NTNU | Norwegian University of Science and Technology

PROG 2007- Mobile Programming

Android Studio Applications with Kotlin

View binding

- View binding makes it easier to reference Views created in xml
- it generates a *binding class* for each XML layout file present in your project
- instance of a binding class contains direct references to all views that have an ID in the corresponding layout.
- To enable view binding set the viewBinding build option to true in the module-level build.gradle file
- More information: https://developer.android.com/topic/libraries/view-binding

```
android {
    …
    buildFeatures {
        viewBinding = true
    }
}
```

View binding - usage

- Add and id to all the XML elements you want to access in code

```
<TextView
    android:id="@+id/textViewWeight"
    … />
```

- Import automatically created binding class in Kotlin file

```
import com.example.bmi_calculator.databinding.ActivityMainBinding
```

- Inflate binding

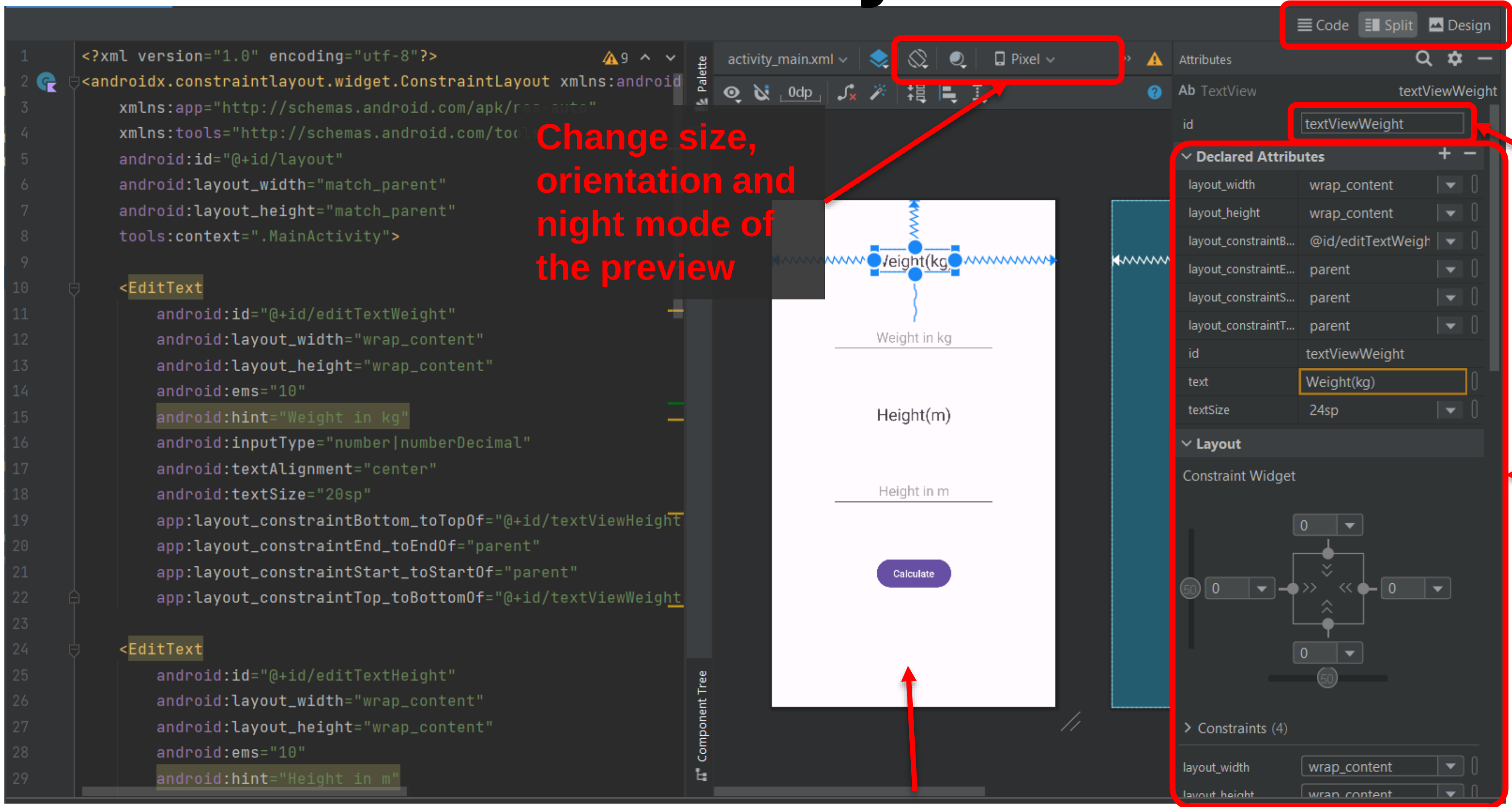
```
binding = ActivityMainBinding.inflate(layoutInflater)
setContentView(binding.root)
```

- Easily access elements by their id

```
binding.textViewWeight.text = “Please enter your Weight“
val height = binding.editTextHeight.text.toString().toFloat()
```

Android Studios Layout Editor

edit the file in code or design view



```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/too
    android:id="@+id/layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/editTextWeight"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Weight in kg"
        android:inputType="number|numberDecimal"
        android:textAlignment="center"
        android:textSize="20sp"
        app:layout_constraintBottom_toTopOf="@+id/textViewHeight
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textViewWeight

    <EditText
        android:id="@+id/editTextHeight"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Height in m"
```

Ab TextView — textViewWeight

id: textViewWeight

Declared Attributes

layout_width	wrap_content
layout_height	wrap_content
layout_constraintB...	@id/editTextWeigh
layout_constraintE...	parent
layout_constraintS...	parent
layout_constraintT...	parent
id	textViewWeight
text	Weight(kg)
textSize	24sp

Layout

Constraint Widget

Constraints (4)

layout_width	wrap_content
layout_height	wrap_content

Change size, orientation and night mode of the preview

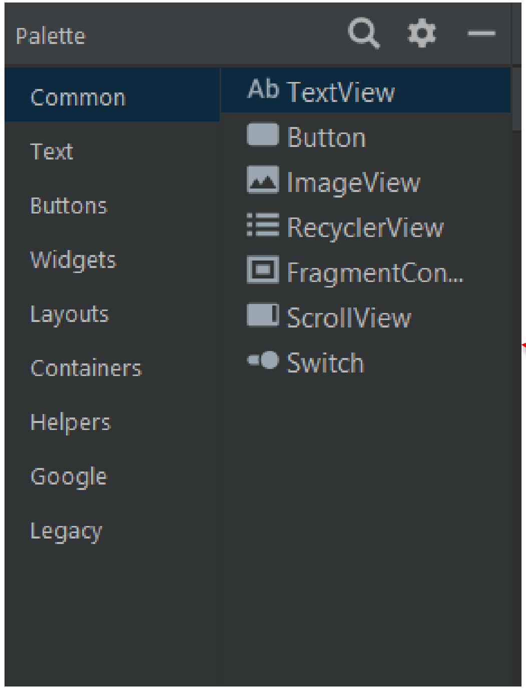
Set the id of the current view

Change attributes like color, padding, margin gravity…

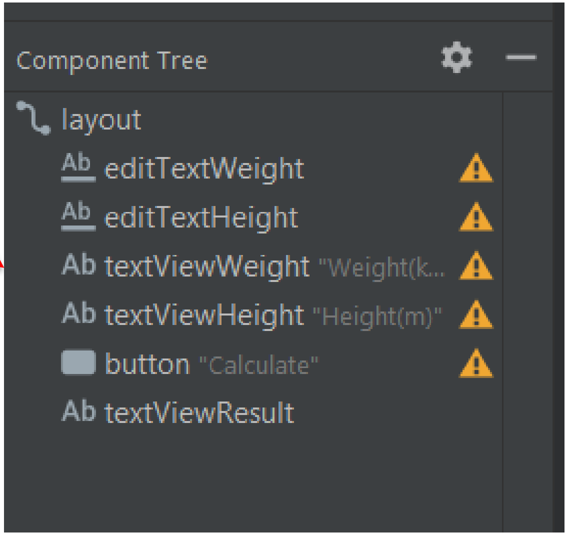
Use drag and drop functionality

NTNU | Norwegian University of Science and Technology

Android Studios Layout Editor



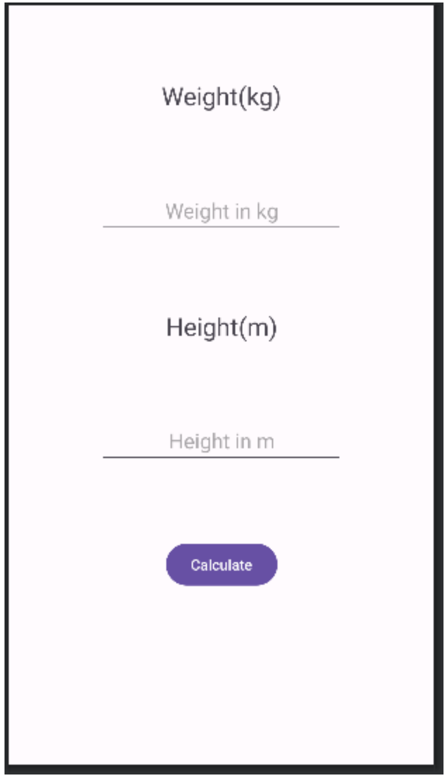
Overview of the Layout structure and possible problems

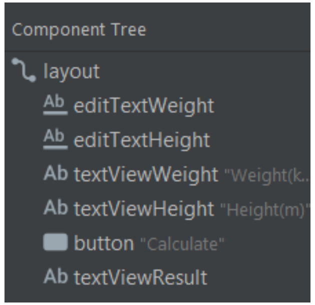


Choose different views to build the UI

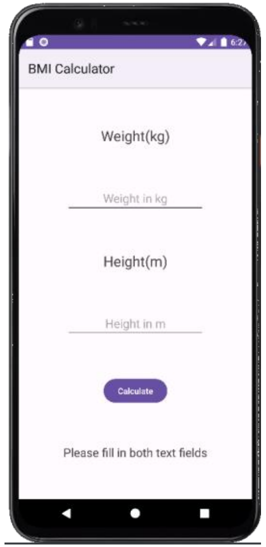
- Dp: density-independent pixels
- 1dp = 0.15875 mm
- Do not use absolute values for positions

BMI Calculator App





Component Tree
- layout
  - editTextWeight
  - editTextHeight
  - textViewWeight "Weight(k..
  - textViewHeight "Height(m)"
  - button "Calculate"
  - textViewResult



Please fill in both text fields

NTNU | Norwegian University of Science and Technology

```kotlin
package com.example.bmi_calculator

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import com.example.bmi_calculator.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {
   lateinit var binding: ActivityMainBinding

   override fun onCreate(savedInstanceState: Bundle?) {
      super.onCreate(savedInstanceState)

      binding = ActivityMainBinding.inflate(layoutInflater)
      setContentView(binding.root)

      binding.button.setOnClickListener {
         try {
            val weight = binding.editTextWeight.text.toString().toFloat()
            val height = binding.editTextHeight.text.toString().toFloat()

            if (weight > 0 && height > 0) {
               val bmi = weight / (height * height)
               binding.textViewResult.text = "Your BMI is $bmi"
            } else {
               binding.textViewResult.text =
                  "Your weight and height can't be zero"
            }
         }catch(e: NumberFormatException){
            binding.textViewResult.text = "Please fill in both text fields"
         }
      }
   }
}
```

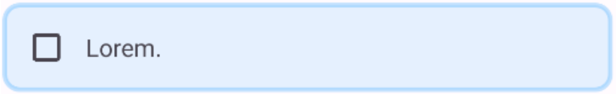
ToDo List App – RecyclerView

- Use RecyclerView to create an efficient scrollable list
- To add content to a recycler view add an adapter and a layout
- The layout can also be specified in XML
- You can set up an own layout file for the single list elements
- The adapter class manages to pass the data of the currently visible elements to the views

ToDo List Adapter

```kotlin
package com.example.todolist

import android.view.LayoutInflater
import android.view.ViewGroup
import androidx.recyclerview.widget.RecyclerView
import com.example.todolist.databinding.TodoItemBinding

class TodoListAdapter(private val list: List<String>): RecyclerView.Adapter<TodoListAdapter.ToDoViewHolder>()
{
   //Class for managing a single list item
   class ToDoViewHolder(private var binding: TodoItemBinding): RecyclerView.ViewHolder(binding.root) {
      fun onBind(text: String){
         //fill UI elements with data, add click listeners
         binding.textView.text = text
      }
   }

   override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): TodoListAdapter.ToDoViewHolder {
      val binding = TodoItemBinding.inflate(LayoutInflater.from(parent.context), parent, false)
      return ToDoViewHolder(binding)
   }

   override fun onBindViewHolder(holder: TodoListAdapter.ToDoViewHolder, position: Int) {
      //Pass data at list position to the ViewHolder
      holder.onBind(list[position])
   }

   override fun getItemCount(): Int {
      return list.size
   }
}
```

Todo_item layout



```xml
<?xml version="1.0" encoding="utf-8"?>
<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_margin="5dp"
   android:background="@drawable/background_item"
   android:orientation="horizontal"
   android:layout_gravity="center"
   android:padding="5dp">

   <CheckBox
      android:id="@+id/checkBox"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_gravity="center_vertical" />

   <TextView
      android:id="@+id/textView"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_gravity="center_vertical"
      android:padding="2dp"
      android:textSize="16sp"
      tools:text="@tools:sample/lorem" />
</LinearLayout>
```

Use tools:text to generate dummy text in the preview

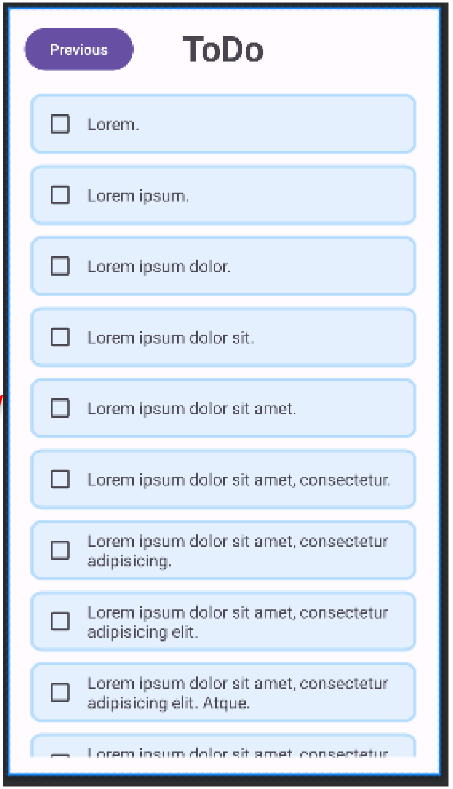
ToDo List App – RecyclerView

```
//Instantiate adapter and add it to recycler view
val adapter = TodoListAdapter(todoList)
binding.recyclerView.adapter = adapter
//Use this if you haven't set up an a layout manager in xml
val linearLayoutManager = LinearLayoutManager(context)
linearLayoutManager.orientation = LinearLayoutManager.VERTICAL
binding.recyclerView.layoutManager = linearLayoutManager
```

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
  …>
   <Button
     … />
   <androidx.recyclerview.widget.RecyclerView
      android:id="@+id/recyclerView"
      android:layout_width="0dp"
      android:layout_height="0dp"
      android:layout_marginTop="16dp"
      app:layoutManager="androidx.recyclerview.widget.LinearLayoutManager"
      tools:listitem="@layout/todo_item"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toBottomOf="@+id/textView2" />
   <TextView
     …/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Set up a layout manager

Use tools:listitem to generate a preview in the Layout editor



- [ ] Lorem.
- [ ] Lorem ipsum.
- [ ] Lorem ipsum dolor.
- [ ] Lorem ipsum dolor sit.
- [ ] Lorem ipsum dolor sit amet.
- [ ] Lorem ipsum dolor sit amet, consectetur.
- [ ] Lorem ipsum dolor sit amet, consectetur adipisicing.
- [ ] Lorem ipsum dolor sit amet, consectetur adipisicing elit.
- [ ] Lorem ipsum dolor sit amet, consectetur adipisicing elit. Atque.
- [ ] Lorem ipsum dolor sit amet, consectetur

NTNU | Norwegian University of Science and Technology

Links

- View bindings: https://developer.android.com/topic/libraries/view-binding
- Recycler view: https://developer.android.com/develop/ui/views/layout/recyclerview
- Constraint Layout https://developer.android.com/develop/ui/views/layout/constraint-layout