

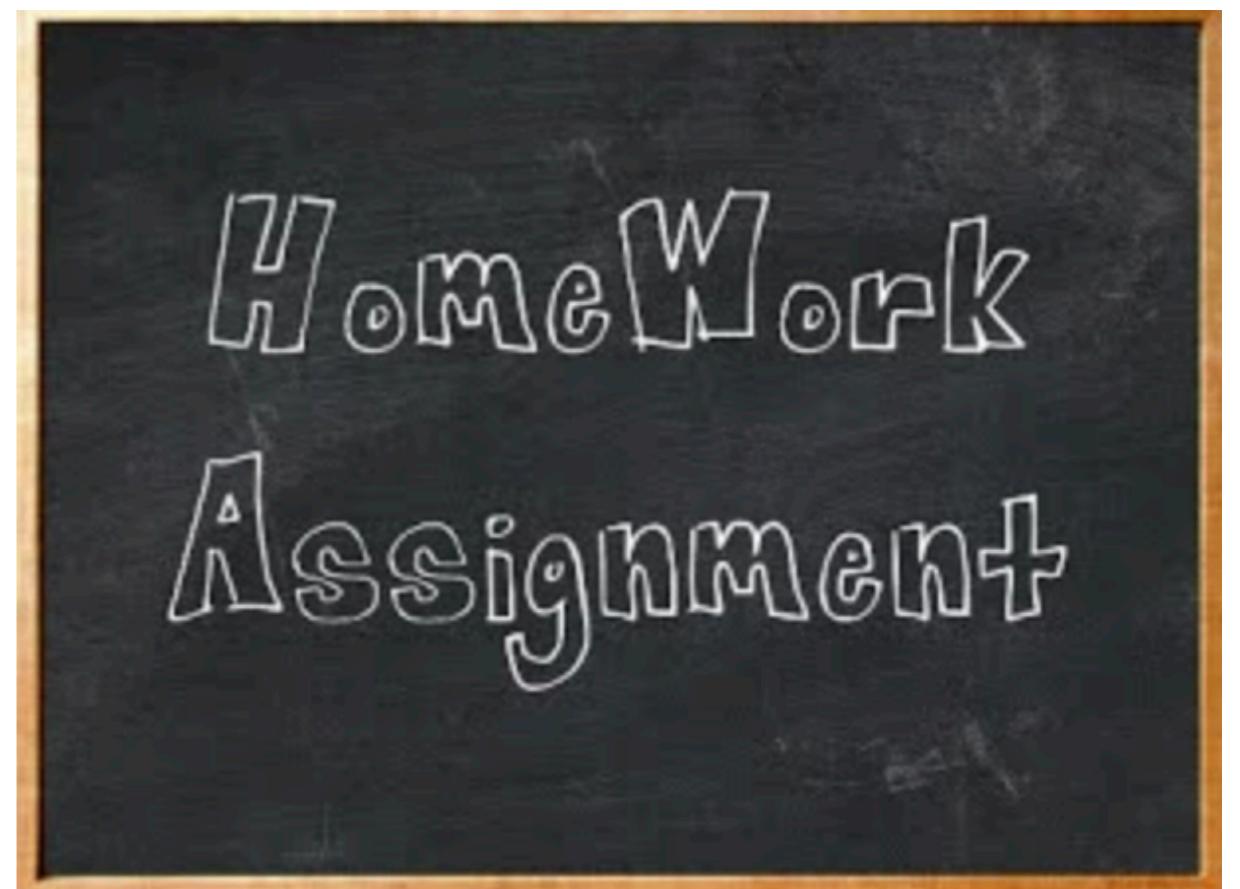
Lecture #2

Lists and Rest Resources

Mobile Applications 2018-2019

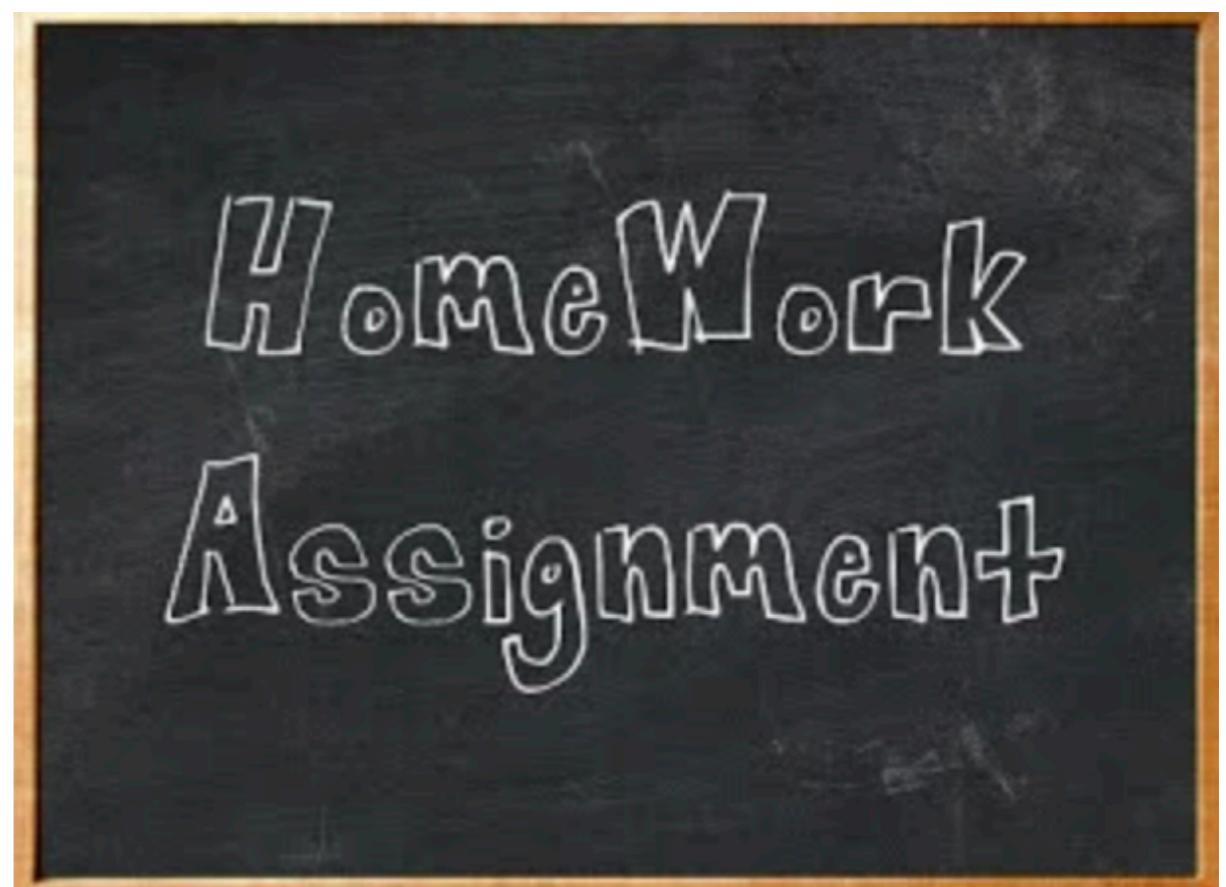
Homework assignments

- First assignment - project details
- Two projects
 - CRUD Application (in two flavors)
 - Bonuses



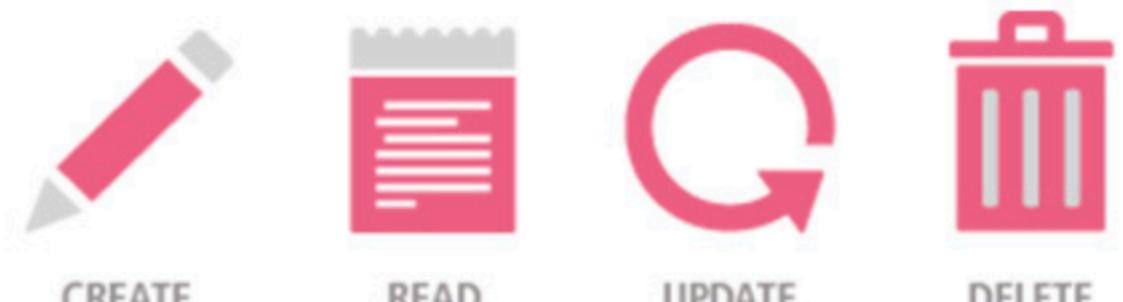
Homework assignments

- 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

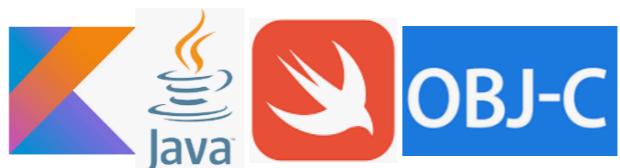


Non-Native

C R U D

CRUD Application

Native



CREATE



READ



UPDATE



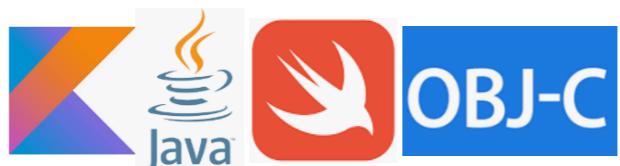
DELETE

Non-Native

C R U D

CRUD Application

Native



CREATE



READ

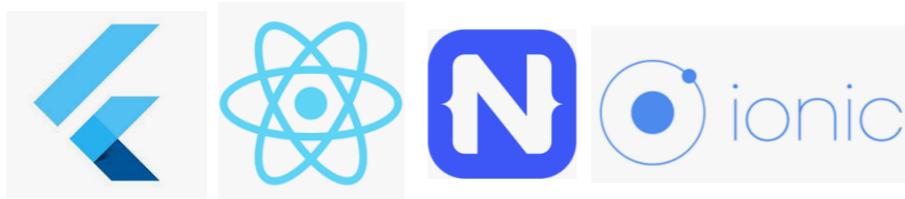


UPDATE



DELETE

Non-Native



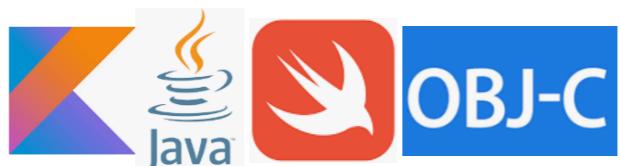
C R U D



Other

CRUD Application

Native



CREATE



READ

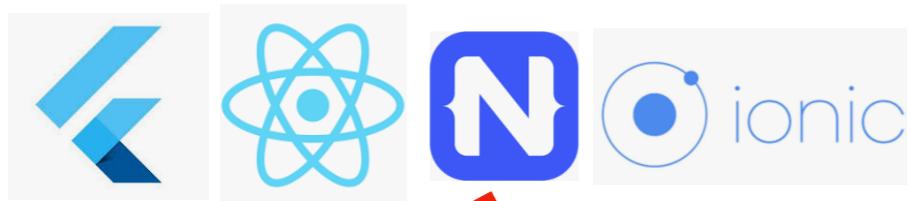


UPDATE



DELETE

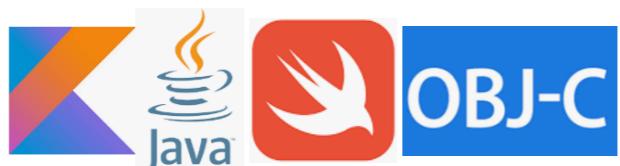
Non-Native



C R U D

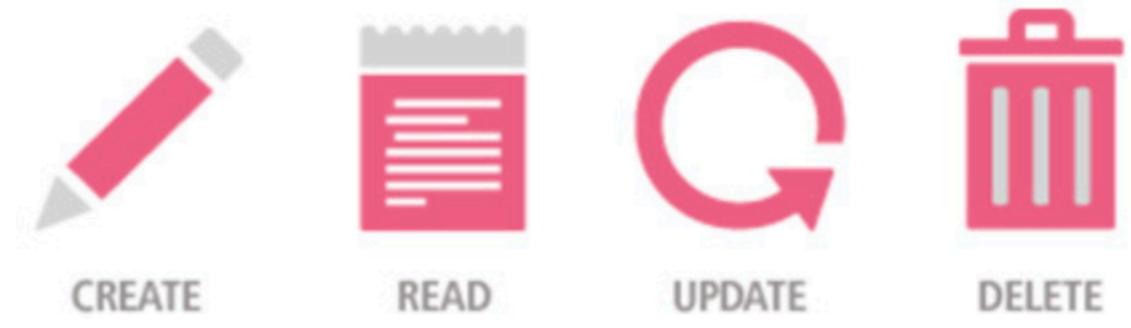
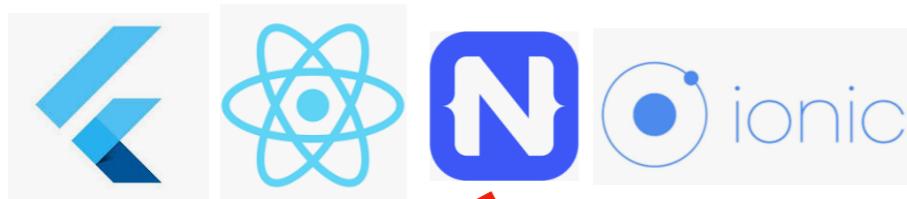
CRUD Application

Native



AND

Non-Native



C R U D



Native

Non-Native



Native



Non-Native



+1p Authentication

Native



Non-Native

B O N U S

Native

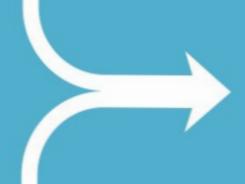
OR 

Non-Native

+1p Authentication
**2p Cloud Persistence*

B O N U S

Native

OR 

Non-Native

+1p Animations

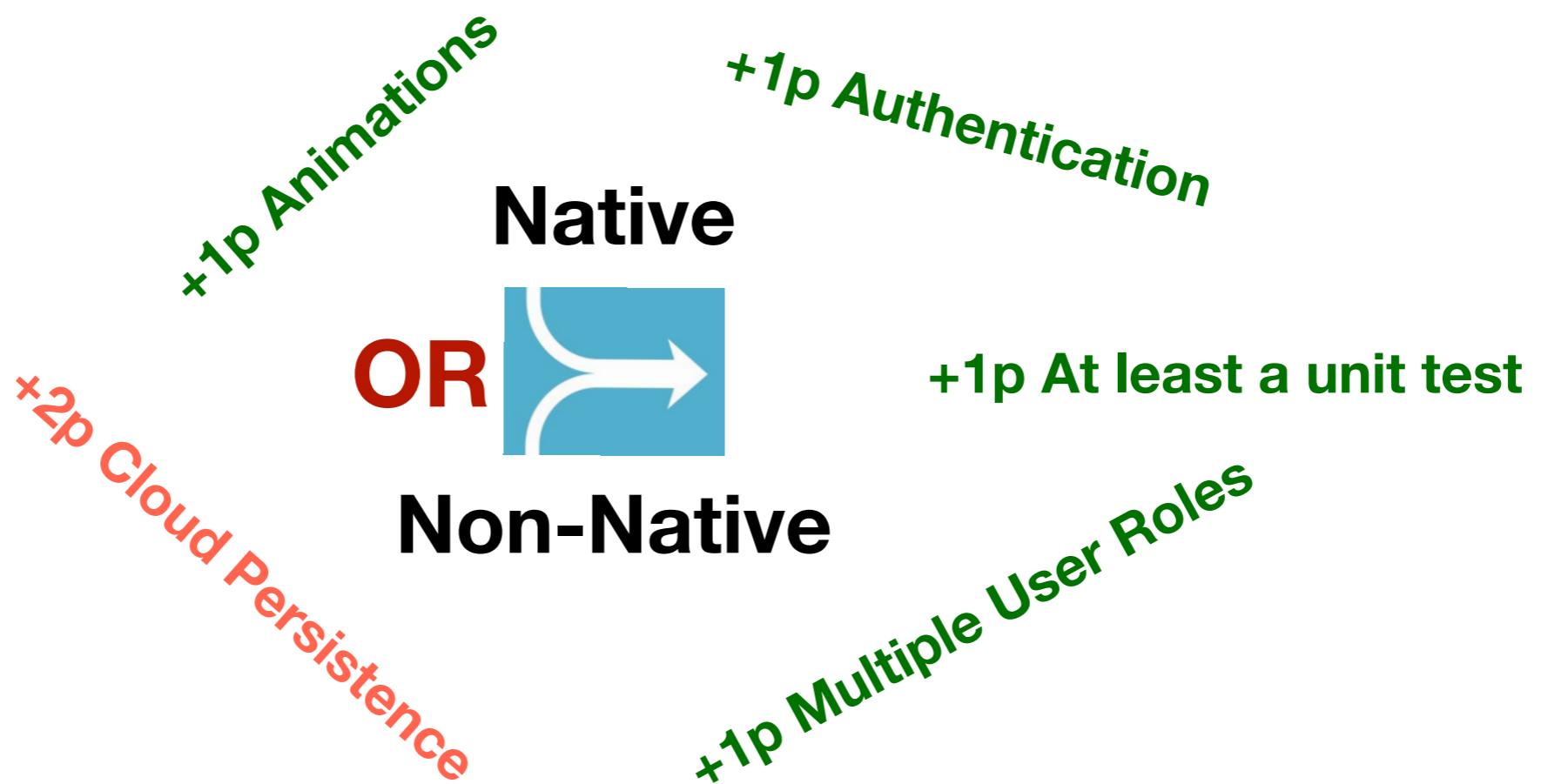
+1p Authentication

+2p Cloud Persistence

B O N U S

+1p Animations
Native
OR
Non-Native
+1p Authentication
+1p Multiple User Roles
+2p Cloud Persistence

B O N U S





+1p Published on
Google Store

+1p Bug
tracking

+1p Uses
Material Design

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

Native



Non-Native

+1p Published on
Play Store

+1p Integrated
ads

+1p At least a unit test

+1p InApp
Purchases

+2p Has more than
50 downloads
on the store

+1p Animations

+1p Authentication

+1p Multiple User Roles

+2p Cloud Persistence



+1p Published on
Google Store

+1p Bug
tracking

+1p Uses
Material Design

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

Native



Non-Native

+1p Animations

+2p Cloud Persistence

+1p Authentication

+1p Multiple User Roles

+1p Published on
Play Store

+1p Integrated
ads

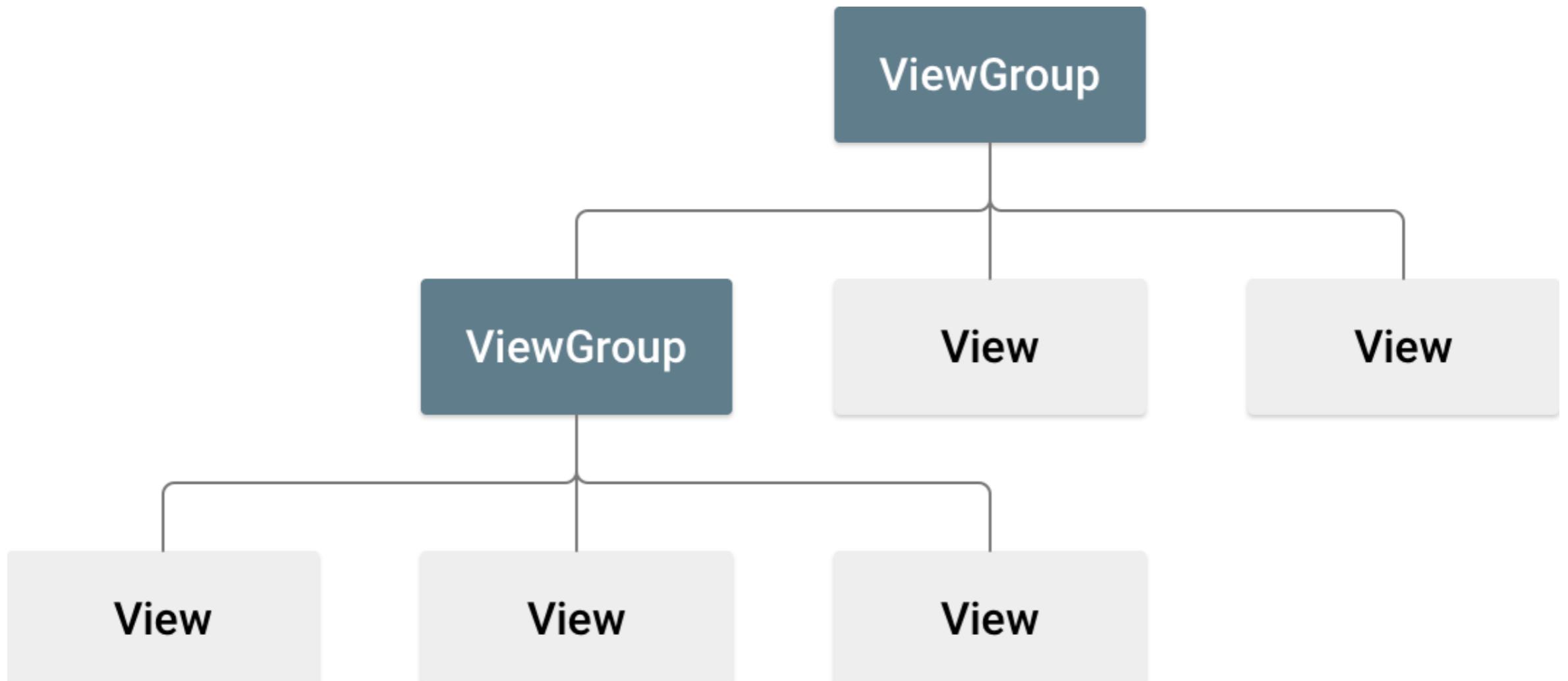
+1p At least a unit test

+1p InApp
Purchases

+2p Has more than
50 downloads
on the store

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>
```



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)
}
```



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)
}
```



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)
}
```



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)
}
```



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"
/>
...
```

```
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)
}
```



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!")
}
```



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!")
}
```



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)
}

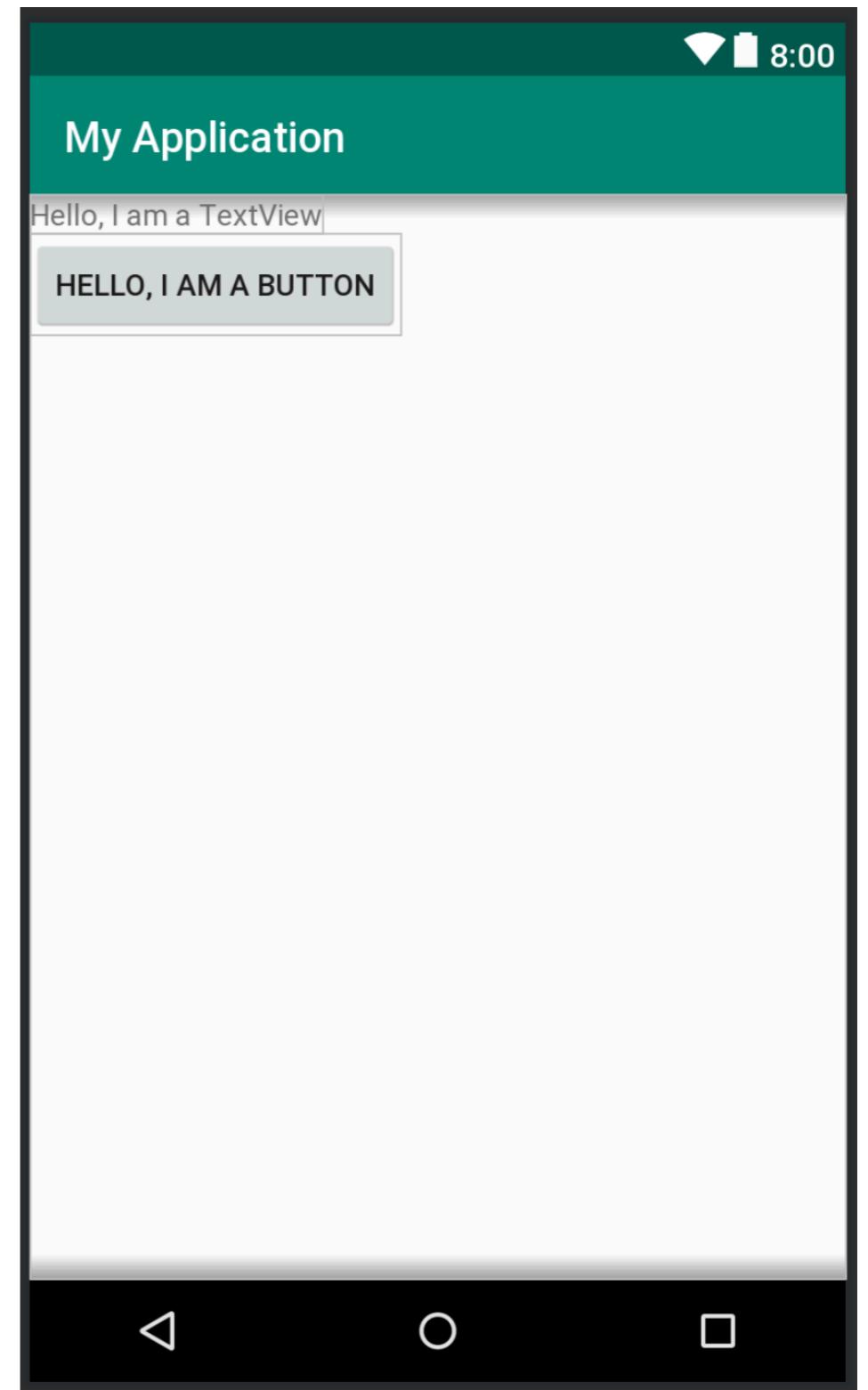
fun sendMessage(view: View) {
    logd("Ready!")
}
```



Using Android KTX

```
...<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!")
}
```



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"  
    />  
...  
  
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!")  
}
```



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"  
    />  
...
```

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

```
fun sendMessage(view: View) {  
    logd("Ready!")  
}
```



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



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



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



DEMO

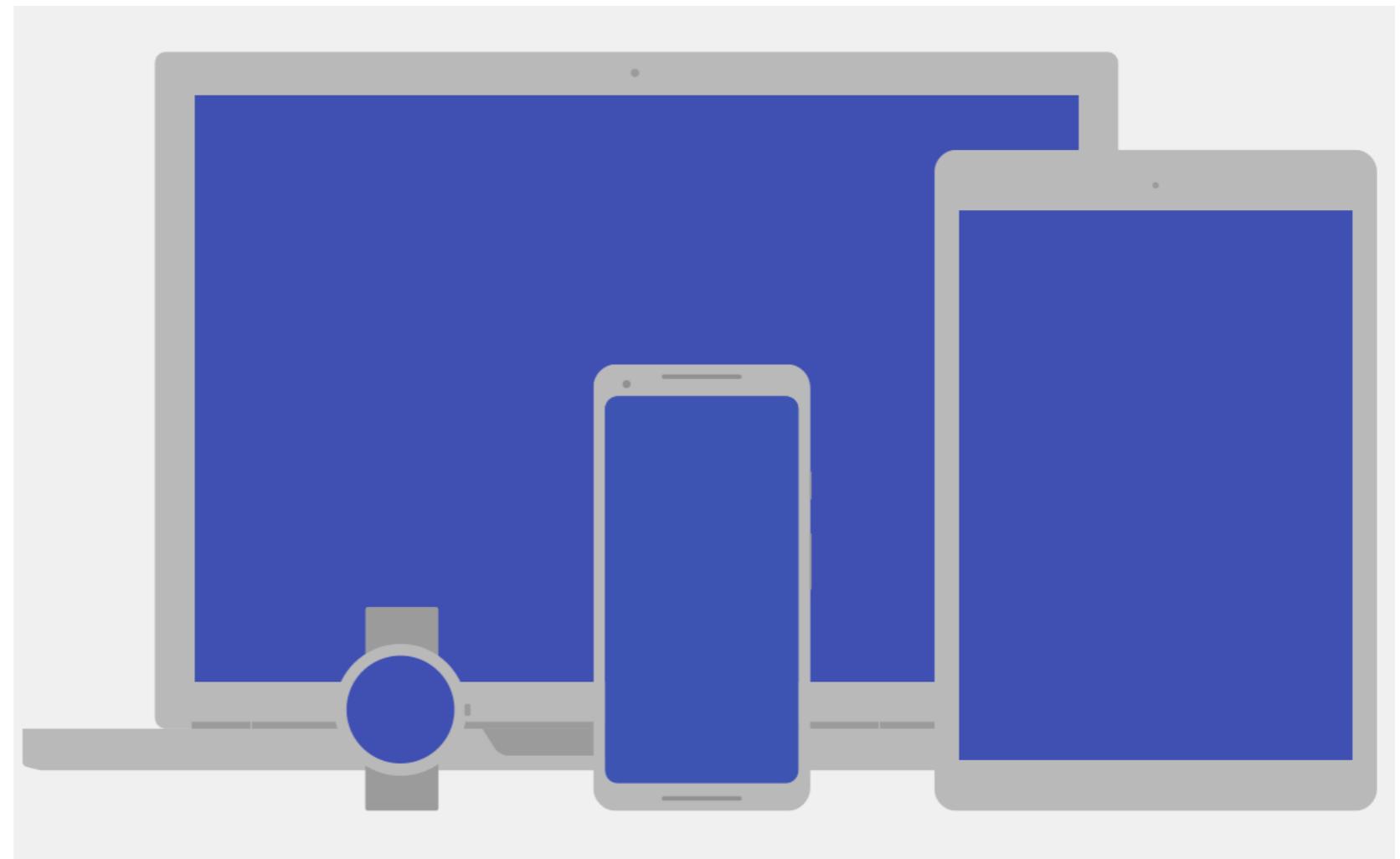
Using Android KTX

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



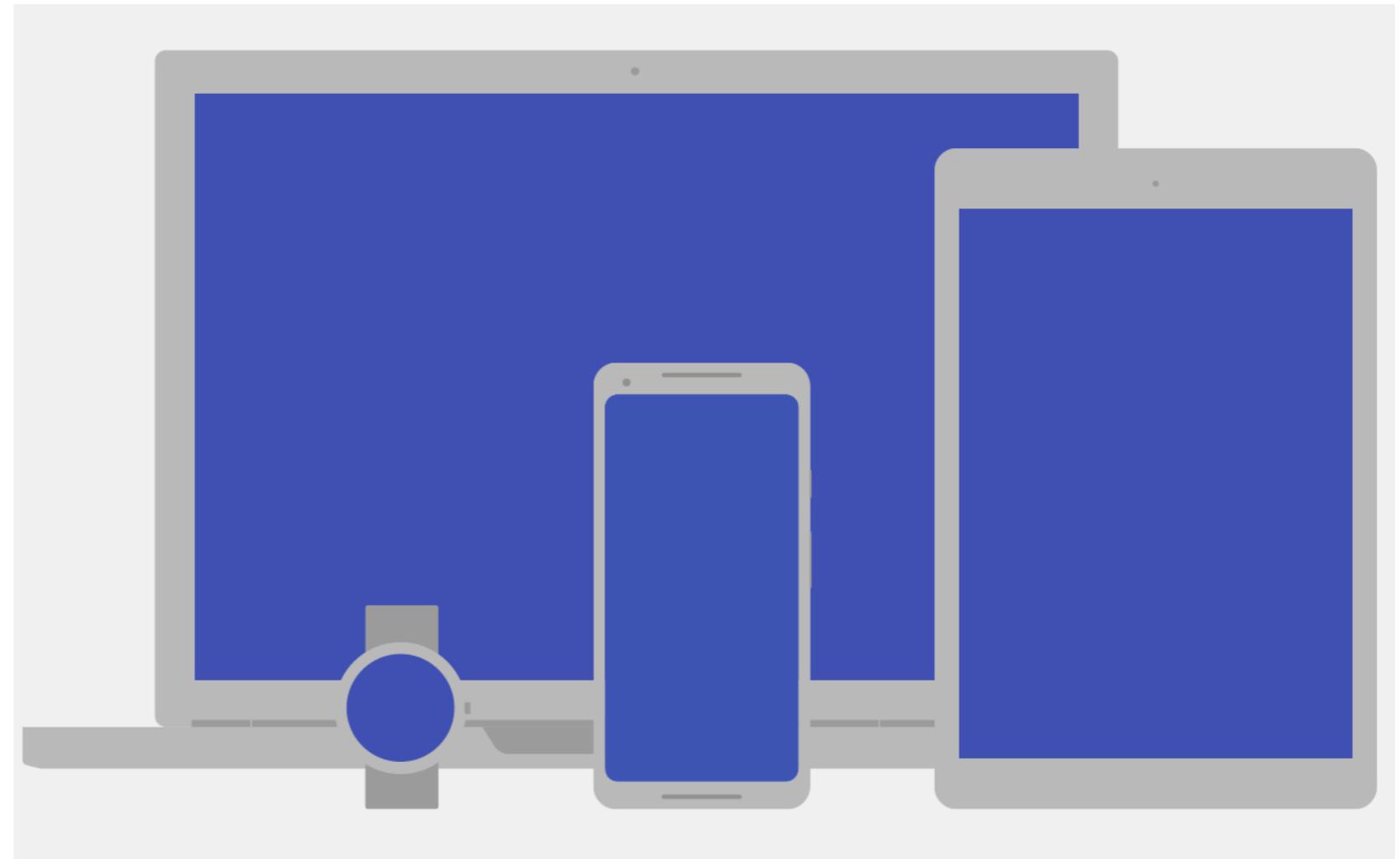
<https://developer.android.com/kotlin/ktx>

Supporting different screen sizes



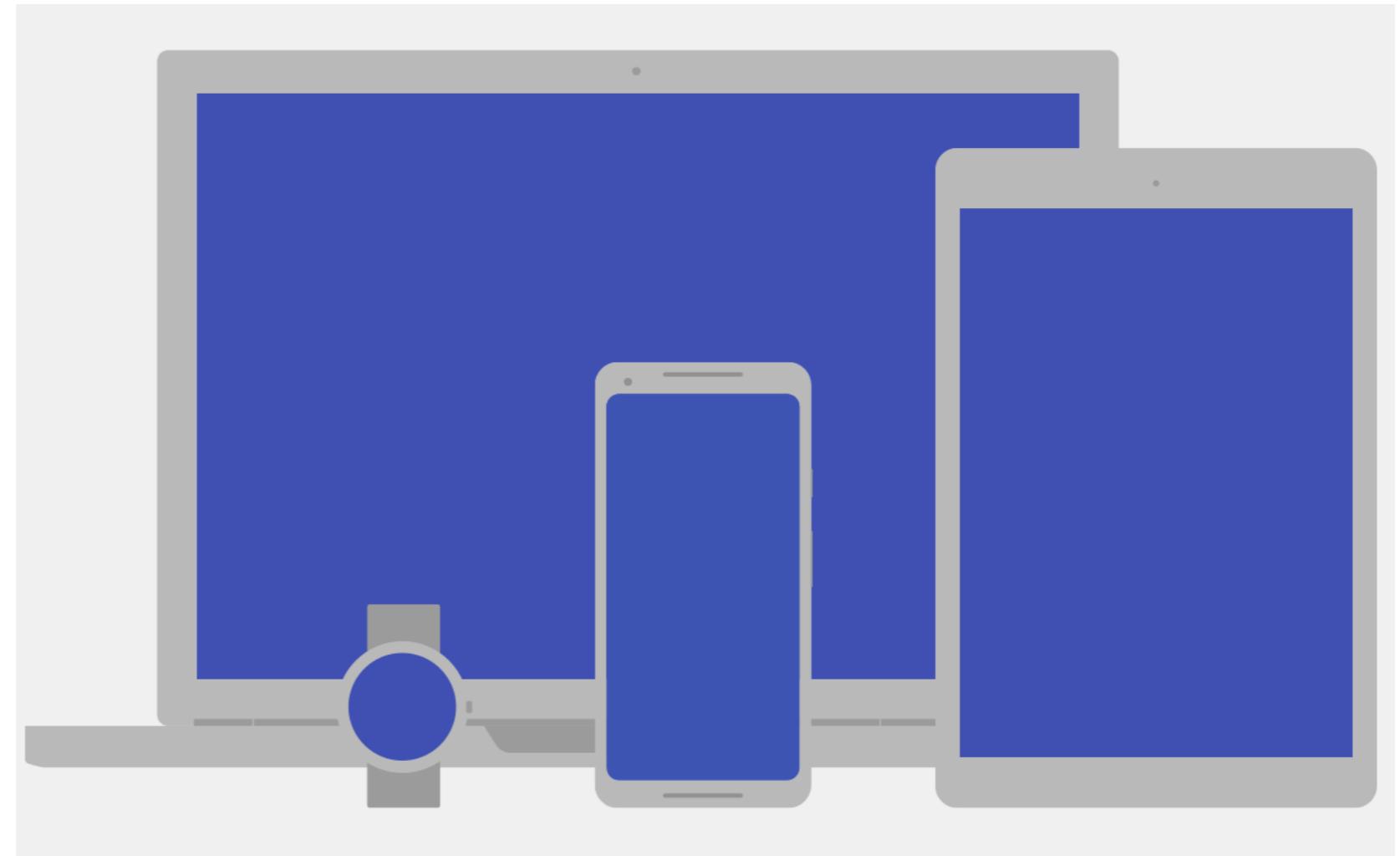
Supporting different screen sizes

- Flexible layouts



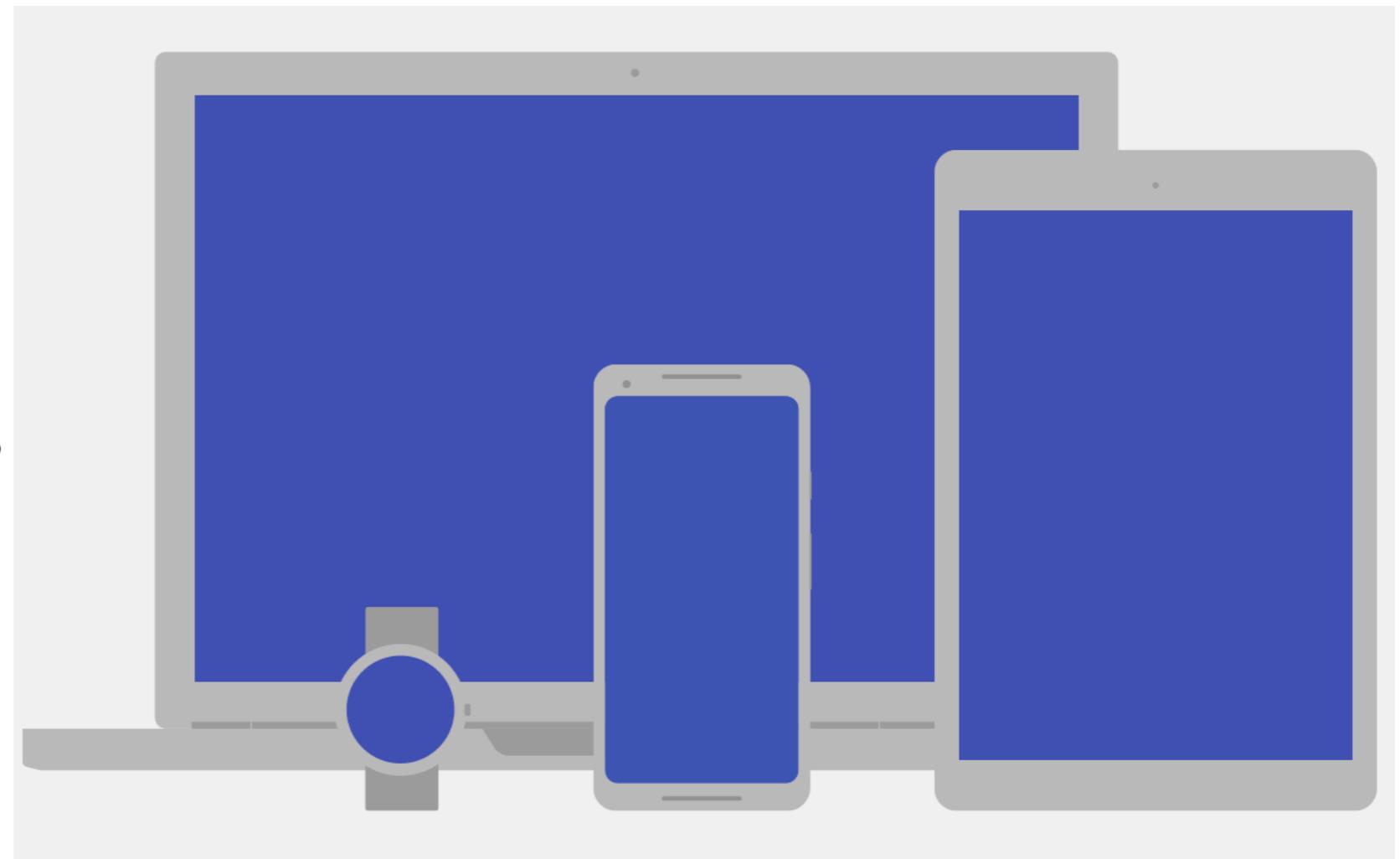
Supporting different screen sizes

- Flexible layouts
- Alternative layouts



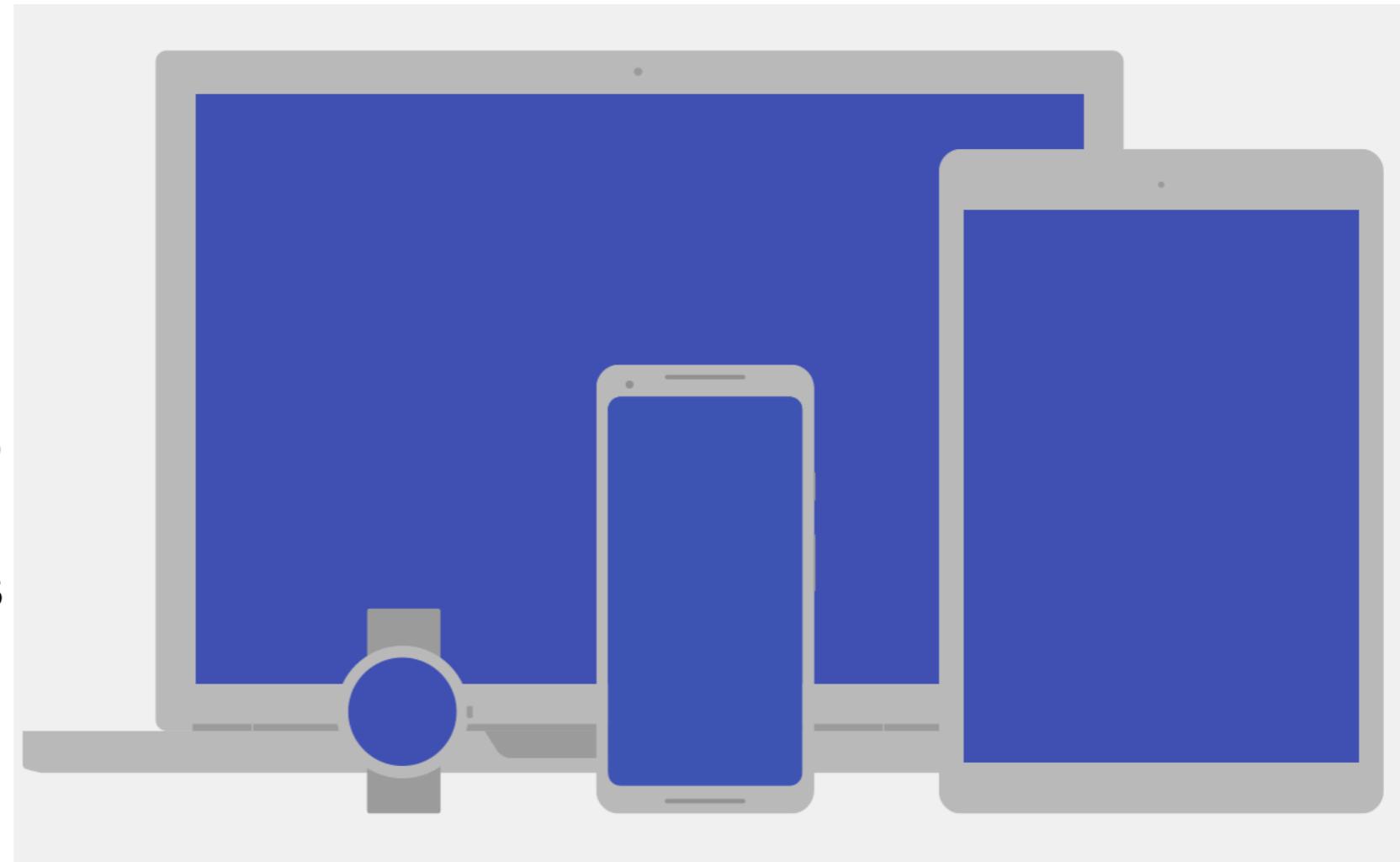
Supporting different screen sizes

- Flexible layouts
- Alternative layouts
- Stretchable images



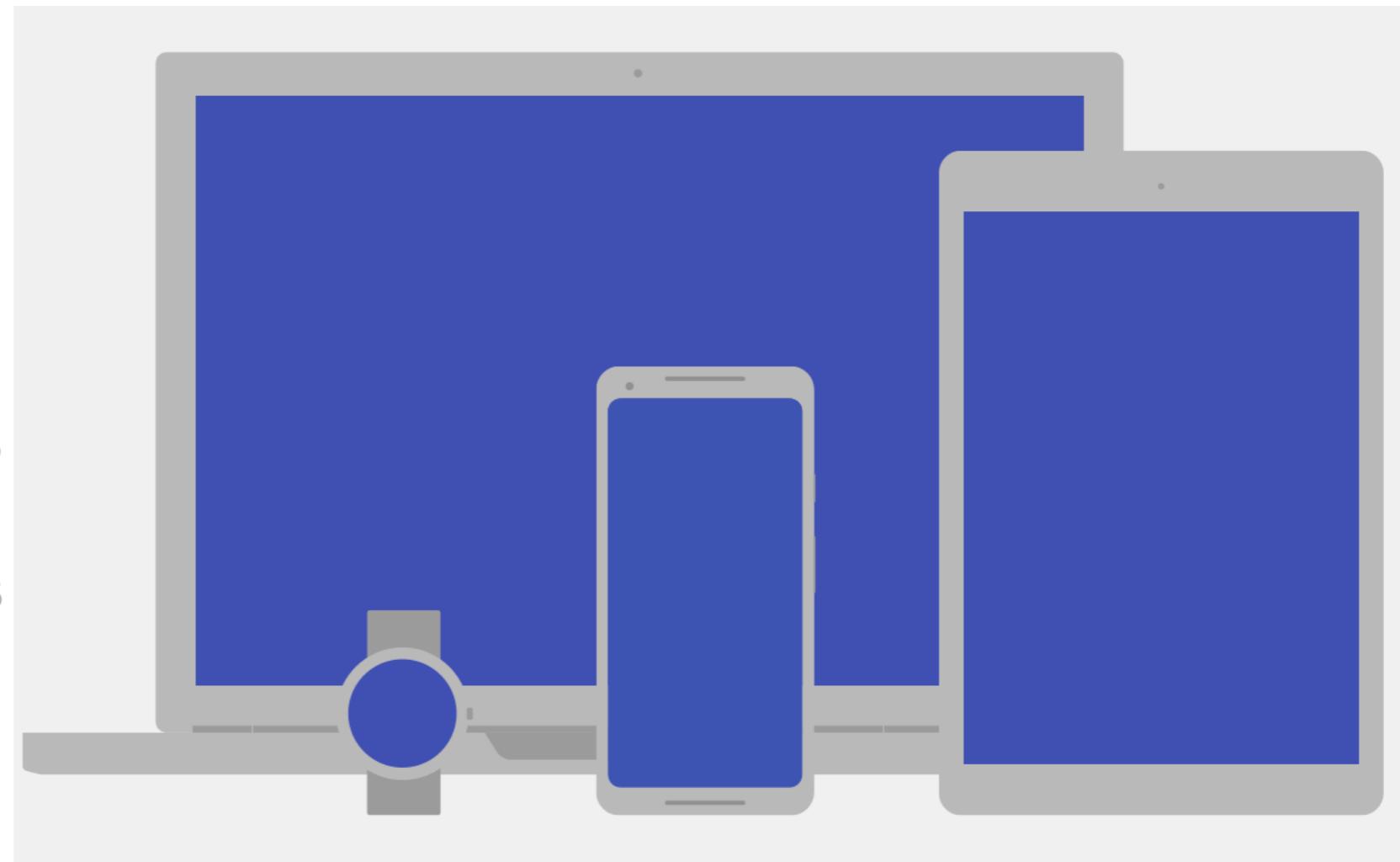
Supporting different screen sizes

- Flexible layouts
- Alternative layouts
- Stretchable images
- Alternative bitmaps



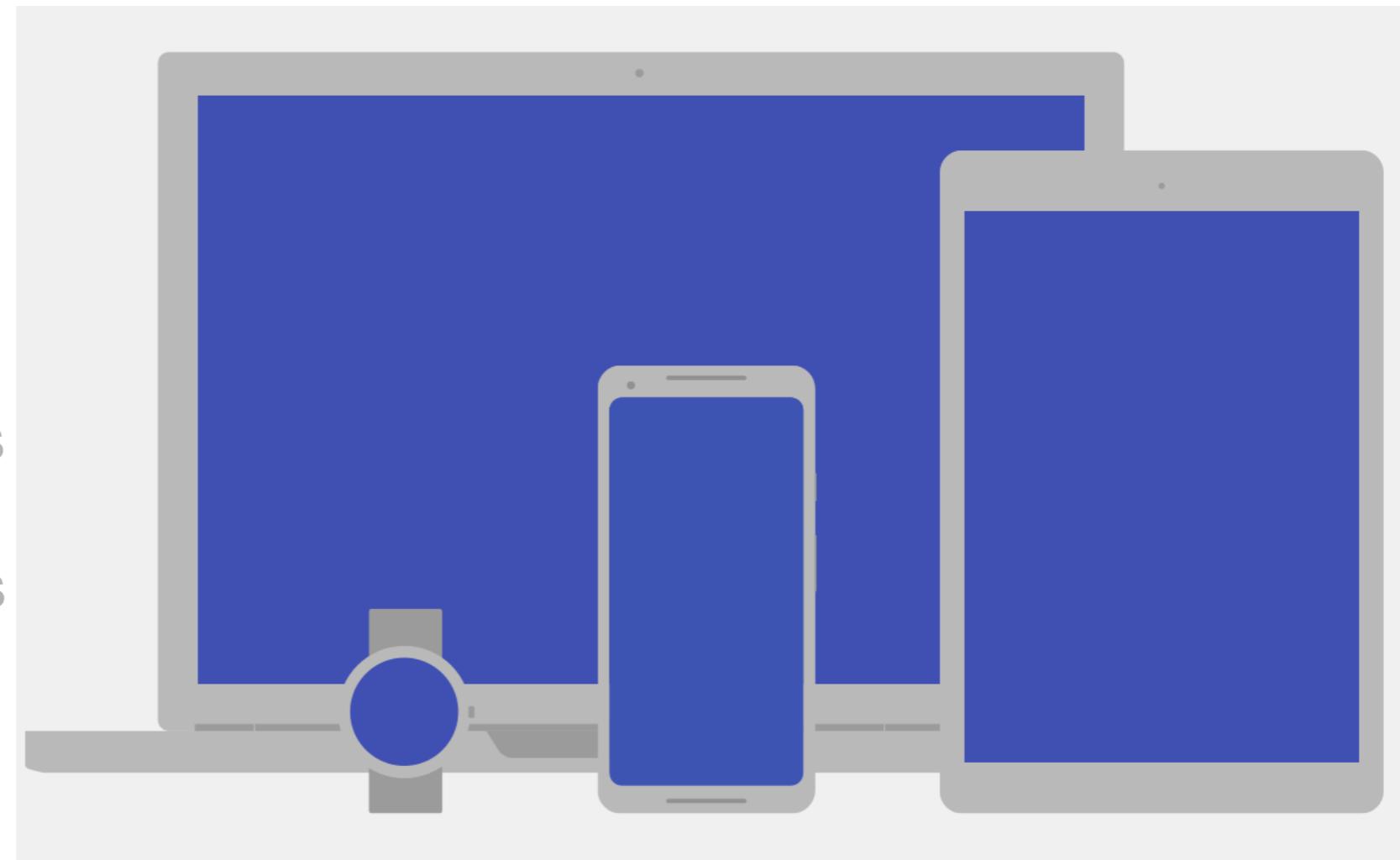
Supporting different screen sizes

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



Supporting different screen sizes

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



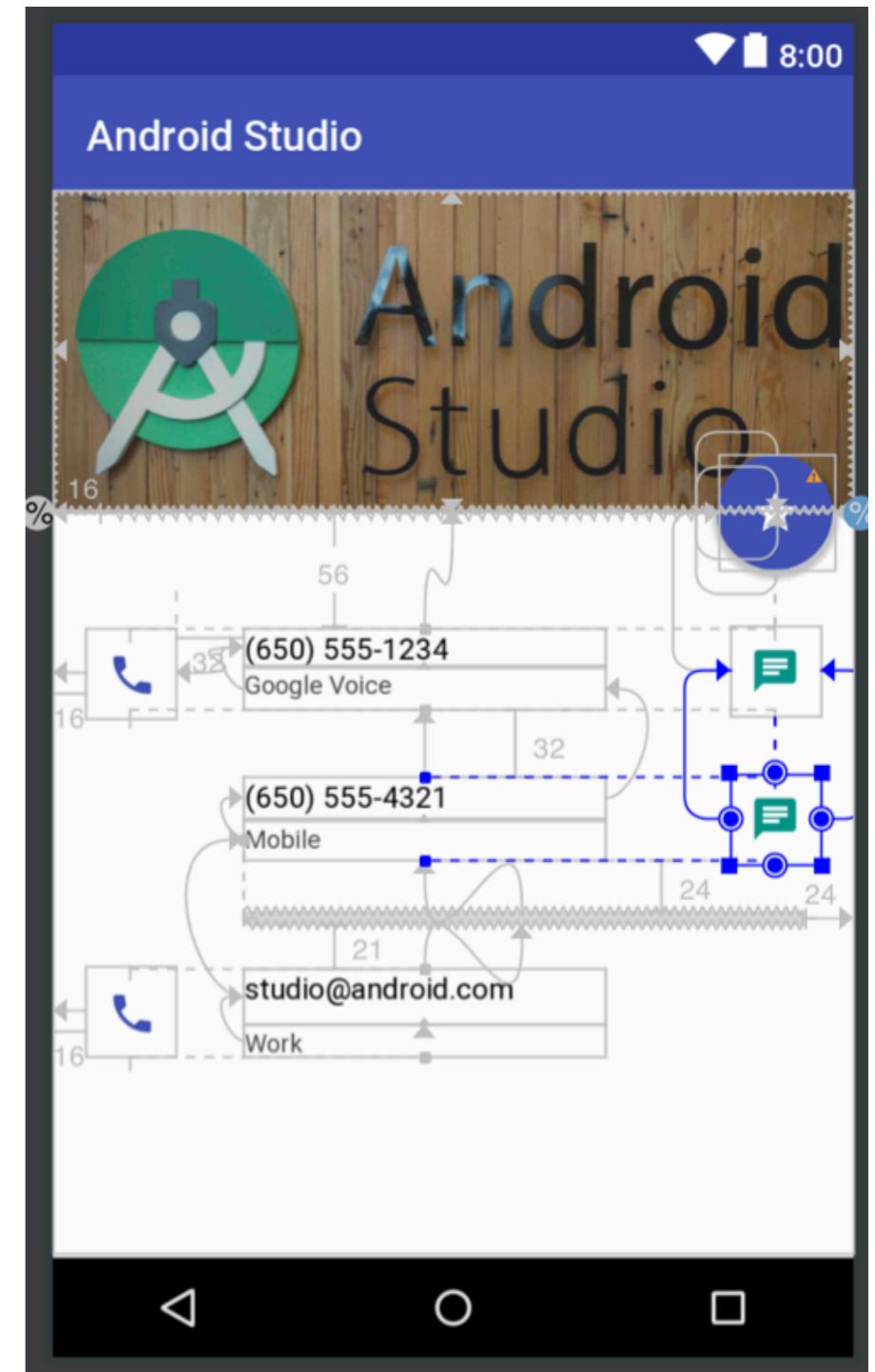
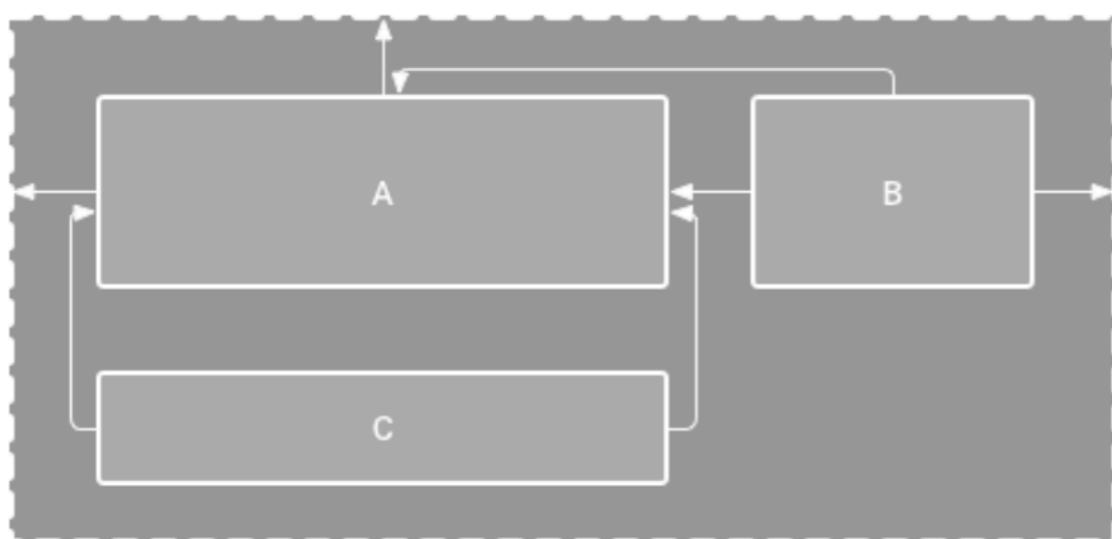
https://developer.android.com/guide/practices/screens_support

Flexible Layouts

ConstraintLayout

In module-level `gradle.build`:

```
repositories {  
    google()  
}  
  
dependencies {  
    implementation  
        'com.android.support.constraint:constraint-layout:1.1.2'  
}
```



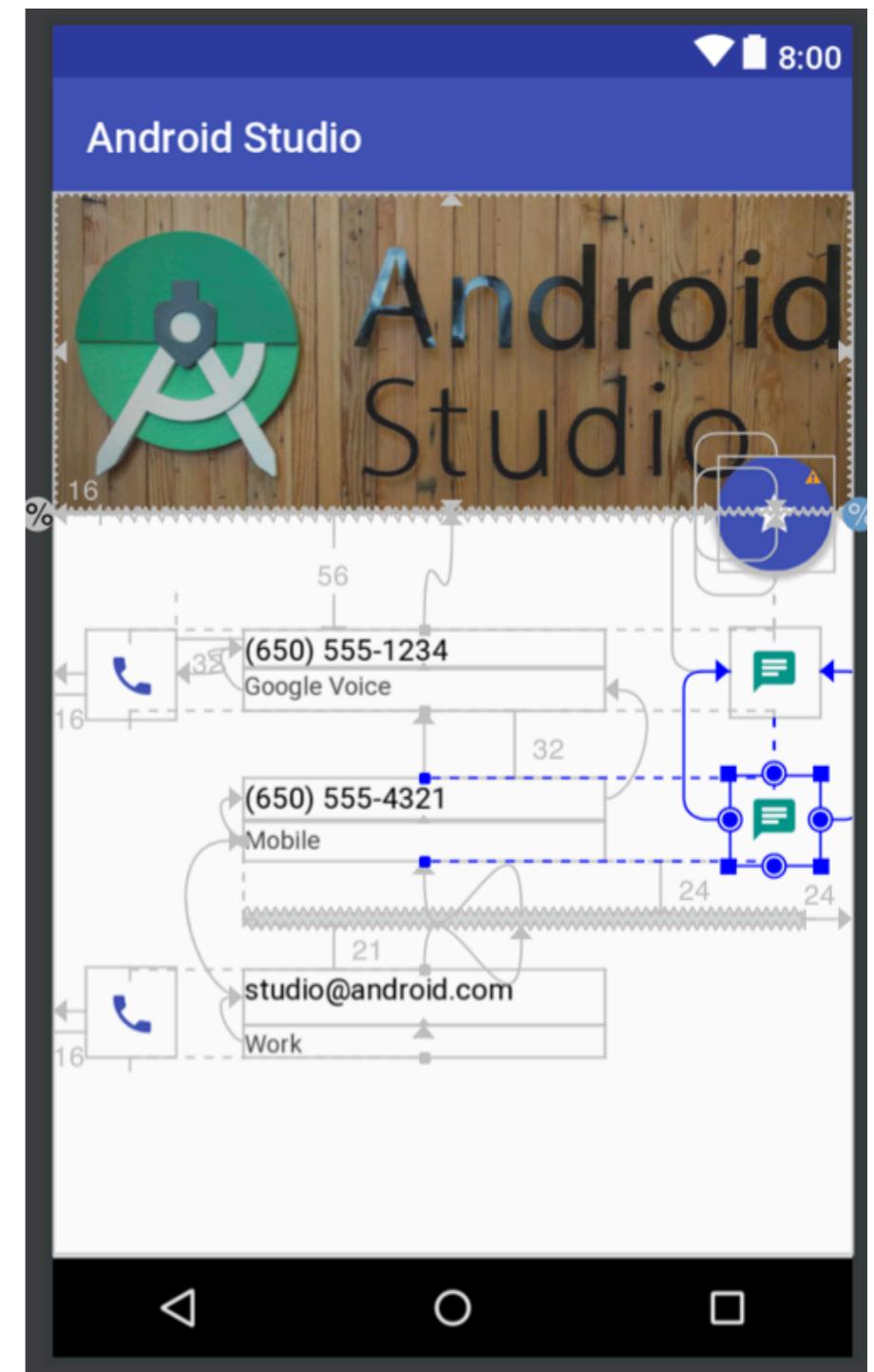
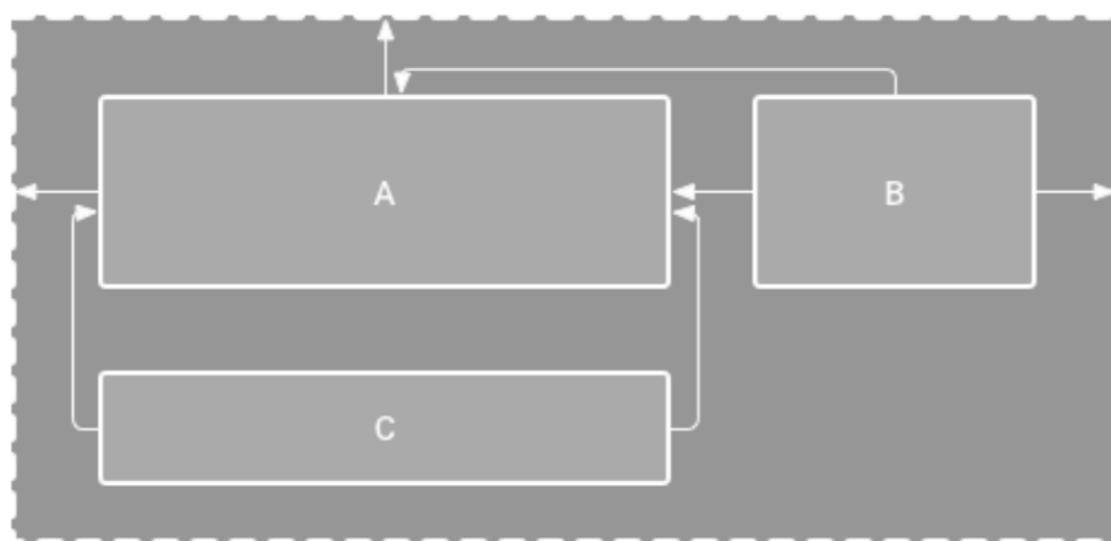
DEMO

Flexible Layouts

ConstraintLayout

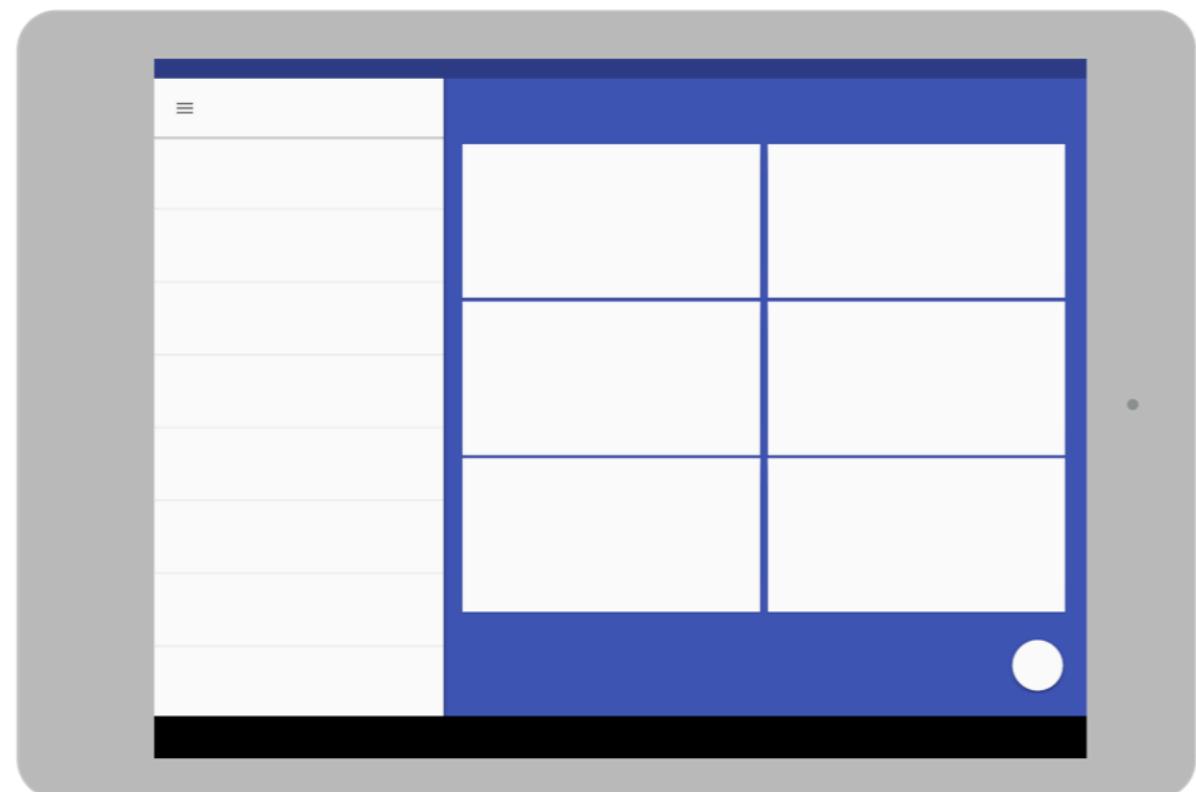
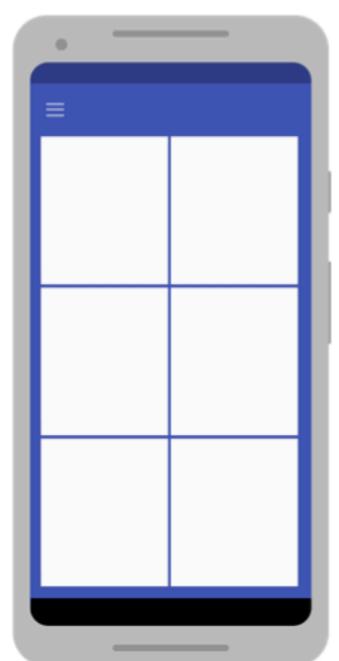
In module-level `gradle.build`:

```
repositories {  
    google()  
}  
  
dependencies {  
    implementation  
        'com.android.support.constraint:constraint-layout:1.1.2'  
}
```



Alternative layouts

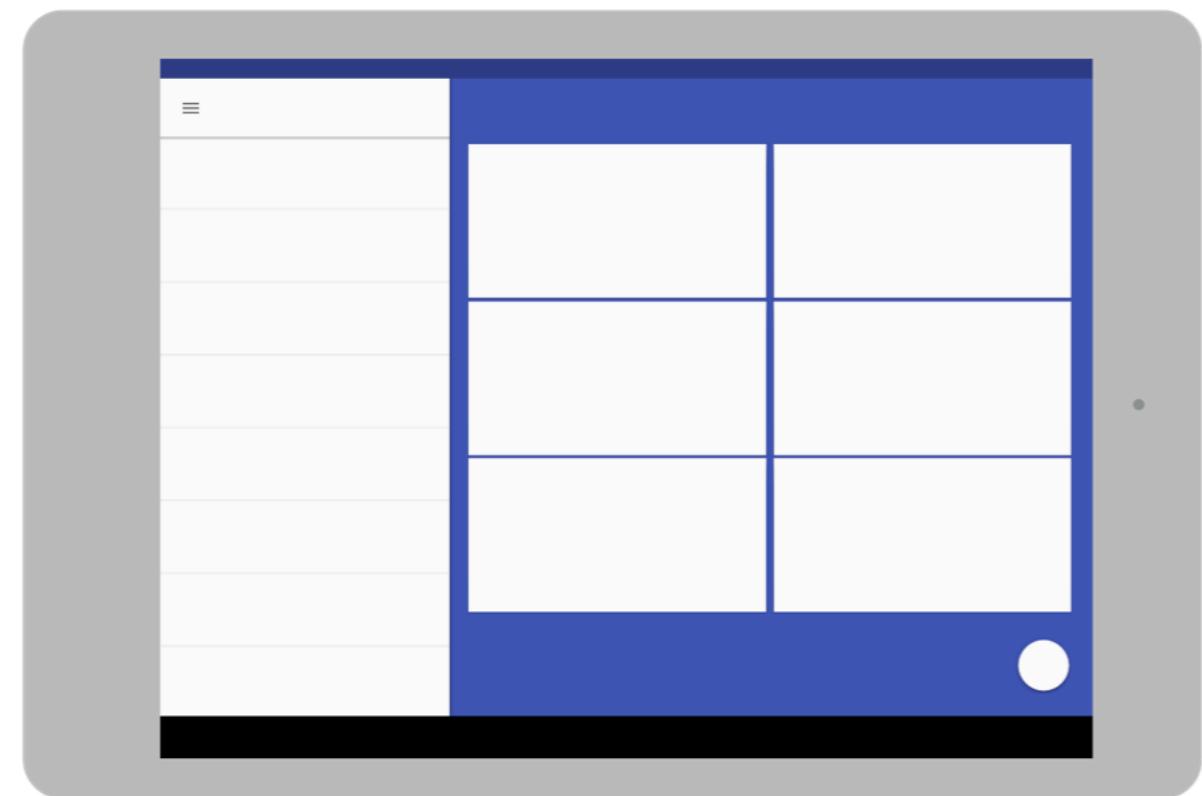
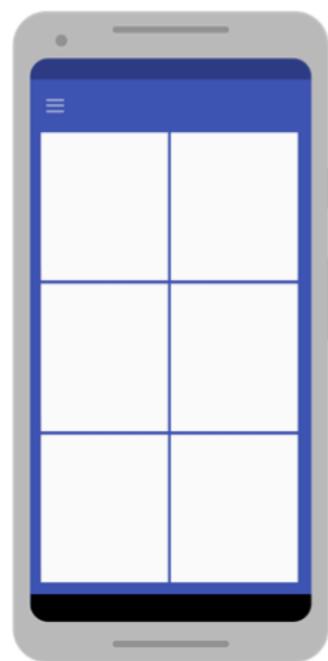
```
res/layout/main_activity.xml           # Default layout  
res/layout-land/main_activity.xml    # When in landscape mode  
res/layout-sw600dp/main_activity.xml  # For 7" tablets  
res/layout-sw600dp-land/main_activity.xml # For 7" tablets in landscape
```



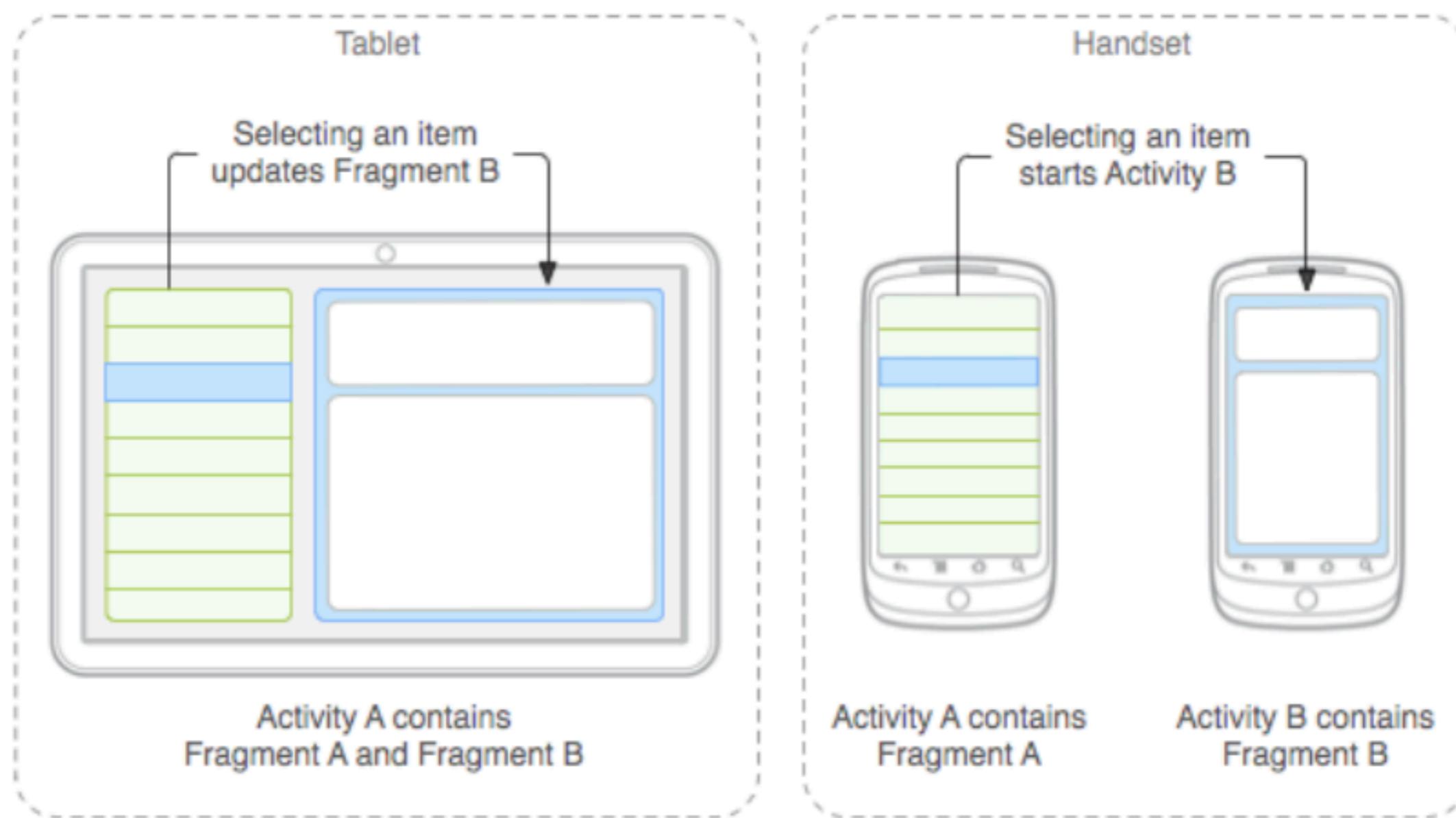
DEMO

Alternative layouts

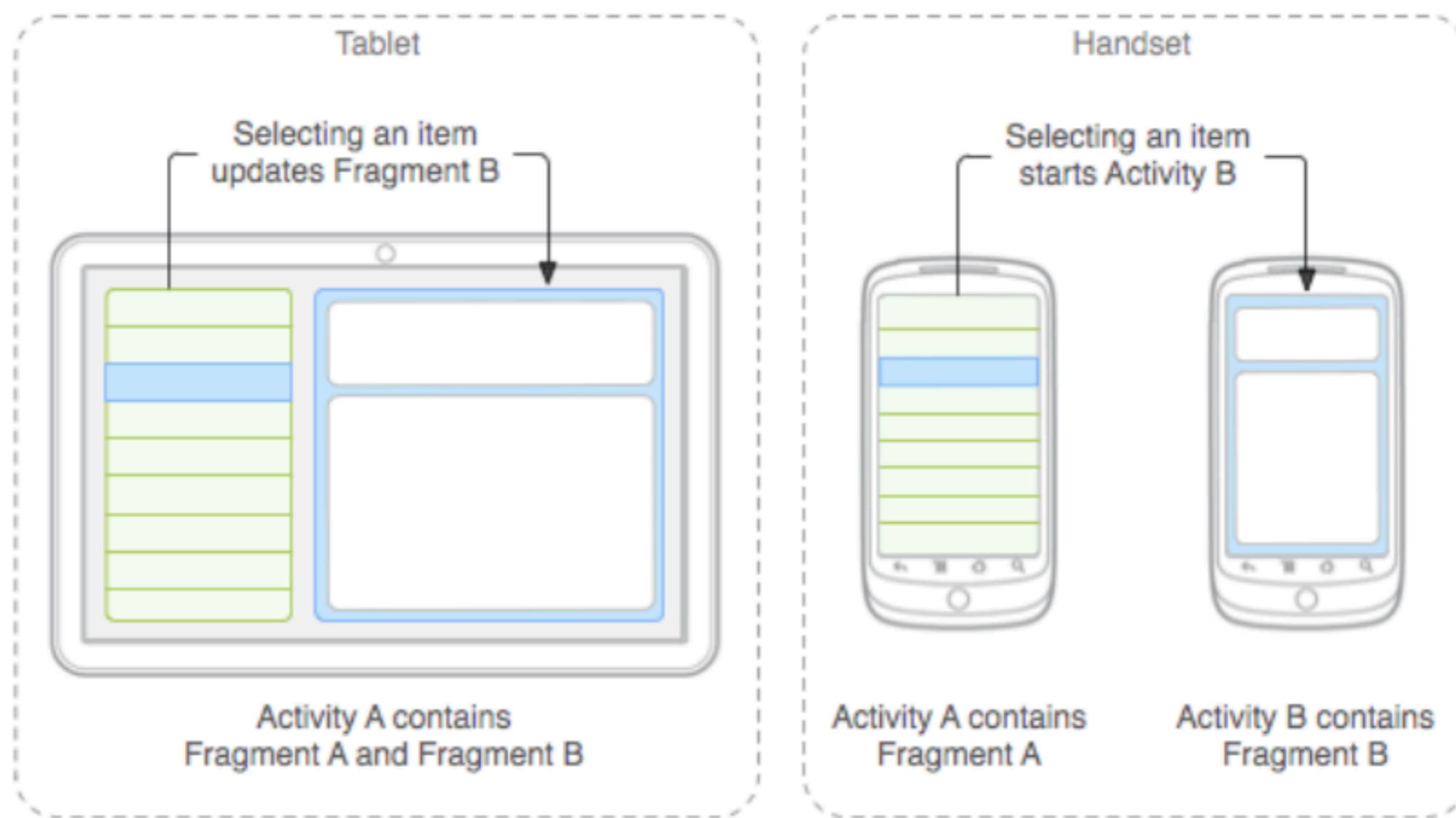
```
res/layout/main_activity.xml           # Default layout  
res/layout-land/main_activity.xml    # When in landscape mode  
res/layout-sw600dp/main_activity.xml # For 7" tablets  
res/layout-sw600dp-land/main_activity.xml # For 7" tablets in landscape
```



Building a Dynamic UI with Fragments

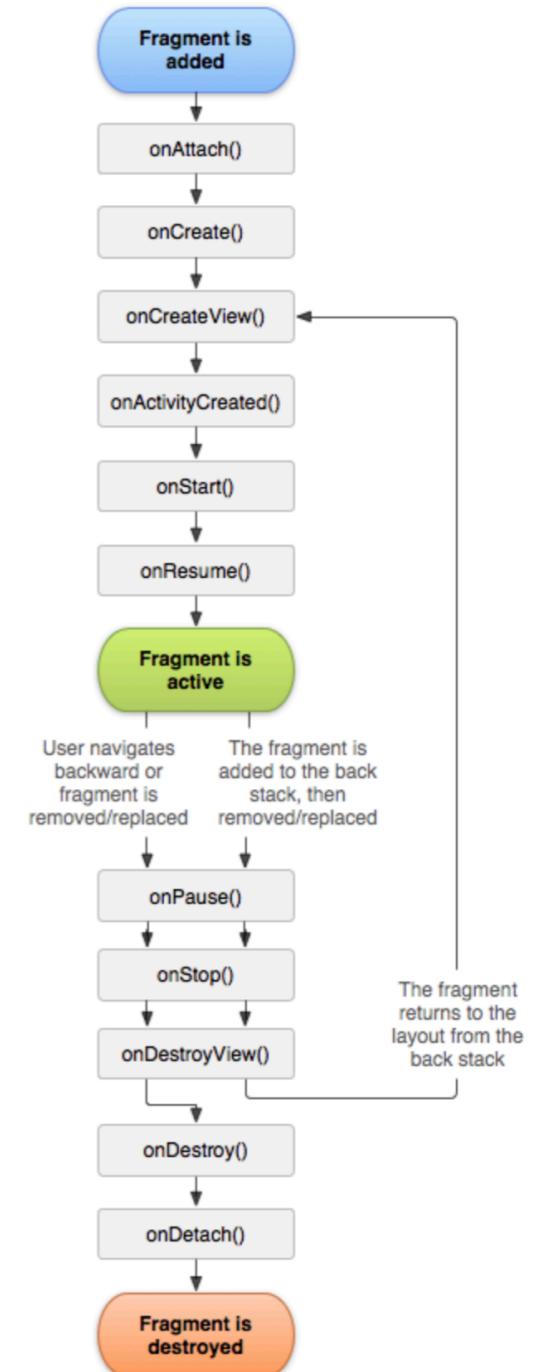


Building a Dynamic UI with Fragments



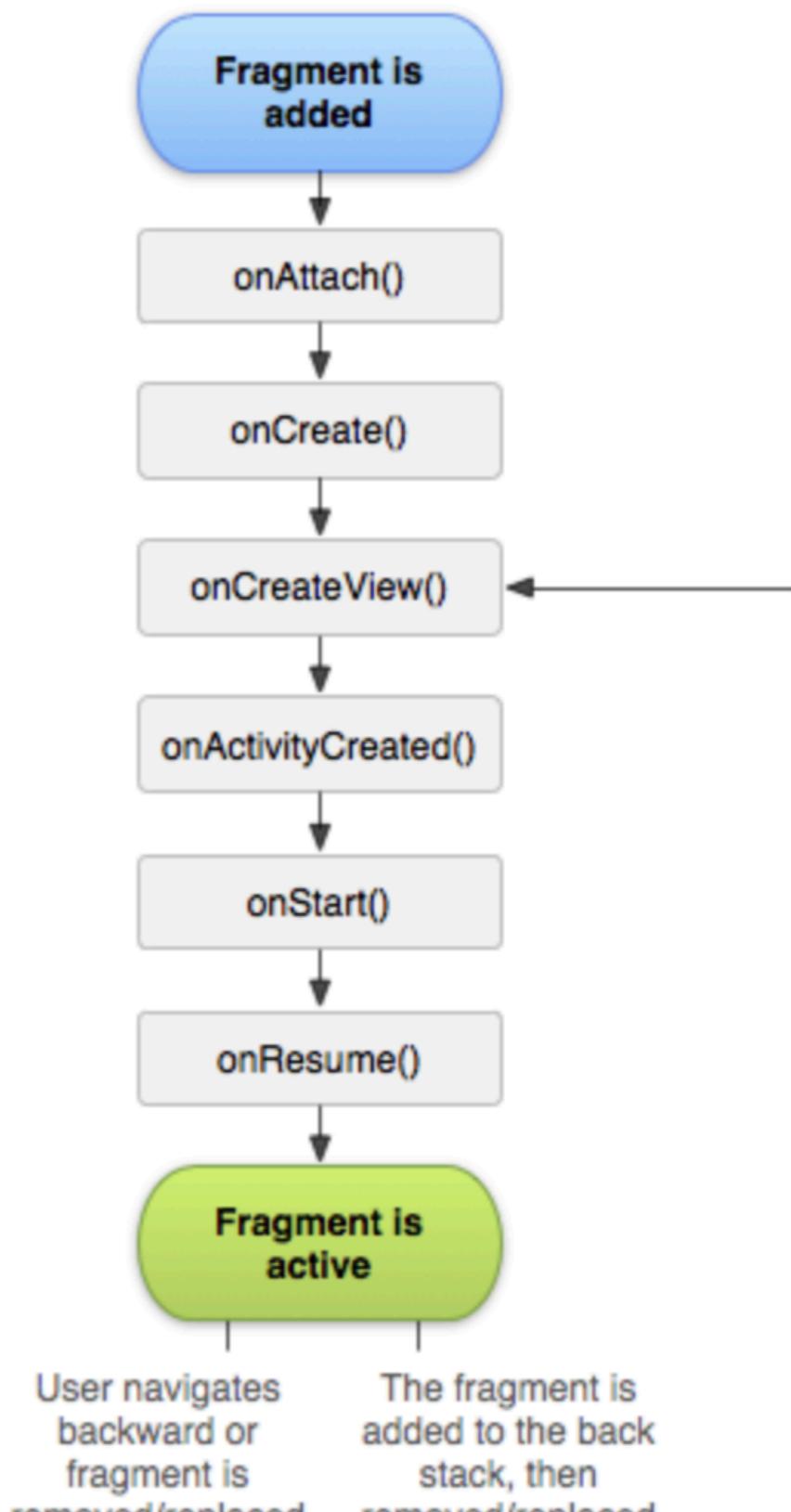
Creating a Fragment

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



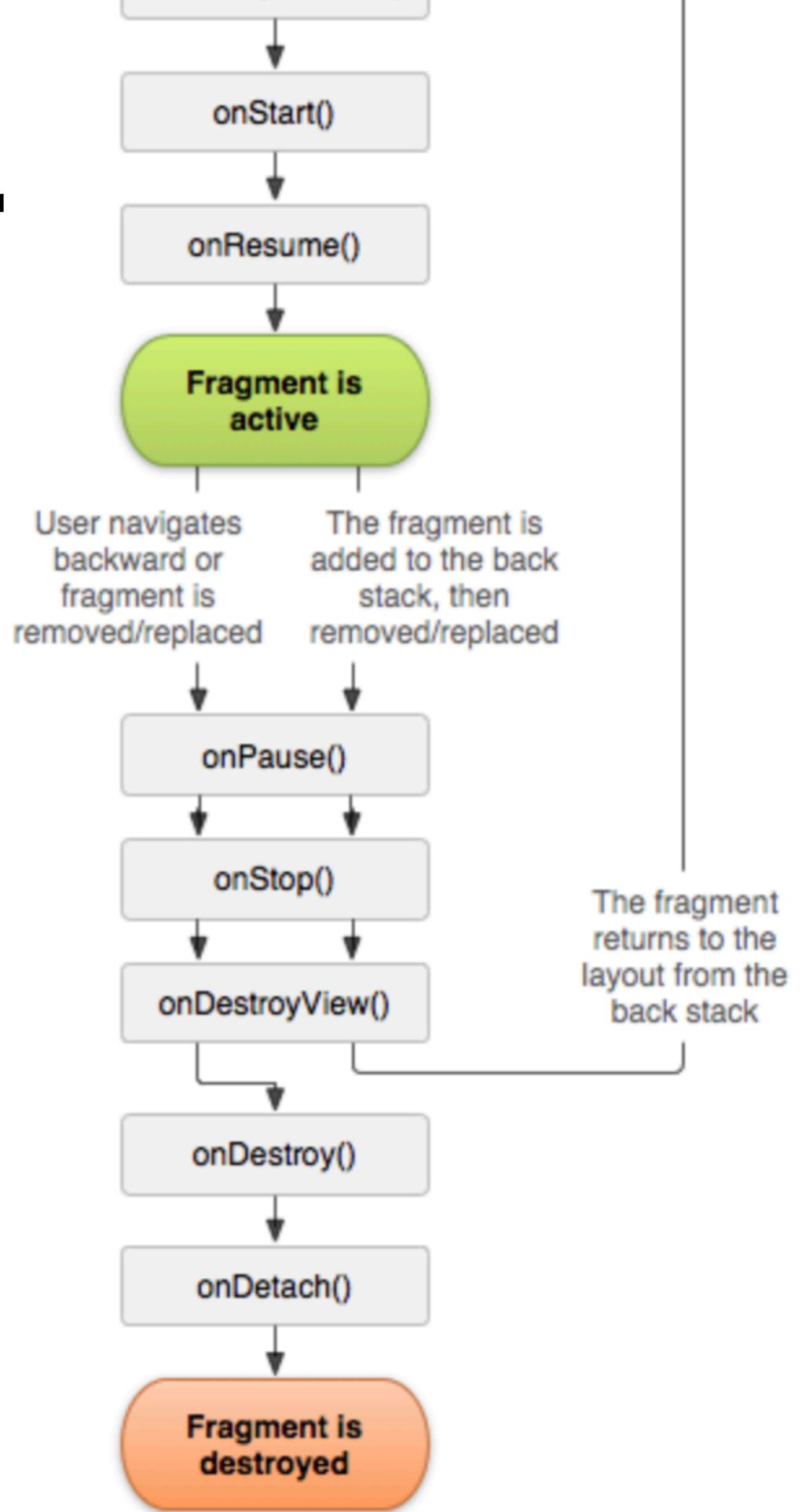
Creating a Fragment

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



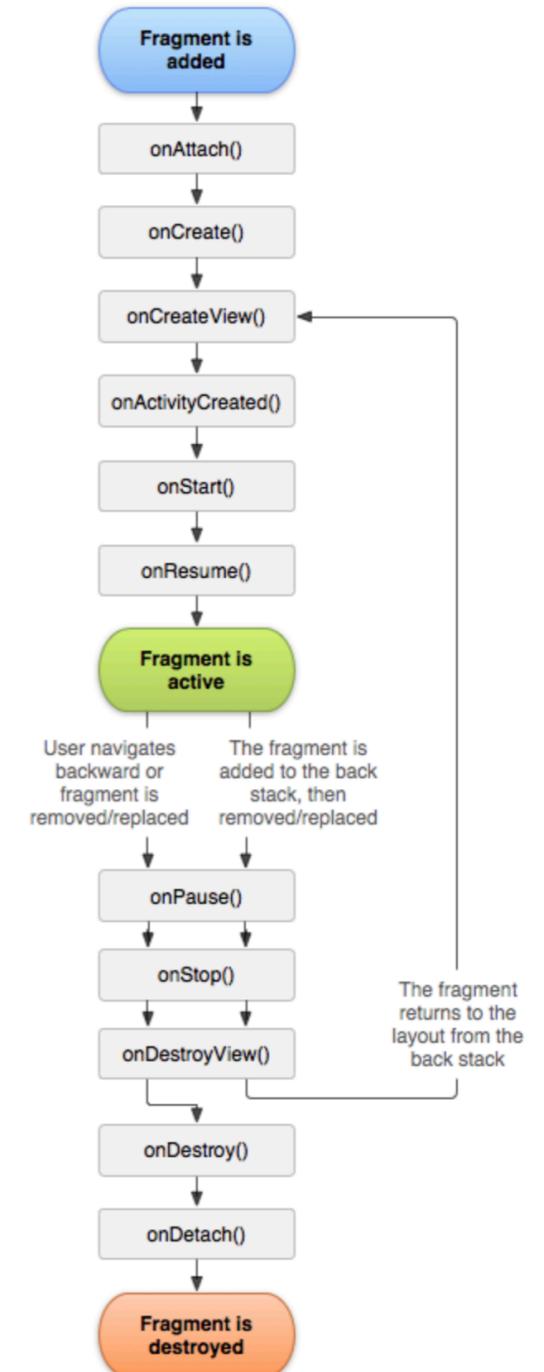
Creating a Fragment

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



Creating a Fragment

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



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)
    }
}
```

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)
    }
}
```

Creatina 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)  
    }  
}  
  
Declare the fragment inside the activity's layout file.  
<?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>
```

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()
```

DEMO

Creating a Fragment

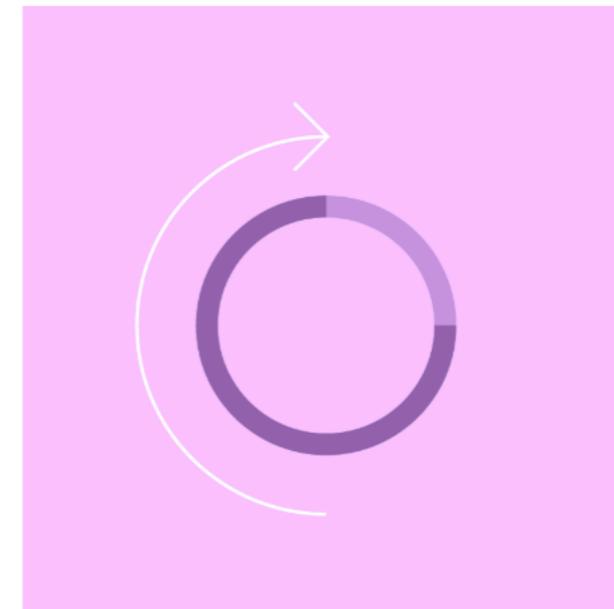
```
class ArticleListFragment : Fragment() {  
  
    override fun onCreateView(  
        inflater: LayoutInflater,  
        container: ViewGroup?,  
        savedInstanceState: Bundle?  
    ): View {  
        // Inflate the layout for this fragment  
        return inflater.inflate(R.layout.example_fragment,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

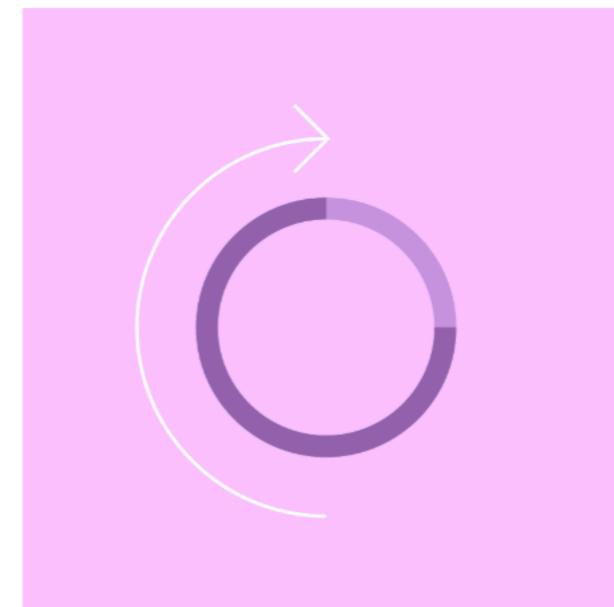
Progress Indicators

- ProgressBar
- RatingBar
- SeekBar



Progress Indicators

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

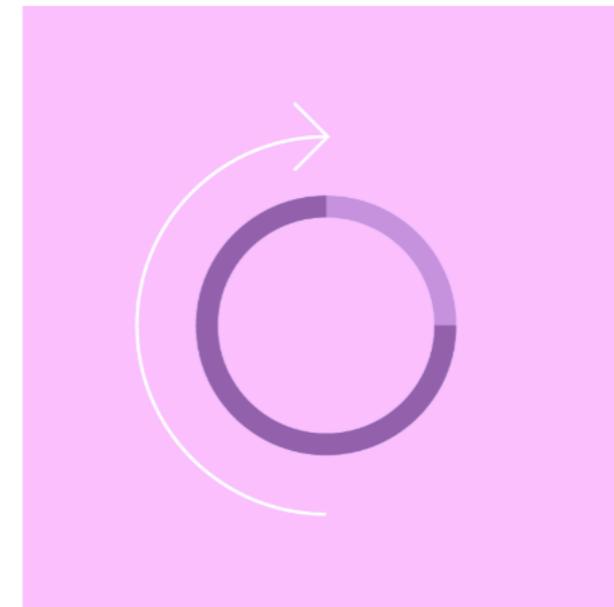


Progress Indicators

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

```
//before starting the action  
indicator.setVisibility(View.VISIBLE)
```

```
//when the action is done  
indicator.setVisibility(View.GONE)
```

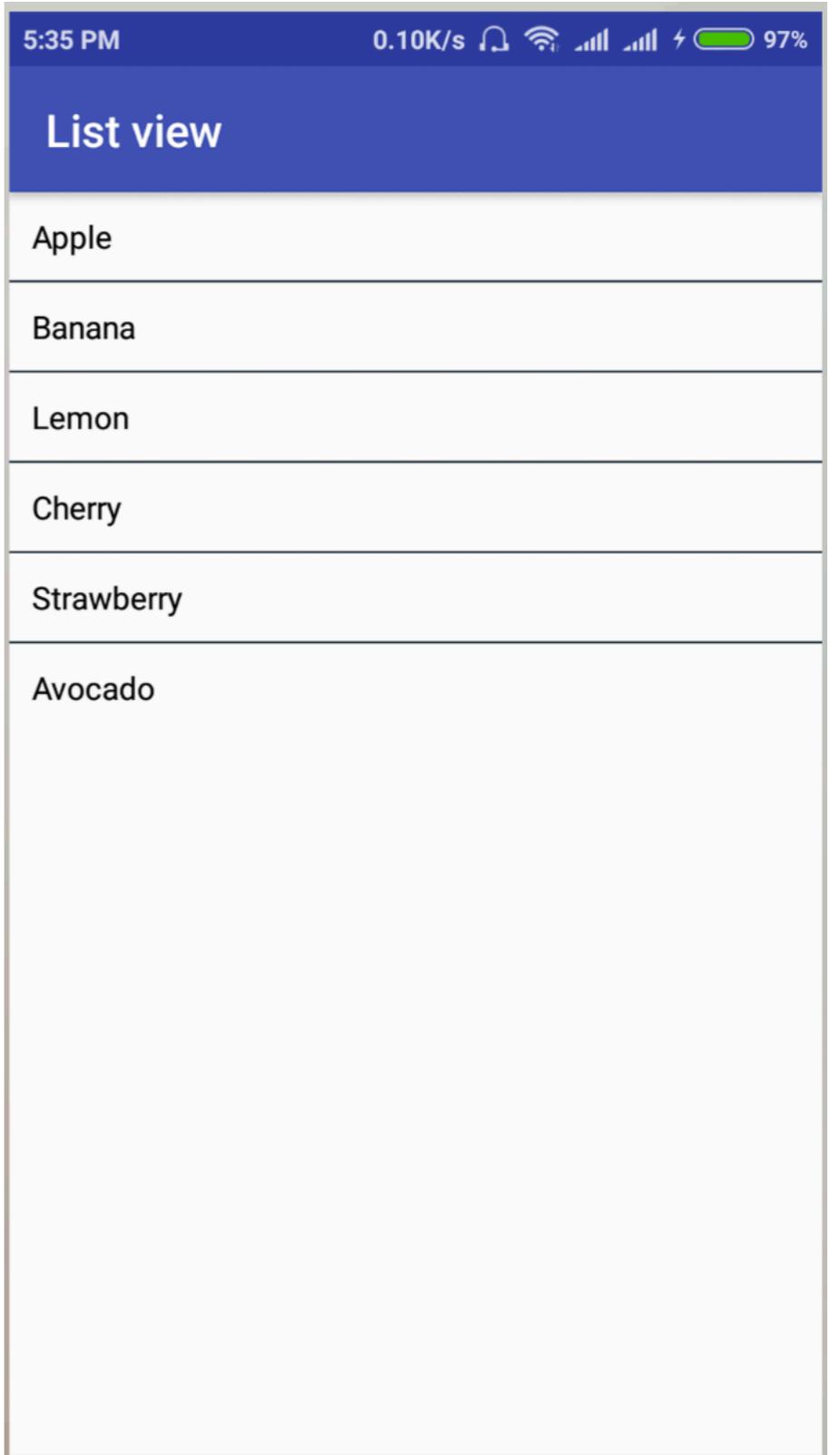


<https://developer.android.com/reference/android/widget/ProgressBar>

Lists

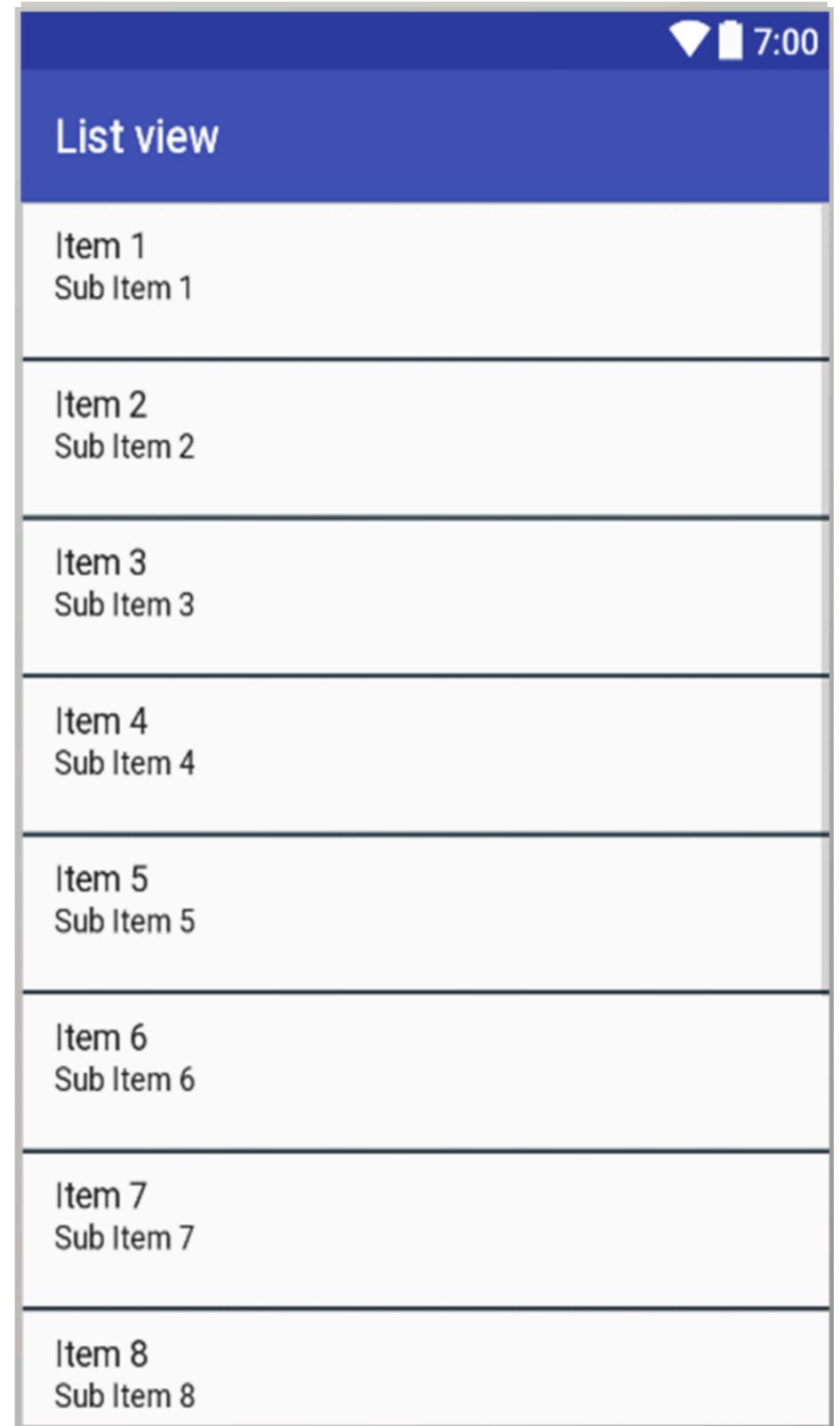
Lists

- ListView



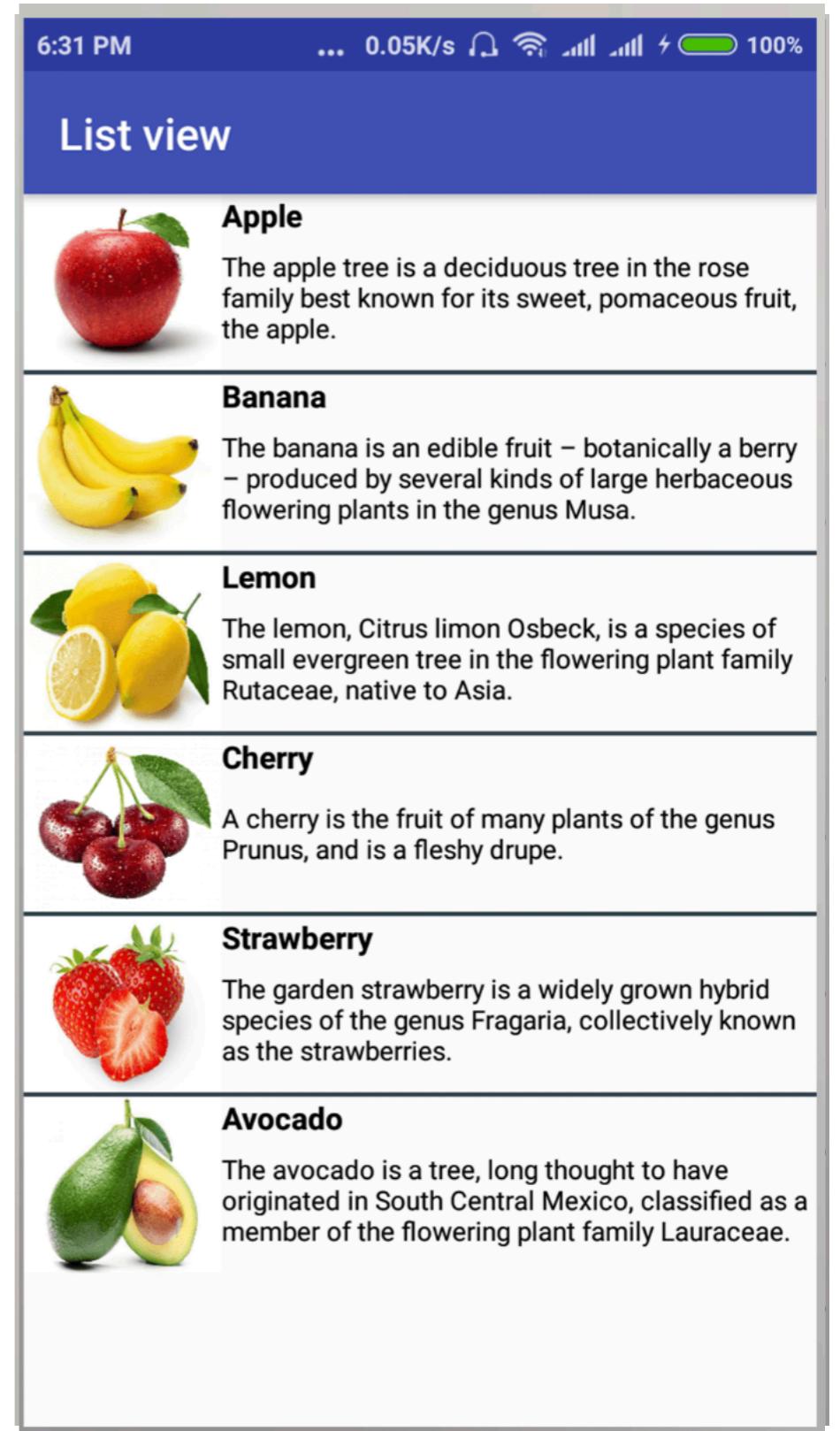
Lists

- ListView
- RecyclerView



Lists

- ListView
- RecyclerView



ListView

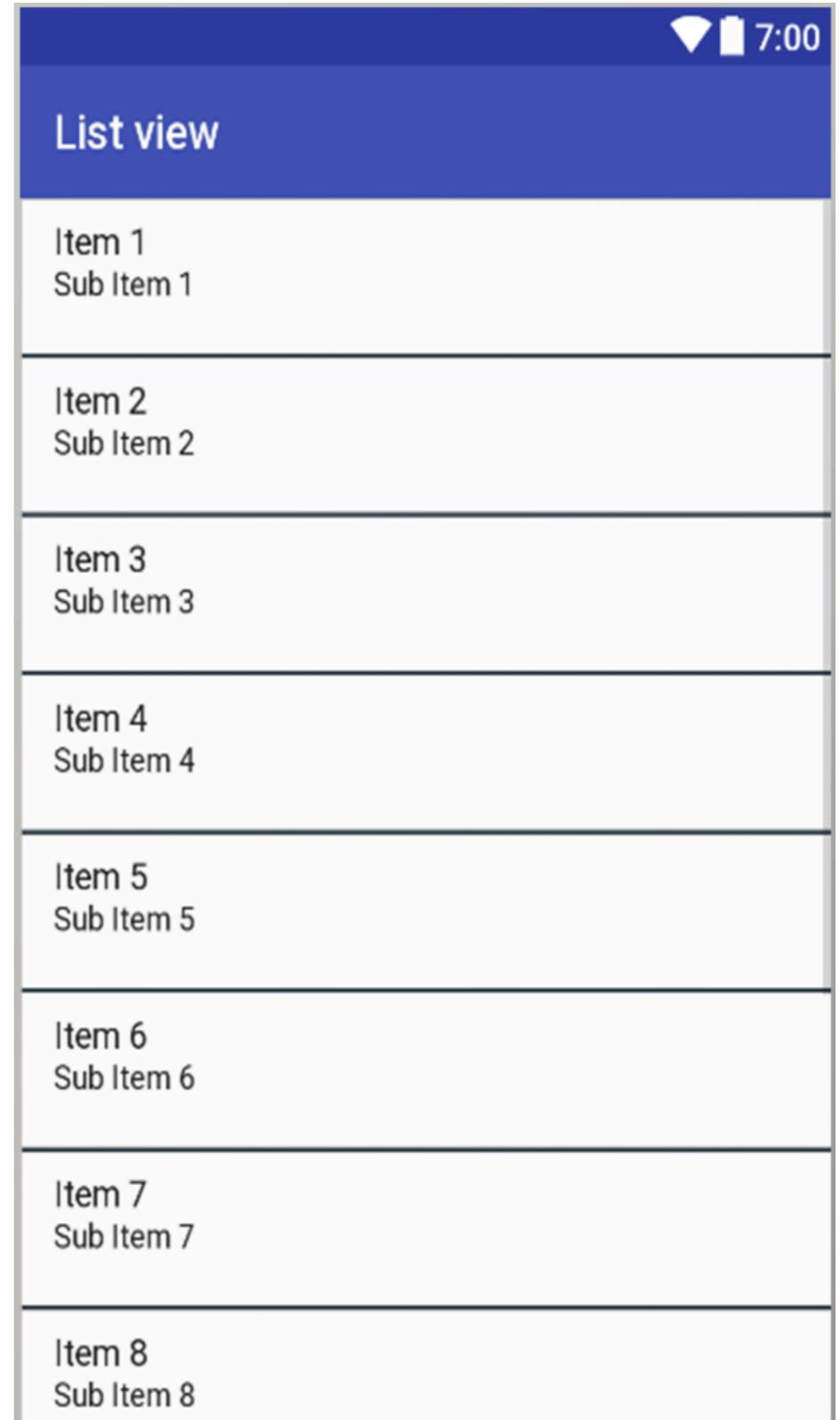
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="8dp"
    android:paddingRight="8dp">
  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

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="@+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>
```

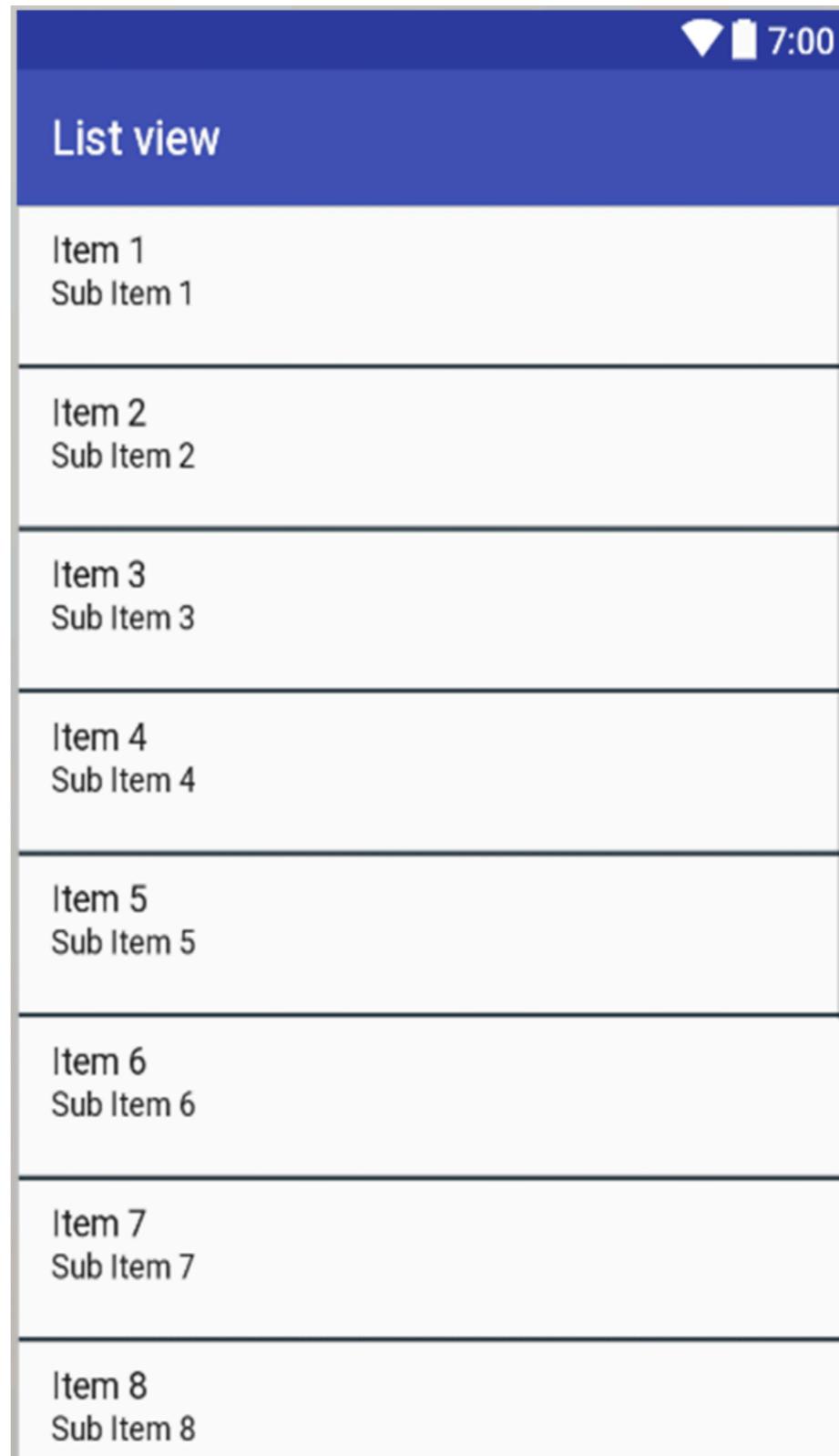


ListView

```
<?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="@+id/myList"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>
</LinearLayout>

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

myList.adapter = arrayAdapter
```

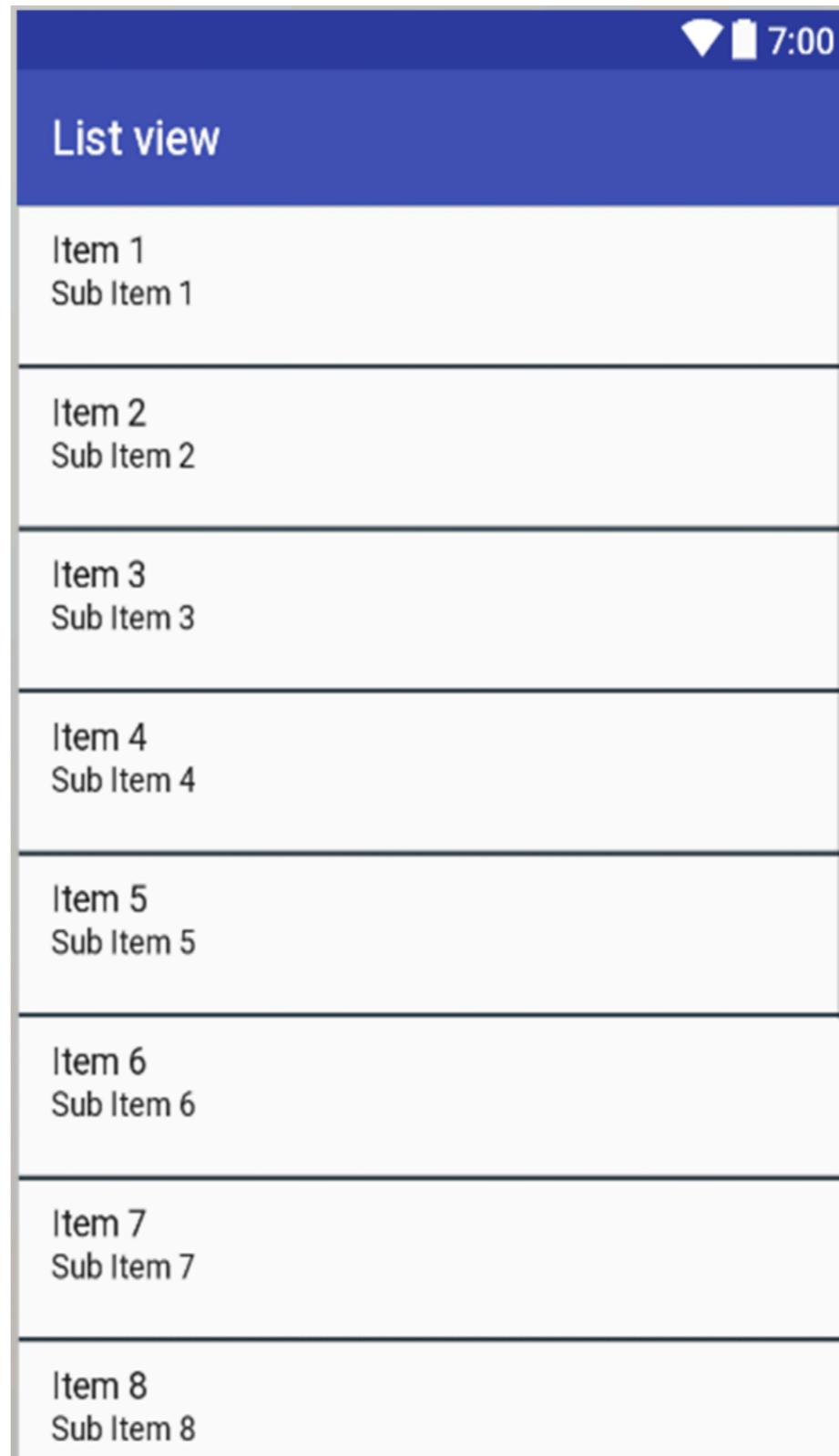


ListView

```
<?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="@+id/myList"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>
</LinearLayout>

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

myList.adapter = arrayAdapter
```



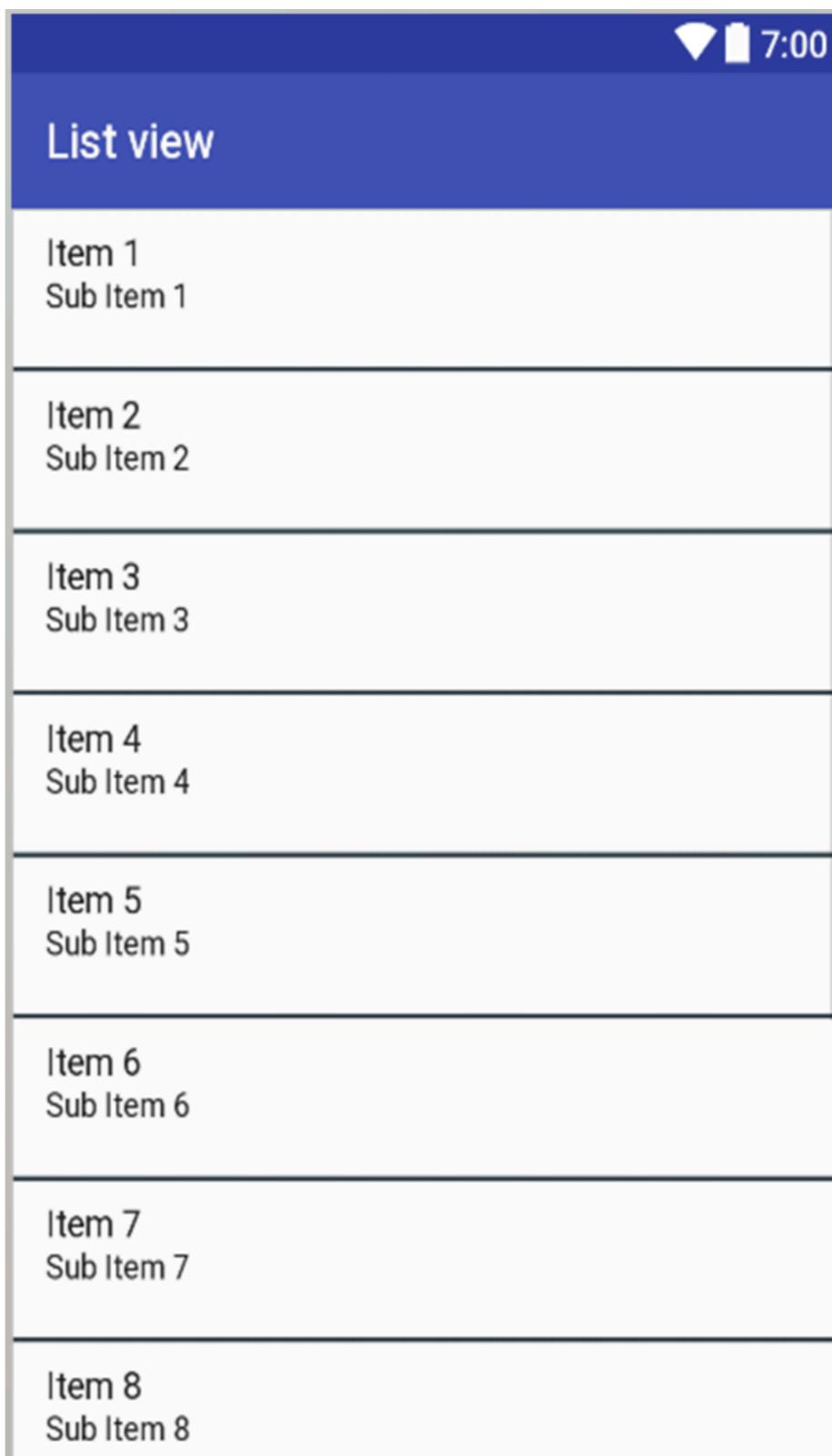
DEMO

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="@+id/myList"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>
</LinearLayout>

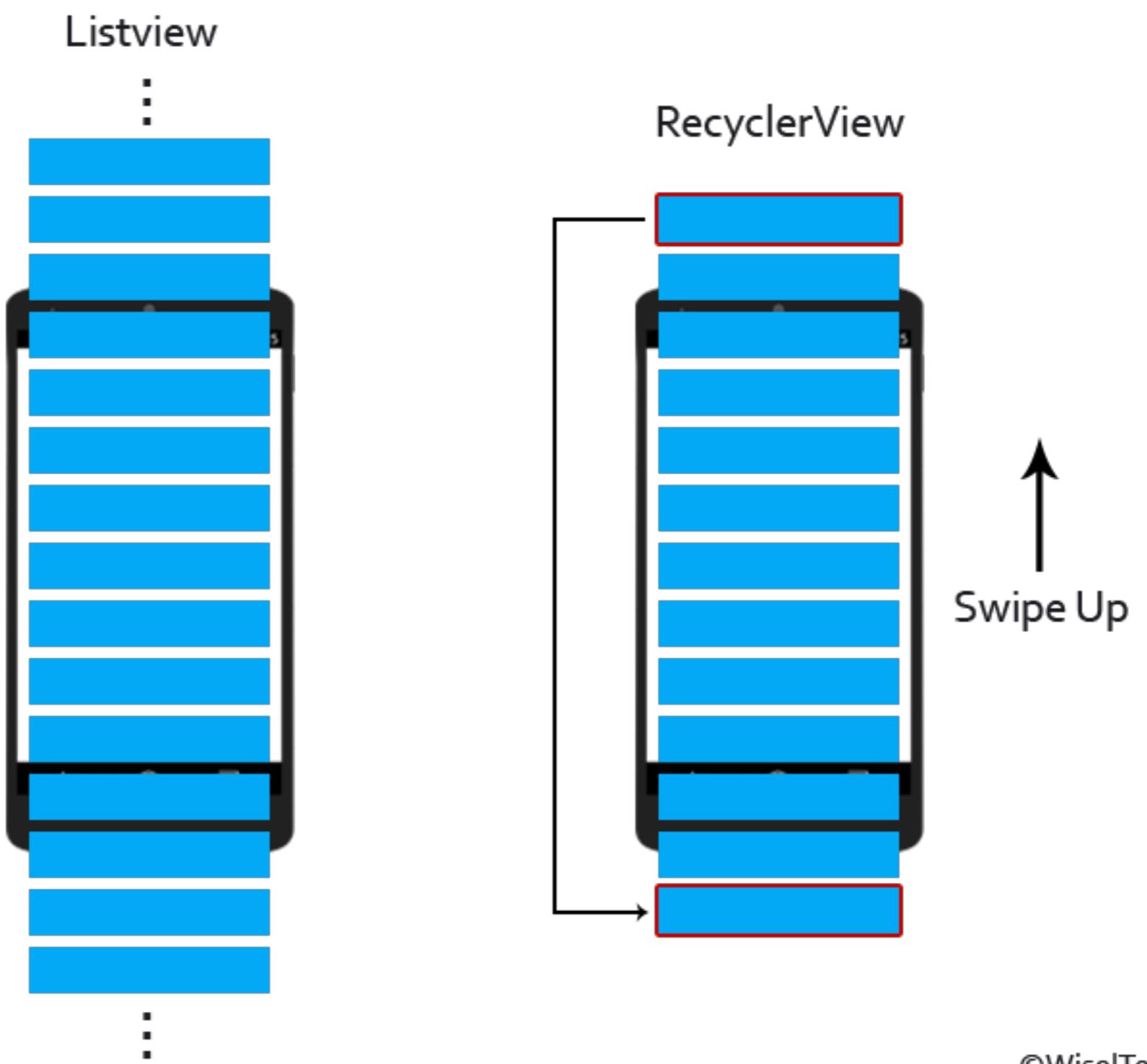
val arrayAdapter = ArrayAdapter<String>(this,
    android.R.layout.simple_list_item_1, arrayList)

myList.adapter = arrayAdapter
```



RecyclerView

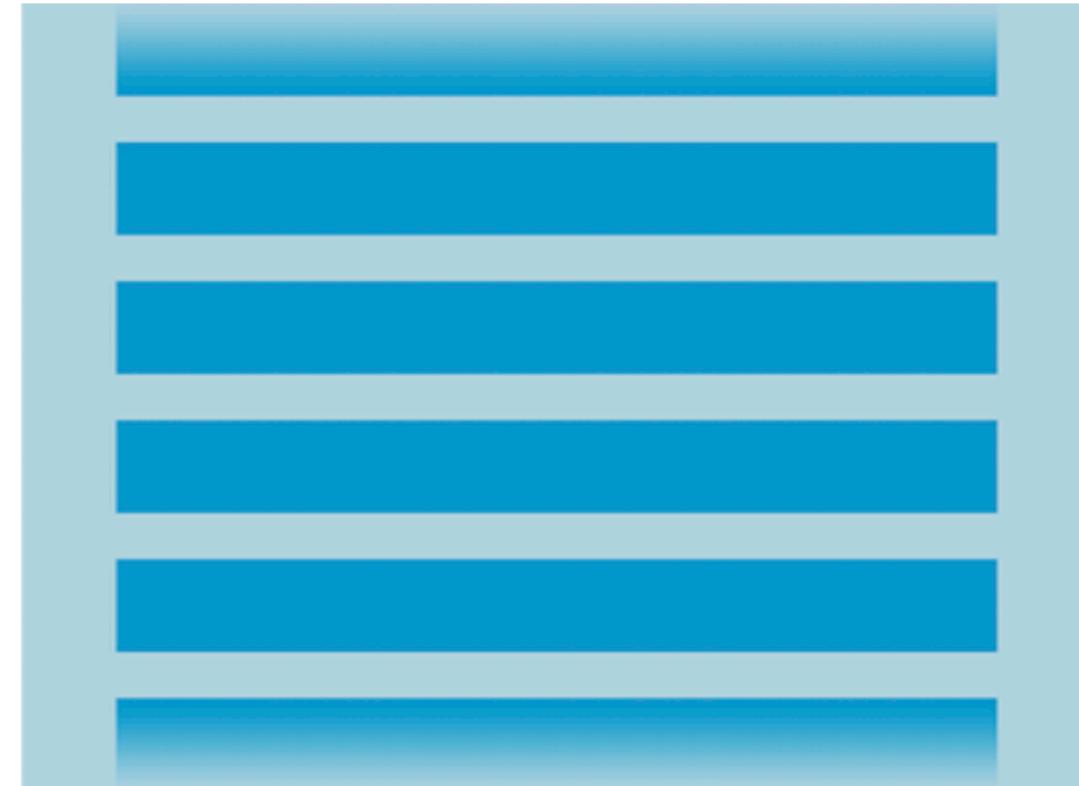
- ListView
- RecyclerView



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"/>

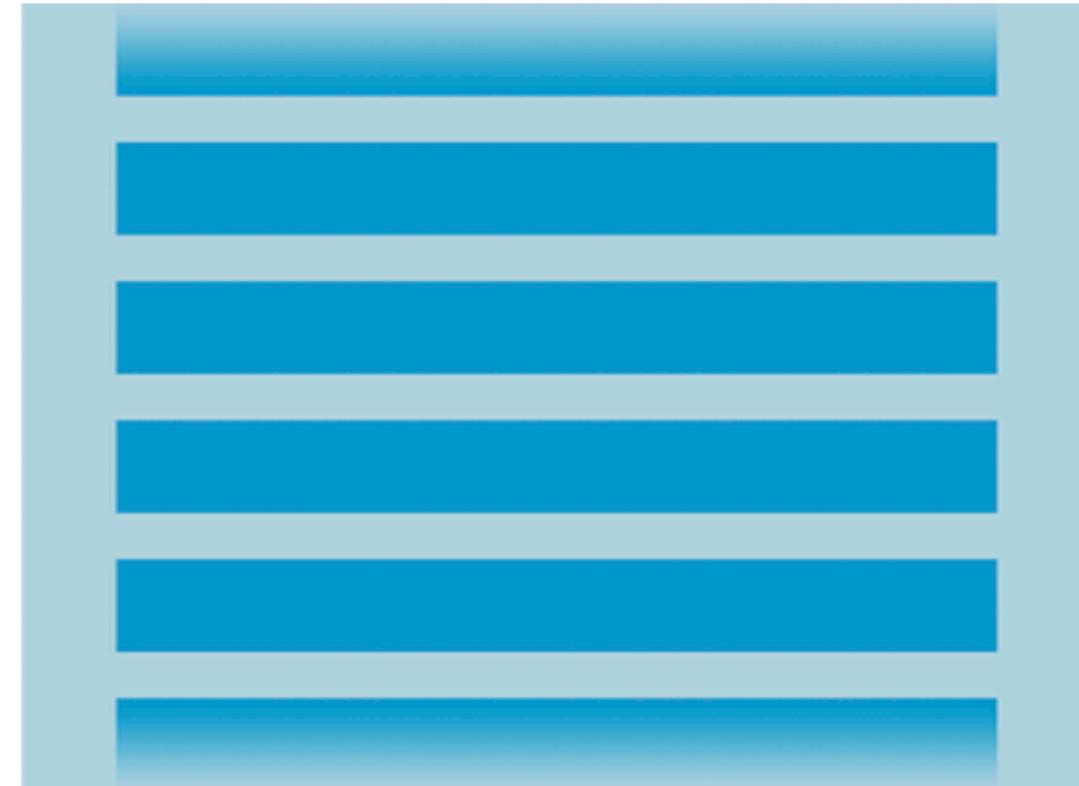
class MyActivity : Activity() {
    private lateinit var recyclerView: RecyclerView
    private lateinit var viewAdapter: RecyclerView.Adapter<*>
    private lateinit var viewManager: RecyclerView.LayoutManager
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        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
        }
    }
    // ...
}
```



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"/>

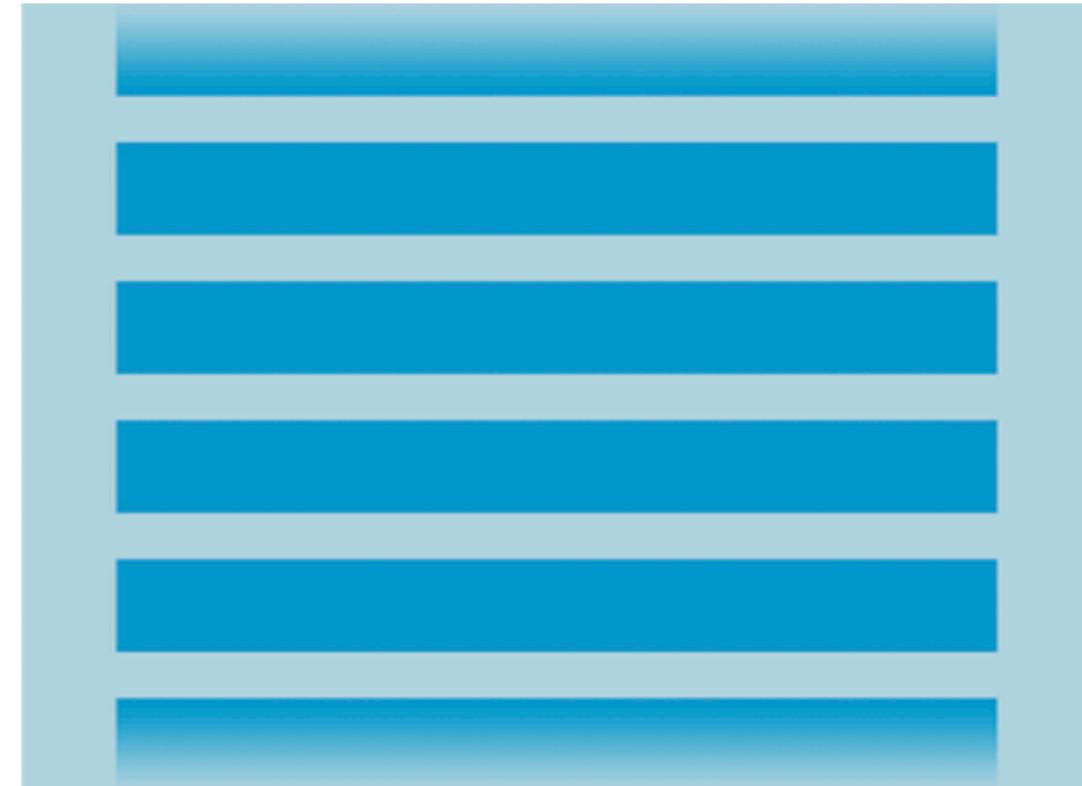
class MyActivity : Activity() {
    private lateinit var recyclerView: RecyclerView
    private lateinit var viewAdapter: RecyclerView.Adapter<*>
    private lateinit var viewManager: RecyclerView.LayoutManager
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        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
        }
    }
    // ...
}
```



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
        }
    }
    // ...
}
```



RecyclerView.Adapter

```
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  
}
```



DEMO

RecyclerView.Adapter

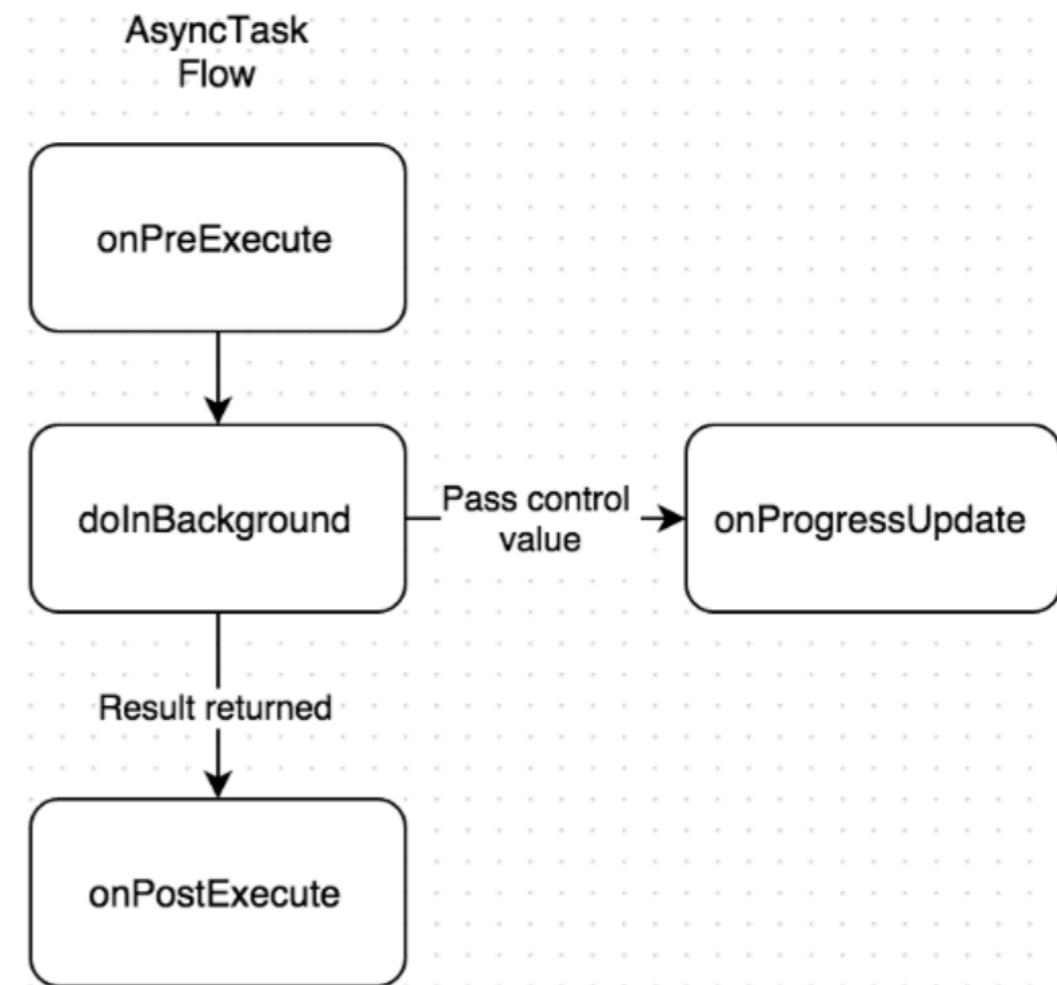
```
class MyAdapter(private val myDataset: Array<String>) :  
    RecyclerView.Adapter<MyAdapter.MyViewHolder>() {  
  
    class MyViewHolder(val textView: TextView) : RecyclerView.ViewHolder(textView)  
  
    override fun onCreateViewHolder(parent: ViewGroup,  
                                     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

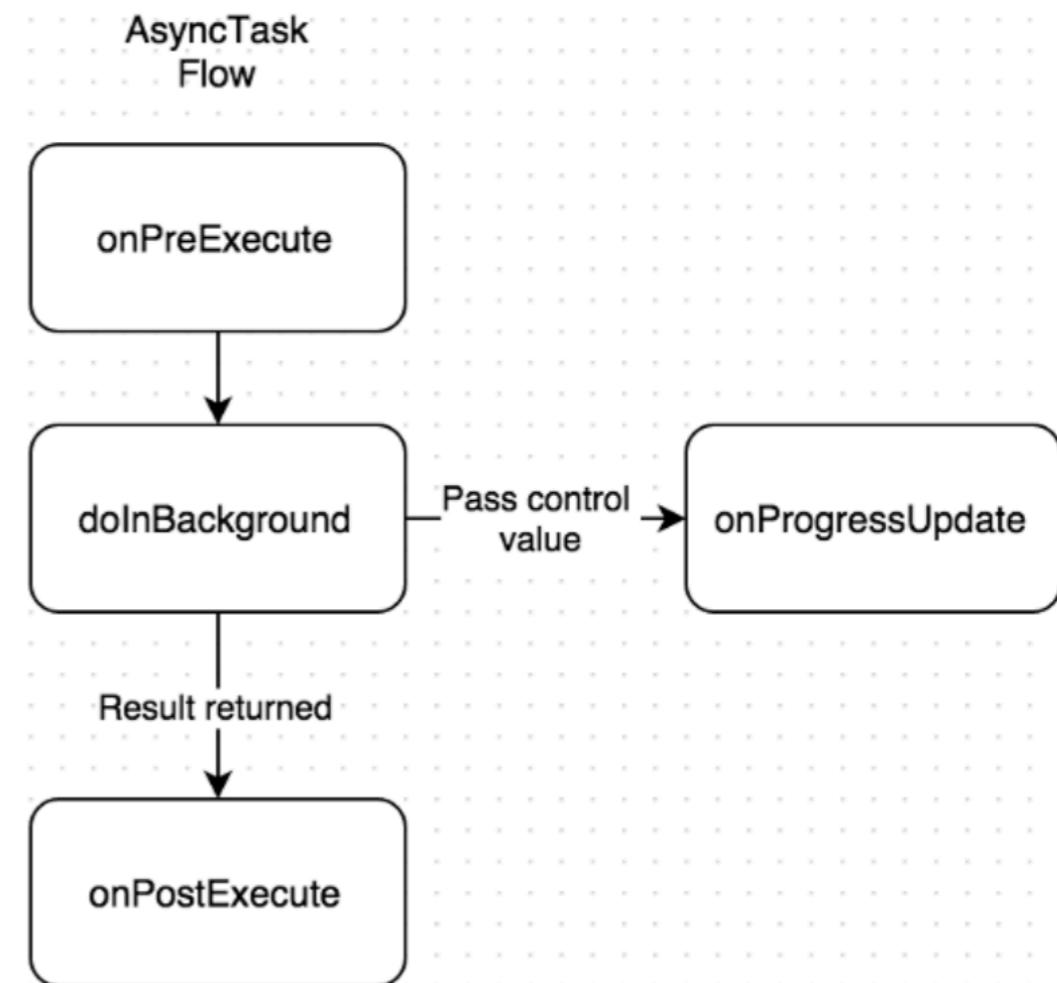
```
class SomeTask():  
    AsyncTask<Void, Int, String>() {  
        override fun doInBackground(  
            vararg params: Void?): String? {  
            // ...  
        }  
  
        override fun onPreExecute() {  
            super.onPreExecute()  
            // ...  
        }  
  
    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



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.onPostExecute(result)  
            // ...  
        }  
    }
```

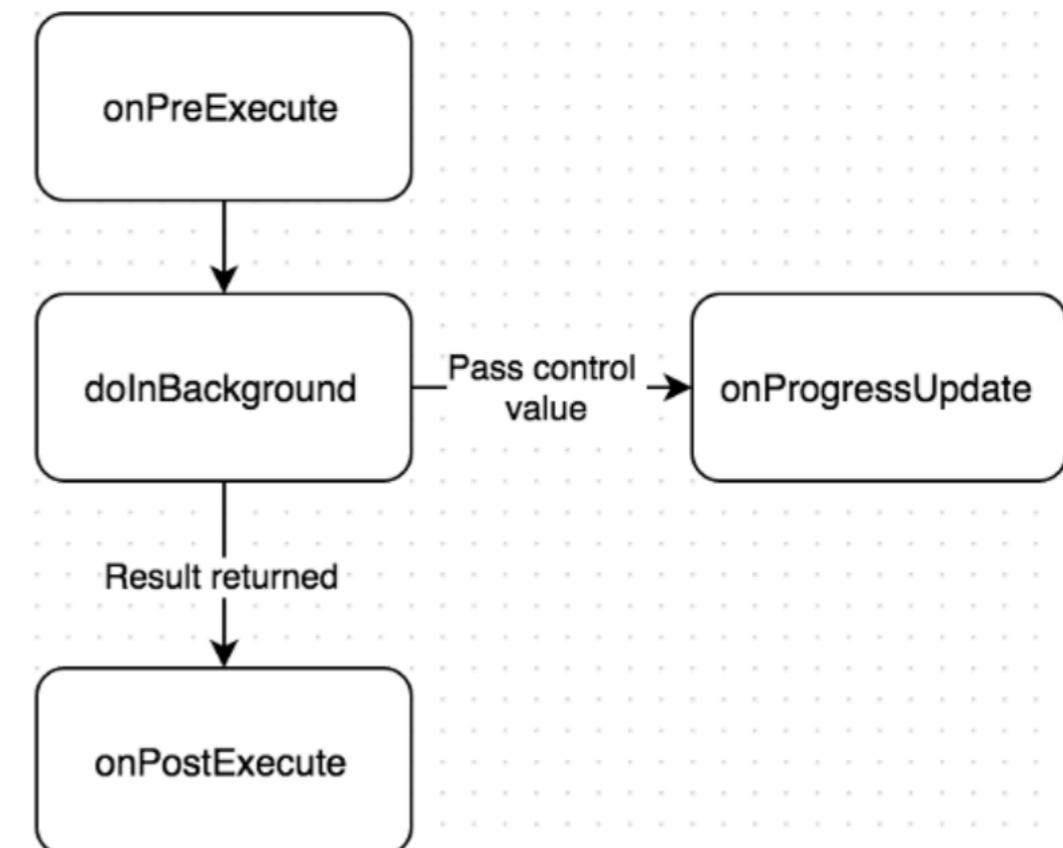


AsyncTask

BackgroundThread

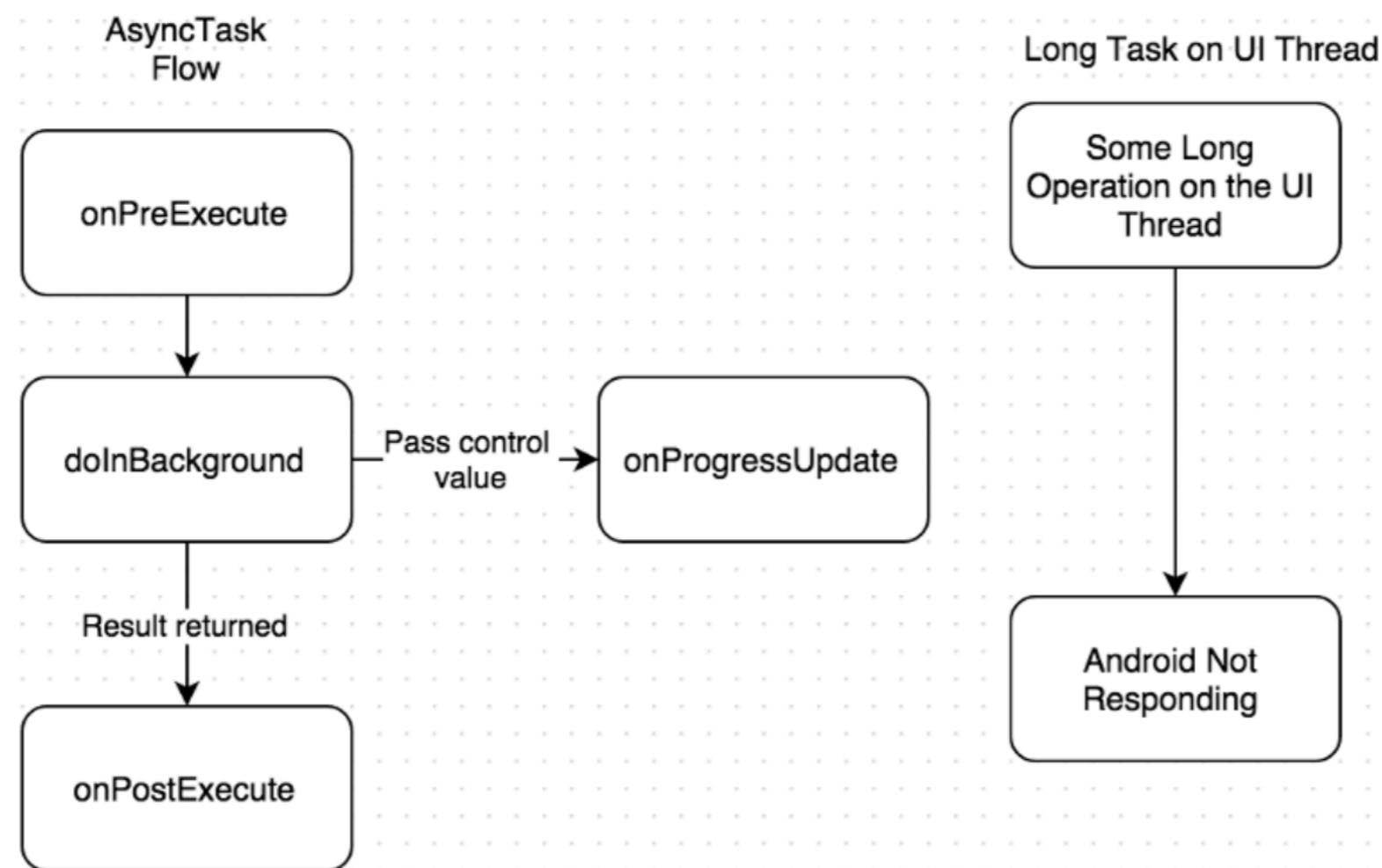
```
class SomeTask():  
    AsyncTask<Void, Int, String>() {  
        override fun doInBackground(  
            vararg params: Void?): String? {  
            // ...  
        }  
  
        override fun onPreExecute() {  
            super.onPreExecute()  
            // ...  
        }  
  
    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

AsyncTask
Flow



UiThread

AsyncTask

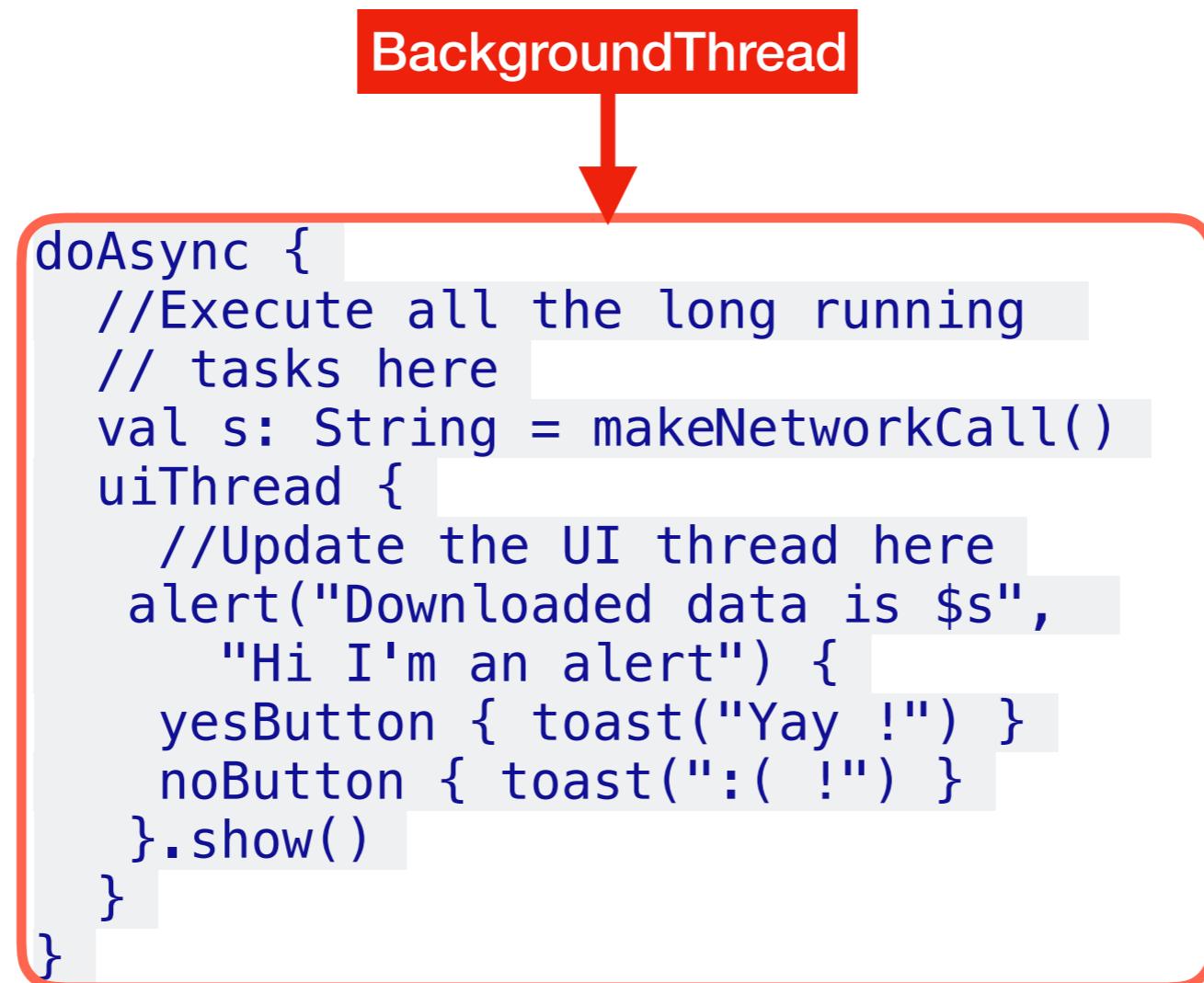


<https://developer.android.com/reference/android/os/AsyncTask>

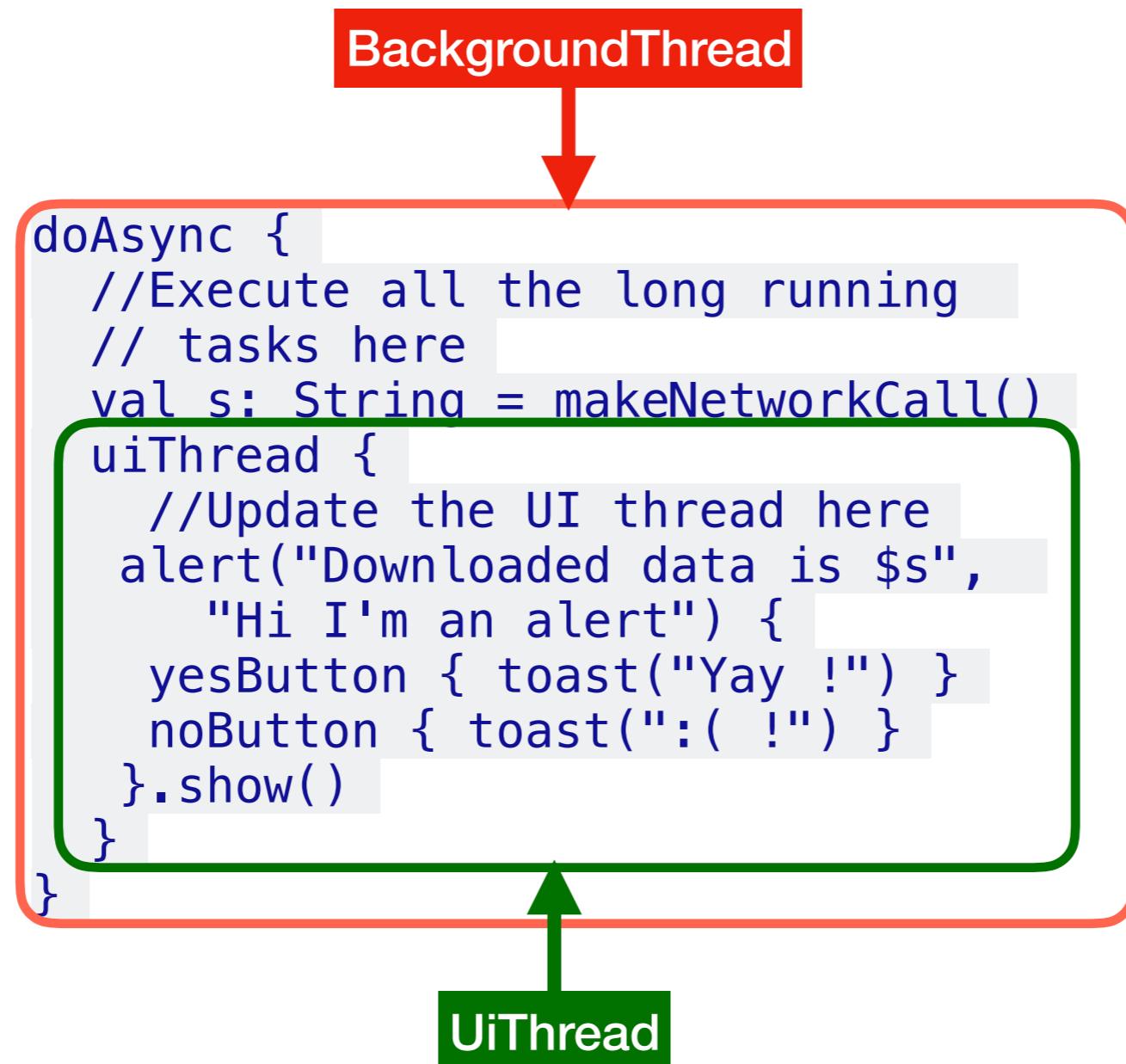
Anko AsyncTask Alternative

```
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()
    }
}
```

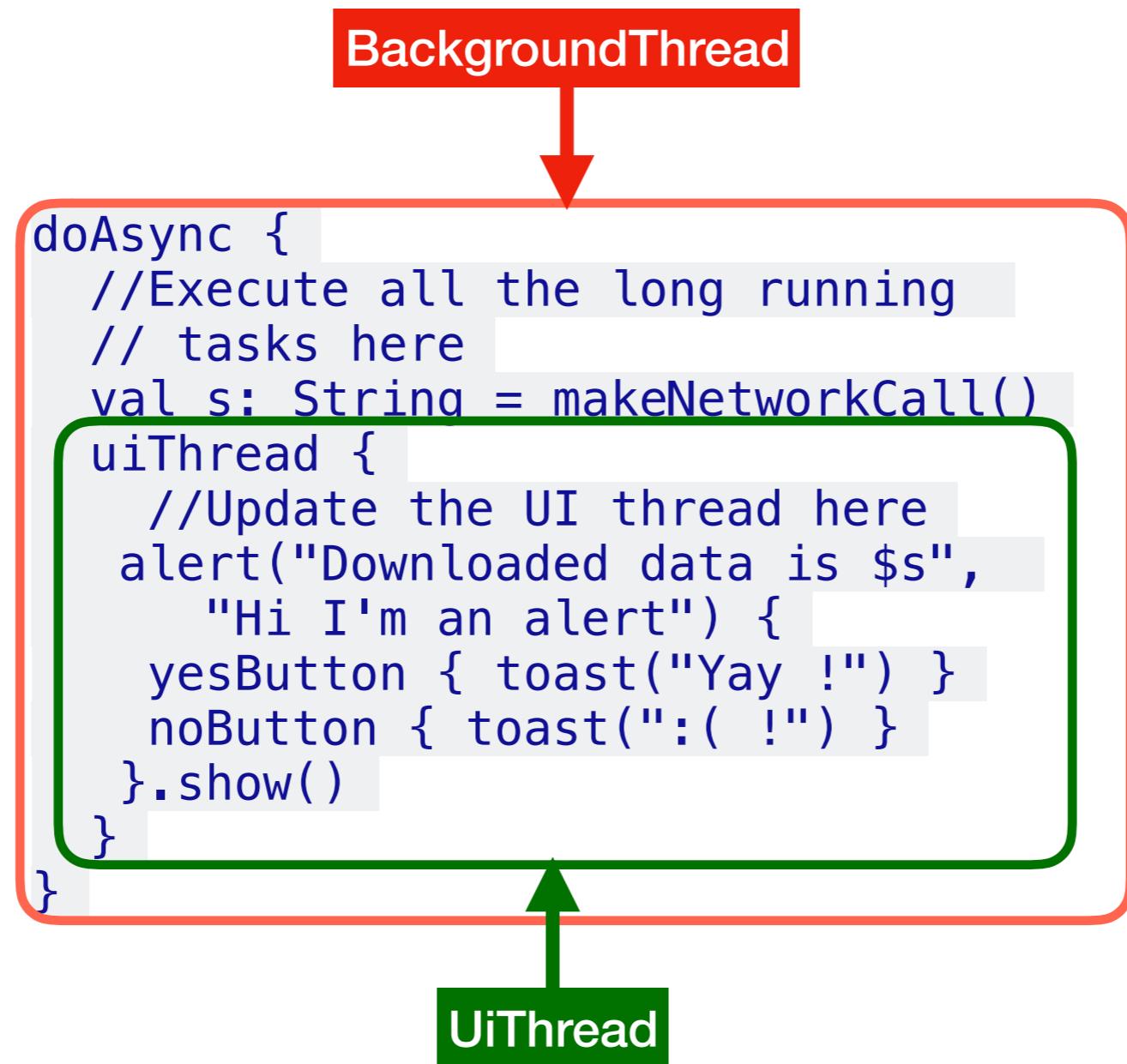
Anko AsyncTask Alternative



Anko AsyncTask Alternative



Anko AsyncTask Alternative



Lecture outcomes

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

