tep 1: Create a New Android Studio Project

1. Open Android Studio.
2. Click on "Start a new Android Studio project".
3. Choose "Empty Activity" and click "Next".
4. Name your application (e.g., `MyFirstApp`), select the save location, and choose the language as Java.
5. Click "Finish" to create the project.

Step 2: Open the Layout File (activity_main.xml) and Add Views

1. In the project view, navigate to `res` > `layout` > `activity_main.xml`.
2. Open `activity_main.xml` in the design or text view.
3. Add the following XML code to include a `TextView`, `EditText`, and `Button` inside a `LinearLayout`:
4.
```xml
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, World!"
        android:textSize="18sp"
        android:textColor="#000000"
        android:layout_gravity="center"/>

    <EditText
        android:id="@+id/editText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter your name"
        android:layout_marginTop="16dp"/>

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Submit"
        android:layout_marginTop="16dp"
        android:layout_gravity="center"/>
</LinearLayout>
```

5. Run the application to see the basic layout.

Step 3: Modify and Customize Views

1. Open `activity_main.xml`.
2. Add an `ImageView` to the layout:

```
3. <LinearLayout
       xmlns:android="http://schemas.android.com/apk/res/android"
       android:layout_width="match_parent"
       android:layout_height="match_parent"
       android:orientation="vertical"
       android:padding="16dp">

       <TextView
           android:id="@+id/textView"
           android:layout_width="wrap_content"
           android:layout_height="wrap_content"
           android:text="Hello, World!"
           android:textSize="18sp"
           android:textColor="#000000"
           android:layout_gravity="center"/>

       <EditText
           android:id="@+id/editText"
           android:layout_width="match_parent"
           android:layout_height="wrap_content"
           android:hint="Enter your name"
           android:layout_marginTop="16dp"/>

       <Button
           android:id="@+id/button"
           android:layout_width="wrap_content"
           android:layout_height="wrap_content"
           android:text="Submit"
           android:layout_marginTop="16dp"
           android:layout_gravity="center"/>

       <ImageView
           android:id="@+id/imageView"
           android:layout_width="wrap_content"
           android:layout_height="wrap_content"
           android:src="@drawable/ic_launcher_foreground"
           android:layout_marginTop="16dp"
           android:layout_gravity="center"/>
   </LinearLayout>
```

Customize the `Button` appearance and handle user input in `MainActivity.java`:

```
package com.example.myapplication;
import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

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


        // Customize Button appearance
        Button button = findViewById(R.id.button);

button.setBackgroundColor(getResources().getColor(R.color.purple_500));
        button.setTextColor(getResources().getColor(R.color.white));

        // Handle EditText input
        EditText editText = findViewById(R.id.editText);
        button.setOnClickListener(v -> {
            String name = editText.getText().toString();
            TextView textView = findViewById(R.id.textView);
            textView.setText("Hello, " + name + "!");
        });
    }
}
```

Step 4: Experiment with Layouts

1. Open `activity_main.xml`.
2. Convert the `LinearLayout` to a `RelativeLayout`:

```
3. <RelativeLayout
       xmlns:android="http://schemas.android.com/apk/res/android"
       android:layout_width="match_parent"
       android:layout_height="match_parent"
       android:padding="16dp">

       <TextView
           android:id="@+id/textView"
           android:layout_width="wrap_content"
           android:layout_height="wrap_content"
           android:text="Hello, World!"
           android:textSize="18sp"
           android:textColor="#000000"
           android:layout_centerHorizontal="true"/>

```

```
    <EditText
        android:id="@+id/editText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter your name"
        android:layout_below="@id/textView"
        android:layout_marginTop="16dp"/>

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Submit"
        android:layout_below="@id/editText"
        android:layout_marginTop="16dp"
        android:layout_centerHorizontal="true"/>

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/ic_launcher_foreground"
        android:layout_below="@id/button"
        android:layout_marginTop="16dp"
        android:layout_centerHorizontal="true"/>
</RelativeLayout>
```

Convert the `RelativeLayout` to a `ConstraintLayout`:

```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:padding="16dp">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, World!"
        android:textSize="18sp"
        android:textColor="#000000"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"/>

    <EditText
        android:id="@+id/editText"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:hint="Enter your name"
        app:layout_constraintTop_toBottomOf="@id/textView"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:layout_marginTop="16dp"/>

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Submit"
        app:layout_constraintTop_toBottomOf="@id/editText"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:layout_marginTop="16dp"/>

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/ic_launcher_foreground"
        app:layout_constraintTop_toBottomOf="@id/button"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:layout_marginTop="16dp"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Run the application after each layout change to observe the differences and discuss the benefits and challenges of each layout approach.