Data-driven layouts

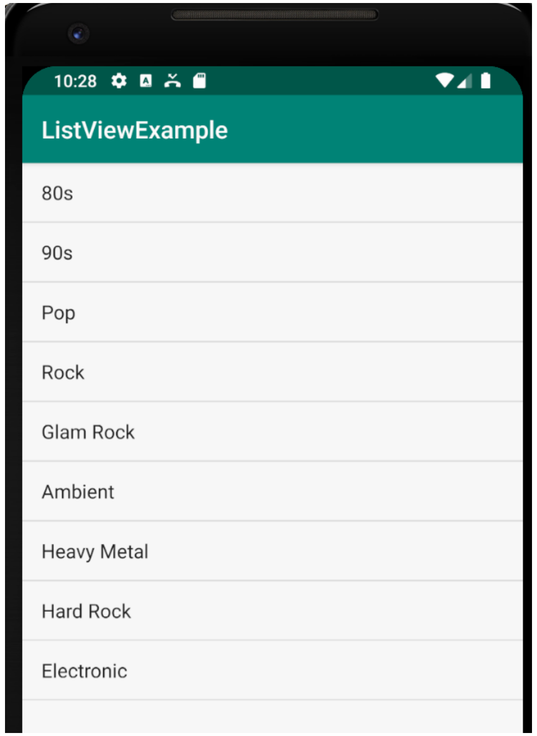
Visualizing data

- *LinearLayout*, *ConstraintLayout*, *RelativeLayout* (and others) are useful for defining the structure of UI
  - The data shown is hard coded

- Other layouts are more useful for showing data to the user
  - Layout dinamically generated
  - *ListView*, *RecyclerView*, *GridView*, *GalleryView*, …

The list is populated from a data source:

80s
90s
Pop
Rock
Glam Rock
Ambient
Heavy Metal
Hard Rock
Electronic



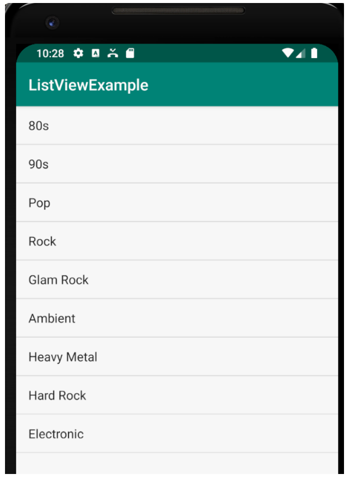
ListViewExample

80s
90s
Pop
Rock
Glam Rock
Ambient
Heavy Metal
Hard Rock
Electronic

Some data-driven layouts

- May want to populate views from a data source (XML file or database)
- Layouts that display repetitive child Views from data source
  - *ListView*
  - *GridView*
  - *ViewPager*
  - …

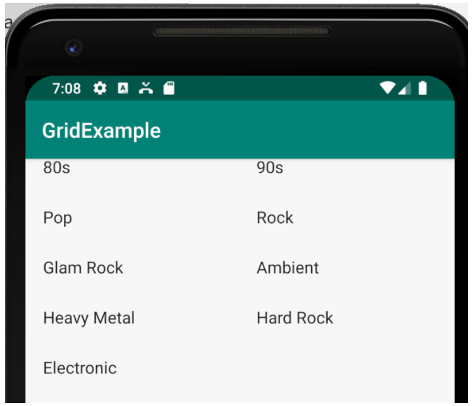
ListView
Rows of entries, pick item, vertical scroll



ListViewExample

80s
90s
Pop
Rock
Glam Rock
Ambient
Heavy Metal
Hard Rock
Electronic

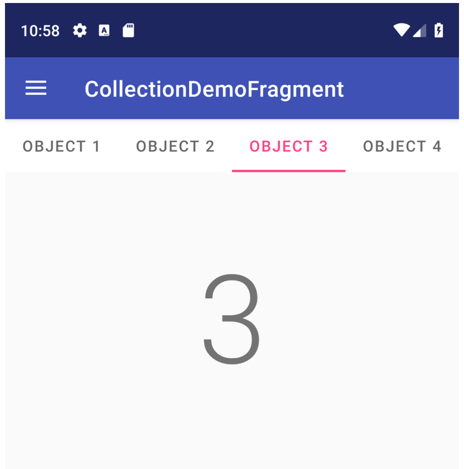
GridView
Items arranged in a number of rows and columns



GridExample

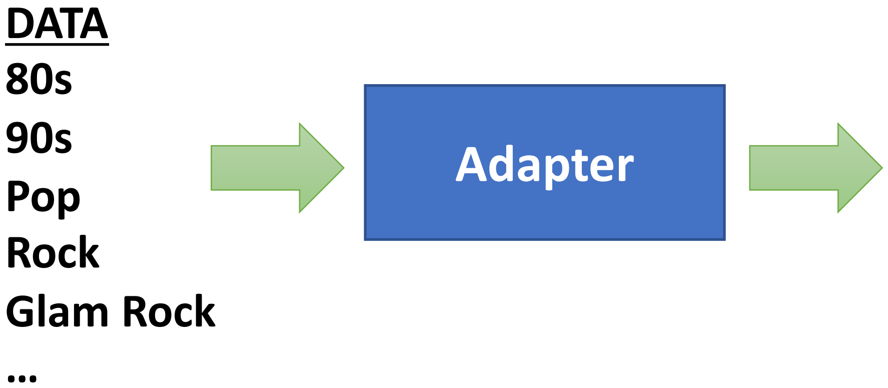
80s	90s
Pop	Rock
Glam Rock	Ambient
Heavy Metal	Hard Rock
Electronic	

ViewPager + TabLayout
Swipe to show new item, can be associated to a TabLayout



Adapter

- **Adapter**: generates widgets from a data source, populates layout
  - E.g. Data is adapted into cells of GridView
- Most common Adapter types:
  - **CursorAdapter**: read from database
  - **ArrayAdapter**: read from resource (e.g. XML file) or Java array



DATA

80s

90s

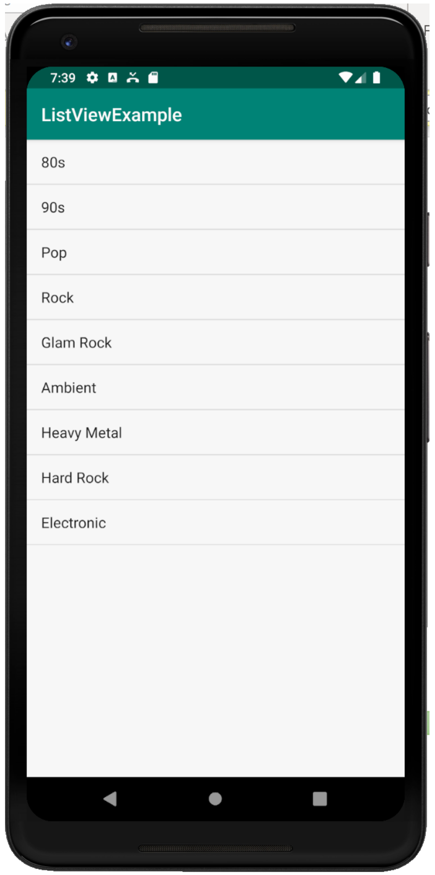
Pop

Rock

Glam Rock

…

Adapter



7:39

ListViewExample

80s

90s

Pop

Rock

Glam Rock

Ambient

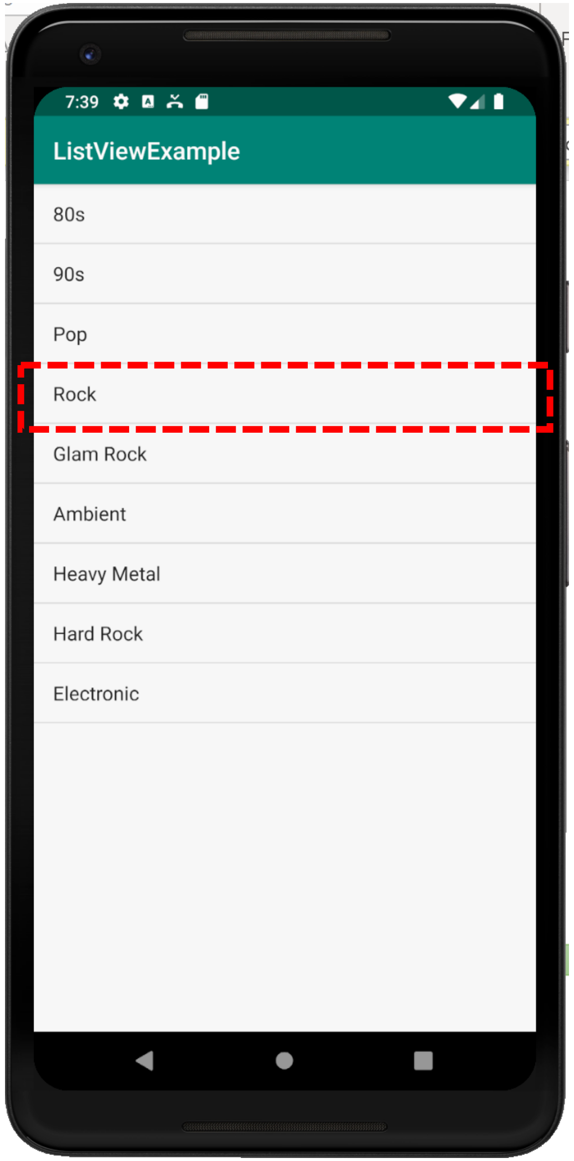
Heavy Metal

Hard Rock

Electronic

Adapter

- When using *Adapter*, a layout (XML format) is defined for each child element (View)
- The adapter
  - Reads in data (list of items)
  - Creates Views (widgets) using layout for each element in data source
  - Fills the containing layout (List, Grid, etc) with the created Views
- Child Views can be as simple as a *TextView* or more complex layouts / controls
  - simple views can be declared in a layout XML file (e.g. *android.R.layout*)



7:39

ListViewExample

80s

90s

Pop

Rock

Glam Rock

Ambient

Heavy Metal

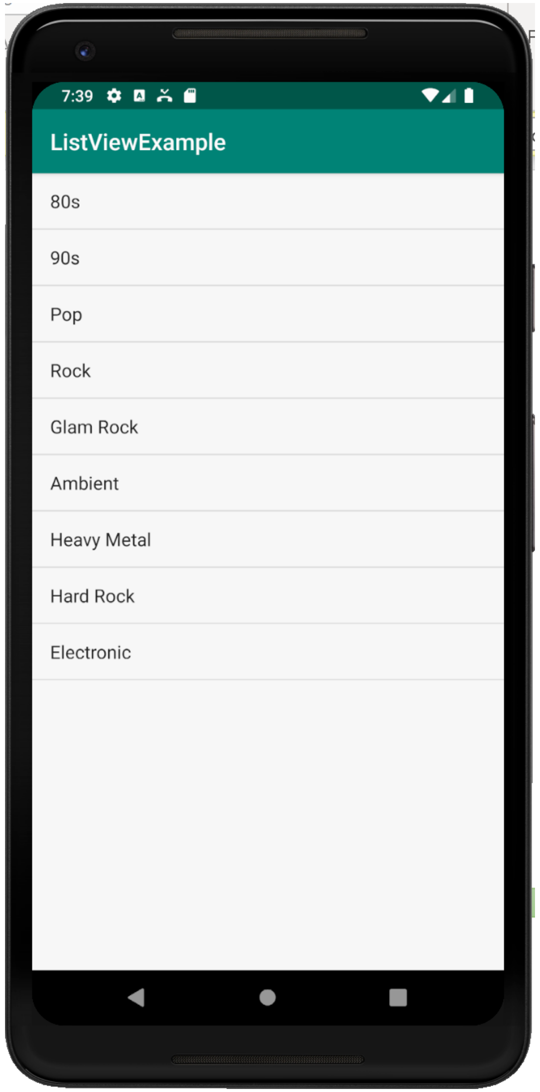
Hard Rock

Electronic

Example: creating ListView with ArrayAdapter

- Create a *ListView* from this array

```
private static final String[] music =
     {"80s", "90s", "Pop","Rock",
      "Glam Rock", "Ambient", "Heavy Metal",
      "Hard Rock", "Electronic"};
```



7:39

ListViewExample

80s

90s

Pop

Rock

Glam Rock

Ambient

Heavy Metal

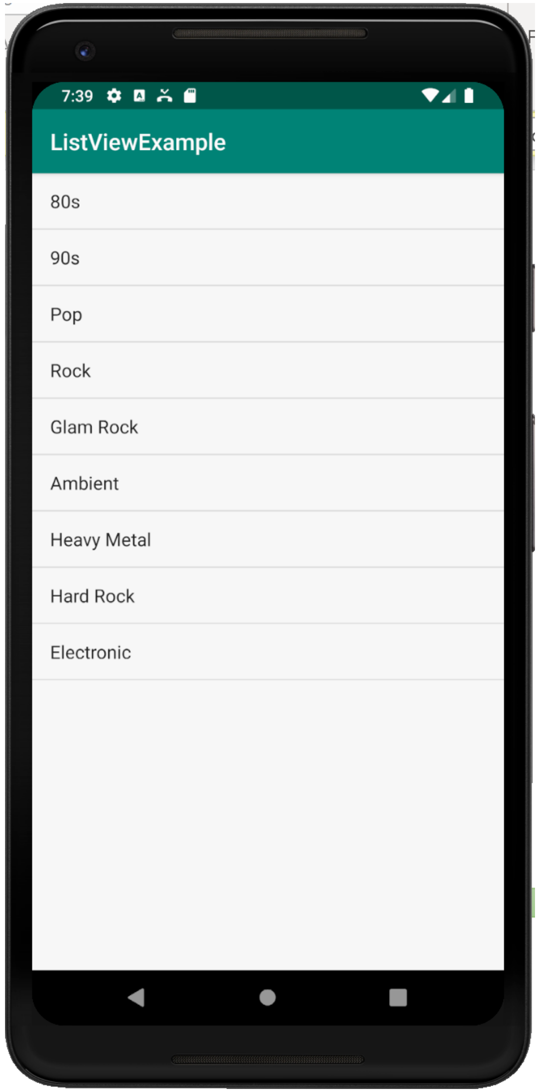
Hard Rock

Electronic

Example: creating ListView with ArrayAdapter

```
<ListView
    android:id="@+id/my_listview"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    . . .
/>
```

In this example the LayoutView fills the screen



ListViewExample

80s

90s

Pop

Rock

Glam Rock

Ambient

Heavy Metal

Hard Rock

Electronic

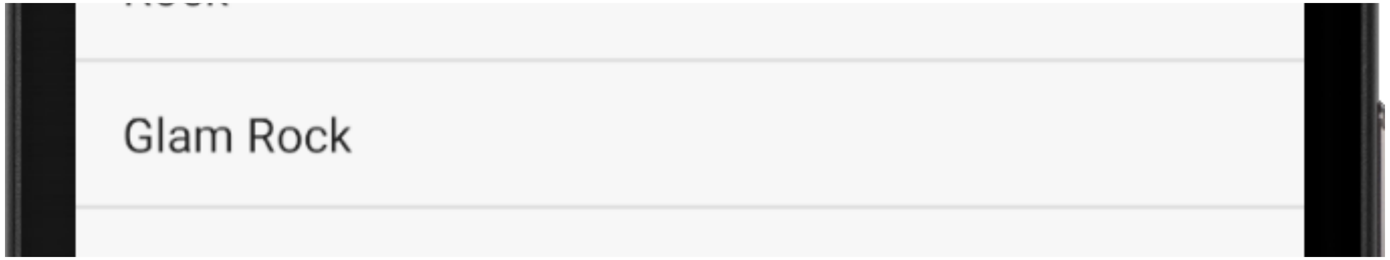
Example: creating ListView with ArrayAdapter

```
public class MainActivity extends AppCompatActivity
                      implements OnItemClickListener {

    private static final String[] music = {"80s", "90s", "Pop",
            "Rock", "Glam Rock", "Ambient",
            "Heavy Metal", "Hard Rock", "Electronic"};
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        ListView lv = (ListView) findViewById(R.id.my_listview);
        lv.setAdapter(new ArrayAdapter<String>(this,
                                               android.R.layout.simple_list_item_1,
                                               music));
        lv.setOnItemClickListener(this);
    }
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        Toast.makeText(this, music[position], Toast.LENGTH_LONG).show();
    }
}
```

Example: creating ListView with ArrayAdapter

- This layout specifies the format of items (font, position of text, etc)

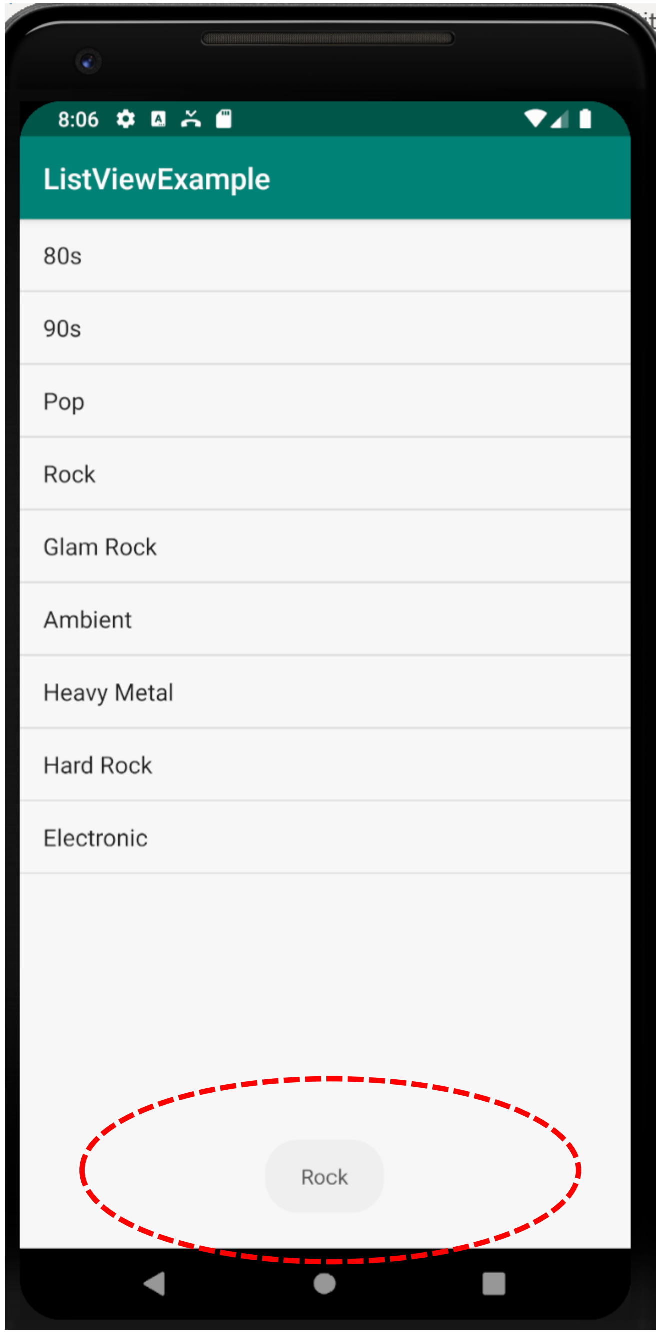
```
android.R.layout.simple_list_item_1
```



Glam Rock

Toast

- A snippet of text that automatically disappears after few seconds
- Used for confirmation (e.g. «message sent», «data saved», …)



ListViewExample

80s

90s

Pop

Rock

Glam Rock

Ambient

Heavy Metal

Hard Rock

Electronic

Rock

References

- CS 528 Mobile and Ubiquitous Computing, WPI
- http://developer.android.com