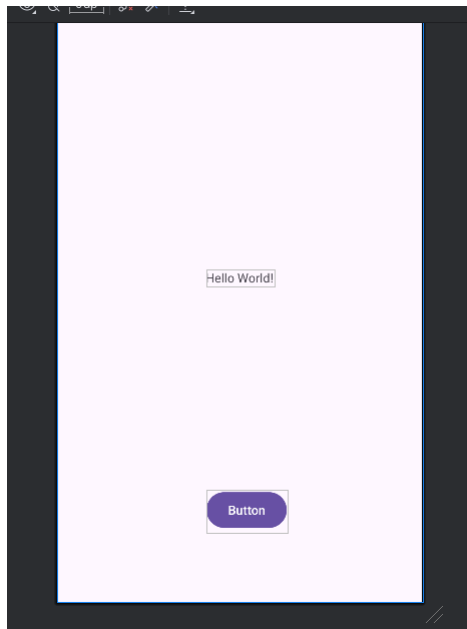


## Практическая работа 3

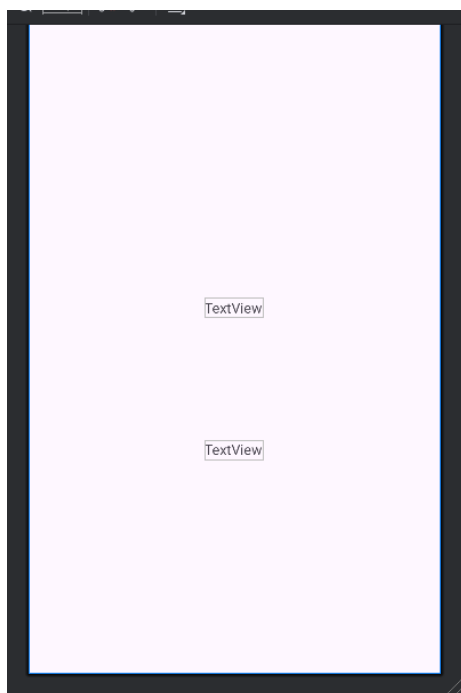
### Задание 1

Создаем 2 activity и оформляем их

act1



act2



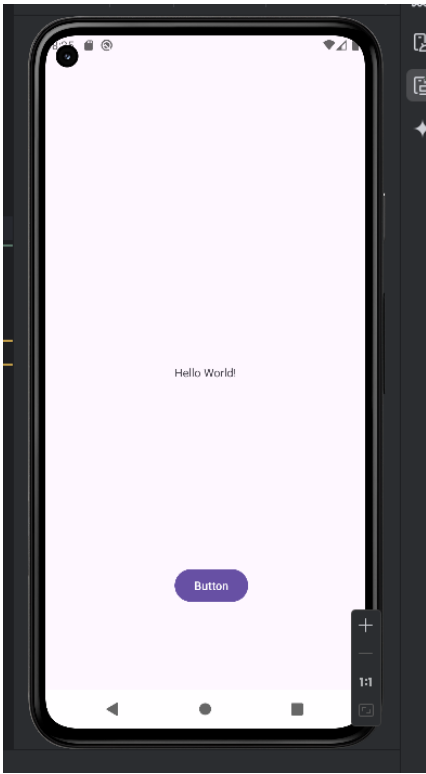
Прописываем код для переноса Extra значений с activity1 на activity2

```
}  
  
var bt:Button = findViewById(R.id.button)  
var txt:TextView = findViewById(R.id.textView)  
  
bt.setOnClickListener{  
    var intent = Intent(packageContext: this, MainActivity2::class.java)  
    intent.putExtra(name: "txt", value: "String проверка")  
    intent.putExtra(name: "num", value: 51)  
    startActivity(intent)  
}
```

Прописываем код для принятия Extra значений с activity1 на activity2

```
var txt1:TextView = findViewById(R.id.textView4)  
var txt2:TextView = findViewById(R.id.textView5)  
val message = intent.getIntExtra(name: "num", defaultValue: 0)  
val message1 = intent.getStringExtra(name: "txt")  
txt1.text = message.toString()  
txt2.text = message1
```

Вывод



## Задание 2 ViewBinding

Чтобы включить ViewBinding введем команду

```
buildFeatures{  
    viewBinding = true  
}
```

в build.gradle.kts

Добавляем / изменяем данные команды

```
binding = ActivityMainBinding.inflate(layoutInflater)
```

```
setContentView(binding.root)
```

Теперь через пропись binding.имя можно использовать значения кнопки и т.д.

Изменяем значения переменных на значения биндов

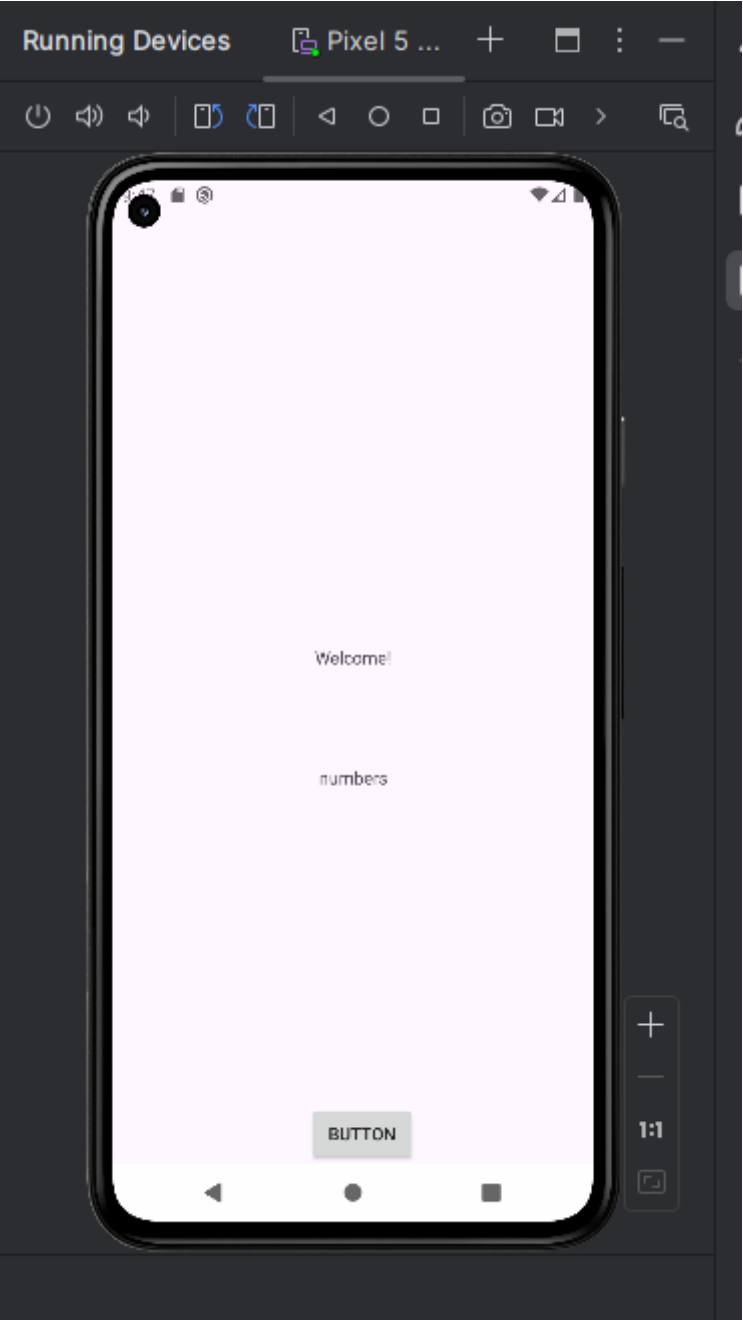
ACT1

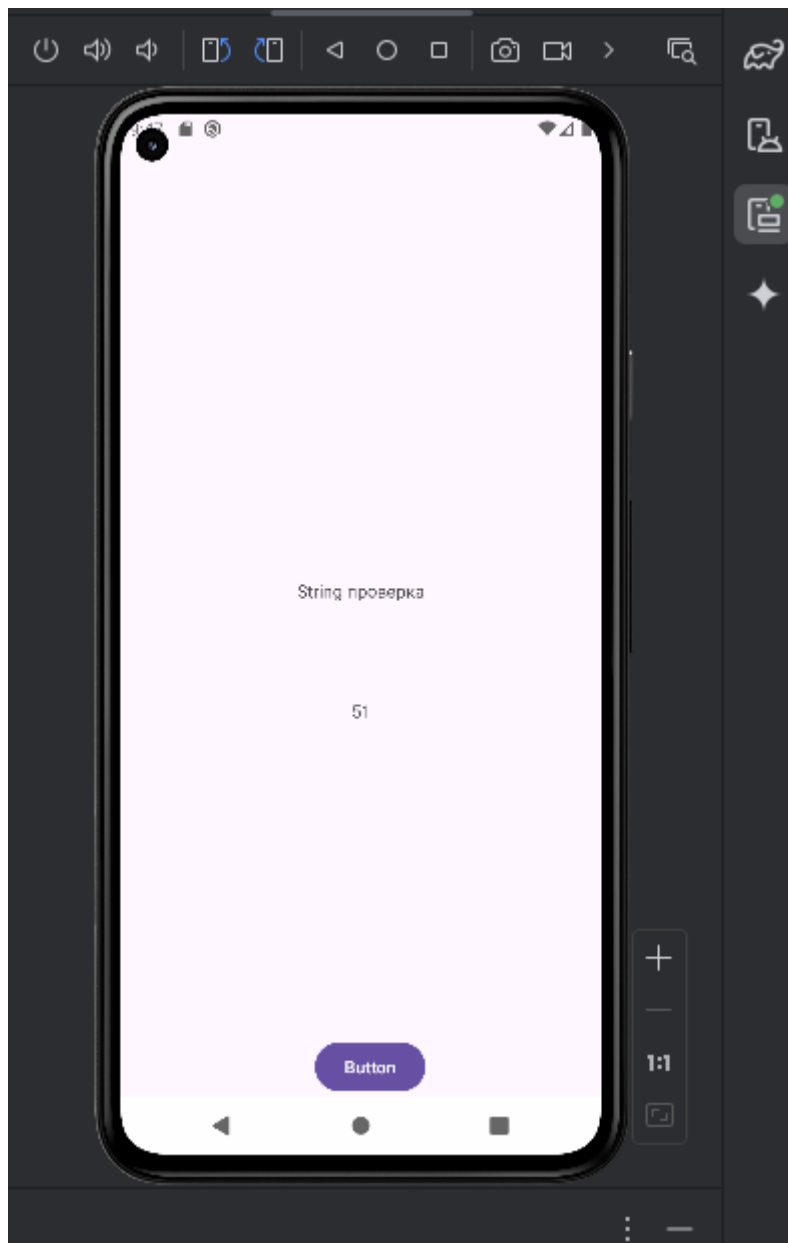
```
binding.textView2.text = "Welcome!"  
  
binding.button.setOnClickListener{  
    var intent = Intent( packageContext: this, MainActivity2::class.java)  
    intent.putExtra( name: "txt", value: "String проверка")  
    intent.putExtra( name: "num", value: 51)  
    startActivity(intent)  
}
```

ACT2

```
val message = intent.getIntExtra( name: "num", defaultValue: 0)  
val message1 = intent.getStringExtra( name: "txt")  
binding.textView3.text = message.toString()  
binding.textView2.text = message1
```

Результат





Ответ: Binding сокращает код, давая возможность обращаться к элементам Activity без объявления переменных, тем самым экономя ресурсы. Я считаю что binding лучше но сложнее в использовании(во время использования можно допустить много ошибок)