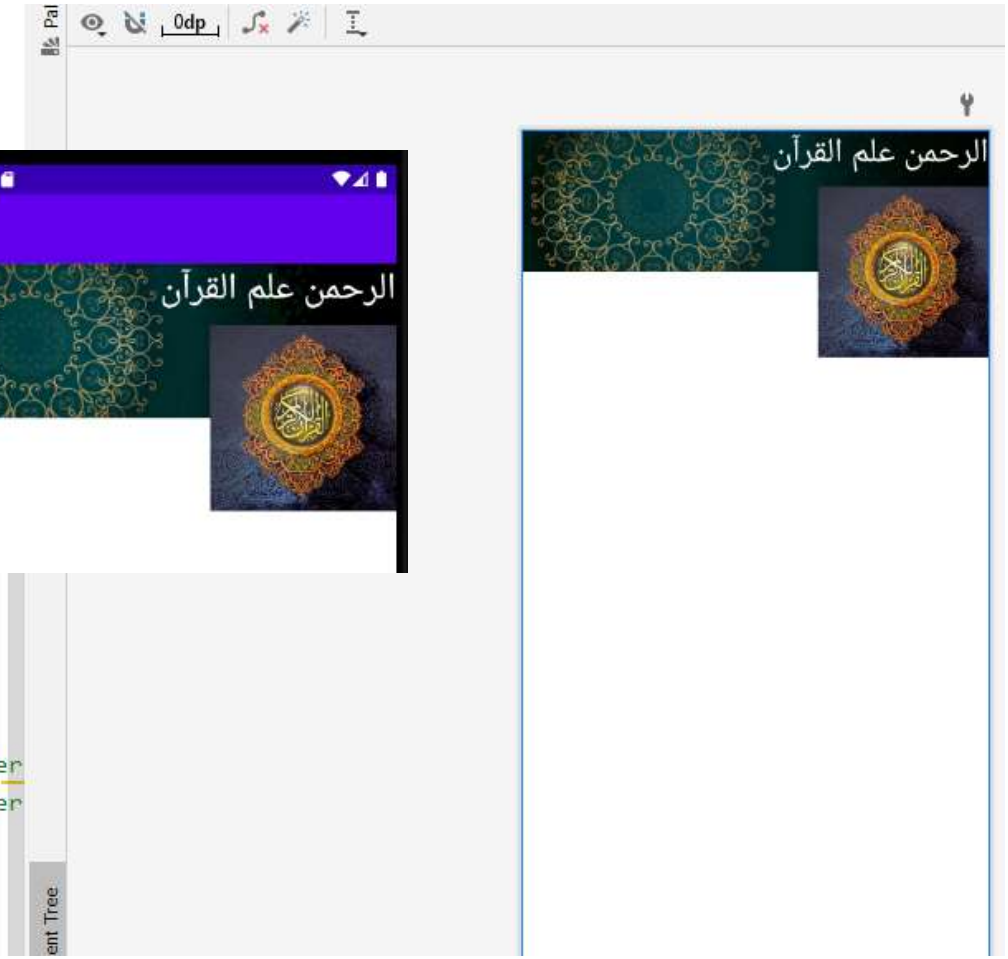
XML Design

```
<ImageView
    android:id="@+id/imageViewHeader"
    android:layout_width="wrap_content"
    android:layout_height="125dp"
    android:layout_marginTop="4dp"
    android:scaleType="centerCrop"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parer
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/islamic_header" />

<ImageView
    android:id="@+id/imageViewSmall"
    android:layout_width="150dp"
    android:layout_height="150dp"
    app:layout_constraintBottom_toBottomOf="@+
    app:layout_constraintEnd_toEndOf="@+id/imag
    app:layout_constraintTop_toBottomOf="@+id/
    app:srcCompat="@drawable/quran" />
</androidx.constraintlayout.widget.ConstraintLayout
```

XML Design

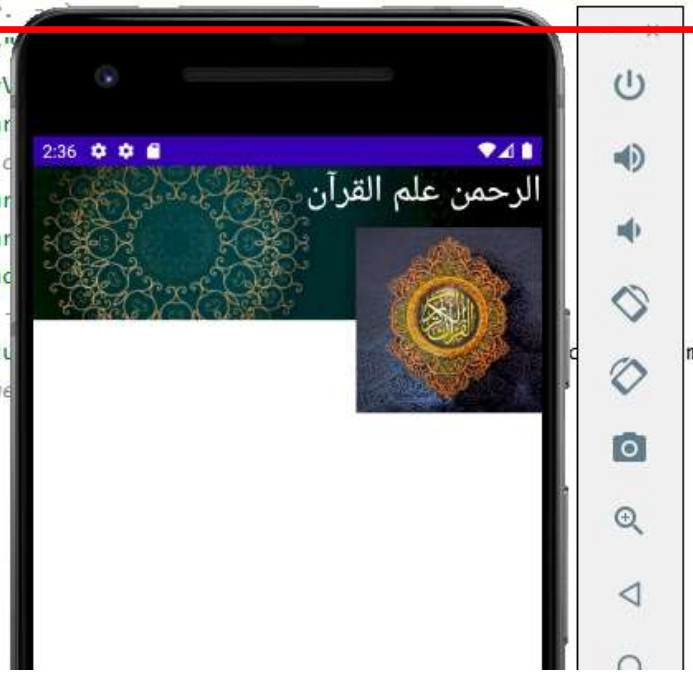
```
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/islamic_header" />

<ImageView
    android:id="@+id/imageViewSmall"
    android:layout_width="150dp"
    android:layout_height="150dp"
    app:layout_constraintBottom_toBottomOf="@+id
    app:layout_constraintEnd_toEndOf="@+id/image
    app:layout_constraintTop_toBottomOf="@+id/im
    app:srcCompat="@drawable/quran" />

<TextView
    android:id="@+id/textViewHeader"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="الرحمن علم القرآن"
    android:textColor="@color/white"
    android:textSize="30sp"
    app:layout_constraintEnd_toEndOf="@+id/imageViewHeader
    app:layout_constraintTop_toTopOf="@+id/imageViewHeader
androidx.constraintlayout.widget.ConstraintLayout>
```





XML Design

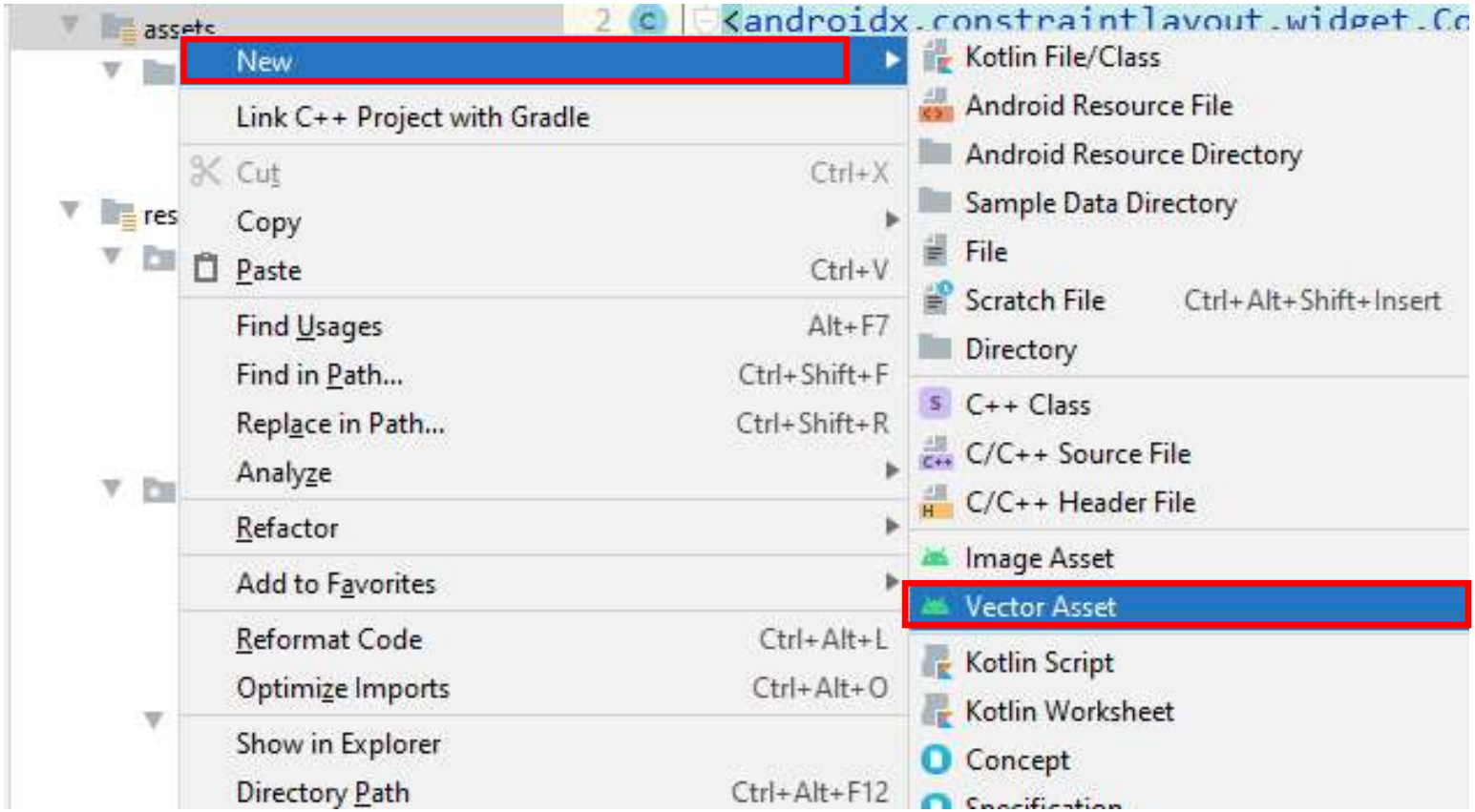
```
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Theme.ASDF" parent="Theme.MaterialComponents.DayNight.NoActionBar">
        <!-- Primary brand color. -->
        <item name="colorPrimary"
        <item name="colorPrimaryV
        <item name="colorOnPrimar
        <!-- Secondary brand colo
        <item name="colorSecondar
        <item name="colorSecondar
        <item name="colorOnSecond
        <!-- Status bar color. --
        <item name="android:statu                                    maryVariant</item>
        <!-- Customize your theme
    </style>
</resources>
```



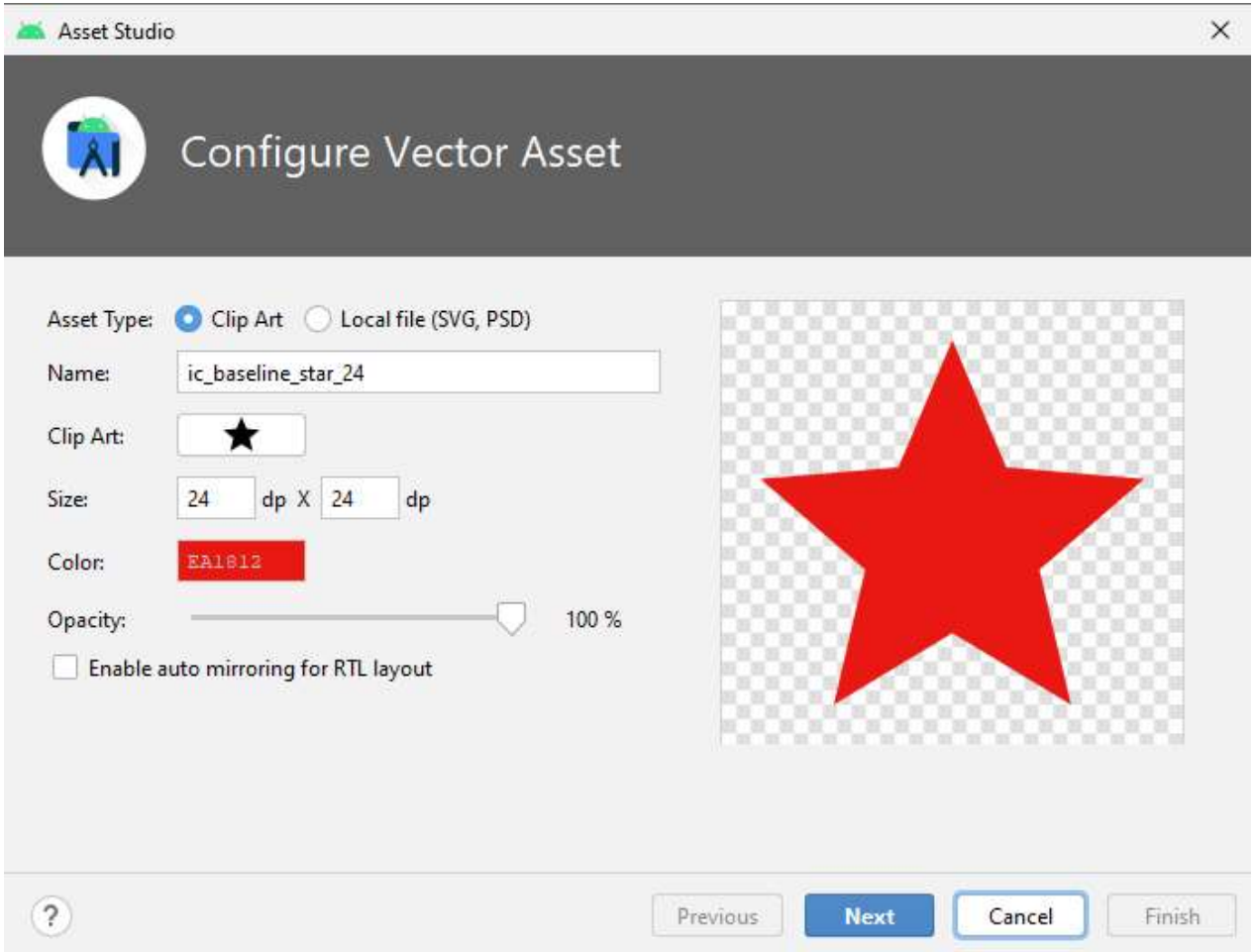
XML Design

```
xmlns:app="http://schemas.android.com/apk/res-a
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".DesignActivity">

<TextView
    android:id="@+id/textViewHeader1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="وب کو ذکر اللہ سے منور کرو
    android:textColor="#00FD0A"
    android:textSize="22sp"
    app:layout_constraintEnd_toStartOf="@+id/im
    app:layout_constraintStart_toStartOf="paren
    app:layout_constraint Top_toBottomOf="@+id/i

<ImageView
    android:id="@+id/imageViewHeader"
    android:layout_width="wrap_content"
    android:layout_height="125dp"
    android:scaleType="centerCrop"
    app:layout_constraintEnd_toEndOf="parent"
```

Vector Asset



Vector Asset



Vector Asset

```
<TextView
    android:id="@+id/textViewHeader"
    android:layout_width="wrap_conter
    android:layout_height="wrap_conte
    android:text="الرحمن علم القرآن"
    android:textColor="@color/white"
    android:textSize="30sp"
    app:layout_constraintEnd_toEndOf=
    app:layout_constraintTop_toTopOf=

<ImageView
    android:id="@+id/imageView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintEnd_toEndOf="@+id/textViewHeader1
    app:layout_constraintStart_toStartOf="@+id/textViewHea
    app:layout_constraintTop_toBottomOf="@+id/textViewHead
    app:srcCompat="@drawable/ic_baseline_star_24" />

/androidx.constraintlayout.widget.ConstraintLayout>
```

Vector Asset

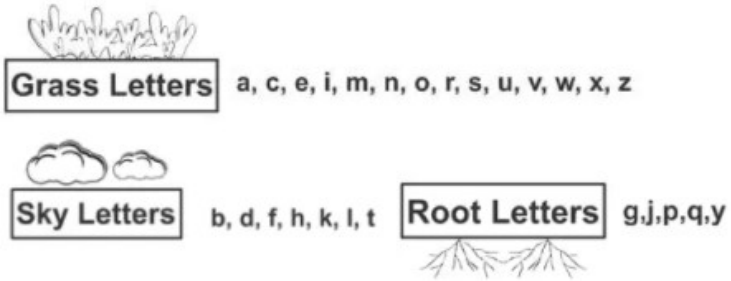
Vector Asset

Font in XML

قلوب کو ذکر اللہ سے منور کرو
12
24
24
ImageV
قلوب کو ذکر اللہ سے منور کرو
12
24

Resources

- https://developer.android.com
- https://developer.android.com/guide/topics/ui/declaring-layout
- https://developer.android.com/guide/topics/ui/look-and-feel/fonts-in-xml

Task 2

- https://www.tcf.org.pk/zakat/zakat-calculator/
- “Siam” and “Salah” calculator for decease
  - نیز ایک نماز اور ایک روزہ کا فدیہ صدقہ فطر کی مقدار کے برابر ہے۔ فدیہ ادا کرنے کی صورت میں یومیہ چھ نمازوں (پنج وقتہ فرض نماز اور وتر) کے حساب سے فدیہ ادا کرنا ہوگا۔
  - **اور ایک صدقہ فطر تقریباً پونے دو کلو گندم یا آٹا یا اس کی قیمت ہے**
- Unit conversion Calculator
- Alphabet converter for kids
- Color recognizer for kids
- Sky, Grass and root letters recognizer for kids