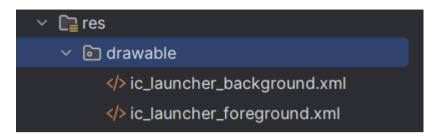
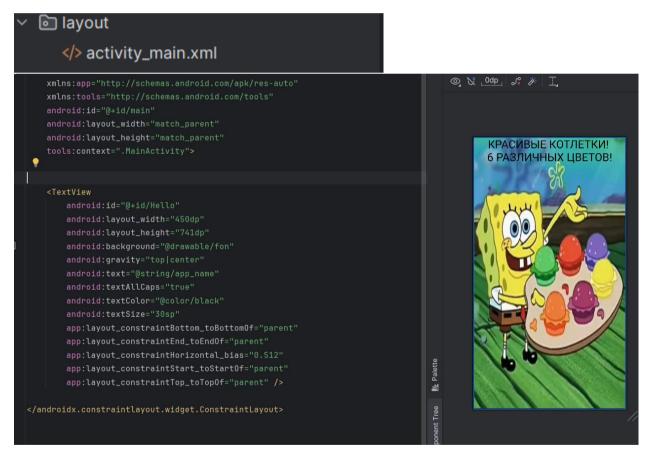
## Практическая работа (Ресурсы)

1.Папка в которой хранится задний фон и передний план



2.Папка в которой находится Layout – это класс, управляющий как его компоненты будут располагаться и выглядеть на экране телефона в готовом приложении.



3.mipmap- предрассчитанный, оптимизированный набор изображений связанных с одной текстурой и предназначенный для увеличения скорости рендеринга и улучшения качества изображения.

```
    ✓ ic_launcher (6)
    ic_launcher.webp (hdpi)
    ic_launcher.webp (mdpi)
    ic_launcher.webp (xhdpi)
    ic_launcher.webp (xxhdpi)
    ic_launcher.webp (xxxhdpi)
    ic_launcher.xml (anydpi-v26)
    ic_launcher_round (6)
    ic_launcher_round.webp (hdpi)
    ic_launcher_round.webp (mdpi)
    ic_launcher_round.webp (xhdpi)
    ic_launcher_round.webp (xxxhdpi)
    ic_launcher_round.webp (xxxhdpi)
    ic_launcher_round.webp (xxxhdpi)
    ic_launcher_round.xml (anydpi-v26)
```

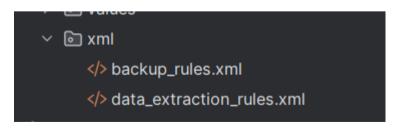
#### 4. Находится цвет текста и текст

```
  values
  </> colors.xml
  </> strings.xml
```

```
<resources>
    <string name="app_name">Красивые котлетки! \n6 различных цветов!</string>
    <string name="app_name2">Введите цвет котлетки</string>
</resources>
```



## 5.Хранение xml



#### Кнопка

1. Создал кнопку



2. Создал новую активность, создал VideoView для воспроизведения видео в активности

```
@ MainActivity2.kt
                                                               MainActivi
activity_main.xml
                   activity_main2.xml ×
    <?xml version="1.0" encoding="utf-8"?>
    <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="h</pre>
        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=".MainActivity2">
        <VideoView
             android:id="@+id/videoView"
             android:layout_width="1080dp"
             android:layout_height="800dp"
             android:layout_gravity="center"
             tools:ignore="MissingConstraints"
             tools:layout_editor_absoluteX="-505dp"
             tools:layout_editor_absoluteY="-12dp"
             />
    </androidx.constraintlayout.widget.ConstraintLayout>
```



3. В первой активности добавляем переход к активности 2

```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
            insets
        }
        val btn:Button=findViewById(R.id.btn_1)
        btn.setOnClickListener{
            val intent=Intent( packageContext: this, MainActivity2::class.java)
            startActivity(intent)
        }
    }
}
```

4. Теперь при нажатии кнопки перекидывает на вторую активность

# Практическая работа Intent

## Переход между Активностями

1. Передача с первой активности

```
val editText = findViewById<EditText>(R.id.edit)
val btn:Button = findViewById<Button>(R.id.btn_1)

btn.setOnClickListener{
   val intent = Intent( packageContext: this, MainActivity2::class.java)
   val enteredText = editText.text.toString().trim()
   intent.putExtra( name: "text",enteredText)
   startActivity(intent)
}
```

2. На вторую активность

```
val receivedText = intent.getStringExtra( name: "text")
val textView = findViewById<TextView>(R.id.textView2)
textView.text = receivedText
```

3.Результат



