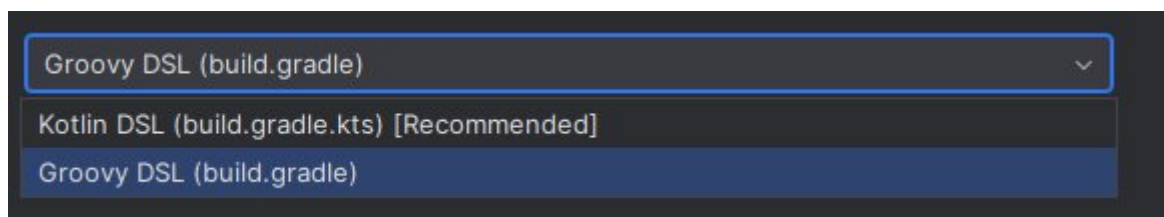
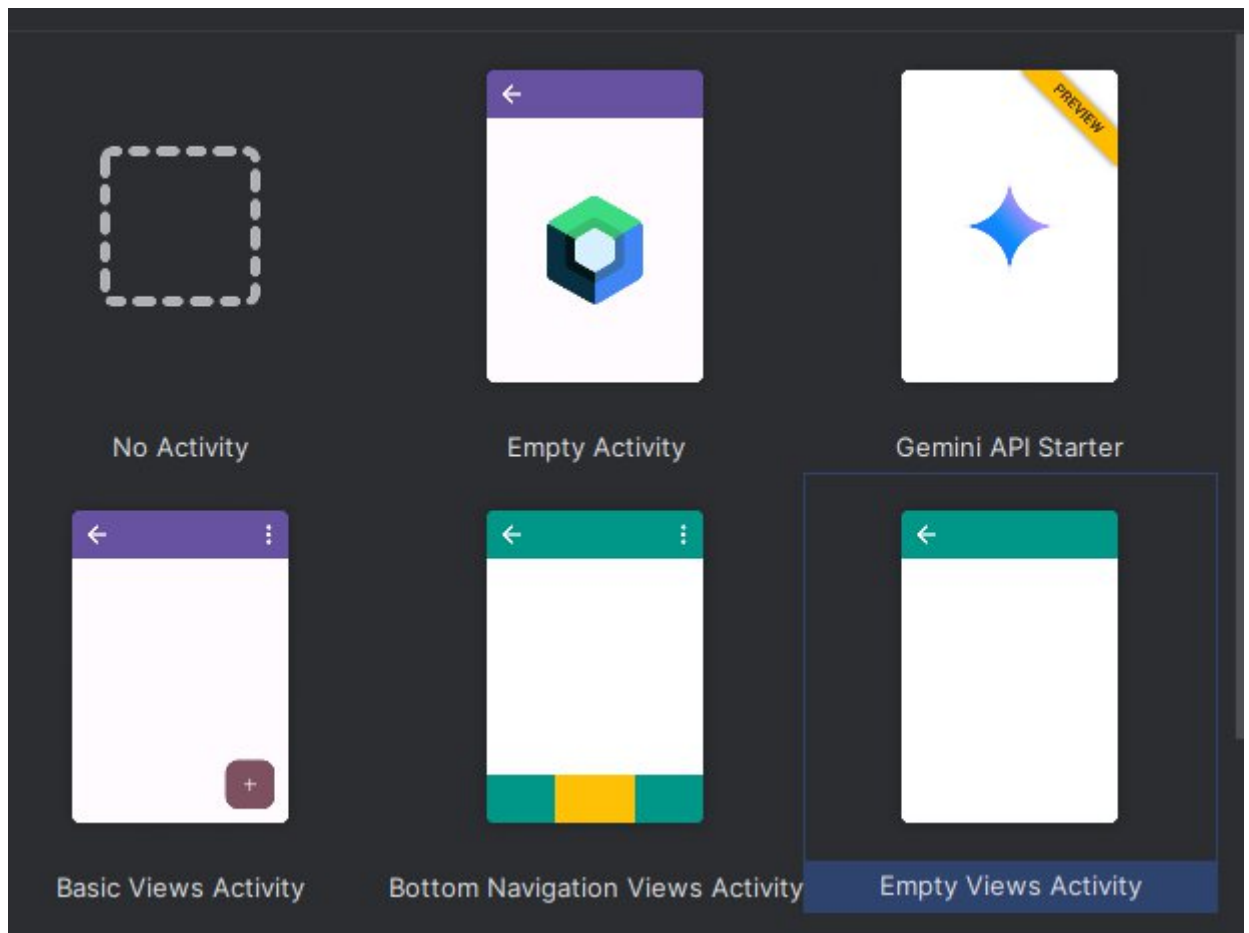
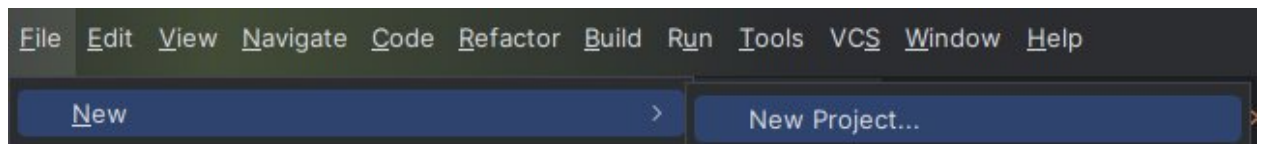


Создание проекта



После создания удалить это:

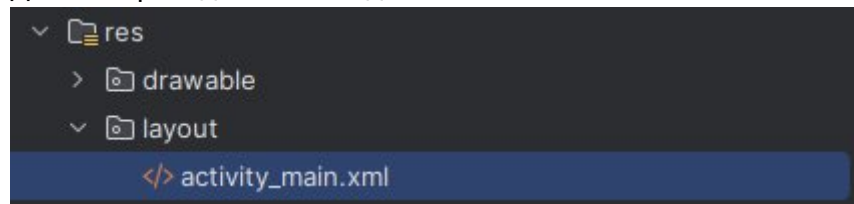
```
EdgeToEdge.enable( $this$enableEdgeToEdge: this);  
ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {  
    Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());  
    v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);  
    return insets;  
});
```

По итогу должно остаться так:

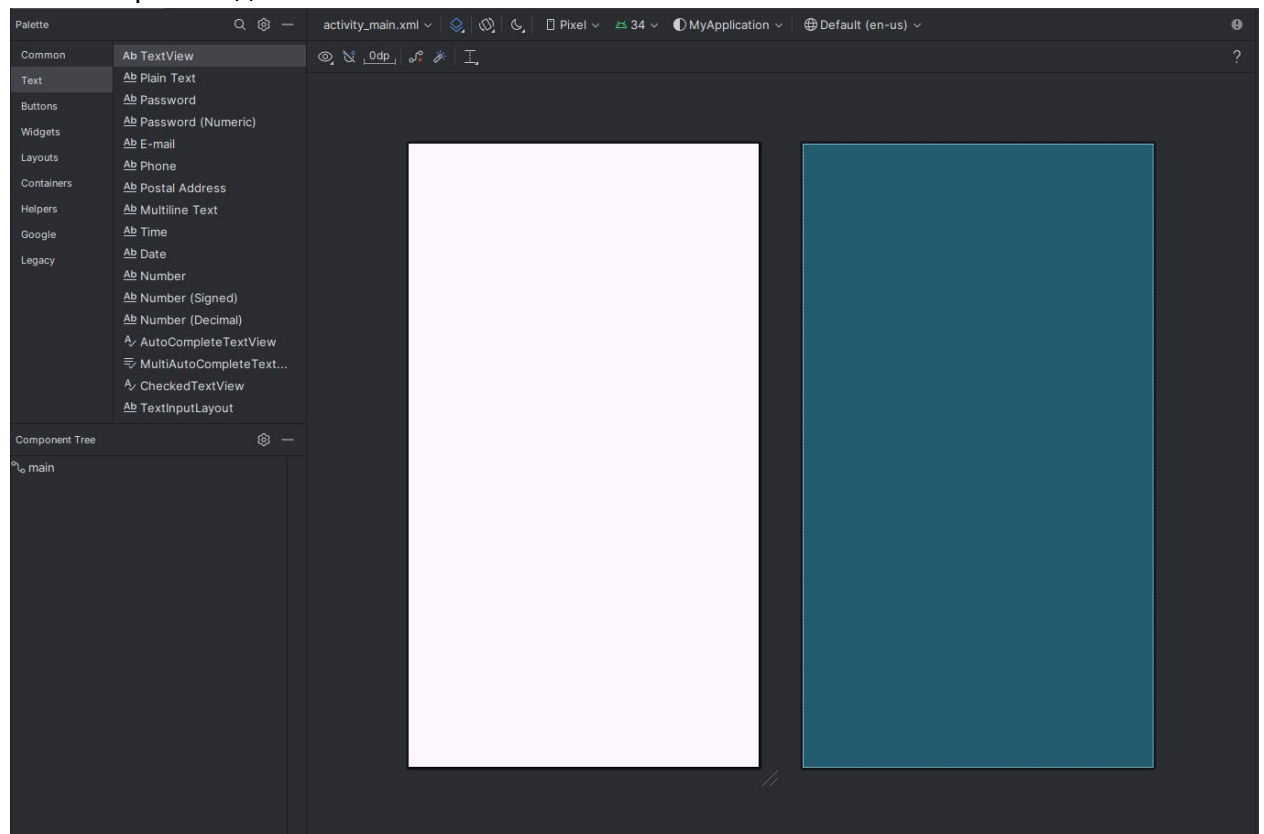
```
public class MainActivity extends AppCompatActivity {

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

Далее переходим на ваш дизайн:

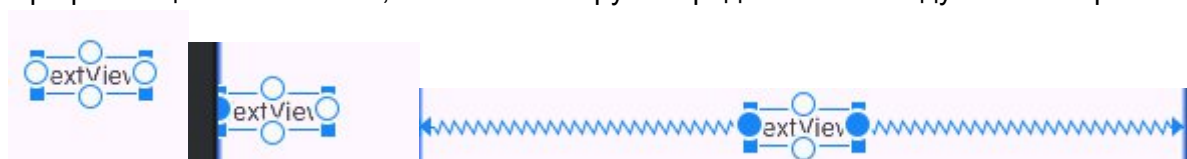


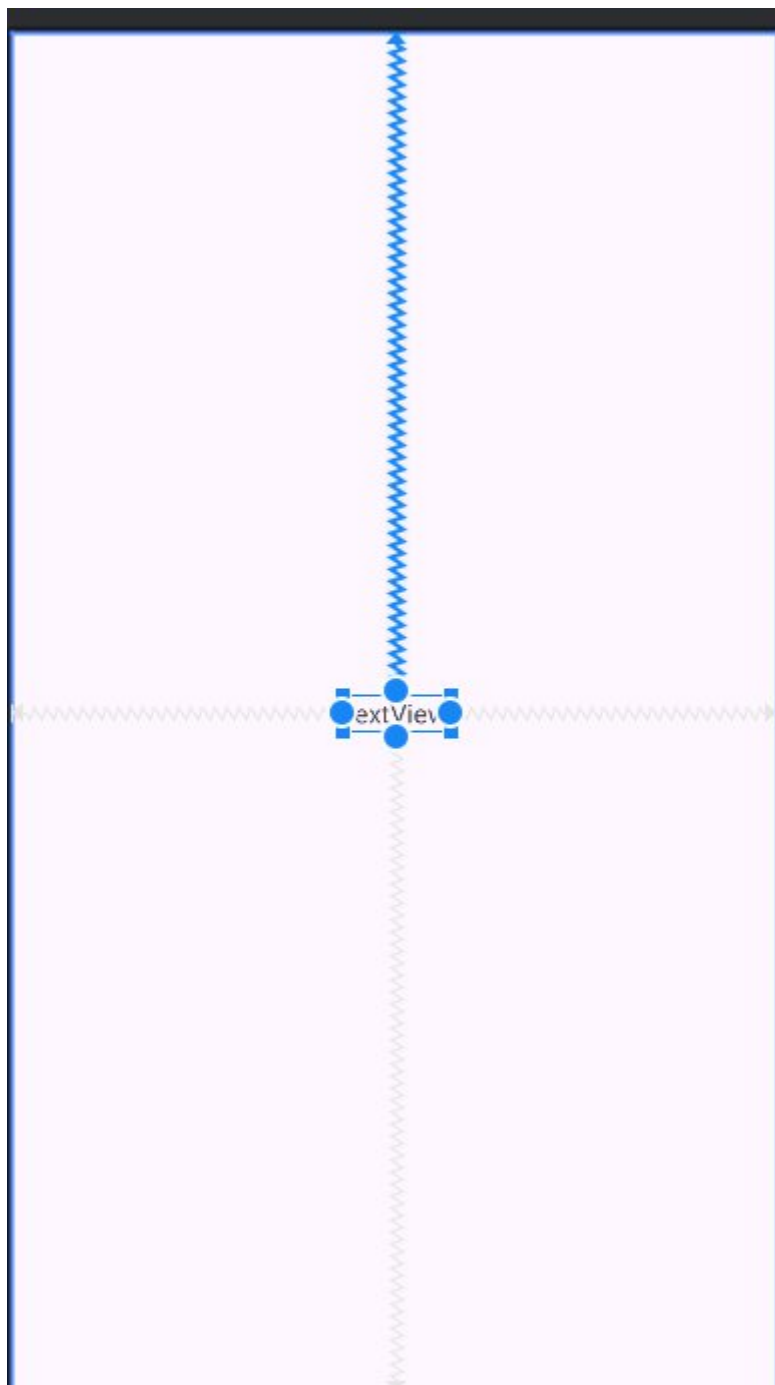
У вас откроется данное меню:



Тут мы можем разместить интересующие нас элементы

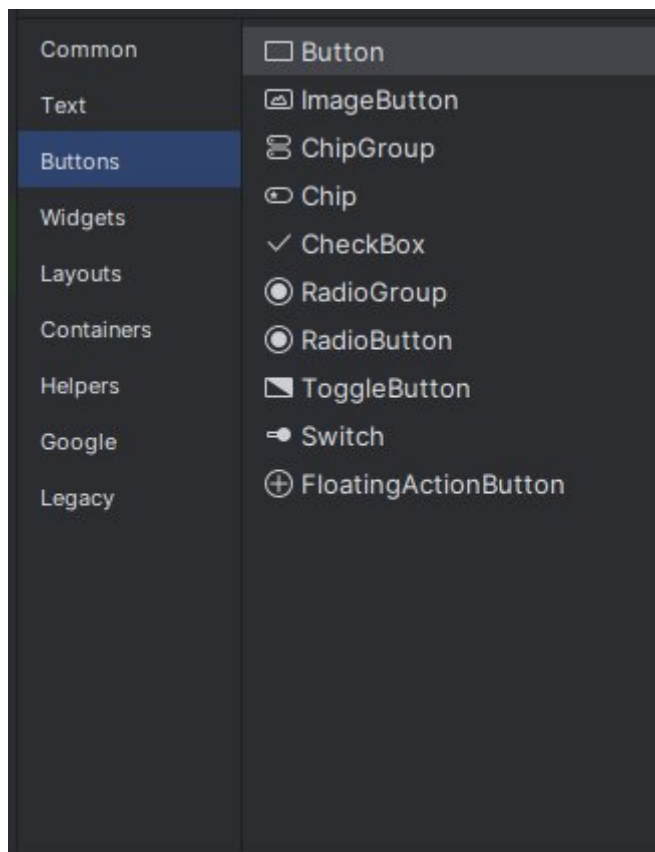
При размещении элемента, потяните его круги определения в каждую часть экрана:





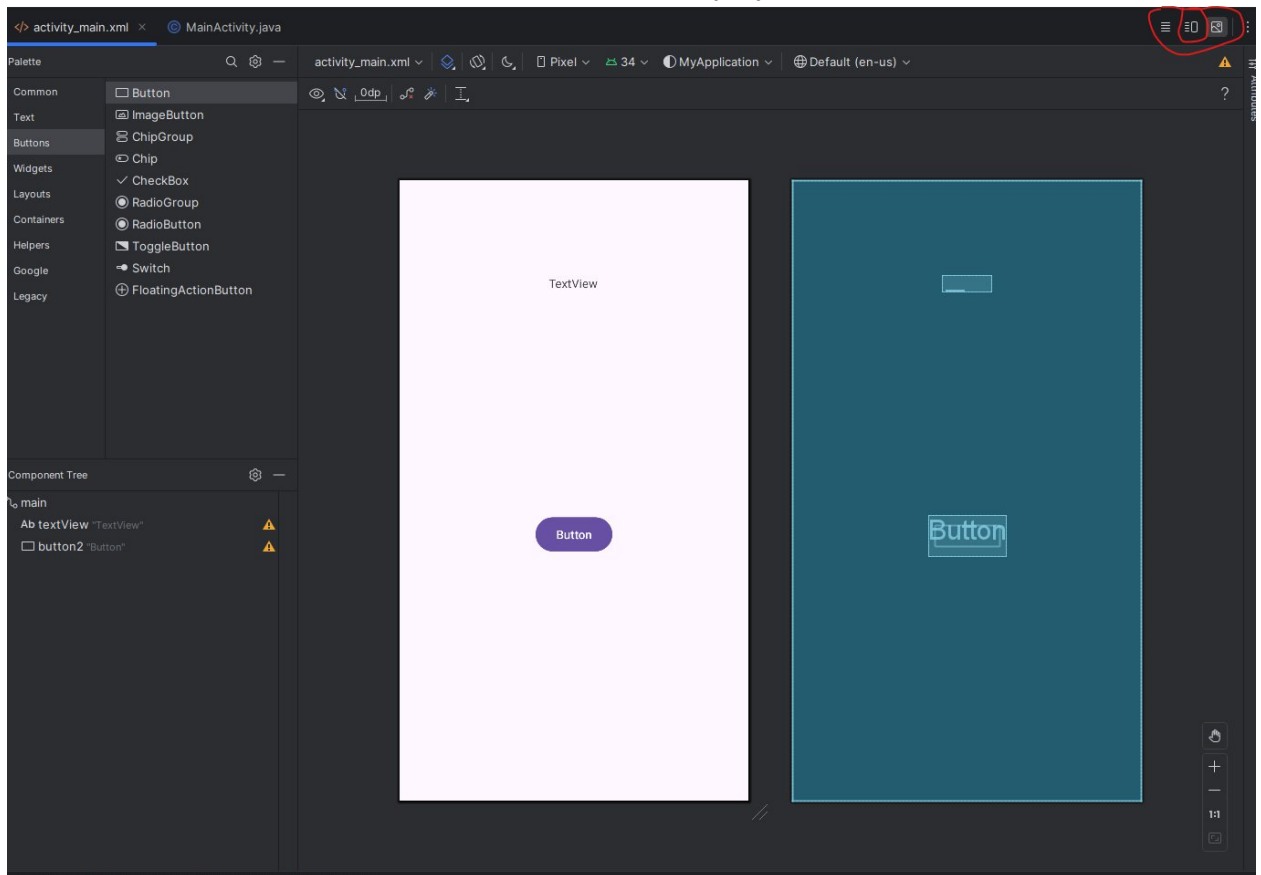
Так ваше приложение будет знать о том, где размещён ваш элемент. Теперь вы свободно можете его перемещать по экрану.

Давайте добавим кнопку. Для этого в палетке переходим в раздел Buttons и выбираем Button:

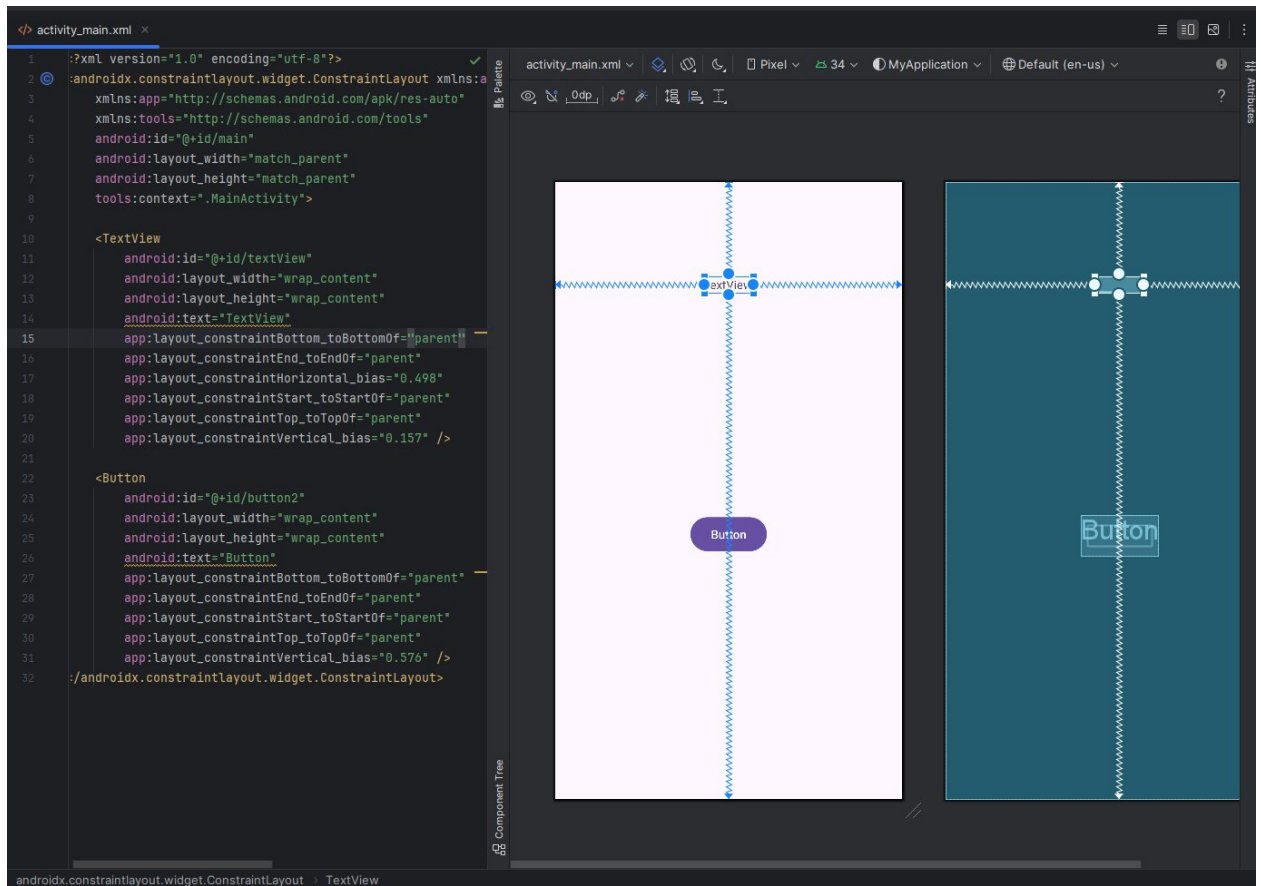


Выполняем все те же действия, как и с текстом.

Далее нажимаем на split дизайн в правом верхнем углу



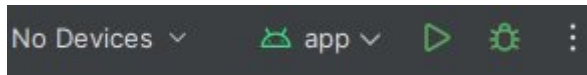
У вас откроется это меню:



Тут вы можете выполнять махинации с вашими элементами с помощью кода. Нас интересует android:text и android:id

Измените текст и id у элементов на те, что вам захочется(или можете оставить базовый).

Можете запустить ваш проект, нажав на кнопку, рядом с app на верхней панели:



Вернёмся к вам в MainActivity. И пропишем ваши элементы в коде:

```
public class MainActivity extends AppCompatActivity {
    TextView textView; 1 usage
    Button button; 1 usage
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        textView = findViewById(R.id.textView);
        button = findViewById(R.id.button);
    }
}
```

Если при написании findViewById у вас будет подчёркиваться красным и вы уверены, что всё написали правильно, то просто не обращайте на него внимание. При сборке и запуске приложения всё исправится.

Давайте объявим вашей кнопке обработчик нажатия. Сделать это можно 2 способами:

```
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

    }
});
button.setOnClickListener(v ->{

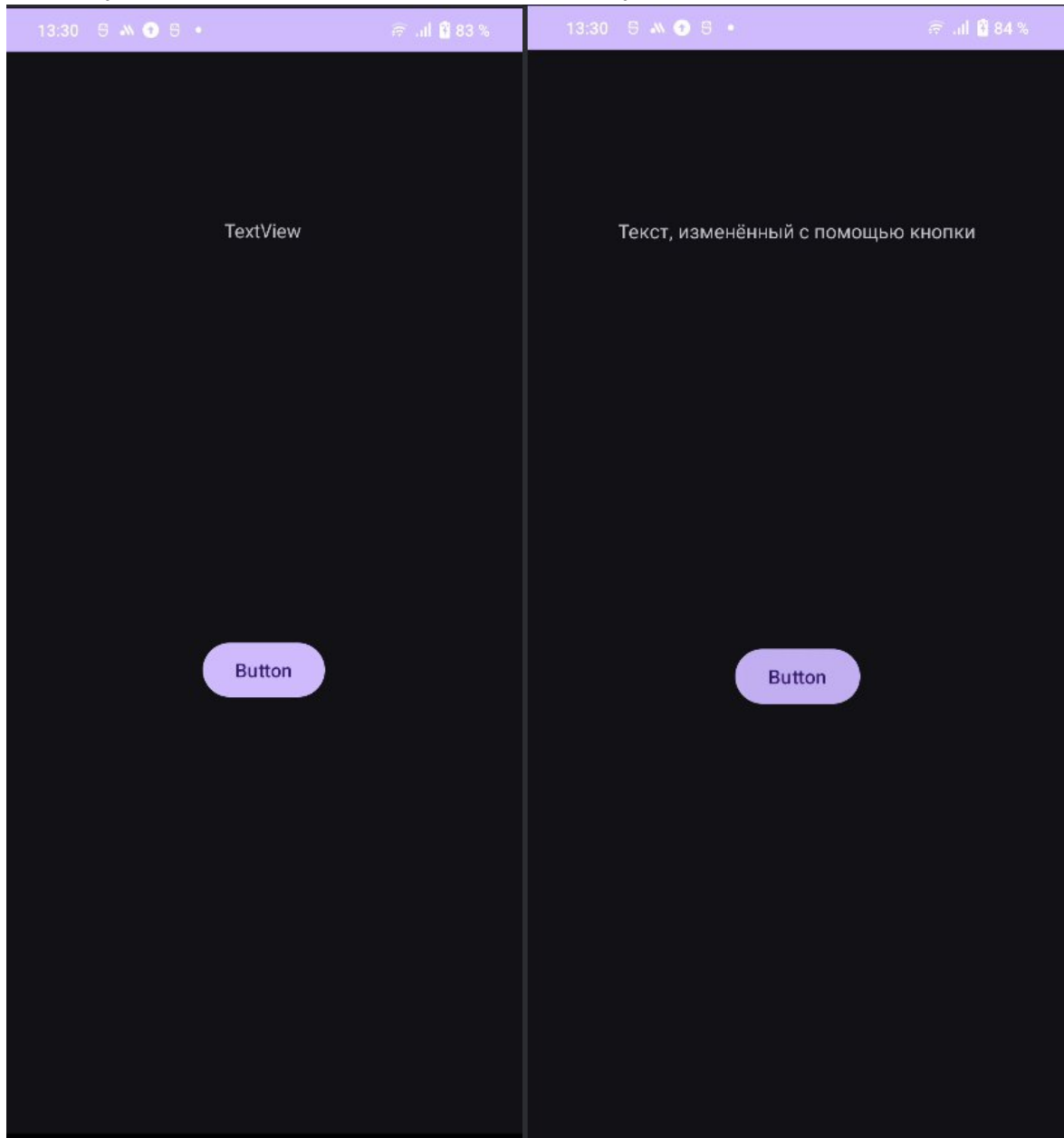
});
```

Выбирайте, какой вам больше нравится. Но от себя советую использовать второй.

Давайте заставим кнопку при нажатии изменять текст. Пишем следующий код:

```
button.setOnClickListener(v ->{
    textView.setText("Текст, изменённый с помощью кнопки");
});
```

При запуске вашего приложения нажимаем кнопку и видим, как изменяется текст:



Далее, давайте заставим ваше приложение выводить всплывающее сообщение. для этого возвращаемся в MainActivity и дописываем:

```
button.setOnClickListener(v ->{  
    textView.setText("Текст, изменённый с помощью кнопки");  
    Toast.makeText(context: this, text: "Нажал на кнопку", Toast.LENGTH_LONG).show();  
});
```

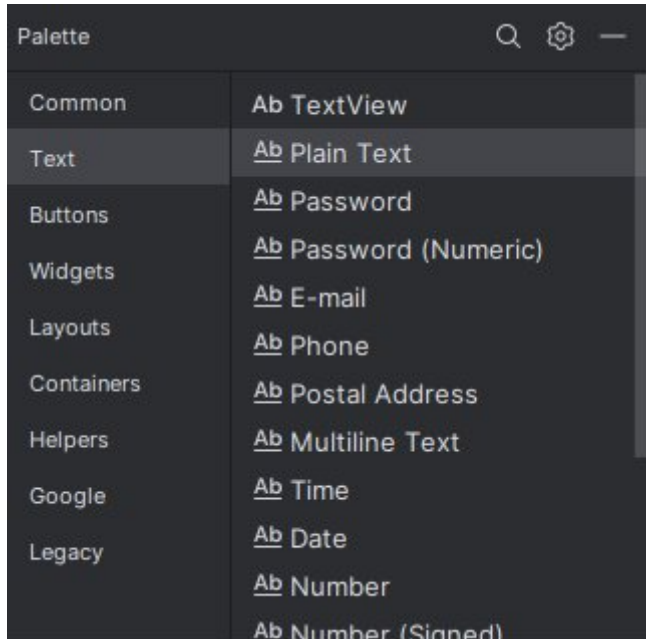
Текст, изменённый с помощью кнопки

Button



Нажал на кнопку

Теперь, давайте попробуем вводить текст. Для этого возвращаемся в ваш дизайн и в палетке выбираем Text и PlainText.



Делаем всё те же махинации, что и с текстом и кнопкой. А также уберём текст у editText и пропишем hint. Hint позволит разместить текст, который будет подсказывать, что писать человеку и при вводе оно будет пропадать.

```
android:hint="Напиши текст сюда"
```

А также в MainActivity пропишем EditText.

```
TextView textView; 2 usages
Button button; 2 usages
EditText editText; 1 usage
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    textView = findViewById(R.id.textView);
    button = findViewById(R.id.button);
    button.setOnClickListener(v ->{
        textView.setText("Текст, изменённый с помощью кнопки");
        Toast.makeText(context: this, text: "Нажал на кнопку", Toast.LENGTH_LONG).show();
    });
    editText = findViewById(R.id.editTextText);
}
```

Получить текст из editText можно с помощью getText(). И обязательно нужно перевести этот текст в string, с помощью toString().

```
editText.getText().toString();
```

Задание

Добавьте ещё один `TextView` и `EditTextView`, а также 2 кнопки.

Сделайте, чтобы в ваш первый `TextView` вводился текст из первого `EditTextView` при нажатии кнопки и `EditTextView` очищается. Во второй `TextView` вводился текст из второго `EditTextView` при нажатии второй кнопки и `EditTextView` очищается.

При нажатии 3 кнопки, оба `TextView` должны удалить свой текст.