

CoolDroid

My First Android Project



Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs

 Phone and Tablet

Minimum SDK API 18: Android 4.3 (Jelly Bean)

Lower API levels target more devices, but have fewer features available.

By targeting API 18 and later, your app will run on approximately **66.9%** of the devices that are active on the Google Play Store.[Help me choose](#) Wear

Minimum SDK API 21: Android 5.0 (Lollipop)

 TV

Minimum SDK API 21: Android 5.0 (Lollipop)

 Android Auto Glass

Minimum SDK Glass Development Kit Preview

Wir verwenden ein Minimum SDK, welches schon die neuen APIs unterstützt.

Cancel

Previous

Next

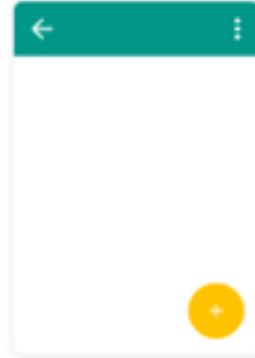
Finish



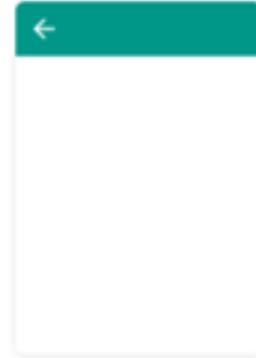
Add an activity to Mobile



Add No Activity



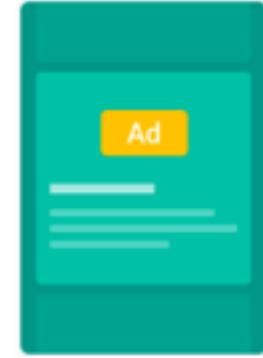
Blank Activity



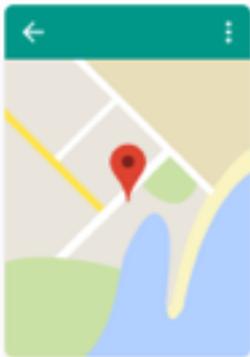
Empty Activity



Fullscreen Activity



Google AdMob Ads Activity



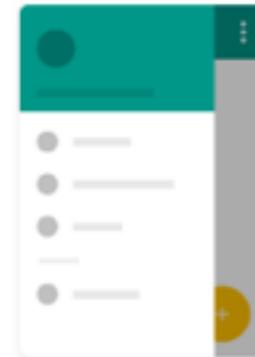
Google Maps Activity



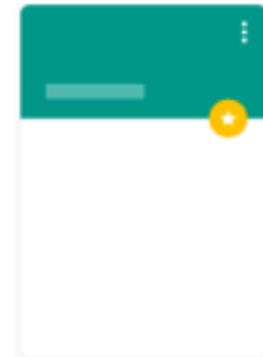
Login Activity



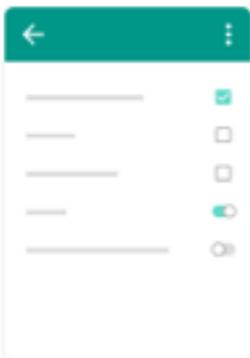
Master/Detail Flow



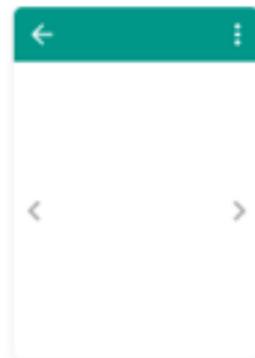
Navigation Drawer Activity



Scrolling Activity



Settings Activity



Tabbed Activity

Cancel

Previous

Next

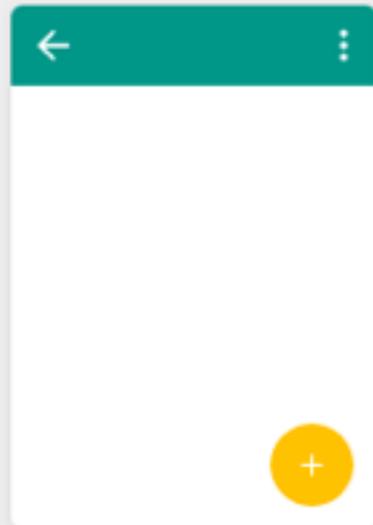
Finish



Customize the Activity



Creates a new blank activity with an app bar.



Blank Activity

Activity Name:

Layout Name:

Title:

Menu Resource Name:

Use a Fragment

The name of the activity class to create

Cancel

Previous

Next

Finish

Palette

Nexus 4 NoActionBar MainActivity 23

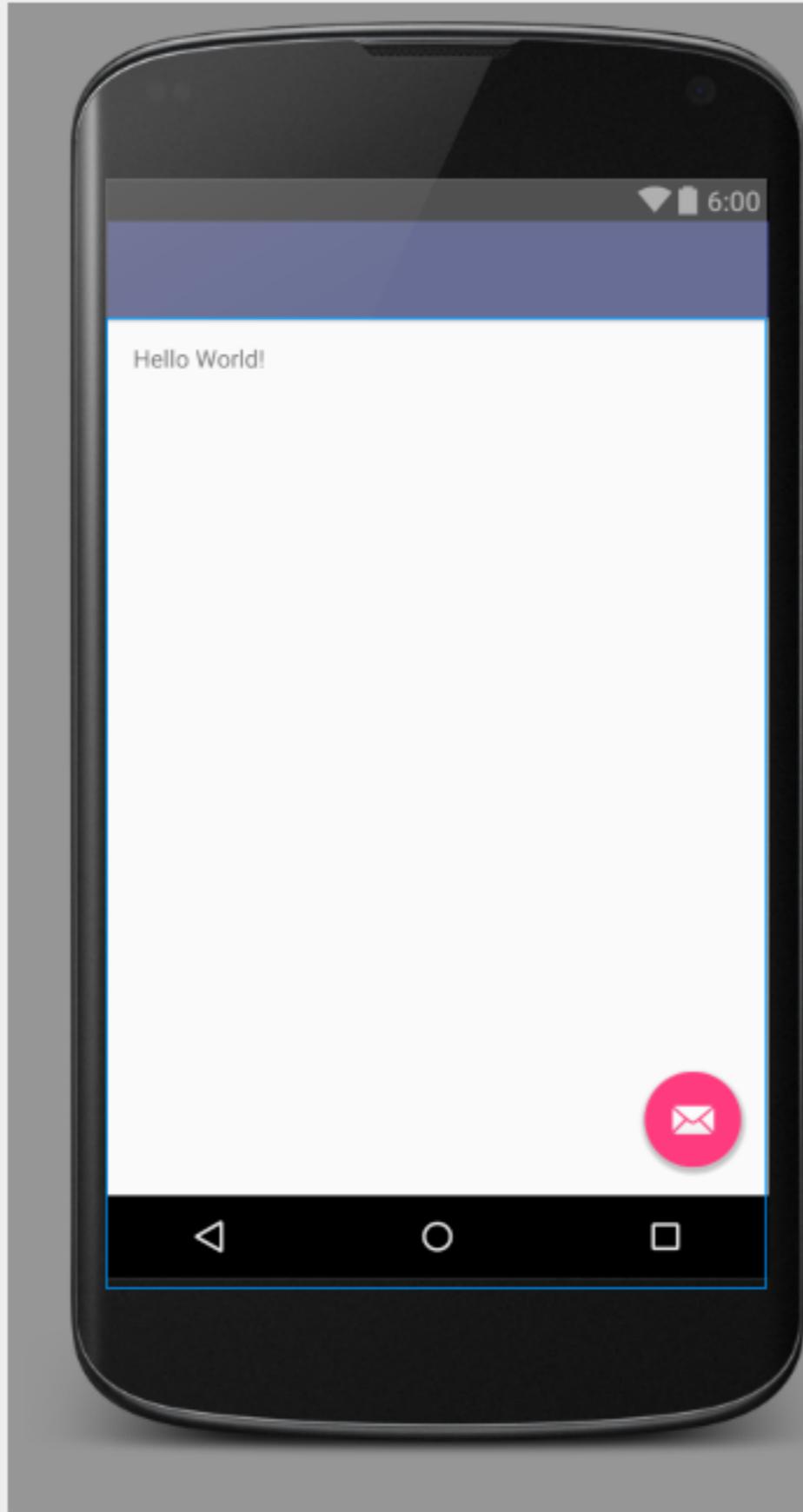
Component Tree

- Shown in @layout/activity_main
 - RelativeLayout
 - TextView - "Hello World!"

Wir ersetzen dieses Layout durch ein LinearLayout

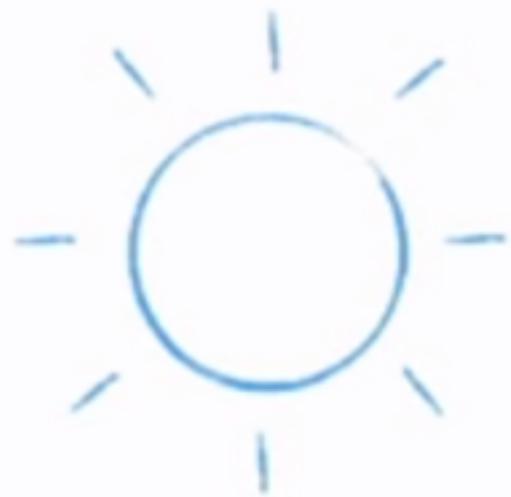
Properties

layout:width	match_parent
layout:height	match_parent
style	
accessibilityLiveRegion	
accessibilityTraversal	
accessibilityTraversal	
alpha	
background	
backgroundTint	
backgroundTintMode	
clickable	<input type="checkbox"/>
contentDescription	
contextClickable	<input type="checkbox"/>
elevation	
focusable	<input type="checkbox"/>
focusableInTouchMod	<input type="checkbox"/>
foreground	



- Layouts**
- FrameLayout
 - LinearLayout (Horizontal)
 - LinearLayout (Vertical)
 - TableLayout
 - TableRow
 - GridLayout
 - RelativeLayout
- Widgets**
- Plain TextView
 - Large Text
 - Medium Text
 - Small Text
 - Button
 - Small Button
 - RadioButton
 - CheckBox
 - Switch
 - ToggleButton
 - ImageButton
 - ImageView
 - ProgressBar (Large)
 - ProgressBar (Normal)
 - ProgressBar (Small)
 - ProgressBar (Horizontal)
 - SeekBar
 - RatingBar
 - Spinner
 - WebView
- Text Fields**
- Plain Text
 - Person Name
 - Password
 - Password (Numeric)
 - E-mail
 - Phone
 - Postal Address

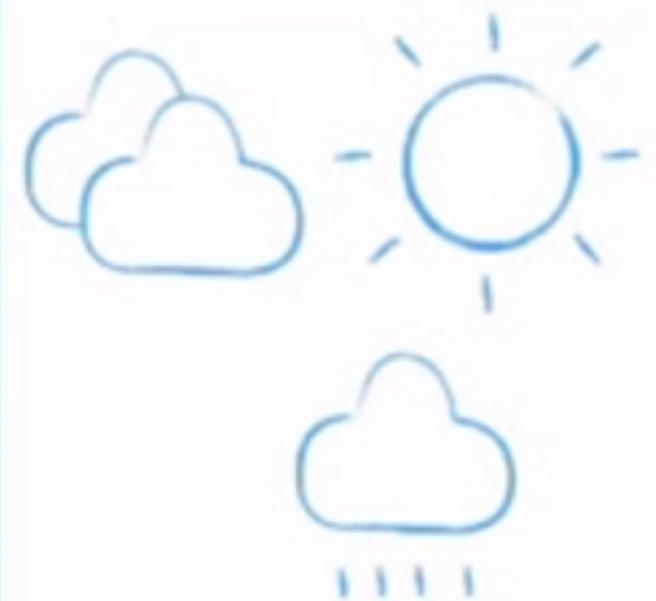
FRAME LAYOUT



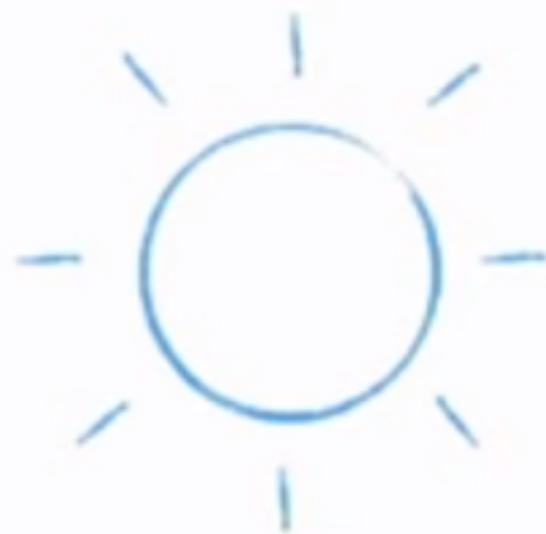
LINEAR LAYOUT



RELATIVE LAYOUT



FRAME LAYOUT



USEFUL FOR SIMPLE LAYOUTS,
WITH A SINGLE VIEW OR STACK
OR VIEWS. VIEWS ARE ALL
ALIGNED AGAINST THE **FRAME**
BOUNDARIES ONLY

LINEAR LAYOUT



PERFECT FOR STACKING VIEWS
VERTICALLY OR HORIZONTALLY,
ONE AFTER ANOTHER.

LINEAR LAYOUT



RELATIVE LAYOUT



SOPHISTICATED LAYOUT THAT
ALLOWS THE POSITIONING OF
VIEWS **RELATIVE** TO OTHER
VIEWS OR THE BOUNDARIES OF
THE VIEW

RELATIVE LAYOUT



RELATIVE LAYOUT

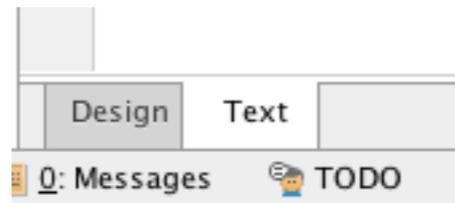


Layout ändern

1

Zuerst TextView und RelativeLayout löschen:

1. auf Text umschalten



2. Bereich markieren und löschen

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context="at.htl_leonding.coolroid.MainActivity"
    tools:showIn="@layout/activity_main">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!" />

</RelativeLayout>
```

2

Anschließend
<LinearLayout ...>
eintragen

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

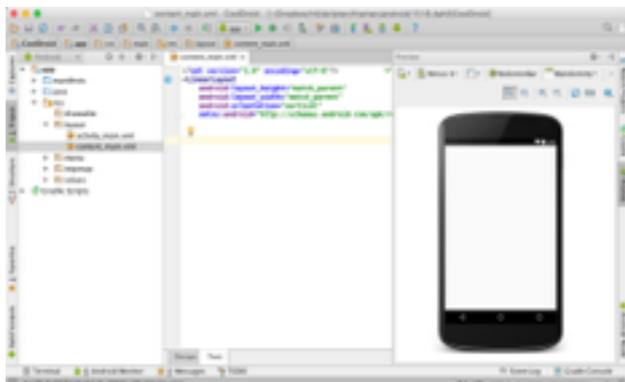


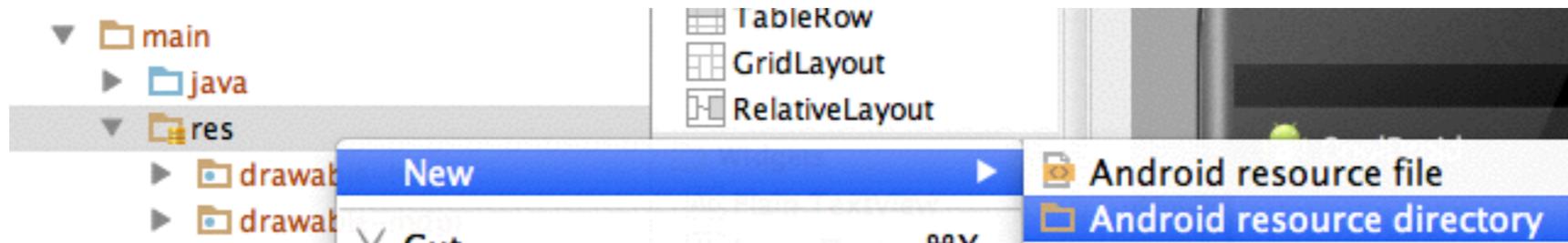
Bild anzeigen

1. Ressourcenverzeichnis anlegen
2. Bild in das Ressourcenverzeichnis kopieren
3. ImageView - Element ins Layout einfügen
4. Ressource an ImageView binden

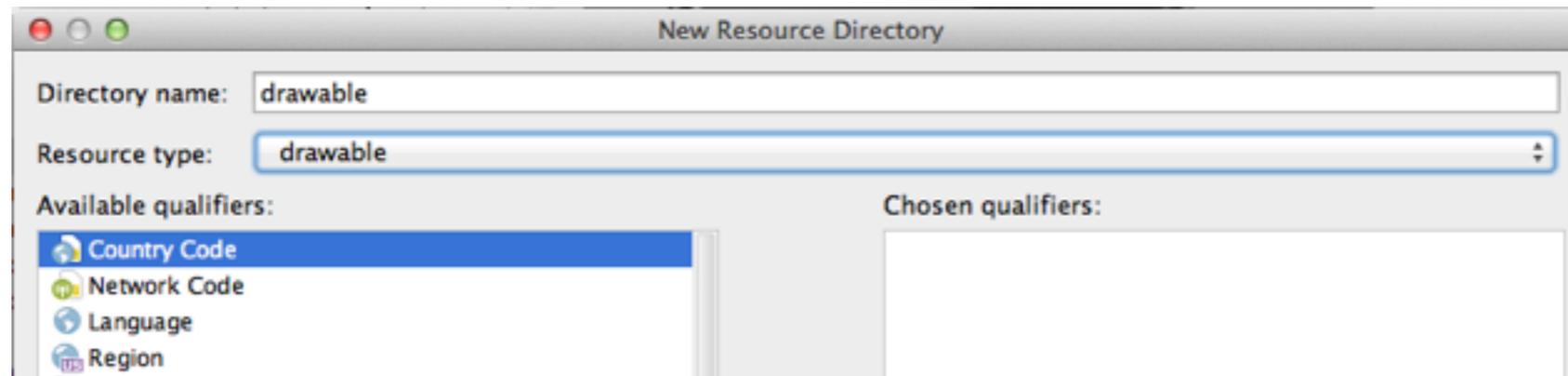


Ressourcenverzeichnis erstellen (falls noch nicht vorhanden)

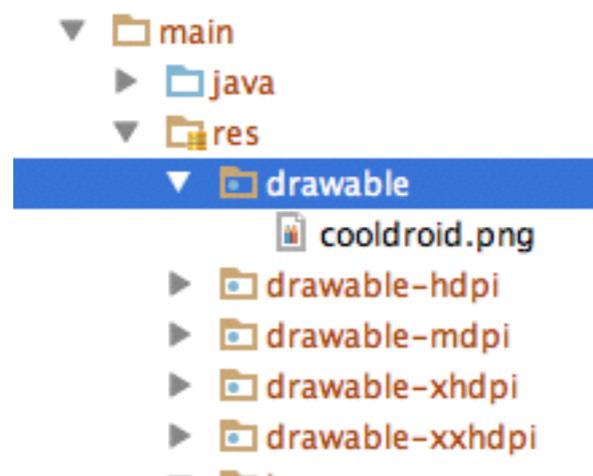
1



2



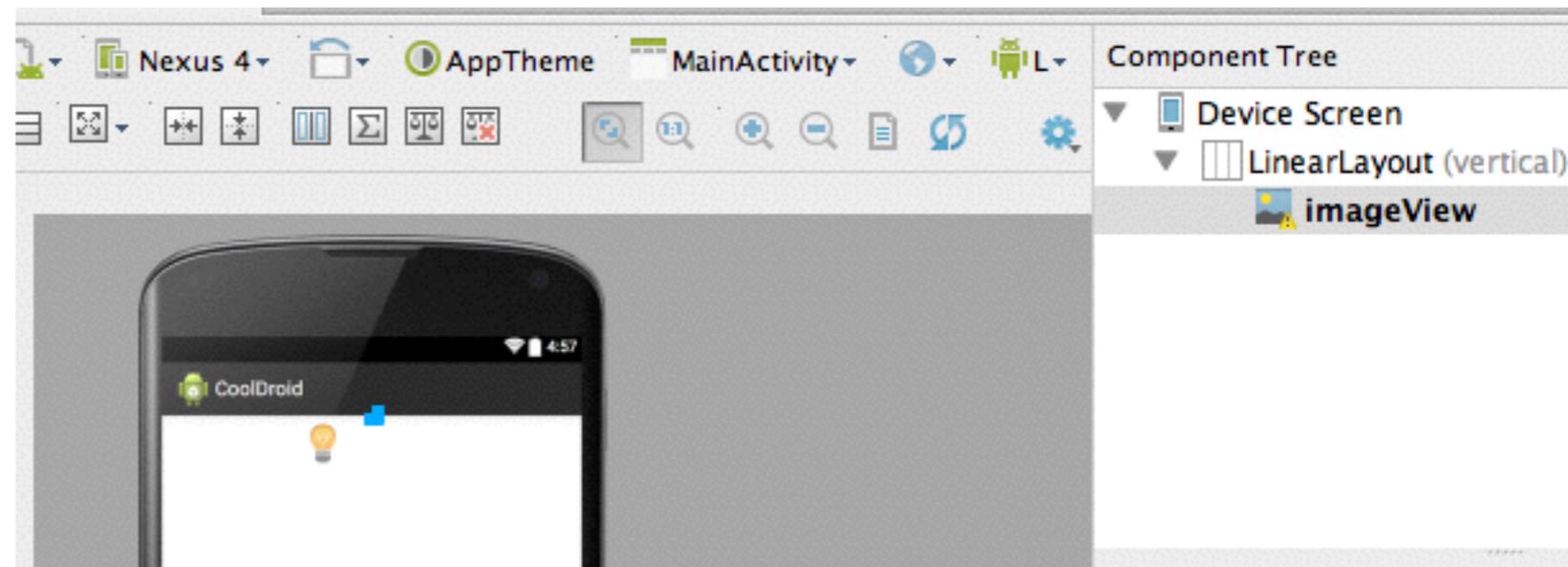
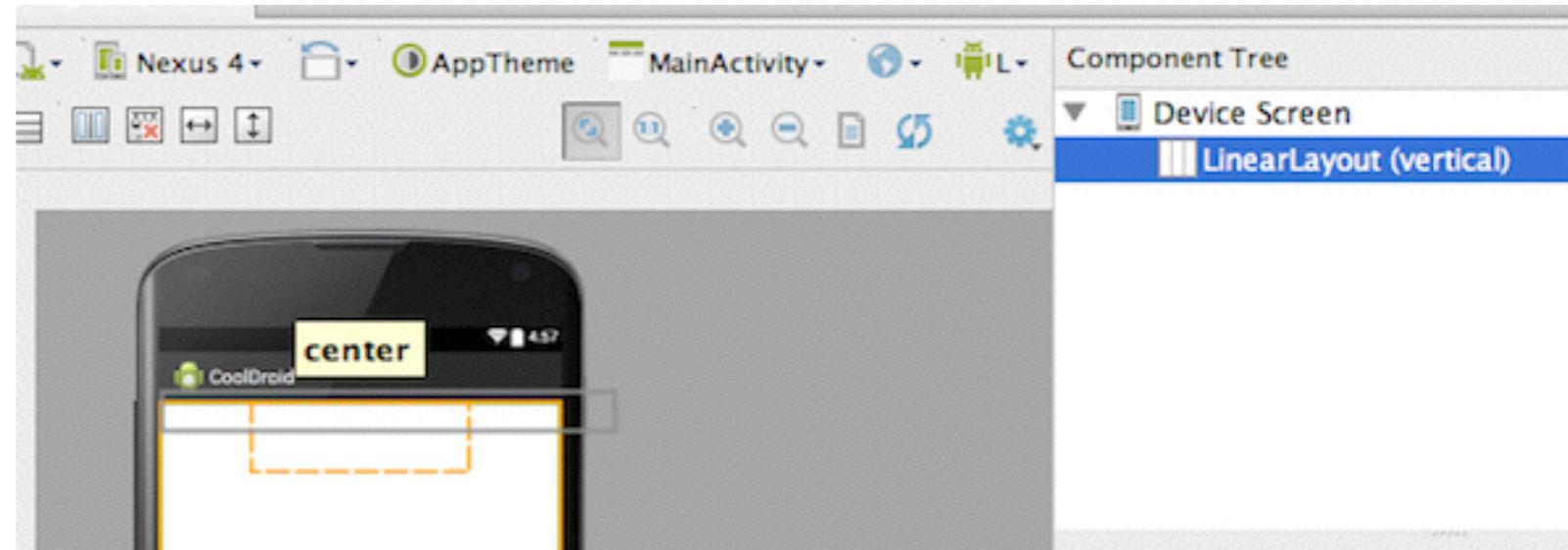
3



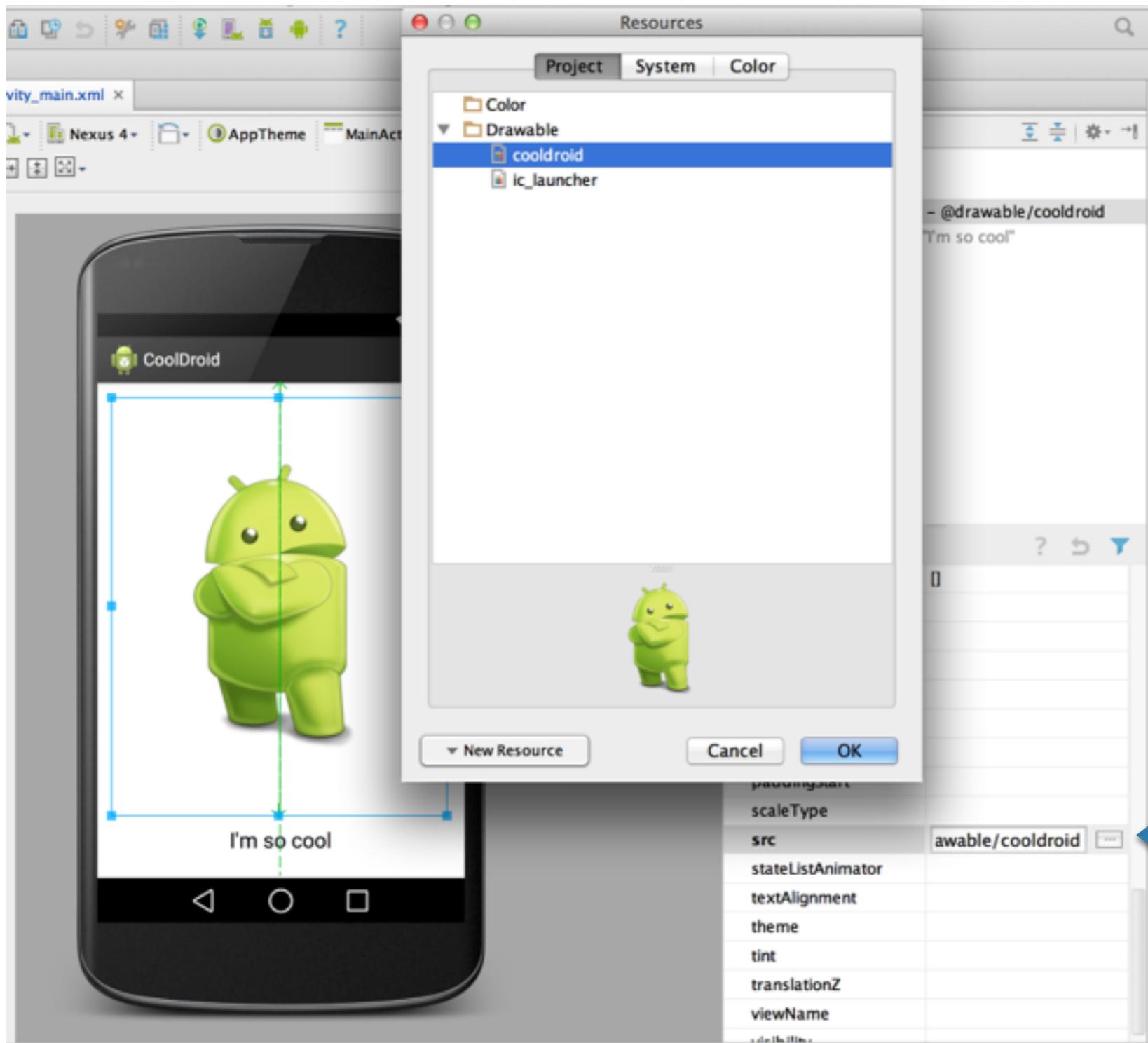
- Kopieren mit Drag'n Drop vom Finder oder Windows Explorer etc. (<Strg> zum Kopieren)

ImageView

- Mit Drag'n Drop das ImageView Element an die gewünschte Position setzen (Center)



Resource an ImageView binden





I'm so cool

1. Verkleinern des Drohten an den Ziehpunkten
2. Einfügen eines "Large Text" Elements darunter

Properties	
onClick	
padding	[10dp, ?, ?, ?, ?]
all	10dp
left	
top	
right	
bottom	
paddingEnd	
paddingStart	
password	<input type="checkbox"/>
phoneNumber	<input type="checkbox"/>
shadowColor	
singleLine	<input type="checkbox"/>
stateListAnimator	
text	I'm so cool

1

2

New String Value Resource

Resource name: textview_welcome_text

Resource value: I'm so cool

File name: strings.xml

Create the resource in directory:

values

values-w820dp

Cancel OK

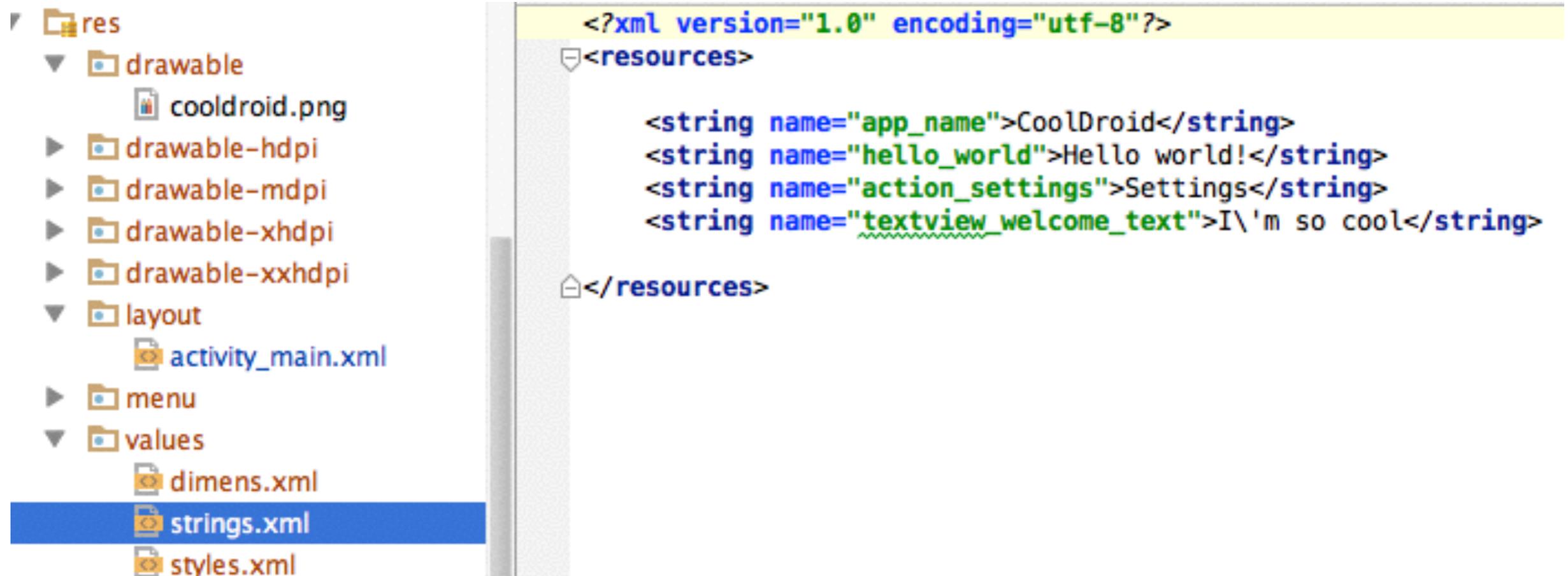
3

Text kann direkt eingetragen werden.
Besser: mit String Ressourcen
Auf ... klicken

stateListAnimator	
text	@string/textview_welcome_text
textAlignment	

4

String Ressourcen

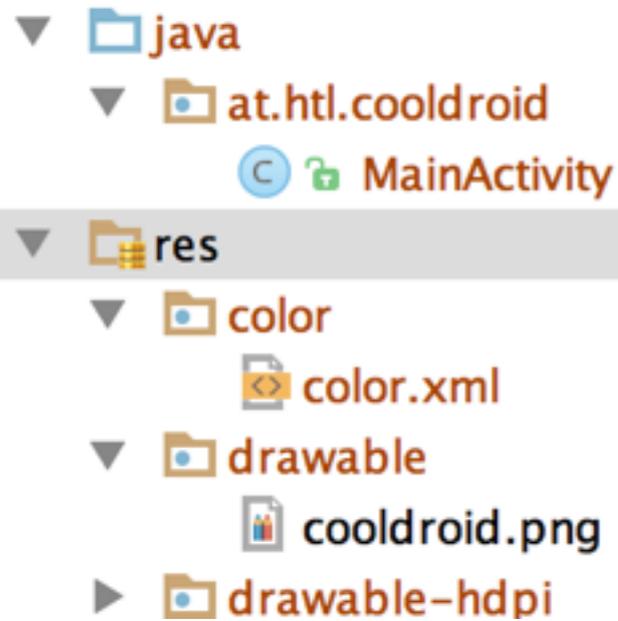


- Können direkt in die xml-Datei eingetragen werden
- Werden zur Übersetzung verwendet zB values-de/strings.xml usw.

Padding beim Textfeld

▼ padding	[10dp, ?, ?, ?, ?]
all	10dp <input type="button" value="..."/>
left	
top	
right	
bottom	

Farb-Ressource anlegen



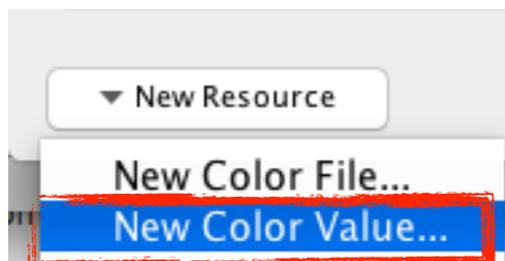
stateListAnimator

text

I'm so cool

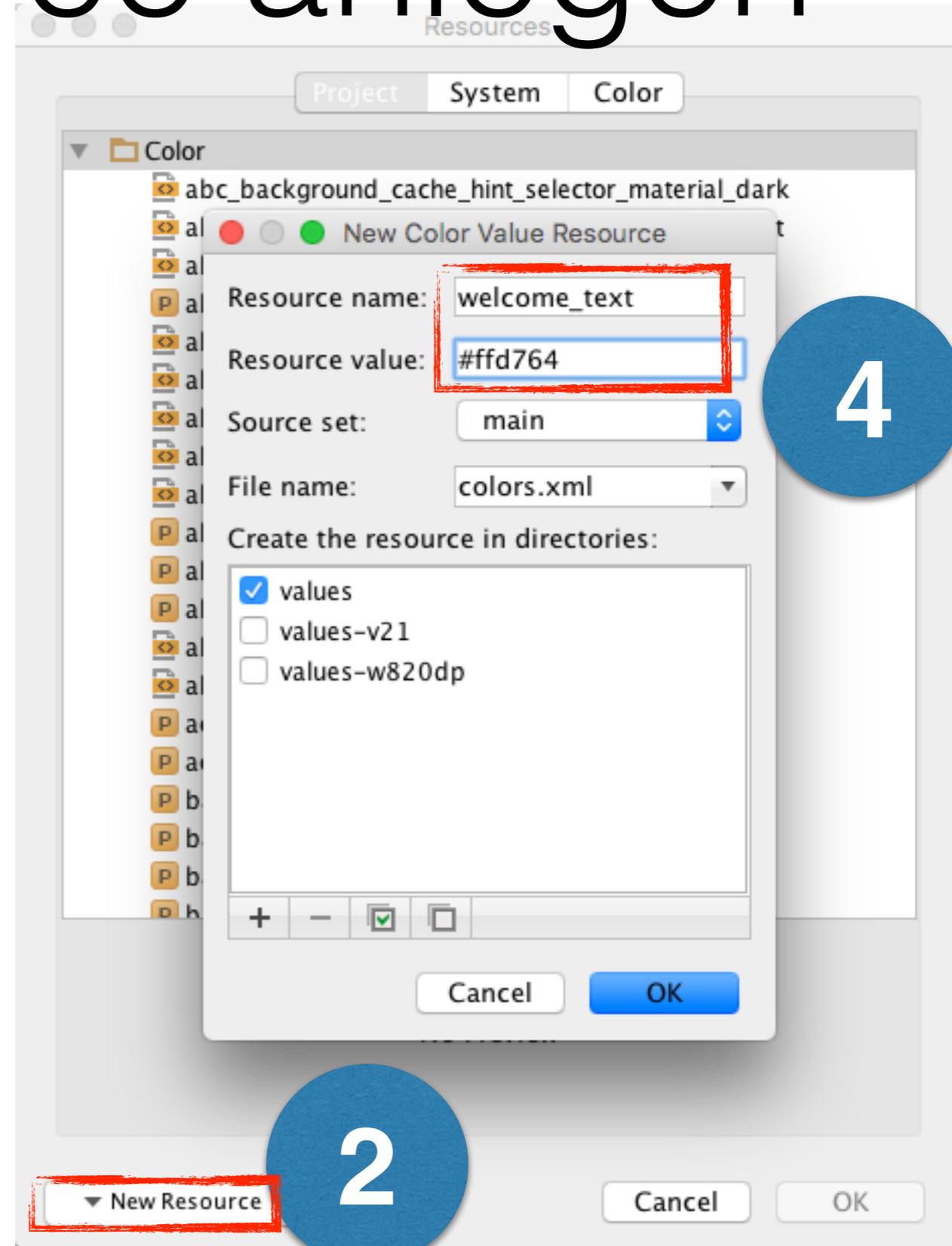


1



3

http://www.w3schools.com/tags/ref_colorpicker.asp



4

2

New Resource

Cancel

OK

The screenshot shows an IDE window with a tab titled 'colors.xml'. The XML content is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <color name="welcome_text">#99ff33</color>
</resources>
```

The line containing the color definition is highlighted in yellow. A 'Choose Color' dialog box is open over the XML editor. The dialog features a color preview bar at the top showing a bright green color. Below the bar, the color is represented in ARGB format with the following values: A: 255, R: 153, G: 255, B: 51. The hex code '#FF99FF33' is displayed in a text field. A large circular color wheel is in the center, with a small square marker indicating the selected color. To the right of the wheel is a vertical slider for adjusting the alpha (opacity) value. At the bottom of the dialog, there is a grid of 18 empty color swatches and two buttons: 'Cancel' and 'Choose'.

Öffnen einer Farbpalette bei Klick auf diese Farb-Preview

activity_main.xml

Hier wird eine neue Id (+) mit Namen ImageView in R.java angelegt.

Obwohl eine String-Resource verwendet wird, wird der korrekte Text angezeigt

```
activity_main.xml x
<LinearLayout
  android:orientation="vertical"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:weightSum="1">

  <ImageView
    android:layout_width="318dp"
    android:layout_height="343dp"
    android:id="@+id/imageView"
    android:layout_gravity="center_horizontal"
    android:src="@drawable/cooldroid" />

  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="I'm so cool"
    android:id="@+id/textView"
    android:layout_gravity="center_horizontal"
    android:padding="10dp"
    android:textColor="@color/welcome_text"
    android:textSize="30dp"/>

</LinearLayout>
```

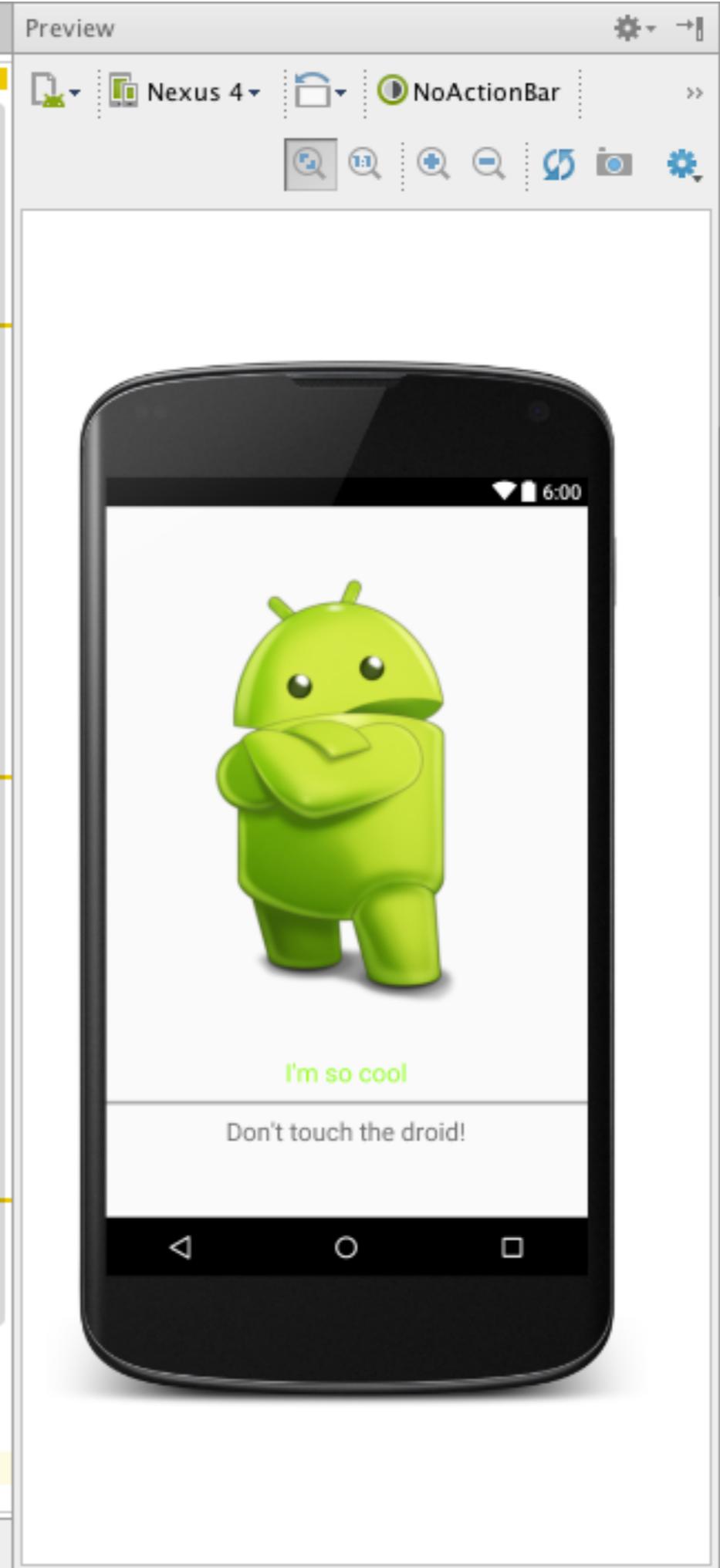
```
content_main.xml x
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_height="match_parent"
  android:layout_width="match_parent"
  android:orientation="vertical"
  android:weightSum="1">

  <ImageView
    android:id="@+id/img_droid"
    android:layout_width="400dp"
    android:layout_height="400dp"
    android:layout_gravity="center_horizontal"
    android:src="@drawable/cooldroid"
    android:layout_marginTop="30dp" />

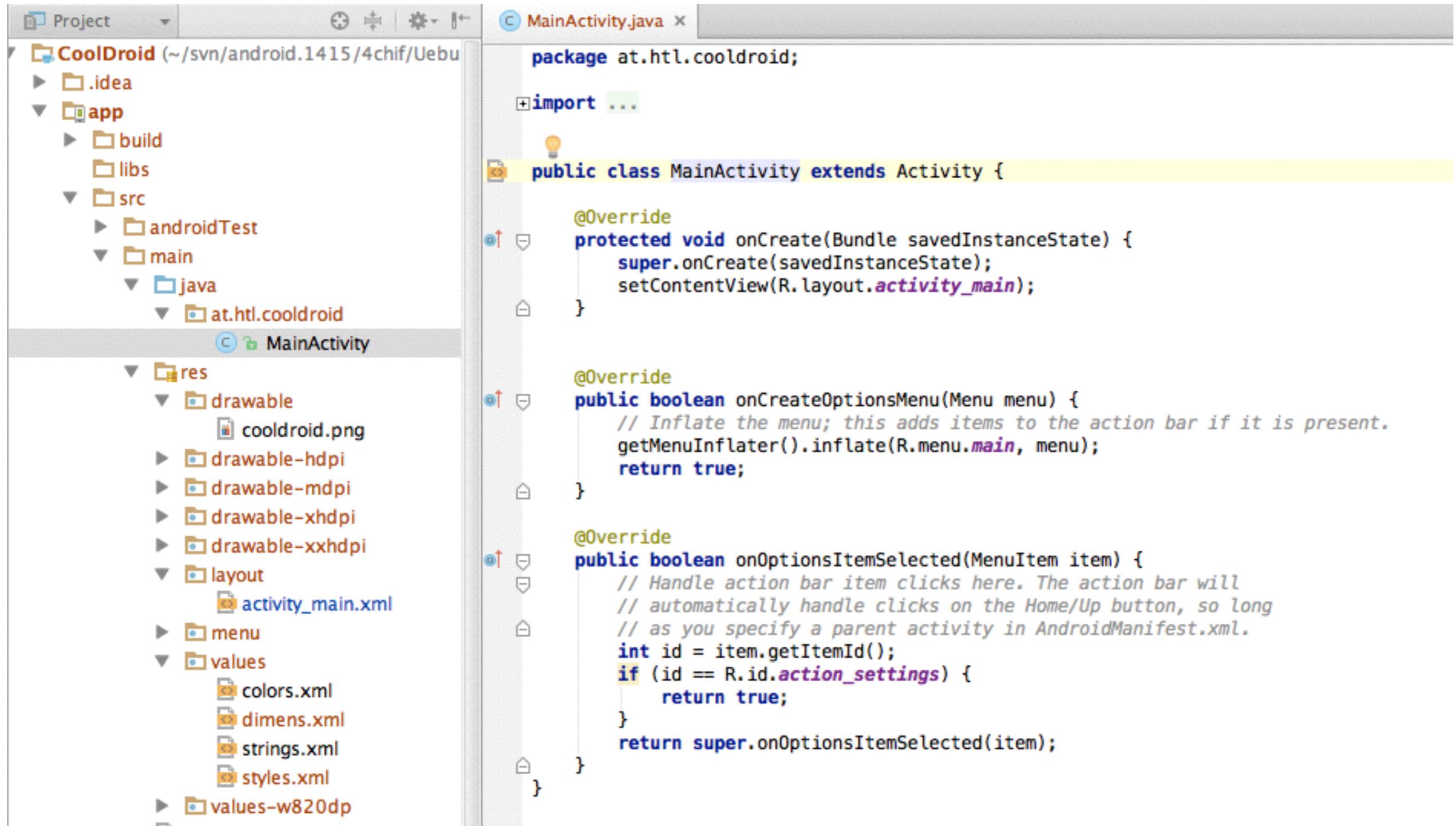
  <TextView
    android:id="@+id/textView_welcome_text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:padding="10dp"
    android:text="@string/textview_welcome_text"
    android:textColor="@color/welcome_text"
    android:textSize="20dp"
    />

  <View
    android:layout_width="match_parent"
    android:layout_height="3dp"
    android:background="@android:color/darker_gray"
    />

  <TextView
    android:id="@+id/textview_droid_output_text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:padding="10dp"
    android:text="@string/abc_droid_output_text"
    android:textSize="20dp"
    />
</LinearLayout>
```



Activities



The screenshot displays an IDE interface for an Android project named "CoolDroid". The left sidebar shows the project structure, including folders for ".idea", "app", "build", "libs", "src", "androidTest", "main", "java", "res", "drawable", "layout", "menu", "values", and "values-w820dp". The "MainActivity" class is selected in the "java" folder.

The right pane shows the code for "MainActivity.java":

```
package at.htl.cooldroid;

import ...

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();
        if (id == R.id.action_settings) {
            return true;
        }
        return super.onOptionsItemSelected(item);
    }
}
```

MainActivity 1

```
public class MainActivity extends Activity {

    private TextView textViewMessage;
    private ImageView imageViewDroid;
    private int counter = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        initializeApp();
    }

    private void initializeApp() {
        textViewMessage = (TextView) findViewById(R.id.textview_droid_output_text);
        imageViewDroid = (ImageView) findViewById(R.id.img_droid);

        imageViewDroid.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                touchDroid();
            }
        });
    }
}
```

MainActivity 2

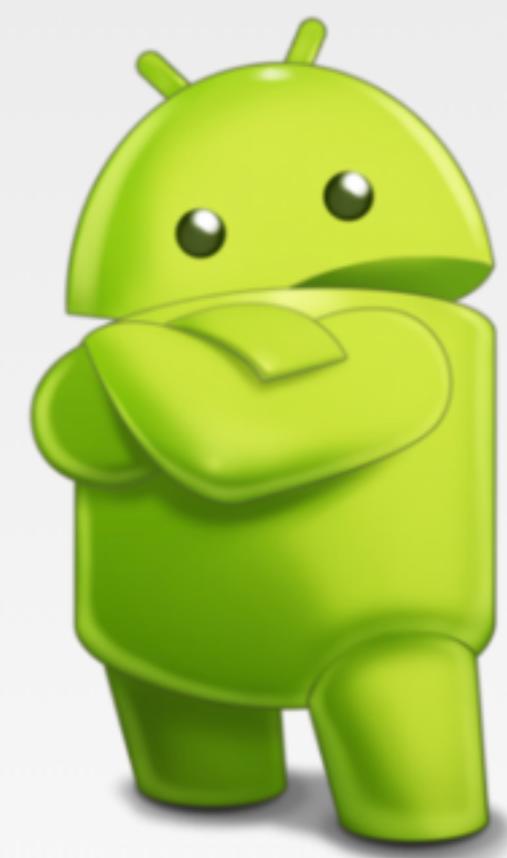
```
private void touchDroid() {  
    counter++;  
    String temp = getStringForDisplay(counter);  
    textViewMessage.setText(String.format("You touched the droid %s", temp));  
}
```

```
private String getStringForDisplay(int count) {  
    String temp;  
    switch (count) {  
        case 1:  
            temp = "once";  
            break;  
        case 2:  
            temp = "twice";  
            break;  
        default:  
            temp = String.format("%d times", count);  
    }  
    return temp;  
}
```



I'm so cool

Don't touch the droid!



I'm so cool

You touched the droid once

HTL Leonding

Schön, hier zu lernen