

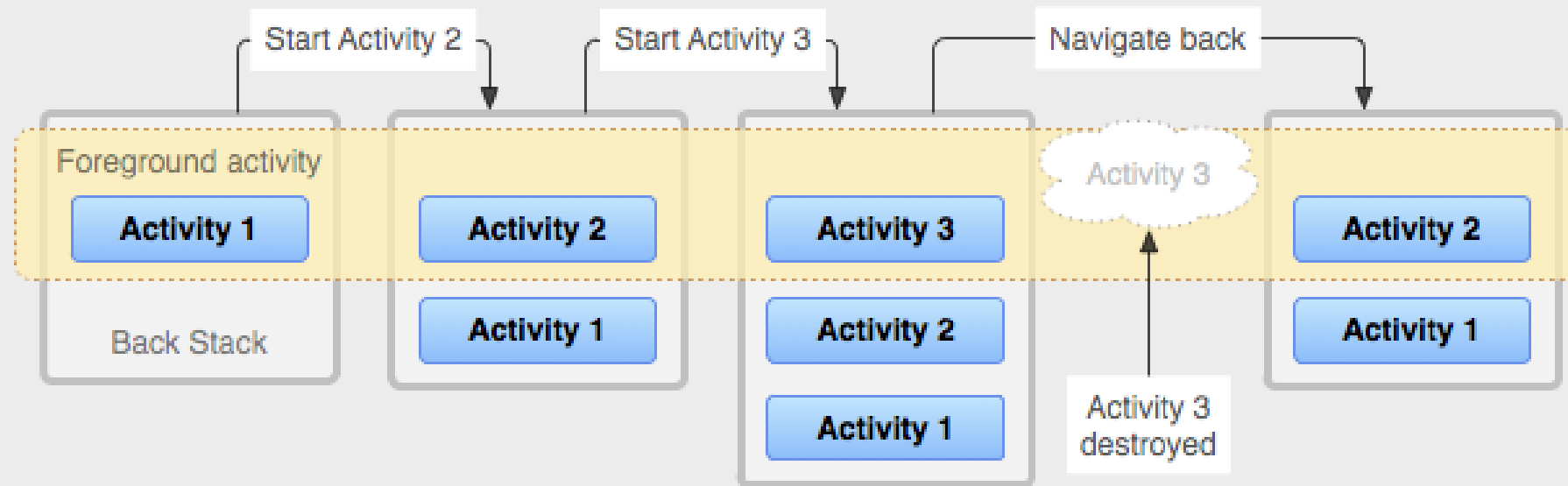
## Task and Back Stack

### **Task**

A task is a collection of activities that users interact with when performing a certain job.

### **Back Stack**

The activities are arranged in a stack (the "back stack"), in the order in which each activity is opened.



## Resources

- Split resources from code
  - Layout/strings/images/...
- Resources are all kept in a “res”-folder
- Different resources for different languages or screen sizes
- Default/Fallback-Resources
  - When not found → App crash (runtime)

### **drawable**

Everything that has something to do with graphics (images, xml)

### **layout**

Layout files for your activities, list items, ...

### **menu**

Menu definitions

### **values**

Simple values like strings for translations

### **mipmap**

Launcher Icon

## Resources – Providing alternative Resources

Same folder structure as default resources

Additional qualifiers

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

Same filename as default resource

```
res/  
  drawable/  
    icon.png  
    background.png  
  drawable-hdpi/  
    icon.png  
    background.png
```

Default

HDPI Devices



### Qualifier Examples

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

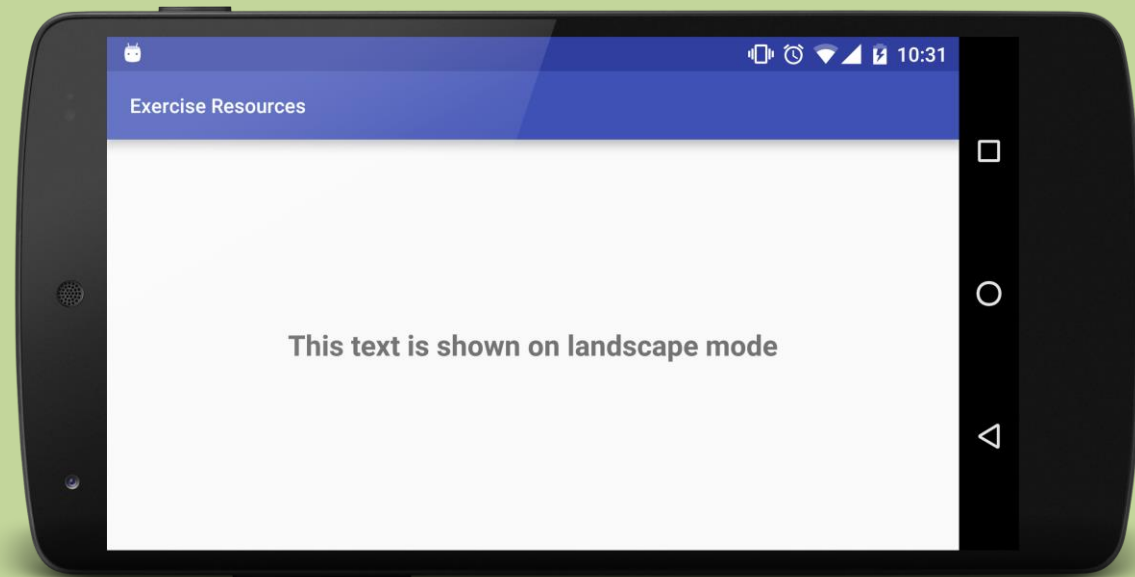
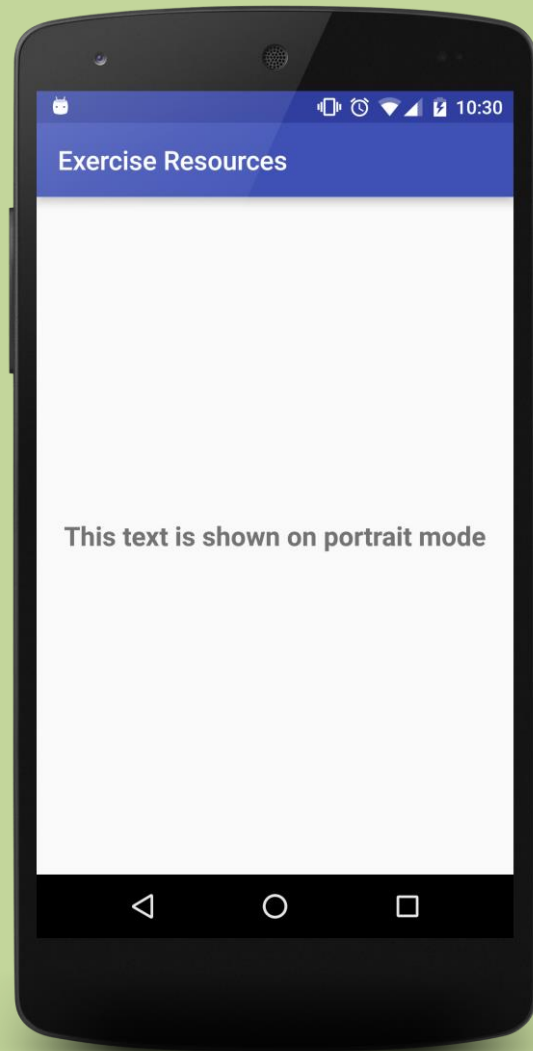
### Other qualifiers

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



### 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



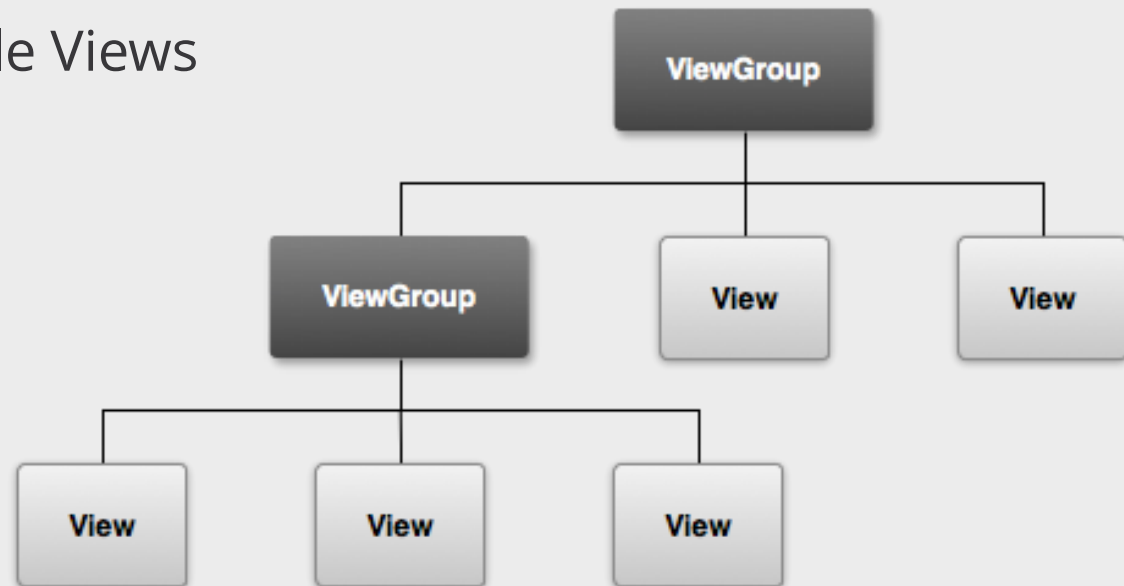
**Aufgaben:**

1. Verstehe wie Android die korrekte Resource lädt
2. Verstehe warum und wann du Ressourcen einsetzen musst
3. Stelle sicher dass die Demo-App in der Landscape-Ansicht einen anderen Text darstellt als im Portrait-Mode

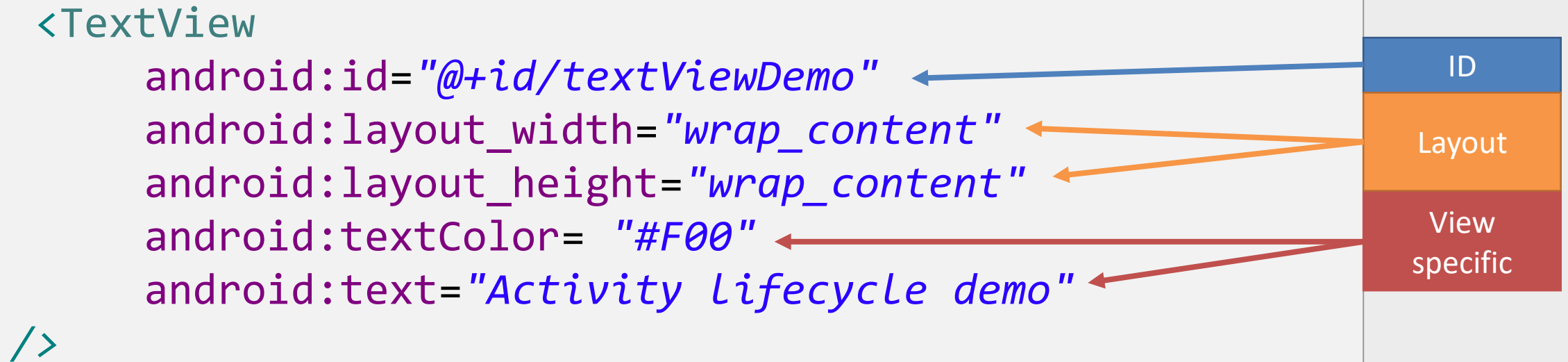
**Projekt: Exercise\_Resources**

## User Interface

- Layouts are created in XML
  - Placed in res/layouts-folder
- Everything is a View
  - TextView, EditText, Button, Spinner, etc.
- ViewGroup is a container for multiple Views



## View Attributes



## 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"/>
```

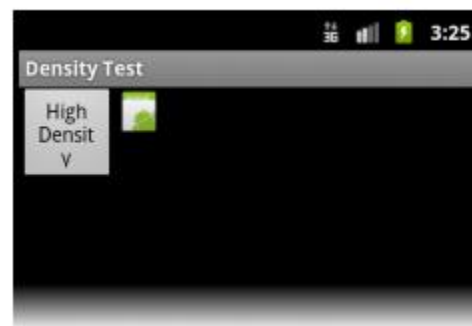
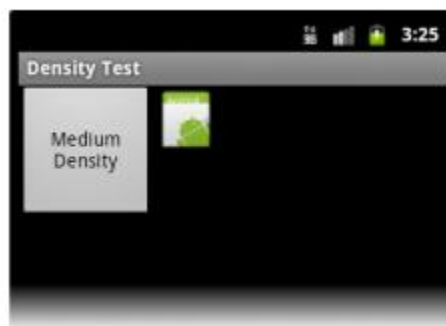
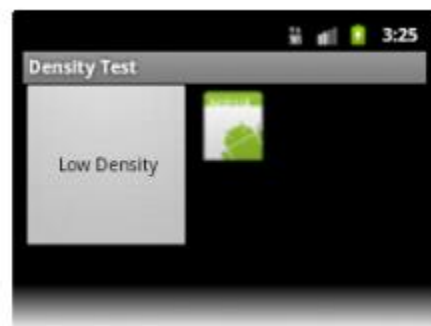
## Layout Attributes



- Define how views are measured
- `layout_width` / `layout_height` are mandatory to every view
- Depend on the ViewGroup the view does belong to
- Multiple measurement units available

## Layout Attributes – 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 Attributes – 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



### **match\_parent (früher fill\_parent)**

Die View nimmt soviel Platz ein, wie das Parent-Element zur Verfügung stellt.

### **wrap\_content**

Die View nimmt genau so viel Platz ein, damit der Inhalt dargestellt werden kann.

# Layout Attributes

## View Groups

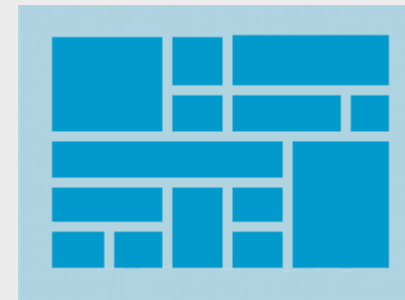
FrameLayout



RelativeLayout



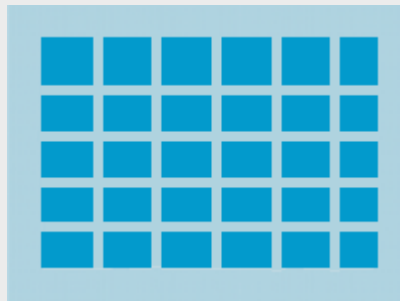
GridLayout



LinearLayout



TableLayout



...

## View Groups – 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="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>
```



Orientation (vertical/horizontal)



```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/demoTextView"
        android:text="@string/hello_world"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

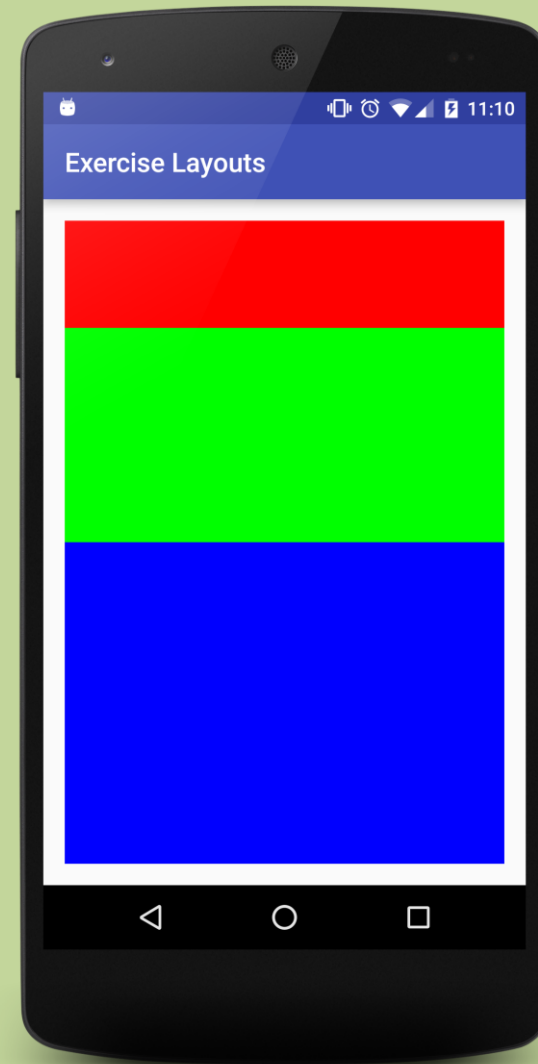
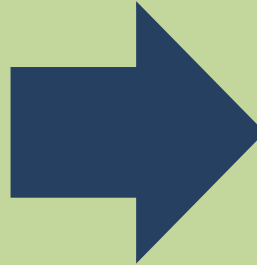
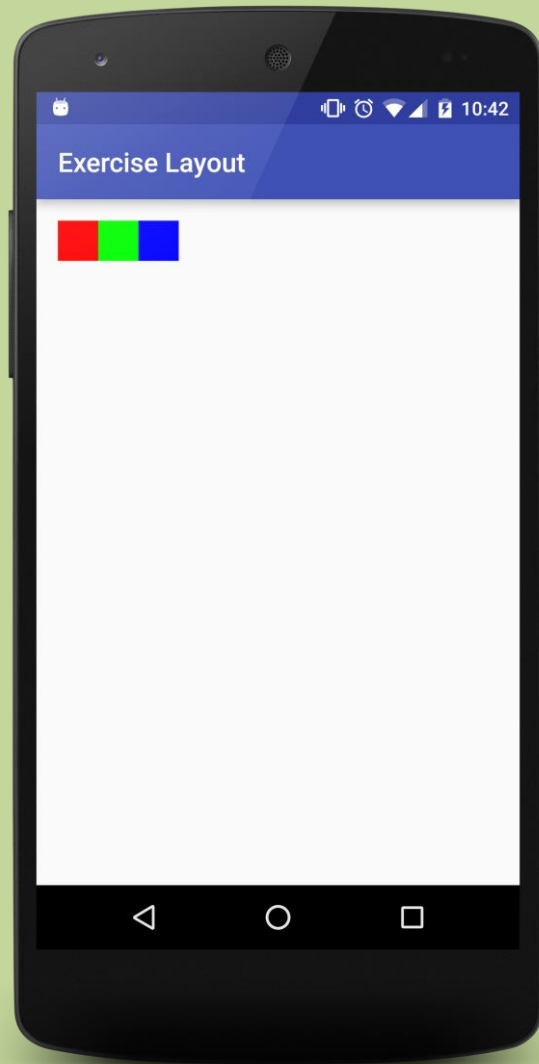
    <TextView
        android:id="@+id/secondDemoTextView"
        android:layout_toRightOf="@id/demoTextView"
        android:layout_alignParentBottom="true"
        android:text="@string/hello_world"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

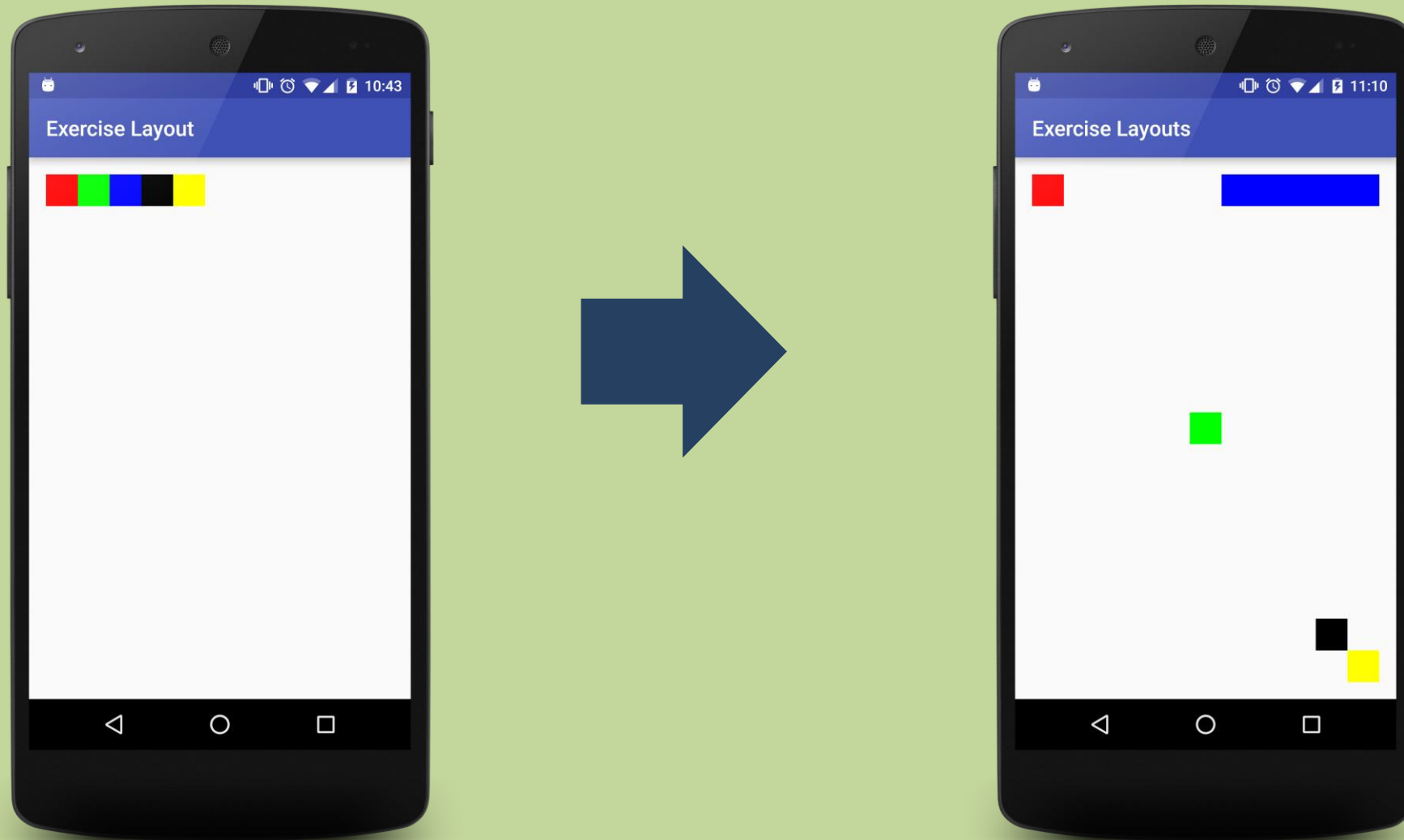
</RelativeLayout>
```

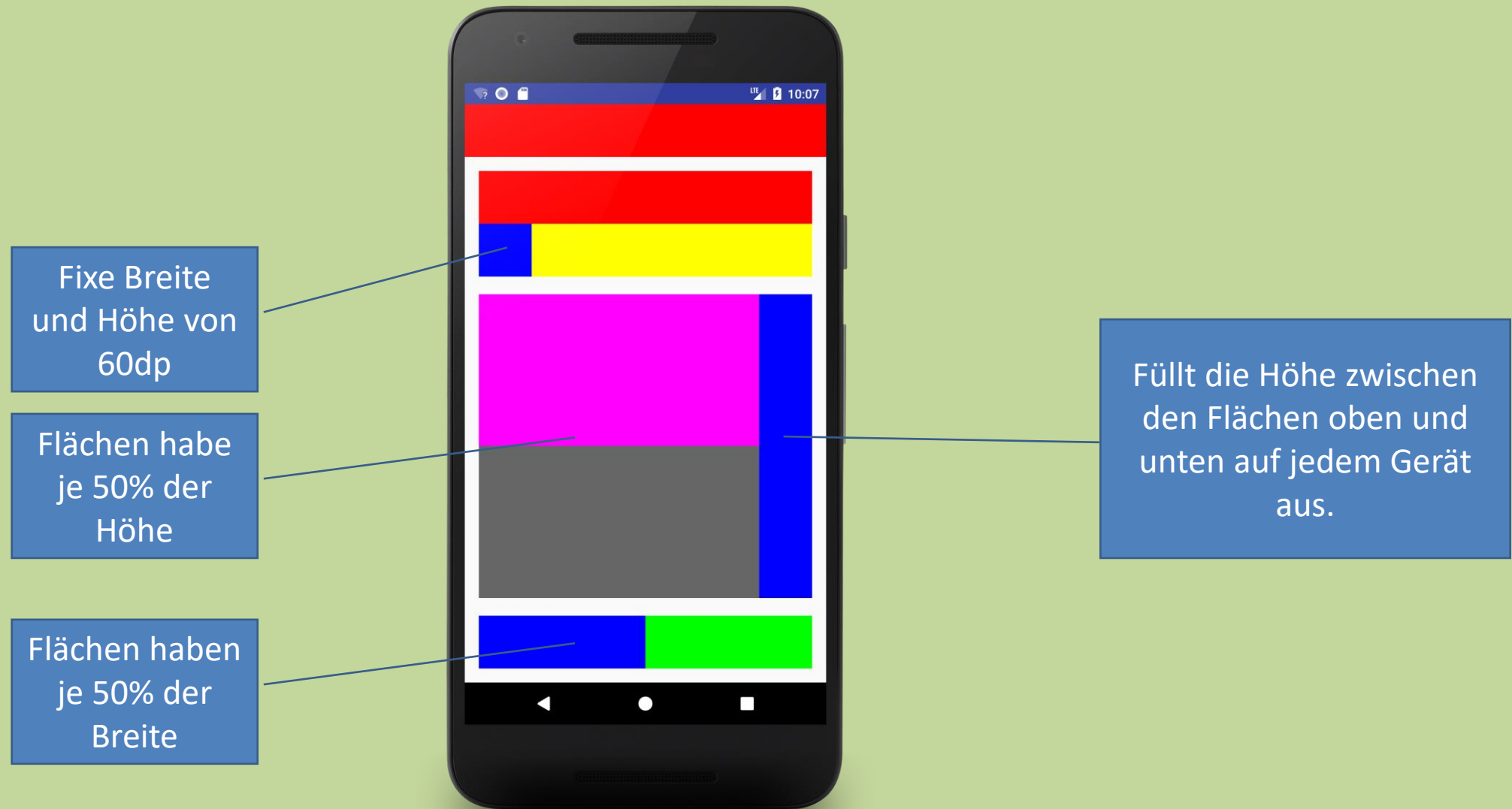


Align right of “demoTextView”

Align to bottom of parent  
(RelativeLayout itself)







## **Tipps LinearLayout:**

- Layout orientation attribute
- Use android:layout\_weight

## **Tipps RelativeLayout:**

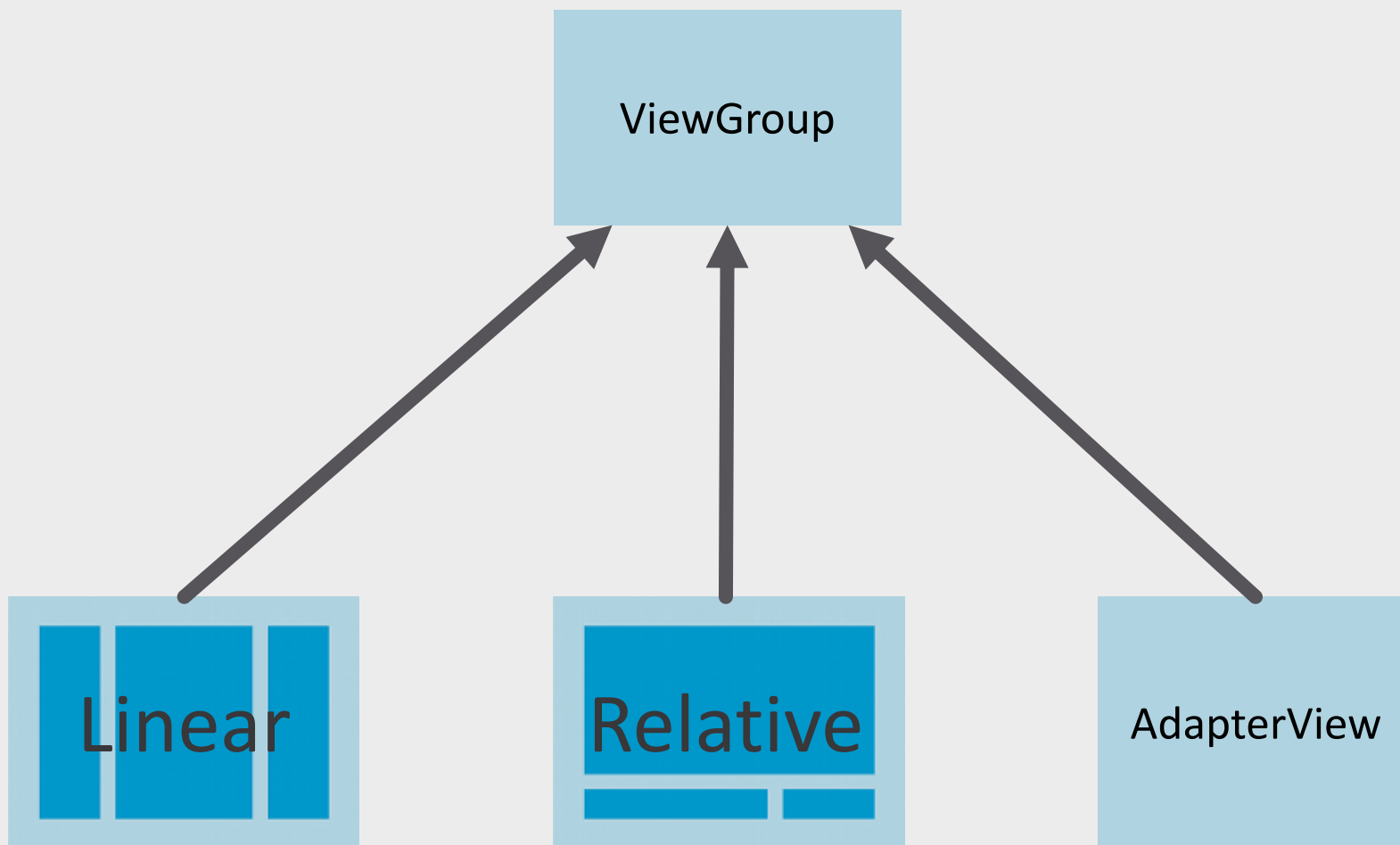
- Use android:layout\_XXX-Attributes
- Reference other views by id “@id/otherViewName”
- Use example in UIElements-project as reference
- Use intellisense to find all possible attributes
- Maybe you have to reorder the elements

## **Tipps App Layout**

- Kombinieren sie das LinearLayout und das RelativeLayout wo nötig.

## **Projekt: Exercise\_Layouts**

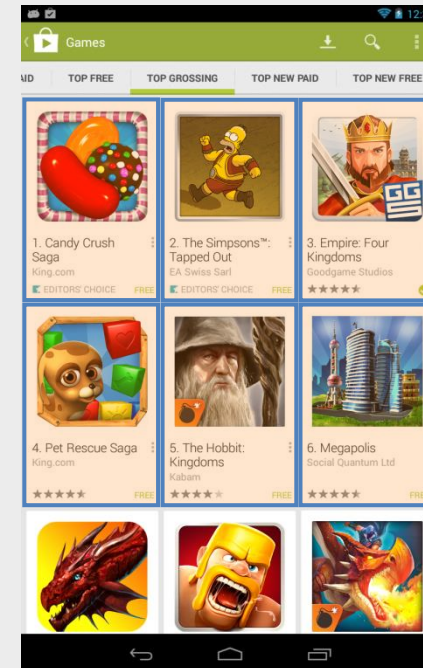
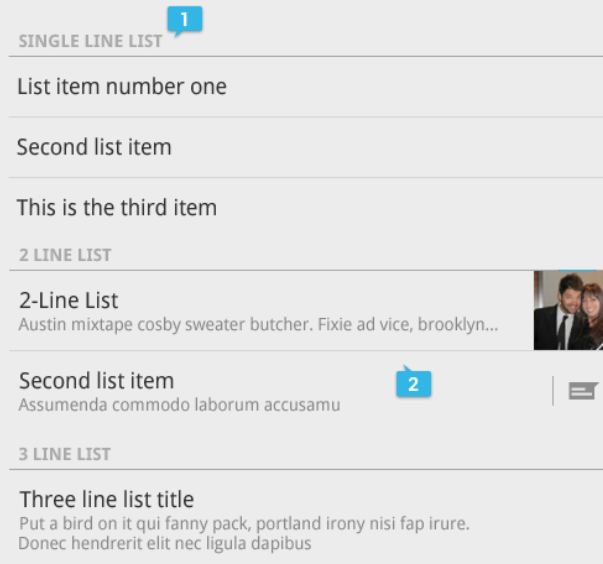
## Adapter View



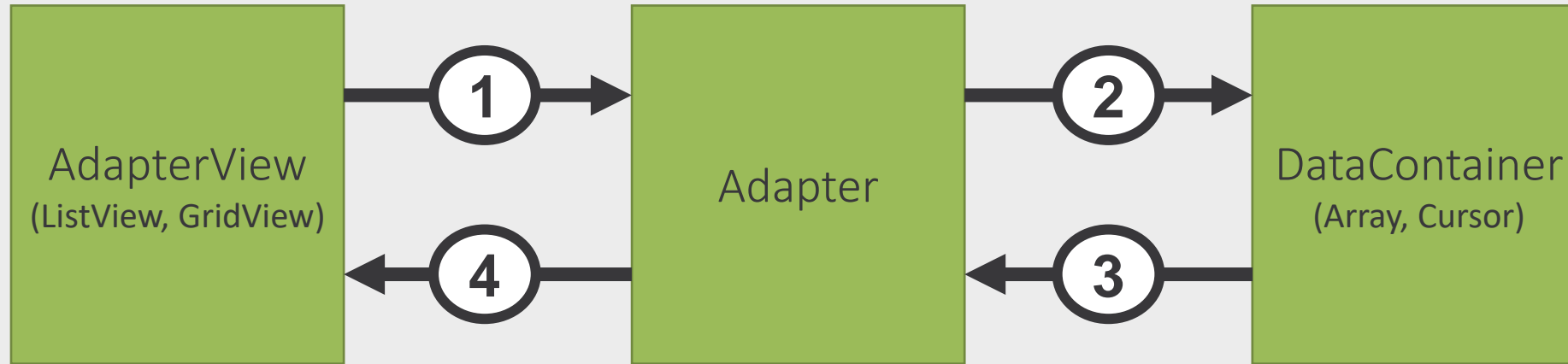
# Adapter View

# Paixon

- Display data in a List/Grid
- Dynamic views







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

## Implement interface Adapter

- ArrayAdapter, BaseAdapter, SimpleAdapter, SpinnerAdapter, ...

## Methods to implement

- getCount()
- getItem(int position)
- getItemId(int position)
- getViewType(int position)
- getViewTypeCount()
- **getView(int position, View convertView, ViewGroup parent)**

## Adapter - 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)

### Was has to be done in this method?

- Create the view when “convertView” is null
- Set data to the given view

### How can multiple types of views be implemented?

- Use getViewTypeCount() to define how many different View-Types you have
- Use getViewType() to tell android which View-Type the current item should be

## Adapter - Create Views in getView(...)

### How can views defined as XML-Resource be instantiated?

- 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);
```

## 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
3. Bei jedem zweiten Eintrag soll die ID auf der rechten Seite angezeigt werden.
4. Warum muss bei `idTextView.setText(person.getId() + "")` die entsprechende ID in einen String umgewandelt werden?

## Projekt: Exercise\_AdapterView

