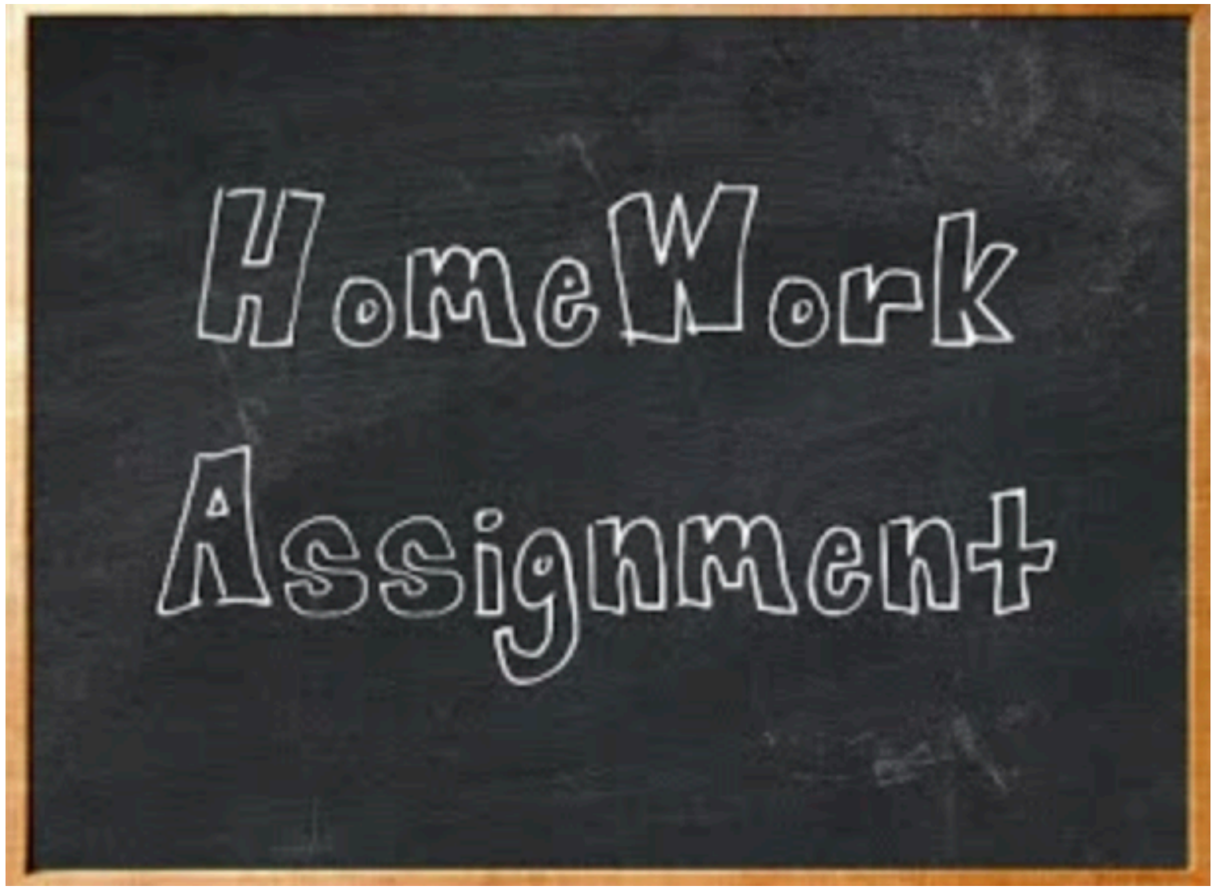
Lecture #2
Lists and Rest Resources

Mobile Applications 2019-2020

Homework assignments

- First assignment - project details **Due: 2nd laboratory**
- Two projects
  - CRUD Application (in two flavors) **Due: last laboratory before holiday**
  - Bonuses

Due: last laboratory



CRUD Application

Native Java OBJ-C

AND

Non-Native

ionic

Multi platform

Other

+1p Published on
Google Store

+1p Animations

+1p Authentication

+1p Published on
Play Store

+1p Bug
tracking

Native

+1p Integrated
ads

+1p At least a unit test

+2p Cloud Persistence

+1p Uses
Material Design

Non-Native

+1p Multiple User Roles

+1p InApp
Purchases

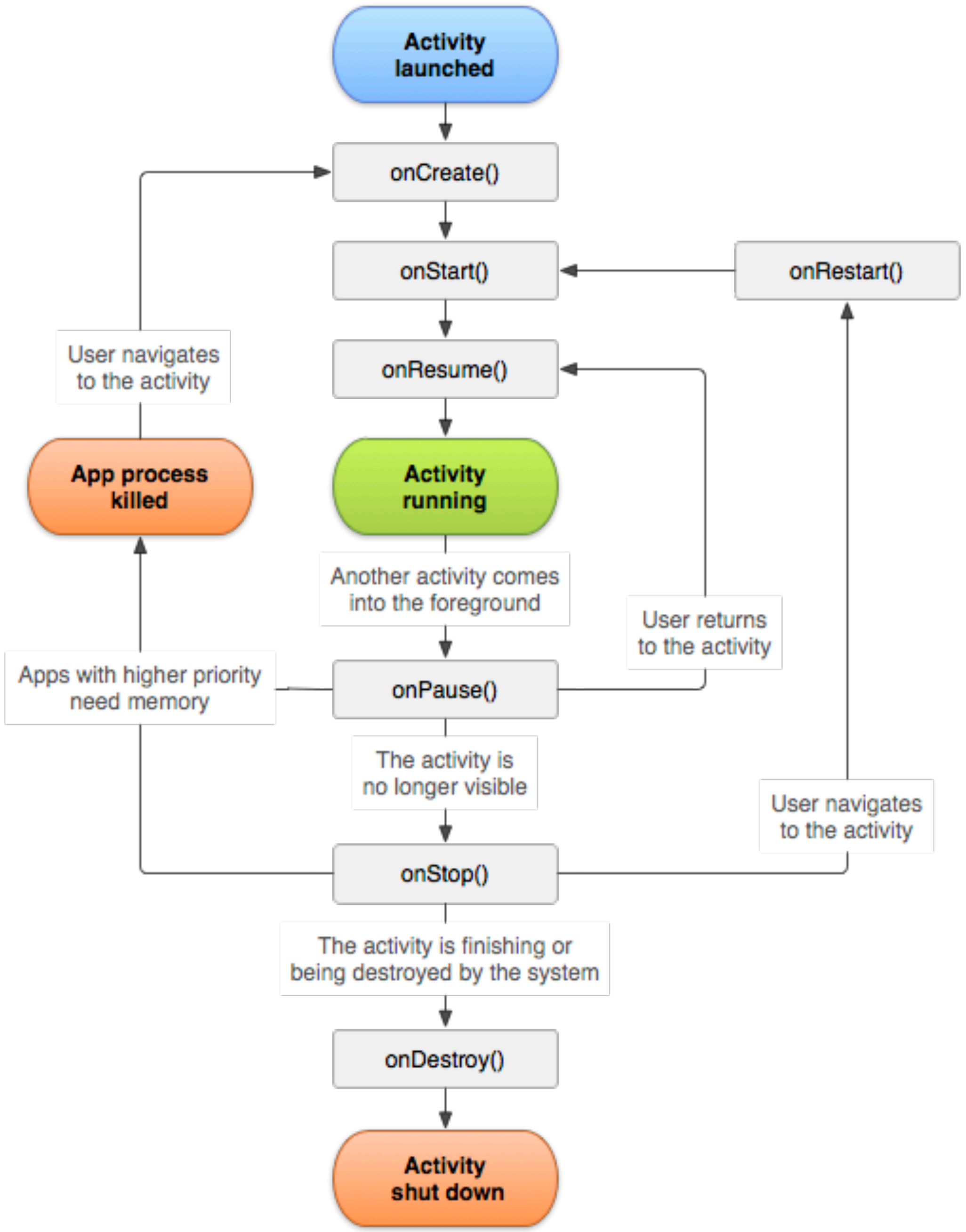
+1p Uses
Rx{Java|Swift|Kotlin}

+2p Has more than
50 downloads
on the store

https://www.cs.ubbcluj.ro/~dan/ma/

DEMO

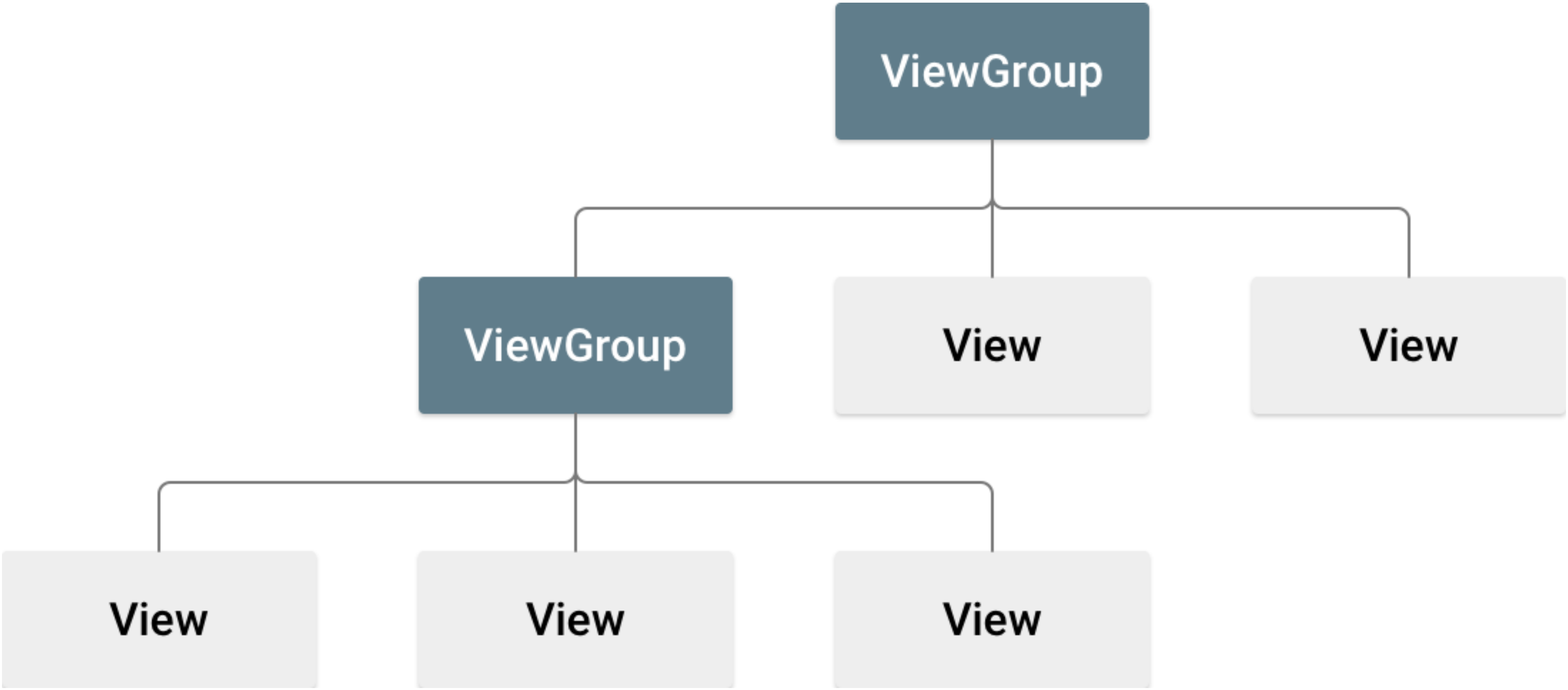
Lifecycle


Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

https://developer.android.com/guide/components/activities/activity-lifecycle

Layouts

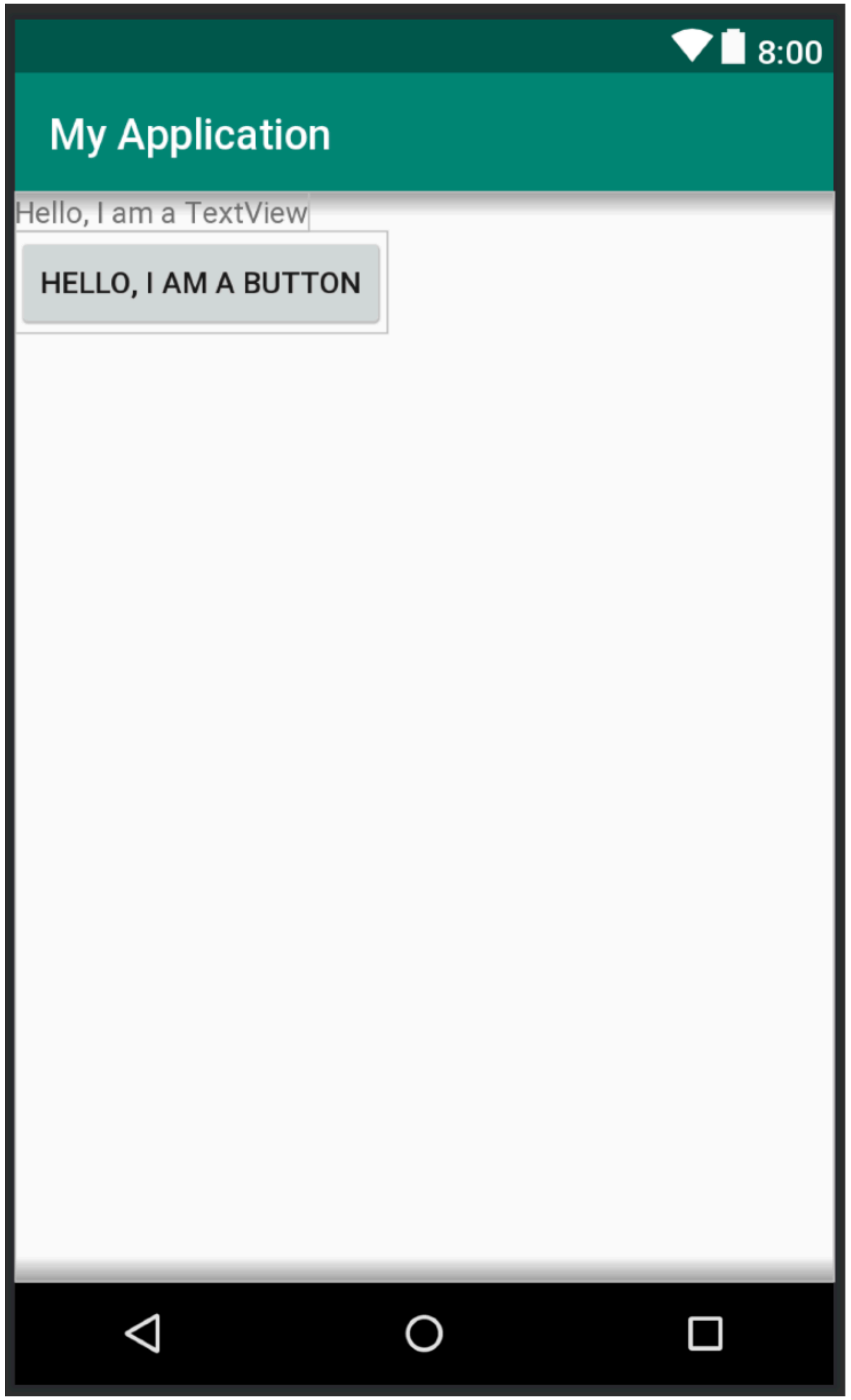
https://developer.android.com/guide/topics/ui/declaring-layout



XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://
schemas.android.com/apk/res/android"
              android:layout_width="match_parent"
              android:layout_height="match_parent"
              android:orientation="vertical">
    <TextView android:id="@+id/text"
              android:layout_width="wrap_content"
              android:layout_height="wrap_content"
              android:text="Hello, I am a
TextView"/>
    <Button android:id="@+id/button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hello, I am a Button"/>
</LinearLayout>
```

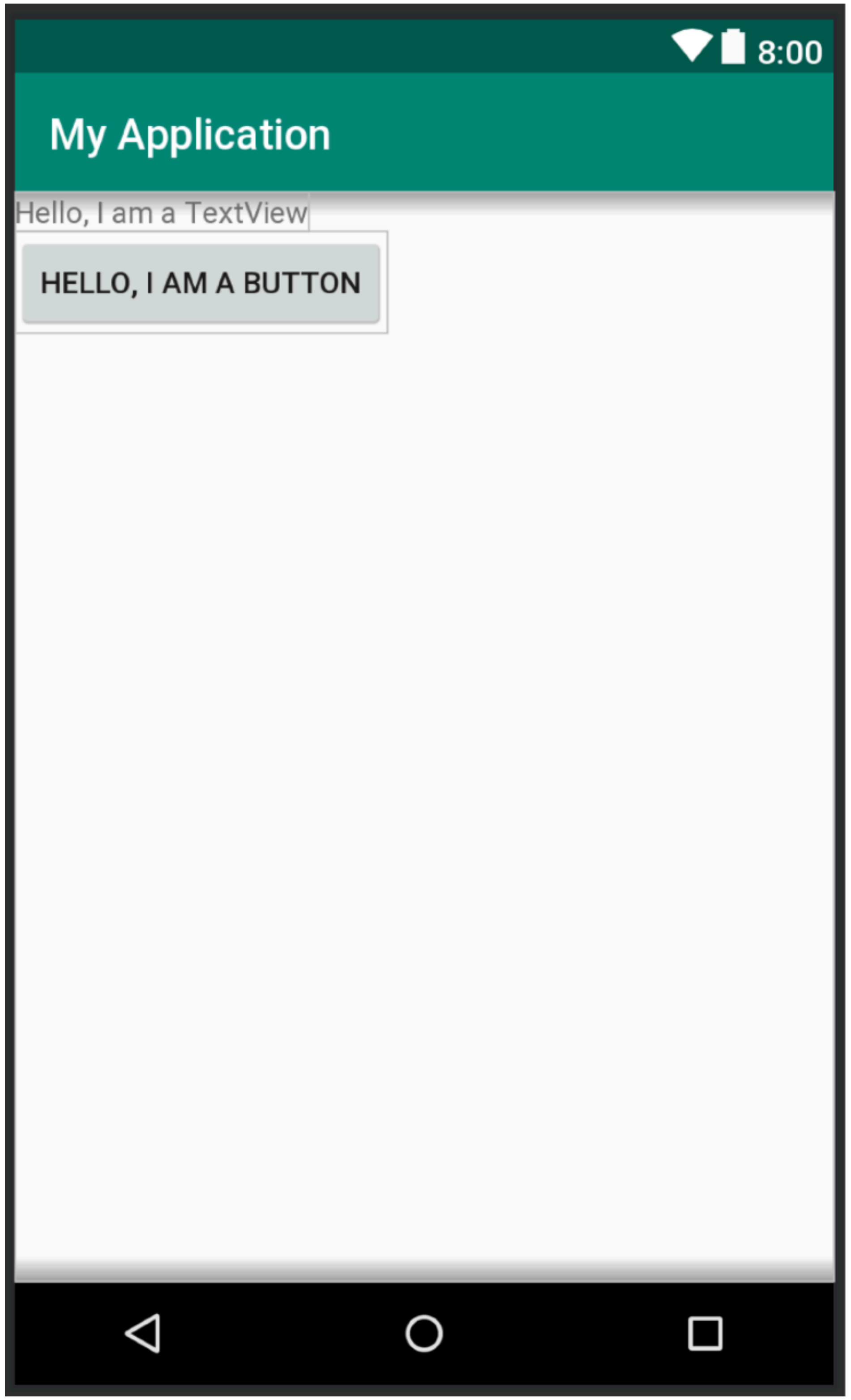
```
fun onCreate(savedInstanceState: Bundle) {
     super.onCreate(savedInstanceState)
     setContentView(R.layout.main_layout)
}
```



Accessing Assets

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://
schemas.android.com/apk/res/android"
              android:layout_width="match_parent"
              android:layout_height="match_parent"
              android:orientation="vertical">
    <TextView android:id="@+id/text"
              android:layout_width="wrap_content"
              android:layout_height="wrap_content"
              android:text="Hello, I am a TextView"/>
    <Button android:id="@+id/button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hello, I am a Button"/>
</LinearLayout>
```

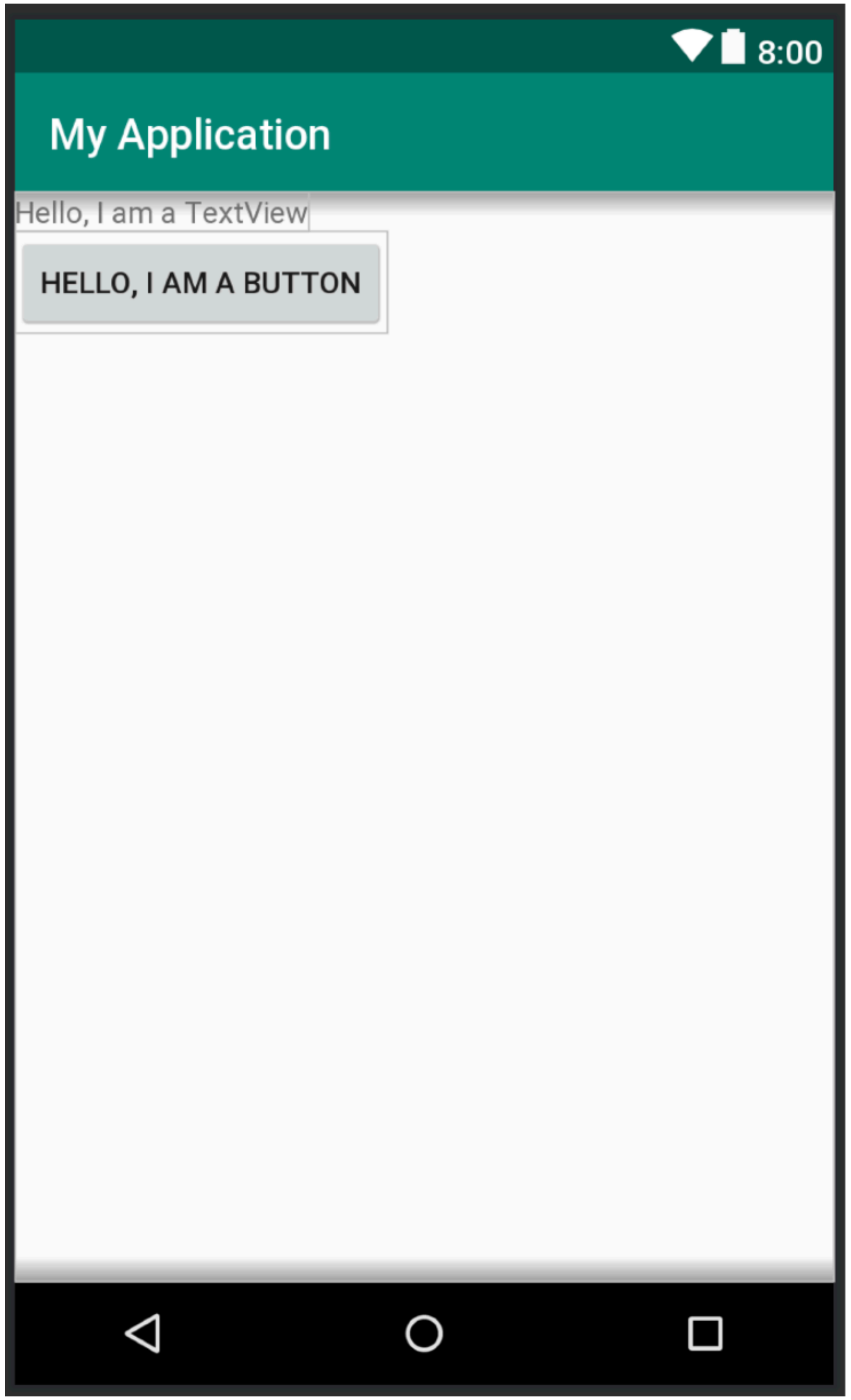
```
fun onCreate(savedInstanceState: Bundle) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.main_layout)
    val myButton: Button = findViewById(R.id.button)
}
```



Add Event Handler

```
...
    <Button android:id="@+id/button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hello, I am a Button"
            android:onClick="sendMessage"
...
```

```
fun onCreate(savedInstanceState: Bundle) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.main_layout)
    val myButton: Button = findViewById(R.id.button)
}
fun sendMessage(view: View) {
   logd("Ready!")
}
```

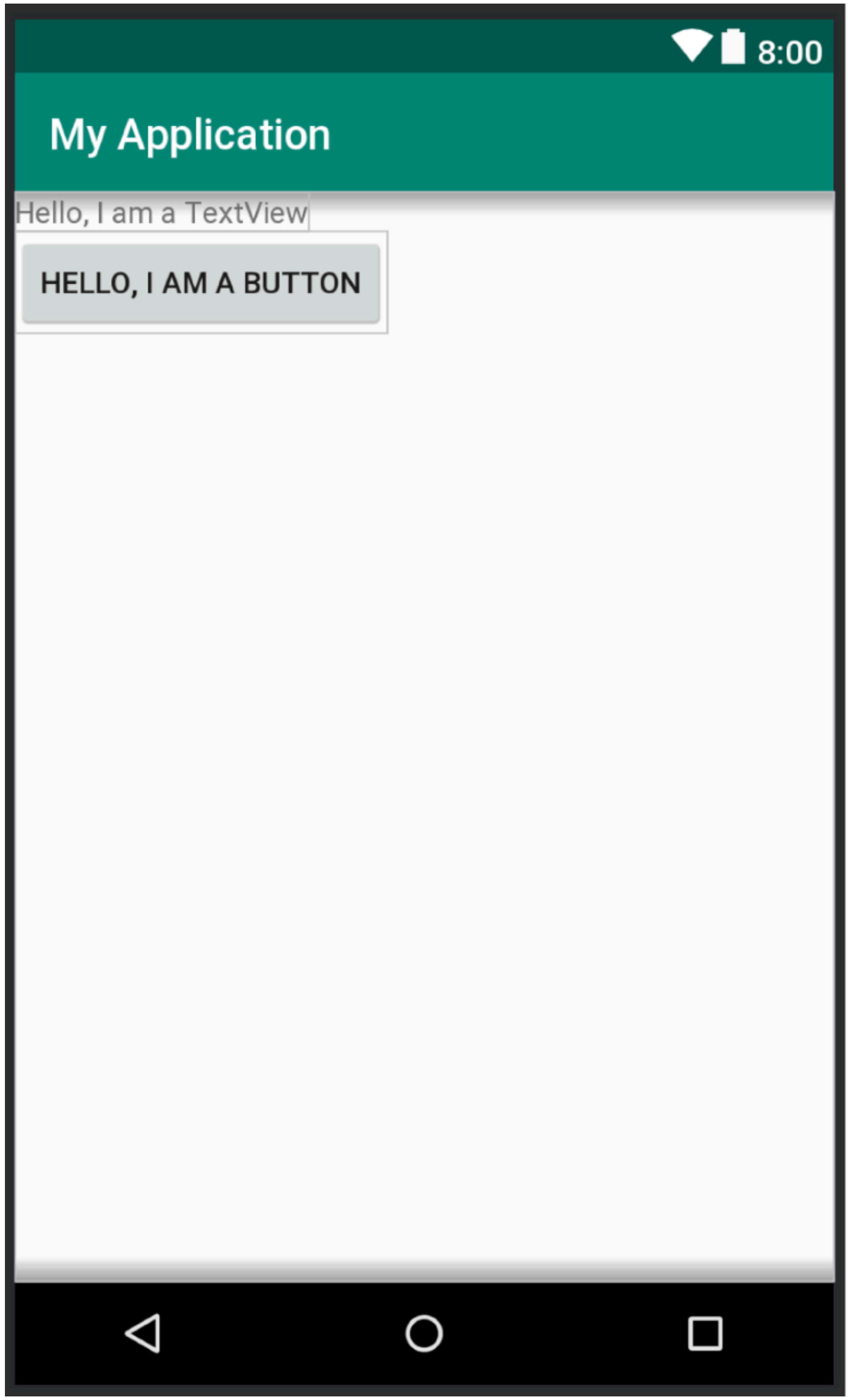


DEMO

Using Android KTX

```
import kotlinx.android.synthetic.main.activity_main.*
...
    <Button android:id="@+id/button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hello, I am a Button"
            android:onClick="sendMessage"
   />
...
```

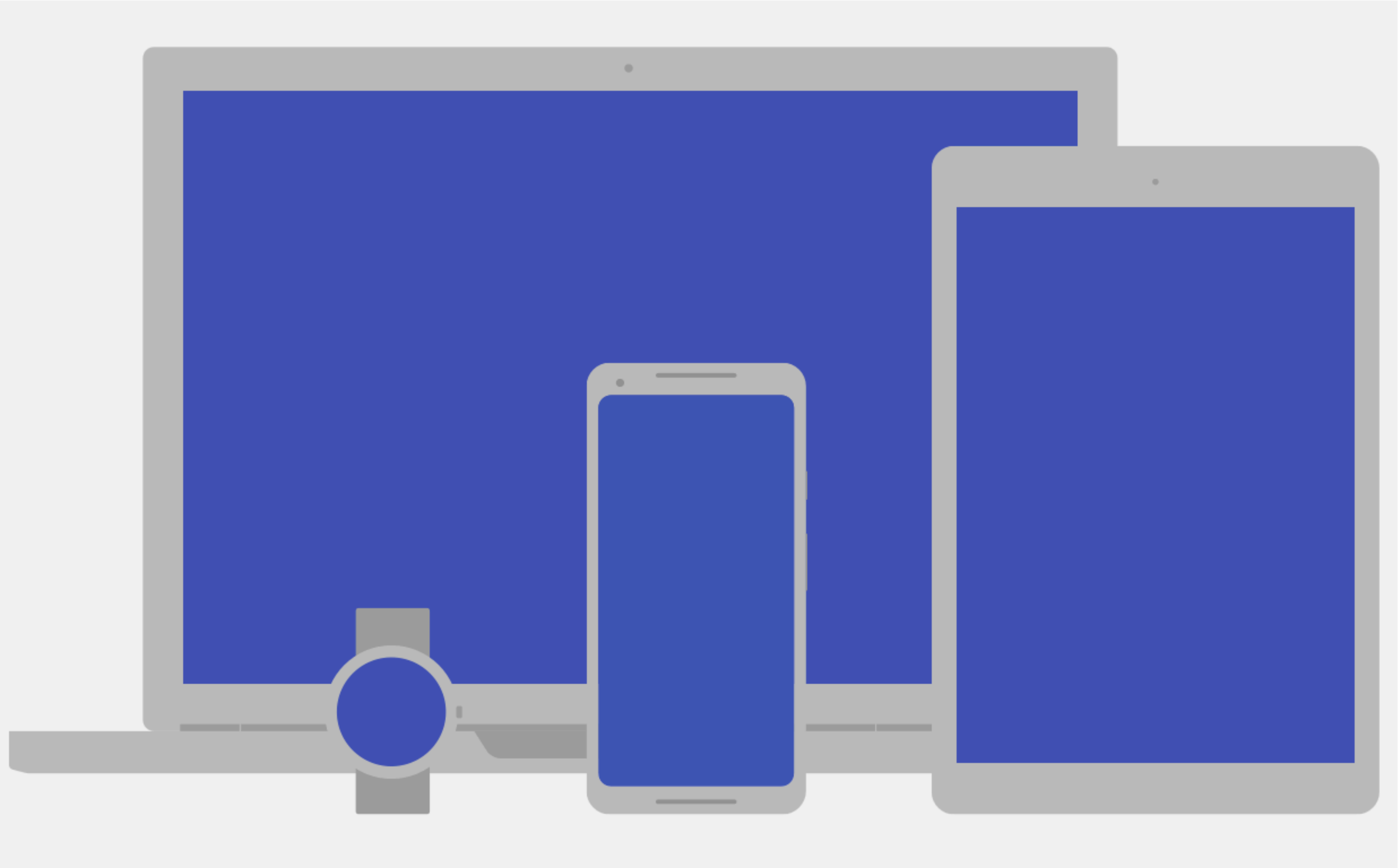
```
fun onCreate(savedInstanceState: Bundle) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.main_layout)
    button.setOnClickListener {
      text.text = "From editText: ${editText.text.toString()}"
      button.text = "Update"
    }
}
```



https://developer.android.com/kotlin/ktx

Supporting different screen sizes

- Flexible layouts
- Alternative layouts
- Stretchable images
- Alternative bitmaps
- Vector graphics



https://developer.android.com/guide/practices/screens_support

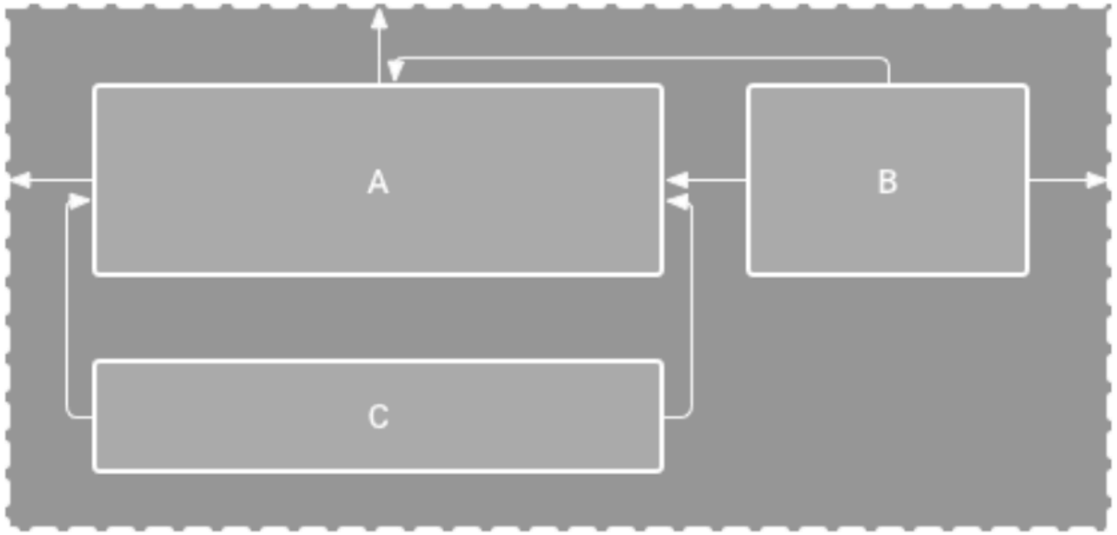
DEMO

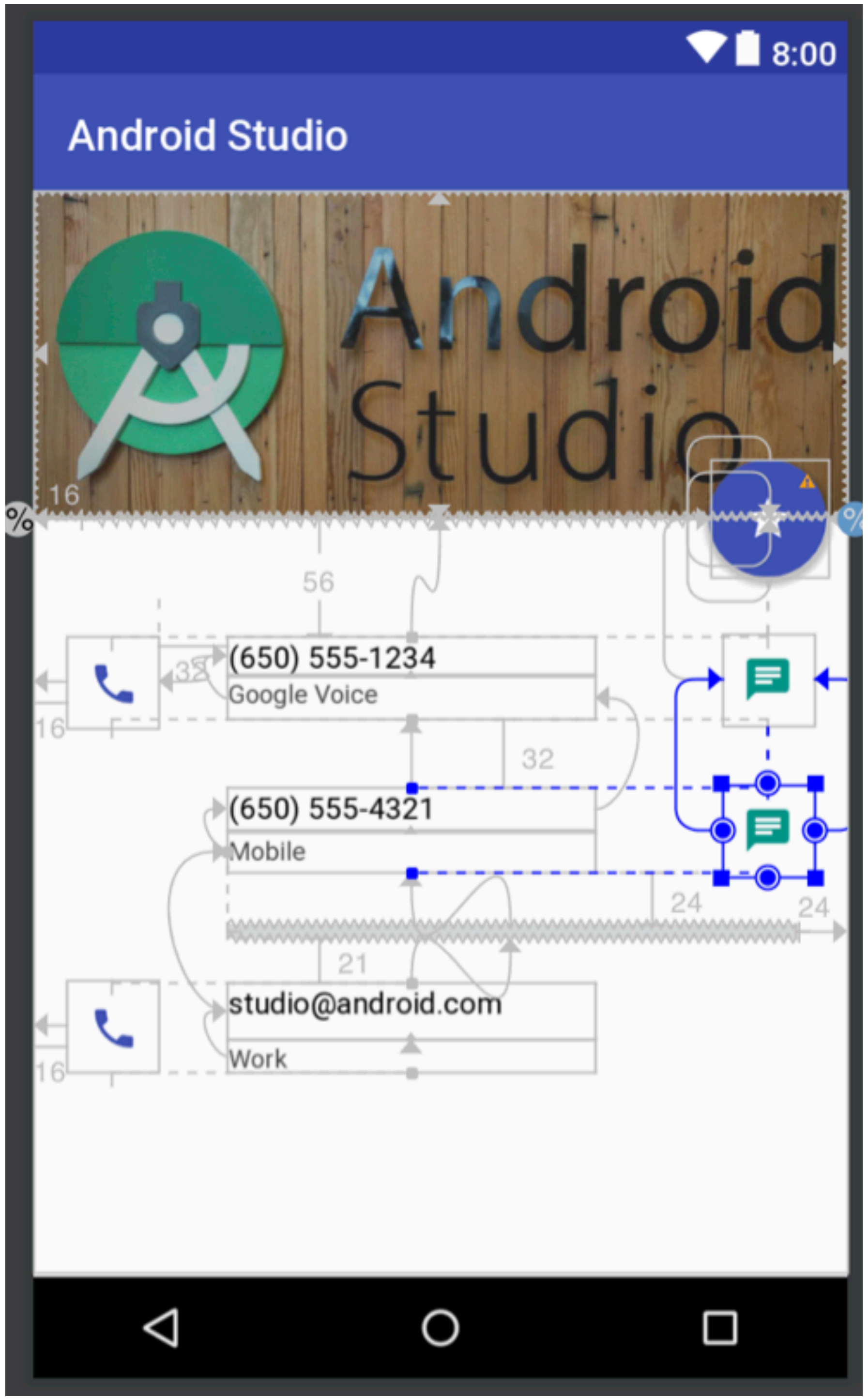
Flexible Layouts

ConstraintLayout

In module-level **gradle.build:**

```
repositories {
    google()
}

dependencies {
 implementation
 'com.android.support.constraint:constraint-layout:1.1.2'
}
```



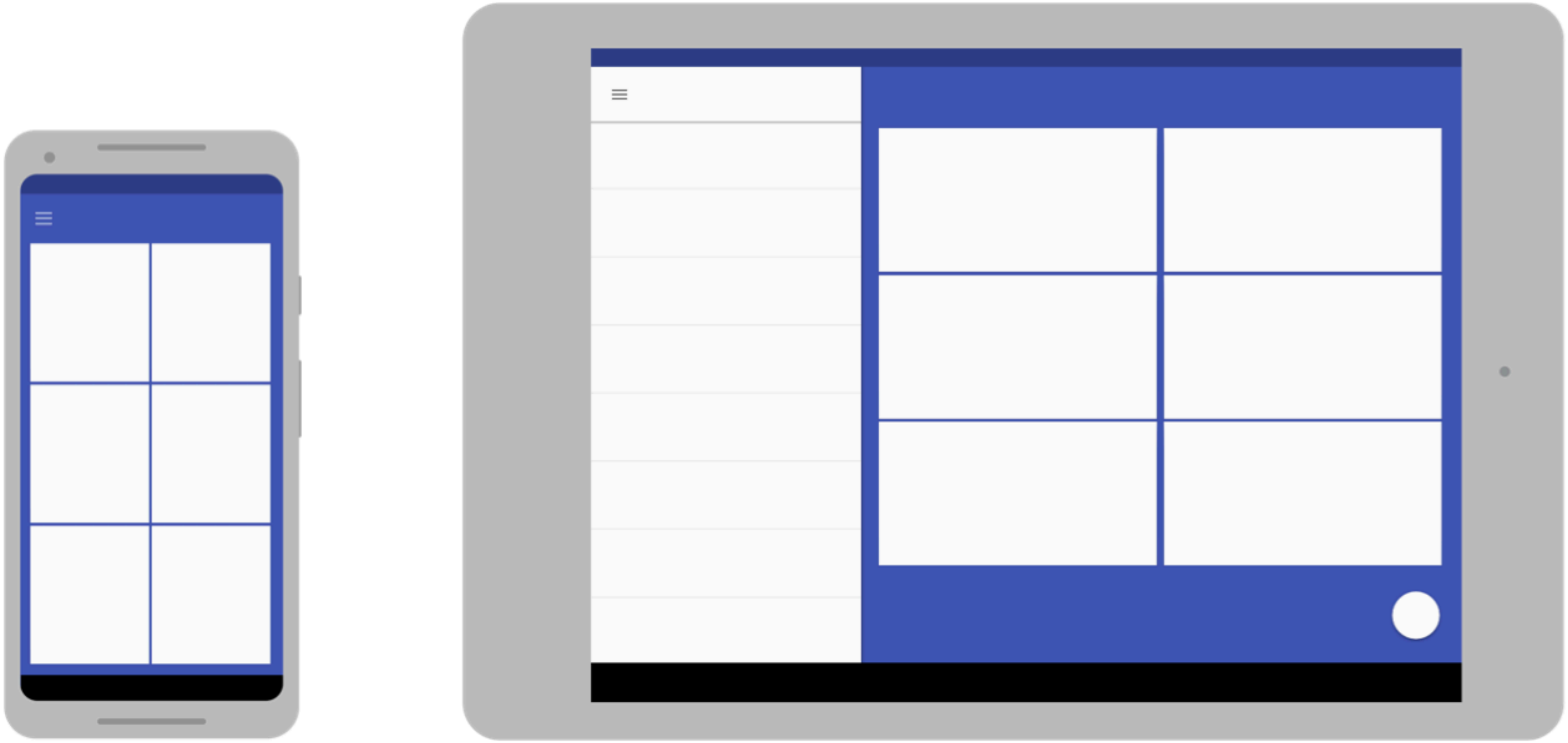


https://developer.android.com/training/constraint-layout/

DEMO

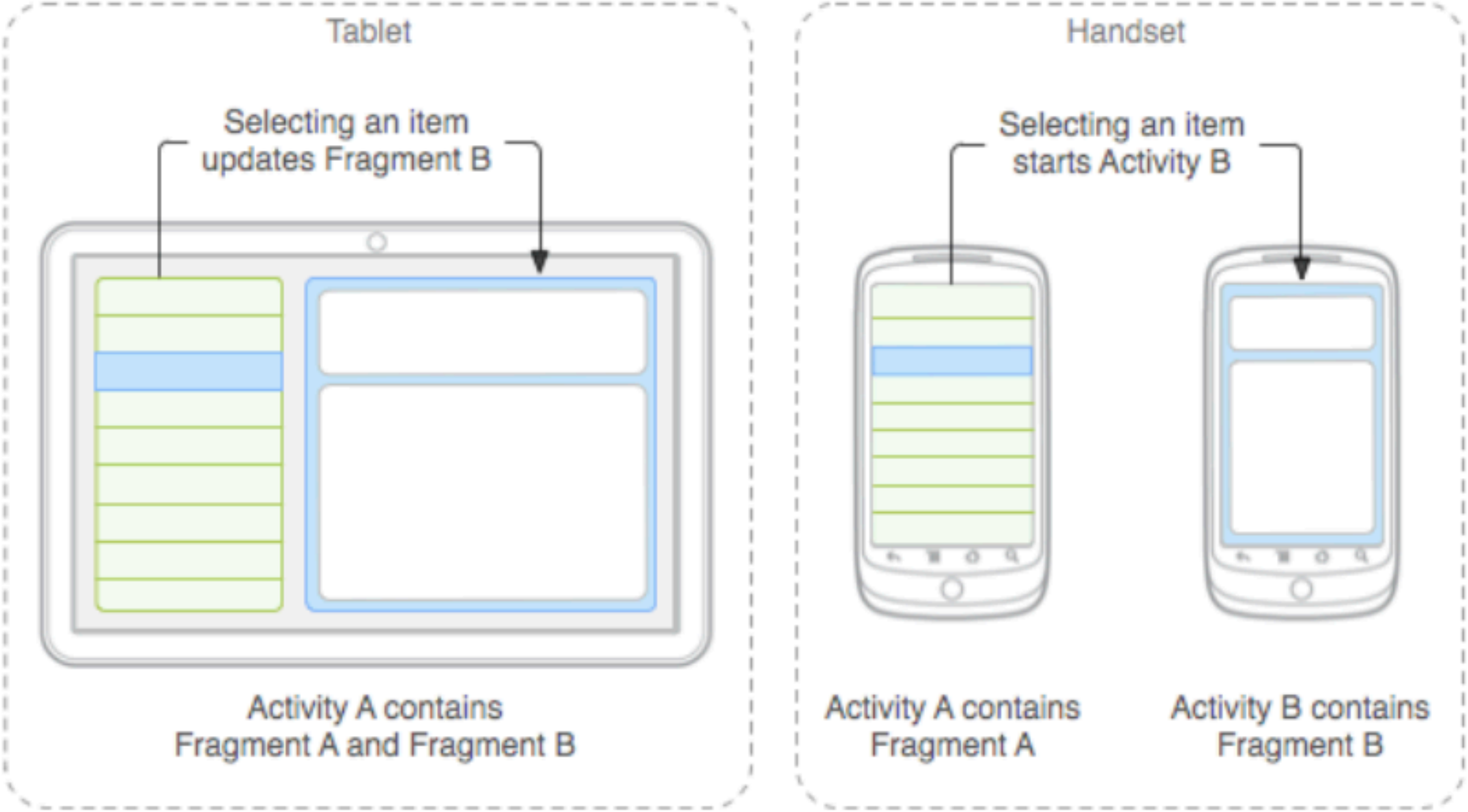
Alternative layouts

```
res/layout/main_activity.xml                # Default layout
res/layout-land/main_activity.xml           # When in landscape mode
res/layout-sw600dp/main_activity.xml        # For 7″ tablets
res/layout-sw600dp-land/main_activity.xml   # For 7″ tablets in landscape
```



https://developer.android.com/training/multiscreen/screensizes#alternative-layouts

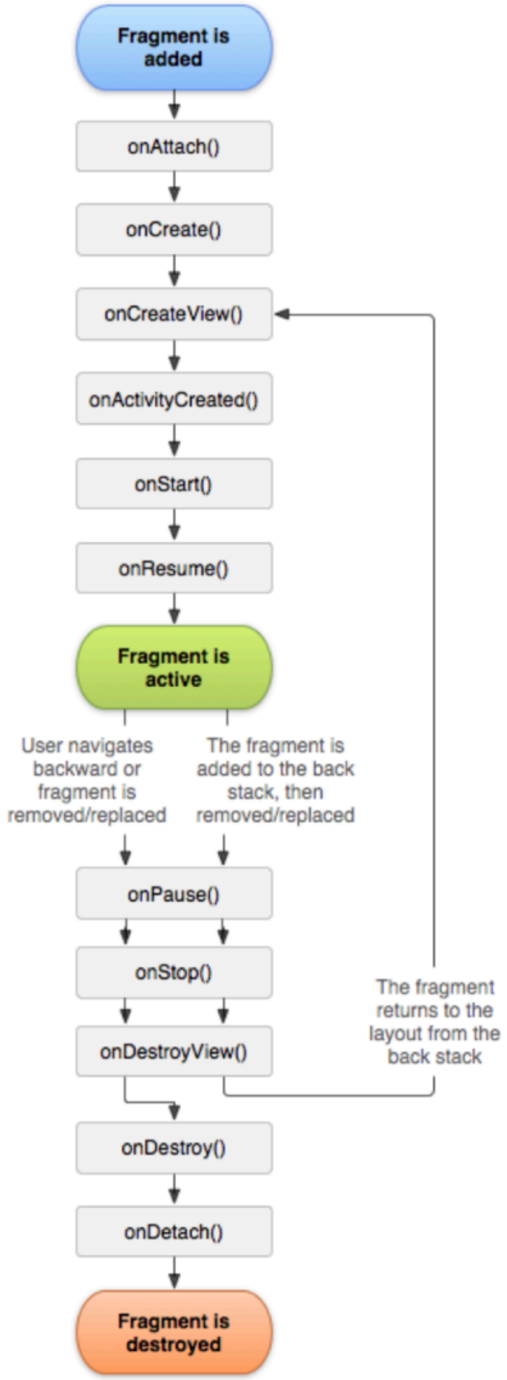
Building a Dynamic UI with Fragments


Tablet
Selecting an item updates Fragment B
Activity A contains Fragment A and Fragment B
Handset
Selecting an item starts Activity B
Activity A contains Fragment A
Activity B contains Fragment B

https://developer.android.com/guide/components/fragments

Creating a Fragment

- New callbacks
  - onAttach
  - onCreateView
  - onActivityCreated
  - onDestroyView
  - onDetach



Fragment is added

onAttach()

onCreate()

onCreateView()

onActivityCreated()

onStart()

onResume()

Fragment is active

User navigates backward or fragment is removed/replaced

The fragment is added to the back stack, then removed/replaced

onPause()

onStop()

onDestroyView()

The fragment returns to the layout from the back stack

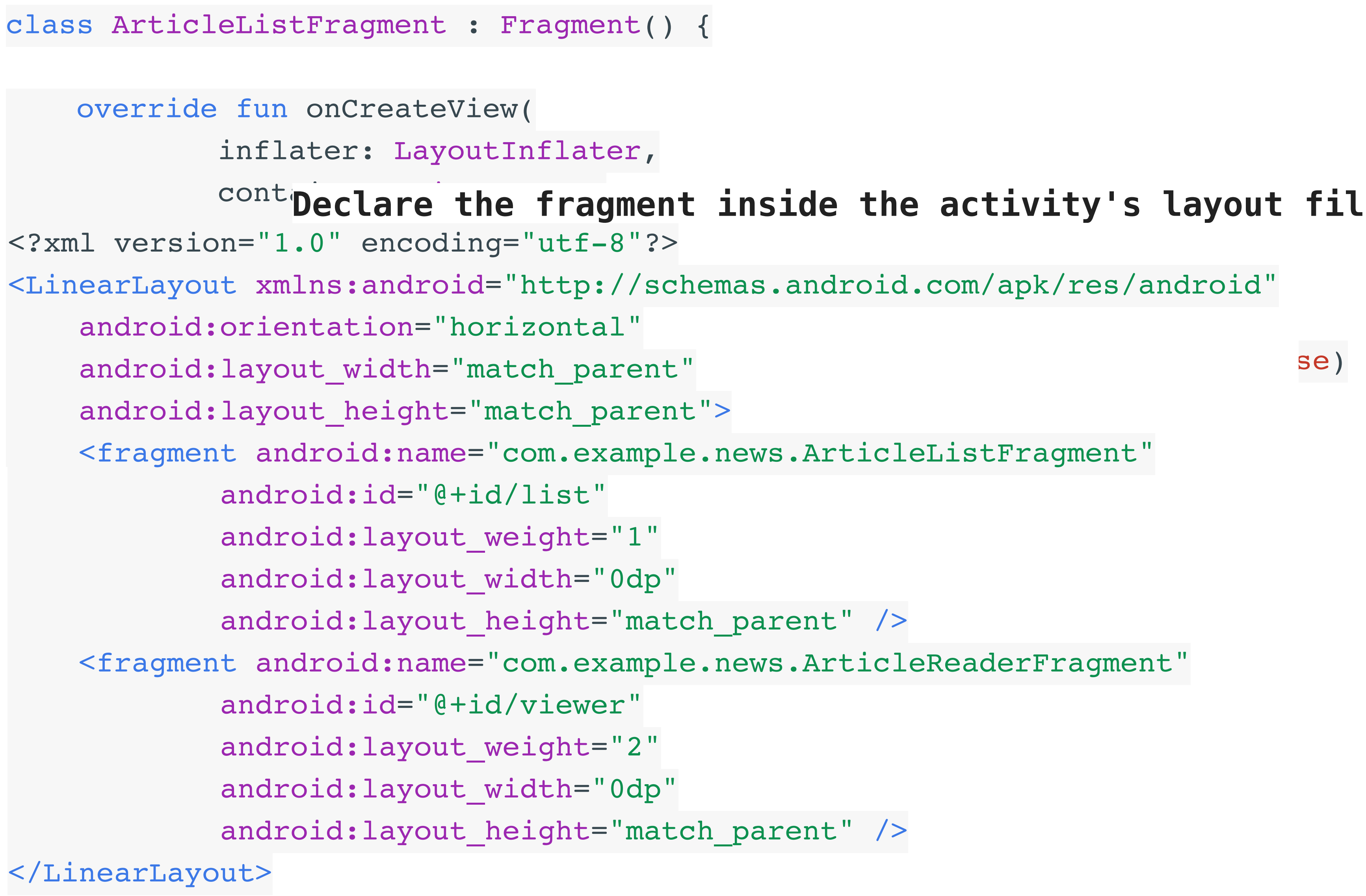
onDestroy()

onDetach()

Fragment is destroyed

https://developer.android.com/reference/android/support/v4/app/Fragment

Creating a Fragment



```kotlin
class ArticleListFragment : Fragment() {

    override fun onCreateView(
            inflater: LayoutInflater,
            cont
                                                                        se)
```

Declare the fragment inside the activity's layout file.

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <fragment android:name="com.example.news.ArticleListFragment"
            android:id="@+id/list"
            android:layout_weight="1"
            android:layout_width="0dp"
            android:layout_height="match_parent" />
    <fragment android:name="com.example.news.ArticleReaderFragment"
            android:id="@+id/viewer"
            android:layout_weight="2"
            android:layout_width="0dp"
            android:layout_height="match_parent" />
</LinearLayout>
```

https://developer.android.com/reference/android/support/v4/app/Fragment

DEMO

Creating a Fragment

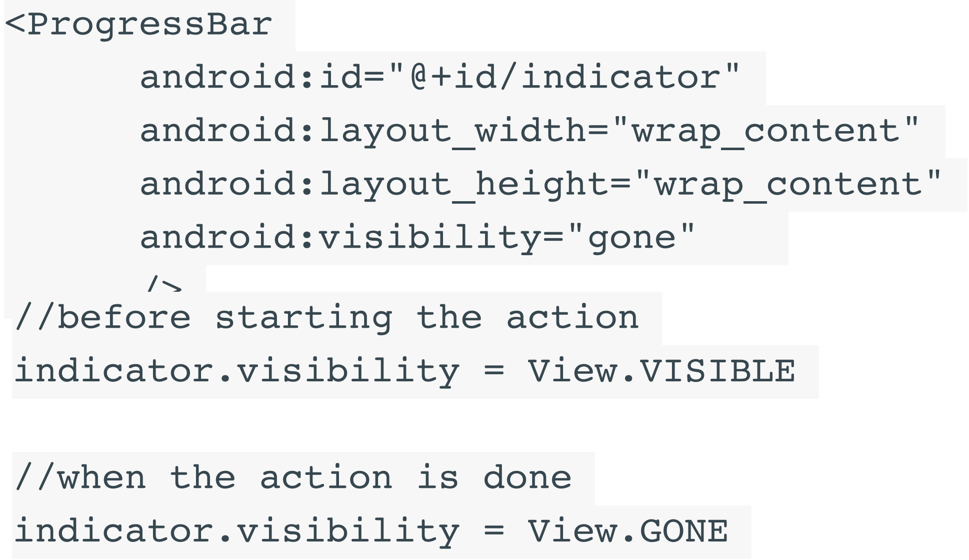
```
class ArticleListFragment : Fragment() {

    override fun onCreateView(
            inflater: LayoutInflater,
            container: ViewGroup?,
            savedInstanceState: Bundle?
    ): View {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.example_fragment, container, false)
    }
}
```

Or, programmatically add the fragment to an existing ViewGroup

```
val fragmentManager = supportFragmentManager
val fragmentTransaction = fragmentManager.beginTransaction()


val fragment = ArticleListFragment()
fragmentTransaction.add(R.id.fragment_container, fragment)
fragmentTransaction.commit()
```

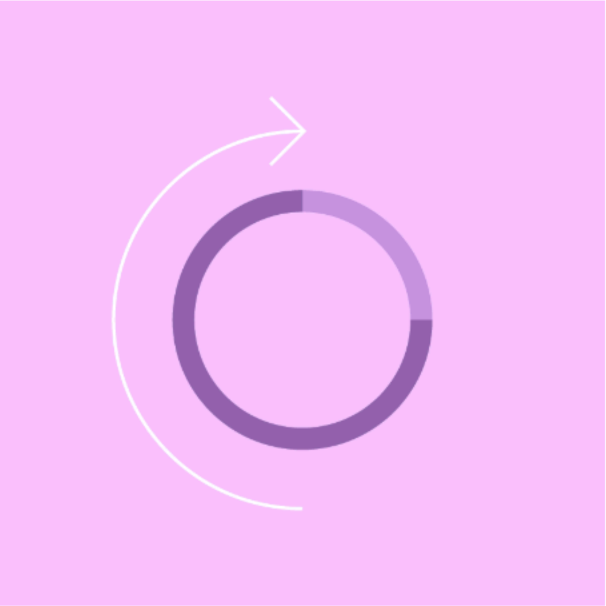
https://developer.android.com/reference/android/support/v4/app/Fragment

Progress Indicators



```
<ProgressBar
      android:id="@+id/indicator"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:visibility="gone"
      />
```

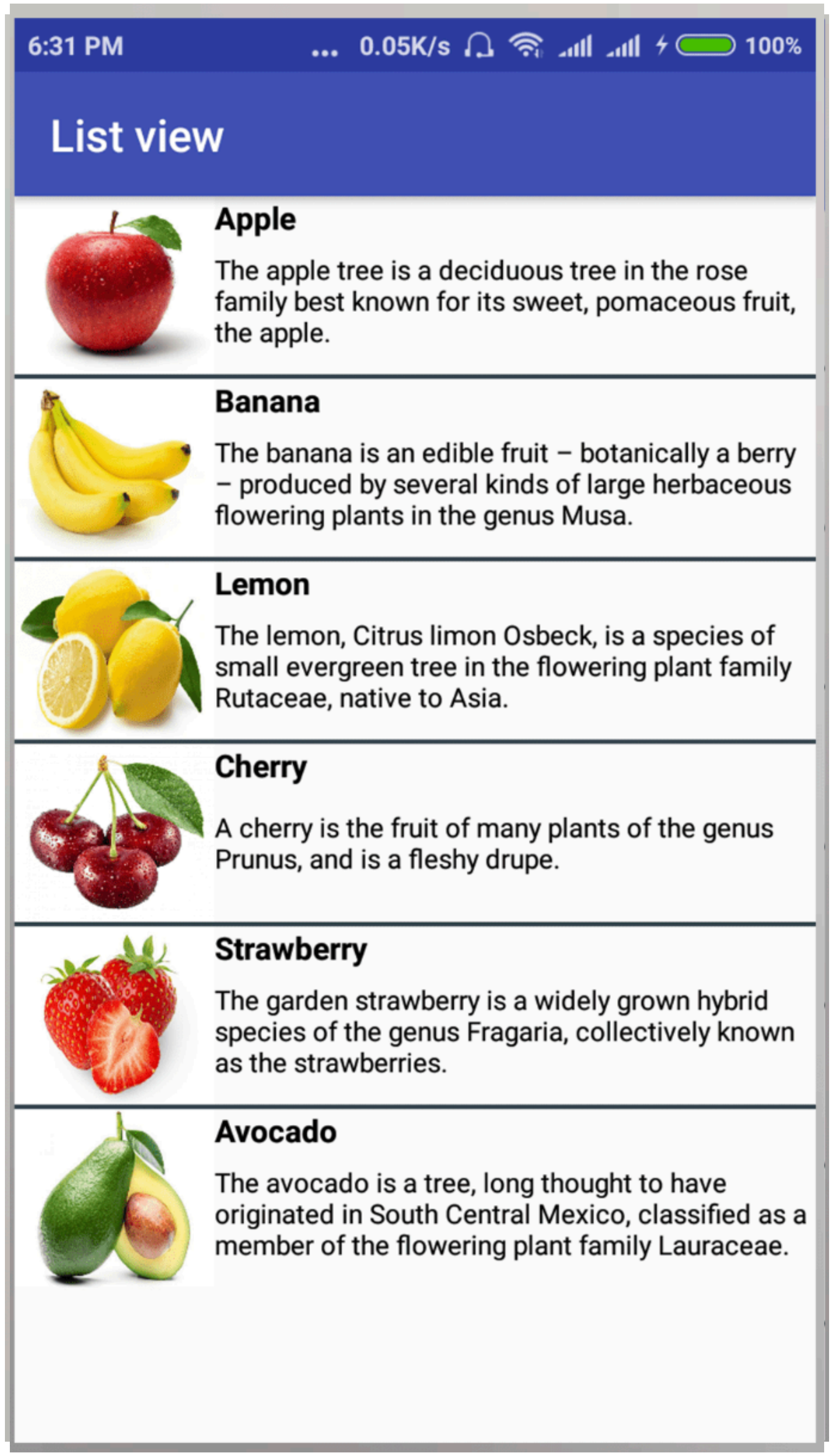
```
//before starting the action
indicator.visibility = View.VISIBLE

//when the action is done
indicator.visibility = View.GONE
```



https://developer.android.com/reference/android/widget/ProgressBar

Lists

- ListView
- RecyclerView



List view

Apple

The apple tree is a deciduous tree in the rose family best known for its sweet, pomaceous fruit, the apple.

Banana

The banana is an edible fruit – botanically a berry – produced by several kinds of large herbaceous flowering plants in the genus Musa.

Lemon

The lemon, Citrus limon Osbeck, is a species of small evergreen tree in the flowering plant family Rutaceae, native to Asia.

Cherry

A cherry is the fruit of many plants of the genus Prunus, and is a fleshy drupe.

Strawberry

The garden strawberry is a widely grown hybrid species of the genus Fragaria, collectively known as the strawberries.

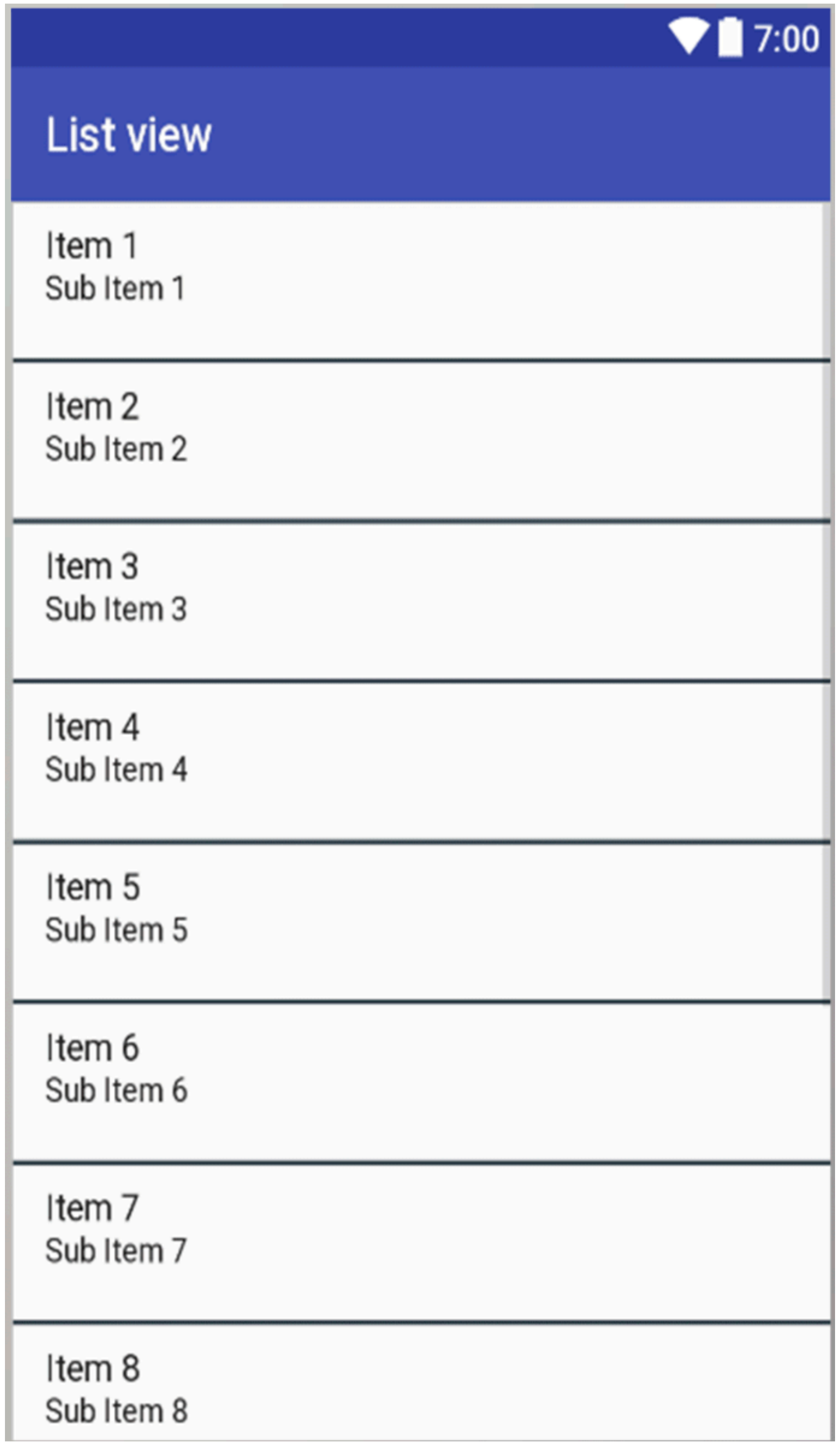
Avocado

The avocado is a tree, long thought to have originated in South Central Mexico, classified as a member of the flowering plant family Lauraceae.

ListView

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:paddingLeft="8dp"
        android:paddingRight="8dp">
    <ListView android:id="@android:id/list"
              android:layout_width="match_parent"
              android:layout_height="match_parent"/>
</LinearLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
 <LinearLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
     android:layout_width="match_parent"
     android:layout_height="wrap_content"
     android:orientation="vertical">
     <TextView android:id="@+id/text"
         android:textSize="16sp"
         android:textStyle="bold"
         android:layout_width="match_parent"
         android:layout_height="wrap_content"/>
     <TextView android:id="@+id/subText"
       ...
 </LinearLayout>
```



7:00

List view

Item 1
Sub Item 1

Item 2
Sub Item 2

Item 3
Sub Item 3

Item 4
Sub Item 4

Item 5
Sub Item 5

Item 6
Sub Item 6

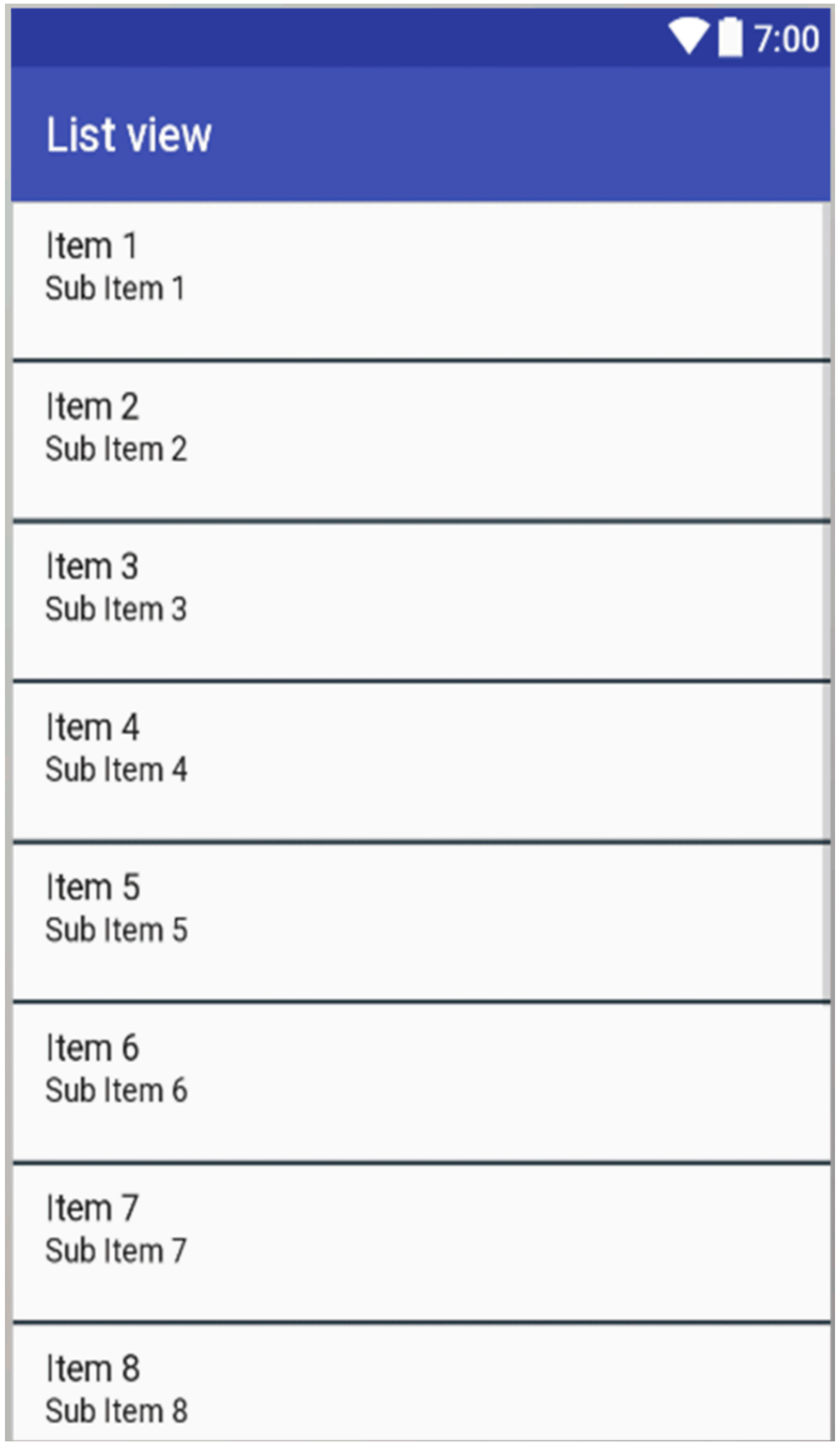
Item 7
Sub Item 7

Item 8
Sub Item 8

DEMO

ListView

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="..."
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:paddingLeft="8dp"
        android:paddingRight="8dp">
    <ListView android:id="@android:id/myList"
              android:layout_width="match_parent"
              android:layout_height="match_parent"/>
</LinearLayout>
```



7:00

List view

Item 1
Sub Item 1

Item 2
Sub Item 2

Item 3
Sub Item 3

Item 4
Sub Item 4

Item 5
Sub Item 5

Item 6
Sub Item 6

Item 7
Sub Item 7

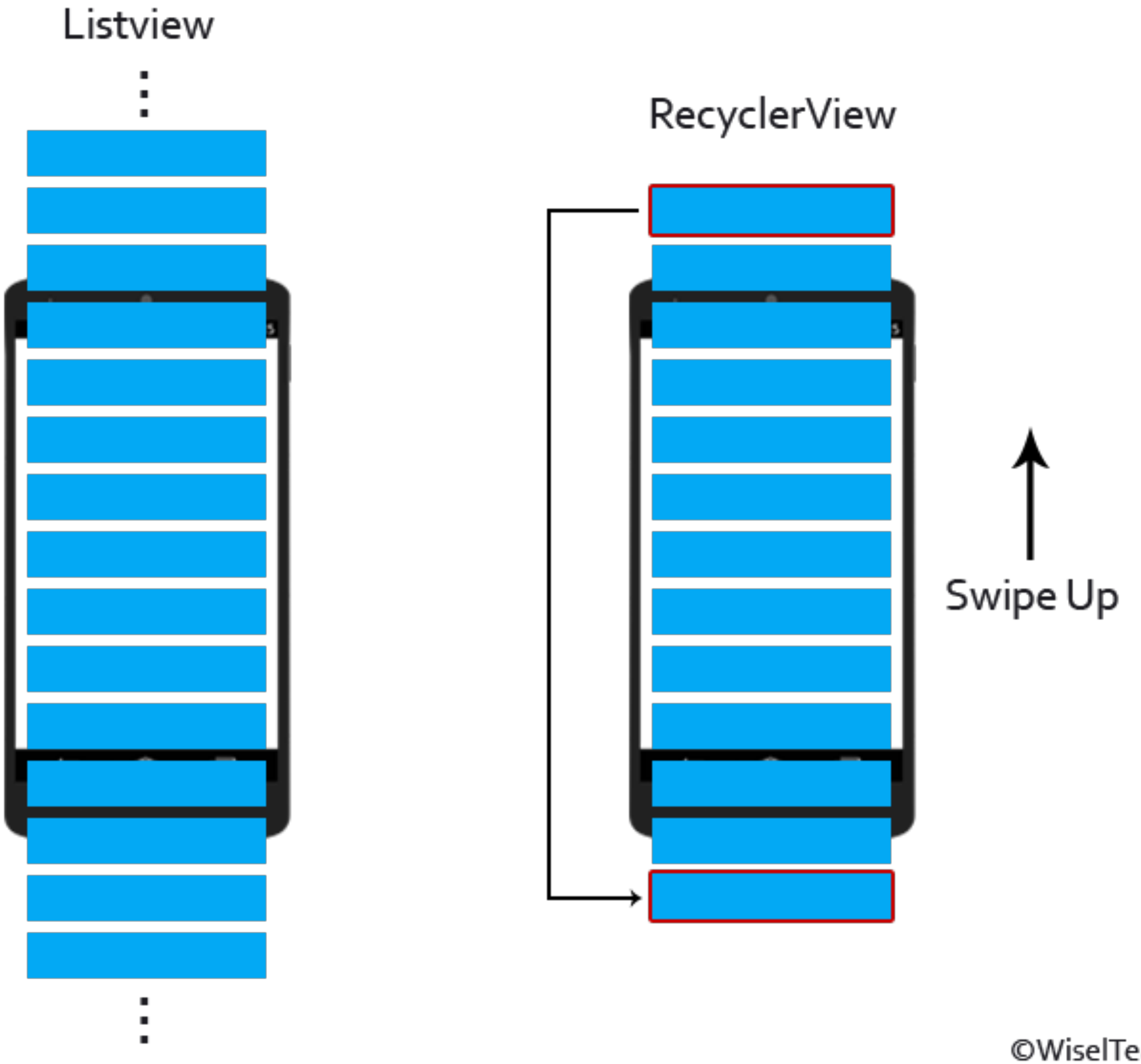
Item 8
Sub Item 8

```
val arrayAdapter = ArrayAdapter<String>(this,
  android.R.layout.simple_list_item_1, arrayList)

myList.adapter = arrayAdapter
```

https://developer.android.com/guide/topics/ui/layout/listview

RecyclerView

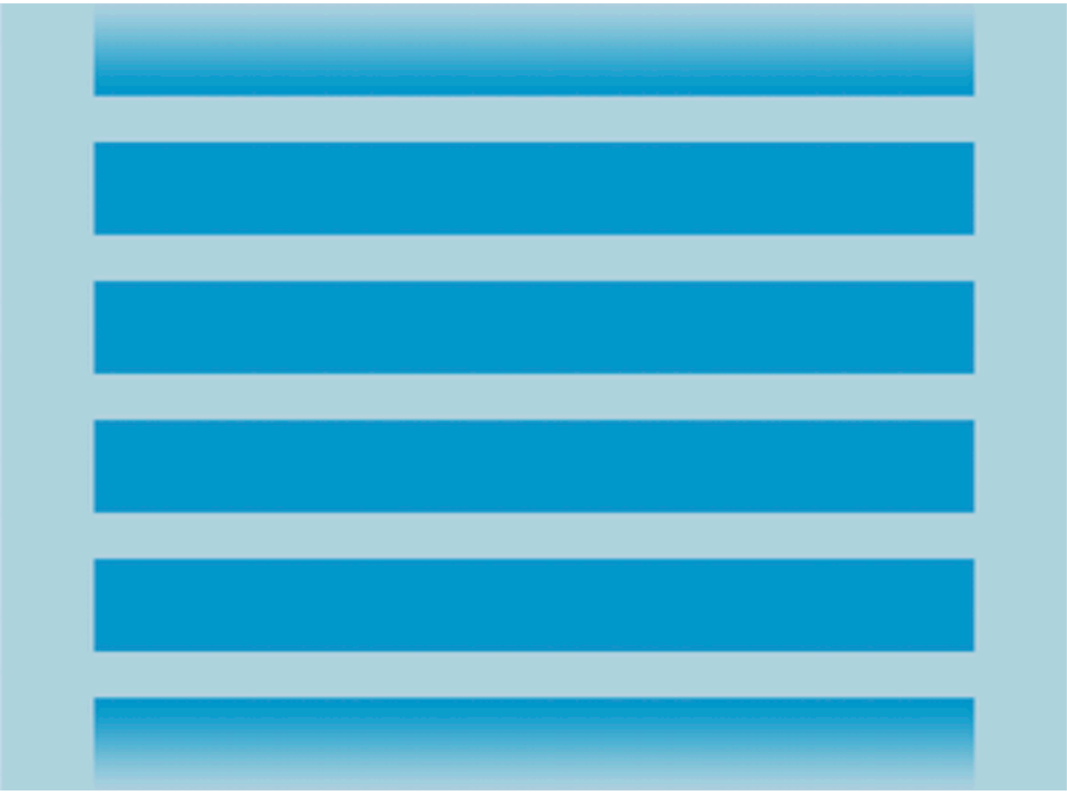
- ListView
- RecyclerView



©WiselTeach

RecyclerView

```
<?xml version="1.0" encoding="utf-8"?>
<!-- A RecyclerView with some commonly used attributes -->
<android.support.v7.widget.RecyclerView
    android:id="@+id/my_recycler_view"
    android:scrollbars="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"/>
```

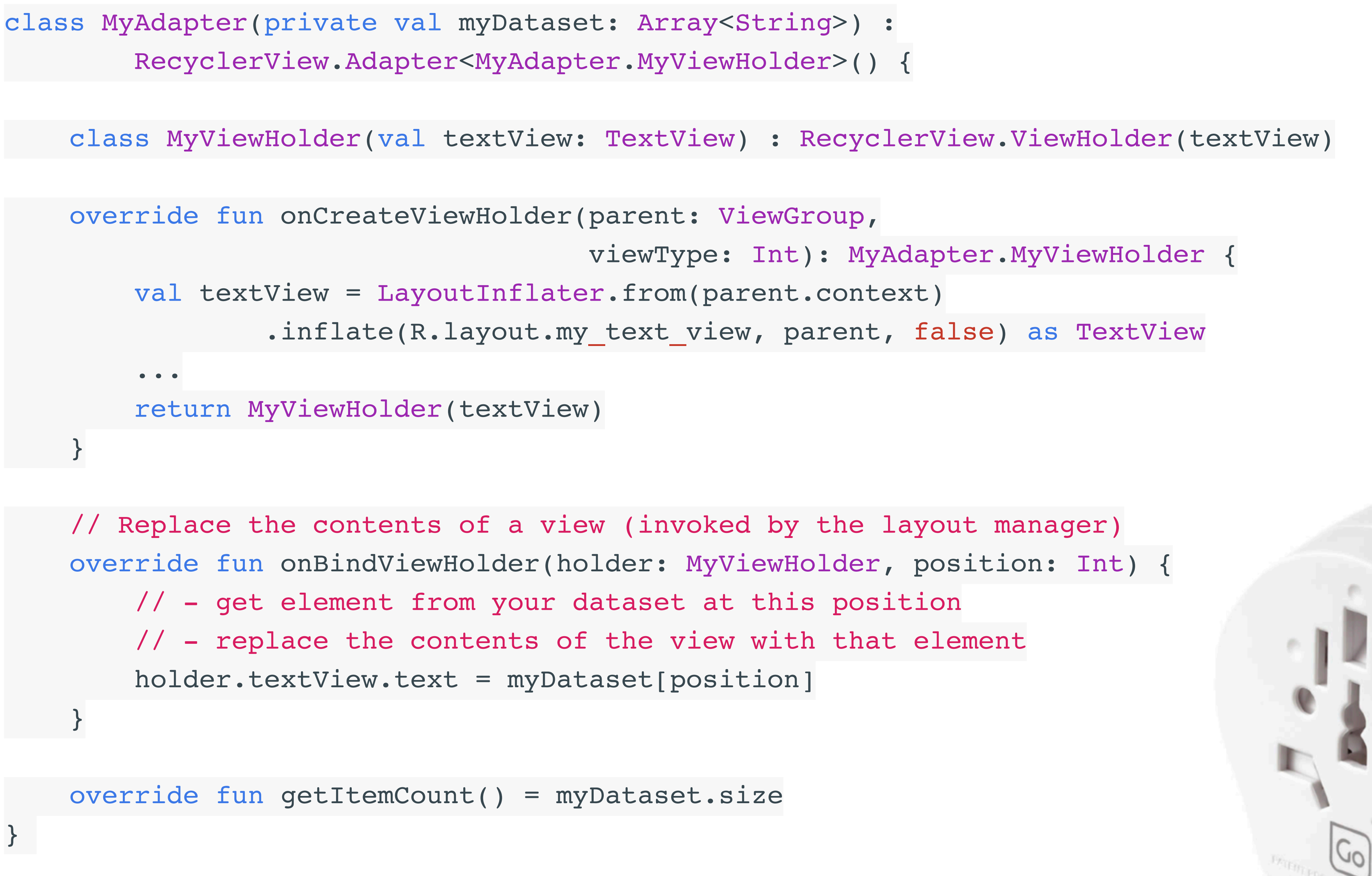


```
class MyActivity : Activity() {
    private lateinit var recyclerView: RecyclerView
    private lateinit var viewAdapter: RecyclerView.Adapter<*>
    private lateinit var viewManager: RecyclerView.LayoutManager
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.my_activity)
        viewManager = LinearLayoutManager(this)
        viewAdapter = MyAdapter(myDataset)
        recyclerView = findViewById<RecyclerView>(R.id.my_recycler_view).apply {
            setHasFixedSize(true)
            layoutManager = viewManager
            adapter = viewAdapter
        }
    }
    // ...
}
```

https://developer.android.com/guide/topics/ui/layout/recyclerview

DEMO

RecyclerView.Adapter

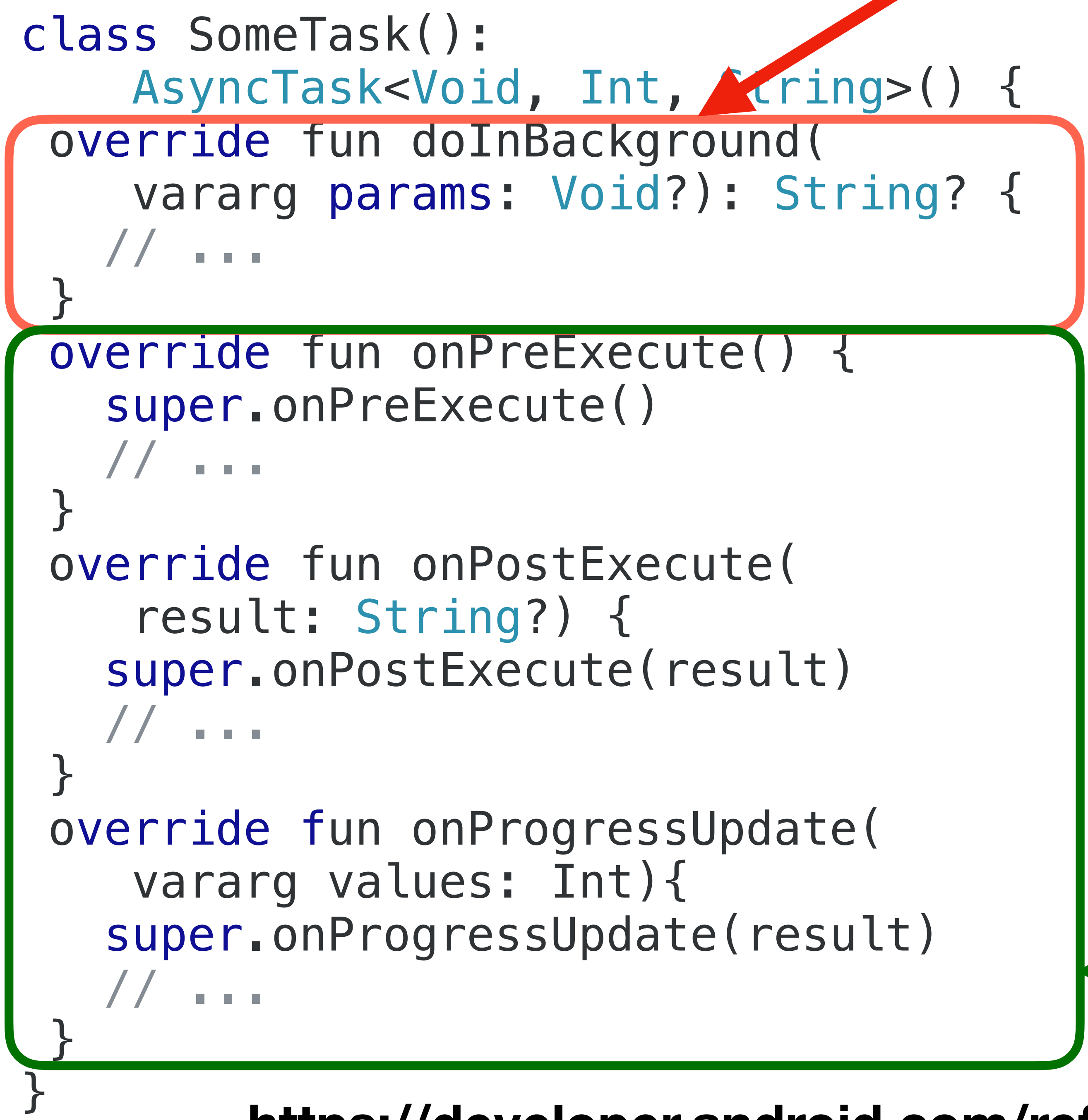


```kotlin
class MyAdapter(private val myDataset: Array<String>) :
        RecyclerView.Adapter<MyAdapter.MyViewHolder>() {

    class MyViewHolder(val textView: TextView) : RecyclerView.ViewHolder(textView)

    override fun onCreateViewHolder(parent: ViewGroup,
                                    viewType: Int): MyAdapter.MyViewHolder {
        val textView = LayoutInflater.from(parent.context)
                .inflate(R.layout.my_text_view, parent, false) as TextView
        ...
        return MyViewHolder(textView)
    }

    // Replace the contents of a view (invoked by the layout manager)
    override fun onBindViewHolder(holder: MyViewHolder, position: Int) {
        // - get element from your dataset at this position
        // - replace the contents of the view with that element
        holder.textView.text = myDataset[position]
    }

    override fun getItemCount() = myDataset.size
}
```

https://developer.android.com/guide/topics/ui/layout/recyclerview

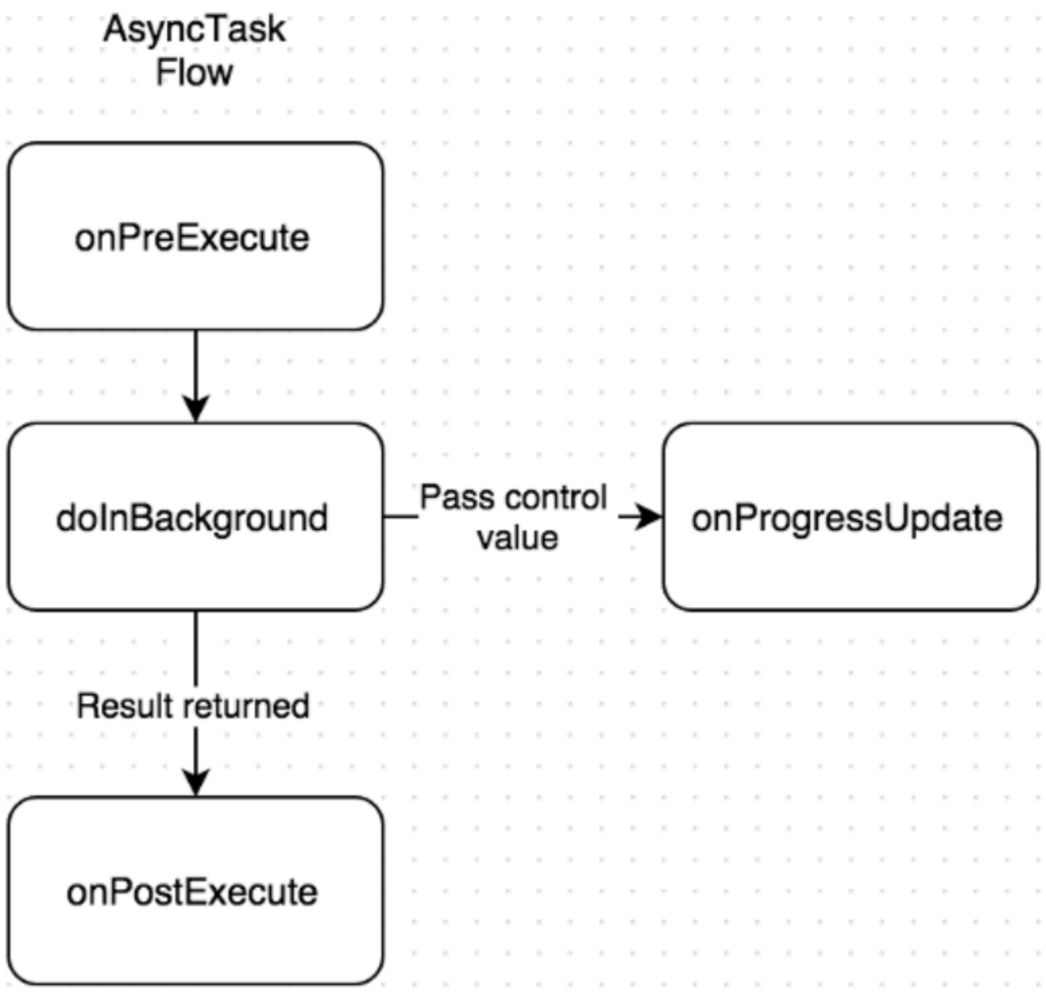
AsyncTask

BackgroundThread



```
class SomeTask():
    AsyncTask<Void, Int, String>() {
  override fun doInBackground(
     vararg params: Void?): String? {
    // ...
  }
  override fun onPreExecute() {
    super.onPreExecute()
    // ...
  }
  override fun onPostExecute(
     result: String?) {
    super.onPostExecute(result)
    // ...
  }
  override fun onProgressUpdate(
     vararg values: Int){
    super.onProgressUpdate(result)
    // ...
  }
}
```

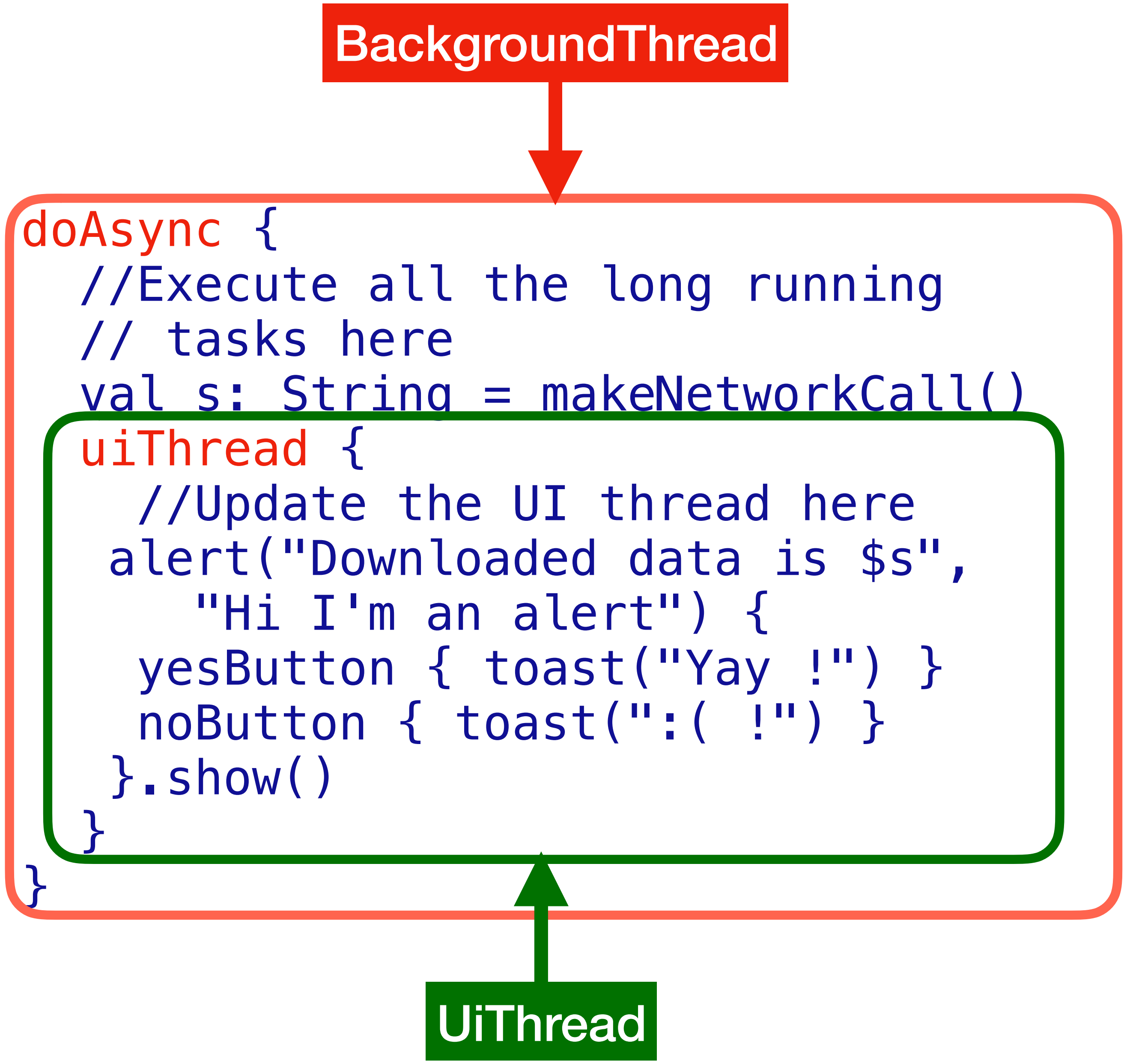
UiThread



https://developer.android.com/reference/android/os/AsyncTask

DEMO

Anko AsyncTask Alternative



BackgroundThread

```
doAsync {
  //Execute all the long running
  // tasks here
  val s: String = makeNetworkCall()
  uiThread {
    //Update the UI thread here
   alert("Downloaded data is $s",
      "Hi I'm an alert") {
    yesButton { toast("Yay !") }
    noButton { toast(":( !") }
   }.show()
  }
}
```

UiThread

https://github.com/Kotlin/anko/wiki/Anko-Coroutines

Lecture outcomes

- Support different screen, using layouts and fragments
- ListView, RecyclerView, Progress Indicators
- Retrieve data on background threads