

Dice Roller

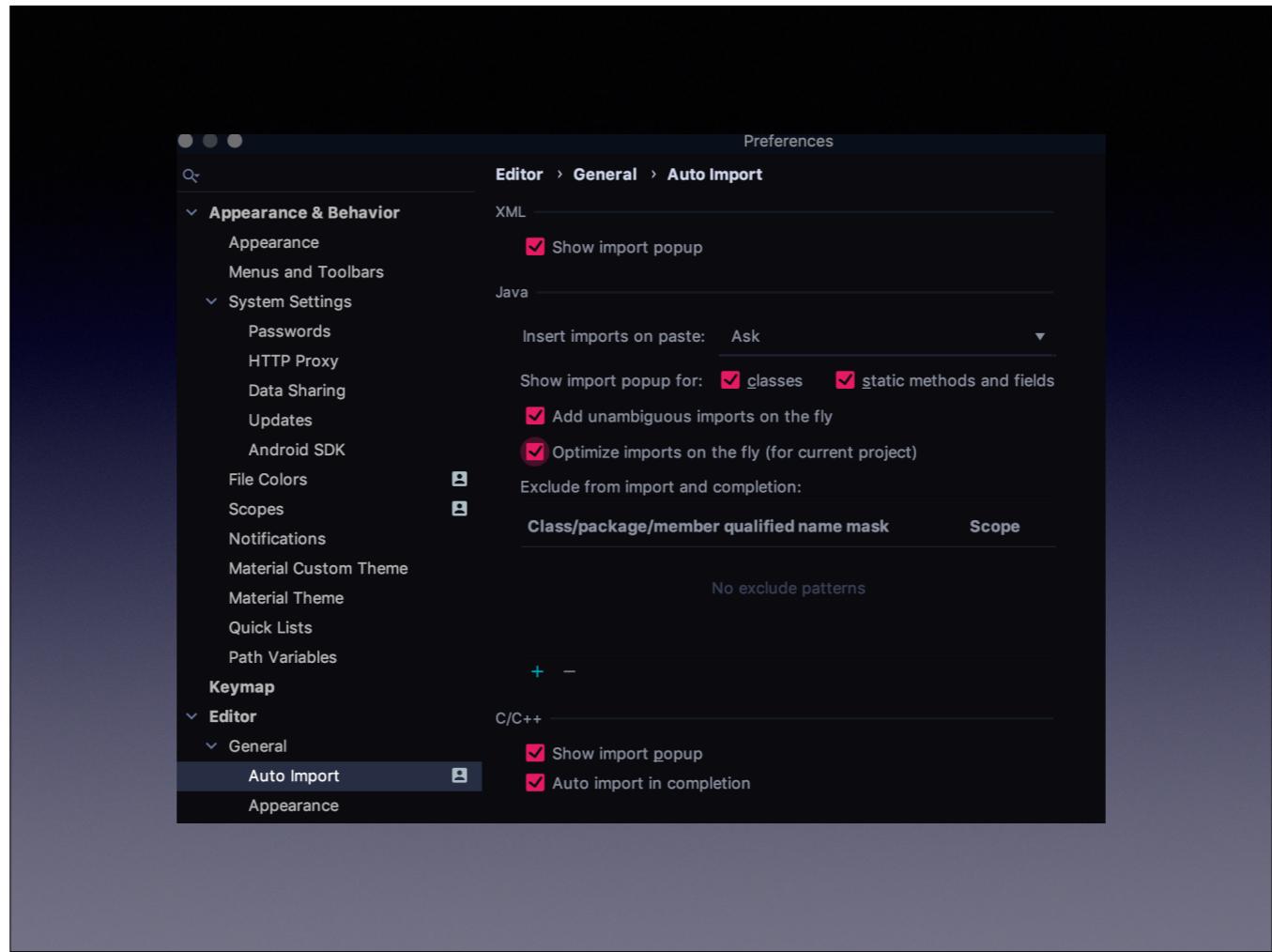
IDE - Android Studio

Objective

- * Learn the IDE
- * Build our first App
- * Dice Roller on two phases:
 - * Click a button will generate a number 1-6
 - * Click a button will show a dice with a number 1-6

Installation

- * <https://developer.android.com/studio/>
- * [https://developer.android.com/studio/
install](https://developer.android.com/studio/install)
- * [https://developer.android.com/studio/
#windows](https://developer.android.com/studio/#windows)
- * [Emulator](#)



IDE

- * Kotlin files for ur app logic
- * Res folder for static content
- * Manifest file that tell the OS how to run ur app
 - * Activities
- * Gradle build files

Let's Build Dice Roller

- 1.Add a button, set text to "roll"
- 2.Add a Text field
- 3.When button is clicked, Change text value to "1"
- 4.Replace the Text with a Dice Image

Step #1

activity_main.xml

Layout Inflation

MainActivity.kt

```
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center_vertical"  
    android:orientation="vertical">
```

LinearLayout

```
    <TextView  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:layout_gravity="center_horizontal"  
        android:textSize="30sp"  
        android:text="1" />
```

TextView

```
    <Button  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:layout_gravity="center_horizontal"  
        android:text="@string/roll" />
```

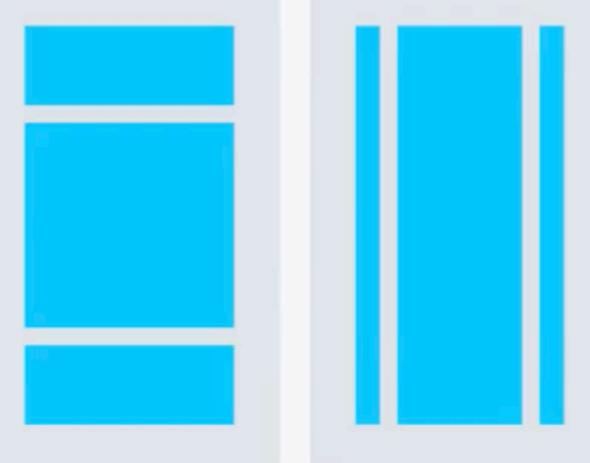
Button

```
</LinearLayout>
```

LinearLayout

Vertical
LinearLayout Horizontal
LinearLayout

- * Is a ViewGroup
- * A container of other views



activity_main.xml

```
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center_vertical"  
    android:orientation="vertical">  
  
    <TextView  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:layout_gravity="center_horizontal"  
        android:textSize="30sp"  
        android:text="1" />  
  
    <Button  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:layout_gravity="center_horizontal"  
        android:text="@string/roll" />  
  
</LinearLayout>
```

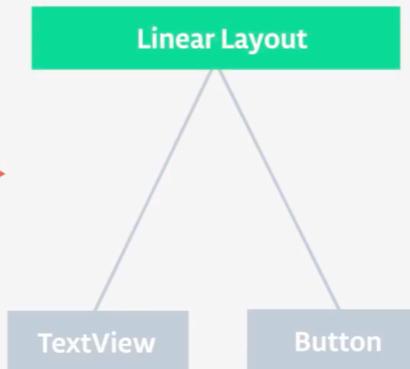
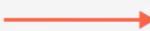
Layout Inflation

MainActivity.kt

Linear Layout

TextView

Button

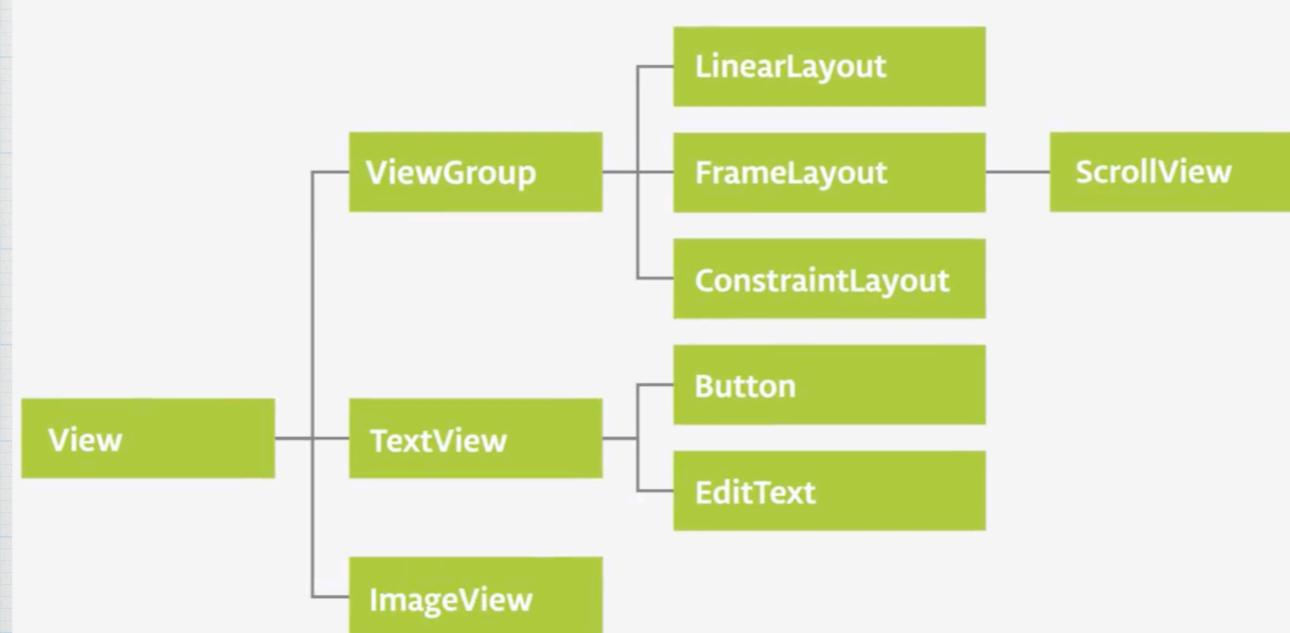


Scroll is Another View

* Scrollable view



Layout & Views



```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center_vertical"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:textSize="30sp"
        android:text="1" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:text="@string/roll" />

</LinearLayout>
```

XML eXtensible Markup Language

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

Note

To: Tove

From: Jani

Heading: Reminder

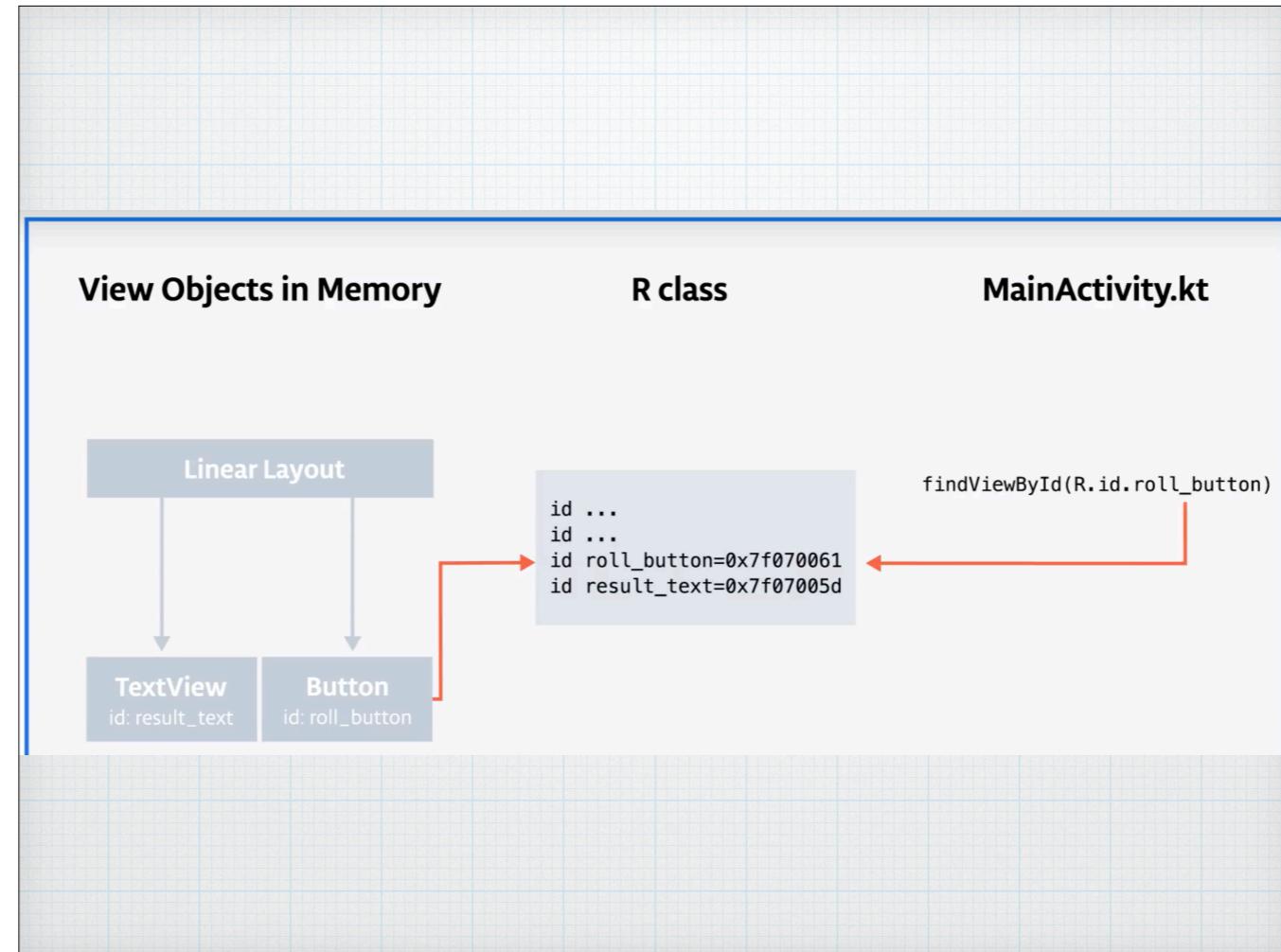
Body: Don't forget me this weekend!

* Example xmlexample.xml

Step #2, set the onClickListener

```
<Button  
    android:id="@+id/roll_button"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"/>
```

```
class MainActivity : AppCompatActivity() {  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        val rollButton: Button = findViewById(R.id.roll_button)  
        rollButton.setOnClickListener { it: View!  
            Toast.makeText( context: this, text: "button clicked", Toast.LENGTH_SHORT).show()  
        }  
    }  
}
```



Step #3, Roll the Dices

```
<TextView  
    android:id="@+id/result_text"  
    android:layout_width="wrap conte
```

```
        val rollButton: Button = findViewById(R.id.roll_button)  
        rollButton.setOnClickListener { it: View!  
            rollDice()  
        }  
  
    private fun rollDice() {  
        val randomInt = Random().nextInt( bound: 6 ) + 1  
  
        val resultText: TextView = findViewById(R.id.result_text)  
        resultText.text = randomInt.toString()  
    }
```

Step #3, Add the ImageView

- * Add the images to res/drawable
- * Replace TextView w/ ImageView

```
<ImageView  
    android:id="@+id/dice_image"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center_horizontal"  
    android:src="@drawable/empty_dice"  
    tools:src="@drawable/dice_1" />
```

Step #3, Add the ImageView...cont

* Update rollDice

```
private fun rollDice() {  
    val randomInt = Random().nextInt(bound: 6) + 1  
    val drawableResource = when (randomInt) {  
        1 -> R.drawable.dice_1  
        2 -> R.drawable.dice_2  
        3 -> R.drawable.dice_3  
        4 -> R.drawable.dice_4  
        5 -> R.drawable.dice_5  
        else -> R.drawable.dice_6  
    }  
  
    val diceImage: ImageView = findViewById(R.id.dice_image)  
    diceImage.setImageResource(drawableResource)  
}
```

More On Views

- * Layouts
- * ViewGroup
- * View
- * LinearLayout
- * ScrollView