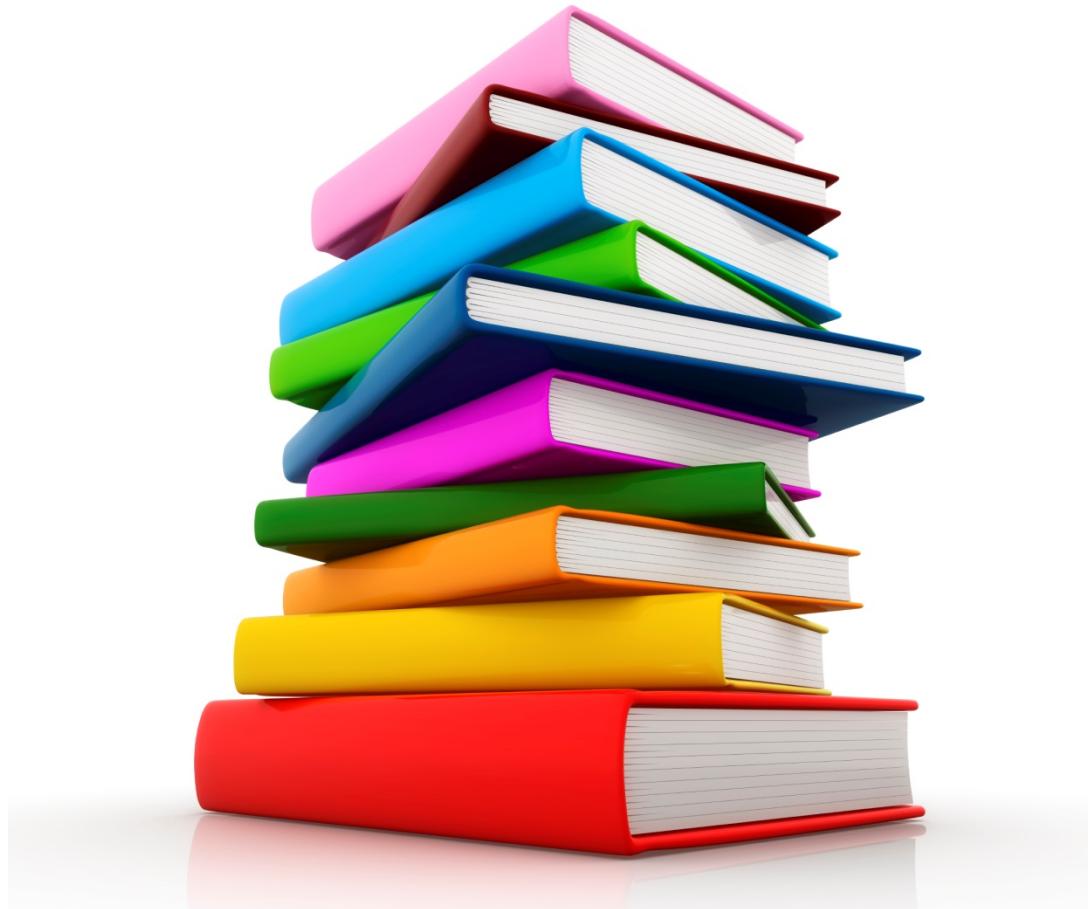


# Resources



# Resources

## Overview

- Split resources from code
  - Layout/strings/images/...
- Resources are all kept in a “res”-folder
- Different resources for different languages or screen sizes
- Default-Resources (Fallback)
  - Used when no special case is defined
  - When not found → App crash (runtime)
- Alternative-Resources
- Used for special case
  - for example landscape is different than portrait

# Resources

## Providing Default resources

- Specific folders in “res” for specific resources
- No other folder-names supported

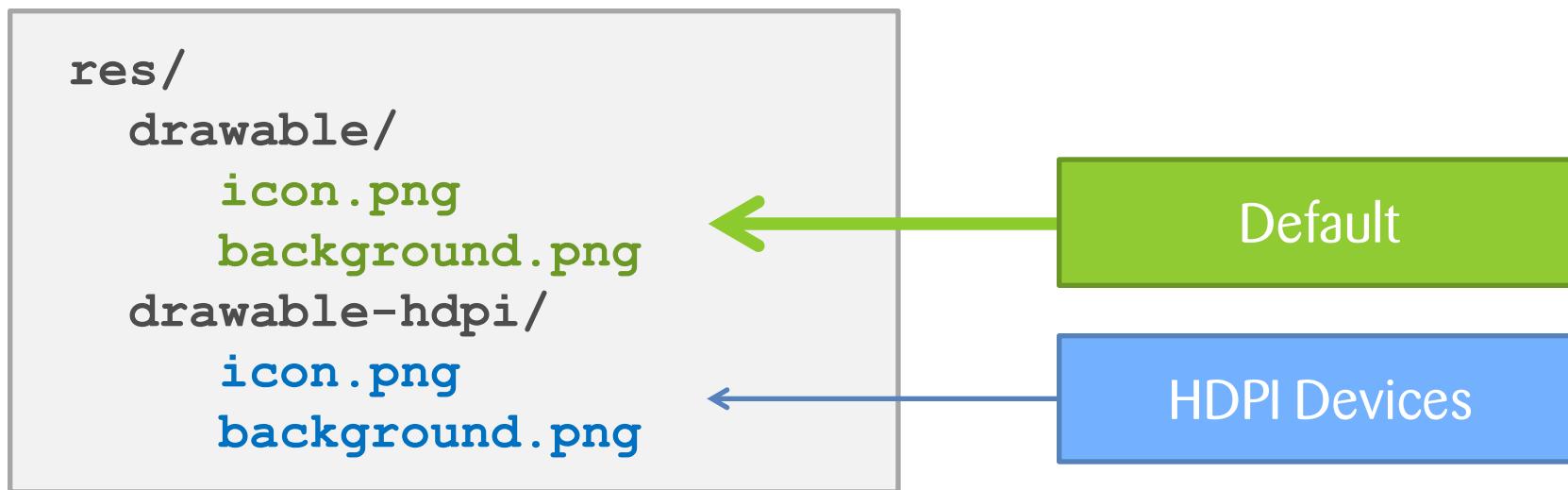
Foldername	Resources
animator	XML files for property animations
anim	XML files for tween animations (View animations)
color	XML files that define state list of colors
drawable	Bitmap-files or XML
layout	XML Layout files
menu	XML files defining Option/Context/Sub-Menu
raw	Other files
values	XML files with simple values like strings, integers
xml	Other XML-files that can be read with <a href="#">Resources.getXML()</a>

# Resources

## Providing Alternative Resources

- Same folder structure as default resources
- Additional qualifiers

```
<resourcesfolder-name>-<qualifier>.
```



# Resources

## Available qualifiers

Qualifier	Example
Language / Region	values-en ; values-de-CH
Screen Size	layout-small ; layout-large
Screen Orientation	layout-land ; layout-port
Screen pixel density (dpi)	layout-mdpi ; layout-hdpi ; drawable-hdpi
Platform version (API Level)	values-v10 ; layout-v8

### Other qualifiers:

Mobile Country Code (MCC); Layout Direction; Smallest Width; Available Width; Available Height; Screen aspect; UI Mode; Night Mode; Touchscreen type; Keyboard availability; Primary text input method; Navigation key availability; Primary non touch navigation method

# Resources

## Providing Alternative Resources

### Qualifier Rules

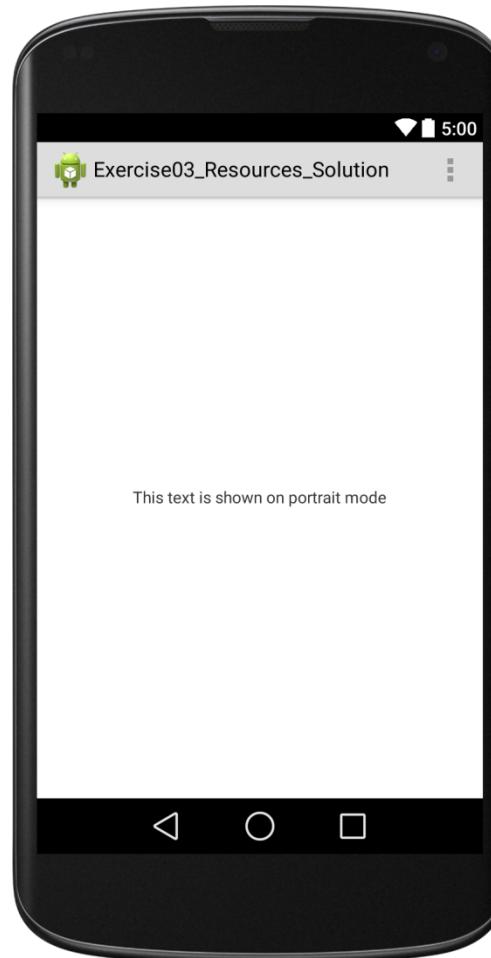
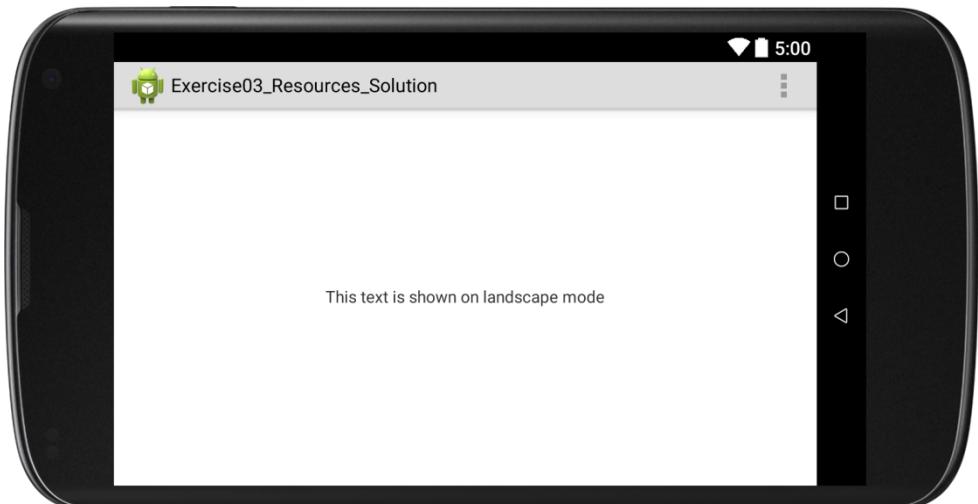
Multiple qualifiers for single resource possible

- For example: layout-de-CH-hdpi
- Order of chained qualifiers is important
  - <http://developer.android.com/guide/topics/resources/providing-resources.html#AlternativeResources>
- Case insensitive
- Multiple qualifier of same type not supported
- For example: “layout-hdpi-mdpi” is not allowed

# Exercise 3

## Exercise03\_Resources

- Show different layout in landscape
  - Use CTRL + F12 to change emulator orientation
  - Show a different text on landscape mode



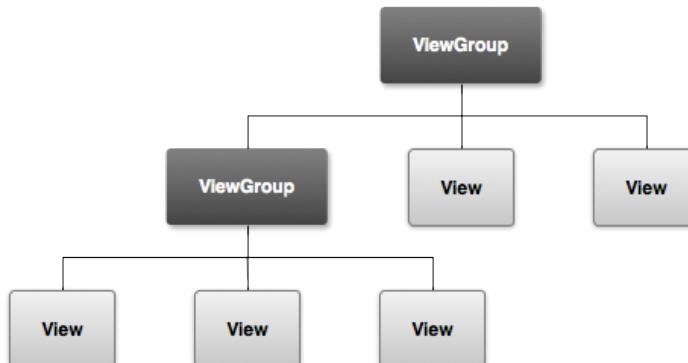
# User Interface



# User interface

## «layouts»

- Layouts are created in XML or in Layoueditor (prefer XML at the beginning to get a better understanding)
  - Placed in `res/layouts`-folder
- Everything is a View
  - `TextView`, `EditText`, `Button`, `Spinner`, etc.
- `ViewGroup` is a container for multiple Views



# User interface

## XML Elements and Attributes for layouts

- Layout\_width / layout\_height mandatory
- Id-attribute to link (find) the view

```
<TextView  
    android:id="@+id/textViewDemo"  
    android:textColor= "#F00"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Activity Lifecycle demo" />
```

# User interface

## ID-Attribute

Create a new Id with “@+id/NameOfTheId”

```
<TextView  
    android:id="@+id/textViewDemo"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"/>
```

Use an existing Id with “@id/NameOfTheId”

```
<TextView  
    android:layout_above="@id/theOtherTextView"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"/>
```

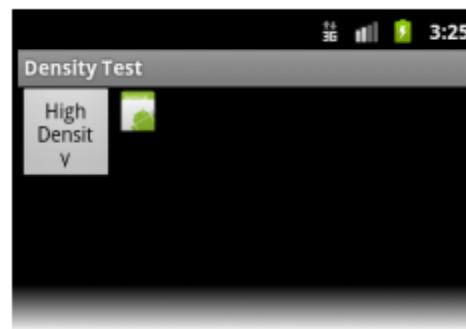
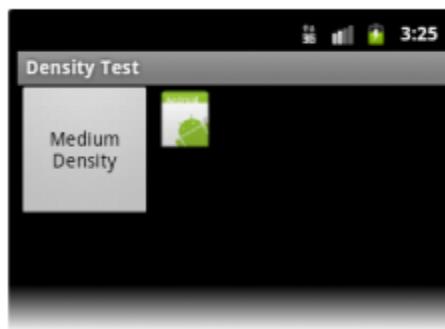
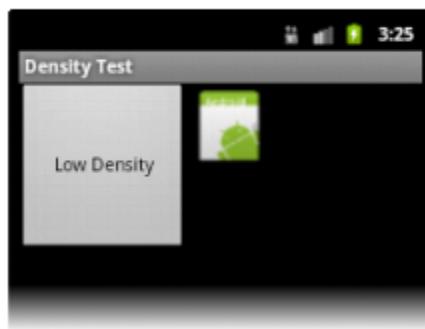
# Layout Parameter

- Define how views are measured
- Depend on the ViewGroup the view does belong to
- layout\_width / layout\_height are mandatory to every view
- Multiple measurement units available

# Layout Parameter

## Measurement units

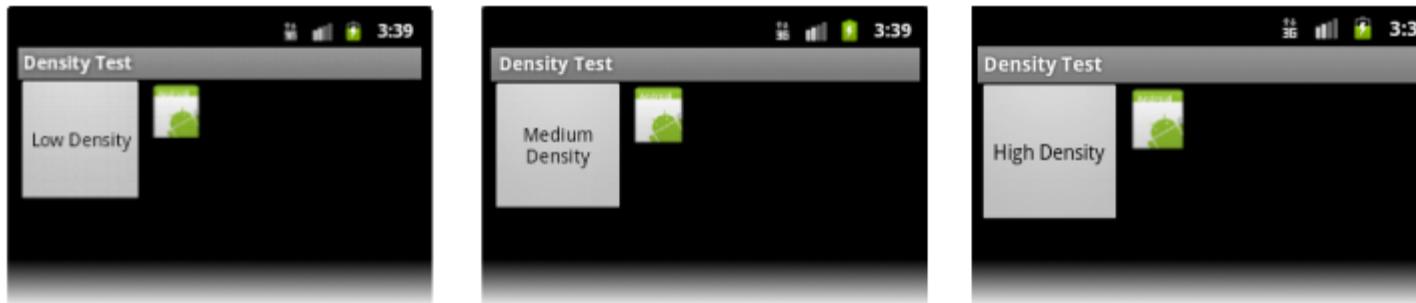
Shortcut	Name	Description
px	Pixels	Real pixel on screen
in	Inches	Size in inches
mm	Millimeters	Size in millimeters
pt	Points	1/72 in



# Layout Parameter

## Measurement units

Shortcut	Name	Description
dp/dip	Density – independent pixel	Abstract unit based on the current screen density
sp	Scale – independent pixel	Same as dp but scaling with font preference



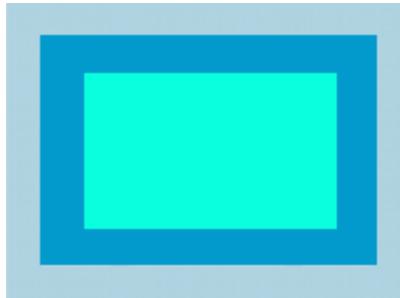
# View Groups



Bild Quelle: <http://www.kunocreative.com/blog/bid/86368/Top-Tips-and-Tools-to-Measure-Social-Media-ROI>

# View Groups

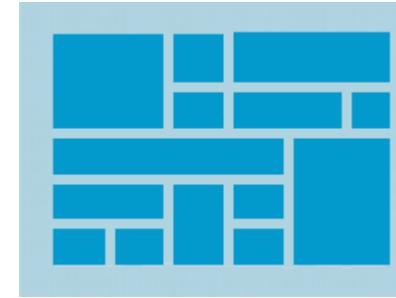
**FrameLayout**



**RelativeLayout**



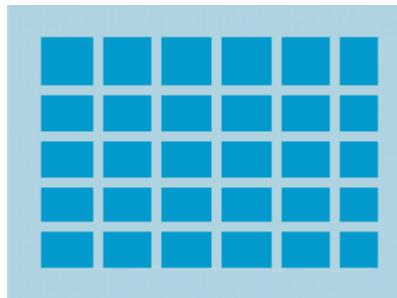
**GridLayout**



**LinearLayout**



**TableLayout**



# View Groups

## Linear Layout

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_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>
```

Orientation

# View Groups

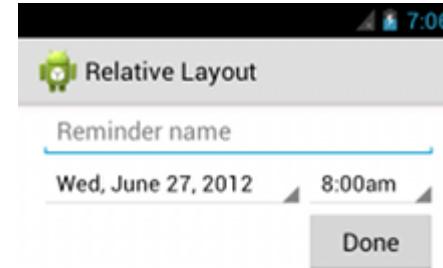
## Relative Layout

- Align views relative to each other
  - android:layout\_toLeftOf="@+id/otherView"
  - android:layout\_below="@+id/otherView"
- Align views relative to parent
  - android:layout\_alignParentLeft="true"
  - android:layout\_alignParentBottom="true"
- Alignment parameter to override width / height

# View Groups

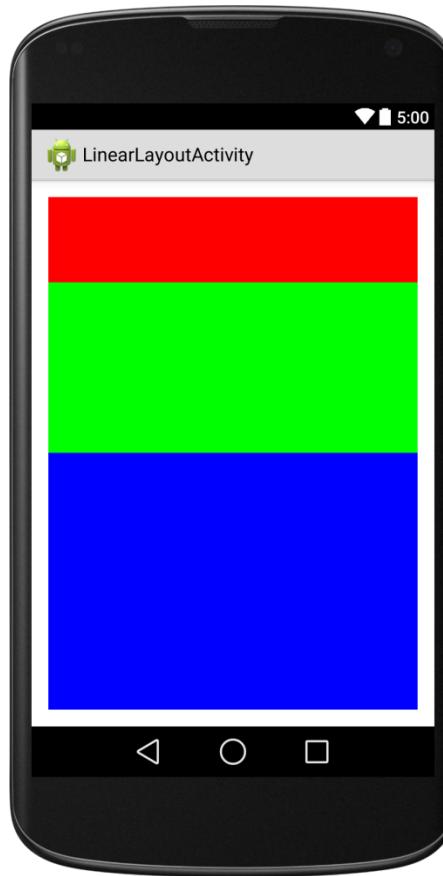
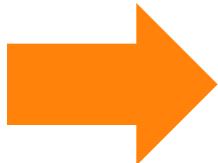
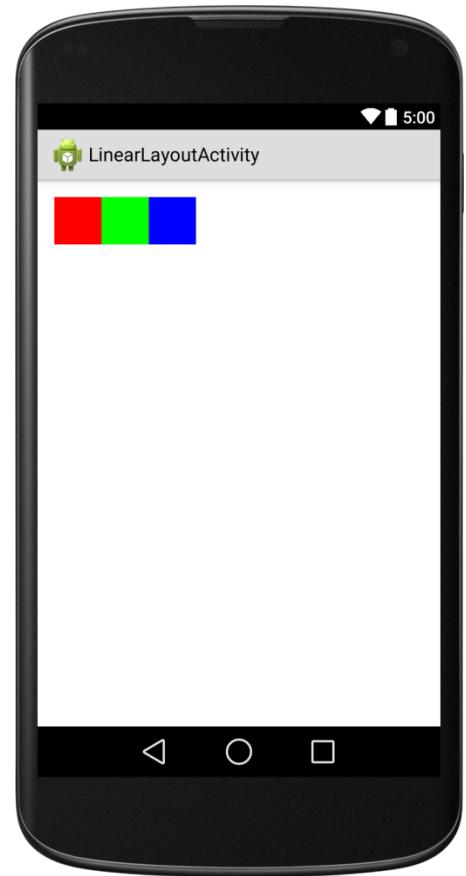
## Relative Layout

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="16dp"
    android:paddingRight="16dp" >
    <EditText
        android:id="@+id/name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/reminder" />
    <Spinner
        android:id="@+id/dates"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_below="@+id/name"
        android:layout_alignParentLeft="true"
        android:layout_toLeftOf="@+id/times" />
    <Spinner
        android:id="@+id/times"
        android:layout_width="96dp"
        android:layout_height="wrap_content"
        android:layout_below="@+id/name"
        android:layout_alignParentRight="true" />
    <Button
        android:layout_width="96dp"
        android:layout_height="wrap_content"
        android:layout_below="@+id/times"
        android:layout_alignParentRight="true"
        android:text="@string/done" />
</RelativeLayout>
```



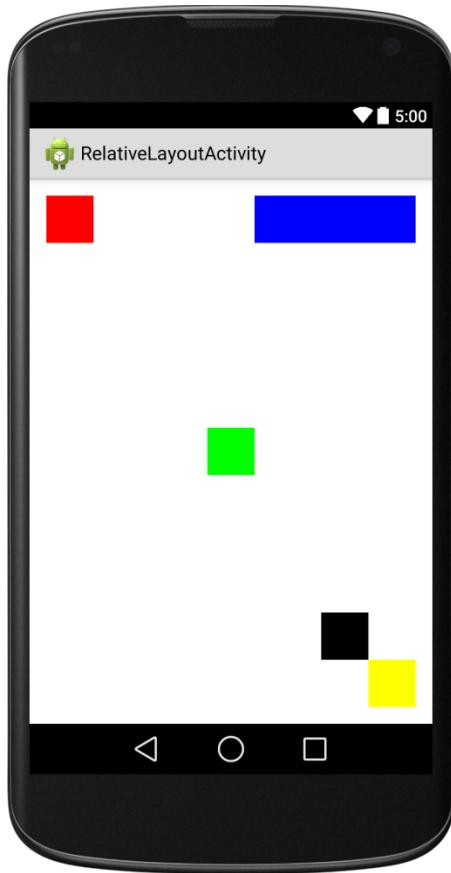
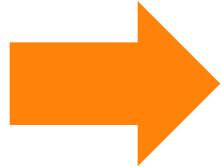
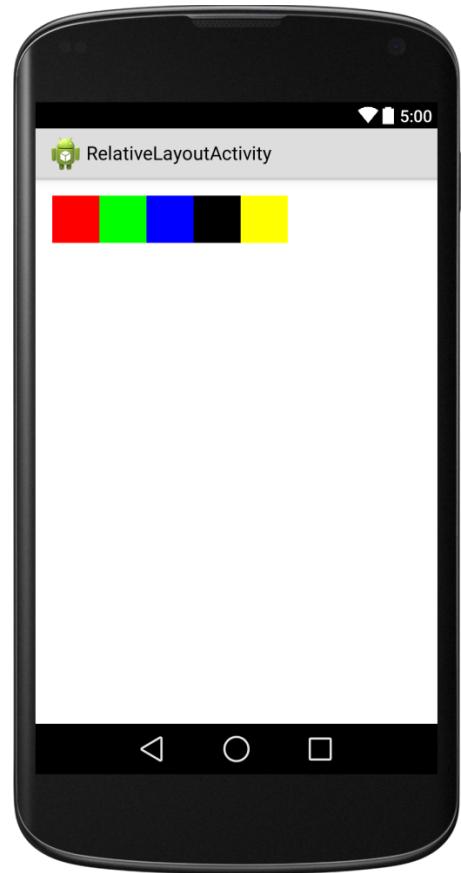
# Exercise 4

Exercise04\_Loops → LinearLayout



# Exercise 4

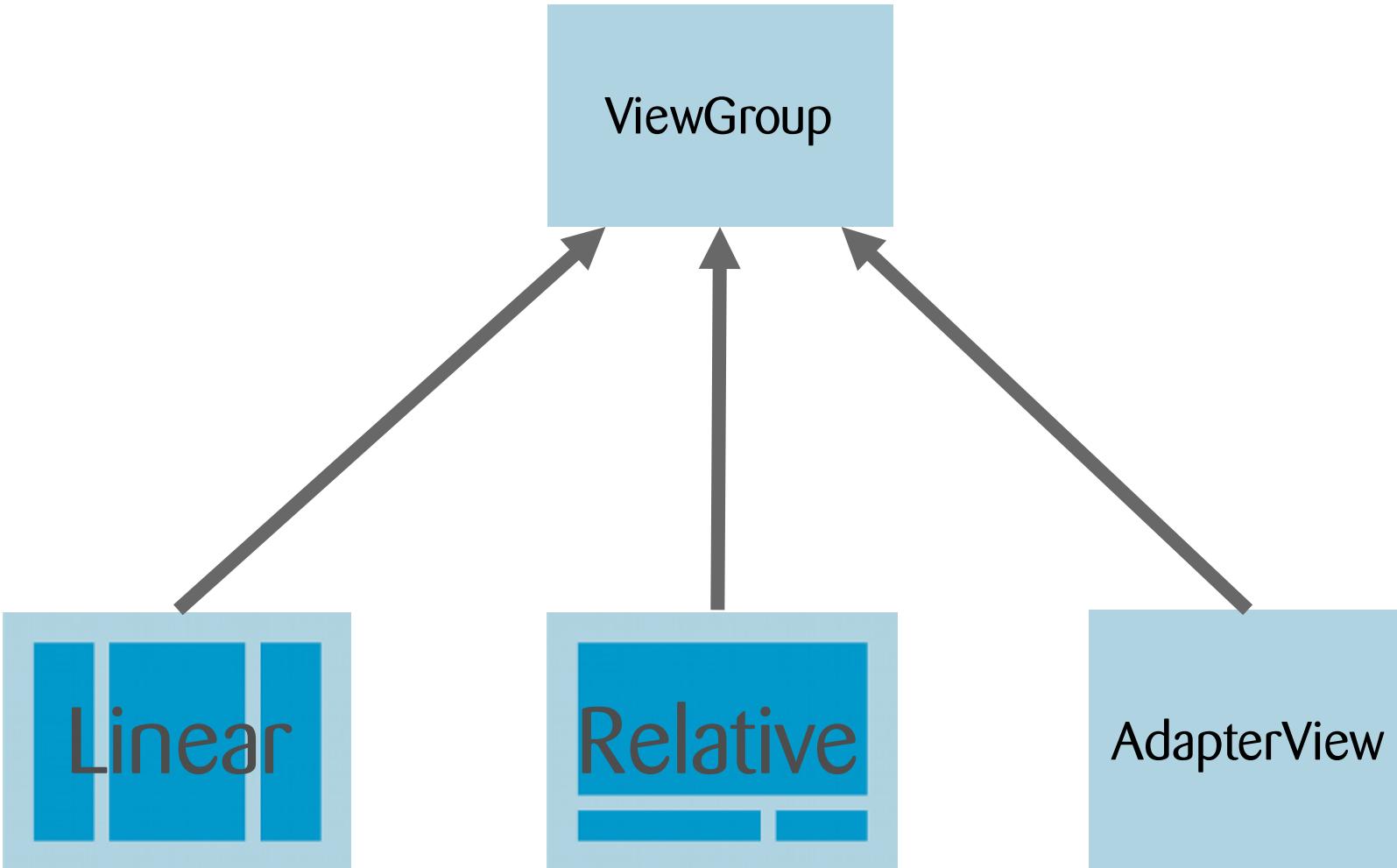
Exercise04\_Loops → Relative Layout



# AdapterView



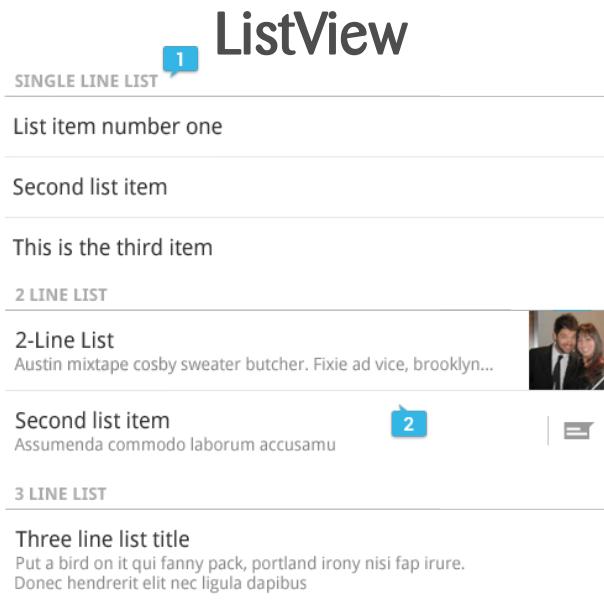
# Adapter View



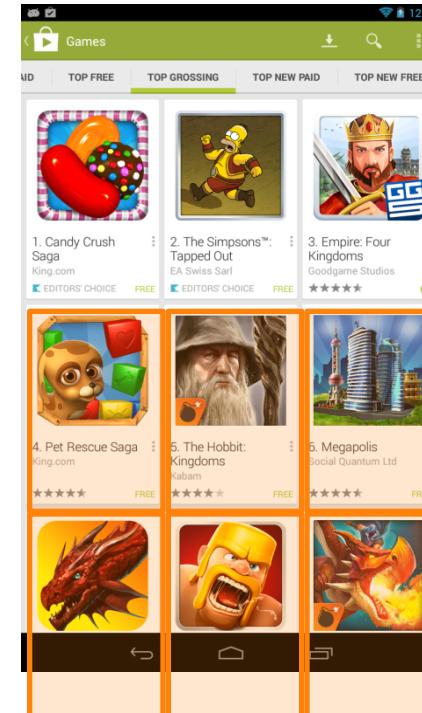
# AdapterView

## Types

- AdapterView



## GridView

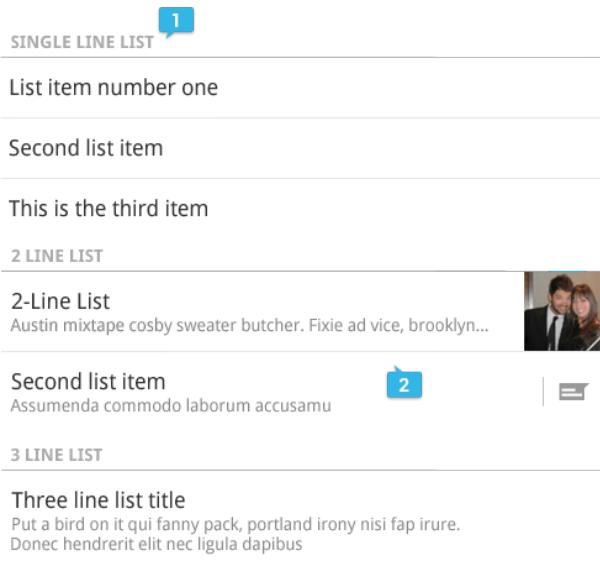


# AdapterView

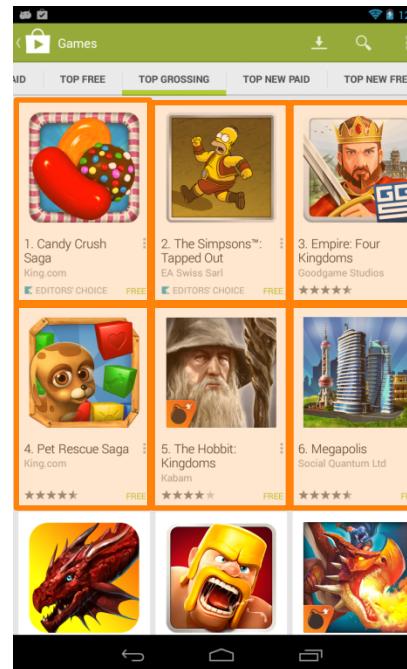
## Types

- Display data in a List/Grid
- Dynamic views

### ListView

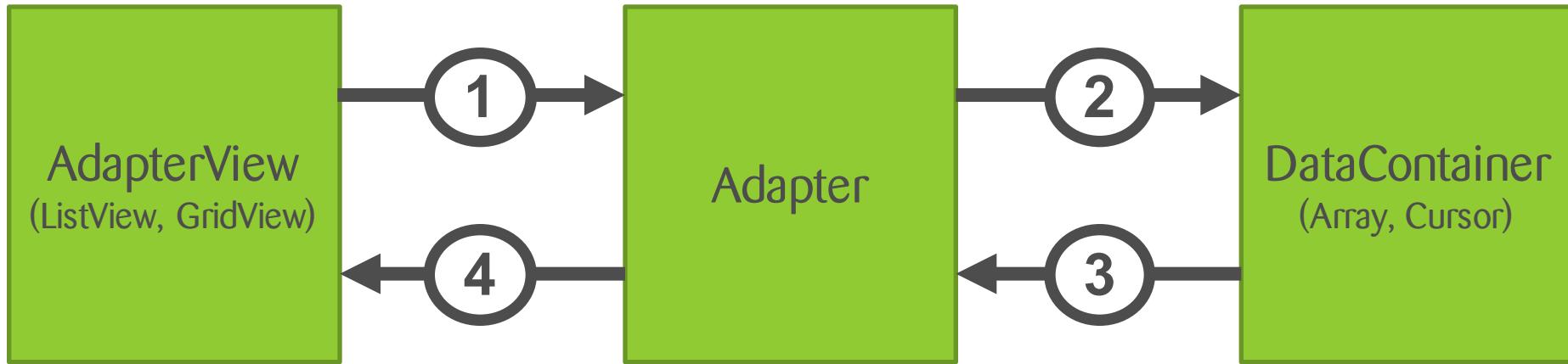


### GridView



# Adapter View

Connect data to AdapterView



1. AdapterView requests Views to be shown
2. Adapter requests data to be shown
3. DataContainer delivers data entries
4. Adapter creates views for every data entry

# Adapter View

## Connect data to AdapterView

- Implement interface Adapter
  - ArrayAdapter, BaseAdapter, SimpleAdapter, SpinnerAdapter, ...
- Methods to implement
  - getCount()
  - getItem(int position)
  - getItemId(int position)
  - getView(int position, View convertView, ViewGroup parent)
  - getViewType(int position)
  - getViewTypeCount()

# Adapter View

## Signature of getView(...)

```
public abstract View getView(int position, View convertView, ViewGroup parent)
```

position: Position of the item within the adapters data-set

convertView: Cached view to be reused

parent: Reference to the parent (AdapterView)

- Create view when “convertView” is null
- Set data to the given view
- When multiple types of views
- → Make sure getViewType() / getViewTypeCount() is implemented

# Adapter View

Create Views in `getView(...)`

How to create Views?

- Use a `LayoutInflater` (System-Service)
- Use `inflate(int layout, ViewGroup root, bool attachToRoot)`

```
String name = Context.LAYOUT_INFLATER_SERVICE;
LayoutInflater infl = (LayoutInflater) context.getSystemService(name);
View view = infl.inflate(R.layout.spaced_list_item, parent, false);
```

# Exercise: Exercise05\_AdapterView

## Aufgaben:

1. Sämtliche Klassen analysieren und Zusammenhang zwischen MainActivity (main\_activity.xml), PersonAdapter und der ListView verstehen
2. Bug beheben, dass der Name und die Adresse nicht angezeigt werden

## Verwendete API Klassen:

- Activity
  - Wird benötigt um einen einzelnen Screen anzuzeigen
- ListView
  - Eine AdapterView welche in main\_activitx.xml definiert wurde und sämtliche Personen aus “PersonRepository” anzeigen soll
- BaseAdapter
  - Basis implementation des Adapter-Interface
- LayoutInflater
  - Wird benötigt um in der getView()-Methode des Adapters die benötigten Views zu erstellen