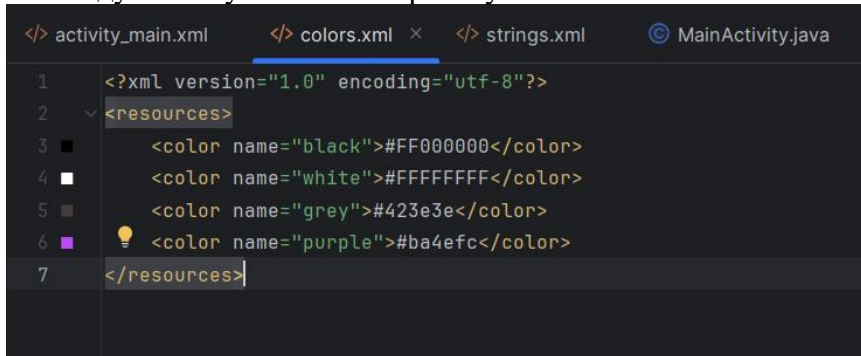


## Практика: Введение. Работа с макетами

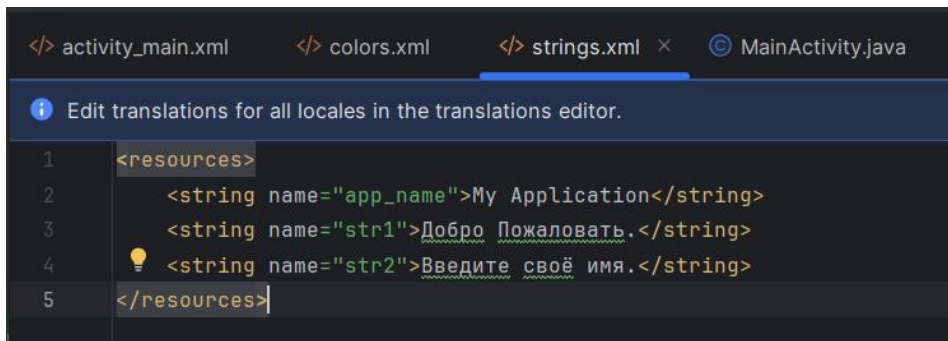
Кто работал: Наглов Виктор Ис221

### Ход работы:

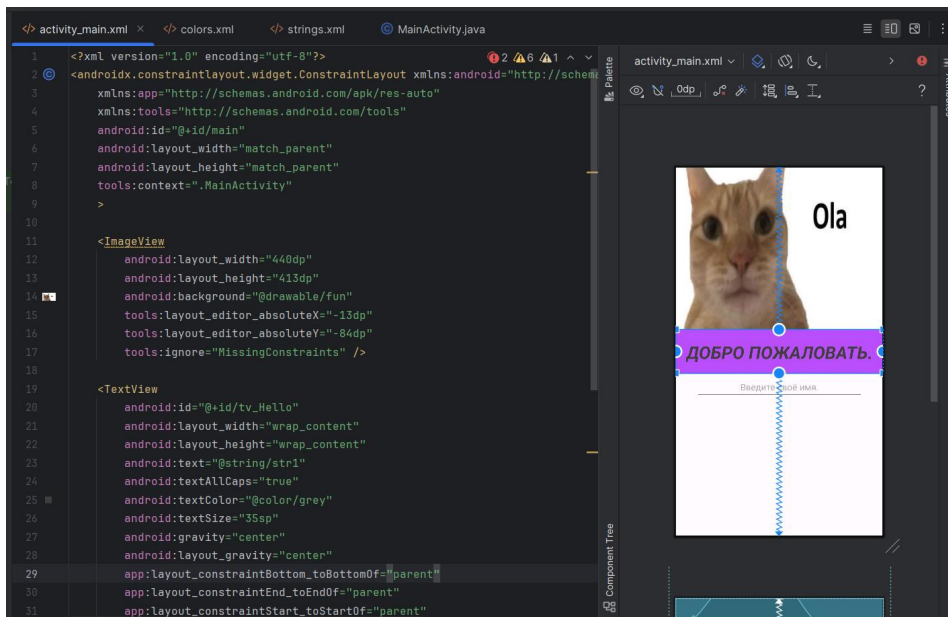
1. Создаем новый пустой проект в Android Studio оставив название My Appli.
2. Знакомимся с ключевыми файлами, с которыми нам нужно будет работать.
3. Следуя тексту выполняем работу



```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
    <color name="grey">#423e3e</color>
    <color name="purple">#ba4efc</color>
</resources>
```



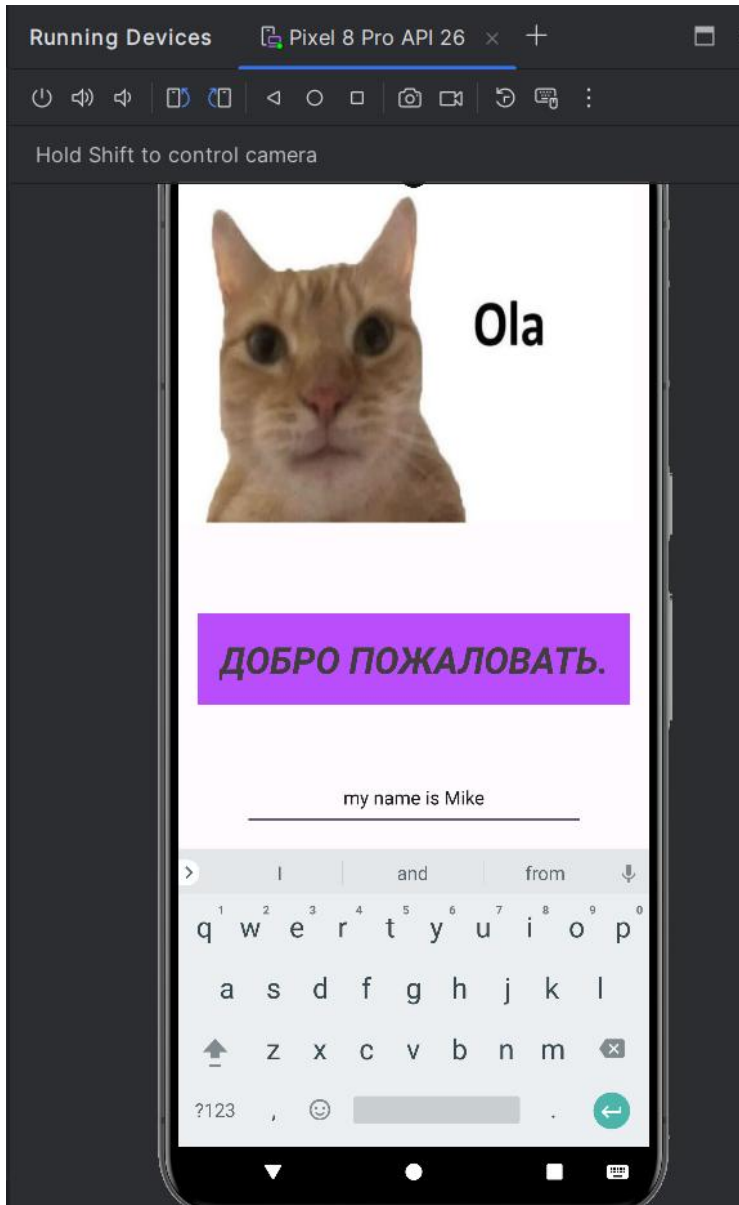
```
<resources>
    <string name="app_name">My Application</string>
    <string name="str1">Добро Пожаловать.</string>
    <string name="str2">Введите своё имя.</string>
</resources>
```



```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    >
    <ImageView
        android:layout_width="440dp"
        android:layout_height="413dp"
        android:background="@drawable/fun"
        tools:layout_editor_absoluteX="-13dp"
        tools:layout_editor_absoluteY="-84dp"
        tools:ignore="MissingConstraints" />
    <TextView
        android:id="@+id/tv_Hello"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/str1"
        android:textAllCaps="true"
        android:textColor="@color/grey"
        android:textSize="35sp"
        android:gravity="center"
        android:layout_gravity="center"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
    >
```

Вместо заднего фона установил просто фото, т.к. картинка искажалась.

По итогу получили это:



**Вывод:** в ходе работы ознакомился с приложением, основными функциями и эмулятором. Как оказалось принцип работы и визуальные составляющие Android Studio, не особо отличаются от Microsoft Visual Studio, что в своем роде помогло в ходе работы.

Под конец работы мы получили приложение в котором мы можем вставлять некий текст в текстовое поле и можем наслаждаться картинкой.

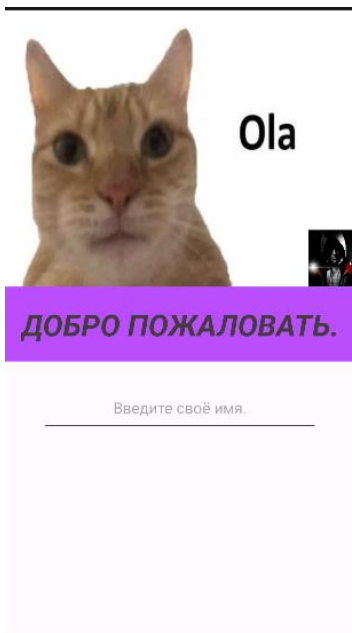
## Практика: Практическая работа (Ресурсы)

Кто работал: Наглов Виктор Ис221

### Ход работы:

1. Следуя тексту работы начнем выполнение.
2. Задание 1. Рассмотрим папки категории res, посмотрим за что они отвечают и что можно сделать с ними.
  - а) drawable - Папки в этой категории отвечают за хранение различных типов изображений в соответствии с требованиями приложения. В них могут содержаться ресурсы изображений, фигур, градиентов.

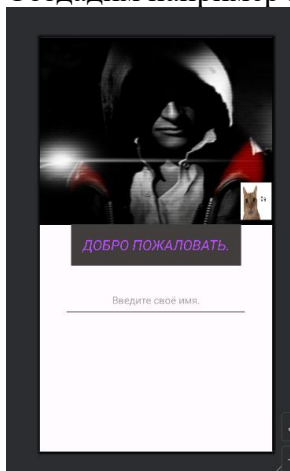
Мы можем например сменить картинку фона или же добавить новую картинку



Как мы теперь видим, нас теперь встречает не только кот но и маленький Алекс

- б) layout - файлы разметки для интерфейсов.

Создадим например второй интерфейс в котором изменим цвета/фото/шрифты

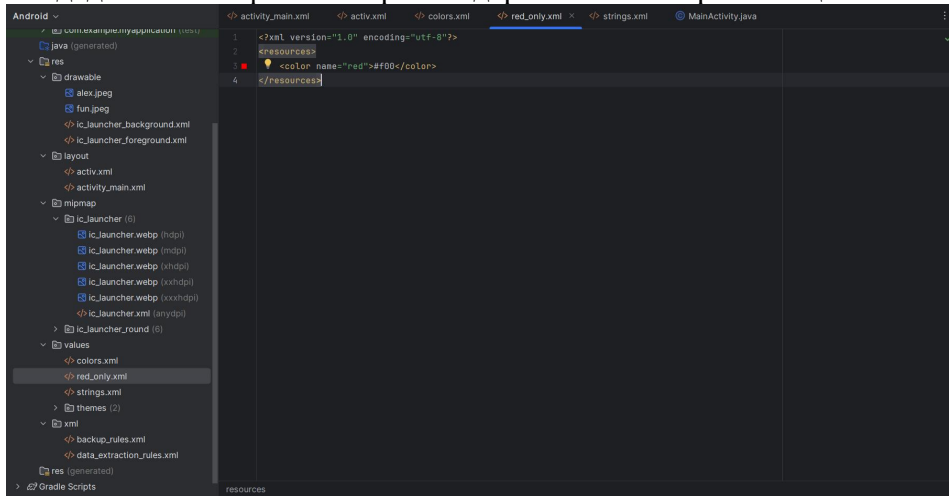


c) **miptar** — иконки приложения.

d) **values** — файлы атрибутов, строк и другие ресурсы, необходимые для приложения.

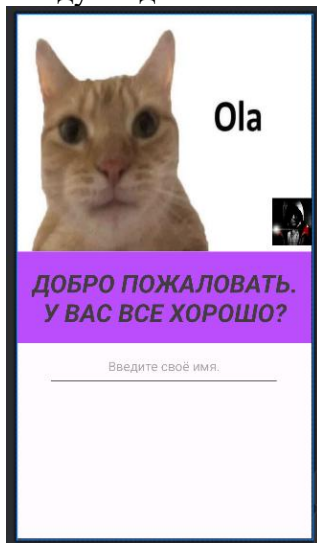
Здесь мы можем создать как свой файл с новыми ресурсами или же изменить текущие **string** и **color**

Создадим новый файл который содержит только красный цвет

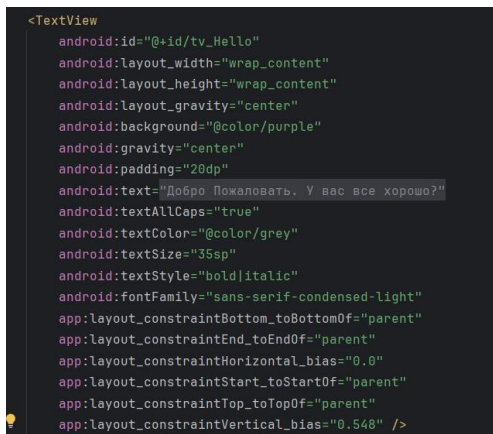


### 3. Задание 2.

Следуя заданию сменим выводимый текст



4. Просмотрим основные атрибуты **TextView**, с которыми в ходе работы мы уже успели познакомиться.



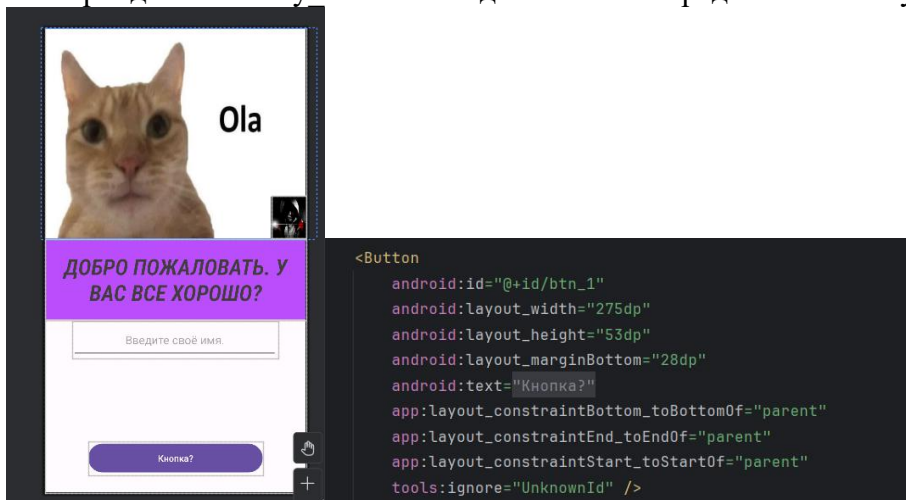
**Вывод:** в ходе этой работы я разобрался какие папки категории **res** за что отвечают и что в них можно создавать.

## Практика: Кнопка

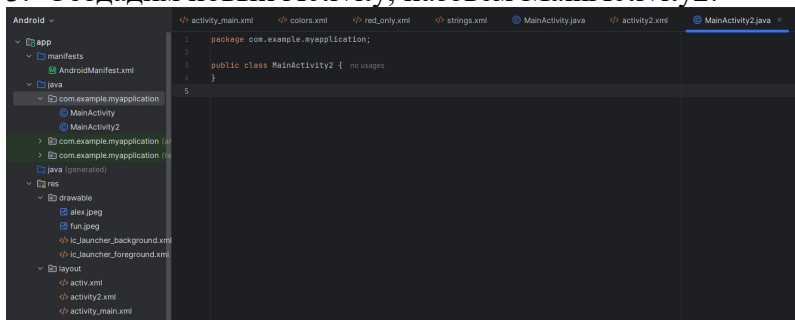
Кто работал: Наглов Виктор Ис221

### Ход работы:

1. Следуя тексту работы начнем выполнение.
2. Перейдем в activity main.xml и добавим непосредственно нашу кнопку.



3. Создадим новый Activity, назовем MainActivity2:

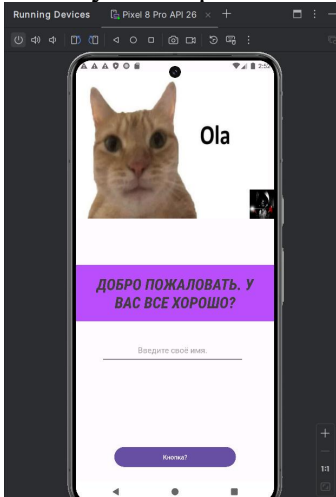


Он у нас естественно пока что пустой.

4. Перейдем в MainActivity, который является основным и сделаем возможность перехода в MainActivity2.



5. Запустим приложение и проверим.



Как видим кнопка присутствует. Если мы на нее нажмем то приложение выключится т.к. мы переходим в MainActivity2, который является пустым.

**Вывод:** в этой практике я узнал как создавать кнопку

## Практика: Практическая работа Intent. Переход между Активностями

Кто работал: Наглов Виктор Ис221

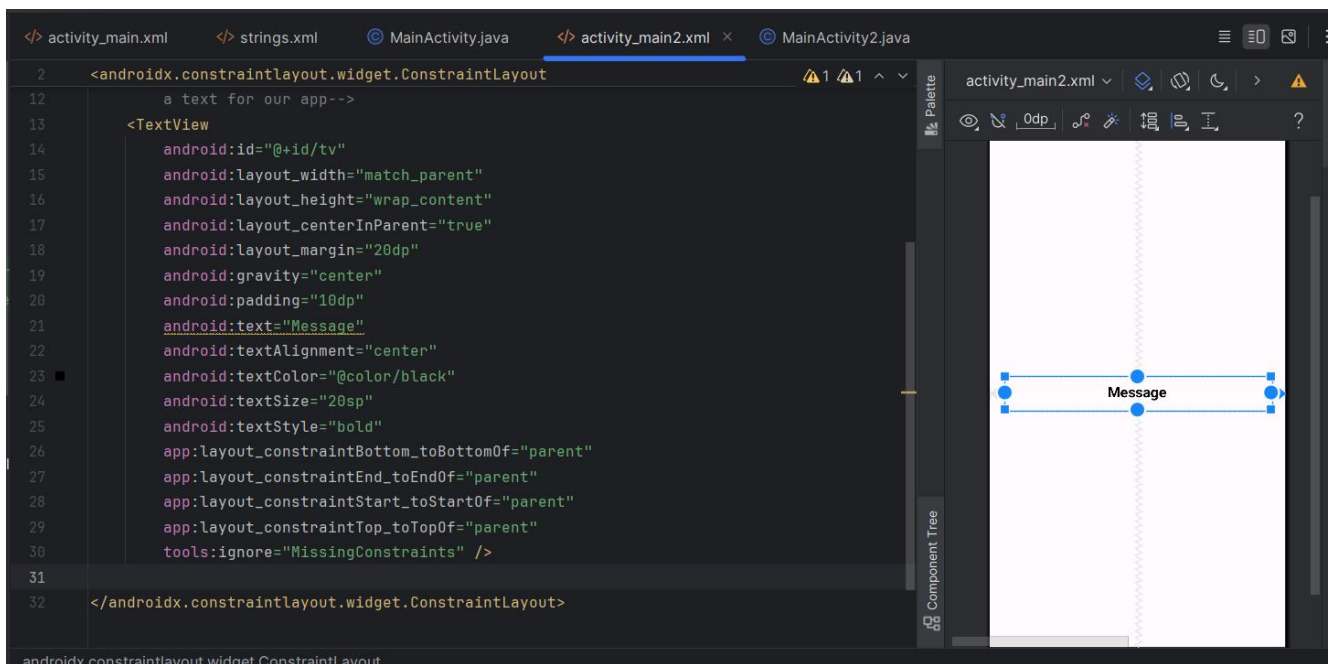
### Ход работы:

1. Следуя тексту работы начнем выполнение.

1. Добавляем дополнительные строки кода в MainActivity, MainActivity2 и xml файл MainActivity2, для создания передачи данных с одного Activity в другой.

```
16 public class MainActivity extends AppCompatActivity {
17     protected void onCreate(Bundle savedInstanceState) {
18         super.onCreate(savedInstanceState);
19         EdgeToEdge.enable(this);
20         setContentView(R.layout.activity_main);
21         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
22             Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
23             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
24             return insets;
25         });
26
27         Button btn = findViewById(R.id.btn_1);
28         btn.setOnClickListener(new View.OnClickListener() {
29             @Override
30             public void onClick(View v) {
31                 EditText poluch = findViewById(R.id.edtMsg);
32                 String text = poluch.getText().toString();
33                 Intent intent = new Intent(packageContext, MainActivity2.class);
34                 intent.putExtra("name", text);
35                 startActivity(intent);
36             }
37         });
38     }
39 }
40 }
```

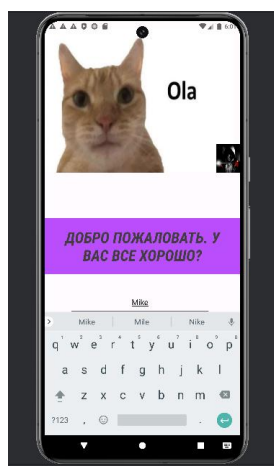
```
17 public class MainActivity2 extends AppCompatActivity {
18     protected void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         EdgeToEdge.enable(this);
21         setContentView(R.layout.activity_main2);
22         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
23             Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
24             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
25             return insets;
26         });
27
28         TextView text = findViewById(R.id.tv);
29         Intent intent = getIntent();
30         String message = intent.getStringExtra("name");
31         text.setText(message);
32     }
33 }
34 }
```



Второй Активности, по своей задумке должен просто выводить введенное в первом Активности имя.

2. Проверяем работоспособность

Введем имя Mike



3. Нажимаем на кнопку и смотрим что нам выведет



Как видим вывелось “Mike”, что подтверждает что все работает.

**Вывод:** в этой практике я выяснил как сделать переход с одной активности в другую с переносом значений используя кнопку. Несмотря на наличие “начального” материала в практике, пришлось также поискать в интернете некоторые моменты.



Листинги:

## MainActivity:

```
package com.example.myapplication;
import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {
    @SuppressLint("MissingInflatedId")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        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;
        });
        Button btn = findViewById(R.id.btn_1);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                EditText poluch = findViewById(R.id.idEdtMsg);
                String text = poluch.getText().toString();
                Intent intent = new Intent(MainActivity.this, MainActivity2.class);
                intent.putExtra("text", text);
                startActivity(intent);
            }
        });
    }
}
```

## MainActivity2:

```
package com.example.myapplication;
import android.annotation.SuppressLint;
import android.os.Bundle;
import android.widget.TextView;
import android.content.Intent;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity2 extends AppCompatActivity {
    @SuppressLint("MissingInflatedId")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main2);
        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;
        });
        TextView text = findViewById(R.id.tv);
        Intent intent = getIntent();
        String message = intent.getStringExtra("text");
        text.setText(message);
    }
}
```

## Activity\_main.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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <ImageView
        android:id="@+id/imageView1"
        android:layout_width="434dp"
        android:layout_height="333dp"
        android:background="@drawable/fun"
        tools:ignore="MissingConstraints"
        tools:layout_editor_absoluteX="-7dp"
        tools:layout_editor_absoluteY="-3dp" />

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="55dp"
        android:layout_height="66dp"
        android:background="@drawable/alex"
        app:layout_constraintBottom_toBottomOf="@+id/imageView1"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="@+id/imageView1"
        app:layout_constraintVertical_bias="1.0"
        tools:ignore="MissingConstraints" />

    <TextView
        android:id="@+id/tv_Hello"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:background="@color/purple"
        android:gravity="center"
        android:padding="20dp"
        android:text="@string/str1"
        android:textAllCaps="true"
        android:textColor="@color/grey"
        android:textSize="35sp"
        android:textStyle="bold|italic"
        android:fontFamily="sans-serif-condensed-light"
        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="0.548" />

    <EditText
        android:id="@+id/idEdtMsg"
        android:layout_width="323dp"
        android:layout_height="58dp"
        android:layout_margin="40dp"
        android:gravity="center"
        android:hint="@string/str2"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.375"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="@+id/imageView1"
        app:layout_constraintVertical_bias="0.712"
        tools:ignore="MissingConstraints" />

    <Button
        android:id="@+id/btn_1"
        android:layout_width="275dp"
        android:layout_height="53dp"
        android:layout_marginBottom="28dp"
        android:text="@string/str3"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        tools:ignore="UnknownId" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

## activity\_main2.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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
>
    <!--on below line we are creating
         a text for our app-->
    <TextView
        android:id="@+id/tv"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:layout_margin="20dp"
        android:gravity="center"
        android:padding="10dp"
        android:text="Message"
        android:textAlignment="center"
        android:textColor="@color/black"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:ignore="MissingConstraints" />

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